Order-based error for managing ensembles of surrogates in derivative-free optimization

Charles Audet *  Michael Kokkolaras †  Sébastien Le Digabel ‡  
Bastien Talgorn §

May 30, 2016

Abstract

We investigate surrogate-assisted strategies for 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, 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 two 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: Derivative-free optimization, ensemble of surrogates, MADS, order error.

---

*Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal and GERAD, Montréal, Canada.  
charles.auDET@GERAD.CA  
†Department of Mechanical Engineering, McGill University and GERAD, Montréal, Canada.  
michael.kokkolaras@mcgill.ca  
‡Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal and GERAD, Montréal, Canada.  
sebastien.le.digabel@GERAD.CA  
§Department of Mechanical Engineering, McGill University and GERAD, Montréal, Canada.  
bastien.talgorn@mail.mcgill.ca
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 [4]. In addition, derivatives may either be unavailable or require significant effort to be approximated reliably. In this case, derivative-free optimization algorithms [15] may offer a valuable alternative to gradient-based techniques. Finally, simulations may fail to return a value and 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

\[
\begin{align*}
\min_{\mathbf{x} \in \mathcal{X}} & \quad f(\mathbf{x}) \\
\text{subject to} & \quad c_j(\mathbf{x}) \leq 0, \quad j \in J = \{1, 2, \ldots, m\},
\end{align*}
\]

where \(\mathcal{X}\) is a subset of \(\mathbb{R}^n\) typically defined by bound constraints \(\mathbf{x} \leq \mathbf{x} \leq \mathbf{x}\) and the functions \(f\) and \(c_j\) are evaluated by means of blackbox simulation models. We employ the mesh-adaptive direct search (MADS) algorithm [7] that relies on the search-and-poll paradigm [9]. The poll rigorously ensures convergence toward a solution satisfying some local optimality conditions, while the search can implement any method likely to improve the efficiency of the optimization process.

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 [38], variable neighborhood search [5], particle swarms [55], genetic algorithms [25] and surrogate-based [9, 14, 17, 26].

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 [6, 29] of a design, and/or considering a diversification term that will favor unexplored areas of the design space [49, 50].

The challenge is to build surrogate models that adequately approximate the blackbox outputs. Commonly used models include Gaussian processes [43, 45], in particular Kriging models [9, 11, 18, 29]. 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 [14, 16, 19], which rely on trust regions [2, 13, 21]. Finally, other surrogate modeling techniques include radial basis functions [30, 33, 59], splines [58] and kernel smoothing [1, 27].

Several studies have shown that no single surrogate modeling technique can be deemed the best on a consistent basis [1, 20, 22, 54]. To address this issue, our approach is to
build an ensemble of surrogates, in which several surrogate models of different types are constructed \([1, 34, 36, 41, 52, 54, 56, 57]\). 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, \([54]\) uses four different empirical methods to determine them directly from error metrics. On the contrary, \([1]\) and \([56]\) 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) \([3, 24, 41, 54]\). 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 \(af(x) + b\) of \(f(x)\) 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 and on two engineering multidisciplinary 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

\[
\mathcal{M}_t = \{x + \Delta_t^m Dz, z \in \mathbb{N}^{n_D}, x \in X_t\}
\]

where \(\Delta_t^m \in \mathbb{R}_+\) is the mesh size parameter at iteration \(t\), \(X_t = \{x_1, x_2, \ldots, x_p\} \subset \mathcal{X}\) is the set of points evaluated previously, and \(D\) 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 \(X_t\) are used to build a surrogate model for each of the blackbox outputs. For a given \(x \in \mathcal{X}\), the models
denoted by \( \hat{f} \) and \( \hat{c}_j \) provide estimates of \( f(x) \) and \( c_j(x) \). The surrogate-based search is based on the premise that a solution of the surrogate problem

\[
\min_{x \in X} \hat{f}(x) \\
\text{subject to } \hat{c}_j(x) \leq 0 \quad \forall j \in J,
\]

will be a promising candidate of the original Problem \( \hat{P} \). The search proceeds as follows: at each iteration of the MADS algorithm, the surrogates models are updated to account for the most recent observations available. Then, a solution of Problem \( \hat{P} \) is computed and denoted by \( x^S_t \). 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 [8]).

The point \( x^S_t \) is then projected on the mesh \( M_t \) using the method described in Section 2.2; this process yields the point \( x^P_t \) which is then evaluated using the blackbox simulation that defines Problem \( \hat{P} \). 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 \( S_{\text{Poll},t} \) of candidates is built within a distance \( \Delta^p_t > 0 \) of the incumbent solution \( x^*_t \). 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. 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 \( x^*_t \) the incumbent solution of the original problem, \( x^S_t \) the non-projected solution of the surrogate problem optimization, and \( x^P_t \) the projection of \( x^S_t \) onto the mesh

\[
M_t = \bigcup_{x \in X_t} M_{t,x},
\]

where or each point \( x \) of \( X_t \), \( M_{t,x} = \{ x + \Delta^p x Dz, z \in \mathbb{N}^{n_D} \} \) is a submesh.

The projection of \( u \in X \) on the submesh \( M_{t,x} \) can then be defined as

\[
\text{proj}_X(u) \in \arg\min_{v \in M_{t,x}} \| u - v \|_1,
\]

which can be performed by rounding operations on the components of \( u \) and \( x \).
Algorithm 1 The MADS optimization algorithm.

[1] Initialization
\[ t \leftarrow 0 \]
Set initial poll and mesh sizes \( \Delta_p \geq \Delta_m > 0 \)
Initialize \( X_0 \) with starting points
Evaluate \( f(x) \) and \( \{c_j(x)\}_{j \in J} \) \( \forall x \in X_0 \)

[2] Search
Use \( X_t \) to build \( \hat{f} \) and \( \{\hat{c}_j\}_{j \in J} \)
\( x_{S}\) ← Solution of \((\hat{P})\)
\( x_{P} \) ← Projection of \( x_{S} \) onto mesh \( M_t \)
Evaluate \( f(x_{P}) \) and \( \{c_j(x_{P})\}_{j \in J} \)
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 \( x \in S_{Poll,t} \)
\[ \begin{align*}
&\text{Evaluate } f(x) \text{ and } \{c_j(x)\}_{j \in J} \\
&\text{If success, goto [4]} \\
\end{align*} \]
end

[4] Updates
\[ t \leftarrow t + 1 \]
Update \( \Delta_m, \Delta_p, M_t, x^{*}_t \) and \( X_t \)
If no stopping condition is met, goto [2]

In previous work, the candidate returned by the search was \( \text{proj}_{x^{*}_t}(x_{S}^t) \), i.e., the projection of the surrogate problem solution on \( M_{t,x^{*}_t} \) \[14, 50\]. As we generally have \( M_{t,x^{*}_t} \subset M_t \) with a strict inclusion, many interesting points that lie on the mesh \( M_t \) 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 \( x_{S}^t \) is not necessarily the best candidate according to \( \hat{f} \) and \( \{\hat{c}_j\}_{j \in J} \).

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
\[ \Pi = \{u \in \mathbb{R}^n : u_i = \pm \Delta_i^m, \forall i = 1...n\} \subset \mathbb{R}^n, \quad (4) \]
and we build the set \( S_{\Pi} \subset \Pi \) by drawing randomly \( \min\{2^n, 100n\} \) different vectors of \( \Pi \). These vectors allow to perturb the design \( x_{S}^t \) so that not only the closest point to \( x_{S}^t \) is considered during the projection. Secondly, we build a set of projection candidates
\[ S_{Proj} = \{x : x = \text{proj}_{x^{*}_t}(x_{P}^t + u), u \in S_{\Pi}, x^{'} \in X\}. \quad (5) \]
Note that there may exist several couples \((u, x^{'}))\) that lead to the same value of \( \text{proj}_{x^{*}_t}(x_{S}^t + u) \), so the training of this set includes the suppression of duplications. Even so, it can be prohibitive to compute \( \hat{f}(x) \) and \( \{\hat{c}(x)\}_{j \in J} \) for all \( x \in S_{Proj} \), 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.
set to no more than 100n candidates. While using this algorithm, we will favor points close to $x^*_t$. Finally, all the points of $S_{Proj}$ are evaluated with $\hat{f}$ and $\{\hat{c}_j\}_{j \in J}$, and the best feasible design (or, if unavailable, the best infeasible design) is selected as the candidate $x^*_t$ of the search step at iteration $t$ of MADS.

3 Overview of selected surrogate modeling techniques

From a set of $p$ observations $[X, y]$, with $X = \{x_1, ..., x_p\} \subset \mathbb{R}^n$ and $y_i = y(x_i)$, 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 $y(x)$ for $x \notin X$. 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) \[1, 41, 44\], kernel smoothing (KS) \[1, 27\], and radial basis functions (RBFs) \[1, 10, 41, 44, 57\].

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 $\hat{y}(x)$ would have taken if the model $\hat{y}$ had been built without the observation $[x_i, y(x_i)]$. Thus, we define $\hat{y}^{(i)}$ the surrogate model built by leaving out the observation $[x_i, y(x_i)]$ for $x_i \in X$. We also define the cross-validation vector $\hat{y}^{cv}$ such that $\hat{y}^{cv}_i = \hat{y}^{(i)}(x_i)$.

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 \[37, 47, 23, 42\]

$$\hat{y}(x) = \sum_{j=1}^{q} \alpha_j h_j(x), \quad (6)$$

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

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

where $r_{ridge} \geq 0$ is a regularization (or ridge \[42\]) 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 $H$ can be built from the training points $X = \{x_1, x_2, ..., x_p\}$

$$H = \begin{bmatrix} h_1(x_1) & \ldots & h_q(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_p) & \ldots & h_q(x_p) \end{bmatrix}. \quad (7)$$
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 $\alpha$ are the solution of the invertible problem

$$\left(\mathbf{H}^\top \mathbf{H} + r_{\text{ridge}} \mathbf{I}_q\right)\alpha = \mathbf{H}^\top y. \tag{8}$$

It follows that

$$\alpha = \mathbf{A}^{-1} \mathbf{H}^\top y. \tag{9}$$

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

$$\hat{y}^{\text{cv}} = y - \text{diag}(\mathbf{P})^{-1} \mathbf{P} y, \tag{10}$$

where $\mathbf{P}$ is the projection matrix such that $\mathbf{P} = \mathbf{I}_p - \mathbf{H} \mathbf{A}^{-1} \mathbf{H}^\top$ and $\text{diag}(\mathbf{P})$ is a diagonal matrix such that $\text{diag}(\mathbf{P})_{ii} = P_{ii}$. 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 $\alpha$ for other functions $y$ (for example, each output of the blackbox simulation) as well as the cross-validation values.

### 3.1.1 Polynomial response surfaces

Polynomial response surface are linear models for which the basis functions are polynomials. For a PRS of degree $d$, the set of basis functions $\{h_j\}_{j=1,...,q}$ 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 $\phi : \mathbb{R}_\geq 0 \rightarrow \mathbb{R}$ is a kernel function. The coefficients of the RBF models are typically obtained by solving the linear system

$$\begin{bmatrix} \mathbf{H}^{\text{RBF}} & \mathbf{H}^{\text{PRS}} \\ \mathbf{H}^{\text{PRS}}^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{\text{RBF}} \mathbf{y} \end{bmatrix}, \tag{11}$$

where $\mathbf{H}^{\text{RBF}}$ is a symmetric matrix such that $H_{ij} = \phi(d(\mathbf{x}_i, \mathbf{x}_j))$ and $\mathbf{H}^{\text{PRS}}$ 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 the reduction of the number of basis functions by carefully selecting a subset of the training points $S(\mathbf{X}) = \{\mathbf{x}_i\}_{i=1,...,q^{\text{RBF}}} \subset \mathbf{X}$. This subset contains $q^{\text{RBF}} < p$ training points of $\mathbf{X}$, which will be used as centers of the radial basis functions. The matrix $\mathbf{H}^{\text{RBF}}$ is then defined in $\mathbb{R}^{p \times q^{\text{RBF}}}$ such that $H_{ij}^{\text{RBF}} = \phi(d(\mathbf{x}_i, \mathbf{x}^*_j)) \forall i = 1,...,p$ and $\forall j = 1,...,q^{\text{RBF}}$. The design matrix is defined as $\mathbf{H} = [\mathbf{H}^{\text{RBF}} \mathbf{H}^{\text{PRS}}]$, where $\mathbf{H}^{\text{PRS}}$ is the PRS design matrix of size $p \times q^{\text{PRS}}$. Consequently, the total number of basis functions in this model is $q = q^{\text{RBF}} + q^{\text{PRS}}$. We chose $q^{\text{RBF}} = \min\{p/2, 10n\}$. That means that only $q^{\text{RBF}}$ radial basis function are selected; however, they are merged with $q^{\text{PRS}}$ PRS basis functions. The PRS used in this work is of degree 1. That means that $q^{\text{PRS}} = n + 1$. As the system is overdetermined, we do not need
to add orthogonality constraints as in Equation (11). The coefficients $\alpha$ are computed from the normal equations.

To build the set $S(X)$, we use a greedy algorithm that is computationally efficient and robust in selecting $p_S$ points in the set $X$. 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 $S(X)$ by selecting the point of $X$ that maximizes the distance to $S(X)$. 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 $x^*$. The greedy selection algorithm is presented in Appendix A. It takes as input the training set $X$ and the incumbent solution $x^*$; the number of selected kernels is set to $\min\{p/2, 100\}$.

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

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:

$$\hat{y}(x) = \frac{\sum_{i=1}^{p} \phi(d(x, x_i)) y_i}{\sum_{i=1}^{p} \phi(d(x, x_i))}. \quad (12)$$

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.

4 Ensembles of surrogates

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

$$\hat{y}(x) = \sum_{k=1}^{k_{\text{max}}} w_k \hat{y}_k(x), \quad (13)$$

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

Many approaches have been proposed to choose the weights $w$. Most of them rely on calculating an error metric $E_k$ for each surrogate $\hat{y}_k$, and then setting $w_k \propto g(E_k) \geq 0$, where $g$ depends on the chosen method [56, 151]. For example, [51] proposes three weight selection methods WTA1, WTA2 and WTA3:

- WTA1 : $w_k \propto E_{\text{sum}} - E_k$, \quad (14)
- WTA2 : $w_k \propto \frac{1}{E_k} = E_{\text{min}}$, \quad (15)
- WTA3 : $w_k \propto (E_k + \alpha E_{\text{mean}})^\beta$, \quad (16)
where $E_{\text{sum}}$, $E_{\text{min}}$, and $E_{\text{mean}}$ are respectively the sum, minimum and mean of $\{E_k\}_{k=1}^{k_{\text{max}}}$. For WTA3, the values $\alpha < 1$ and $\beta < 0$ must be provided by the user; [54] 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 $E$ 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 [40] 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 $r_{\phi}$ 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.

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

### 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 RBFs 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 $\hat{f}(x)$ and the real blackbox output $f(x)$ (idem...
for the constraints). Several studies have shown that there exists no single surrogate model

The idea investigated in this work consists of building several surrogate models (using dif-

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)

\[
E_{RMSE} = \sqrt{\frac{1}{p} \sum_{i=1}^{p} (y(x_i) - \hat{y}(x_i))^2}.
\] (17)

This error metric is broadly used in many scientific contexts, including surrogate model

selection [56]. 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 and Kriging models are generally interpolating, which means that the model

error is zero at the training points. Thus, the \( E_{RMSE} \) will be zero; however, this does not

guarantee accurate predictions outside the training set.

\subsection{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 [11, 20, 48]. The principle is to partition the set of available
data \( X \) 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 \( x_i \in X \), the training set \( X^{(-i)} = X \setminus \{x_i\} \)
is used to build the surrogate model \( \hat{y}^{(-i)} \). This model is then tested on the point \( x_i \). The

LOO technique has two main advantages over other cross-validation methods. Firstly, the

training sets \( X^{(-i)} \) are almost identical to \( X \), 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” \( \hat{y}^{(-i)}(x_i) \) 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 [42].

The PRESS metric used in [3, 51] 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

\[
E_{PRESS} = \sqrt{\frac{1}{p} \sum_{i=1}^{p} (y(x_i) - \hat{y}^{(-i)}(x_i))^2}.
\] (18)

Cross-validation improvements have been proposed in [12, 39]. In these works, the cross-

validation error is evaluated with various number of training points and the error is expan-

dated to the entire set \( X \). 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

\[ f(x) \leq f(x') \Leftrightarrow \hat{f}(x) \leq \hat{f}(x'), \quad \forall x, x' \in \mathcal{X}, \quad (19) \]
\[ c_j(x) \leq 0 \Leftrightarrow \hat{c}_j(x) \leq 0, \quad \forall x \in \mathcal{X}, \forall j \in J. \quad (20) \]

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:

\[
\mathcal{E}_{OE} = \left\{ \begin{array}{ll}
\frac{1}{p^2} \sum_{i=1}^{p} \sum_{l=1}^{p} \theta(f(x_i) - f(x_l), \hat{f}(x_i) - \hat{f}(x_l)) & \text{for the objective function } f, \\
\frac{1}{p} \sum_{i=1}^{p} \theta(c_j(x_i), \hat{c}_j(x_i)) & \text{for a constraint function } c_j,
\end{array} \right. 
\]

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

\[
\theta(a, b) = (a \leq 0) \text{ xor } (b \leq 0), \quad (22)
\]

and xor is the logical exclusive or operator (\(A \text{ xor } B = 1\) if the two Boolean 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

\[
\mathcal{E}_{OECV} = \left\{ \begin{array}{ll}
\frac{1}{p^2} \sum_{i=1}^{p} \sum_{l=1}^{p} \theta(f(x_i) - f(x_i-1), \hat{f}(x_i) - \hat{f}(x_i-1)) & \text{for the objective function } f, \\
\frac{1}{p} \sum_{i=1}^{p} \theta(c_j(x_i), \hat{c}_j(x_i)) & \text{for a constraint function } c_j.
\end{array} \right. 
\]

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

\[
y(x) = \left\{ \begin{array}{ll}
x^2 & \text{if } x \leq 1/2, \\
1 & \text{otherwise},
\end{array} \right. \quad (24)
\]

and we consider the training points \(X = \{\pm 1/k; k = 1...100\}\) 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 $x = 1/2$. Such outliers are quite common in blackbox optimization, and can be encountered, for example, when the blackbox crashes. 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 2), 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 $x \in \{-1, -1/2, +1/5, +1/2, +1\}$ 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.

The values of the four metrics for each of the seventeen surrogate models listed in Table 1 are presented in Figure 1. Test functions that include Branin-Hoo, Camelback, Rosenbrock 9, Hartman 3 and Hartman 6 were used in [1, 20, 54]. Figure 2 reports results for the Hartman 6 test function; it confirms that different metrics can lead to different surrogate model selection.

Table 2: The test function of Equation (24).
Figure 1: Comparison of metrics on the test function of Equation (24). The number in the bars denotes model rank according to the used metric.

Figure 2: Comparison of metrics on the Hartman 6 test function. The number in the bars denotes model rank according to the used metric.
5 Computational experiments

We test the surrogate ensembles and associated model selections on three sets of problems. First, we use ten analytical benchmark problems. Then, we consider two engineering design applications, namely the “simplified wing” [53] and the “aircraft range” [31] problems. For each of these two design applications, we conduct 50 optimization runs using different initial guesses. For each optimization run, a budget of 1000\((n + 1)\) 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 “Quad” refers to a MADS search step that uses local quadratic models as described in [14]. The remaining four types are ensembles where the metric utilized to select the best model differs (“RMSE”, “OE”, “PRESS” or “OECV”).

Table 3: Surrogate or ensemble types.

<table>
<thead>
<tr>
<th>#</th>
<th>Model/ensemble</th>
<th>Legend</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>None</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>Quad</td>
<td>□</td>
</tr>
<tr>
<td>3</td>
<td>Select RMSE</td>
<td>✷</td>
</tr>
<tr>
<td>4</td>
<td>Select OE</td>
<td>■</td>
</tr>
<tr>
<td>5</td>
<td>Select PRESS</td>
<td>▲</td>
</tr>
<tr>
<td>6</td>
<td>Select OECV</td>
<td>✔</td>
</tr>
</tbody>
</table>

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. Moreover, up to four starting points (if available) are provided for the solution of the surrogate problem: the current best feasible and best infeasible design of the main problem, and the best non-projected feasible and infeasible designs of the previous surrogate problem optimization. All MADS executions are conducted using the NOMAD [32] implementation version 3.7.1.

5.1 Quantifying deviation from the best known solution

For each optimization run \(\rho \in \{1, ..., \rho_{\text{max}}\}\), 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 \(\rho\) is denoted by

\[
f^*_\rho = \min_{s,i} f_{s,\rho,i}. \tag{25}\]

The worst first feasible objective value of an optimization run \(\rho\) is defined as

\[
f^w_\rho = \max_{s,i} f_{s,\rho,i} \quad \text{s.t.} \quad f_{s,\rho,i} < +\infty. \tag{26}\]

In practice, for an optimization run \(\rho\) 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

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

(27)

This definition of discrepancy differs from that of [40] 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 [40] are provided in Appendix B.

5.2 Analytical benchmark problems

The ten analytical benchmark problems are listed in Table 4. These problems are solved using each of the six surrogate approaches listed in Table 3; median discrepancy curves are depicted in Figure 3.

Table 4: Analytical benchmark optimization problems.

<table>
<thead>
<tr>
<th>Name</th>
<th>n</th>
<th>m</th>
<th>Bounds</th>
<th>Smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAD6</td>
<td>5</td>
<td>7</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CRESCENT</td>
<td>10</td>
<td>2</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>SNAKE</td>
<td>2</td>
<td>2</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>HS24</td>
<td>2</td>
<td>3</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>HS36</td>
<td>3</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HS37</td>
<td>3</td>
<td>2</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HS73</td>
<td>3</td>
<td>3</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>HS101</td>
<td>7</td>
<td>4</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HS102</td>
<td>7</td>
<td>4</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HS103</td>
<td>7</td>
<td>4</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

We first observe that having no search step in MADS (“None”) yields poor performance, which confirms the motivation for surrogate-based search. Secondly, the ensemble choices based on the “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 7 of Appendix B. Based on these results, only “Quad”, “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 the drag of a wing by optimizing its geometry [53]. It is a multidisciplinary design optimization (MDO) problem that involves structures and aerodynamics. The problem is smooth but has many local minima. The best objective function value reported in [53] is \( f^* = -16.61 \).

The problem formulation is

\[
\begin{align*}
\text{min} & \quad \text{wing drag} \\
\text{subject to} & \quad \text{shear stress} \leq 73,200 \text{ psi}, \\
& \quad \text{tensile stress} \leq 47,900 \text{ psi}, \\
& \quad \text{total weight} \leq \text{total lift}. \\
\end{align*}
\]

(28)

The design variables, their bounds and known optimal values are listed in Table 5.

Table 5: Simplified wing problem design variables.

<table>
<thead>
<tr>
<th>Design variable</th>
<th>( x )</th>
<th>( \bar{x} )</th>
<th>( x^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wing span</td>
<td>30</td>
<td>45</td>
<td>44.132</td>
</tr>
<tr>
<td>Root chord</td>
<td>6</td>
<td>12</td>
<td>6.758</td>
</tr>
<tr>
<td>Taper ratio</td>
<td>0.28</td>
<td>0.50</td>
<td>0.282</td>
</tr>
<tr>
<td>Angle of attack at root</td>
<td>-1</td>
<td>3</td>
<td>3.0</td>
</tr>
<tr>
<td>Angle of attack at tip</td>
<td>-1</td>
<td>3</td>
<td>0.718</td>
</tr>
<tr>
<td>Tube external diameter</td>
<td>1.6</td>
<td>5.0</td>
<td>4.03</td>
</tr>
<tr>
<td>Tube thickness</td>
<td>0.3</td>
<td>0.79</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The median discrepancy curves for the three surrogate types (“Quad”, “Select PRESS” and “Select OECV”) are depicted in Figure 3.
“Select PRESS” shows a slight advantage on “Quad”. “Select OECV” performs very well. The final median discrepancy is more than $10^8$ times smaller than those of the other two. The most interesting feature of these results is that the discrepancy continues to decrease during the latest part of the optimization and does not flatten out. To understand this behavior, Figure 5 displays the chosen models at each MADS iteration for one of the optimization runs of the simplified wing problem.

This run is chosen because the budget of blackbox evaluations was exhausted for both “Select PRESS” and “Select OECV”. For each solver, the Figure displays the chosen models for each of the 4 blackbox outputs (wing drag, shear stress, tensile stress and total weight).

In this run, “Select PRESS” reached $f = -16.6056$ after 7333 blackbox evaluation (i.e., 917 groups of $n + 1$ evaluations) while “Select OECV” reached $f = -16.6057$ after 7663 blackbox evaluation (i.e., 958 groups of $n + 1$ evaluations). We observe that model selection varies greatly for “Select OECV” in comparison to “Select PRESS”. As described in Section 4.2, the order-based error metrics are more able to focus on small variations of the blackbox output, for example in areas with many training points.

Figure 4: Median discrepancy curves for the 50 simplified wing optimization runs.
Figure 5: Selected models in one optimization run of the simplified wing problem.
5.4 Aircraft range problem

The “aircraft range” problem is also an MDO problem that considers aerodynamics, structures and propulsion for designing a supersonic business jet [31]. 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 [50]. The problem formulation is

\[
\begin{align*}
\max & \quad \text{aircraft range} \\
\text{subject to} & \quad \text{normalized stress} \leq 1.09 \ (5 \text{ constraints}), \\
& \quad \text{pressure gradient} \leq 1.04 \ \text{Pa.m}^{-1}, \\
& \quad 0.5 \leq \text{engine scale factor} \leq 1.5, \\
& \quad \text{normalized engine temperature} \leq 1.02, \\
& \quad \text{throttle setting} \leq \text{max throttle.}
\end{align*}
\]  

Table 6: Aircraft range problem design variables.

<table>
<thead>
<tr>
<th>Variables</th>
<th>( \underline{x} )</th>
<th>( \overline{x} )</th>
<th>( x^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taper ratio</td>
<td>0.1</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>Wingbox cross-section</td>
<td>0.75</td>
<td>1.25</td>
<td>0.75</td>
</tr>
<tr>
<td>Skin friction coeff.</td>
<td>0.75</td>
<td>1.25</td>
<td>0.75</td>
</tr>
<tr>
<td>Throttle</td>
<td>0.1</td>
<td>1.0</td>
<td>0.156</td>
</tr>
<tr>
<td>Thickness/chord</td>
<td>0.01</td>
<td>0.09</td>
<td>0.06</td>
</tr>
<tr>
<td>Altitude</td>
<td>30000</td>
<td>60000</td>
<td>60000</td>
</tr>
<tr>
<td>Mach number</td>
<td>1.4</td>
<td>1.8</td>
<td>1.4</td>
</tr>
<tr>
<td>Aspect ratio</td>
<td>2.5</td>
<td>8.5</td>
<td>2.5</td>
</tr>
<tr>
<td>Wing sweep</td>
<td>40</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>Wing surface area</td>
<td>50</td>
<td>1500</td>
<td>1500</td>
</tr>
</tbody>
</table>

The median discrepancy curves are depicted in Figure 6. 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 9 of Appendix B). 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.
6 Concluding remarks

We propose a framework for managing ensembles of surrogates (i.e., model selection) for surrogate-based search in MADS using a new order-based metric. Our motivation for solving a surrogate optimization problem in the search step of MADS is to use surrogate models for acquiring as much information as possible while using the blackboxes for making optimization decisions in what we term surrogate-assisted blackbox optimization using derivative-free algorithms (such as 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 two 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. This is particularly important because, as the optimization process unfolds, more and more training points become available in the area near the optimal solution, making thus early outliers less and less relevant. Therefore, order-based metrics favor models that are able to represent the subtle variations of the objective and constraint functions in the areas close to the optimal solution.

On the contrary, metric 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.

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

This work has been supported by FRQNT grant 2015-PR-182098; the authors are grateful for this support, which does not constitute an endorsement of the opinions expressed in this paper. The authors would also like to thank Robin Tournemenne (doctoral candidate at École Centrale de Nantes) for his insightful questions and comments that contributed significantly to improving the search algorithm used in this work.

References


A Greedy selection algorithm

Algorithm 2 Greedy selection.

Input: $S_{in}$, $x_0$ and $p_{out}$

[1] Initialization
- Randomly draw $x_{new}$ in $S_{in}\backslash\{x^*\}$
- $S_{out} \leftarrow \{x_{new}\} \cup (x_0 \cap S_{in})$
- $\lambda \leftarrow 3$
- $\lambda_{min} \leftarrow 0.01$

- while $\text{card}(S_{out}) < p_{out}$ and $\lambda > \lambda_{min}$
  - Select $x_{new} \in \operatorname{arg\max}_{x \in S_{in}} d(x, S_{out}) - \lambda d(x, x_0)$
  - if $d(x_{new}, S_{out}) = 0$
    - $\lambda \leftarrow 0.99\lambda$
  - else
    - $S_{out} \leftarrow S_{out} \cup \{x_{new}\}$
  - 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. 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 $x_0$. Since these two goals can be conflicting, a trade-off parameter $\lambda$ is introduced. When many points that are close to $x_0$ have been selected, it is possible that the goal of selecting even more points close to $x_0$ will lead to selecting a point that is already selected. In that case, the trade-off coefficient $\lambda$ is decreased.

B 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:

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

where $1$ 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 $\tau$, the data profile is the curve corresponding to the ratio of solved problems after $i$ groups of $n + 1$ evaluations [40]. For each set of optimization runs, we show the data profiles for $\tau \in \{10^{-3}, 10^{-5}, 10^{-7}\}$. 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 7: Data profiles for the ten analytical problems; $\tau = 10^{-3}$ (left), $10^{-5}$ (middle) and $10^{-7}$ (right).

Figure 8: Data profiles for the 50 optimization runs of the simplified wing problem; $\tau = 10^{-3}$ (left), $10^{-5}$ (middle) and $10^{-7}$ (right).

Figure 9: Data profiles for the 50 optimization runs of aircraft range problem; $\tau = 10^{-3}$ (left), $10^{-5}$ (middle) and $10^{-7}$ (right).