Problem Formulations for Simulation-based Design Optimization using Statistical Surrogates and Direct Search

Bastien Talgorn∗ Sébastien Le Digabel† Michael Kokkolaras‡

February 17, 2014

Abstract

Typical challenges of simulation-based design optimization include unavailable gradients and unreliable approximations thereof, expensive function evaluations, numerical noise, multiple local optima and the failure of the analysis to return a value to the optimizer. One possible remedy to alleviate these issues is to use surrogate models in lieu of the computational models or simulations and derivative-free optimization algorithms. In this work, we use the R dynaTree package to build statistical surrogates of the blackboxes and the direct search method for derivative-free optimization. We present different formulations for the surrogate problem considered at each search step of the Mesh Adaptive Direct Search (MADS) algorithm using a surrogate management framework. The proposed formulations are tested on twenty analytical benchmark problems and two simulation-based multidisciplinary design optimization problems. Numerical results confirm that the use of statistical surrogates in MADS improves the efficiency of the optimization algorithm.

Keywords: Simulation-based design optimization; mesh adaptive direct search (MADS); surrogate management framework; statistical surrogates.

1 Introduction

We consider the optimization problem

\[
\min_{x \in \mathcal{X}} f(x) \\
\text{subject to } c_j(x) \leq 0, \ j \in J,
\]

where \(J = \{1, 2, \ldots, m\}\) and \(\mathcal{X}\) is a subset of \(\mathbb{R}^n\) typically defined by bound constraints. The functions \(f\) and \(c_j, \ j \in J\), map \(\mathbb{R}^n\) to \(\mathbb{R} \cup \{\infty\}\) and are evaluated using blackboxes. A blackbox is typically a simulation-based function that can be evaluated but whose internal structure is unknown and/or inaccessible. Typical challenges associated with blackboxes include numerical noise, multi-modality, high computational cost, and failure to return a value, e.g., when the simulation crashes. However, the most frequent feature of blackboxes is that gradients are unavailable and their approximations are unreliable.

∗GERAD (Group for Research in Decision Analysis), a multi-university research center
†GERAD and Département de mathématiques et génie industriel, École Polytechnique de Montréal
‡GERAD and Department of Mechanical Engineering, McGill University
Derivative-free methods [1], and in particular direct-search algorithms such as GPS [2, 3] or MADS [4], have been developed to handle these issues. These two algorithms rely on the search-and-poll paradigm. The search can implement any method to evaluate a finite number of trial points. It favors exploration of the design space and allows users to implement any appropriate method, given their knowledge of the problem. If the search fails to find an improvement, the poll proposes trial points around the incumbent solution following rigid polling conditions. These points can then be evaluated by the blackbox in any order, and the evaluation can be interrupted if an improved solution is found. The poll rigorously ensures convergence of the algorithm toward a local optimum.

A promising alternative to direct-search algorithms is surrogate-based algorithms that rely solely on surrogate models [5, 6, 7, 8, 9, 10]. From the points previously evaluated, we build surrogate models that approximate the objective and constraint functions. These models are then used to propose promising candidates. This implies the formulation and solution of an optimization problem called the surrogate problem. Much effort can be devoted to solving this problem since the computational cost of a surrogate evaluation is negligible compared to the blackbox. These methods have proved to be efficient, but they have mostly been used on smooth or unconstrained problems.

Our work is based on our belief that surrogates would be more efficient if they were integrated into the search of direct-search methods as described in the surrogate management framework [11]. The search would then involve building or updating the surrogate models and solving the surrogate problem to propose a candidate. Simple implementations of this framework have been presented for the unconstrained case in [11, 12] and for the constrained case in [13, 14]. However, only the simplest formulation of the surrogate problem was considered in these implementations, namely the optimization of a model of the objective function subject to models of the constraint functions.

The contribution of this work consists of new formulations of the surrogate problem that exploit the different capabilities of statistical surrogate modeling methods and in particular the dynaTree library [6, 15]. Eight formulations are proposed. Three of them are constrained while the others quantify the relevance of a candidate via a single statistical criterion. Six of these formulations emphasize the exploration of the design space, and all of them handle nonconvex and nonsmooth constrained problems.

The paper is organized as follows. Section 2 describes the MADS algorithm, the implementation of the surrogate-based search, and the dynaTree models. Section 3 presents the eight new surrogate problem formulations. Section 4 compares the performance of the formulations by means of twenty benchmark problems defined using analytical expressions and two simulation-based multidisciplinary design optimization (MDO) problems related to aircraft design. Section 5 provides concluding remarks.

2 Background: MADS and dynaTree

2.1 Mesh adaptive direct search (MADS)

Mesh adaptive direct search (MADS) [4] is an algorithm for blackbox optimization that can handle nonlinear constraints. It ensures convergence to a solution satisfying local optimality conditions based on the Clarke calculus for nonsmooth functions [16]. At each MADS iteration \( k \), trial points are evaluated on the mesh \( M_k \) defined as

\[
M_k = \{ x + \Delta_k^m D z : z \in \mathbb{N}^{m_D}, x \in X_k \} \subset \mathbb{R}^n
\]

where \( \Delta_k^m \) is a mesh size parameter, the columns of \( D \in \mathbb{R}^{n \times n_D} \) form a positive spanning set of \( n_D \) directions in \( \mathbb{R}^n \) [17], and \( X_k = \{ x^1, x^2, \ldots \} \subset \mathbb{R}^n \) denotes the set of already evaluated
points, called the cache. MADS relies on a search-and-poll paradigm, named after the two steps that constitute each iteration.

The search is an optional step during which several different methods can be used to propose candidates anywhere on the mesh $M_k$. These methods can be heuristic in the sense that they can be guided by user insight into the problem at hand. Alternatively, more systematic methods, such as genetic algorithms [18], variable neighborhood search [19], particle swarms [20], or Latin hypercube-based design of experiments [21] can be used during the search step. In this work, we focus on surrogate-based search methods [11, 13, 22, 14].

The data $[X_k, f(X_k), c_1(X_k), ..., c_J(X_k)]$ are used to build statistical surrogate models of $f$ and $c_j$, $j \in J$, by modeling each blackbox output as a random variable. Statistical surrogate models provide information about the mean, variance, and probability density function of the modelled random variable. In this way, we can compute statistical measurements of the favor areas with uncertain blackbox outputs. Using the formulations presented in this paper (Section 3), we find a candidate $x_k^{SP}$ (where $SP$ denotes the surrogate problem) by solving a subproblem in a secondary execution of MADS. This candidate is then projected onto the current mesh $M_k$ to preserve the original convergence properties described in [4].

During the poll step, a set of candidates, called the poll set, is defined as $P_k = \{x_k + d : d \in D_k\}$, where $D_k$ is a set of polling directions based on combinations of directions in $D$. The poll size parameter $\Delta_k^p = \max_{d \in D_k} ||d||$ defines the maximum norm of the directions of $D_k$. MADS controls $\Delta_k^m$ and $\Delta_k^p$ so that $\Delta_k^m$ decreases faster than $\Delta_k^p$, which causes the set of poll directions to grow dense in the unit sphere, once normalized. This allows polling in all possible directions of $\mathbb{R}^n$.

The set of trial points $T_k = x_k^{SP} \cup P_k$ is evaluated by the blackbox opportunistically, which means that if evaluating a point leads to an improvement, the evaluation of $T_k$ is interrupted. Since this strategy makes the evaluation order-critical, the relevance criterion used in the search step is also used to sort the points in $T_k$. This is performed via a filter-like mechanism described in [23].

After the trial-point evaluations, MADS updates the poll and mesh size parameters depending on the success of the iteration. The incumbent solution and the cache $X_k$ are then updated, and a new iteration begins. The optimization terminates when the convergence criteria are satisfied, which means that the mesh size parameter is smaller than the machine precision or the evaluation budget is exhausted. Figure 1 illustrates the complete algorithm.

### 2.2 The dynaTree library

The dynaTree library [6, 15] is used in this work to build statistical surrogate models. It is based on a Bayesian framework for parameter-free regression on nonsmooth data. From the data $[X, y(X)]$, dynaTree provides statistical information on $y(x)$, namely the mean $\hat{y}(x)$, the standard deviation $\hat{\sigma}(x)$, and the cumulative density function $P[y(x) < y_0], \forall y_0 \in \mathbb{R}$. Unlike Kriging, dynaTree does not consider $y$ to be a continuous or stationary Gaussian process. Consequently, dynaTree is a non-interpolating method, which means that $x \in X \not\Rightarrow \hat{y}(x) = y(x)$. Such methods are best suited for the approximation of nonsmooth data [6, 24]. Specifically, dynaTree implements a piecewise linear regression that allows global and robust regression in the presence of noncontinuous data or first-order discontinuities.

This regression relies on trees. As illustrated in the one-dimensional example (Figure 2), a tree implements a partition of the design space $\mathcal{X}$. Each interior node implements a partitioning criterion, and each leaf represents a partition of $\mathcal{X}$. In each leaf $\eta$, a linear regression
[1] Initialization
Set initial mesh and poll sizes $\Delta^m_0, \Delta^p_0 > 0$
Set starting point $x_0 \in \mathcal{X}$
$X_0 \leftarrow \emptyset$
$k \leftarrow 0$

[2] Search
Build or update dynaTree models
$x_{SP}^k \leftarrow$ surrogate problem solution
Project $x_{SP}^k$ onto the mesh $M_k$

[3] Poll
Build poll directions $D_k$ and trial points $P_k$

Build trial set $T_k = x_{SP}^k \cup P_k$
Sort $T_k$ according to the surrogate models
Perform opportunistic evaluation of $T_k$
$k \leftarrow k + 1$
Update mesh and poll size $\Delta^m_k, \Delta^p_k$
Update incumbent solution $x_k$
Update cache $[X_k, f(X_k), c_1(X_k), ..., c_m(X_k)]$
goto [2] if no stopping condition is met

Figure 1: Optimization algorithm.

model is built using the data $[X_\eta, y(X_\eta)]$ where $X_\eta = X \cap \eta$. In the one-dimensional example of Figure 2, the plot shows 24 data points, the partitioning of the design space, the piecewise linear prediction, and the standard deviation of $y(x)$. The diagram below the plot depicts the tree associated with the partition of the interval $[0, 25]$.

Figure 2: dynaTree regression on 24 data points in $\mathbb{R}$.
The Bayesian approach enables the computation of the likelihood $L(\eta)$ of each leaf $\eta$, which quantifies the ability of the model to fit the data in $\eta$. Then, a prior $\pi(T)$ allows us to penalize overly complicated arborescences [25, 26]. $\pi(T)$ is defined by considering that each leaf $\eta$ can be split with a probability $p_{\text{split}}(T, \eta)$ that grows with the depth of the leaf. The prior is the likelihood of the tree in relation to this splitting probability:

$$
\pi(T) = \prod_{\eta \in \text{Leafs}(T)} p_{\text{split}}(T, \eta) \prod_{\eta \in \text{Interior}(T)} p_{\text{split}}(T, \eta).
$$

Finally, the likelihood $L(T)$, which quantifies the quality of $T$, is defined as:

$$
L(T) = \pi(T) \prod_{\eta \in \text{Leafs}(T)} L(\eta).
$$

Although one tree is sufficient to build a surrogate model, dynaTree generates a set of trees. This increases the likelihood of finding several efficient partitions and allows more robust regression. A particle learning sequential Monte Carlo algorithm [27, 28] adapts the set of trees to the observations, by reproducing and modifying the trees with the best likelihood. The modifications consist of three equally probable operations: splitting a leaf, merging two leaves, or making no change. Once the set is built, predictions are made by averaging all the trees. The interested reader can refer to [15] for more details.

This method offers several advantages: the Bayesian framework can handle noisy data and provides statistical information, while the space partitioning allows us to refine the model in areas of interest. Moreover, the selection of the number of trees allows us to find a balance between accuracy and computational cost. However, dynaTree is likely to be outperformed by Kriging and polynomial regression on smooth functions. In addition, the complexity of the space partitioning grows exponentially with the dimension of the design space: dynaTree cannot normally be used when $n > 10$.

3 Formulations of the surrogate problem

The statistical information provided by the models built using dynaTree is used to compute other measures of candidate relevance. At most $m + 1$ surrogate models are built to evaluate the objective function and the $m$ constraints. The mean and standard deviation of the surrogate objective and constraints are denoted $\hat{f}$ and $\hat{\sigma}_f$ and $\hat{c_j}$ and $\hat{\sigma}_j$, respectively.

3.1 Direct surrogate of the original problem

The simplest surrogate formulation results from using surrogate models (in lieu of blackboxes) to evaluate the objective and constraints of the original problem $(P)$:

$$
\min_{x \in \mathcal{X}} \hat{f}(x)
$$
subject to $\hat{c}_j(x) \leq 0 \quad \forall j \in J$.

This formulation can be generalized to perform exploration of the design space. In [15], Taddy et al. propose solving unconstrained problems by sequentially solving the surrogate problem

$$
\min_{x \in \mathcal{X}} -EI(x) - \lambda \hat{\sigma}_f(x),
$$

There may be situations where the properties of the objective function or some of the constraints do not require the construction and use of surrogate models, e.g., if one of these functions is smooth and inexpensive and has an analytical expression.
where $EI$ is some *expected improvement* function, and $\lambda$ is an exploration parameter empirically chosen in $[0, 1]$. We use this concept to formulate the surrogate problem

$$
\min_{x \in \mathcal{X}} \tilde{f}(x) - \lambda \hat{\sigma}_f(x)
$$

subject to $\tilde{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0 \ \forall j \in J.$

This formulation is denoted $(\text{SP1-F}\sigma)$. $\text{SP1}$ denotes the first surrogate problem formulation, $F$ indicates that the objective of the surrogate problem is based on the surrogate model of the objective function, and $\sigma$ indicates that the variance of the surrogate model is taken into account for the exploration.

Taddy et al. use the values $\lambda \in \{1/100, 1/10, 1\}$ because the literature [8, 15] considers searching with $\lambda = 0$ to be *myopic*. We will consider the values $\lambda \in \{0, 1/100, 1/10, 1\}$. Large values of $\lambda$ imply that the search will favor candidates with an uncertain objective, which are in the ill-explored or nonsmooth areas of $\mathcal{X}$, generally outside the current attraction basin. In this formulation, for $\lambda > 0$, the feasible space is extended by the uncertainties in the constraints; the uncertainties in the objective are considered as potential improvements of the objective. Note that Formulation $(\text{SP1-F}\sigma)$ is equivalent to the problem of Equation $(\hat{P})$ when $\lambda = 0$.

### 3.2 Probability of feasibility

Another way to handle the constraints is to use the cumulative density function provided by the surrogate model to estimate the probability of a point being feasible. The value $P[c_j(x) \leq 0]$ is provided by the model and represents the probability that the constraint $c_j$ is satisfied at $x$. An estimation of the probability that $x$ is feasible is computed by

$$
P(x) = \prod_{j \in J} P[c_j(x) \leq 0] \approx P[c_j(x) \leq 0, \forall j \in J].
$$

(2)

If the constraints are statistically independent, the above approximation is exact. Since we are considering blackbox output, it cannot be assumed that there is no correlation between the constraints; however, $P(x)$ is the best approximation available. It is worth mentioning that the probability of feasibility of a point can also be estimated by building a model of an aggregate constraint $h(x)$ and by computing $P(x) = P[h(x) \leq 0]$. Several definitions are possible:

$$
h(x) = \sum_{j \in J} \max \{c_j(x); 0\}^2,
$$

(3)

$$
h(x) = \max_{j \in J} \{c_j(x)\}, \text{ or }
$$

(4)

$$
h(x) = \begin{cases} 
1 & \text{if } c_j(x) \leq 0, \ \forall j \in J, \\
0 & \text{otherwise.}
\end{cases}
$$

(5)

Aggregate constraints enable modeling feasibility by building just one surrogate model rather than $m$. This can reduce the computational time and avoid the question of the independence of the constraints. However, it also implies that fewer data contribute to building the model, which makes it less accurate than multiple-constraint surrogate models. Preliminary tests with dynaTree models have shown that building one model per constraint is more efficient. Thus, we use Equation (2) to treat the constraints in this study.

This estimation of the probability of feasibility can be used as a chance constraint, meaning that a candidate must satisfy a constraint on $P(x)$, regardless of its objective. This leads to
the formulation of the surrogate problem

$$\min_{x \in \mathcal{X}} \hat{f}(x) - \lambda \hat{\sigma}_f(x)$$
subject to $P(x) \geq p_c.$

(SP2-FσP)

This formulation indicates that only candidates likely to be feasible will be evaluated. As a consequence, the candidates will remain distant from the boundary of the feasible domain and will approach it only when $\sigma_j$ decreases. The choice of $p_c$ can depend on the problem size and the number of constraints. If $p_c$ is too high, the constraint on $P$ can be impossible to satisfy, particularly at the beginning of the optimization when the constraints are uncertain. However, if $p_c$ is too low, the candidate will rarely be in the feasible domain, leading to an inefficient search. In this study, we choose $p_c = 0.5$, which means that after the entire optimization run half of the points $x_k^{SP}$ will be feasible, ensuring improvement of the models inside and outside the feasible domain.

### 3.3 Expected improvement

*Improvement* is defined by

$$I(x) = \max\{f_{min} - f(x), 0\},$$

where $f_{min}$ is the objective function value of the currently best feasible point [9]. In the context of global optimization, evaluating a point that does not improve the objective is not considered counterproductive since this evaluation enhances the information about the problem [7, 8]. Evaluating a point that improves the objective is a step forward in the optimization and narrows the area where a global optimum can be found. Thus, the utility of evaluating a new point is always positive. This principle is manifested in the definition of $EI$, which is considered a major relevance criterion in global optimization [7, 8, 15]:

$$EI(x) = \mathbb{E}[I(x)] = \int_0^{+\infty} I \phi_f(f_{min} - I) dI,$$

where $\phi_f$ is the probability density function of $f$, provided by the surrogate model. The formulation described in (1) can be adapted to handle constraints:

$$\min_{x \in \mathcal{X}} -EI(x) - \lambda \hat{\sigma}_f(x)$$
subject to $\hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0.$

(SP3-EIσ)

The expected feasible improvement ($EFI$) considers in a single scalar criterion the objective and feasibility of a candidate [9]:

$$EFI(x) = EI(x) \times P(x).$$

The $EFI$ represents a tangible measure of the relevance of a candidate in the context of constrained optimization. A promising candidate can be found by maximizing the $EFI$. This leads to an unconstrained formulation of the surrogate problem:

$$\min_{x \in \mathcal{X}} - EFI(x).$$

(SP4-EFI)

Maximizing the $EFI$ is an efficient method, but it would also be interesting to incorporate the exploration term proposed in (1) and used in the previous formulations:

$$\min_{x \in \mathcal{X}} - EFI(x) - \lambda \hat{\sigma}_f(x).$$

(SP5-EFIσ)
The drawback of (SP5-\(EFl\sigma\)) is that the exploration term \(\hat{\sigma}_f(x)\) does not take into account uncertainties in the constraints. To address this, \(\hat{\sigma}_f(x)\) could be replaced by a norm on \([\hat{\sigma}_f(x), \hat{\sigma}_1(x), \ldots, \hat{\sigma}_n(x)]\), but the uncertainty in the value of \(c_j\) is less significant than the uncertainty in the feasibility of the candidate. Given that the event “\(x\) is feasible” follows a Bernoulli law of probability \(P(x)\), its variance is \(P(x)P(x) \in [0, 1/4]\). Thus, we propose a measure of the uncertainty in the feasibility \((\mu)\):

\[
\mu(x) = 4 \cdot P(x) \cdot \overline{P(x)}.
\]  

(7)

The multiplication by four is intended to normalize \(\mu\) in \([0, 1]\). The larger \(\mu\) is, the more uncertain is the feasibility of the candidate, which means that we cannot predict whether the candidate is feasible. \(\mu(x)\) is maximal for \(P(x) = 1/2\) and null for \(P(x) \in \{0,1\}\). Figure 3 illustrates this concept for a single constraint \(c(x)\).

Figure 3: Probability of feasibility and uncertainty in feasibility. \(\mu(x)\) is maximal for \(x = 4\), where \(\hat{c}(x) = 0\). In the neighborhood of \(x = 7\), despite the sharp variation in \(c\), the feasibility is predictable, so \(\mu(x)\) is small.

Using this measure, two formulations are derived. In the first formulation, the exploration term is multiplied by \(\mu(x)\):

\[
\min_{x \in X} -EFl(x) - \lambda \hat{\sigma}_f(x) \mu(x).
\]  

(\(SP6-EFl\mu\))

This formulation encourages a search for candidates that are uncertain both in the objective and the feasibility. The drawback is that a promising candidate that is uncertain in only one of the two measures (objective or feasibility) will not be considered. To address this, the crossed formulation (\(SP7-EFIC\)) is introduced:

\[
\min_{x \in X} -EFl(x) - \lambda \left(EI(x)\mu(x) + P(x)\hat{\sigma}_f(x)\right).
\]  

(\(SP7-EFIC\))

The first part of the exploration term, \(EI(x)\mu(x)\), favors candidates that have a promising objective and unpredictable feasibility. In contrast, \(P(x)\hat{\sigma}_f(x)\) favors candidates with an uncertain objective and good feasibility.
3.4 Probability of improvement

The last formulation, based on [7], focuses on the probability of improvement (PI):

\[ PI(x) = \mathbb{P}(I(x) > 0). \]

As for the EFI, the feasibility of the point can be taken into account by defining the probability of feasible improvement (PFI):

\[ PFI(x) = PI(x) \times P(x). \]

This leads to the surrogate problem formulation

\[ \min_{x \in X} - PFI(x). \]  \hspace{1cm} (SP8-PFI)

Unlike the EFI, maximizing the PFI will bring small but certain improvements to the solution. As the optimization proceeds, the PFI will remain significant, but the related improvement will become negligible.

4 Numerical experiments

The proposed formulations are tested on a set of 20 analytical problems and 2 MDO applications related to aircraft design. These 22 problems are constrained; they may involve nonsmooth functions, may have several local optima, and may exhibit some numerical noise. To generate more results for the 2 MDO applications, we run a total of 100 optimizations by specifying 50 different feasible starting points for each application, using Latin hypercube sampling [21]. Thus, each formulation is tested on three sets of 20, 50, and 50 optimization runs, respectively. For each optimization, we allow 1000 blackbox evaluations, but the optimization can stop earlier if the convergence criterion for the mesh size is satisfied. The two MDO applications have a computational time of 60 ms (Simplified Wing) and 5 ms (Aircraft Range) per evaluation on a standard desktop PC (Intel Core i7-2600, 16 Gb).

The numerical results were obtained using the MADS implementation of the NOMAD software package [17, 29] and the R dynamicTree library [6] for building the statistical surrogate models. The different formulations are compared to MADS without a search step, and to MADS with the use of quadratic models inside the search step as defined in [13], which is denoted “Quad.” The formulations (SP1-Fσ), (SP2-FσP), and (SP3-EIσ) are tested for \( \lambda \in \{0, 0.01, 0.1, 1\} \). The formulations (SP5-EFIσ), (SP6-EFIμ), and (SP7-EFIIC) are equivalent to formulation (SP4-EFI) for \( \lambda = 0 \); therefore, they are tested for \( \lambda \in \{0.01, 0.1, 1\} \). A total of \( S = 25 \) formulations are tested in this work, as summarized in Table 4.

4.1 Problem description

The 20 analytical constrained optimization problems are described in Table 4.1, where \( f^* \) denotes the optimum. Problems HS73 and HS114 include a linear equality constraint, which has been used to eliminate one optimization variable. Thus, only the inequality constraints have to be treated explicitly.

The Simplified Wing problem [33] involves optimizing the geometry of a wing to minimize drag. The two disciplines involved are wing structures and aerodynamics. This problem is smooth but has many local minima. The best objective found in this study (for all formulations and initial guesses) is \( f^* = -16.60 \). The best objective value reported in [33] is
Table 1: List of formulations.

<table>
<thead>
<tr>
<th>(Eq.) Section</th>
<th>Formulation</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MADS</td>
<td>MADS with no search [4]</td>
<td>N.A.</td>
</tr>
<tr>
<td>Quad</td>
<td>MADS with quadratic model [13]</td>
<td>N.A.</td>
</tr>
<tr>
<td>(SP1-(F_\sigma)) 3.1</td>
<td>( \min_{x \in \mathcal{X}} \hat{f}(x) - \lambda \hat{\sigma}_f(x) ) ( \text{st: } \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0 )</td>
<td>( \lambda \in {0, \frac{1}{100}, \frac{1}{10}, 1} )</td>
</tr>
<tr>
<td>(SP2-(F_\sigma P)) 3.2</td>
<td>( \min_{x \in \mathcal{X}} \hat{f}(x) - \lambda \hat{\sigma}_f(x) ) ( \text{st: } P(x) \geq p_f )</td>
<td>( \lambda \in {0, \frac{1}{100}, \frac{1}{10}, 1} )</td>
</tr>
<tr>
<td>(SP3-(E_\sigma)) 3.3</td>
<td>( \min_{x \in \mathcal{X}} -EI(x) - \lambda \hat{\sigma}_f(x) ) ( \text{st: } \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0 )</td>
<td>( \lambda = 0 )</td>
</tr>
<tr>
<td>(SP4-(E_{FI})) 3.3</td>
<td>( \min_{x \in \mathcal{X}} -E_{FI}(x) )</td>
<td>( \lambda = 0 )</td>
</tr>
<tr>
<td>(SP5-(E_{FI\sigma})) 3.3</td>
<td>( \min_{x \in \mathcal{X}} -E_{FI}(x) - \lambda \hat{\sigma}_f(x) )</td>
<td>( \lambda \in {0, \frac{1}{100}, \frac{1}{10}, 1} )</td>
</tr>
<tr>
<td>(SP6-(E_{FI\mu})) 3.3</td>
<td>( \min_{x \in \mathcal{X}} -E_{FI}(x) - \lambda \hat{\sigma}_f(x) \mu(x) )</td>
<td>( \lambda \in {0, \frac{1}{100}, \frac{1}{10}, 1} )</td>
</tr>
<tr>
<td>(SP7-(E_{FIC})) 3.3</td>
<td>( \min_{x \in \mathcal{X}} -E_{FI}(x) - \lambda \left( EI(x)\mu(x) + P(x)\hat{\sigma}_f(x) \right) )</td>
<td>( \lambda = 0 )</td>
</tr>
<tr>
<td>(SP8-(P_{FI})) 3.4</td>
<td>( \min_{x \in \mathcal{X}} -P_{FI}(x) )</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

\( f^* = -16.65 \). The problem can be summarized as

\[
\begin{align*}
\min & \quad \text{Wing drag} \\
\text{subject to} & \quad \text{Shear stress} \leq 73,200 \text{ psi} \\
& \quad \text{Tensile stress} \leq 47,900 \text{ psi} \\
& \quad \text{Sum of the weights} \leq \text{total lift}.
\end{align*}
\]

Two structural constraints guarantee wing integrity, and a constraint on the lift ensures sustentation. Table 4.1 lists the \( n = 7 \) design optimization variables, their bounds, and the known optimal values. Five of the variables are related to the aerodynamics properties of the wing; the two other describe its structure.

The Aircraft Range problem [34] considers the design of a supersonic business jet by taking into account aerodynamics, structure, and propulsion. The problem is not smooth and has several local optima. The best objective value found in this work for all formulations and initial guesses is \( f^* = -3964.20 \). The best objective value reported in [33] is \( f^* = -3963.98 \). The range of the aircraft must be maximized while satisfying \( m = 10 \) constraints related to
Table 2: List of analytical optimization problems.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Source</th>
<th>n</th>
<th>m</th>
<th>Bounds</th>
<th>Smooth</th>
<th>$f^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G2</td>
<td>[30]</td>
<td>10</td>
<td>3</td>
<td>yes</td>
<td>no</td>
<td>-0.740466</td>
</tr>
<tr>
<td>2</td>
<td>MAD6</td>
<td>[31]</td>
<td>5</td>
<td>8</td>
<td>no</td>
<td>no</td>
<td>0.101831</td>
</tr>
<tr>
<td>3</td>
<td>PENTAGON</td>
<td>[31]</td>
<td>6</td>
<td>16</td>
<td>no</td>
<td>no</td>
<td>-1.85962</td>
</tr>
<tr>
<td>4</td>
<td>SNAKE</td>
<td>[31]</td>
<td>2</td>
<td>3</td>
<td>no</td>
<td>yes</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>HS24</td>
<td></td>
<td>2</td>
<td>4</td>
<td>no</td>
<td>yes</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>HS34</td>
<td></td>
<td>3</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
<td>-0.833795</td>
</tr>
<tr>
<td>7</td>
<td>HS36</td>
<td></td>
<td>3</td>
<td>2</td>
<td>yes</td>
<td>yes</td>
<td>-3300</td>
</tr>
<tr>
<td>8</td>
<td>HS37</td>
<td></td>
<td>3</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
<td>-3455.51</td>
</tr>
<tr>
<td>9</td>
<td>HS64</td>
<td></td>
<td>3</td>
<td>2</td>
<td>no</td>
<td>yes</td>
<td>6299.94</td>
</tr>
<tr>
<td>10</td>
<td>HS66</td>
<td></td>
<td>3</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
<td>0.532397</td>
</tr>
<tr>
<td>11</td>
<td>HS72</td>
<td></td>
<td>4</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
<td>727.701</td>
</tr>
<tr>
<td>12</td>
<td>HS73</td>
<td>[31]</td>
<td>3</td>
<td>4</td>
<td>no</td>
<td>no</td>
<td>29.8944</td>
</tr>
<tr>
<td>13</td>
<td>HS86</td>
<td>[31]</td>
<td>5</td>
<td>11</td>
<td>no</td>
<td>yes</td>
<td>-32.2879</td>
</tr>
<tr>
<td>14</td>
<td>HS93</td>
<td></td>
<td>6</td>
<td>3</td>
<td>no</td>
<td>yes</td>
<td>135.075961</td>
</tr>
<tr>
<td>15</td>
<td>HS101</td>
<td></td>
<td>7</td>
<td>7</td>
<td>yes</td>
<td>yes</td>
<td>1809.76</td>
</tr>
<tr>
<td>16</td>
<td>HS102</td>
<td></td>
<td>7</td>
<td>7</td>
<td>yes</td>
<td>yes</td>
<td>911.88</td>
</tr>
<tr>
<td>17</td>
<td>HS103</td>
<td></td>
<td>7</td>
<td>7</td>
<td>yes</td>
<td>yes</td>
<td>543.67</td>
</tr>
<tr>
<td>18</td>
<td>HS104</td>
<td></td>
<td>8</td>
<td>7</td>
<td>yes</td>
<td>yes</td>
<td>4.02305</td>
</tr>
<tr>
<td>19</td>
<td>HS105</td>
<td></td>
<td>8</td>
<td>4</td>
<td>yes</td>
<td>yes</td>
<td>1136.36</td>
</tr>
<tr>
<td>20</td>
<td>HS114</td>
<td></td>
<td>9</td>
<td>7</td>
<td>yes</td>
<td>no*</td>
<td>-1192.28</td>
</tr>
</tbody>
</table>

Table 3: Design optimization variables for the Simplified Wing MDO problem.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Bounds</th>
<th>$x^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower</td>
<td>Upper</td>
</tr>
<tr>
<td>Wing span</td>
<td>30</td>
<td>45</td>
</tr>
<tr>
<td>Root chord</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>Taper ratio</td>
<td>0.28</td>
<td>0.50</td>
</tr>
<tr>
<td>Angle of attack at root</td>
<td>-1</td>
<td>3</td>
</tr>
<tr>
<td>Angle of attack at tip</td>
<td>-1</td>
<td>3</td>
</tr>
<tr>
<td>Tube external diameter</td>
<td>1.6</td>
<td>5.0</td>
</tr>
<tr>
<td>Tube thickness</td>
<td>0.3</td>
<td>0.79</td>
</tr>
</tbody>
</table>

structure, engine, and performance. The problem can be summarized as

$$\text{max} \quad \text{Aircraft range}$$
$$\text{subject to} \quad \text{Normalized stress} \leq 1.09 \ (5 \ \text{constraints})$$
$$0.96 \leq \text{Normalized wing twist} \leq 1.04$$
$$\text{Pressure gradient} \leq 1.04 \ \text{Pa.m}^{-1}$$
$$0.5 \leq \text{Eng. scale factor} \leq 1.5$$
$$\text{Normalized engine temperature} \leq 1.02$$
$$\text{Throttle setting} \leq \text{max throttle,}$$

where the max throttle is computed based on the altitude and Mach number using polynomial
regression on Pratt & Whitney data [35]. The problem has $n = 10$ design optimization variables, listed in Table 4.1 along with their bounds and known optimal values. Seven of them describe the wing aerodynamic properties. The others describe the flight conditions: engine command, altitude, and speed.

Table 4: Design optimization variables for the Aircraft Range MDO problem.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Bounds</th>
<th>$x^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower</td>
<td>Upper</td>
</tr>
<tr>
<td>Taper ratio</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>Wingbox cross-section</td>
<td>0.75</td>
<td>1.25</td>
</tr>
<tr>
<td>Skin friction coeff.</td>
<td>0.75</td>
<td>1.25</td>
</tr>
<tr>
<td>Throttle</td>
<td>0.1</td>
<td>1.0</td>
</tr>
<tr>
<td>Thickness/chord</td>
<td>0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>Altitude</td>
<td>30000</td>
<td>60000</td>
</tr>
<tr>
<td>Mach number</td>
<td>1.4</td>
<td>1.8</td>
</tr>
<tr>
<td>Aspect ratio</td>
<td>2.5</td>
<td>8.5</td>
</tr>
<tr>
<td>Wing sweep</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>Wing surface area</td>
<td>50</td>
<td>1500</td>
</tr>
</tbody>
</table>

4.2 Quantifying deviation from the best known solution

Comparisons between the formulations are performed independently on the three sets of optimization runs using statistics of deviation from the best known solution, data profiles, and performance profiles [36].

In each set, the optimization runs are denoted $p \in \{1, \ldots, P\}$, with $P = 20$ for the set of analytical problems and $P = 50$ for each of the MDO problems. The dimension of the design space is denoted $n_p$. $f_p^*$ is the best feasible objective value found among all formulations for optimization run $p$. To take into account the different values of $n_p$, the progress of the optimization is represented by the number $i \in \{1, \ldots, i_{\text{max}}\}$ of groups of $(n_p + 1)$ evaluations, which is equivalent to the number of simplex gradient estimates (SGEs) [36].

The formulations are denoted by $s \in \{1, \ldots, S\}$, where $S = 25$. For each formulation $s$, the best value of the objective for optimization run $p$ after $i$ SGEs is denoted $f_{p,s,i}$, which is infinite if no feasible point has been found. The best possible objective value $f^* = 0$ for the SNAKE problem has not been reached by any of the formulations; thus, $f_p^* \neq 0$ for all optimization runs. This allows to define the relative deviation from the best known solution as

$$d_{p,s,i} = \min \left\{ \frac{f_{p,s,i} - f_p^*}{|f_p^*|}, 1 \right\}.$$ 

To remove outliers, the deviation is bounded. This allows us to compute deviation statistics.

4.3 Statistics of deviation from the best known solution

For formulation $s$, the average deviation from the best known solution of all optimization runs is defined as

$$d_{s}^{\text{mean}} = \frac{1}{P} \sum_{p=1}^{P} d_{p,s,i_{\text{max}}}.$$
The maximum deviation $d_{s}^{\text{max}}$ and the standard deviation of the deviation $d_{s}^{\text{std}}$ are defined accordingly. Table 4.3 reports deviation statistics (in %) for each formulation and for the three sets of optimization runs. In each column, the formulations that are better (worse) than MADS or Quad are followed by the sign (+) (preceded by the sign (−)). The best value of each column is highlighted in bold.

Table 5: Relative deviation (%) for the three sets of optimization runs. Formulations that are better (worse) than MADS or Quad are followed by the sign (+) (preceded by the sign (−)). The best value in each column is highlighted in bold.

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Analytical problems</th>
<th>MDO Simp. Wing.</th>
<th>MDO Aircraft Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$</td>
<td>$d_{m}^{\text{max}}$</td>
<td>$d_{m}^{\text{mean}}$</td>
</tr>
<tr>
<td>MADS</td>
<td>N.A.</td>
<td>100</td>
<td>28.7</td>
</tr>
<tr>
<td>Quad</td>
<td>N.A.</td>
<td>100</td>
<td>27.4</td>
</tr>
<tr>
<td>(SP1-Fs)</td>
<td>0.0</td>
<td>100</td>
<td>21.6</td>
</tr>
<tr>
<td>(SP2-FsP)</td>
<td>0.0</td>
<td>100</td>
<td>26.2</td>
</tr>
<tr>
<td>(SP3-Elo)</td>
<td>0.0</td>
<td>100</td>
<td>20.9</td>
</tr>
<tr>
<td>(SP4-Eff)</td>
<td>0.0</td>
<td>100</td>
<td>28.0</td>
</tr>
<tr>
<td>(SP5-Eff)</td>
<td>0.0</td>
<td>100</td>
<td>25.3</td>
</tr>
<tr>
<td>(SP6-Eff)</td>
<td>0.0</td>
<td>100</td>
<td>26.6</td>
</tr>
<tr>
<td>(SP7-Eff)</td>
<td>0.0</td>
<td>100</td>
<td>28.1</td>
</tr>
<tr>
<td>(SP8-Jff)</td>
<td>N.A.</td>
<td>100</td>
<td>25.7</td>
</tr>
</tbody>
</table>

### 4.4 Data and performance profiles

The data and performance profiles are based on the proportion of problems solved by formulation $s$, after $i$ groups of $(n+1)$ evaluations, for a given precision $\tau$:

$r_{s,i}(\tau) = \frac{1}{P} \# \{ p \in \{1,...,P\} : d_{p,s,i} \leq \tau \},$

where $\tau$ represents the tolerance on the deviation $d_{p,s,i}$. If the tolerance decreases, the number of optimization runs $p$ satisfying the condition $d_{p,s,i} \leq \tau$ will also decrease. For a given $\tau$, the proportion $r_{s,i}(\tau)$ varies depending on the formulation $s$ and on the number $i$ of SGEs. As the optimization proceeds, the proportion is likely to increase since more evaluations are performed. For a given $\tau$ and $i$, $r_{s1,i}(\tau) > r_{s2,i}(\tau)$ means that formulation $s_1$ yields better results than $s_2$. In each profile, the proportion $r_{s,i}(\tau)$ is plotted for several formulations $s$ in order to compare them.

In the data profiles, the value of $i$ varies in order to compare the formulations at various times of the optimization. The tolerance $\tau$ is fixed and can take the values $\{10^{-1}, 10^{-3}, 10^{-7}\}$.
Each curve in the profile represents the function $i \rightarrow r_{s,i}(\tau)$ for a formulation $s$. The $x$-axis specifies the number $i$ and the $y$-axis indicates the ratio $r_{s,i}(\tau)$.

In the performance profiles, $\tau$ varies in order to compare the formulations for various tolerances. The progress of the optimization is fixed at $i = i_{\text{max}}$, which enables a comparison of the formulations in terms of performance. The tolerance $\tau$ varies in $[10^{-7}, 10^{-1}]$. Each curve in the profile represents the function $\tau \rightarrow r_{s,i_{\text{max}}}(\tau)$ for a formulation $s$. The $x$-axis represents the tolerance $\tau$. As in the data profiles, the $y$-axis indicates the proportion $r_{s,i}(\tau)$.

These profiles are linked since $r_{s,i_{\text{max}}}(\tau)$ appears both at the end of the data profile for the precision $\tau$ and in the performance profile at abscissa $\tau$.

Given the number of formulations presented, the profiles can be displayed in a visually meaningful manner only for a small number of formulations. Therefore, for each set of optimization runs, the data and performance profiles are plotted for MADS, Quad, and two formulations: the best and the worst according to the mean deviation $d_{s}^{\text{mean}}$ (Figures 4, 5, and 6).

![Data profile, $\tau = 10^{-1}$](image1)

![Data profile, $\tau = 10^{-3}$](image2)

![Data profile, $\tau = 10^{-7}$](image3)

![Performance profile after 1000$n$ evaluations](image4)

Figure 4: Data and performance profiles for the set of analytical problems. MADS and Quad are used as a reference; (SP1-$F_\sigma$, $\lambda = 1.0$) and (SP3-$E_{\lambda}$, $\lambda = 0.01$) are the formulations with the best and worst mean deviation, respectively.
4.5 Discussion

If the convergence criteria are met, the optimization algorithm stops before the budget of evaluations is consumed. On the three sets of optimization runs, the mean number of SGEs per run is 396, 615, and 416, respectively. We observe a slight negative correlation (-4%, -4%, and -6% on the three sets) between the mean deviation of a formulation $s$ and the mean number of SGEs. This illustrates the need to explore the design space: if no mechanism enables a search for a better solution outside the current attraction basin, the algorithm may converge to a local optimum.

All the formulations have a deviation of 100% on at least one of the following analytical problems: HS101, HS102, HS103, and SNAKE. Thus, the maximum deviation is 100% for all the formulations. For the mean deviation, all the formulations but one perform better than MADS. The formulations (SP1-$F\sigma$), (SP2-$F\sigma P$), and (SP6-$E F_l \mu$) perform better than MADS and Quad. In particular, formulation (SP3-$E l \sigma$, $\lambda = 0.01$) with $\lambda = 1$ decreases the mean deviation.

Figure 5: Data and performance profiles for the simplified wing MDO problem. MADS and Quad are used as a reference; (SP3-$E l \sigma$, $\lambda = 0.01$) and (SP3-$E l \sigma$, $\lambda = 1.0$) are the formulations with the best and worst mean deviation, respectively.
deviation by more than half. The data profiles show that the solution of the worst statistical formulation \( \text{SP1-FI}, \lambda = 1 \) is at most 5% worse than that of the MADS algorithm. The best statistical formulation \( \text{SP3-EI}, \lambda = 0.01 \) performs better than or as well as Quad for tolerances between \( \tau = 10^{-1} \) and \( \tau = 10^{-3} \), but it is outperformed by Quad for small tolerances \( \tau = 10^{-7} \). This illustrates the fact that, on analytical problems, once a good attraction basin is found, Quad can converge quickly toward a local optimum.

The statistical formulations exhibit a significant advantage for the simplified-wing MDO runs. All but one of the formulations yield a smaller deviation than that of MADS and Quad. Most formulations provide a reduction of more than 10% in the mean deviation. The data and performance profiles show that Quad outperforms MADS for large tolerances, but the opposite occurs for small tolerances. Similarly, if we consider the best and worst statistical formulations according to the mean deviation, formulation \( \text{SP3-EI}, \lambda = 1 \) outperforms \( \text{SP3-EI}, \lambda = 0.01 \) for large tolerances, and conversely for small tolerances.

The steep curves between \( \tau = 10^{-1} \) and \( \tau = 10^{-3} \) in the performance profile of Figure 5
illustrate the existence of multiple local minima. Since the problem is not noisy, if the algorithm finds the proper attraction basin, it can quickly reach an accuracy of $10^{-7}$. Otherwise, it is unlikely to reach a deviation smaller than $10^{-3}$. There is no variation in the proportion of solved problems below $\tau = 10^{-4}$.

Finally, for the aircraft-range MDO executions, the mean and maximum deviations yielded by the statistical formulations can be up to three times higher than that of MADS or Quad. However, some formulations are very efficient: 10 formulations are better, 6 are five times better, and (SP5-$\text{EFI}_\sigma$) with $\lambda = 0.1$ divides the maximum deviation by 22 and the mean deviation by 29. For the data profiles in Figure 6, (SP5-$\text{EFI}_\sigma$, $\lambda = 1$) shows a significant advantage: for $\tau = 0.1$, a proportion of 100% is reached in less than 100 SGEs. The worst formulation, (SP5-$\text{EFI}_\sigma$, $\lambda = 0.1$), is outperformed by MADS and Quad for $\tau = 10^{-1}$ and $\tau = 10^{-3}$ but performs better for $\tau = 10^{-7}$. This illustrates the efficiency of a robust regression on noisy functions. In this problem, the statistical formulations show a significant advantage over MADS and Quad.

5 Conclusion

This work introduces several novel problem formulations for using statistical surrogates and the MADS derivative-free optimization algorithm in a surrogate management framework for simulation-based design optimization. These formulations take advantage of the statistical features of the surrogate and emphasize the exploration of the design space. The presented surrogate management framework formulations can be used with any direct search method based on the search-and-poll paradigm. They have been implemented using the dynaTree library to build the statistical surrogate models, and were tested on 20 analytical problems and 2 simulation-based MDO problems. They generally perform as good as or better than existing formulations but seem to exhibit significant advantages when used to solve nonsmooth, noisy and nonconvex problems.

In future work, the surrogate search could be improved by updating the exploration parameter $\lambda$ depending on the result of the search and on the smoothness of the blackbox outputs. A similar strategy could be applied to the parameter $p_c$ involved in the chance constraint of formulation (SP2-$\text{F}_{\sigma P}$). The correlation between the constraints could be analyzed to provide a more accurate estimation of the probability of feasibility. Finally, further experimentations may allow to build a decision process to chose the most promising formulation depending on the characteristics and features of different problems and applications. Finally, we would like to note that the integration of the dynaTree statistical surrogate modeling tool with the MADS algorithm will be available in a future release of the free NOMAD software package [17, 29].

Acknowledgements

This work is partially supported by NSERC Discovery Grants 418250-2012 and 436193-2013 and by a GERAD postdoctoral fellowship; such support does not constitute an endorsement by the sponsors of the opinions expressed in this article. The authors would like to thank Prof. Charles Audet of GERAD and École Polytechnique for his useful comments and Prof. Robert Gramacy of the University of Chicago for his help with implementing dynaTree within NOMAD.
References


