Les Cahiers du GERAD ISSN: 0711–2440

Order-based error for managing ensembles of surrogates in mesh adaptive direct search

C. Audet, M. Kokkolaras
S. Le Digabel, B. Talgorn

G–2016–36

May 2016
Revised: February 2018

CITATION ORIGINALE / ORIGINAL CITATION

C. Audet, M. Kokkolaras, S. Le Digabel, B. Talgorn, Order-based error for managing ensembles of surrogates in mesh adaptive direct search, Journal of Global Optimization, 70, 645–675, 2018.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Order-based error for managing ensembles of surrogates in mesh adaptive direct search

Charles Audet [a]
Michael Kokkolaras [b]
Sébastien Le Digabel [a]
Bastien Talgorn [b]

[a] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec), Canada*

[b] *GERAD & Department of Mechanical Engineering, McGill University, Montréal (Québec), Canada*

charles.audet@gerad.ca
michael.kokkolaras@mcgill.ca
sebastien.le.digabel@gerad.ca
bastien.talgorn@mail.mcgill.ca

May 2016
Revised: February 2018
Les Cahiers du GERAD
G–2016–36

Abstract: We investigate surrogate-assisted strategies for global derivative-free optimization using the mesh adaptive direct search (MADS) blackbox optimization algorithm. In particular, we build an ensemble of surrogate models to be used within the search step of MADS to perform global exploration, and examine different methods for selecting the best model for a given problem at hand. To do so, we introduce an order-based error tailored to surrogate-based search. We report computational experiments for ten analytical benchmark problems and three engineering design applications. Results demonstrate that different metrics may result in different model choices and that the use of order-based metrics improves performance.

Keywords: Derivate-free optimization, ensemble of surrogates, MADS, order error

---

Acknowledgments: This work has been supported partially by FRQNT grant 2015-PR-182098; B. Talgorn and M. Kokkolaras are grateful for the partial support of NSERC/Hydro-Québec grant EGP2 498903-16; such support does not constitute an endorsement by the sponsors of the opinions expressed in this article. The authors would like to thank Robin Tournemenne (École Centrale de Nantes) for his insightful questions and comments that contributed significantly to improve the search algorithm used in this work. The authors would also like to express their gratitude to Stéphane Alarie (IREQ) for his support, which made this work possible.

1 Introduction

In many engineering design optimization problems, the objective and constraint functions are evaluated using simulation models. These models are often not accessible by the user, and so the structure of the optimization problem cannot be identified in order to be exploited. Such problems are called blackbox optimization problems [6]. In addition, derivatives may either be unavailable or require significant effort to be approximated reliably. In this case, derivative-free optimization algorithms [20] may offer a valuable alternative to gradient-based techniques. Finally, simulations may be computationally expensive. In such cases, surrogate models of the blackboxes may be built and used to obtain information in a more robust and inexpensive, yet possibly less accurate, manner.

We consider the constrained engineering design optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}} & f(\mathbf{x}) \\ \text{s.t.} & c_j(\mathbf{x}) \leq 0, \quad j \in J = \{1, 2, \ldots, m\}, \end{array} \tag{P}$$

where $\mathcal{X}$ is a subset of $\mathbb{R}^n$ typically defined by bound constraints $\underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}$ and the functions f and c_j, $j \in J$, are evaluated by means of blackbox simulation models. We employ the mesh-adaptive direct search (MADS) algorithm [9] that relies on the search-and-poll paradigm [12]. The poll rigorously ensures global convergence toward a solution satisfying some local optimality conditions, while the search can implement any method likely to improve the efficiency and global exploration of the optimization process in order to escape local optima.

Our research focuses on developing surrogate-assisted strategies integrated in MADS. We build/update surrogate models that are used to obtain a next-iterate candidate by solving a surrogate optimization problem in the search step. We also use the surrogate models to rank the candidates proposed by the poll step. Then, we select the next iterate by evaluating the blackbox simulation at some carefully chosen trial points. In this manner, we generate as much information as possible using the inexpensive and robust surrogate models while making decisions using the blackbox simulation models. This approach has served as the foundation for a body of literature that includes various types of search methods: Latin hypercube-based design of experiments [49], variable neighborhood search [7], particle swarms [71], genetic algorithms [31] and surrogate-based [12, 19, 22, 33].

The principle of a surrogate-based search is to use previously evaluated points for building and calibrating surrogate models to predict blackbox output at new trial points. The idea is that a minimizer of the objective function surrogate subject to constraint surrogates will be a potentially interesting candidate for the original optimization Problem (P). Other approaches that are not considered in this work consist of computing the Expected Improvement [8, 36] of a design, and/or considering a diversification term that will favor unexplored areas of the design space [64, 65].

The search step therefore consists of solving the *surrogate problem*, i.e., the minimization of the surrogate of the objective function subject to the constraint surrogates. To find a minimizer of this problem, an inner instance of MADS is used which relies heavily on Variable Neighborhood Search (VNS) [7] and Latin Hypercube (LH) sampling [49]. In our computational experiments in Section 5, a large budget of model evaluations (10,000) is allocated for solving the surrogate problem at each search step, which favors a global solution of the surrogate problem. As a consequence, the main instance of MADS is more likely to converge towards a global minimizer of the main problem, however, this convergence is contingent upon the accuracy of the chosen surrogate models.

The challenge is to build surrogate models that adequately approximate the blackbox outputs. Commonly used models include Gaussian processes [56, 58], in particular Kriging models [12, 16, 23, 36]. While these models may be very useful, the process of choosing their parameters can be tedious and generate significant computational overhead. Another common surrogate approach is to build local quadratic models [19, 21, 24], which rely on trust regions [4, 18, 26]. Finally, other surrogate modeling techniques include radial basis functions [38, 42, 59, 60, 75], splines [74] and kernel smoothing [2, 34].

Several studies have shown that no single surrogate modeling technique can be deemed the best on a consistent basis [2, 25, 27, 29]. To address this issue, our approach is to build an ensemble of surrogates, in which several surrogate models of different types are constructed [2, 13, 14, 29, 43, 45, 53, 54, 67, 72, 73]. The quality of each model is assessed to either select the best for each individual blackbox output, or to construct an aggregate model using a weighted combination of all surrogate models. A fundamental aspect of an ensemble of surrogates is that the models are built without any attempt to fine-tune their parameters. In the approaches proposed so far to compute aggregate model weights, [29] uses four different empirical methods to determine them directly from error metrics. On the contrary, [2] and [72] compute the weights to minimize error metrics.

In this work, we select the best surrogate model according to an error metric. Commonly used metrics include the Root Mean Square Error (RMSE) and the Predicted Residual Sum of Squares (PRESS) [5, 29, 30, 53]. However, our opinion is that RMSE or PRESS may not be the most suitable metrics for selecting models in a surrogate-based optimization (SBO) context. For example, metrics based on sums of squares will penalize models that do not fit outliers well, even though oftentimes these outliers are not critical with respect to the optimizer. In general, model accuracy (favored by RMSE or PRESS metrics) may not be as critical as the location of the optimizer; for example, the surrogate $af(\mathbf{x}) + b$ of $f(\mathbf{x})$ with $a > 0$ is perfectly suitable for SBO even though it has a high RMSE or PRESS value. Our main concern in SBO is to find the correct optimizer, not necessarily the correct optimum. Therefore, we propose a novel class of order-based error metrics that is tailored specifically for SBO; these metrics quantify the ability of a model to predict which of two points has the best objective function, or to predict whether a given point is feasible.

The paper is structured as follows. Section 2 presents a high-level description of the MADS optimization algorithm, and surrogate-based search. Section 3 provides an overview of modeling techniques used to build ensembles of surrogates. In Section 4, ensembles of surrogates are discussed in more detail along with several error metrics (including the newly proposed order-based error metric) used to assess predictive capabilities. Section 5 demonstrates the performance of the new metric by means of several computational experiments on a set of analytical problems, on three engineering design applications. Concluding remarks are drawn in Section 6.

2 The MADS algorithm

At each iteration t of MADS, several candidates are evaluated. To guarantee the convergence of the algorithm towards a solution satisfying some optimality conditions, all candidates must lie on a discretization of the design space called the mesh and defined, at iteration t, by

$$\mathcal{M}_t = \{\mathbf{x} + \Delta_t^m \mathbf{D}\mathbf{z}, \mathbf{z} \in \mathbb{N}^{n_D}, \mathbf{x} \in \mathbf{X}_t\}$$

where $\Delta_t^m \in \mathbb{R}_+$ is the mesh size parameter at iteration t, $\mathbf{X}_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\} \subset \mathcal{X}$ is the set of points evaluated previously, and $\mathbf{D}$ is a matrix whose columns form a positive spanning set of n_D directions in $\mathbb{R}^n$.

2.1 Search and poll

Each MADS iteration consists of an optional search and a mandatory poll step. We use the search step to solve a surrogate optimization problem, i.e., solve the optimization problem using surrogate models of the objective and constraints. The training points $\mathbf{X}_t$ are used to build a surrogate model for each of the blackbox outputs. For a given $\mathbf{x} \in \mathcal{X}$, the models denoted by $\hat{f}$ and $\hat{c}_j$ provide estimates of $f(\mathbf{x})$ and $c_j(\mathbf{x})$. The surrogate-based search is based on the premise that a solution of the surrogate problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}} & \hat{f}(\mathbf{x}) \\ \text{s.t.} & \hat{c}_j(\mathbf{x}) \leq 0, \quad j \in J, \end{array} \tag{$\hat{P}$}$$

will be a promising candidate of the original Problem (P). The search proceeds as follows: at each iteration of the MADS algorithm, the surrogate models are updated to account for the most recent observations available.

Then, a solution of Problem ($\hat{P}$) is computed and denoted by $\mathbf{x}_t^S$. It is the best feasible solution of ($\hat{P}$) or, if no feasible design was found, the best infeasible solution (i.e., the solution with the smallest constraint violation value, as defined in [10]).

The point $\mathbf{x}_t^S$ is then projected on the mesh $\mathcal{M}_t$ using the method described in Section 2.2; this process yields the point $\mathbf{x}_t^P$ which is then evaluated using the blackbox simulation that defines Problem (P). If this candidate leads to an improvement of the solution, the surrogate-based search is repeated. Otherwise, the algorithm proceeds to the poll step.

In the poll step, a set $S_{Poll,t}$ of candidates is built within a distance $\Delta_t^p > 0$ of the incumbent solution $\mathbf{x}_t^*$. Candidates of the poll step are ordered depending on how well they perform on the surrogate Problem ($\hat{P}$); they are then evaluated using the blackbox simulation. These evaluations are made opportunistically, which means that if a candidate leads to an improvement of the solution, the MADS iteration is considered a success and the other candidates of the poll are not evaluated. This opportunistic strategy implies that for each MADS iteration, the computational cost and the magnitude of the improvement both depend on the order in which the candidates of the poll step are evaluated. Specifically, the process of evaluating candidates from most to least promising allows for a larger improvement of the solution within fewer blackbox evaluations.

At the end of iteration t, the mesh size parameter Δ_{t+1}^m is updated based on the outcome of iteration t. The standard MADS algorithm [10] proposes to increase the mesh size parameter when the iteration is successful (i.e., when a point better than the incumbent solution $\mathbf{x}_t^*$ has been found) in order to accelerate the movement towards more interesting regions of the design space. However, in the case where the success of the iteration is obtained via the search steps, which performs global exploration, it is not relevant to increase the mesh size. Consequently, we adopt the following scheme: if the search is a success, the mesh size remains at the same value and the poll step is skipped (i.e., $\Delta_{t+1}^m \leftarrow \Delta_t^m$); if the poll is a success, the mesh size is increased (i.e. $\Delta_{t+1}^m \leftarrow 4\Delta_t^m$); finally, if the iteration is a failure, the mesh size is reduced (i.e. $\Delta_{t+1}^m \leftarrow 0.25\Delta_t^m$). This scheme is permitted in MADS [9]. An overview of MADS is provided in Algorithm 1.

2.2 Projection on the mesh

The search is required to propose finitely many candidates lying on the discrete mesh to preserve the convergence properties of MADS. However, we have observed that the search is often more efficient when the surrogate optimization process is allowed to evaluate points that do not lie on the mesh. Consequently, the candidate returned by the optimization of the surrogate problem is often not on the mesh and the last step of the surrogate-based search projects this point onto the mesh.

We denote $\mathbf{x}_t^*$ the incumbent solution of the original problem, $\mathbf{x}_t^S$ the non-projected solution of the surrogate problem optimization, and $\mathbf{x}_t^P$ the projection of $\mathbf{x}_t^S$ onto the mesh

$$\mathcal{M}_t = \bigcup_{\mathbf{x} \in \mathbf{X}_t} \mathcal{M}_{t,\mathbf{x}}.$$

where for each point $\mathbf{x}$ of $\mathbf{X}_t$, $\mathcal{M}_{t,\mathbf{x}} = \{\mathbf{x} + \Delta_t^m \mathbf{D}\mathbf{z}, \mathbf{z} \in \mathbb{N}^{n_D}\}$ is a submesh.

The projection of $\mathbf{u} \in \mathcal{X}$ on the submesh $\mathcal{M}_{t,\mathbf{x}}$ can then be defined as

$$\text{proj}_{\mathbf{x}}(\mathbf{u}) \in \underset{\mathbf{v} \in \mathcal{M}_{t,\mathbf{x}}}{\text{argmin}} \|\mathbf{u} - \mathbf{v}\|_1,$$

which can be performed by rounding operations on the components of $\mathbf{u}$ and $\mathbf{x}$.

In previous work, the candidate returned by the search was $\text{proj}_{\mathbf{x}_t^*}(\mathbf{x}_t^S)$, i.e., the projection of the surrogate problem solution on $\mathcal{M}_{t,\mathbf{x}_t^*}$ [19, 65]. As we generally have $\mathcal{M}_{t,\mathbf{x}_t^*} \subset \mathcal{M}_t$ with a strict inclusion, many interesting points that lie on the mesh $\mathcal{M}_t$ may be discarded during this projection step. Moreover, this definition of the projection may be misleading as the point of the mesh that is the closest to $\mathbf{x}_t^S$ is not necessarily the best candidate according to $\hat{f}$ and $\{\hat{c}\}_{j \in J}$.

Algorithm 1 The MADS optimization algorithm.

[1] Initialization
$t \leftarrow 0$
Set initial poll and mesh sizes $\Delta_0^p \geq \Delta_0^m > 0$
Initialize $\mathbf{X}_0$ with starting points
Evaluate $f(\mathbf{x})$ and $\{c_j(\mathbf{x})\}_{j\in J}$ for all $\mathbf{x} \in \mathbf{X}_0$
[2] Search
Use $\mathbf{X}_t$ to build $\hat{f}$ and $\{\hat{c}_j\}_{j\in J}$
$\mathbf{x}_t^S \leftarrow$ Solution of $(\hat{P})$
$\mathbf{x}_t^P \leftarrow$ Projection of $\mathbf{x}_t^S$ onto mesh $\mathcal{M}_t$
Evaluate $f(\mathbf{x}_t^P)$ and $\{c_j(\mathbf{x}_t^P)\}_{j\in J}$
If success, `goto` **[4]**
[3] Poll
Build poll set $S_{Poll,t}$
Sort $S_{Poll,t}$ according to $\hat{f}$ and $\{\hat{c}_j\}_{j\in J}$
for $\mathbf{x} \in S_{Poll,t}$
  Evaluate $f(\mathbf{x})$ and $\{c_j(\mathbf{x})\}_{j\in J}$
  If success, `goto` **[4]**
end
[4] Updates
$t \leftarrow t + 1$
Update Δ_t^m, Δ_t^p, $\mathcal{M}_t$, x_t^* and $\mathbf{X}_t$
If no stopping condition is met, `goto` **[2]**

To address these problems, we propose a novel projection step which consists of the following four steps. First, we consider the set of perturbation vectors

$$\Pi = \{\mathbf{u} \in \mathbb{R}^n : u_i = \pm\Delta_t^m,\ i = 1, 2, \ldots, n\} \subset \mathbb{R}^n,$$

and we build the set $S_\Pi \subset \Pi$ by drawing randomly $\min\{2^n, 100n\}$ different vectors of Π. These vectors allow to perturb the design $\mathbf{x}_t^S$ so that not only the closest point to $\mathbf{x}_t^S$ is considered during the projection. Secondly, we build a set of projection candidates

$$S_{Proj} = \{\mathbf{x} : \mathbf{x} = \text{proj}_{\mathbf{x}'}(\mathbf{x}_t^P + \mathbf{u}), \mathbf{u} \in S_\Pi, \mathbf{x}' \in \mathbf{X}\}.$$

Note that there may exist several couples $(\mathbf{u}, \mathbf{x}')$ that lead to the same value of $\text{proj}_{\mathbf{x}'}(\mathbf{x}_t^S + \mathbf{u})$, so the training of this set includes the suppression of duplications. Even so, it can be prohibitive to compute $\hat{f}(\mathbf{x})$ and $\{\hat{c}(\mathbf{x})\}_{j\in J}$ for all $\mathbf{x} \in S_{Proj}$, so the third step of the projection is to perform a greedy selection with Algorithm 2 (see Appendix A) to reduce the size of this set to no more than $100n$ candidates. While using this algorithm, we will favor points close to $\mathbf{x}_t^S$. Finally, all the points of S_{Proj} are evaluated with $\hat{f}$ and $\{\hat{c}\}_{j\in J}$, and the best feasible design (or, if unavailable, the best infeasible design) is selected as the candidate $\mathbf{x}_t^P$ of the search step at iteration t of MADS.

3 Overview of selected surrogate modeling techniques

From a set of p observations $[\mathbf{X}, \mathbf{y}]$, with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\} \subset \mathbb{R}^n$ and $\mathbf{y}_i = y(\mathbf{x}_i)$, where $y : \mathbb{R}^n \to \mathbb{R}$ is either the objective function f or one of the constraint functions $\{c_j\}_{j\in J}$, it is possible to build a surrogate model $\hat{y}$ which can be used to predict the value of $y(\mathbf{x})$ for $\mathbf{x} \notin \mathbf{X}$. In this section, we describe three types of surrogate models and how to use them to build an ensemble of surrogates: Polynomial response surfaces (PRSs) [2, 53, 57], kernel smoothing (KS) [2, 34], and radial basis functions (RBFs) [2, 15, 53, 57, 59, 60, 73].

As discussed in Section 4.1, it is useful to have access to cross-validation values in order to quantify the predictive capability of a surrogate model. These consist of the values that $\hat{y}(\mathbf{x}_i)$ would have taken if the model $\hat{y}$ had been built without the observation $[\mathbf{x}_i, y(\mathbf{x}_i)]$. Thus, we define $\hat{y}^{(-i)}$ the surrogate model built by leaving out the observation $[\mathbf{x}_i, y(\mathbf{x}_i)]$ for $\mathbf{x}_i \in \mathbf{X}$. We also define the cross-validation vector $\hat{\mathbf{y}}^{cv}$ such that $\hat{y}_i^{cv} = \hat{y}^{(-i)}(\mathbf{x}_i)$.

Since PRS and RBF models both fall in the category of linear models (not to be confused with linear regression), Section 3.1 provides an overview of these types of models.

3.1 Linear models

A model $\hat{y}$ is said to be linear if it can be expressed as a linear combination of basis functions [28, 46, 55, 62]

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{q} \alpha_j h_j(\mathbf{x}),$$

where $h_j(x) : \mathbb{R}^n \to \mathbb{R}$ is a (possibly non-linear) basis function. The coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_q]^\top$ are computed to minimize the regularized quadratic error

$$\sum_{\mathbf{x}\in\mathbf{X}} (y(\mathbf{x}) - \hat{y}(\mathbf{x}))^2 + r_{ridge}\|\boldsymbol{\alpha}\|_2^2,$$

where $r_{ridge} \geq 0$ is a regularization (or ridge [55]) parameter. This parameter ensures that the linear system is invertible, even if there are more basis functions than training points or if the training points are aligned. In case the number of training points does not exceed the number of basis functions ($p \leq q$), the model is constructed only if the ridge coefficient is not equal to zero. Otherwise, the model is considered as not ready and will be built only when more training points are available.

For such models, the *design matrix* $\mathbf{H}$ can be built from the training points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\}$

$$\mathbf{H} = \begin{bmatrix} h_1(\mathbf{x}_1) & \ldots & h_q(\mathbf{x}_1) \\ \vdots & & \vdots \\ h_1(\mathbf{x}_p) & \ldots & h_q(\mathbf{x}_p) \end{bmatrix}.$$

Note that the matrix $\mathbf{H}$ is independent from the output $y(\mathbf{X})$. To solve this kind of problem, we consider only the cases where the number of training points is larger than the number of basis functions (i.e., the number of unknown coefficients) or where the ridge coefficient is not equal to zero. In these two cases, the use of Ordinary Least Squares (OLS) is required. The coefficients $\boldsymbol{\alpha}$ are the solution of the invertible problem

$$\underbrace{(\mathbf{H}^\top\mathbf{H} + r_{ridge}\mathbf{I}_q)}_{\mathbf{A}}\boldsymbol{\alpha} = \mathbf{H}^\top\mathbf{y}.$$

It follows that

$$\boldsymbol{\alpha} = \mathbf{A}^{-1}\mathbf{H}^\top\mathbf{y}.$$

An interesting property of this kind of model is that the cross-validation vector can be computed by

$$\hat{\mathbf{y}}^{cv} = \mathbf{y} - diag(\mathbf{P})^{-1}\mathbf{P}\mathbf{y},$$

where $\mathbf{P}$ is the *projection* matrix [55] such that $\mathbf{P} = \mathbf{I}_p - \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top$ and $diag(\mathbf{P})$ is a diagonal matrix such that $diag(\mathbf{P})_{ii} = P_{ii}$. Once the design matrices $\mathbf{H}$ and $\mathbf{A}$ are built and $\mathbf{A}^{-1}$ has been computed for a given function y, it is inexpensive to compute the coefficients $\boldsymbol{\alpha}$ for other functions $\mathbf{y}$ (for example, each output of the blackbox simulation) as well as the cross-validation values.

3.1.1 Polynomial response surfaces

Polynomial response surfaces are linear models for which the basis functions are polynomials. For a PRS of degree d, the set of basis functions $\{h_j\}_{j=1,2,\ldots,q}$ is a basis of the polynomial vector space of degree d in $\mathbb{R}^n$.

3.1.2 Radial basis functions

RBF models rely on basis functions of the form $h_j(\mathbf{x}) = \phi(d(\mathbf{x}_j, \mathbf{x}))$, where d is a distance function (in our case the Euclidean distance) and $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a kernel function. The coefficients of the RBF models are typically obtained by solving the linear system [2, 15, 30, 55, 73]

$$\begin{bmatrix} \mathbf{H}^{RBF} & \mathbf{H}^{PRS} \\ (\mathbf{H}^{PRS})^\top & 0 \end{bmatrix} \boldsymbol{\alpha} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}, \tag{1}$$

where $\mathbf{H}^{RBF}$ is a symmetric matrix such that $H_{ij} = \phi(d(\mathbf{x}_i, \mathbf{x}_j))$ and $\mathbf{H}^{PRS}$ is the design matrix of a PRS of degree 1.

The main disadvantage of this method is that the computational cost of building an RBF model can become prohibitive for a large number of training points. We propose a more efficient method which we have named RBFI (where "I" stands for "Incomplete"). RBFIs are based on the reduction of the number of basis functions by carefully selecting a subset of the training points $S(\mathbf{X}) = \{\mathbf{x}_i^s\}_{i=1,2,\ldots,q^{RBF}} \subset \mathbf{X}$. This subset contains $q^{RBF} < p$ training points of $\mathbf{X}$, which will be used as centers of the radial basis functions. The matrix $\mathbf{H}^{RBF}$ is then defined in $\mathbb{R}^{p \times q^{RBF}}$ such that $H_{ij}^{RBF} = \phi(d(\mathbf{x}_i, \mathbf{x}_j^s))$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q^{RBF}$. The design matrix is defined as $\mathbf{H} = [\mathbf{H}^{RBF}\ \mathbf{H}^{PRS}]$, where $\mathbf{H}^{PRS}$ is the PRS design matrix of size $p \times q^{PRS}$. Consequently, the total number of basis functions in this model is $q = q^{RBF} + q^{PRS}$. We chose $q^{RBF} = \min\{p/2, 10n\}$. That means that only q^{RBF} radial basis function are selected; however, they are merged with q^{PRS} PRS basis functions. The PRS used in this work is of degree 1. That means that $q^{PRS} = n + 1$. As the system is overdetermined, we do not need to add orthogonality constraints as in Equation (1). The coefficients $\boldsymbol{\alpha}$ are computed from the normal equations.

To build the set $S(\mathbf{X})$, we use a greedy algorithm that is computationally efficient and robust in selecting q^{RBF} points in the set $\mathbf{X}$. This algorithm is based on two requirements. First, the selected points must be spread across the design space. To do so, points are added greedily to $S(\mathbf{X})$ by selecting the point of $\mathbf{X}$ that maximizes the distance to $S(\mathbf{X})$. Secondly, to allow a better representation of the model in areas of interest, the number of selected points must be larger next to the incumbent solution $\mathbf{x}^*$. The greedy selection algorithm is presented in Appendix A. It takes as input the training set $\mathbf{X}$, the incumbent solution x^*, as well as the desired number of selected points q^{RBF}.

We use two types of kernel in this study. First, the Gaussian kernel $\phi(d) = \exp\left(-\frac{r_\phi^2 d^2}{d_{mean}^2}\right)$, where d_{mean} is the mean Euclidean distance between each pair of points of $S(\mathbf{X})$ and r_ϕ is a parameter of the model. Secondly, we use the poly-harmonic kernels of degree 1 ($\phi(d) = d$) and degree 2 ($\phi(d) = \log(d)d^2$, with $\phi(0) = 0$).

3.2 Kernel smoothing

Kernel smoothing models consist of a weighted sum of the training points, where the weight decreases with the distance to the training point:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{p} \phi(d(\mathbf{x}, \mathbf{x}_i)) y_i}{\sum_{i=1}^{p} \phi(d(\mathbf{x}, \mathbf{x}_i))}.$$

The advantage of KS is that the computation is immediate. It does not require a linear system inversion. One of the drawbacks is that KS rarely respects the training set, and has a tendency to "undershoot", i.e., low values may be estimated higher and high values may be estimated lower. However, despite their tendency to undershoot, we observe that KS models typically tend to respect the order of the output, which means that they are able to accurately predict which of two points has the best objective function value. Kernel smoothing predictions are computed using the same Gaussian kernel as for RBFI models, except that d_{mean} is the mean Euclidean distance between each pair of points of $\mathbf{X}$.

4 Ensembles of surrogates

For a function y to be modelled, we build a set of k_{max} surrogate models $\hat{y}_k$. These models can then be aggregated to a single model by

$$\hat{y}(\mathbf{x}) = \sum_{k=1}^{k_{max}} w_k \hat{y}_k(\mathbf{x}),$$

where $\mathbf{w} = [w_1, w_2, \ldots, w_{k_{max}}]$ is a weight vector such that $w_k \geq 0$ and $\sum_{k=1}^{k_{max}} w_k = 1$.

Many approaches have been proposed to choose the weights $\mathbf{w}$. Most of them rely on calculating an error metric $\mathcal{E}_k$ for each surrogate $\hat{y}_k$, and then setting $w_k \propto g(\mathcal{E}_k) \geq 0$, where g depends on the chosen method [2, 29, 72]. For example, [29] proposes three weight selection methods WTA1, WTA2 and WTA3:

$$\begin{aligned} \text{WTA1}: &\quad w_k \propto \mathcal{E}_{sum} - \mathcal{E}_k, \\ \text{WTA2}: &\quad w_k \propto \mathbb{1}_{\mathcal{E}_k = \mathcal{E}_{min}}, \\ \text{WTA3}: &\quad w_k \propto (\mathcal{E}_k + \alpha \mathcal{E}_{mean})^{\beta}, \end{aligned}$$

where $\mathcal{E}_{sum}$, $\mathcal{E}_{min}$ and $\mathcal{E}_{mean}$ are respectively the sum, minimum and mean of $\{\mathcal{E}_k\}_{k=1,2,\ldots,k_{max}}$. For WTA3, the values $\alpha < 1$ and $\beta < 0$ must be provided by the user; [29] recommends $\alpha = 0.05$ and $\beta = -1$. The WTA2 method is tantamount to selecting the best model; if there are several surrogates with the same error, equal weights are assigned to these models.

Another approach is to compute the weights that minimize the error metric $\mathcal{E}$ of the aggregated model. However, we have observed that under the constraints $0 \leq w_k \leq 1$ and $\sum_k w_k = 1$, the optimization of the weights tends to be equivalent to WTA2. Some works [61] have added a diversity term to favor sharing the weights among all surrogate models; however, we have observed that this did not perform as well as selecting the best models. Consequently, in this work, we will use the WTA2 method. The remaining question is then which error metric to use.

The set of models used in this work to build an ensemble of surrogates is listed in Table 1. This set of surrogate models has been selected empirically after testing diverse models. The wide range of values for the degree of the PRS and the shape parameter r_ϕ of the Gaussian kernel allows this set to be adapted to reflect different situations and yet be of a reasonable size.

Table 1: Surrogate model types.

#	Model	Param. 1	Param. 2
1	PRS	Degree = 1	$r_{ridge} = 0$
2		1	10^{-3}
3		2	0
4		2	10^{-3}
5		3	0
6		6	10^{-3}
7	KS	$r_\phi = 0.1$	Gaussian kernel
8		0.3	
9		1.0	
10		3.0	
11		10	
12	RBFI	$r_\phi = 0.3$	Gaussian kernel
13		1.0	
14		3.0	
15		10	
16		Degree = 1	Poly-harmonic kernel
17		2	

4.1 Quantifying model quality

A large number of surrogate model techniques exist, which makes it difficult to select the best. Moreover, most surrogate modeling techniques include parameters that must be selected or optimized (degree for PRSs, kernel type and shape for RBFIs and KS) since they have a large impact on their "performance". The efficiency of the surrogate-based search depends largely on the discrepancy between the prediction $\hat{f}(\mathbf{x})$ and the real blackbox output $f(\mathbf{x})$ (idem for the constraints). Several studies have shown that there exists no single surrogate model technique that is consistently best for all problems [2, 25, 29].

The idea investigated in this work consists of building several surrogate models (using different modeling techniques and different parameters) and selecting, for each blackbox output, the model with the smallest

discrepancy. We call a *metric* a measure that allows to quantify this discrepancy. The most common metric is the root mean square error (RMSE)

$$\mathcal{E}_{RMSE} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\Big(y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)\Big)^2}.$$

This error metric is broadly used in many scientific contexts, including surrogate model selection [72]. However, it can fail to quantify the predictive accuracy of a surrogate model since the accuracy of the model is evaluated on the points that were used to build it. For example, RBF (unlike RBFI) and Kriging models are generally interpolating, which means that the model error is zero at the training points. Thus, the $\mathcal{E}_{RMSE}$ will be zero; however, this does not guarantee accurate predictions outside the training set.

4.1.1 Cross-validation

Cross-validation methods can be useful in addressing this issue of assessing the accuracy of a surrogate model at other points [2, 25, 63]. The principle is to partition the set of available data $\mathbf{X}$ to two disjoint sets: the training set and the testing set. The training set is used to build the surrogate model. The accuracy of this model is then assessed using the testing set. This process can be repeated for different training and testing sets. Such methods generally provide a more robust estimation of the predictive capability of surrogate models. However, they may be computationally expensive as they require the construction of numerous models. Moreover, a varying size of the training set may have an adverse impact on the surrogate modeling method.

Considering this, the *leave-one-out* (LOO) cross-validation method seems to be promising and straightforward to implement. For each point $\mathbf{x}_i \in \mathbf{X}$, the training set $\mathbf{X}^{(-i)} = \mathbf{X} \setminus \{\mathbf{x}_i\}$ is used to build the surrogate model $\hat{y}^{(-i)}$. This model is then tested on the point $\mathbf{x}_i$. The LOO technique has two main advantages over other cross-validation methods. Firstly, the training sets $\mathbf{X}^{(-i)}$ are almost identical to $\mathbf{X}$, which allows the behavior of $\hat{y}^{(-i)}$ to be as close as possible to that of $\hat{y}$. Secondly, this method does not explicitly require all the models $\hat{y}^{(-i)}$ to be built: the "validation value" $\hat{y}^{(-i)}(\mathbf{x}_i)$ can often be computed for all i with only a small additional computational effort. Specifically, as described in Section 3.1, if the surrogate model is built by means of solving a linear system, the validation values can be obtained by computing explicitly the inverse of the matrix of this system [55].

The PRESS metric used in [5, 66] is a sum of squares of cross-validation values. To compare it to the RMSE metric, we take the square root of its average and define

$$\mathcal{E}_{PRESS} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\Big(y(\mathbf{x}_i) - y^{(-i)}(\mathbf{x}_i)\Big)^2}.$$

Cross-validation improvements have been proposed in [17, 50]. In these works, the cross-validation error is evaluated with various numbers of training points and the error is extrapolated to the entire set $\mathbf{X}$. This takes into account the fact that the model behavior depends on the number of training points, especially if that number is small. However, these methods are too computationally expensive to be considered in this work.

4.1.2 Order-based error (OE) metrics

The main contribution of this work is to propose a novel class of metrics that is tailored to blackbox surrogate optimization. We start from the observation that Problems (P) and ($\hat{P}$) have the same solution(s) if the following two conditions are satisfied

$$f(\mathbf{x}) \leq f(\mathbf{x}') \Leftrightarrow \hat{f}(\mathbf{x}) \leq \hat{f}(\mathbf{x}'), \text{ for all } \mathbf{x}, \mathbf{x}' \in \mathcal{X},$$

$$c_j(\mathbf{x}) \leq 0 \Leftrightarrow \hat{c}_j(\mathbf{x}) \leq 0, \quad \text{for all } \mathbf{x} \in \mathcal{X} \text{ and } j \in J.$$

These conditions will generally not be satisfied. Therefore, we propose a discrepancy metric $\mathcal{E}_{OE}$ (where OE stands for Order Error) that is based on the quantification of the violation of these conditions on the training points:

$$\mathcal{E}_{OE} = \begin{cases} \frac{1}{p^2}\sum_{i=1}^{p}\sum_{l=1}^{p}\theta\Big(f(\mathbf{x}_i)-f(\mathbf{x}_l), \hat{f}(\mathbf{x}_i)-\hat{f}(\mathbf{x}_l)\Big) & \text{for the objective } f, \\ \\ \frac{1}{p}\sum_{i=1}^{p}\theta\Big(c_j(\mathbf{x}_i), \hat{c}_j(\mathbf{x}_i)\Big) & \text{for a constraint } c_j, \end{cases}$$

where $\theta : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\theta(a, b) = (a \leq 0) \text{ xor } (b \leq 0),$$

and xor is the logical *exclusive or* operator (A xor $B = 1$ if the two Booleans A and B differ, otherwise it is equal to zero).

$\mathcal{E}_{OE}$ is equal to zero if the conditions are satisfied on the training points. We observe that $\mathcal{E}_{OE}$ is bounded above by 1, and a value $\mathcal{E}_{OE} > 0.5$ indicates that the surrogate model is less accurate than its opposite function. As for the $\mathcal{E}_{RMSE}$ metric, an interpolating model will yield $\mathcal{E}_{OE} = 0$, without yielding necessarily accurate predictions. To address this issue, we propose the use of cross-validation, and define the order-based metrics

$$\mathcal{E}_{OECV} = \begin{cases} \frac{1}{p^2}\sum_{i=1}^{p}\sum_{l=1}^{p}\theta\Big(f(\mathbf{x}_i)-f(\mathbf{x}_l), \hat{f}^{(-i)}(\mathbf{x}_i)-\hat{f}^{(-l)}(\mathbf{x}_l)\Big) & \text{for the objective } f, \\ \\ \frac{1}{p}\sum_{i=1}^{p}\theta\Big(c_j(\mathbf{x}_i), \hat{c}_j^{(-i)}(\mathbf{x}_i)\Big) & \text{for a constraint } c_j. \end{cases}$$

4.2 Comparison of the metrics

In this section, we compute different error metrics for several surrogate models of some test functions to illustrate how they can lead to different choice decisions. We first propose a simple academic, easy to reproduce test function

$$y(x) = \begin{cases} x^2 & \text{if } x \leq 1/2, \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

and we consider the training points $\mathbf{X} = \{\pm 1/k, k = 1, 2, \ldots, 100\}$ which emulate the sequence of iterates, i.e., the density increases as we get closer to the minimizer. On this set of training points, the function y is nearly a quadratic function with just one outlier at $x = 1/2$. Note that this outlier has a rather small value, which makes the representation of this function easier.

We build two ensembles of surrogates for which the best models will be selected using either the PRESS metric or the OECV metric. When looking at the general shape of the function (Figure 1), we see that the model "Select PRESS" (which selected the simple model "PRS 2" with $r_{ridge} = 10^{-3}$) seems to be very accurate and very close to y. However, the model "Select OECV" (which selected the simple model "KS 10") has a very nonsmooth behavior. It is clear that the cross-validation error will be especially high for the KS model at the training points $\mathbf{x} \in \{-1, -1/2, +1/5, +1/2, +1\}$ because KS models do not extrapolate. However, when we "focus" in the region $[-0.1; +0.1]$, we see that the model "Select OECV" fits the data very well. On the contrary, for the model "Select PRESS", while the error is small, the fit is shifted to the left due to the outlier at $x = 1/2$. As a consequence, unlike the minimizer of the model "Select PRESS", the minimizer of the model "Select OECV" is very close to the minimizer of y, which is the most important feature for surrogate-based optimization.

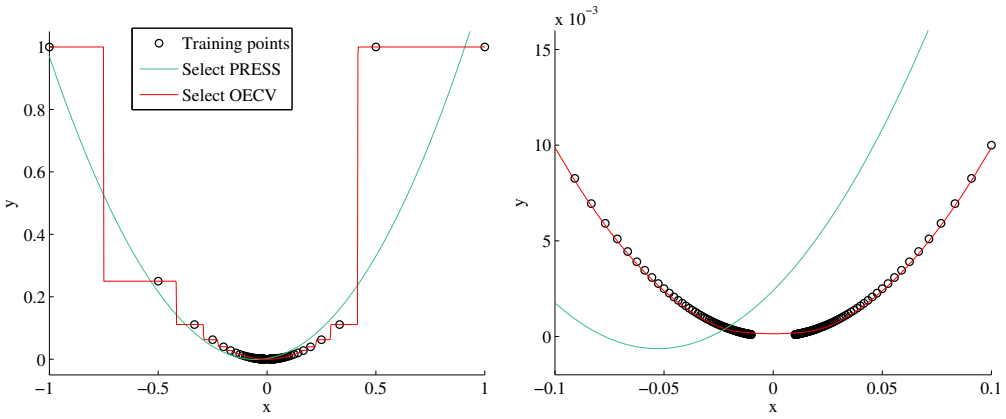

Figure 1: The test function of Equation (2); the left plot represents all of the training points as well as the two surrogate models; the right plot represents the same data, but with a zoom near the minimizer of function y.

The values of the four metrics for each of the seventeen surrogate models listed in Table 1 are presented in Figure 2. Test functions that include Branin-Hoo, Camelback, Rosenbrock 9, Hartman 3 and Hartman 6

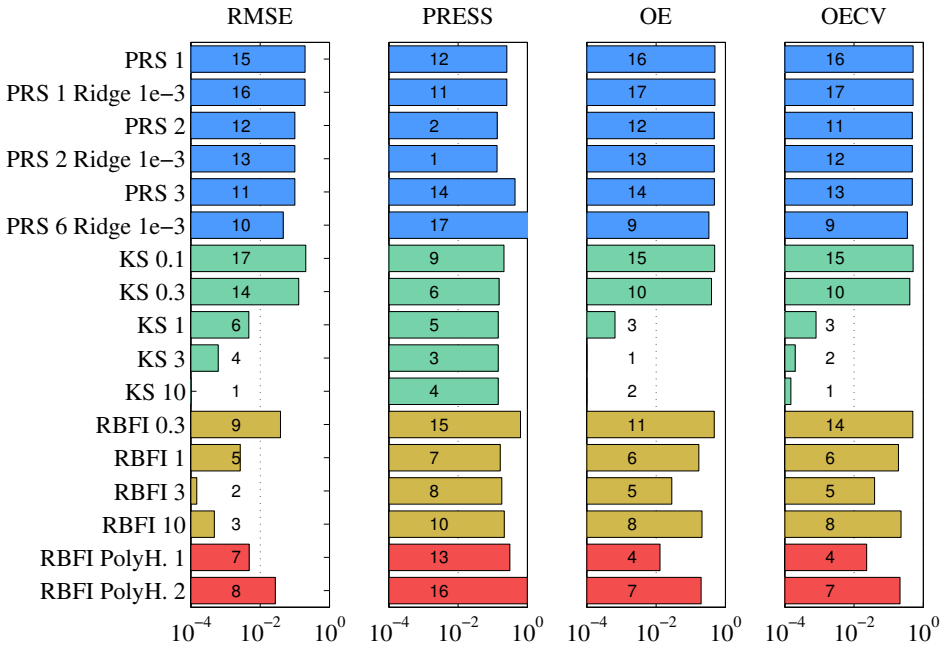

Figure 2: Comparison of metrics on the test function of Equation (2). The number in the bars denotes model rank according to the used metric.

were used in [2, 25, 29]. Figure 3 reports results for the Hartman 6 test function; it confirms that different metrics can lead to different surrogate model selection.

5 Computational experiments

We test the surrogate ensembles and associated model selections on four sets of problems. First, we use ten analytical benchmark problems. Then, we consider three engineering design applications, namely, the "simplified wing" [70], the "aircraft range" [3, 39] and the "Lockwood" [33, 37, 47, 48] problems. Table 2 summarizes the main properties of the three engineering design optimization problems.

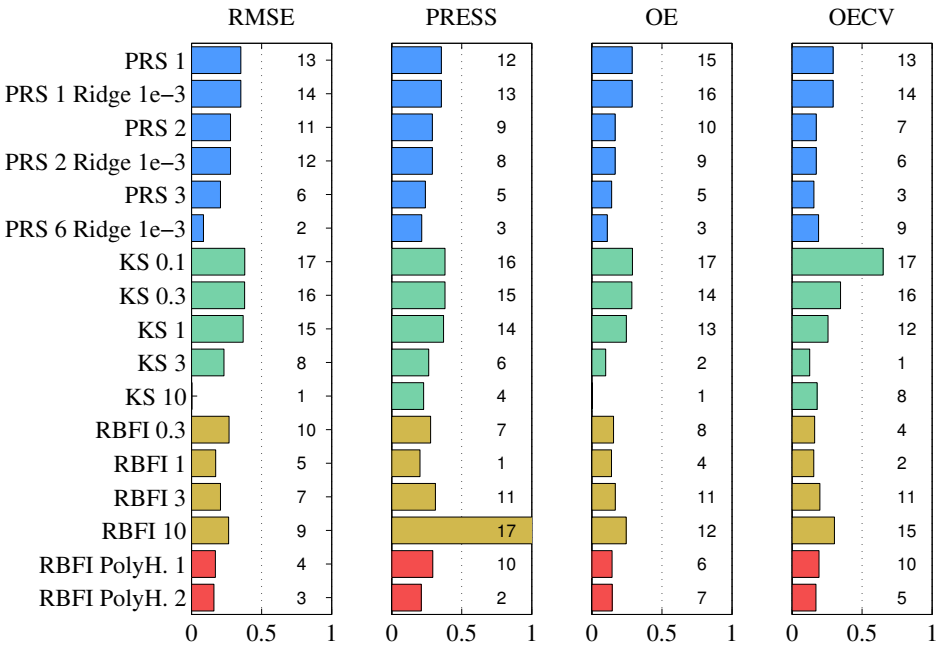

Figure 3: Comparison of metrics on the Hartman 6 test function. The number in the bars denotes model rank according to the used metric.

Table 2: Main properties of the three engineering design optimization problems. The blackbox evaluation times are evaluated on a 3.4 Ghz Intel Core i7-2600 with 16 Gbytes of RAM.

Problem name	n	m	Eval. budget	Eval. time
Simplified wing	7	4	8000	44ms
Aircraft range	10	11	11000	3ms
Lockwood	6	5	700	1.6s

For each of these three problems, 50 different starting points were generated through Latin hypercube (LH) sampling [49]. Then, each problem is solved for each of these starting points. This benchmarking strategy allows for the comparison of solvers from various starting points of the design space. It is worth noting that most of these points will be distant from the global minimizer of the problem. For each optimization run, a budget of $1000(n+1)$ evaluations is allocated, with the sole exception being the more time-expensive Lockwood problem, for which a budget of $100(n+1)$ evaluations is allocated. The number of variables n differs from one problem to another. The surrogates or ensembles tested are listed in Table 3: The method "None" refers to MADS without a surrogate search. The method "Quad" refers to a MADS search step that uses local quadratic models as described in [19]. The acronym "VNS" refer to MADS with the Variable Neighborhood Search (VNS) search strategy [7] designed to escape from local minima by favoring global exploration [51]. The remaining four types are ensembles where the metric utilized to select the best model differs ("RMSE", "OE", "PRESS" or "OECV").

For the cases where an ensemble of surrogates is used, the surrogate Problem ($\hat{P}$) is solved with an inner instance of MADS in which no search step is employed. A budget of 10,000 surrogate model evaluations is allocated for each solution of ($\hat{P}$). The initial mesh and poll size of the second MADS instance (the instance that is utilized to solve the surrogate problem) are chosen equal to the current mesh and poll size of the main MADS instance. Up to four starting points (if available) are provided for the solution of the surrogate problem. The two first possible starting points are the current best feasible and infeasible solutions of the main problem (P). The two last possible starting points are the best non-projected feasible and infeasible designs of the surrogate problem optimization performed during the search step of iteration $t-1$. These

Table 3: List of compared solvers.

#	Solver	Legend
1	None	
2	Quad	
3	VNS	
4	Select RMSE	
5	Select OE	
6	Select PRESS	
7	Select OECV	

two last starting points reuse the information available from the previous search step in order to improve the local convergence of the inner MADS instance.

Moreover, this inner MADS instance strongly relies on LH sampling and VNS search to improve the chances of finding a global optimum of the surrogate problem ($\hat{P}$) and allows the main MADS instance to escape from local attraction basins. All MADS executions are conducted using the NOMAD [41] implementation version 3.7.1, using the default values for the algorithmic parameters that are not involved in defining the configurations of Table 3.

5.1 Quantifying deviation from the best known solution

For each optimization run $\rho \in \{1, 2, \ldots, \rho_{\max}\}$, we denote $f_{s,\rho,i}$ the best objective value found for solver s after i groups of $n+1$ evaluations. If no feasible point is found, we assign $f_{s,\rho,i} = +\infty$. The best solution found among all runs for optimization run ρ is denoted by

$$f_\rho^* = \min_{s,i} f_{s,\rho,i}.$$

The *worst first feasible* objective value of an optimization run ρ is defined as

$$f_\rho^w = \max_{\substack{s,i \\ f_{s,\rho,i}<+\infty}} f_{s,\rho,i}.$$

In practice, for an optimization run ρ and solver s, the value $f_{s,\rho,i}$ is piecewise constant with respect to i. We make the assumption that it takes more than one finite value. This allows to define the relative discrepancy to the best known solution of run ρ for solver s after i groups of $n+1$ evaluations

$$\delta_{s,\rho,i} = \frac{f_{s,\rho,i} - f_\rho^*}{f_\rho^w - f_\rho^*} \in [0; 1] \cup \{+\infty\}.$$

This definition of discrepancy differs from that of [52] to take into account constrained optimization problems. Specifically, it requires that at least two feasible designs of different objective function value have been found by any solver for each run. For each set of runs, we report the value of the median discrepancy after i groups of $n+1$ evaluations. The use of the median instead of the mean is motivated by wide range of magnitude that the discrepancy values can take. Data profiles [52] are provided in Appendix C.

5.2 Analytical benchmark problems

The ten analytical benchmark problems are listed in Table 4. These problems are solved using each of the seven solvers in Table 3. Median discrepancy curves are depicted in Figure 4.

We first observe that having no search step in MADS ("None") yields poor performance, which confirms the motivation for surrogate-based search. Second, the "VNS" performs well in terms of final result, but requires a large blackbox evaluation budget to become competitive. Third, the ensemble choices based on the

Table 4: Analytical benchmark optimization problems.

Name		n	m	Bounds	Smooth
MAD6	[44]	5	7	no	no
CRESCENT	[1, 10]	10	2	no	yes
SNAKE	[10, 19]	2	2	no	yes
HS24		2	3	yes	yes
HS36		3	1	yes	yes
HS37		3	2	yes	yes
HS73	[35]	3	3	yes	no
HS101		7	4	yes	yes
HS102		7	4	yes	yes
HS103		7	4	yes	yes

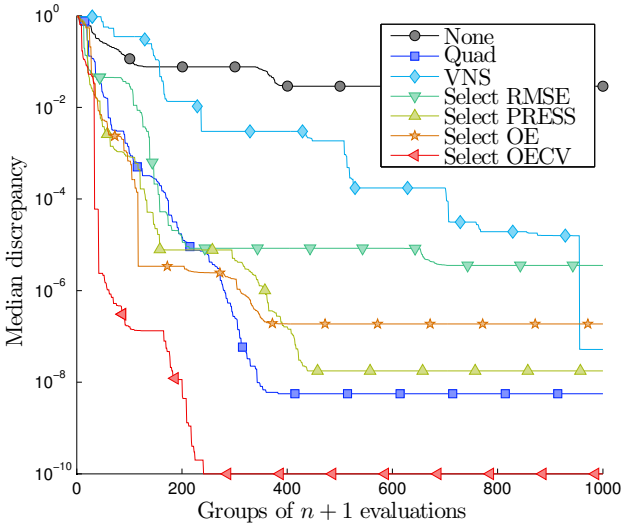

Figure 4: Median discrepancy curves for the ten analytical benchmark problems.

"Select RMSE" and "Select OE" metrics do not perform as well as their cross-validation counter-parts "Select PRESS" and "Select OECV". This illustrates the importance of using cross-validation in the error metrics. For the non cross-validation metrics, we observe that "Select OE" performs better than "Select RMSE". Similarly, "Select OECV" performs better than "Select PRESS". This illustrates that the order-based error approach is more efficient than the quadratic error approach on this set of runs. Finally, we observe that "Select OECV" is the most efficient of all surrogate-based searches for these analytical problems. The median discrepancy reaches 0 after 250 groups of $n+1$ evaluations as this algorithm finds the best solution for eight out of the ten problems. It is the only algorithm that performs better than MADS with local quadratic search. These observations are confirmed by the data profiles provided in Figure 9 of Appendix C. Based on these results, only "Quad", "VNS", "Select PRESS" and "Select OECV" will be tested on the two engineering design problems.

5.3 Simplified wing problem

The "simplified wing" problem considers the minimization of the drag of a wing by optimizing its geometry [70]. It is a multidisciplinary design optimization (MDO) problem that involves structures and aerodynamics. The aerodynamic analysis considers an RAE2822 airfoil in transonic flight and the 2D flow equations with boundary layers are solved with BGK [11]. The structural analysis considers a wing discretized at 1000 stations. Shear and axial stresses created by lift and drag forces are computed at each station. The multidisciplinary analysis is solved with a fixed point method which stops after 100 iterations, or when the normalized variation of the analysis variables is smaller than 10^{-8}. The objective function is computed with a variant

of the Breguet range equation [40]. The problem is smooth but has many local minima. The best objective function value reported in [70] is $f^* = -16.61$. The problem formulation is

$$\begin{array}{rl} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^7} & \text{wing drag} \\ \text{s.t.} & \left\{ \begin{array}{lcl} \text{shear stress} & \leq & 73{,}200 \text{ psi} \\ \text{tensile stress} & \leq & 47{,}900 \text{ psi} \\ \text{total weight} & \leq & \text{total lift} \end{array} \right. . \end{array}$$

The design variables, their bounds (set $\mathcal{X}$) and known optimal values are listed in Table 5. The median discrepancy curves for the three surrogate types ("Quad", "Selec PRESS" and "Select OECV") and for the "VNS" algorithm are depicted in Figure 5.

Table 5: Simplified wing problem design variables.

Design variable	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$	$\mathbf{x}^*$
Wing span	30	45	44.132
Root chord	6	12	6.758
Taper ratio	0.28	0.50	0.282
Angle of attack at root	-1	3	3.0
Angle of attack at tip	-1	3	0.718
Tube external diameter	1.6	5.0	4.03
Tube thickness	0.3	0.79	0.3

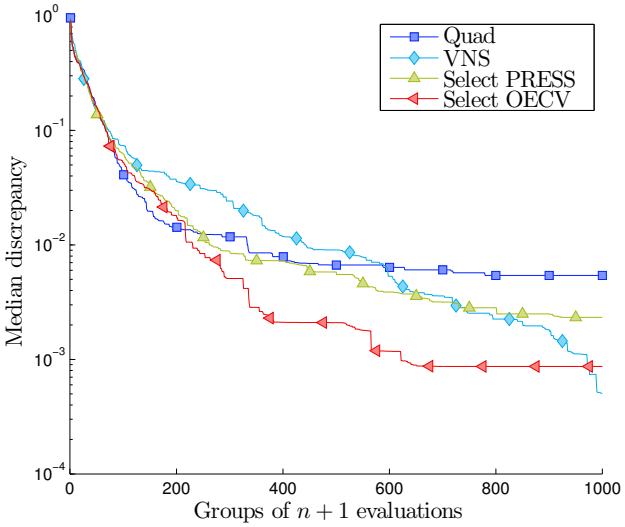

Figure 5: Median discrepancy curves for the 50 simplified wing optimization runs.

"Select PRESS" shows a slight advantage over "Quad" and "Select OECV" generally performs better than the other solvers. At the very end of the evaluation budget, the "VNS" solver overtakes the "Select OECV" method and leads to a slightly better median discrepancy.

5.4 Aircraft range problem

The "aircraft range" problem is also an MDO problem that considers aerodynamics, structures, propulsion and performance for designing a supersonic business jet [3, 39]. The multidisciplinary analysis is handled with a fixed point method that implements the multidisciplinary feasible optimization (MDF) approach. The process terminates after 100 fixed point iterations, or when the normalized variation in variables shared by several disciplines is smaller than 10^{-12}. The variables relative to the thickness of the wing are not considered

here, and are fixed at the optimal value indicated in [68]. The problem is not smooth and has several local optima. The best objective value found in this work is $f^* = -3,964.20$ which coincides with that of [65]. The problem formulation is

$$\max_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^{10}} \text{ aircraft range}$$

$$\text{s.t.} \begin{cases} \text{normalized stress} & \leq & 1.09 \text{ (5 constraints)} \\ \text{pressure gradient} & \leq & 1.04 \text{ Pa.m}^{-1} \\ \text{engine scale factor} & \geq & 0.5 \\ \text{engine scale factor} & \leq & 1.5 \\ \text{normalized engine temperature} & \leq & 1.02 \\ \text{throttle setting} & \leq & \text{max throttle} \quad . \end{cases}$$

Design variables, their bounds (set $\mathcal{X}$) and known optimal values are listed in Table 6. The median discrepancy curves are depicted in Figure 6.

Table 6: Aircraft range problem design variables.

Design variable	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$	$\mathbf{x}^*$
Taper ratio	0.1	0.4	0.4
Wingbox cross-section	0.75	1.25	0.75
Skin friction coeff.	0.75	1.25	0.75
Throttle	0.1	1.0	0.156
Thickness/chord	0.01	0.09	0.06
Altitude	30000	60000	60000
Mach number	1.4	1.8	1.4
Aspect ratio	2.5	8.5	2.5
Wing sweep	40	70	70
Wing surface area	50	1500	1500

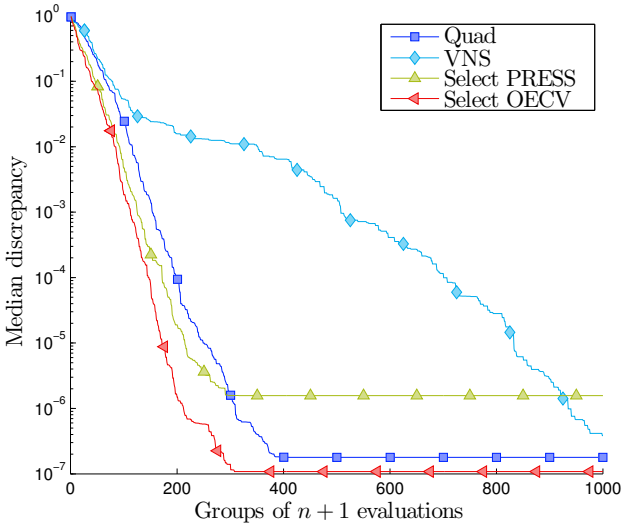

Figure 6: Median discrepancy for the set of aircraft range runs.

Though the difference is not as profound as in the previous tests, "Select OECV" is once again the best-performing search, especially for high-precision tolerances (see data profile in Figure 11 of Appendix C). For this problem, "Select PRESS" has a slight advantage on "Quad" at the early stages of the optimization process, but this trend is reversed later on.

5.5 Lockwood

The "Lockwood" problem [33, 37, 47, 48] consists of minimizing the cost of the extraction of pollutant from the soil of the Lockwood site via six decontamination wells in order to prevent two plumes from expanding. The $n = 6$ design variables of this problem are described in Table 7. The objective function of the problem is linear and represents the extraction cost. This structure is not exploited and the objective is treated as a blackbox. The feasible domain is non-connected and defined by four simulation-based inequality constraints which are linked two by two in order to approximate equality constraints of the form $-10^{-3} \leq Q \leq +10^{-3}$ where Q is the steady-state flux of pollutant from each of the two plumes. As observed in [32], even with its linear objective, this problem poses a difficult global optimization challenge due to the presence of these constraints. We allocate a budget of $100(n+1)$ blackbox evaluations for each run of this problem.

Table 7: Lockwood problem design variables.

Pumping rates (m^3/day)	$\underline{\mathbf{x}}$	$\overline{\mathbf{x}}$	$\mathbf{x}^*$
Plume A, Well 1	0	566	9.712
Plume A, Well 2	0	566	149.744
Plume B, Well 3	0	566	382.695
Plume B, Well 4	0	566	59.6640
Plume B, Well 5	0	566	20.6464
Plume B, Well 6	0	566	22.9182

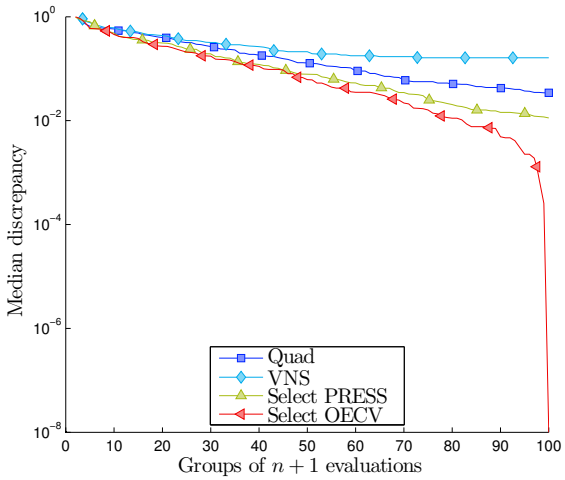

Figure 7: Median discrepancy for the set of Lockwood runs.

The best known objective value (reported in [47]) is \$23,714, which is reached after 993 blackbox evaluations. With a budget of only 700 blackbox evaluations, 72% of the "Select OECV" runs surpass this objective and one of the runs reaches an objective value of \$22,805. This solver performs better than any of the other solvers. In particular, it reaches the best objective function value f^* for more than half of the runs, which makes the median discrepancy reach zero. Figure 7 presents the value of the median discrepancy for the four solvers "QUAD", "VNS", "Select PRESS" and "Select OECV".

We define a model selection map $S_{i,k}$ as the number of times that the model k (among the 17 models listed in Table 1) is selected after i groups of $n+1$ blackbox evaluations over the 50 runs. Figure 8 depicts the selection maps for the objective and constraint functions c_1 and c_3 of the Lockwood problem for the solvers "Select PRESS" and "Select OECV". Notice how "Select PRESS" captured the linear nature of the objective while "Select OECV" did not, at least directly. This is due to the fact that many different surrogate techniques achieved an order error of zero on this linear function. As observed in Figure 1 and in [69], the PRESS metric tends to overestimate the efficiency of PRS models, whereas the OECV metric is more prone to using KS and RBF models, which in turn favor multi-modal representation of the blackbox outputs.

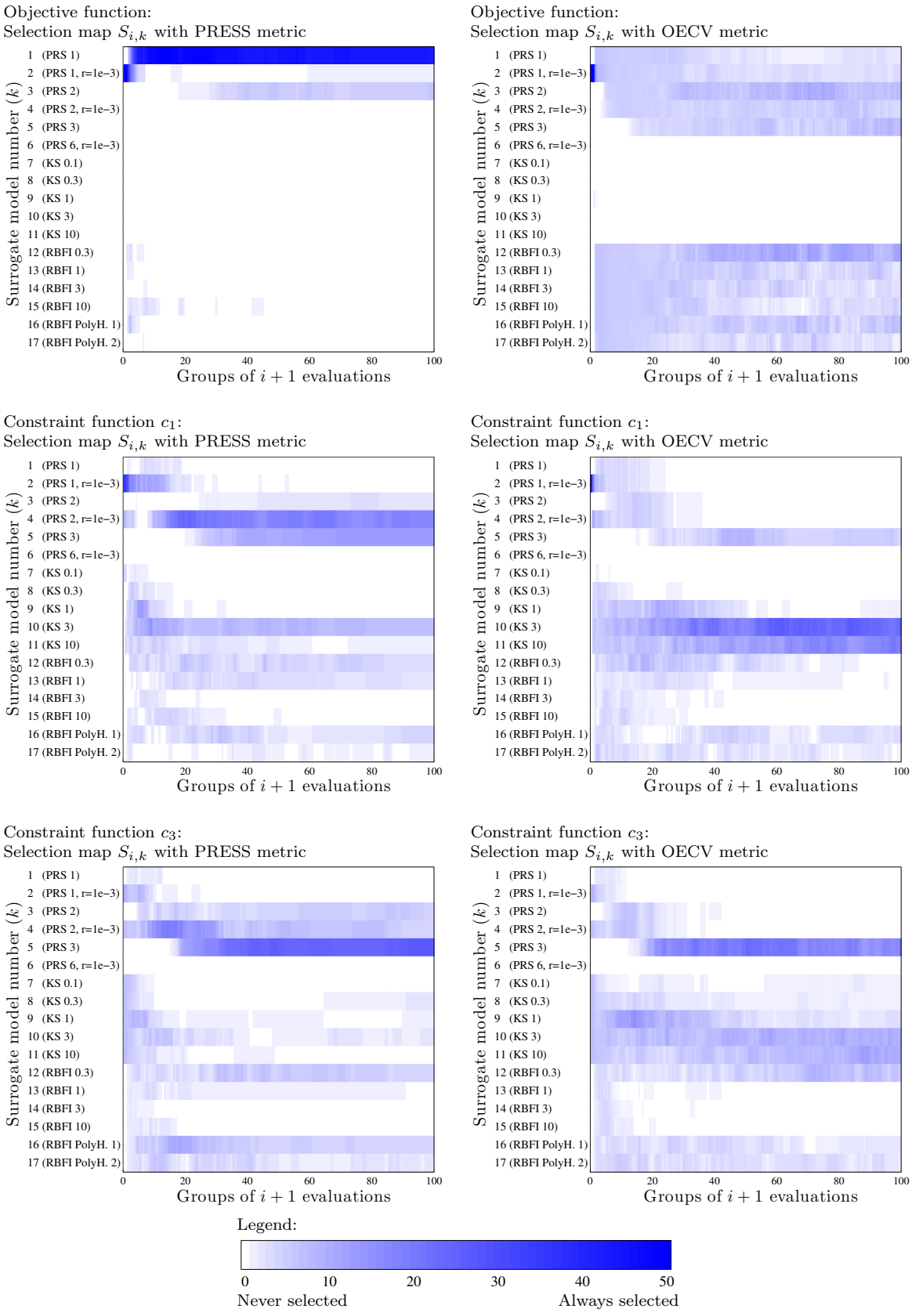

Figure 8: Model selection map $S_{i,k}$ for "Select PRESS" and "Select OECV" solvers on the Lockwood problem; a selection map indicates how often the model k was selected during blackbox evaluation number $i(n+1)$ over the 50 runs. Darker tones indicate a model selected more frequently. The left (resp. right) column represents the selection maps with the "Select PRESS" (resp. "Select OECV") solver. The first row represents the objective function while the second (resp. third) row represents the first constraint function c_1 (resp. c_3).

6 Concluding remarks

We propose a global optimization method based on ensembles of surrogates. These ensembles are managed with a new order-based metric and used for surrogate-based exploration in the search step of MADS.

We consider ensembles of surrogates based on three types of modeling techniques, including a novel and efficient method to build RBF models. The ensembles of surrogates are used in the search step of MADS with an improved projection method that allows to conserve the convergence properties of MADS without reducing its efficiency.

The main contribution of this work is the introduction of an order-based error metric tailored to surrogate-based search. This error metric involves cross-validation to ensure that the models have adequate prediction capabilities.

Using ten analytical benchmark problems and three engineering design applications, we demonstrated that order-based error metrics perform better than others as they favor models that are relevant for the majority of training points, i.e., outliers do not contribute inadequately in order-based metrics. In the early stage of the optimization, the use of order-based metrics ensures an adequate representation of multi-modal functions, thus improving the global exploration capabilities of the algorithm. Moreover, as the optimization process unfolds, models selected with order-based metrics provide more accurate candidates of local minimizers. These candidates allow for more efficient local convergence.

On the contrary, metrics such as the RMSE and the PRESS will favor models which may overemphasize outliers. This allows a good representation of the general shape of the blackbox functions at the early stages of the optimization process when few data are available. However, the importance of the outliers does not decrease fast enough for these metrics as more data become available. Consequently, these metrics will often favor the selection of PRS models, which will lead to a poor representation of multi-modal functions and will make the search step unable to move away from local minimizers. Finally, note that a fourth engineering application (trumpet optimization) is detailed in [69] and that it confirms the efficiency of the ensembles of surrogates based on the order error.

Appendices

Appendix A Greedy selection algorithm

Algorithm 2 Greedy selection.

Input : S_{in}, $\mathbf{x}_0$ and p_{out}
[1] Initialization
  Randomly draw $\mathbf{x}_{new}$ in $S_{in} \backslash \{\mathbf{x}_0\}$
  $S_{out} \leftarrow \{\mathbf{x}_{new}\} \cup (\mathbf{x}_0 \cap S_{in})$
  $\lambda \leftarrow 3$
  $\lambda_{min} \leftarrow 0.01$
[2] Greedy selection
  while $\text{card}(S_{out}) < p_{out}$ and $\lambda > \lambda_{min}$
    $\mathbf{x}_{new} \in \underset{\mathbf{x} \in S_{in}}{\text{argmax}} \; d(\mathbf{x}, S_{out}) - \lambda d(\mathbf{x}, \mathbf{x}_0)$
    if $d(\mathbf{x}_{new}, S_{out}) = 0$
      $\lambda \leftarrow 0.99\lambda$
    else
      $S_{out} \leftarrow S_{out} \cup \{\mathbf{x}_{new}\}$
    end
  end
Return S_{out}

This greedy algorithm is used in Section 2.2 to filter the set of projection candidates and in Section 3.1.2 to select a subset of the training points around which the radial basis functions will be centered. For the projection step, the inputs of Algorithm 2 are:

$$\begin{aligned} S_{in} &\leftarrow S_{Proj} \text{ (Set of projection candidates)}, \\ \mathbf{x}_0 &\leftarrow \mathbf{x}_t^S \text{ (Non projected solution of } (\hat{P})), \\ p_{out} &\leftarrow 100n. \end{aligned}$$

For the selection of the RBF kernels, the inputs of Algorithm 2 are:

$$\begin{aligned} S_{in} &\leftarrow \mathbf{X} \text{ (Set of points previously evaluated)}, \\ \mathbf{x}_0 &\leftarrow \mathbf{x}_t^* \text{ (Incumbent solution of } (P)), \\ p_{out} &\leftarrow q^{RBF} \text{ (Number of radial basis functions, } q^{RBF} = \min\{p/2, 10n\}). \end{aligned}$$

In both cases, the goal of this algorithm is to select a set S_{out} of p_{out} points from the set $S_{in} \in \mathbb{R}^n$. The set S_{out} must be spread as widely as possible in $\mathbb{R}^n$ while favoring points that are close to a target $\mathbf{x}_0$. Since these two goals can be conflicting, a trade-off parameter λ is introduced. When many points that are close to $\mathbf{x}_0$ have been selected, it is possible that the goal of selecting even more points close to $\mathbf{x}_0$ will lead to selecting a point that is already selected. In that case, the trade-off coefficient λ is decreased.

Appendix B Description of the analytical problems

B.1 MAD6 [44]

The original formulation uses seven variables, but two linear equality constraints allow to express the problem with five variables as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^5} \max_{1\le i\le 163} f_i(\mathbf{x})$$
$$\text{s.t.} \begin{cases} c_1(\mathbf{x}) = -x_1 + 0.4 \le 0 \\ c_2(\mathbf{x}) = x_1 - x_2 + 0.4 \le 0 \\ c_3(\mathbf{x}) = x_2 - x_3 + 0.4 \le 0 \\ c_4(\mathbf{x}) = x_3 - x_4 + 0.4 \le 0 \\ c_5(\mathbf{x}) = x_4 - x_5 + 0.4 \le 0 \\ c_6(\mathbf{x}) = -x_4 + x_5 - 0.6 \le 0 \\ c_7(\mathbf{x}) = x_4 - 2.1 \le 0 \end{cases}$$

where, for $1 \le i \le 163$,

$$f_i(\mathbf{x}) = \frac{1}{15} + \frac{2}{15}\left(\left(\sum_{k=1}^{5} \cos(2\pi x_k \sin v_i)\right) + \cos\left(2\pi(1+x_4)\sin v_i\right) + \cos(7\pi \sin v_i)\right)$$

and

$$v_i = \frac{\pi}{180}\left(8.5 + \frac{i}{2}\right).$$

$$\mathbf{x}_0 = \begin{bmatrix} 0.5 \\ 1 \\ 1.5 \\ 2 \\ 2.5 \end{bmatrix}, \ f(\mathbf{x}_0) = 0.22052, \ \mathbf{x}^* = \begin{bmatrix} 0.4 \\ 0.819839074 \\ 1.219839074 \\ 1.69398531 \\ 2.09398531 \end{bmatrix}, \ f(\mathbf{x}^*) = 0.101831.$$

Best value achieved in this work: $f = 0.101831$.

B.2 CRESCENT [1, 10]

$$\min_{\mathbf{x}\in\mathbb{R}^{10}} \quad x_{10}$$
$$\text{s.t.} \quad \begin{cases} c_1(\mathbf{x}) = \sum_{i=1}^{10} (x_i - 1)^2 - 100 \leq 0 \\ c_2(\mathbf{x}) = \sum_{i=1}^{10} (x_i + 1)^2 - 100 \leq 0 \end{cases}$$

$$\mathbf{x}_0 = [10\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^\top, \ f(\mathbf{x}_0) = 0,$$
$$\mathbf{x}^* = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ -9]^\top, \ f(\mathbf{x}^*) = -9.$$

Best value achieved in this work: $f = -9$.

B.3 SNAKE [10, 19]

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad \sqrt{(x_1 - 20)^2 + (x_2 - 1)^2}$$
$$\text{s.t.} \quad \begin{cases} c_1(\mathbf{x}) = \sin x_1 - \frac{1}{10} - x_2 \leq 0 \\ c_2(\mathbf{x}) = x_2 - \sin x_1 \leq 0 \end{cases}$$

$$\mathbf{x}_0 = [0\ -10]^\top, \ f(\mathbf{x}_0) = \infty \ (\mathbf{x}_0 \text{ is infeasible}),$$
$$\mathbf{x}^* = [20.02887\ 0.92434]^\top, \ f(\mathbf{x}^*) = 0.08098094.$$

Best value achieved in this work: $f = 0.08098094$. [1]

B.4 HS24 [35]

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad \frac{(x_1-3)^2-9}{27\sqrt{3}} x_2^3$$
$$\text{s.t.} \quad \begin{cases} c_1(\mathbf{x}) = -\frac{x_1}{\sqrt{3}} + x_2 \leq 0 \\ c_2(\mathbf{x}) = -x_1 - \sqrt{3}x_2 \leq 0 \\ c_3(\mathbf{x}) = +x_1 + \sqrt{3}x_2 - 6 \leq 0 \\ x_1, x_2 \geq 0 \end{cases}$$

$$\mathbf{x}_0 = [1\ 0.5]^\top, \ f(\mathbf{x}_0) = -0.0133646,$$
$$\mathbf{x}^* = [3\ 1.7320508071]^\top, \ f(\mathbf{x}^*) = -1.$$

Best value achieved in this work: $f = -1$.

[1] There is an error in the objective function formulation of the SNAKE problem in [65].

B.5 HS36 [35]

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^3} & -x_1x_2x_3 \\ \text{s.t.} & \begin{cases} c_1(\mathbf{x}) = x_1 + 2x_2 + 2x_3 - 72 \leq 0 \\ 0 \leq x_1 \leq 20 \\ 0 \leq x_2 \leq 11 \\ 0 \leq x_3 \leq 42 \end{cases} \end{array}$$

$$\begin{aligned} \mathbf{x}_0 &= [10\ 10\ 10]^\top,\ f(\mathbf{x}_0) = -1000, \\ \mathbf{x}^* &= [20\ 11\ 15]^\top,\ f(\mathbf{x}^*) = -3300. \end{aligned}$$

Best value achieved in this work: $f = -3300$.

B.6 HS37 [35]

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^3} & -x_1x_2x_3 \\ \text{s.t.} & \begin{cases} c_1(\mathbf{x}) = x_1 + 2x_2 + 2x_3 - 72 \leq 0 \\ c_2(\mathbf{x}) = -x_1 - 2x_2 - 2x_3 \leq 0 \\ 0 \leq x_1, x_2, x_3 \leq 42 \end{cases} \end{array}$$

$$\begin{aligned} \mathbf{x}_0 &= [10\ 10\ 10]^\top,\ f(\mathbf{x}_0) = -1000, \\ \mathbf{x}^* &= [24\ 12\ 12]^\top,\ f(\mathbf{x}^*) = -3456. \end{aligned}$$

Best value achieved in this work: $f = -3456$.

B.7 HS73 [35]

In the original problem, the value of x_4 is computed directly by eliminating one linear equality constraint ($x_4 = 1 - x_1 - x_2 - x_3$). Constraint c_3 is added to handle the bound $x_4 \geq 0$.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^3} & 24.55x_1 + 26.75x_2 + 39x_3 + 40.5(1 - x_1 - x_2 - x_3) \\ \text{s.t.} & \begin{cases} c_1(\mathbf{x}) = -2.3x_1 - 5.6x_2 - 11.1x_3 - 1.3(1 - x_1 - x_2 - x_3) + 5 \leq 0 \\ c_2(\mathbf{x}) = -12x_1 - 11.9x_2 - 41.8x_3 - 52.1(1 - x_1 - x_2 - x_3) + 21 \\ \qquad +1.645\sqrt{0.28x_1^2 + 0.19x_2^2 + 20.5x_3^2 + 0.62(1 - x_1 - x_2 - x_3)^2} \leq 0 \\ c_3(\mathbf{x}) = x_1 + x_2 + x_3 - 1 \leq 0 \\ 0 \leq x_1, x_2, x_3 \leq 1 \end{cases} \end{array}$$

$$\mathbf{x}_0 = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix},\ f(\mathbf{x}_0) = 31.14,\ \mathbf{x}^* = \begin{bmatrix} 0.6355215683 \\ 0 \\ 0.3127018814 \end{bmatrix},\ f(\mathbf{x}^*) = 29.8944.$$

Best value achieved in this work: $f = 29.8944$.

B.8 HS101, 102 and 103 [35]

$$
\min_{\mathbf{x}\in\mathbb{R}^7} \quad \frac{10x_1x_4^2x_7^a}{x_2x_6^3} + \frac{15x_3x_4}{x_1x_2^2x_5\sqrt{x_7}} + \frac{20x_2x_6}{x_1^2x_4x_5^2} + \frac{25x_1^2x_2^2\sqrt{x_5}x_7}{x_3x_6^2}
$$

$$
\text{s.t.} \quad \begin{cases}
c_1(\mathbf{x}) = \dfrac{0.5\sqrt{x_1}x_7}{x_3\sqrt{x_6}} + \dfrac{0.7x_1^3x_2x_6\sqrt{x_7}}{x_3^2} + \dfrac{0.2x_3x_6^{2/3}\sqrt[4]{x_7}}{x_2\sqrt{x_4}} - 1 \leq 0 \\
c_2(\mathbf{x}) = \dfrac{2x_1x_5\sqrt[3]{x_7}}{x_3^{3/2}x_6} + \dfrac{0.1x_2x_5}{x_6\sqrt{x_3}\sqrt{x_7}} + \dfrac{x_2\sqrt{x_3}x_5}{x_1} + \dfrac{0.65x_3x_5x_7}{x_2^2x_6} - 1 \leq 0 \\
c_3(\mathbf{x}) = \dfrac{2x_1x_5\sqrt[3]{x_7}}{x_3^{3/2}x_6} + \dfrac{0.1x_2x_5}{\sqrt{x_3}x_6\sqrt{x_7}} + \dfrac{x_2\sqrt{x_3}x_5}{x_1} + \dfrac{0.65x_3x_5x_7}{x_2^2x_6} - 1 \leq 0 \\
c_4(\mathbf{x}) = \dfrac{0.2x_2\sqrt{x_5}\sqrt[3]{x_7}}{x_1^2x_4} + \dfrac{0.3\sqrt{x_1}x_2^2x_3\sqrt[3]{x_4}\sqrt[4]{x_7}}{x_5^{2/3}} + \dfrac{0.4x_3x_5x_7^{3/4}}{x_1^3x_2^2} + \dfrac{0.5x_4\sqrt{x_7}}{x_3^2} - 1 \leq 0 \\
0.1 \leq x_1, x_2, \ldots, x_6 \leq 10 \\
0.01 \leq x_7 \leq 10
\end{cases}
$$

The value of the parameter a varies between the three problems:

$$
\begin{cases}
\text{HS101: } a = -1/4\ ; \\
\text{HS102: } a = -1/8\ ; \\
\text{HS103: } a = -1/2\ .
\end{cases}
$$

The starting point is $\mathbf{x}_0 = [6\ 6\ 6\ 6\ 6\ 6\ 6]^\top$ with the value $f(\mathbf{x}_0) = +\infty$ ($\mathbf{x}_0$ is infeasible) and the best know solutions are:

$$
\mathbf{x}^*_{\text{HS101}} = \begin{bmatrix} 3.1921264708 \\ 0.7569354058 \\ 2.5119021207 \\ 5.6530753445 \\ 0.871294938 \\ 1.261887906 \\ 0.0558117342 \end{bmatrix}, \quad \mathbf{x}^*_{\text{HS102}} = \begin{bmatrix} 3.3669180827 \\ 0.7679140551 \\ 2.797994961 \\ 4.1843330406 \\ 0.8791473174 \\ 1.0784651788 \\ 0.0328185121 \end{bmatrix}, \quad \mathbf{x}^*_{\text{HS103}} = \begin{bmatrix} 2.2198703858 \\ 0.6133345902 \\ 2.105352648 \\ 4.5835873238 \\ 0.8698142544 \\ 1.1828253718 \\ 0.046094896 \end{bmatrix}
$$

with

$$
f(\mathbf{x}^*_{\text{HS101}}) = 1948.02,\ f(\mathbf{x}^*_{\text{HS102}}) = 1495.48,\ f(\mathbf{x}^*_{\text{HS103}}) = 3367.69.
$$

The best values achieved in this work are 2480.94, 1950.26, and 3367.69, for HS101, HS102, and HS103, respectively.

Appendix C Data profiles

For a given tolerance $\tau \leq 1$, the *ratio of solved problems* for solver s after i groups of $n+1$ evaluations is defined as:

$$
r_{s,i}(\tau) = \frac{1}{\rho_{\max}} \sum_{\rho=1}^{\rho_{\max}} \mathbb{1}\Big(\delta_{s,\rho,i} \leq \tau\Big),
$$

where $\mathbb{1}$ is the indicator function. Note that if $\tau = 1$, a problem is considered solved if a feasible solution has been found. For a given value of τ, the data profile is the curve corresponding to the ratio of solved problems after i groups of $n+1$ evaluations [52]. For the set of analytical problems and for the two sets of MDO runs (simplified wing and aircraft range), we show the data profiles for $\tau \in \{10^{-3}, 10^{-5}, 10^{-7}\}$ (See Figures 9, 10 and 11). For the Lockwood problem runs, to which is allocated a much smaller budget of evaluations, we show the data profiles for $\tau \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ (See Figure 12). Note that for the medium discrepancy

curve, the lower the curve value, the better the performance. However, for the data profiles, the higher the better.

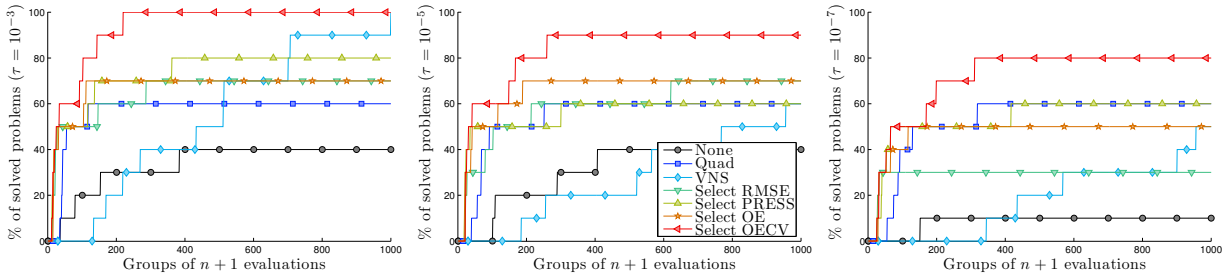

Figure 9: Data profiles for the ten analytical problems; $\tau = 10^{-3}$ (left), 10^{-5} (middle) and 10^{-7} (right).

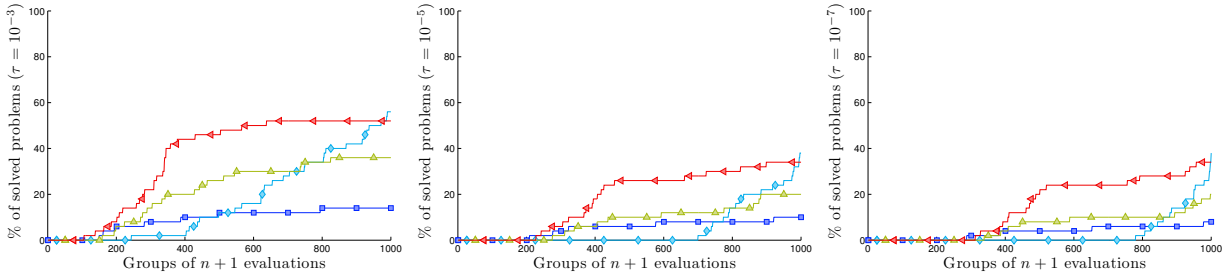

Figure 10: Data profiles for the 50 optimization runs of the simplified wing problem; $\tau = 10^{-3}$ (left), 10^{-5} (middle) and 10^{-7} (right).

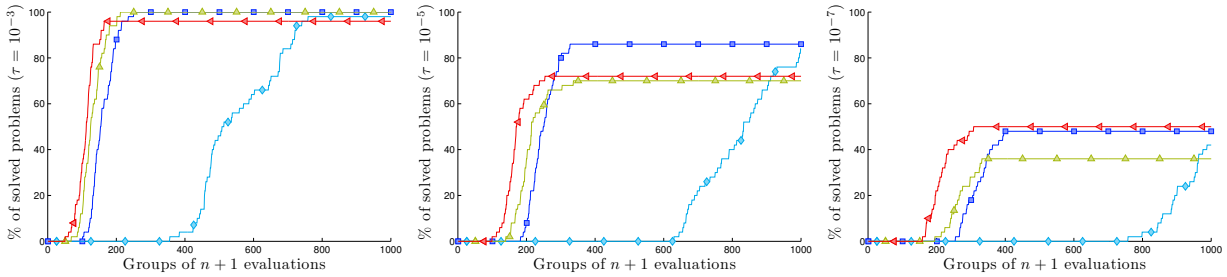

Figure 11: Data profiles for the 50 optimization runs of aircraft range problem; $\tau = 10^{-3}$ (left), 10^{-5} (middle) and 10^{-7} (right).

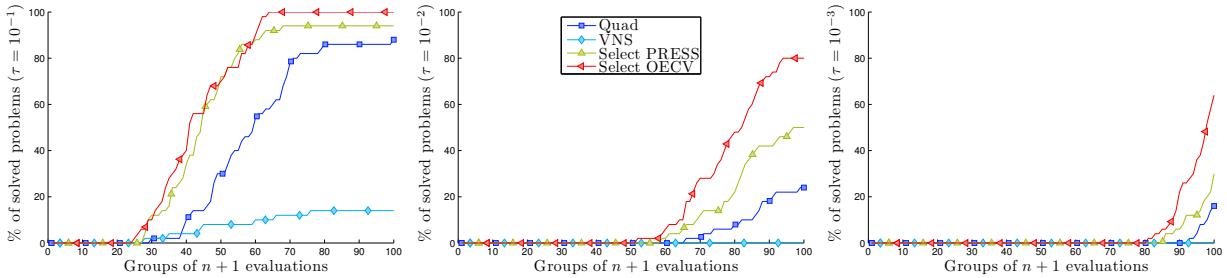

Figure 12: Data profiles for the 50 optimization runs of Lockwood problem; $\tau = 10^{-1}$ (left), 10^{-2} (middle) and 10^{-3} (right).

References

[1] M.A. Abramson, C. Audet, J.E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. SIAM Journal on Optimization, 20(2):948–966, 2009.

[2] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. Structural and Multidisciplinary Optimization, 37(3):279–294, 2009.

[3] J.S. Agte, J. Sobieszczanski-Sobieski, and R.R.J. Sandusky. Supersonic business jet design through bilevel integrated system synthesis. In Proceedings of the World Aviation Conference, volume SAE Paper No. 1999-01-5622, San Francisco, CA, 1999. MCB University Press, Bradford, UK.

[4] N.M. Alexandrov, J.E. Dennis, Jr., R.M. Lewis, and V. Torczon. A trust-region framework for managing the use of approximation models in optimization. Structural and Multidisciplinary Optimization, 15(1):16–23, 1998.

[5] D.M. Allen. The Relationship between Variable Selection and Data Agumentation and a Method for Prediction. Technometrics, 16(1):125–127, 1974.

[6] C. Audet. A survey on direct search methods for blackbox optimization and their applications. In P.M. Pardalos and T.M. Rassias, editors, Mathematics without boundaries: Surveys in interdisciplinary research, chapter 2, pages 31–56. Springer, 2014.

[7] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. Journal of Global Optimization, 41(2):299–318, 2008.

[8] C. Audet, A.J. Booker, J.E. Dennis, Jr., P.D. Frank, and D.W. Moore. A surrogate-model-based method for constrained optimization. Presented at the 8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2000.

[9] C. Audet and J.E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.

[10] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009.

[11] F. Bauer, P. Garabedian, and D. Korn. Supercritical Wing Section III. Springer Berlin Heidelberg, 1977.

[12] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A Rigorous Framework for Optimization of Expensive Functions by Surrogates. Structural and Multidisciplinary Optimization, 17(1):1–13, 1999.

[13] F. Boukouvalaa and C.A. Floudas. ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. Optimization Letters, 11(5):895–913, 2017.

[14] F. Boukouvalaa, M.M. Faruque Hasan, and C.A. Floudas. Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption. Journal of Global Optimization, 67(1):3–42, 2017.

[15] M.D. Buhmann. Radial Basis Functions: Theory and Implementations. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

[16] Y.-C. Chen, C. Wei, and H.-C. Yeh. Rainfall network design using kriging and entropy. Hydrological Processes, 22(3):340–346, 2008.

[17] S. Chowdhury, A. Mehmani, J. Zhang, and A. Messac. Quantifying Regional Error in Surrogates by Modeling its Relationship with Sample Density. In Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2013.

[18] B. Colson. Trust-Region Algorithms for Derivative-Free Optimization and Nonlinear Bilevel Programming. PhD thesis, Département de Mathématique, FUNDP, Namur, Belgium, 2003.

[19] A.R. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software, 28(1):139–158, 2013.

[20] A.R. Conn, K. Scheinberg, and L.N. Vicente. Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[21] A.R. Conn and Ph.L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Gianessi, editors, Nonlinear Optimization and Applications, pages 27–47. Plenum Publishing, New York, 1996.

[22] A.L. Custódio, H. Rocha, and L.N. Vicente. Incorporating minimum Frobenius norm models in direct search. Computational Optimization and Applications, 46(2):265–278, 2010.

[23] A. Dhar and B. Datta. Global optimal design of ground water monitoring network using embedded kriging. Ground Water, 47(6):806–815, 2009.

[24] B.C. Eaves. On quadratic programming. Management Science, 17(11):698–711, 1971.

[25] B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association, 78(382):316–331, 1983.

[26] R. Fletcher, N.I.M. Gould, S. Leyffer, Ph.L. Toint, and A. Wächter. On the global convergence of trust-region SQP-filter algorithms for general nonlinear programming. SIAM Journal on Optimization, 13(3):635–659, 2002.

[27] K.R. Fowler, J.P. Reese, C.E. Kees, J.E. Dennis Jr., C.T. Kelley, C.T. Miller, C. Audet, A.J. Booker, G. Couture, R.W. Darwin, M.W. Farthing, D.E. Finkel, J.M. Gablonsky, G. Gray, and T.G. Kolda. Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. Advances in Water Resources, 31(5):743–757, 2008.

[28] J.D Gergonne. The application of the method of least squares to the interpolation of sequences. Historia Mathematica, 1(4):439–447, 1974.

[29] T. Goel, R.T. Haftka, W. Shyy, and N.V. Queipo. Ensemble of surrogates. Structural and Multidisciplinary Optimization, 33(3):199–216, 2007.

[30] T. Goel and N. Stander. Comparing three error criteria for selecting radial basis function network topology. Computer Methods in Applied Mechanics and Engineering, 198(27-29):2137–2150, 2009.

[31] D.E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman, Boston, 1989.

[32] R.B. Gramacy, G.A. Gray, S. Le Digabel, H.K.H. Lee, P. Ranjan, G. Wells, and S.M. Wild. Modeling an Augmented Lagrangian for Blackbox Constrained Optimization. Technometrics, 58(1):1–11, 2016.

[33] R.B. Gramacy and S. Le Digabel. The mesh adaptive direct search algorithm with treed Gaussian process surrogates. Pacific Journal of Optimization, 11(3):419–447, 2015.

[34] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[35] W. Hock and K. Schittkowski. Test Examples for Nonlinear Programming Codes, volume 187 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1981.

[36] D.R Jones, M. Schonlau, and W.J. Welch. Efficient Global Optimization of Expensive Black Box Functions. Journal of Global Optimization, 13(4):455–492, 1998.

[37] A. Kannan and S.M. Wild. Benefits of Deeper Analysis in Simulation-based Groundwater Optimization Problems. In Proceedings of the XIX International Conference on Computational Methods in Water Resources (CMWR 2012), June 2012.

[38] S. Kitayama, M. Arakawa, and K. Yamazaki. Sequential approximate optimization using radial basis function network for engineering optimization. Optimization and Engineering, 12(4):535–557, 2011.

[39] S. Kodiyalam. Multidisciplinary aerospace systems optimization. Technical Report NASA/CR-2001-211053, Lockheed Martin Space Systems Company, Computational AeroSciences Project, Sunnyvale, CA, 2001.

[40] I. Kroo. Aircraft design: synthesis and analysis; Cruise performance and range. http://adg.stanford.edu/aa241/performance/cruise.html, 2005.

[41] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 37(4):44:1–44:15, 2011.

[42] H.A. Le Thi, A.I.F. Vaz, and L.N. Vicente. Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimization. TOP, pages 1–25, 2011.

[43] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff. A Study on Metamodeling Techniques, Ensembles, and Multi-surrogates in Evolutionary Computation. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07, pages 1288–1295, New York, NY, USA, 2007. ACM.

[44] L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report V-798, ICS AS CR, 2000.

[45] Y. Mack, T. Goel, W. Shyy, R.T. Haftka, and N.V. Queipo. Multiple surrogates for the shape optimization of bluff body-facilitated mixing. In 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2005. AIAA. Paper AIAA-2005-0333.

[46] K.V. Mardia, J.T. Kent, and J.M. Bibby. Multivariate analysis. Probability and mathematical statistics. Academic Press, 1979.

[47] L.S. Matott, K. Leung, and J. Sim. Application of MATLAB and Python optimizers to two case studies involving groundwater flow and contaminant transport modeling. Computers & Geosciences, 37(11):1894–1899, 2011.

[48] L.S. Matott, A.J. Rabideau, and J.R. Craig. Pump-and-treat optimization using analytic element method flow models. Advances in Water Resources, 29(5):760–775, 2006.

[49] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2):239–245, 1979.

[50] A. Mehmani, S. Chowdhury, and A. Messac. Predictive quantification of surrogate model fidelity based on modal variations with sample density. Structural and Multidisciplinary Optimization, 52(2):353–373, 2015.

[51] N. Mladenović and P. Hansen. Variable neighborhood search. Computers and Operations Research, 24(11):1097–1100, 1997.

[52] J.J. Moré and S.M. Wild. Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

[53] J. Müller and R. Piché. Mixture surrogate models based on Dempster-Shafer theory for global optimization problems. Journal of Global Optimization, 51(1):79–104, 2011.

[54] J. Müller and C.A. Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms forcomputationally expensive black-box global optimization problems. Journal of Global Optimization, 60(2):123–144, 2014.

[55] M.J.L. Orr. Introduction to radial basis function networks. Technical report, Center for Cognitive Science, University of Edinburgh, 1996.

[56] N. Peremezhney, E. Hines, A. Lapkin, and C. Connaughton. Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. Engineering Optimization, 46(11):1593–1607, 2014.

[57] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucher. Surrogate-based analysis and optimization. Progress in Aerospace Sciences, 41(1):1–28, 2005.

[58] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2006.

[59] R.G. Regis. Stochastic Radial Basis Function Algorithms for Large-scale Optimization Involving Expensive Black-box Objective and Constraint Functions. Computers and Operations Research, 38(5):837–853, 2011.

[60] R.G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Engineering Optimization, 46(2):218–243, 2014.

[61] R. Shi, L. Liu, T. Long, and J. Liu. An efficient ensemble of radial basis functions method based on quadratic programming. Engineering Optimization, 48(7):1202–1225, 2016.

[62] S.M. Stigler. Gergonne's 1815 paper on the design and analysis of polynomial regression experiments. Historia Mathematica, 1(4):431–439, 1974.

[63] M. Stone. An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):44–47, 1977.

[64] M.A. Taddy, R.B. Gramacy, and N.G. Polson. Dynamic trees for learning and design. Journal of the American Statistical Association, 106(493):109–123, 2011.

[65] B. Talgorn, S. Le Digabel, and M. Kokkolaras. Statistical Surrogate Formulations for Simulation-Based Design Optimization. Journal of Mechanical Design, 137(2):021405–1–021405–18, 2015.

[66] T. Tarpey. A Note on the Prediction Sum of Squares Statistic for Restricted Least Squares. The American Statistician, 54(2):116–118, 2000.

[67] Y. Tenne. An adaptive-topology ensemble algorithm for engineering optimization problems. Optimization and Engineering, 16(2):303–334, 2014.

[68] S. Tosserams, L.F.P Etman, and J.E. Rooda. A classification of methods for distributed system optimization based on formulation structure. Structural and Multidisciplinary Optimization, 39(5):503–517, 2009.

[69] R. Tournemenne, J.-F. Petiot, B. Talgorn, M. Kokkolaras, and J. Gilbert. Brass Instruments Design Using Physics-Based Sound Simulation Models and Surrogate-Assisted Derivative-Free Optimization. Journal of Mechanical Design, 139(4):041401–01–041401–9, 2017.

[70] C. Tribes, J.-F. Dubé, and J.-Y. Trépanier. Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design. Engineering Optimization, 37(8):775–796, 2005.

[71] A.I.F. Vaz and L.N. Vicente. A particle swarm pattern search method for bound constrained global optimization. Journal of Global Optimization, 39(2):197–219, 2007.

[72] F.A.C. Viana, R.T. Haftka, S. Valder, Jr., S. Butkewitsch, and M.F. Leal. Ensemble of Surrogates: a Framework based on Minimization of the Mean Integrated Square Error. In 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials, Schaumburg, IL, 2008.

[73] F.A.C. Viana, R.T. Haftka, and L.T. Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. Journal of Global Optimization, 56(2):669–689, 2013.

[74] G.S. Watson. Smoothing and interpolation by kriging and with splines. Mathematical Geology, 16:601–615, 1984.

[75] S.M. Wild, R.G. Regis, and C.A. Shoemaker. ORBIT: optimization by radial basis function interpolation in trust-regions. SIAM Journal on Scientific Computing, 30(6):3197–3219, 2008.