Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design

Nathalie Bartoli, Thierry Lefebvre, Sylvain Dubreuil, Romain Olivanti, Rémy Priem, Nicolas Bons, Joaquim R.R. A. Martins, Joseph Morlier

Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design

N. Bartoli [a,*], T. Lefebvre [a], S. Dubreuil [a], R. Olivanti [b], R. Priem [a], N. Bons [c], J.R.R.A. Martins [c], J. Morlier [d]

[a] ONERA/DTIS, Université de Toulouse, Toulouse, France
[b] Université de Toulouse, ISAE-SUPAERO, Toulouse, France
[c] University of Michigan, Ann Arbor, MI, 48109, United States
[d] Université de Toulouse, ISAE-SUPAERO-INSA-Mines Albi-UPS, CNRS UMR5312, Institut Clément Ader, Toulouse, France

A B S T R A C T

Surrogate models are often used to reduce the cost of design optimization problems that involve computationally costly models, such as computational fluid dynamics simulations. However, the number of evaluations required by surrogate models usually scales poorly with the number of design variables, and there is a need for both better constraint formulations and multimodal function handling. To address this issue, we developed a surrogate-based gradient-free optimization algorithm that can handle cases where the function evaluations are expensive, the computational budget is limited, the functions are multimodal, and the optimization problem includes nonlinear equality or inequality constraints. The proposed algorithm—super efficient global optimization coupled with mixture of experts (SEGOMOE)—can tackle complex constrained design optimization problems through the use of an enrichment strategy based on a mixture of experts coupled with adaptive surrogate models. The performance of this approach was evaluated for analytic constrained and unconstrained problems, as well as for a multimodal aerodynamic shape optimization problem with 17 design variables and an equality constraint. Our results showed that the method is efficient and that the optimum is much less dependent on the starting point than the conventional gradient-based optimization.

Keywords:
Surrogate modeling
Global optimization
Multimodal optimization
Mixture of experts
Aerodynamic shape optimization
Wing design

1. Introduction

In aerodynamic shape optimization, realistic wing design problems require both a large number of design variables and constraints. A typical problem involves a few tens of design variables and objective and constraint functions that require significant CPU time. Because the computational budget is usually restricted, only a limited number of objective and constraint function evaluations (a few hundreds) can be performed to find the best design. The emergence of adjoint methods [1–3] was a considerable breakthrough in the field of aerodynamic shape optimization because it renders the cost of computing objective and constraint gradients independent of the number of design variables. In conjunction with gradient-based optimizers, this provides an efficient way to converge to a local minimum in a high-dimensional problem. Although airfoil and wing design optimization problems are unimodal [4,5], once the wing planform is allowed to change, the design space becomes multimodal [6]. More generally, in many engineering design optimization problems, we do not know *a priori* whether the design space is multimodal or not and performing a multistart is often intractable in terms of CPU time. When the gradient information is not available, an interesting option is to use gradient-free optimizers known as derivative-free optimization (DFO) methods [7–10]. Stochastic DFO methods mostly fall into the category of evolutionary algorithms and are efficient to solve constrained problems [11]. One of the most promising evolutionary algorithms that can handle constraints is the covariance matrix adaptation evolution strategy (CMA-ES) [12] and its variants [13–15]. They have gained popularity in the solution of high-dimensional problems. Their only drawback is still the large number of function evaluations required to find the optimum [16]. Deterministic DFO methods, such as mesh adaptive direct search (MADS), coordinate search (CS), and generalized pattern search (GPS), have been proposed [10]. These methods have been implemented in different packages such as NOMAD [17], HOPSPACK [18], and the DFL Library [19]. Deterministic DFO methods perform local searches and therefore depend on the initial starting point. For

* Corresponding author.
E-mail address: nathalie.bartoli@onera.fr (N. Bartoli).

https://doi.org/10.1016/j.ast.2019.03.041

solving a multimodal problem, a multistart approach is generally used and the number of function evaluations increases, often with more than a thousand calls. In addition, some DFO algorithms, such as NOMAD, artificially increase the number of inequality constraints to manage equality constraints, making convergence even more difficult. Surrogate-based optimization (SBO) or Bayesian optimization (BO) approaches build an inexpensive approximation of the original function that can be used to rapidly find an approximate optimum [20–23]. In BO, an acquisition function is built from Gaussian process approximations of the objective and constraint functions [24]. For some BO frameworks [17,25], inequality constraints are an issue. Most of the cited algorithms consider equality constraints as two inequality constraints, which becomes intractable for large numbers of constraints. The augmented Lagrangian framework ALBO [26] handles mixed-constrained problems; however, it is still not suitable for solving large-scale problems in a reasonable time. Various reviews on SBO can be found in the literature [27–30].

Another BO alternative is based on sequential enrichment applied to efficient global optimization (EGO) [21] using an adaptive surrogate model. EGO uses a kriging model (also called Gaussian process in the machine learning community [31]) as a substitute for high-fidelity models, taking advantage of the prediction of the variance that is built into these models to inform the adaptive sampling [32–34]. Wessing and Preuss [16] performed a comparison between EGO and CMA-ES for multimodal unconstrained problems. They concluded that EGO is effective in finding multiple optima when the budget of function evaluations is limited. To handle constrained problems, Sasena et al. [35] proposed an extension of EGO called SuperEGO. SBO has been applied to aerodynamic shape optimization problems in previous research efforts [36,37], some of which have used adaptive sampling [38–40]. Despite the works cited above, the number of design variables that have been handled so far is insufficient for wing design optimization and multimodality with a restricted budget is still an issue.

For realistic aircraft wing shape optimization problems, the required number of design variables exceeds 200 [4], and, therefore, trying to directly solve the problems using EGO with a conventional kriging approach is not feasible. We recently proposed to use EGO with a new type of kriging model adapted to high-dimensional problems (namely, KPLS and KPLS+K) [41,42]. The resulting optimization approach, which we call SEGOKPLS(+K), was demonstrated in problems with up to 50 design variables that were solved using approximately a hundred function evaluations [43]. To handle nonlinear functions that vary significantly within a wide domain, researchers have proposed to cluster the data and construct an assembly of local surrogates, known as *mixture of experts* (MOE), which facilitate global optimization [44–48].

The first contribution of this study is the extension of the SEGOKPLS algorithm to handle highly nonlinear and non-smooth functions by using MOE with KPLS or KPLS+K models as the local experts. We have previously presented an approach combining SuperEGO and MOE (SEGOMOE), with preliminary results for analytic functions [49]. We start by constructing surrogate models for the objective and constraint functions by combining automatic clustering and best expert selection. Then, we approach the solution iteratively by balancing the exploration and exploitation phases with a new proposed criterion for the acquisition function. Analytic test cases are presented to compare SEGOMOE with other DFO algorithms such as COBYLA, NOMAD, or ALBO.

The second contribution is the comparison of the proposed approach with a gradient-based algorithm for an aerodynamic shape optimization problem. This problem is based on a benchmark developed by the AIAA Aerodynamic Design and Optimization Discussion Group (ADODG) [50]. The case that we solve is a simplified version of ADODG Case 6, which is a subsonic wing design problem with a multimodal objective. The aerodynamic model is expensive, requiring high-performance parallel computing resources to solve the Reynolds-averaged Navier–Stokes equations using computational fluid dynamics (CFD). The objective is to minimize the drag coefficient at a given lift coefficient. The design variables are the angle of attack, twist distribution, and dihedral distribution of the wing, for a total of 17 design variables. Multiple optima have been identified by using a gradient-based optimizer starting from different design points. The goal of this study was to compare the effectiveness of SEGOMOE versus an established gradient-based approach in converging to the global optimum.

We start the rest of this paper by summarizing the previously developed methods that are relevant to the present work and by introducing a new enrichment criterion (i.e., WB2S). We demonstrate the ability of the proposed approach to find the global optima for five analytic test cases that are multimodal. Then, we demonstrate the effectiveness of SEGOMOE in finding the global optimum of the aerodynamic shape optimization problem and discuss how it compares to a gradient-based method.

2. The SEGOMOE approach

In this section, we describe the proposed approach (SEGOMOE), which solves the constrained optimization problem

$$\begin{cases} \min\limits_{\boldsymbol{x}\in\Omega} \quad y(\boldsymbol{x}) \\ \text{s.t.} \quad c_1(\boldsymbol{x}) \le 0, \ldots, c_m(\boldsymbol{x}) \le 0, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ defines the design space. Equality constraints can be considered a particular case where $c_i(x) = 0$.

We start this section with a description of previously developed techniques (Section 2.1) and follow with a description of the novel contributions in the present work (Section 2.2).

2.1. Background on SEGO

The approach proposed in this paper builds upon SEGO, which, in turn, is based on EGO. Therefore, we start with an overview of EGO and follow with an explanation of how constraints are handled in SEGO. We also include a description of a more local infill criterion (WB2). Finally, we describe the techniques that we use to handle high-dimensional design spaces efficiently (namely, KPLS and KPLS+K).

2.1.1. Original EGO

As mentioned in the Introduction, the proposed algorithm was inspired by EGO [21]. The main idea of the EGO approach is to assume that the unknown objective function is a realization of a Gaussian process (GP) (also known as kriging [31,51]). This requires an initial design of experiments (DOE) of n_{DOE} points $\boldsymbol{X} = \left\{\boldsymbol{x}^{(i)},\ i = 1, \ldots, n_{\text{DOE}}\right\}$ with $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, where the objective function is evaluated to obtain the samples $\boldsymbol{y} = \left\{y(\boldsymbol{x}^{(i)}),\ i = 1, \ldots, n_{\text{DOE}}\right\}$ with $y(\boldsymbol{x}^{(i)}) \in \mathbb{R}$. Then, we can build a conditioned GP that approximates the objective function at any point $\boldsymbol{x}$ by a Gaussian random variable $\hat{Y}(\boldsymbol{x})$ with a mean of

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}_{\boldsymbol{x}\boldsymbol{X}}^t \boldsymbol{R}^{-1} \left(\boldsymbol{y} - \boldsymbol{1}\hat{\mu}\right) \tag{2}$$

and a standard deviation of

$$\hat{s}^2(\boldsymbol{x}) = \hat{\sigma}^2 \left(\boldsymbol{1} - \boldsymbol{r}_{\boldsymbol{x}\boldsymbol{X}}^t \boldsymbol{R}^{-1} \boldsymbol{r}_{\boldsymbol{x}\boldsymbol{X}}\right), \tag{3}$$

where $\boldsymbol{1}$ is an $n_{\text{DOE}} \times 1$ column vector of 1's, $\boldsymbol{r}_{\boldsymbol{x}\boldsymbol{X}} = \{k(\boldsymbol{x}, \boldsymbol{x}^{(i)}),\ i = 1, \ldots, n_{\text{DOE}}\}$, $\boldsymbol{R}$ is the covariance matrix of components $\boldsymbol{R}_{ij} = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$, and $k(.,.)$ is a given covariance function. In this work, we use the squared exponential covariance function

$$k(\boldsymbol{x},\boldsymbol{x}') = \hat{\sigma}^2 \prod_{i=1}^{d} \exp\left(-\theta_i \left(\boldsymbol{x}_i - \boldsymbol{x}'_i\right)^2\right)$$
$$\forall \theta_i \in \mathbb{R}^+, \forall i \in [1,\ldots,d]. \tag{4}$$

The scalar parameters $\hat{\mu}$, $\hat{\sigma}$, and θ_i, $i = 1,\ldots,d$, can be found using a number of methods; here, we use likelihood maximization [22,31]. Using this Gaussian approximation, the EGO algorithm initially proposed by Jones et al. [21] iteratively adds points to the DOE to increase the accuracy of the GP. To find new points $\boldsymbol{x}^{\text{new}}$ that help the overall optimization, the EGO algorithm uses the *expected improvement* (EI) criterion,

$$\text{EI}(\boldsymbol{x}) = \mathbb{E}\left[\max(0, y_{\min} - \hat{Y}(\boldsymbol{x}))\right], \tag{5}$$

where $y_{\min}$ is the minimum value of the objective function over the DOE, i.e., $y_{\min} = \min_{i\in[1,\ldots,n_{\text{DOE}}]} y(\boldsymbol{x}^{(i)})$. Because $\hat{Y}(\boldsymbol{x})$ is a Gaussian random variable defined by its mean (2) and its variance (3), the EI criterion can be written as

$$\text{EI}(\boldsymbol{x}) = \begin{cases} \left(y_{\min} - \hat{y}(\boldsymbol{x})\right)\Phi\left(\frac{y_{\min}-\hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x})\phi\left(\frac{y_{\min}-\hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right), \\ \quad \text{if } \hat{s} > 0 \\ 0, \quad \text{if } \hat{s} = 0, \end{cases} \tag{6}$$

where $\phi(.)$ is the probability density function and $\Phi(.)$ is the cumulative distribution function of the standard normal distribution. This expression highlights the trade-off between *exploitation* of the Gaussian surrogate model and *exploration* of the design space. If $\Phi(.)$ is large when $\hat{y}(\boldsymbol{x})$ is small compared to $y_{\min}$, then the EI criterion promotes exploitation. On the other hand, if $\phi(.)$ is large when $\hat{s}(\boldsymbol{x})$ is large, then it promotes exploration.

Each iteration of the EGO algorithm consists of three main steps:

1. Construct a conditioned GP defined by a mean (2) and the variance (3) based on a DOE of size n_{DOE}.
2. Maximize the expected improvement (6).
3. Add a new point to the DOE ($\boldsymbol{x}^{\text{new}}$) and evaluate it ($y^{\text{new}}$).

This iterative process is repeated from an initial DOE until convergence. The usual stopping criterion is the maximum number of function evaluations corresponding to a computational budget.

2.1.2. Handling constraints

The original EGO algorithm outlined above was designed to minimize unconstrained functions. Because most engineering design problems are subject to constraints, it is crucial to have a sound strategy to handle design constraints. Although we can always add constraint functions to the objective as penalties, this approach is inefficient and inaccurate. To address this issue, Sasena et al. [52] proposed an approach they called super-efficient global optimization (SEGO), which can solve general nonlinearly mixed constrained problems. As in the EGO approach, the objective function y is approximated by a GP. To consider the m nonlinear constraints c_i, $i = 1,\cdots,m$, we construct a surrogate model for each constraint c_i, denoted by $\hat{c}_i$, usually based on the same DOE as that used for the objective function. This results in the following constrained optimization problem:

$$\max_{\boldsymbol{x}\in\Omega_f} \quad \text{EI}(\boldsymbol{x}), \tag{7}$$

where the feasible domain Ω_f is defined by the nonlinear constraints, i.e.,

$$\Omega_f := \{x \in \mathbb{R}^d : \hat{c}_1(\boldsymbol{x}) \le 0, \ldots, \hat{c}_m(\boldsymbol{x}) \le 0\}, \tag{8}$$

where equality constraints, $\hat{c}_i(x) = 0$, could replace or be added to the set of inequality constraints above. At each iteration, the point $\boldsymbol{x}^{\text{new}}$ that solves this optimization problem is added to the DOE and the process is repeated until the convergence. Even if a point $\boldsymbol{x}^{\text{new}}$ is not feasible, evaluating the true functions adds information to the DOE.

In this procedure, the iterative construction of the various surrogate models (one for the objective function and m for the constraints) is driven only by the EI of the objective function. Therefore, if the surrogate models of the constraints are not accurate enough, the accuracy of the optimum is compromised.

In some examples, the optimal solution of Eq. (7) cannot satisfy the "true" constraint functions $c_i(\boldsymbol{x})$ because only the mean value of the GP $\hat{c}_i(\boldsymbol{x})$ is used to approximate the constraints during the optimization process. To consider the associated error estimation $\hat{s}^2_{c_i}(\boldsymbol{x})$, we should implement different strategies. For handling constraints in a Bayesian optimization framework considering the error estimation, various approaches are possible: probabilistic [53], expected violation [54], predictive entropy search with constraints [55], or slack variables with Lagrangian formulation to handle mixed-constraint problems [26]. Work is still in progress when it comes to addressing this challenge [56].

2.1.3. Infill criterion

The optimization problem (7) is multimodal, and it is challenging to find the global maximum without incurring a large computational cost. To improve the efficiency of this optimization, Sasena et al. [52] recommended to use the following criterion named "locating the regional extreme" proposed by Watson and Barnes [57]:

$$\text{WB2}(\boldsymbol{x}) = -\hat{y}(\boldsymbol{x}) + \text{EI}(\boldsymbol{x}), \tag{9}$$

where the mean value of the Gaussian surrogate is subtracted from the EI. The result is that this criterion (WB2) is less multimodal than the EI, which eases the solution of the global optimization, as demonstrated by Sasena [58].

Nonetheless, the WB2 criterion can, in certain cases, prevent the algorithm from finding the global minimum, yielding a local minimum instead. In fact, the magnitude of the term $\text{EI}(\boldsymbol{x})$ is expected to decrease during the iterative process, as the GP surrogate model becomes more accurate in the promising areas of the design space. Thus, after a few iterations, the WB2 criterion is only driven by the mean value of the GP surrogate model, which makes the algorithm focus only on the *exploitation* of the surrogate model rather than on the *exploration* of the design space. To address these issues, we developed a new criterion that builds on WB2, which is described in Section 2.2.1.

2.1.4. Handling a large number of design variables

One of the issues with SEGO is the scalability with the number of design variables d. To construct the GP of the objective and the constraint functions, we have to estimate the hyperparameters θ_i, $i = 1,\ldots,d$ of the covariance function (4). The estimation of these hyperparameters by maximization of the likelihood function can be time-consuming, especially when the number of design variables is high ($d > 10$). This is because the likelihood function is multimodal, requiring a large number of inversions of the correlation matrix to maximize it.

A recently developed surrogate technique, kriging with partial least squares (KPLS), was proposed by Bouhlel et al. [41] to handle the large number of variables for Gaussian processes (up to 100 variables). This technique uses the partial least squares (PLS) method to reduce the number of hyperparameters and the size of the estimation problem. Bouhlel et al. [42] subsequently improved KPLS by adding a new step in the construction of the surrogate

model, which improves the accuracy for high-dimensional problems (KPLS+K). These approaches were used within the SEGO algorithm, and they were demonstrated to be efficient for problems with up to 50 design variables [43].

In this study, we use KPLS and KPLS+K to model both the objective function and the constraints. The use of the KPLS(+K) method in the SEGO algorithm is what enables us to solve nonlinearly constrained optimization problems with a large number of design variables ($d > 10$).

2.2. SEGOMOE and its improvements

In this section, we describe the improvements we made to SEGOMOE that constitute the contributions of the present study. These contributions include a new enrichment technique that improves the overall performance of the optimization, and an MOE approach that improves the accuracy of the surrogate model over a wide range in the design space.

2.2.1. New infill criterion

As discussed in Section 2.1.3, the WB2 criterion (9) improves some of the issues of EI; however, the lack of scaling between the EI term and the prediction of the model can compromise the exploration properties of the method. Therefore, we add the scaling

$$\mathrm{WB2S}(\boldsymbol{x}) = s\,\mathrm{EI}(\boldsymbol{x}) - \hat{y}(\boldsymbol{x}), \tag{10}$$

where s is a non-negative scale, EI is given by Eq. (9), and $\hat{y}(\boldsymbol{x})$ is given by Eq. (2).

This new criterion (WB2S) has two objectives:

1. Keep the exploration property of the expected improvement metric over the feasible design space Ω_f.
2. Keep the original properties of the WB2 metric by smoothing it to facilitate optimization.

The second condition is fulfilled as in the original WB2 criterion by penalizing the expected improvement with the term $-\hat{y}(\boldsymbol{x})$ in Eq. (10). The first condition is more difficult to satisfy and we use the following heuristic. Intuitively, this condition can be translated to

$$\arg\max_{\boldsymbol{x}\in\Omega_f} \mathrm{EI}(\boldsymbol{x}) \approx \arg\max_{\boldsymbol{x}\in\Omega_f} \mathrm{WB2S}(\boldsymbol{x}). \tag{11}$$

To enforce this condition, we need to at least ensure that, for $\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\Omega_f} \mathrm{EI}(\boldsymbol{x})$, the following inequality is verified: $s\mathrm{EI}(\boldsymbol{x}^\star) > \hat{y}(\boldsymbol{x}^\star)$. As a consequence, we define $s = \beta|\hat{y}(\boldsymbol{x}^\star)|/\mathrm{EI}(\boldsymbol{x}^\star)$, where $\beta > 1$. This heuristic approach does not guarantee that Eq. (11) is true because this depends on the variation of $y(\boldsymbol{x}^\star)$ over Ω_f. However, by setting a relatively large value for the parameter β, we expect this heuristic to give an acceptable approximation.

Finally, a new approximation is performed to improve the computational efficiency of the proposed criterion. Finding $\boldsymbol{x}^\star$ over Ω_f is a difficult task, and we prefer to redefine $\boldsymbol{x}^\star$ with the following approximation: $\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\boldsymbol{X}_0} \mathrm{EI}(\boldsymbol{x})$, where the finite subset $\boldsymbol{X}_0 \subset \Omega_f$ contains the starting points used in the multistart optimization of the criterion. The following procedure is used to compute $s > 0$:

1. Compute EI for each starting point in the optimization (in a multistart approach).
2. Evaluate the prediction of the surrogate model for the point with the highest EI.
3. Compute the scale s such that

$$s = \begin{cases} \beta \frac{|\hat{y}(x_{\mathrm{start,EI_{max}}})|}{\mathrm{EI}(x_{\mathrm{start,EI_{max}}})} \text{ if } \mathrm{EI}\left(x_{\mathrm{start,EI_{max}}}\right) \neq 0 \\ 1 \text{ if } \mathrm{EI}\left(x_{\mathrm{start,EI_{max}}}\right) = 0, \end{cases} \tag{12}$$

where $\beta > 1$ is a scaling factor. In our experiments, we have found that a fixed value of $\beta = 100$ works well; however, this could be adjusted during optimization if the EI values are too small compared to the $\hat{y}(\boldsymbol{x})$ values. Some analytic experiments will be presented in Section 3.3 to study the sensitivity analysis of the β parameter.

To illustrate the different behaviors associated with each criterion, we provide a one-dimensional example in Fig. 1. The objective function, its associated GP mean, and the GP uncertainty (with a confidence interval of 99%) are plotted in Fig. 1(a). In Fig. 1(c), the exploration phase is less obvious with the WB2 criterion compared to that with the EI or the WB2S criterion in Figs. 1(b) and 1(d), respectively. The maximum value of WB2 is located close to the minimum of the kriging prediction (see Fig. 1 at $x \approx 0.4$), which illustrates that, in that case, WB2 focuses more on the exploitation of the surrogate model rather than on the exploration. In contrast, the EI and WB2S criteria reach their maximum at $x = 0$, which allows to explore a promising area of the design space. The two advantages of WB2S are illustrated in Fig. 1(d). Similarly to the EI criterion, the WB2S criterion facilitates exploration of the three local maxima. On the other hand, the WB2 criterion is more unimodal, which prevents the optimizer from getting stuck in flat areas in the interval $x \in [0.8, 1]$, where the gradient is zero.

2.2.2. Mixture of experts

One of the main contributions of this study is the combination of MOE with EGO. The motivation for using MOE comes from industrial optimization problems for which both the objective function and the constraints might be strongly nonlinear, discontinuous, or both. In such cases, a good approximation over the whole design space by a kriging model might be inaccurate or require a large-size DOE.

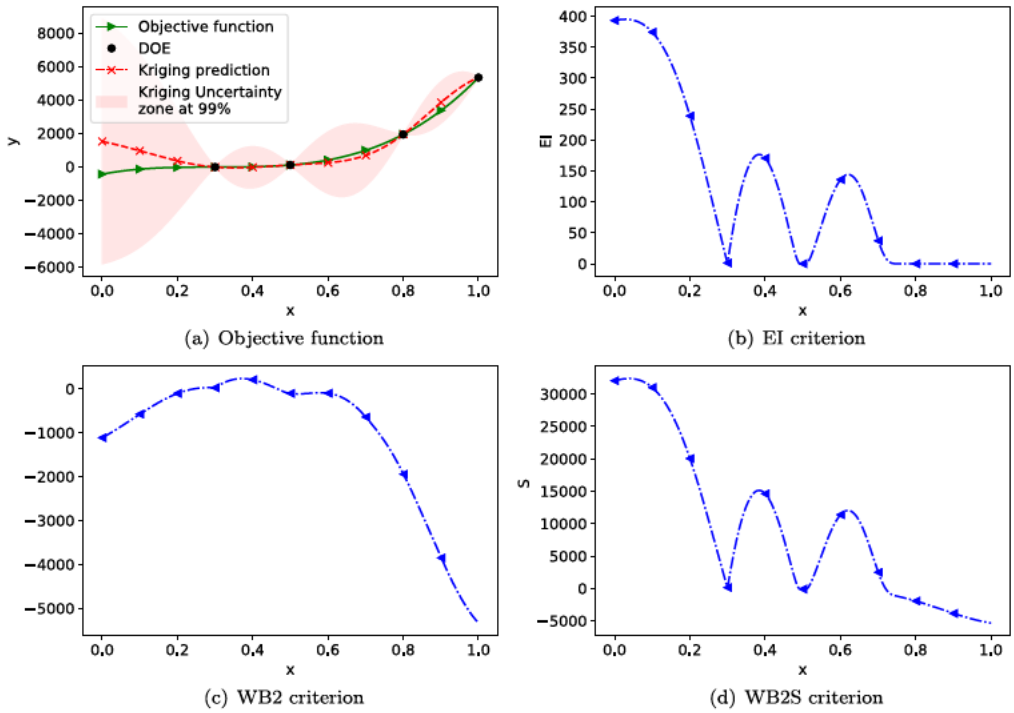
To deal with the approximation of highly nonlinear functions, some researchers have proposed the MOE technique [59,60]. The key idea is to construct different local approximations (experts) for different domains in the design space. These local approximations can be tailored to deal with disparate local trends in the function, including flat regions, discontinuities, and strong nonlinearities.

We use MOE with KPLS(+K) surrogate models as the experts and implemented these methods in the Surrogate Modeling Toolbox [61]. MOE relies on the expectation-maximization algorithm for Gaussian mixture models [62]. The input space is clustered together with its output value by means of a parameter estimation of the joint distribution. A local expert (e.g., polynomial fit, radial basis functions, and kriging) is then built on each cluster, and all the local experts are then combined using the Gaussian mixture model parameters found by the expectation-maximization algorithm to obtain a global model.

In this approach, the Gaussian mixture model is used to combine the data to both partition the input space and derive the mixing proportion. To perform the clustering, we need n inputs, $\boldsymbol{X} = \left\{\boldsymbol{x}^{(i)},\ i = 1, \ldots, n\right\}$, and the corresponding outputs, $\boldsymbol{y} = \left\{y(\boldsymbol{x}^{(i)}),\ i = 1, \ldots, n\right\}$. Therefore, we can only know the cluster posterior probabilities of vectors like $(\boldsymbol{x}^{(i)}, y(\boldsymbol{x}^{(i)})) \in \mathbb{R}^{d+1}$. To predict the cluster posterior probabilities of a sample knowing only its inputs, we must project each multivariate Gaussian function k in the Gaussian mixture model (trained in $d+1$ dimensions) onto the input space, which has d dimensions.

Thus, for each cluster k, we create a multivariate Gaussian function in $(d+1)$-dimensional space with the covariance matrix,

$$\boldsymbol{\Gamma}_k = \begin{pmatrix} \boldsymbol{\Gamma}_k^X & \nu_k \\ \nu_k^T & \xi_k \end{pmatrix}, \tag{13}$$



(a) Objective function (b) EI criterion

(c) WB2 criterion (d) WB2S criterion

Fig. 1. Explicit infill criteria for a one-dimensional example. The objective function in green is approximated by the kriging mean (red curve) from a four-point DOE. The confidence interval of 99% is represented by the pink shaded area. The criterion (EI/WB2/WB2S) is given by the blue curve. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where $\boldsymbol{\Gamma}_k$ is the covariance matrix of $(\boldsymbol{X}, \boldsymbol{y})$, $\boldsymbol{\Gamma}_k^X \in \mathbb{R}^d$ is the covariance matrix of $\boldsymbol{X}$, $\nu_k \in \mathbb{R}$ is $\mathrm{Cov}(\boldsymbol{X}, \boldsymbol{y})$, and $\xi_k \in \mathbb{R}$ is $\mathrm{Var}(\boldsymbol{y}, \boldsymbol{y})$. The mean vector of the multivariate Gaussian function is

$$\boldsymbol{\mu}_k = \begin{pmatrix} \boldsymbol{\mu}_k^X \\ \mu_k^y \end{pmatrix}, \tag{14}$$

where $\boldsymbol{\mu}_k^X$ represents the $\boldsymbol{x}$-coordinates of the mean $\boldsymbol{\mu}_k$ and μ_k^y is the y-coordinates of the mean.

The clustering is based on the spatial location and function values (supervised approach between the inputs $\boldsymbol{X}$ and the output $\boldsymbol{y}$) as in the initial version of Bettebghor et al. [45]. If available, some other clustering criteria could be used, such as a derivative value (supervised approach between the inputs $\boldsymbol{X}$ and the outputs $\partial \boldsymbol{y}/\partial \boldsymbol{x}$) to obtain a better indicator for the heterogeneity in the function, as proposed by Liem et al. [60]. In our MOE toolbox, the user can provide the full or partial Jacobian as a criterion for the MOE clustering.

Thanks to hyperplane projection and linear recombination, the posterior probabilities of each cluster can be predicted and local models can be performed. When local models $\hat{f}_i$ are known, the global model is

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{K} \mathbb{P}(\kappa = i | X = \boldsymbol{x}) \hat{f}_i(\boldsymbol{x}), \tag{15}$$

which is the classical probability expression for MOE. In the equation above, K is the number of Gaussian components, $\mathbb{P}(\kappa = i | X = \boldsymbol{x})$ is the probability that X lies in cluster i knowing that $X = \boldsymbol{x}$, and $\hat{f}_i$ is the local expert built on cluster i. This probability is called the *gating network*.

2.2.3. Recombination of the infill criterion with the mixture of experts

Another key contribution of the proposed approach is the strategy used to recombine the local experts and compute the infill criterion. In MOE, the Gaussian mixture model is used to both partition the input space and derive the mixing proportion.

Based on Eq. (15), the global model can be rewritten as

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{K} \alpha_i \hat{f}_i(\boldsymbol{x}), \tag{16}$$

where K is the number of Gaussian components. This number is chosen automatically to minimize the generalization error on a validation data set [49]. The coefficients α_i are given by

$$\alpha_i = \mathbb{P}(\kappa = i | X = \boldsymbol{x}) \quad \forall i \in 1 \dots K$$

for the smooth recombination

or

$$\alpha_i = \left\{ \begin{array}{l} = 1 \text{ if } i = \arg \max_{k \in [1, \cdots, K]} \mathbb{P}(\kappa = k | X = \boldsymbol{x}) \\ = 0 \text{ otherwise} \end{array} \right\}$$

for the hard recombination (17)

Their computation leads to two different approximation models: smooth recombination and discontinuous recombination.

Smooth recombination considers all the cluster posterior probabilities when choosing the Gaussian laws to compute these quantities. The output is given by the sum of the different model outputs weighted by the cluster posterior probabilities. The resulting mixture of experts is also normally distributed, assuming the local experts are independent random variables that are normally distributed (and therefore also jointly so). The uncertainties of these models follow the Gaussian functions. The Jacobian of the surrogate model can be predicted if the analytic Jacobians of the local models are known, as in the case of kriging-based models. Thus, we can perform a gradient-based optimization, thanks to the smoothness of the surrogate model.

Discontinuous recombination—also known as hard recombination—considers the maximum of the cluster posterior probabilities; in which case, $\mathbb{P}(\kappa = i | X = x)$ is computed using characteristic functions for the clusters that are equal to 0 or 1. Thus, only

outputs given by a local model corresponding to the cluster are allowed. This method can be efficient for discontinuous functions. Moreover, the mixture of experts can also predict the uncertainty: for one sample, we choose the uncertainty corresponding to its cluster.

For both recombinations, the mixture of experts based on kriging models approximates functions with a heterogeneous behavior, providing a global model with a prediction of both the Jacobian and the uncertainty. The proposed algorithm automatically chooses the best type of recombination, according to a cross-validation procedure. The MOE can be written as the sum

$$\sum_{i=1}^{K} \alpha_i \mathcal{N}(\hat{y}_i, \hat{s}_i^2) = \mathcal{N}\left(\sum_{i=1}^{K} \alpha_i \hat{y}_i, \sum_{i=1}^{K} \alpha_i^2 \hat{s}_i^2\right) = \mathcal{N}\left(\hat{y}, \hat{s}^2\right), \tag{18}$$

where α_i is defined by Eq. (17), and $\hat{y}_i(\boldsymbol{x})$ and $\hat{s}_i^2(\boldsymbol{x})$ are given by Eqs. (2) and (3), respectively. Thus, the MOE can also predict the uncertainty. This information is useful to define the infill criterion used in the EGO or SEGO algorithm. The EI, WB2, and WB2S criteria defined by Eqs. (6), (9), and (10), respectively, are adapted for the MOE models using

$$\mathrm{EI}_{\mathrm{MOE}}(\boldsymbol{x}) = \begin{cases} \left(y_{\min} - \hat{y}(\boldsymbol{x})\right) \Phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(\boldsymbol{x}) \phi\left(\frac{y_{\min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right), \\ \quad \text{if } \hat{s} > 0 \\ 0, \quad \text{if } \hat{s} = 0, \end{cases} \tag{19}$$

where $\phi(.)$ is the probability density function and $\Phi(.)$ is the cumulative distribution function of the standard normal distribution $\mathcal{N}(0, 1)$. The only difference between Eq. (6) and Eq. (19) is the nature of $\hat{y}$. Because MOE has multiple models, $\hat{y}$ is derived from Eq. (18) and expressed as a combination of kriging-based local experts, i.e.,

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{K} \alpha_i \hat{y}_i(\boldsymbol{x}), \tag{20}$$

where $\hat{y}_i(\boldsymbol{x})$ is given by Eq. (2). The associated variance $\hat{s}^2$ is expressed in the same way, i.e.,

$$\hat{s}^2(\boldsymbol{x}) = \sum_{i=1}^{K} \alpha_i^2 \hat{s}_i^2(\boldsymbol{x}), \tag{21}$$

where $\hat{s}_i^2$ is given by Eq. (3). Because the infill criterion considers only the objective function with the knowledge of the uncertainty estimation (see Eq. (3)), we have a wider choice for the constraint surrogate models. These could be a single surrogate or a mixture of local experts. Analytic derivatives of the infill criterion (EI, WB2, or WB2S) are also available and computed for gradient-based optimization.

2.2.4. The SEGOMOE algorithm

Once recombination is achieved, MOE is used in combination with the SEGO algorithm to solve global optimization problems subject to nonlinear constraints involving a large number of design variables ($d > 10$) and potentially highly nonlinear objective and constraint functions. Figure 2 shows a flowchart of the SEGOMOE algorithm used in this study. The main steps in the algorithm are as follows:

1. Construct the initial DOE and build the associated MOE models for the objective and constraint functions.
2. Maximize the infill criterion (EI, WB2, or the new criterion defined in Section 2.2.1) subject to the design constraints and variable bounds, and propose the new enrichment point.

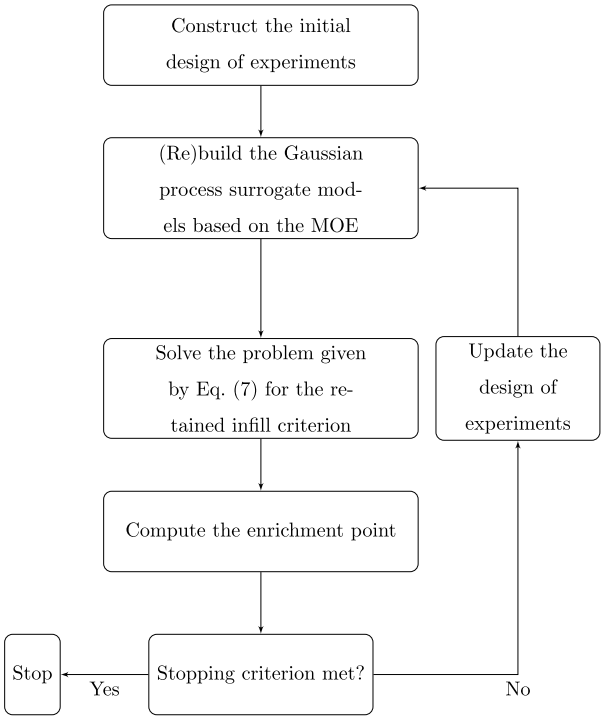


Fig. 2. Overview of the SEGOMOE algorithm.

3. Compute the values of the objective and constraint functions at the new enrichment point.
4. Check if the new enrichment point is in the feasible domain or not, and identify inactive, active, and violated constraints.
5. Return to Step 2 and update the DOE until the stopping criterion is met. A common criterion is the maximum number of function calls corresponding to the available computational budget.

To solve the constrained optimization problem (with inequality and/or equality constraints) in Step 2, we can use a gradient-based algorithm or a derivative-free optimizer. Because we use local optimizers, we perform a multistart optimization with different starting points (10 points by default). The gradient-based optimizers use the analytic Jacobian of the MOE (as described in Section 2.2.3) to compute the gradients of both the objective function and the constraints of the global optimization problem (7).

Regarding the number of clusters involved in the MOE, Bartoli et al. [49] compared the efficiency of SEGOMOE with K clusters versus one cluster on the MOPTA test case [63]. It was found that the number of function evaluations could be decreased by automatically choosing the number of clusters K (see [49] for more details on the proposed strategy) and that, in every test case studied, the use of MOE in the EGO approach never increases the number of function evaluations required for convergence. As a consequence, in the following, the SEGOMOE framework is used with the automatic cluster number approach already presented by Bartoli et al. [49]. The algorithm is implemented within the NASA OpenMDAO framework [64] and, additionally, can use surrogates available within the Surrogate Modeling Toolbox [61].

3. Analytic benchmark problems

This section presents numerical results that demonstrate the ability of SEGOMOE to solve multimodal analytic optimization problems. A more realistic application is presented in Section 4. Here, five multimodal analytic optimization problems (three unconstrained problems and two constrained problems) are con-

sidered. The three unconstrained problems are well known two-dimensional benchmark analytic functions: the six-hump camel-back function, the Michalewicz function, and the Ackley function. The first constrained problem is a two-dimensional analytic problem proposed by Parr et al. [65] that has two design variables and one inequality constraint. The solution consists of one global minimum and two local minima. The second constrained problem is a four-dimensional analytic problem proposed by [26] that has a (known) linear objective and two constraints: one inequality and one equality. Before dealing with the numerical results, we review first the performance criteria based either on the objective value at the optimal point or on the global minimum location and the choice of the β parameter for the WB2S criterion (see Section 2.2.1). We also introduce derivative-free optimizers to compare their performance with that of SEGOMOE. We choose to consider COBYLA, NOMAD, and ALBO, which are three derivative-free optimizers available as an open-source toolbox that can handle nonlinear constraints.

3.1. Performance criteria

We use three performance criteria to compare the results for SEGOMOE with those for the other optimization algorithms (COBYLA, NOMAD, and ALBO).

1. Percentage of converged runs
2. Mean number of function evaluations for converged runs
3. Standard deviation of the number of function evaluations for converged runs.

The convergence is assessed by measuring the error in the objective value or the proximity of the design variables. When the objective value is used, the relative error is

$$\mathrm{RE} = \frac{|f^*_{\mathrm{opt}} - f^*_{\mathrm{ref}}|}{|f^*_{\mathrm{ref}}|}, \tag{22}$$

where f^*_{opt} is the best point given by each algorithm and f^*_{ref} is the known reference solution given in Appendix A. When using the proximity of the design variables, we measure the difference between two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ using

$$\mu_{\mathrm{prox}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = 1 - \frac{1}{d}\sum_{i=1}^{d} \frac{|x_{2_i} - x_{1_i}|}{\mathrm{ub}_i - \mathrm{lb}_i}, \tag{23}$$

where ub_i and lb_i denote the upper bound and the lower bound, respectively, of the ith design variable. The distances are scaled to $[0, 1]$ to confer the same weight on the design variables.

In the test cases presented here, one of the solutions is the reference and the other one is the optimal point provided by the algorithm. To ensure that an optimization is converged, we check that $1 - \mu_{\mathrm{prox}}(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq \epsilon$, where ϵ is a given threshold value (10^{-3} by default). The type of validation and the convergence threshold are specified for each problem. It is crucial to look at this criterion because the other two are only computed for converged runs.

3.2. Derivative-free optimizers: COBYLA, NOMAD, and ALBO

To evaluate the surrogate-based strategies, we consider the other derivative-free algorithms (COBYLA, NOMAD, and ALBO) for the constrained and unconstrained analytic test cases. COBYLA (Constrained Optimization BY Linear Approximation) is a trust region optimization method that uses an approximating linear interpolation model of the objective and constraint functions [66]. NOMAD is a derivative-free algorithm based on a C++ implementation of the mesh adaptive direct search (MADS) algorithm [17], which is designed to solve difficult blackbox optimization problems. NOMAD discretizes the design space into a mesh that refines adaptively to find better successive solutions. According to Audet et al. [67], NOMAD is intended for time-consuming blackbox simulation with a small number of variables. Another BO algorithm, ALBO [26], is used here to handle mixed-constrained problems. ALBO combines an unconstrained BO framework with the classical augmented Lagrangian (AL) framework [68]. Originally designed for the equality constraint problems [69], it has been extended to inequality constraints by means of the slack variables. The ALBO procedure is the same as the AL framework except that the minimization of the AL function is replaced by the maximization of an acquisition function. The new acquisition function is not given explicitly but only through an estimation method. We found that ALBO is not suitable for the solution of large-scale problems in a reasonable time.

We tested COBYLA and NOMAD on three well-known two-dimensional multimodal unconstrained problems and tested ALBO on a four-dimensional mixed-constrained problem to compare the convergence to the global minimum and the number of function evaluations.

Both COBYLA and NOMAD require the user to specify the maximum number of function evaluations. For COBYLA, this is set to 500 evaluations, which is much higher than the number of evaluations required to reach the convergence in the analytic test cases we solve. For NOMAD, we try different values for the maximum number of evaluations keeping in mind that our budget is limited to 1000 calls. If no stopping criterion is specified, NOMAD stops as soon as the mesh size reaches a given tolerance. Because COBYLA and NOMAD require a starting point, it may be necessary to use multistart to find the global optimum of these multimodal functions. To use the same set of starting points between the two algorithms, we generate an initial set of 100 points using Latin hypercube sampling (LHS) [70].

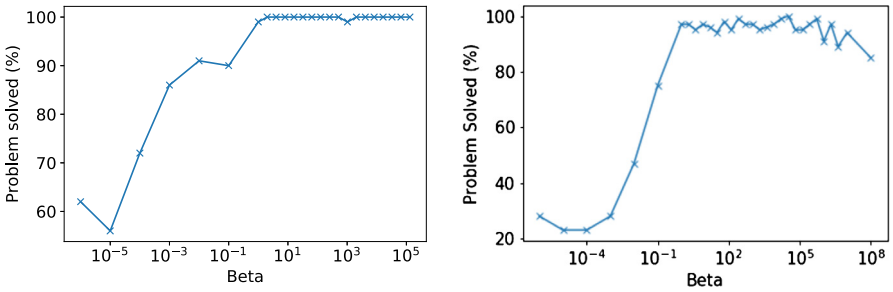
As explained earlier, SEGOMOE uses an LHS strategy to generate the initial DOE to build the GP approximations of the objective and the constraints. Thus, to compare the three infill criteria (EI, WB2, and WB2S), we use the same set of initial LHS-generated points for each criterion with different numbers of sampling points (5, 10, 20, and 30 points). This set of initial LHS-generated points is also used for ALBO, which, like SEGOMOE, requires an initial sample of points to build a GP approximation. The size of the sampling set is also a studied parameter.

We used the Python library Scipy [71] for COBYLA, the Python NOMAD framework [72], and DiceOptim [73] with the default settings for ALBO.

3.3. Sensitivity analysis of the WB2S infill criterion

To find the relevant values for the β parameter in the WB2S infill criterion (defined in Section 2.2.1), we performed a sensitivity analysis study for two analytic functions (Michalewicz and Ackley) defined in Appendices A.2 and A.1. The sensitivity analysis study consisted of 100 runs for 32 values of $\beta \in [10^{-6}, 10^{9}]$, resulting in a total of 3200 runs for each function. An initial DOE of 5 points was used for the two test cases. The percentage of solutions obtained for varying β is plotted in Fig. 3.

The success rate for the Michalewicz problem was 100% for $\beta > 1$, as shown in Fig. 3(a). This was expected because s in Eq. (12) increases with β and the WB2S criterion tends to the EI criterion when β tends to infinity, as can be seen in Eq. (10). On the other hand, for $\beta < 1$, the WB2S criterion tends to the surrogate-based criterion (SBO approach), which minimizes the kriging-based surrogate model without any exploration. Therefore, small values of β lead to poor performance in both problems.



(a) Michalewicz (b) Ackley

Fig. 3. Effect of the value of β in the WB2S infill criterion. One hundred DOE points (initial set with 5 points) were tested with a total of 3200 runs for each function.

For the Ackley function, the percentage of problems solved with WB2S initially increased with increasing β and then tended to decrease for $\beta > 10^5$ (see Fig. 3(b)). As a consequence, one can see that the range of β values leading to a higher success rate is quite large $\beta \in [10, 10^5]$. Thus, even if this range is problem dependent (especially the upper bound, as illustrated by the comparison of Fig. 3(a) and Fig. 3(b)), one might be confident in choosing an intermediate value for β. In the following, $\beta = 100$ is retained.

3.4. Analytic unconstrained multimodal problems

As mentioned above, the main objective of this section is to demonstrate the performance of SEGOMOE for multimodal problems. In particular, we want to compare the performance of the different infill criteria (EI, WB2, and WB2S). We first consider three analytic unconstrained multimodal functions:

- The six-hump camel-back function (defined in Appendix A.3) is a two-dimensional function with six minima—two of which are global—that is smooth and easy to optimize. The convergence is assessed based on the objective value and a threshold of 10^{-3} for the associated relative error given by Eq. (22).
- The Michalewicz function (defined in Appendix A.2) exhibits valleys and ridges with a tunable steepness. The convergence is assessed based on the objective value and a threshold of 10^{-3} for the associated relative error (22).
- The Ackley function (defined in Appendix A.1 for an arbitrary number of dimensions d) is characterized by many local minima, which makes it difficult to optimize. The value of the global minimum is independent of the number of dimensions and is located in an area of steep gradient. We consider the $d = 2$ case. The convergence is assessed based on the proximity of the design variables with the reference solution, as the relative error on the objective function value is not defined in that case ($f(x^*) = 0$). The proximity is computed with Eq. (23) and a threshold is set to 10^{-3}.

For SEGOMOE, as stated above, the same sets of DOE (one set per initial size) are used for all the criteria for a fair comparison. Universal kriging models (with squared exponential correlation functions and linear regression functions) are used as local experts to build the surrogate model of the objective $f(\boldsymbol{x})$. The iteration budget is set to 300 evaluations. Each computation can be stopped early, either because convergence has been reached or because of a singularity arising when enrichment points are too close. The optimization of the infill SEGOMOE criterion is performed using the SLSQP algorithm [74] with a multistart approach. The gradients for SLSQP are computed analytically for speed and accuracy. The convergence is assessed with a threshold of 10^{-3} for the relative error (see Eq. (22)) or the proximity index (see Eq. (23)), with an iteration budget of 300 function evaluations. The results are summarized in Table 1.

Table 1
SEGOMOE results for 3600 runs (100 per combination of DOE size, criterion, and function) for the six-hump camel-back, Michalewicz, and Ackley problems, showing the percentage of runs that converged to the analytic solution (with a relative error threshold or a proximity index of 10^{-3}). The best results are shown in bold.

Criterion	Results	DOE 5	DOE 10	DOE 20	DOE 30
Six-hump camel-back function					
WB2	Converged	94%	**100%**	**100%**	**100%**
	Mean (evaluations)	**23**	**29**	**40**	**43**
	σ (evaluations)	6	7	7	4
EI	Converged	**100%**	**100%**	**100%**	**100%**
	Mean (evaluations)	39	40	42	44
	σ (evaluations)	4	4	**4**	3
WB2S	Converged	**100%**	99%	**100%**	**100%**
	Mean (evaluations)	39	39	42	44
	σ (evaluations)	**3**	**3**	**4**	**3**
Michalewicz function					
WB2	Converged	71%	76%	83%	87%
	Mean (evaluations)	**25**	**31**	**36**	**46**
	σ (evaluations)	**14**	**24**	**15**	**13**
EI	Converged	**100%**	**100%**	**100%**	**100%**
	Mean (evaluations)	47	45	51	59
	σ (evaluations)	36	43	31	21
WB2S	Converged	**100%**	99%	**100%**	**100%**
	Mean (evaluations)	48	46	51	55
	σ (evaluations)	36	33	27	15
Ackley function					
WB2	Converged	32%	58%	90%	91%
	Mean (evaluations)	**41**	**55**	**45**	**50**
	σ (evaluations)	**30**	**38**	27	**17**
EI	Converged	71%	98%	98%	98%
	Mean (evaluations)	95	73	56	80
	σ (evaluations)	96	68	30	56
WB2S	Converged	**97%**	**100%**	**100%**	**100%**
	Mean (evaluations)	107	60	58	68
	σ (evaluations)	90	45	**26**	23

In this table, higher success rates were obtained in decreasing order by WB2S (10 times), EI (8 times), and WB2 (3 times). WB2 seems more efficient when judged on the basis of the mean and standard deviation of the evaluations. However, it is also important to consider its rate of convergence. For the Michalewicz and the Ackley function, a significant number of runs did not converge (between 70% and 40% for the initial DOEs with fewer than 20 points), although they converged with the EI and the WB2S criterion. This can be explained by the multimodality of the considered functions, as some runs performed with the WB2 criterion ended up converging near the local minima. This issue is mitigated as the size of the initial DOE is increased. The use of more points improves exploitation and reduces the likelihood of being trapped near the local optima. The EI and the WB2S criterion performed similarly across all sizes of the DOE in terms of both rate of con-

Table 2
Six-hump, Michalewicz, and Ackley results using COBYLA or NOMAD (100 runs). Different stopping criteria relative to the maximum number of blackbox evaluations were tested. The best results are shown in bold.

Algorithm	COBYLA	NOMAD		
MAX_BB_EVAL	500	100	300	without
Six-hump camel-back				
Converged	77%	93%	**95%**	**95%**
Mean (evaluations)	**51**	100	294	308
σ (evaluations)	5	**0**	10	25
Michalewicz				
Converged	29%	87%	**89%**	**89%**
Mean (evaluations)	**60**	100	290	300
σ (evaluations)	27	**0**	13	26
Ackley				
Converged	38%	98%	**100%**	**100%**
Mean (evaluations)	**59**	100	300	499
σ (evaluations)	5	**0**	**0**	12

vergence and mean number of evaluations. The standard deviation of the WB2S was slightly better for DOE sizes ranging from 10 to 30. This behavior was expected, as explained in Section 2.2.1.

The comparison results between COBYLA and NOMAD are listed in Table 2. For COBYLA, the convergence to the global minima was obtained with a significantly lower success rate compared to that for SEGOMOE. However, when COBYLA converges, it requires roughly the same number of function evaluations (between 51 and 60) as that of SEGOMOE. Table 2 shows also the NOMAD results for the three performance criteria as a function of the stopping criterion chosen (MAX_BB_EVAL for a maximum number of blackbox evaluations of 100, 300, and no limit). Thus, the number of evaluations with NOMAD was much higher than that with the SEGOMOE approach with a similar success rate.

The efficiency of SEGOMOE was validated for three multimodal unconstrained problems compared to the two other derivative-free algorithms (COBYLA and NOMAD). In terms of criteria, EI and WB2S yielded similar results in two problems, whereas WB2S performed better in the third one.

3.5. Analytic constrained multimodal problems

To evaluate the ability of SEGOMOE to tackle multimodal and constrained problems, we used the following two test cases:

- The modified Branin problem, which is a two-dimensional nonlinear problem with a single nonlinear constraint [65]. This modification, which was proposed by Parr et al. [65], adapts the original Branin function to have a single global optimum and two local ones, rather than three optima of equal value.

$$\begin{cases} \min\limits_{\boldsymbol{x}\in[-5,10]\times[0,15]} f(\boldsymbol{x}) \\ g(\boldsymbol{x}) \geq 0 \end{cases}, \tag{24}$$

  where $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are as detailed in Appendix A.4. The feasible space for this problem consists of isolated feasible regions within their design space, which often occurs for severely constrained practical cases. The problem features three distinct feasible regions delimited by green curves in Fig. 4, making it an excellent test for constraint strategies [25, 65].
- The Linear-Ackley-Hartman (LAH) mixed-constrained problem with four input dimensions, a linear objective, and two constraints like the one described by Picheny et al. [26]

$$\begin{cases} \min\limits_{\boldsymbol{x}\in[0,1]^4} f(\boldsymbol{x}) \\ g(\boldsymbol{x}) \leq 0 \\ h(\boldsymbol{x}) = 0 \end{cases} \tag{25}$$

  The first inequality constraint is the Ackley function in four dimensions, and the second one is an equality constraint following the Hartman four-dimensional function. The problem is denoted by LAH in the following, and details on the function expressions and the reference solution are given in Appendix A.5.

For these two constrained test cases, we compared SEGOMOE, COBYLA, and NOMAD with the ALBO framework. As the LAH problem is a mixed-constrained problem, the equality constraint $h(\boldsymbol{x})$ in Eq. (25) has been transformed into two inequality constraints for COBYLA and NOMAD. SEGOMOE and ALBO are both able to consider mixed-constraint problems.

In our tests, we sought to quantify the impact of the size of the initial DOE and the choice of infill criterion for SEGOMOE. As for the unconstrained case, the results were obtained by considering universal kriging models (with squared exponential correlation functions and linear regression functions) as local experts to build the surrogate models of both the objective $f(\boldsymbol{x})$ and the constraints $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$. The iteration budget was still set to 300 evaluations. Optimization under constraints of the infill SEGOMOE criterion was still performed using the SLSQP algorithm [74] with a multistart approach. Because the initial DOE was computed using an enhanced stochastic evolutionary (ESE) algorithm [70] for building the LHS, we ran the same case 100 times to obtain statistically significant results. Furthermore, to reduce the bias that could exist between the results obtained with different infill criteria should different batches of initial DOEs be used, we kept the same initial batches for the criteria (size-wise) by means of a hot start feature implemented in SEGOMOE. The same DOEs were used to initialize the ALBO framework.

To compare the performances of the different algorithms (SEGOMOE, ALBO, COBYLA, and NOMAD), we used the criteria defined in Section 3.1. For the modified Branin problem, the convergence was assessed for each feasible point found using the relative error of the objective with respect to the known solution (see Eq. (22)), and a convergence threshold of 10^{-3}. A point is only a priori feasible when the relative violation of the constraint is less than 10^{-4}. For the LAH problem, a convergence threshold of 10^{-3} (see Eq. (23)) on the proximity index was used. The results of the optimizations are summarized in Table 3.

From Table 3, we can see that the convergence success rate increases with the size of the initial DOE for all the criteria considered, reaching nearly 100% for SEGOMOE with WB2 or WB2S when more than 20 points were used. When comparing the three criteria we used, we found that WB2S was significantly better than the other two for sparse initial DOEs (sizes of 5 and 10), for which it had a 69% success rate for modified Branin and 10% for the LAH function. This can be attributed to the fact that, in WB2S, the scaling eases exploration. For all criteria, the mean of the SEGOMOE cost increased with the size of the initial DOE; however, it is important to note that the variance decreased. The lower success rates for the sparser initial DOEs can be explained by the inaccuracy of the surrogate of the constraint, which prevented the enrichment optimizer from searching the promising areas.

If the WB2 and WB2S criteria both gave a convergence success rate close to 100% for all the DOE sizes (10, 20, and 30) with a similar number of function evaluations (mean and variance) for the LAH function, the EI criterion was less efficient on this example. Concerning the other BO algorithm, ALBO converged with a success rate of less than 45% with a large number of function evaluations (greater than 50). The ALBO results on the LAH func-

Table 3
Results for 3200 runs (100 per combination of DOE size and criterion) of the modified Branin and the LAH problems. DOEs range from 5 to 30 points, and the number of function evaluations to reach the minimum value is reported both for SEGOMOE depending on the infill criterion used (EI, WB2, or WB2S) and for ALBO. The number of converged runs to the analytic solution (with a relative error threshold or a proximity index of 10^{-3}) is given in percent. The best results are shown in bold.

Criterion	Results	DOE 5	DOE 10	DOE 20	DOE 30
Modified Branin function					
WB2	Converged	62%	78%	93%	99%
	Mean (evaluations)	38	**39**	53	51
	σ (evaluations)	28	22	20	9
EI	Converged	50%	75%	**96%**	**100%**
	Mean (evaluations)	**27**	45	**49**	**45**
	σ (evaluations)	22	29	21	**8**
WB2S	Converged	**69%**	**76%**	90%	**100%**
	Mean (evaluations)	34	**39**	**49**	47
	σ (evaluations)	20	**17**	**19**	9
ALBO	Converged	41%	32%	35%	19%
	Mean (evaluations)	58	57	70	70
	σ (evaluations)	**15**	13	8	7
LAH function					
WB2	Converged	**100%**	**100%**	**100%**	**100%**
	Mean (evaluations)	**17**	**19**	**28**	**37**
	σ (evaluations)	4	3	**3**	**3**
EI	Converged	96%	86%	80%	68%
	Mean (evaluations)	45	33	43	43
	σ (evaluations)	31	15	15	11
WB2S	Converged	**100%**	**100%**	**100%**	**100%**
	Mean (evaluations)	18	20	29	**37**
	σ (evaluations)	**3**	**2**	**3**	**3**
ALBO	Converged	5%	8%	13%	27%
	Mean (evaluations)	29	50	65	84
	σ (evaluations)	11	7	8	8

tion are quite different from those presented by Picheny et al. [26]; this could be explained by the difference in the relative violation of the constraint, i.e., 10^{-4} instead of 10^{-2}.

To visualize the enrichment process in two dimension, Fig. 4 shows the modified Branin function, where the three feasible regions are shown in green. The global optimum of the modified Branin case is at the border of the feasible region at the bottom right-hand corner of the contour plots. When the WB2S or WB2 infill criterion was used, the solution successfully converged to the global minimum. However, when the conventional EI criterion was used, the solution remained trapped in the feasible region discovered by the initial DOE. This shows that the numerical improvement of WB2S relative to EI eases the enrichment optimization, allowing a better exploration. Moreover, the increased contribution of EI within WB2S, compared to WB2, enabled the optimizer to identify the second feasible region and, therefore, to converge for this particular configuration.

As for the unconstrained test cases, comparisons were performed with the DFO algorithms (COBYLA and NOMAD), as reported in Table 4, where 100 runs (corresponding to 100 different starting points from an LHS) were performed and the same threshold criterion of 10^{-3} for the relative error or the proximity index was used. For the modified Branin problem, the convergence success rate was very low for COBYLA (12%). For NOMAD, the constraint was treated with the Progressive Barrier option [75], which needs to be satisfied only at the solution and not necessarily at the intermediate points (relaxable constraint). The convergence success rate was very low (less than 27%), and the number of function evaluations was still significant (more than 300). These values were compared with fewer than 51 evaluations for the SEGOMOE results (see Table 3). For the LAH problem, COBYLA with 53% convergence success rate had a number of function evaluations quite comparable to the SEGOMOE-EI results. For NOMAD, the equality constraint was treated as two inequality constraints with the Progressive Barrier option [75]. The convergence success rate was 0% for the 10^{-3} threshold and was very low (2%) if we increased the threshold value to 10^{-1} (requiring 500 function evaluations in this case). In this LAH test case, a maximum number of function evaluations `MAX_BB_EVAL` is required to obtain a reasonable a CPU time. The obtained values were compared with fewer than 43 evaluations for the SEGOMOE results (see Table 3). To sum up on these two constrained test cases, SEGOMOE with the EI, WB2, or WB2S criterion was still more efficient than ALBO, NOMAD, or COBYLA in terms of both convergence success rate and number of function evaluations.

Table 4
Modified Branin and LAH problems with COBYLA or NOMAD (100 runs). Different stopping criteria relative to the maximum number of evaluations were tested. The best results are shown in bold.

Algorithms	COBYLA	NOMAD		
`MAX_BB_EVAL`	500	100	300	Without
Modified Branin				
Converged	12%	20%	**27%**	**27%**
Mean (evaluations)	**60**	100	300	641
σ (evaluations)	**6**	0	0	123
LAH				
Converged	**53%**	0%	0%	–
Mean (evaluations)	**42**	–	–	–
σ (evaluations)	**11**	–	–	–

In the five analytic multimodal optimization problems considered in this section, EI and WB2S had a similar behavior in some cases, which justifies the use of a scalar multiplier $s > 1$ in the WB2S formula to promote the exploration phase. In other cases, WB2S gave better results than those of EI; especially when the size of the initial DOE was small, its convergence success rate was close to 100%. When EI found a global optimum, WB2S managed to find it also, although the reciprocity was not always satisfied.

4. Application to wing aerodynamic shape optimization

In the previous sections, we demonstrated the performance of SEGOMOE for analytic cases. We now demonstrate it on a design optimization problem that is more representative of a real-world problem: a nonlinearly constrained wing aerodynamic shape optimization problem where the objective function is multimodal. We compare SEGOMOE with a gradient-based optimizer (SNOPT) that requires a multistart approach to reach the global optimum. These comparisons are based on the performance criteria described in Section 3.1.

The final problem is a simplified version of an aerodynamic shape optimization problem defined by the ADODG, which has one equality constraint and exhibits multimodality.

4.1. High-fidelity aerodynamic shape optimization framework

To perform the aerodynamic shape optimization, we use a framework that combines a Reynolds-averaged Navier–Stokes (RANS) CFD solver, a geometry parametrization engine, and a mesh perturbation algorithm. The CFD solver is ADflow, which uses a second-order finite-volume scheme to solve the compressible Euler equations, laminar Navier–Stokes, and RANS equations (steady, unsteady, and time periodic) on overset meshes [76,77]. The Spalart–Allmaras turbulence model [78] is used to complete the RANS equations. The solver combines a Runge–Kutta method

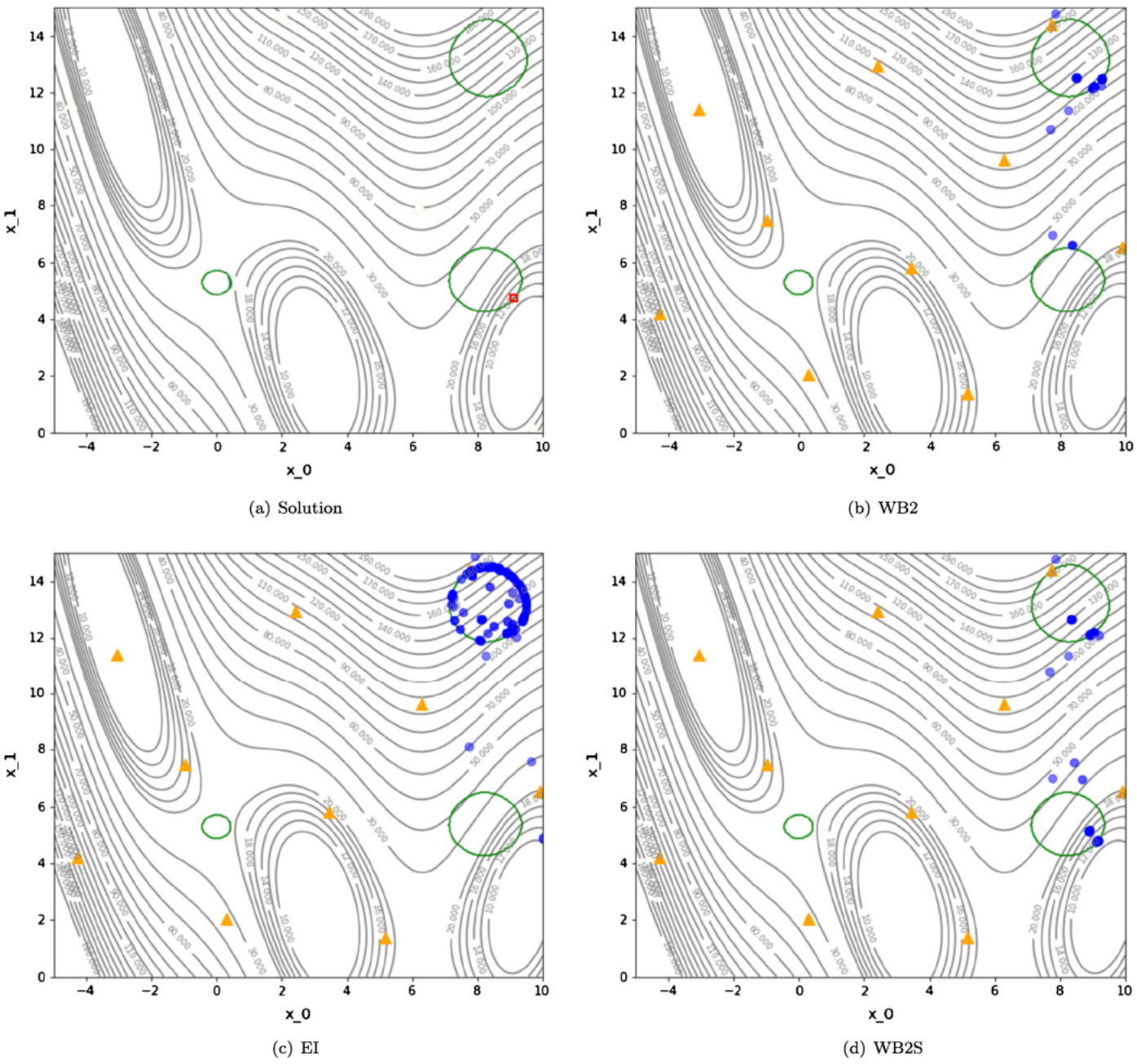


Fig. 4. Contour plots of the modified Branin function showing an example of criterion sensitivity for an initial DOE of 10 points. The color of the enrichment points (blue) fades according to how old they are in the optimization history. The feasible areas are inside the three green circles, and the optimal exact solution is shown by the red square in (a). The enrichment process with WB2, EI, and WB2S is given in (b), (c), and (d), respectively. EI is trapped in a local optimal point, whereas WB2 and WB2S find the global optimum.

for the initial iterations with a Newton–Krylov algorithm that increases the convergence success rate in the later iterations.

ADflow is especially effective when used in conjunction with gradient-based optimization because it efficiently computes accurate gradients with respect to large numbers of design variables using an adjoint method. The adjoint method is implemented using a hybrid approach that selectively uses automatic differentiation to generate the code that computes the partial derivatives in the adjoint equations [77]. The geometry is parametrized using an implementation of free-form deformation (FFD) [79], and the mesh deformation is performed with an efficient analytic inverse distance method [80].

The integration of ADflow into the optimization algorithms is achieved through pyOpt, a common Python interface that facilitates the use of different optimization algorithms [81,82]. The gradient-based optimizer that we use here is SNOPT [83], a sequential quadratic programming optimizer that can handle large-scale nonlinear constrained problems. This aerodynamic shape optimization framework has been used to solve various problems [4, 84–87] and has also been extended to perform aerostructural design optimization [3,88–90]. For this specific study, SEGOMOE was integrated into pyOpt, enabling us to reuse the same Python scripts when benchmarking SEGOMOE against SNOPT.

4.2. Problem definition

The problem is a simplification of the ADODG Case 6 benchmark ("Multimodal Subsonic Inviscid Optimization") [6,50], where

Table 5
Definition of the simplified ADODG Case 6 optimization problem.

	Function/ variable	Description	Quantity	Range
Minimize	C_D	Drag coefficient	1	
with respect to	α	Angle of attack	1	$[-3.0, 6.0]$ (°)
	θ	Twist	8	$[-3.12, 3.12]$ (°)
	δ	Dihedral	8	$[-0.25, 0.25]$ (δ/c)
		Total variables	17	
subject to	$C_L = 0.2625$	Lift coefficient	1	
		Total constraints	1	

we reduce the number of variables. This case was devised to explore the existence of multiple local minima in aerodynamic wing design. The baseline geometry is a rectangular wing with a chord of 1.0 m and a NACA 0012 airfoil cross section with a sharp trailing edge. The semi-span is 3.06 m and the wing is fitted with a rounded wingtip cap. In the full benchmark problem, the optimizer is given freedom to change the twist, chord, dihedral, sweep, span, and sectional shape variables. In the modified version used for this study, we reduced the variables to twist and dihedral, for a total of 17 design variables. The geometry is parameterized using the FFD approach implemented in pyGeo [79], which allows the definition of global design variables with control of the sections of the B-spline control points. Nine sections are defined along the span of the wing with heavier clustering toward the wing tip. The twist variables rotate eight of these sections (excluding the root section) about the quarter-chord. Likewise, each of the eight dihedral variables controls the vertical displacement of one of the spanwise sections, excluding the root section. The angle of attack can be varied to allow the optimizer to satisfy the lift constraint. The objective of the problem is to perform a lift-constrained drag minimization of a simple wing under subsonic flow (M_∞ = 0.5), using the Euler equations. The main characteristics of the optimization problem are summarized in Table 5.

4.3. Gradient-based optimization results

The SEGOMOE approach is compared to SNOPT on the same test case. These gradient-based results, as well as those of other cases related to ADODG Case 6, are presented in more detail by Bons et al. [6]. Here, we just compare the SEGOMOE results for the Euler-based twist and dihedral optimization case, and summarize the corresponding results for completeness. The optimization problem is solved using SNOPT [83] starting from the 15 random shapes shown in Fig. 5(a), where we use a coarse mesh grid (L3 mesh with 180K cells). The average number of iterations required to converge is approximately 100. Nevertheless, in addition to the standard evaluation, ADflow also needs to compute the gradients for each aerodynamic-related function (drag and lift), each of which requires an adjoint solution. Therefore, for this problem, the mean evaluation budget is equivalent to approximately 300 standard ADflow calls. The optimal shapes are shown in Fig. 5(b), and the optimal results are summarized in Table 6. This problem has multiple local optima and a single global one, although the designs are very close in terms of drag value: 39.9091 drag counts for the global optimum and around 40.1980 drag counts for the local ones. Of the 15 runs, 5 converged toward the global optimum, which is characterized by an upward winglet. For the other 10 runs, multiple local minima were found for shapes with a downward winglet.

Table 6
Results obtained with SNOPT and ADflow using the adjoint method. The global optimum is written in bold.

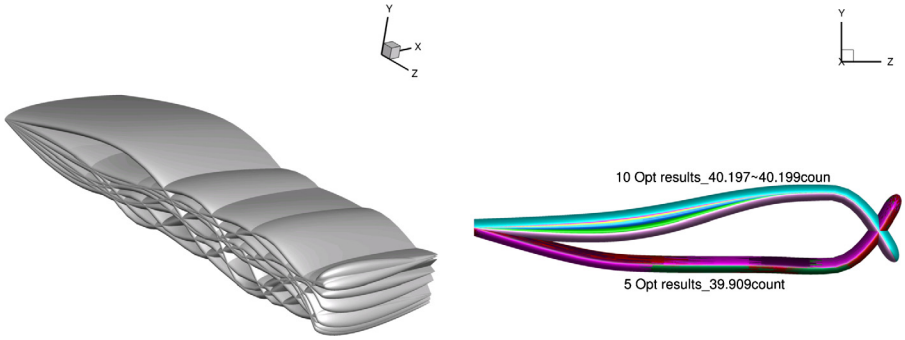
$C_D \times 10^4$ (drag counts)	Number of runs	Mean constraint violation
39.9091	**5**	7×10^{-10}
40.1971	3	1×10^{-9}
40.1972	3	1×10^{-8}
40.1975	2	6×10^{-10}
40.1986	1	8×10^{-10}
40.1995	1	5×10^{-10}

The radar chart in Fig. 6 shows the values of the 17 design variables for the various optima. We can see that all local minima differ only slightly on the twist parameters and on the dihedral parameters in the inner part of the wing. The multimodality of this problem, together with the complexity of the equality constraint, makes this a challenging proving ground for SEGOMOE.

4.4. SEGOMOE results

We performed a series of studies similar to those performed for the analytic functions in Sections 3.4 and 3.5. The runs were performed using a fixed evaluation budget of 500 evaluations, including the initial DOE and iterations. Six initial DOE sizes were considered, ranging from $1 \times d = 17$ to $6 \times d = 102$. Given the dimension of the problem, KPLS+K surrogates were selected as local experts of the MOE because we have found in previous studies that they offer the best trade-off between efficiency and accuracy [42]. The tolerance on the C_L was set to 10^{-5}, and the enrichment step was performed using the SLSQP optimizer [74] with 50 randomly picked starting points at each step.

Both the WB2 and the WB2S criterion were tested using the same batches of initial DOEs (18 DOEs per batch, computed using an optimized LHS algorithm) for each size of initial DOE, leading to a total of 216 runs. As previously mentioned, this allows us to



(a) Initial configurations (b) Optimal shapes

Fig. 5. SNOPT results for wing optimization with respect to the twist and dihedral variables. Multiple local minima were found when starting from 15 randomly generated initial configurations.

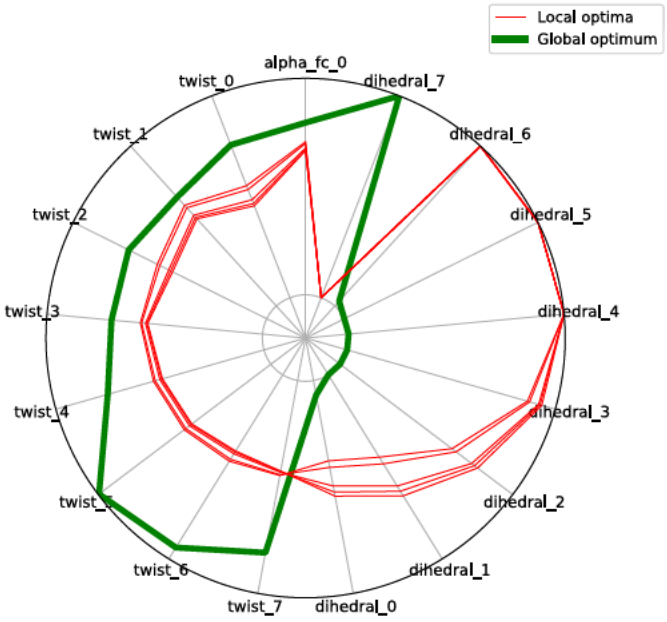


Fig. 6. Radar plot of the optima found by gradient-based optimization: SNOPT global minimum (green) and SNOPT local minima (red).

Table 7
Proximity indices between the local optima and the global one ($C_D = 39.9091$ drag counts) computed with Eq. (23). The best value in bold was obtained for the global optimum.

C_D (drag counts)	**39.9091**	40.1971	40.1972	40.1975	40.1986	40.1995
$\mu_{\text{prox-glob}}$	**1**	0.458	0.449	0.441	0.494	0.505

study the sensitivity with respect to the initial sampling without introducing bias into the results.

The best point obtained for each run is shown in a radar plot that compares the results for different initial DOE sizes and the infill criteria. Only the four most significant plots (out of 12) are shown in Figs. 7 and 8, where we compare the WB2 and WB2S criteria. The optima found by the gradient-based optimizer (SNOPT) in Section 4.3 are also shown for comparison, where the global optimum is shown in green and the five local optima are shown in red.

Because not all runs converge exactly to one of the optimum of the problem within the given evaluation budget, we use the proximity index to quantify the difference between two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ defined by Eq. (23). The proximity indices of the local minima computed with the global optimum as reference ($\mu_{\text{prox-glob}}$) are listed in Table 7. When the point is the global optimum, then $\mu_{\text{prox}} = 1$.

Because two distinct optimal regions exist, proximity indices can be used to assess whether a point belongs more to the "local region" or to the "global one," by comparing its proximity to the global and local optima.

The cost of the optimization is evaluated using the mean, maximum, and minimum number of evaluations corresponding to the best points found. Owing to the relatively small number of runs performed, the minimum and maximum are deemed more relevant than the standard deviation when analyzing the sensitivity to the initial DOE. The accuracy is assessed by computing the mean proximity index of the best points of a batch with the known global optimum ($\mu_{\text{prox-glob}}$), and the feasibility is assessed by computing the mean constraint violation. All the results are compiled in Table 8.

Both the radar plots and the statistics in Table 8 show that the WB2S criterion is more robust than the WB2 criterion in finding the "global region" for all sizes of the initial DOE within the specified budget. For instance, Figs. 7 and 8 compare the behavior of both criteria for $1 \times d$ and $4 \times d$ DOE points. We can see that, for $1 \times d$, only WB2S managed to find the global optimum shape (higher proximity index). For $4 \times d$, all WB2S optimizations converged to the global optimum area, which was not the case with WB2. We can see this by looking at the success rate in finding the global optimum ("Global" in Table 8) and the closely related $\mu_{\text{prox-glob}}$ index defined by Eq. (23).

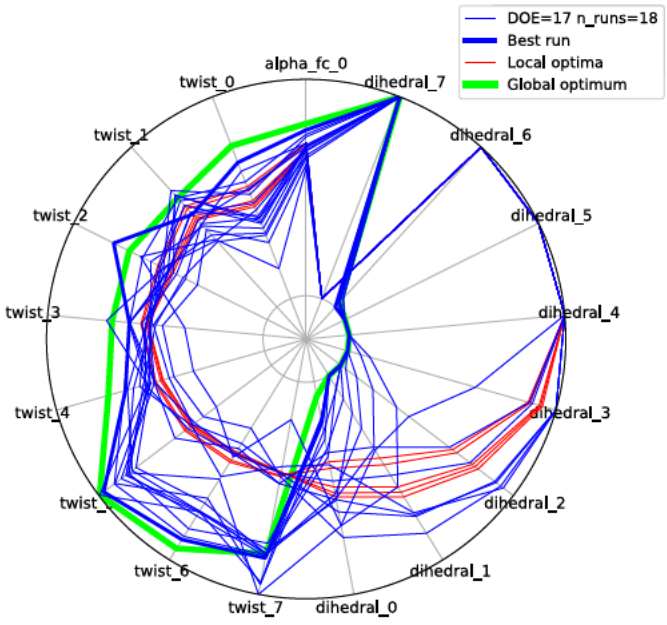
The robustness of WB2S comes at an additional cost of 40 to 120 iterations compared to WB2 in terms of mean evaluations. The sensitivity to the initial DOE can be noticed by looking at the sharp differences between the maximum and the minimum number of evaluations for the WB2S criterion ($\approx$ 200–250). It is nonetheless important to keep in mind that these results are linked to the evaluation budget as the SEGOMOE approach performs exploration without using a stopping criterion other than the budget. For instance, several runs yield their best point on the last iteration, which might be noticeably improved with few additional iterations.

The mean evaluation budget, irrespective of the infill criterion selected, is around 50% higher than that required for one gradient-based optimization with SNOPT and the adjoint method. Likewise and logically, the mean constraint violation obtained with SEGOMOE was also higher than what was achieved by the gradient-based optimization using SNOPT, although it is totally in line with the chosen feasibility tolerance (10^{-5}).

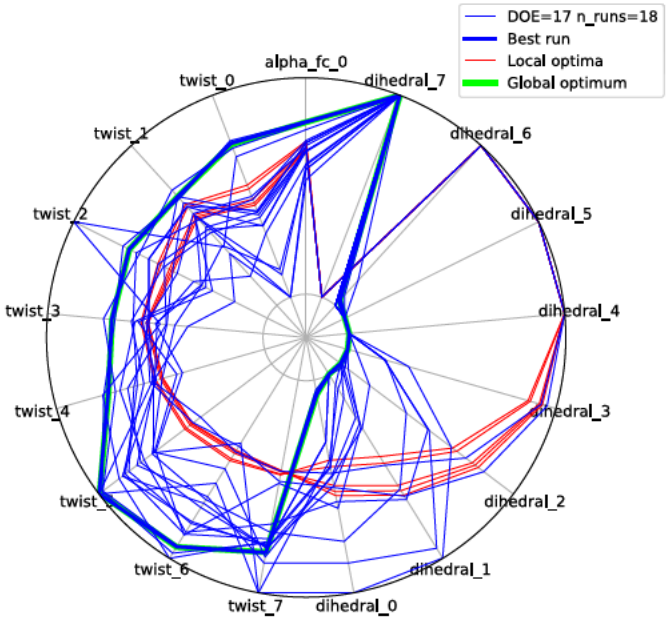
The gradient-based optimization only converged a third of a time to the global optimum. Although SEGOMOE rarely exactly

Table 8
SEGOMOE statistics for the simplified ADODG Case 6 optimization problem for different infill criteria (WB2 and WB2S) and initial DOE sizes. The best results in terms of proximity index or mean evaluations are shown in bold.

Criterion	Results	DOE 17 $= d$	DOE 34 $= 2d$	DOE 51 $= 3d$	DOE 68 $= 4d$	DOE 85 $= 5d$	DOE 102 $= 6d$
WB2	Global	61.1%	77.8%	77.8%	88.9%	83.3%	77.8%
	$\mu_{\text{prox-glob}}$	0.714	0.785	0.801	0.892	0.844	0.801
	Mean evaluations	**382**	**426**	**414**	**409**	**428**	**368**
	Maximum evaluations	500	500	500	497	498	489
	Minimum evaluations	225	273	189	270	254	248
	Mean violation	3.9×10^{-6}	4.0×10^{-6}	3.2×10^{-6}	2.97×10^{-6}	2.64×10^{-6}	3.22×10^{-6}
WB2S	Global	**88.9%**	**94.5%**	**100%**	**100%**	**100%**	**100%**
	$\mu_{\text{prox-glob}}$	**0.808**	**0.856**	**0.888**	**0.954**	**0.934**	**0.949**
	Mean evaluations	399	458	468	477	478	468
	Maximum evaluations	497	500	500	500	498	500
	Minimum evaluations	256	312	335	406	370	408
	Mean violation	3.5×10^{-6}	3.5×10^{-6}	4.5×10^{-6}	3.24×10^{-6}	2.52×10^{-6}	2.76×10^{-6}

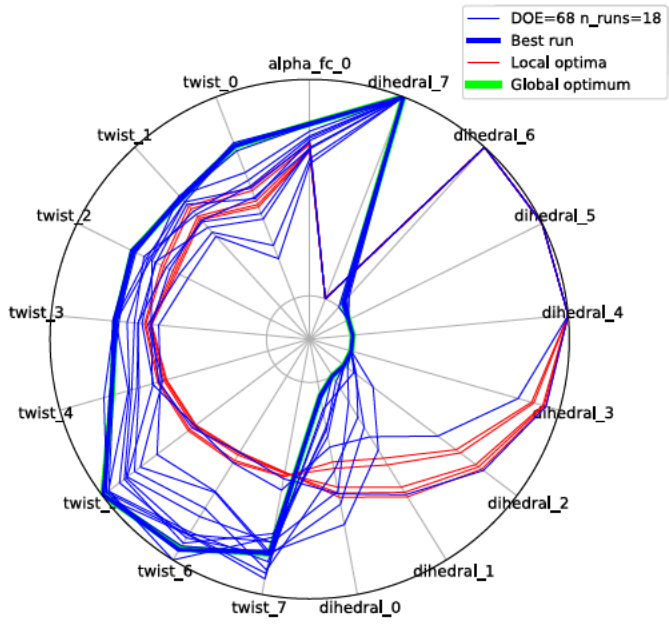


(a) WB2

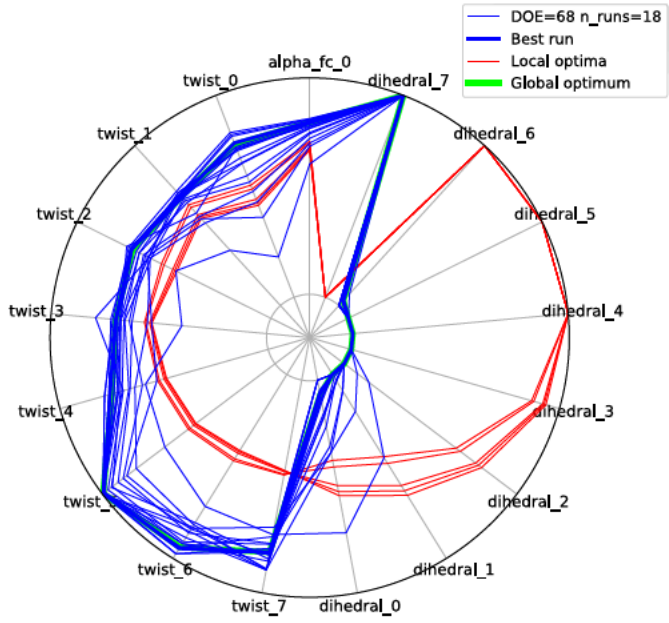


(b) WB2S

Fig. 7. Best points obtained with SEGOMOE (blue) for the initial DOEs of 17 points, as well as for the SNOPT global minimum (green) and SNOPT local minima (red).



(a) WB2



(b) WB2S

Fig. 8. Best points obtained with SEGOMOE (blue) for the initial DOEs of 68 points, as well as for the SNOPT global minimum (green) and SNOPT local minima (red).

converges toward an optimum within the budget, it reliably identifies the neighborhood of the global optimum (189 out of 216 times) with acceptable statistical accuracy. This is especially true when we used the WB2S criterion (105 out of 108 times, or 97%). Table 8 shows that the best size of the initial DOE is $4 \times d$, both in terms of finding the global optimum and in terms of accuracy.

5. Conclusions

We presented a new constrained global optimization methodology, namely, super-efficient global optimization coupled with mixture of experts (SEGOMOE), that can handle nonlinearly constrained multimodal problems at a reasonable computational cost, even when the function evaluations are costly. We built on previous algorithms by developing and implementing a new infill criterion (WB2S) and by implementing an MOE with a new recombination strategy. The use of MOE based on kriging models allows us to have a global surrogate model that approximates heterogeneous functions for the objective and the constraints.

The proposed approach can handle complex design optimization problems through the use of an adaptive surrogate modeling approach based on kriging surrogate models. The ability

of SEGOMOE to tackle multimodal optimization problems (both constrained and unconstrained) was evaluated using five analytic benchmark problems. For the first three problems, we compared SEGOMOE with two other derivative-free algorithms (NOMAD and COBYLA). The number of function evaluations was drastically reduced with the proposed approach, and the convergence success rate was higher: 99% for WB2S, compared to 89% for NOMAD and 29% for COBYLA in the worst cases. In the best cases, WB2S achieved 100%, compared to 100% for NOMAD and 77% for COBYLA. The fourth analytic problem (a modified Branin function) enabled us to obtain statistically significant results for different infill criteria and different initial DOE sizes for SEGOMOE. The results show the overall efficiency of the SEGOMOE algorithm and demonstrate the advantages of WB2S, which performs a better design space exploration, especially for sparse initial DOEs. On this constrained multimodal analytic problem, COBYLA and NOMAD had a small convergence success rate (12% and 27%, respectively) with a significant number of function evaluations. For the mixed-constrained problem considered in four dimensions, NOMAD had a poor convergence success rate (2% to be closed to the reference solution). The ALBO algorithm was compared to the different variants of SEGOMOE in this fifth analytic example and managed to converge to the reference solution with a 27% success rate and a high number of function evaluations (84 calls compared to 37 with SEGOMOE).

We also presented the results for an aerodynamic shape constrained optimization problem based on the ADODG Case 6 benchmark, which exhibits multimodality. An exhaustive study of the infill criteria with various DOE sizes was conducted, and SEGOMOE converged to the vicinity of the global minimum area under a fixed computational budget. Among the criteria, WB2S appeared more robust than WB2 for all DOE sizes, confirming its advantages. Compared to a gradient-based optimizer (SNOPT) that takes advantage of adjoint gradients, and assuming a DOE size of $4 \times d$, SEGOMOE always identified the global minimum for an average budget that is only 50% higher than that of the gradient-based one. The gradient-based approach, on the other hand, only reached the global optimum 33% of the time.

Conflict of interest statement

None declared.

Acknowledgements

This work is part of the activities of the ONERA–ISAE–ENAC joint research group. We would like to thank the following internship students for their work on the SEGOMOE optimizer: Vivien Stilz, Marie Gibert, and Valentin Fievez. We are also grateful to Rémi Lafage for his support with the SEGOMOE framework and the ADflow installation at ONERA, Régine Leconte for her support on the ISAE Cluster, and Youssef Diouane for his suggestions. This work was supported by the AGILE project (Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts), as well as by the European Union Horizon 2020 Program (H2020-MG-2014-2015) under grant agreement number 636202. Additional support was provided by the Air Force Office of Scientific Research (AFOSR) MURI on "Managing multiple information sources of multi-physics systems," program officer Jean–Luc Cambier, award number FA9550-15-1-0038. Joseph Morlier acknowledges the ISAE–SUPAERO foundation, which supported his visiting scholar position at the University of Michigan. This work was also partially supported by the EU project 658570—NextGen Airliners—funded by Marie Skłodowska-Curie Actions (MSCA-IF-2014-EF).

Appendix A

A.1. Ackley function

Objective

$$f(x) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$$

Parameters (usual)

$a = 20 \quad b = 0.2 \quad c = 2\pi$

Search domain

$x_i \in [-32.768, 32.768] \quad \forall i \in [1, \ldots, d]$

Global minimum

$f(x^*) = 0 \quad x^* = [0, \ldots, 0]$

A.2. Michalewicz function

Objective

$$f(x) = -\sum_{i=1}^{d} \sin(x_i) \sin^{2m}\left(\frac{i{x_i}^2}{\pi}\right)$$

Search domain

$x_i \in [0, \pi] \quad \forall i \in [1, \ldots, d]$

Global minimum according to the dimension d

$d = 2 \quad f(x^*) = -1.8013 \quad x^* = [2.20, 1.57]$

$d = 5 \quad f(x^*) = -4.687658$

$d = 10 \quad f(x^*) = -9.66015$

A.3. Six-hump camel-back function

Objective

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

Search domain

$x_1 \in [-3, 3] \quad x_2 \in [-2, 2]$

Global minimum

$f(x^*) = -1.0316 \quad x^* = [0.0898, -0.7126]$ and $x^* = [-0.0898, 0.7126]$

A.4. Modified Branin problem

The modified Branin function [65] is a version of the Branin function normalized to [0, 1] [22]. The constraint function is a normalized version of the `Gomez#3` function [47], with an additional sine wave to increase multimodality:

Objective

$$f(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left[\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 1\right] + \frac{5x_1 + 25}{15}$$

Constraint

$$g(x_1, x_2) = \left(4 - 2.1y^2 + \frac{y^4}{3}\right) y^2 + yz + 4\left(y^2 - 1\right) z^2 + 3\sin[6(1-y)] + 3\sin[6(1-z)] - 6$$

where $y = \frac{x_1 - 2.5}{7.5}$, $z = \frac{x_2 - 7.5}{7.5}$

$$g(x_1, x_2) \geq 0$$

Search domain

$$x_1 \in [-5, 10] \quad x_2 \in [0, 15]$$

Global minimum

$$f(x^*) = 12.005$$

A.5. Linear Ackley–Hartman problem

The objective of the linear Ackley–Hartman function is a simple linear function, the inequality constraint is the Ackley function (centered) and the equality constraint is the "Hartman" function (centered, rescaled) as described by Picheny et al. [26].

Objective

$$f(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} x_i$$

Inequality constraint

$$g(x_1, x_2, x_3, x_4) = 3 + 20\exp\left(-0.2\sqrt{\frac{1}{4}\sum_{i=1}^{4}(3x_i - 1)^2}\right) + \exp\left(\frac{1}{4}\sum_{i=1}^{4}\cos(2\pi(3x_i - 1))\right) - 20 - \exp(1)$$

$$g(x_1, x_2, x_3, x_4) \leq 0$$

Equality constraint

$$h(x_1, x_2, x_3, x_4) = \frac{1}{0.8387}\left[-1.1 + \sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{4} a_{ji}(x_j - p_{ji})^2\right)\right]$$

with

$$\mathbf{C} = \begin{bmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} 10.0 & 0.05 & 3.0 & 17.0 \\ 3.0 & 10.0 & 3.5 & 8.0 \\ 17.0 & 17.0 & 1.70 & 0.05 \\ 3.5 & 0.1 & 10.0 & 10.0 \end{bmatrix},$$

$$\mathbf{p} = \begin{bmatrix} 0.131 & 0.232 & 0.234 & 0.404 \\ 0.169 & 0.413 & 0.145 & 0.882 \\ 0.556 & 0.830 & 0.352 & 0.873 \\ 0.012 & 0.373 & 0.288 & 0.574 \end{bmatrix}.$$

$$h(x_1, x_2, x_3, x_4) = 0$$

Search domain

$$x_i \in [0, 1] \quad \forall i \in [1, \ldots, 4]$$

Global minimum

$$f(x^*) = 0.0516605 \quad \text{and} \quad x^* = [0.0, 0.0, 0.0, 0.0516605]$$

References

[1] C.A. Mader, J.R.R.A. Martins, J.J. Alonso, E. van der Weide, ADjoint: an approach for the rapid development of discrete adjoint solvers, AIAA J. 46 (4) (2008) 863–873, https://doi.org/10.2514/1.29123.

[2] J.R.R.A. Martins, J.T. Hwang, Review and unification of methods for computing derivatives of multidisciplinary computational models, AIAA J. 51 (11) (2013) 2582–2599, https://doi.org/10.2514/1.J052184.

[3] G.K.W. Kenway, J.R.R.A. Martins, Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration, J. Aircr. 51 (1) (2014) 144–160, https://doi.org/10.2514/1.C032150.

[4] Z. Lyu, G.K.W. Kenway, J.R.R.A. Martins, Aerodynamic shape optimization investigations of the common research model wing benchmark, AIAA J. 53 (4) (2015) 968–985, https://doi.org/10.2514/1.J053318.

[5] Y. Yu, Z. Lyu, Z. Xu, J.R.R.A. Martins, On the influence of optimization algorithm and starting design on wing aerodynamic shape optimization, Aerosp. Sci. Technol. 75 (2018) 183–199, https://doi.org/10.1016/j.ast.2018.01.016.

[6] N. Bons, X. He, C.A. Mader, J.R.R.A. Martins, Multimodality in aerodynamic wing design optimization, AIAA J. 57 (3) (2019) 1004–1018, https://doi.org/10.2514/1.J057294.

[7] A.R. Conn, K. Scheinberg, L.N. Vicente, Introduction to Derivative-Free Optimization, vol. 8, SIAM, 2009.

[8] L.M. Rios, N.V. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations, J. Glob. Optim. 56 (3) (2013) 1247–1293.

[9] F. Boukouvala, R. Misener, C.A. Floudas, Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO, Eur. J. Oper. Res. 252 (3) (2016) 701–727.

[10] C. Audet, W. Hare, Derivative-Free and Blackbox Optimization, Springer, 2017.

[11] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[12] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18.

[13] N. Hansen, S. Kern, Evaluating the CMA Evolution Strategy on Multimodal Test Functions, PPSN, vol. 8, Springer, 2004, pp. 282–291.

[14] G.A. Jastrebski, D.V. Arnold, Improving evolution strategies through active covariance matrix adaptation, in: 2006 IEEE Congress on Evolutionary Computation, CEC 2006, IEEE, 2006, pp. 2814–2821.

[15] Y. Diouane, S. Gratton, L.N. Vicente, Globally convergent evolution strategies for constrained optimization, Comput. Optim. Appl. 62 (2) (2015) 323–346.

[16] S. Wessing, M. Preuss, The true destination of EGO is multi-local optimization, in: 20017 IEEE Latin American Conference on Computational Intelligence, LA-CCI, IEEE, 2017, pp. 1–6.

[17] S. Le Digabel, Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm, ACM Trans. Math. Softw. 37 (4) (2011) 44.

[18] T.D. Plantenga, Hopspack 2.0 user manual, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, SAND2009–SAND6265.

[19] G. Fasano, G. Liuzzi, S. Lucidi, F. Rinaldi, A linesearch-based derivative-free approach for nonsmooth constrained optimization, SIAM J. Optim. 24 (3) (2014) 959–992.

[20] J. Močkus, On Bayesian methods for seeking the extremum, in: Optimization Techniques IFIP Technical Conference, 1975, pp. 400–404.

[21] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Glob. Optim. 13 (4) (1998) 455–492.

[22] A. Forrester, A. Sobester, A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, John Wiley & Sons, 2008.

[23] Z. Wang, S. Jegelka, Max-value entropy search for efficient Bayesian optimization, in: 34th International Conference on Machine Learning, vol. 70, 2017, pp. 3627–3635.

[24] M.A. Gelbart, Constrained Bayesian Optimization and Applications, Ph.D. thesis, Harvard, 2015.

[25] V. Picheny, A stepwise uncertainty reduction approach to constrained global optimization, in: AISTATS, 2014, pp. 787–795.

[26] V. Picheny, R.B. Gramacy, S. Wild, S. Le Digabel, Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian, in: Advances in Neural Information Processing Systems, 2016, pp. 1435–1443.

[27] S. Shan, G.G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, Struct. Multidiscip. Optim. 41 (2) (2010) 219–241.

[28] A.I. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (1) (2009) 50–79.

[29] G.G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, J. Mech. Des. 129 (4) (2007) 370–380.

[30] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P.K. Tucker, Surrogate-based analysis and optimization, Prog. Aerosp. Sci. 41 (1) (2005) 1–28.

[31] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, vol. 1, MIT Press, Cambridge, 2006.

[32] G. Matheron, Principles of geostatistics, Econ. Geol. 58 (8) (1963) 1246–1266.

[33] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. (1989) 409–423.

[34] J.P. Kleijnen, Kriging metamodeling in simulation: a review, Eur. J. Oper. Res. 192 (3) (2009) 707–716.

[35] M.J. Sasena, P.Y. Papalambros, P. Goovaerts, The use of surrogate modeling algorithms to exploit disparities in function computation time within simulation-based optimization, in: The Fourth World Congress of Structural and Multidisciplinary Optimization, 2001, pp. 1–6.

[36] F. Palacios, J.J. Alonso, M. Colonno, J. Hicken, T. Lukaczyk, Adjoint-based method for supersonic aircraft design using equivalent area distributions, AIAA Pap. 269 (2012) 2012.

[37] Z.-H. Han, R. Zimmermann, S. Görtz, A New Cokriging Method for Variable-Fidelity Surrogate Modeling of Aerodynamic Data, AIAA Paper 1225, 2010, p. 2010.

[38] T. Benamara, P. Breitkopf, I. Lepot, C. Sainvitu, Adaptive infill sampling criterion for multi-fidelity optimization based on Gappy-POD, Struct. Multidiscip. Optim. 54 (4) (2016) 843–855.

[39] 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, Struct. Multidiscip. Optim. 55 (3) (2017) 925–943.

[40] Y. Ma, W. Zhou, Q. Han, Research of multi-point infill criteria based on multi-objective optimization front and its application on aerodynamic shape optimization, Adv. Mech. Eng. 9 (6) (2017) 1687814017703340.

[41] M.A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction, Struct. Multidiscip. Optim. (ISSN 1615-1488) 53 (5) (2016) 935–952, https://doi.org/10.1007/s00158-015-1395-9.

[42] M.A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, An improved approach for estimating the hyperparameters of the Kriging model for high-dimensional problems through the partial least squares method, Math. Probl. Eng. (2016).

[43] M.A. Bouhlel, N. Bartoli, R.G. Regis, A. Otsmane, J. Morlier, Efficient global optimization for high-dimensional constrained problems by using the Kriging models combined with the partial least squares method, Eng. Optim. (2018) 1–16, https://doi.org/10.1080/0305215X.2017.1419344.

[44] T. Hastie, R. Tibshirani, J. Friedman, J. Franklin, The elements of statistical learning: data mining, inference and prediction, Math. Intell. 27 (2) (2005) 83–85.

[45] D. Bettebghor, N. Bartoli, S. Grihon, J. Morlier, M. Samuelides, Surrogate modeling approximation using a mixture of experts based on EM joint estimation, Struct. Multidiscip. Optim. (ISSN 1615-147X) 43 (2) (2011) 243–259, https://doi.org/10.1007/s00158-010-0554-2.

[46] Y. Tenne, An optimization algorithm employing multiple metamodels and optimizers, Int. J. Autom. Comput. 10 (3) (2013) 227–241.

[47] M. Sasena, P. Papalambros, P. Goovaerts, Global optimization of problems with disconnected feasible regions via surrogate modeling, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, 2002, Paper No. AIAA-2002-5573.

[48] S. Chen, Z. Jiang, S. Yang, W. Chen, Multimodel fusion based sequential optimization, AIAA J. 55 (1) (2016) 241–254.

[49] N. Bartoli, M.-A. Bouhlel, I. Kurek, R. Lafage, T. Lefebvre, J. Morlier, R. Priem, V. Stilz, R. Regis, Improvement of efficient global optimization with application to aircraft wing design, in: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Washington DC, USA, 2016, p. 4001.

[50] AIAA Aerodynamic Design Optimization Discussion Group, http://mdolab.engin.umich.edu/content/aerodynamic-design-optimization-workshop, 2019.

[51] D.G. Krige, A Statistical Approach to Some Mine Evaluations and Allied Problems at the Witwatersrand, Master's thesis, University of Witwatersrand, 1951.

[52] M.J. Sasena, P. Papalambros, P. Goovaerts, Exploration of metamodeling sampling criteria for constrained global optimization, Eng. Optim. 34 (3) (2002) 263–278, https://doi.org/10.1080/03052150211751.

[53] M. Schonlau, W.J. Welch, D.R. Jones, Global Versus Local Search in Constrained Optimization of Computer Models, Lecture Notes-Monograph Series, 1998, pp. 11–25.

[54] C. Audet, J. Denni, D. Moore, A. Booker, P. Frank, A surrogate-model-based method for constrained optimization, in: 8th Symposium on Multidisciplinary Analysis and Optimization, 2000, p. 4891.

[55] J.M. Hernández-Lobato, M.A. Gelbart, R.P. Adams, M.W. Hoffman, Z. Ghahramani, A general framework for constrained Bayesian optimization using information-based search, J. Mach. Learn. Res. 17 (160) (2016) 1–53.

[56] R. Priem, N. Bartoli, Y. Diouane, S. Dubreuil, An adaptive feasibility approach for constrained bayesian optimization with application in aircraft design, in: Engopt: 6th International Conference on Engineering Optimization, Lisboa, Portugal, 2018.

[57] A.G. Watson, R.J. Barnes, Infill sampling criteria to locate extremes, Math. Geol. 27 (5) (1995) 589–608.

[58] M. Sasena, Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations, Ph.D. thesis, University of Michigan, 2002.

[59] D. Bettebghor, N. Bartoli, Approximation of the critical buckling factor for composite panels, Struct. Multidiscip. Optim. 46 (4) (2012) 561–584, http://www.springerlink.com/content/e2855877v3x855t7/.

[60] R.P. Liem, C.A. Mader, J.R.R.A. Martins, Surrogate models and mixtures of experts in aerodynamic performance prediction for mission analysis, Aerosp. Sci. Technol. 43 (2015) 126–151, https://doi.org/10.1016/j.ast.2015.02.019.

[61] M.A. Bouhlel, J.T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J.R.R.A. Martins, A Python surrogate modeling framework with derivatives, Adv. Eng. Softw., https://doi.org/10.1016/j.advengsoft.2019.03.005.

[62] P.S. Bradley, U.M. Fayyad, C. Reina, et al., Scaling clustering algorithms to large databases, in: KDD, 1998, pp. 9–15.

[63] M.F. Anjos, D.R. Jones, MOPTA 2008 benchmark, http://www.miguelanjos.com/jones-benchmark, 2009.

[64] J.S. Gray, J.T. Hwang, J.R.R.A. Martins, K.T. Moore, B.A. Naylor, OpenMDAO: an open-source framework for multidisciplinary design, in: analysis, and optimization, Struct. Multidiscipl. Optim. (2019), https://doi.org/10.1007/s00158-019-02211-z.

[65] J. Parr, A. Keane, A.I. Forrester, C. Holden, Infill sampling criteria for surrogate-based optimization with constraint handling, Eng. Optim. 44 (10) (2012) 1147–1166.

[66] M.J. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, in: Advances in Optimization and Numerical Analysis, Springer, 1994, pp. 51–67.

[67] C. Audet, S. Le Digabel, C. Tribes, NOMAD User Guide, Tech. Rep. G-2009-37, Les cahiers du GERAD, 2009, https://www.gerad.ca/nomad/Downloads/user_guide.pdf.

[68] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, ISBN 978-0-387-30303-1, 2006.

[69] R.B. Gramacy, G.A. Gray, S. Le Digabel, H.K. Lee, P. Ranjan, G. Wells, S.M. Wild, Modeling an augmented Lagrangian for blackbox constrained optimization, Technometrics 58 (2016) 1–11.

[70] R. Jin, W. Chen, A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments, J. Stat. Plan. Inference 134 (1) (2005) 268–287.

[71] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: open source scientific tools for Python, http://www.scipy.org/, 2001. (Accessed 27 April 2016) [Online].

[72] M. Abramson, C. Audet, G. Couture, J. Dennis Jr., S. Le Digabel, C. Tribes, The NOMAD project, Software available at https://www.gerad.ca/nomad/, 2009.

[73] V. Picheny, D. Ginsbourger, O. Roustant, DiceOptim: Kriging-based optimization for computer experiments, https://github.com/cran/DiceOptim, 2014.

[74] D. Kraft, et al., A Software Package for Sequential Quadratic Programming, DFVLR Obersfaffeuhofen, Germany, 1988.

[75] C. Audet, J.E. Dennis Jr., A progressive barrier for derivative-free nonlinear programming, SIAM J. Optim. 20 (1) (2009) 445–472.

[76] G.K.W. Kenway, N. Secco, J.R.R.A. Martins, A. Mishra, K. Duraisamy, An efficient parallel overset method for aerodynamic shape optimization, in: Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum, Grapevine, TX, 2017.

[77] Z. Lyu, G.K. Kenway, C. Paige, J.R.R.A. Martins, Automatic differentiation adjoint of the Reynolds-averaged Navier–Stokes equations with a turbulence model, in: 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, 2013.

[78] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, 1992.

[79] G.K. Kenway, G.J. Kennedy, J.R.R.A. Martins, A CAD-free approach to high-fidelity aerostructural optimization, in: Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, AIAA 2010-9231, 2010.

[80] E. Luke, E. Collins, E. Blades, A fast mesh deformation method using explicit interpolation, J. Comput. Phys. (ISSN 0021-9991) 231 (2) (2012) 586–601, https://doi.org/10.1016/j.jcp.2011.09.021.

[81] R.E. Perez, P.W. Jansen, J.R.R.A. Martins, pyOpt: a python-based object-oriented framework for nonlinear constrained optimization, Struct. Multidiscipl. Optim. 45 (1) (2012) 101–118, https://doi.org/10.1007/s00158-011-0666-3.

[82] G.K.W. Kenway, J.R.R.A. Martins, pyOptSparse – PYthon OPTimization (sparse) framework, https://github.com/mdolab/pyoptsparse, 2018.

[83] P.E. Gill, W. Murray, M.A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, SIAM Rev. 47 (1) (2005) 99–131.

[84] G.K.W. Kenway, J.R.R.A. Martins, Buffet onset constraint formulation for aerodynamic shape optimization, AIAA J. 55 (6) (2017) 1930–1947, https://doi.org/10.2514/1.J055172.

[85] R.P. Liem, J.R.R.A. Martins, G.K. Kenway, Expected drag minimization for aerodynamic design optimization based on aircraft operational data, Aerosp. Sci. Technol. 63 (2017) 344–362, https://doi.org/10.1016/j.ast.2017.01.006.

[86] S. Chen, Z. Lyu, G.K.W. Kenway, J.R.R.A. Martins, Aerodynamic shape optimization of the common research model wing-body-tail configuration, J. Aircr. 53 (1) (2016) 276–293, https://doi.org/10.2514/1.C033328.

[87] G.K.W. Kenway, J.R.R.A. Martins, Multipoint aerodynamic shape optimization investigations of the common research model wing, AIAA J. 54 (1) (2016) 113–128, https://doi.org/10.2514/1.J054154.

[88] G.K.W. Kenway, G.J. Kennedy, J.R.R.A. Martins, Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and derivative computations, AIAA J. 52 (5) (2014) 935–951, https://doi.org/10.2514/1.J052255.

[89] T.R. Brooks, G.K.W. Kenway, J.R.R.A. Martins, Benchmark aerostructural models for the study of transonic aircraft wings, AIAA J. 56 (7) (2018) 2840–2855, https://doi.org/10.2514/1.J056603.

[90] D.A. Burdette, J.R.R.A. Martins, Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization, Aerosp. Sci. Technol. 81 (2018) 192–203, https://doi.org/10.1016/j.ast.2018.08.004.