High-dimensional Black-box Optimization Under Uncertainty

Hadis Anahideh · Jay Rosenberger · Victoria Chen

Received: date / Accepted: date

Abstract Optimizing expensive black-box systems with limited data is an extremely challenging problem. As a resolution, we present a new surrogate optimization approach by addressing two gaps in prior research—unimportant input variables and inefficient treatment of uncertainty associated with the black-box output. We first design a new flexible non-interpolating parsimonious surrogate model using a partitioning-based multivariate adaptive regression splines approach, Tree Knot MARS (TK-MARS). The proposed model is specifically designed for optimization by capturing the structure of the function, bending at near-optimal locations, and is capable of screening unimportant input variables. Furthermore, we develop a novel replication approach called Smart-Replication, to overcome the uncertainty associated with the black-box output. The Smart-Replication approach identifies promising input points to replicate and avoids unnecessary evaluations of other data points. Smart-Replication is agnostic to the choice of a surrogate and can adapt itself to an unknown noise level. Finally to demonstrate the effectiveness of our proposed approaches we consider different complex global optimization test functions from the surrogate optimization literature. The results indicate that TK-MARS outperforms original MARS within a surrogate optimization algorithm and successfully detects important variables. The results also show that although non-interpolating surrogates can mitigate uncertainty, replication is still beneficial for optimizing highly complex black-box functions. The robustness and the quality of the final optimum solution found through Smart-Replication are competitive with that using no replications in environments with low levels of noise and using a fixed number of replications in highly noisy environments.

H. Anahideh
University of Illinois - Chicago
E-mail: hadis@uic.edu

J. Rosenberger
University of Texas at Arlington
E-mail: jrosenbe@uta.edu

V. Chen
University of Texas at Arlington
E-mail: vchen@uta.edu
Keywords Surrogate optimization · Black-box functions · Derivative-free · Limited data · Non-interpolating model · Uncertainty

1 Introduction

Challenging optimization problems appear in many areas of science, technology, and industry. Examples include optimizing designs of wind farms [1], autonomous vehicle control systems [2], green buildings [3], vehicle safety systems [4], molecular structures in pharmaceuticals [5], and material structures [6]. Most of these problems include a complex system of inputs and outputs without well-known information about the underlying system behavior. Optimizing such problems is sometimes referred to as Black-Box Optimization (BBO). In this situation, the underlying function is complex yet unknown, a large number of input variables are involved, and there are substantial interactions between the input variables. On the other hand, the evaluation process includes costly experiments, which can be either based upon computer simulators, such as finite element simulation tools [7], or actual experiments, such as crash simulators [8]. Achieving a near-optimal solution of a high-dimensional expensive black-box function within a limited number of function evaluations is the primary goal of this and other BBO research.

The BBO problem formulation we address in this research has a single objective, (1), and a box constraint, (2), and is given by:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad a \leq x \leq b, \forall x \in \mathbb{R}^d.
\end{align*}
\]

Here \( f(x) \) is the black-box function, and the goal is to obtain a global optimal solution of \( f(x) \) in the feasible input space \( D \), where \( D = \{ x \in \mathbb{R}^d : a_j \leq x_j \leq b_j, \forall j = 1, \ldots, d \} \). \( a, b \in \mathbb{R}^d \) represent the lower bound and the upper bound of \( x \), which is a \( d \)-dimensional vector of input variables.

Since computing \( f(x) \) is often expensive, a practical BBO approach is to employ a surrogate model that approximates \( f(x) \) but is less expensive to evaluate. A surrogate model is a mathematical approximation of the relationship between the input and the output variables. Gaussian Processes (GP), also referred to as Kriging [9], Radial Basis Function (RBF) [10], Regression Trees [11], Multivariate Adaptive Regression Splines (MARS) [12], Artificial Neural Networks [13], and Support Vector Regression [14] are examples of surrogate models based upon statistical modeling. Optimizing the cheap to evaluate surrogate models for BBO is one of the existing well-known derivative-free techniques called surrogate optimization. Surrogate optimization requires careful selection of the candidate points for evaluation to simultaneously improve the approximation performance and find a potential optimum, within a few evaluations. An exploration and exploitation trade-off is required to find the most promising data points for black-box function evaluation. In §2.9, we will elaborate on different input selection strategies for surrogate optimization.

Most of the existing surrogate optimization methods assume that there is no uncertainty in the black-box system, especially in higher dimensional space, and the set of important input variables is known \emph{a priori} [15][16][17]. Both of these
assumptions are often unrealistic in real-world problems [18]. One such example, is
the hyper-parameter tuning of deep learning for image processing [19], which is a
stochastic complex system. In one other example, Pilla et. al. [20] shows that, in a
fleet assignment case study, only 42 out of 1264 variables remain significant. This
research introduces a new surrogate optimization paradigm to address these two
primary concerns. One surrogate model that has the potential to overcome both
of these concerns is MARS [12]. MARS is a parsimonious and non-interpolating
model, implicitly capable of screening unimportant input variables. Moreover, as
a non-interpolating model, MARS is able to mitigate noise effects that are present
in the objective values. MARS has rarely been used in the surrogate optimization
context. Müller et al. [17] studied the performance of MARS as a single surro-
gate and showed that MARS in its original form is inferior compared with other
surrogates, like Radial Basis Functions and Gaussian Processes. However, in this
research, we develop a partitioning-based MARS, called Tree-Knot Multivariate
Adaptive Regression Splines (TK-MARS), to tailor MARS for the purpose of sur-
rogate optimization and improve its performance. In addition, we develop a smart
replication strategy to mitigate the uncertainty of the black-box output. The ro-
bustness and the quality of the final optimum solution found through our proposed
smart replication approach is competitive with using no replications in environ-
ments with low levels of noise and using a fixed number of replications in highly
noisy environments. There are two notable advantages of the smart replication ap-
proach. One, it is agnostic to the choice of a surrogate model, and two, it adapts
to the noise level that is present in the data. Hence, if the noise level is low it does
not replicate, but if the noise level is high it replicates extensively.

The remainder of this paper is organized as follows. We introduce a general
surrogate optimization algorithm and related literature in §2 to highlight the gaps
in prior research. In §3, we describe TK-MARS and the proposed replication ap-
proach for black-box functions under uncertainty. Finally, in §4, we provide results
of computational experiments for the performance of the proposed approach.

2 Background

In this section, we first provide some technical background on surrogate opti-
mization. We then further describe its core components that we focus on in
this research—choosing an appropriate surrogate model (interpolating and non-
interpolating models), as well as an exploration-exploitation Pareto sampling tech-
teque. We also provide a brief background on Classification and Regression Trees
(CART) [11], which are used as a partitioning technique in our proposed modified
MARS approach §3.1.

2.1 General Surrogate Optimization Framework

Surrogate optimization is a derivative-free optimization technique, which applies
cheap to evaluate surrogate models to optimize expensive black-box functions.
Surrogate optimization is referred to as one-shot optimization when it optimizes
over a fixed well-designed pool of sample points [21]. In other settings, surrogate
optimization uses sequential model-based adaptive sampling [15]. Algorithm [1] de-
picts a generic sequential surrogate optimization algorithm, which is the focus of
this research. In this setting, we assume the black-box function is deterministic.
In Step0 a set of N input points is generated using a space-filling technique, such
Algorithm 1 Surrogate Optimization

1: \( I = \{x^1, \ldots, x^N\} \), a set of \( N \) input points in \( D \), selected with a Design of Experiment (DOE) method
2: \( \mathcal{F} = \{f(x^i)|x^i \in I\} \)
3: while Termination criteria not satisfied do
4: \textbf{Surrogate:} Fit a surrogate model \( \hat{f} \) on \( (I, \mathcal{F}) \)
5: \textbf{Sampling:} Determine new candidate points, \( P \)
6: \textbf{Evaluation:} \( \mathcal{F}_P = \{f(x^i)|x^i \in P\} \)
7: \( I = I \cup P; \mathcal{F} = \mathcal{F} \cup \mathcal{F}_P \)
8: \textbf{Best Known Solution (BKS):} \( x^* \in \arg\min_{x^i \in I} f(x) \)
9: end while
10: Return \( x^* = x^o \)

as Latin Hypercube Design (LHD), Sobol sequence \( [22, 23] \), or Orthogonal Array (OA) \( [24] \). Each input point \( x^i \) is evaluated to obtain a function value \( f(x^i) \) in Step 2. Next, the algorithm fits a surrogate model to predict the output of the system using the already evaluated input points, Step 4. The surrogate fitting can be performed in multiple ways, using an interpolating or a non-interpolating surrogate model. Step 5 determines, or samples, promising new input points using exploration, exploitation, or both. The exploration-exploitation trade-off involves identifying points that are in unexplored regions of the input space and points with minimum predicted output values. The sampled input points are evaluated in the black-box system in Step 6. A solution with the current best objective value, referred to as a best known solution (BKS) is determined in Step 8. Surrogate optimization repeats these steps until some termination criteria are met. Finally, it returns a BKS. In this research, we first propose a modified version of MARS to be used in Step 4 of Algorithm 1 and then will investigate the impact of different non-/interpolating surrogate models in \( \S \) 4. We also propose a replication strategy for the cases where the uncertainty is present and will evaluate the performance of different surrogates within the proposed framework.

2.2 Interpolating vs. Non-interpolating Surrogate Models

Choosing an appropriate surrogate model, \( \hat{f} \) (Step 4), is extremely dependent on the characteristics of the black-box function. Surrogate models are classified as interpolating (e.g., often GP \( [9] \), RBF \( [10] \)) or non-interpolating (e.g., polynomial regression models \( [25] \), MARS \( [12] \)). Interpolating models are the most common surrogates used in the surrogate optimization literature. They predict deterministic function values accurately by traversing the output of the input points. By contrast, non-interpolating models smoothly approximate the output under uncertainty \( [26] \). Unlike interpolating surrogate models, non-interpolating surrogate models do not necessarily traverse all the input points to capture the exact behavior of the function. Therefore, in highly uncertain, or noisy, systems, non-interpolating models are preferred to avoid oscillations caused by interpolation \( [27] \). For example, we show an interpolating RBF model in Figure 4(b) and a non-interpolating MARS model in Figure 4(c); both of these models approximate a simple \( \sin(x) \) function under uncertainty. Observe that to capture a given input with two or more different outputs, the interpolation method needs to traverse the points with an extremely large slope. Consequently, interpolation-based surrogate models result in highly fluctuating approximations for the output of a black-box function with uncertainty. In \( \S \) 4, we demonstrate the performance of
2.3 Radial Basis Function (RBF)

Radial Basis Function (RBF) is one of the most common interpolating surrogate models [10]. Assuming \( N \) distinct already evaluated input points, \( x^1, x^2, \ldots, x^N \in \mathbb{R}^d \), the RBF interpolant is of the form, \( \hat{f}(x) = \sum_{i=1}^{N} \lambda^i B(||x-x^i||) + p(x), \forall x \in \mathbb{R}^d \), where the coefficients \( \lambda^i, \forall i = 1, \ldots, N \), are real numbers, \( B(x) \) is a basis function, and \( p(x) \) is a low degree polynomial, which is added to avoid singularity [28].

Although there are multiple forms of the basis function \( B(x) \), we use Multiquadric (MQ) RBF models \(^1\) with the basis function of the form \( B(r) = \sqrt{r^2 + \omega^2} \), where \( \omega = ||.|| \) refers to the Euclidean \( L_2 \) norm, and \( \omega \) is a constant called the shape parameter. A larger shape parameter corresponds to a flatter basis function. In this research, the shape parameter is selected using cross-validation.

2.4 Non-interpolating Radial Basis Function

Jakobsson et al. [29] proposed a modified version of RBF as a non-interpolating surrogate to handle noisy objective functions, \( \tilde{f}(x) = f(x) + \varepsilon \), and avoid oscillating interpolation. The proposed approximation allows the model to deviate from the data points. Specifically, the non-interpolating RBF is determined by minimizing a weighted objective that considers both the standard interpolating RBF objective, or surrogate norm, and the sum of squared errors, or deviation. Let \( \bar{f} \) be the interpolating RBF function with coefficients \( \lambda^i \). The squared semi norm is defined as \( |S|^2 = \sum_{i=1}^{N} \lambda^i \bar{f}(x^i) \). The new optimization problem finds the coefficients \( \lambda^i \) that minimize \( \eta|S|^2 + (1 - \eta)||e||^2 \) subject to \( e_i = \bar{f}(x^i) - \tilde{f}(x^i), \forall i = 1, \ldots, N \), where \( ||e|| \) denotes the \( L_2 \) norm of \( (e_1, \ldots, e_N) \). The trade-off between the surrogate norm and deviation is controlled by \( \eta \), a hyperparameter that can be calibrated via cross-validation. Note that the proposed model cannot handle replicated data points and requires taking an average response if there are any.

2.5 Gaussian Processes for Noisy Observations

Gaussian processes, also known as Kriging, is one of the most common interpolating surrogate models that has been used in the BBO literature. A Gaussian process is a probabilistic model based on Gaussian distributions that are defined over a function \( f \) with a mean function \( \mu \) at each \( x^i \) and a covariance

---

\(^1\) MQ had the highest performance among different RBF basis function types in our preliminary analysis.
function $\Sigma$ at each pair of points $x^i$ and $x^j$. Let $X_N = [x^1, \ldots, x^N]$ represent the matrix of evaluated input data points, let $f$ be a noisy function, and let $y_N = [f(x^1), \ldots, f(x^N)]$ be the sampled output. The posterior distribution of $f$ (the true function being approximated) at any point $x$ is given as $f \sim \mathcal{GP}(\mu, \sigma^2_f)$, where $\mu(x) = \Sigma(X_N, x)^T[\Sigma(X_N, X_N) + \sigma^2_f I_N]^{-1}y_N$, and $\sigma^2_f(x) = \Sigma(x, x) - \Sigma(X_N, x)^T[\Sigma(X_N, X_N) + \sigma^2_f I_N]^{-1}\Sigma(X_N, x)$. $\Sigma(x, x)$ represents a kernel function over the input space, $\Sigma(X_N, X_N)$ is an $N \times N$ matrix of kernel values calculated on the input data, $\Sigma(X_N, x)$ is a $N \times 1$ vector of kernel values, and $\sigma_f$ is the standard deviation of $\varepsilon$. There are different possibilities for the kernel function, such as exponential, squared exponential, and Matern class of covariance functions [30]. The most popular kernel is the squared exponential kernel $\Sigma(x, x') = \sigma^2_f \exp(-0.5|\|x - x'|^2/\sigma^2_m)$, where $\sigma_m$ represents the length scale for predictors. The kernel hyperparameters can also be calibrated using cross-validation.

2.6 Multivariate Adaptive Regression Splines

Multivariate Adaptive Regression Splines (MARS) was introduced by Friedman [12]. MARS is a non-parametric non-interpolating surrogate model. The structure of the MARS model is based on basis functions. The MARS algorithm utilizes these basis functions to construct a piecewise continuous function in the following form:

$$\hat{f}(x, \beta) = \beta_0 + \sum_{m=1}^{M_{\text{max}}} \beta_m B_m(x).$$

(3)

where $x$ is a $d$-dimensional vector of input variables, $\beta_0$ is an intercept coefficient, $M_{\text{max}}$ is the maximum number of linearly independent basis functions, $\beta_m$ is the coefficient for the $m$th basis function, and $B_m(x)$ is a basis function that is either univariate or multivariate with interaction terms. These interaction terms have the following form:

$$B_m(x) = \prod_{l=1}^{L_m} [s_{ml}(x_{j(m,l)} - t_{ml})]^+.$$

(4)

where $L_m$ is the number of interaction terms in the $m$th basis function, $x_{j(m,l)}$ is the $j$th input variable of the $l$th truncated linear function in the $m$th basis function, and $t_{ml}$ is a knot location where the MARS function bends. The constant $s_{ml}$ is the direction of the truncated linear basis function and is either +1 or -1.

Within the context of modeling complex systems, MARS has two significant benefits. First, it can mitigate uncertainty as discussed in §2.2. Second, MARS is intended to be parsimonious and is able to screen unimportant input variables, which frequently occur in real-world complex systems [31]. The MARS algorithm includes a forward selection and a backward elimination procedure. In forward selection, MARS adds the basis functions in pairs for different dimensions, which gives the maximum reduction in the sum of squared error. The process of adding continues until it reaches the maximum number of basis functions, $M_{\text{max}}$. By removing the least effective term at each step, the backward elimination process avoids overfitting. MARS has an embedded dimension reduction technique, which is very useful for BBO where there is no previous understanding of the input
variables and their impact on the output. As a consequence, the final MARS model includes only important input variables.

Nonetheless, the MARS algorithm was developed for approximation purposes but not for optimization. Specifically, in the traditional version of MARS described in Friedman [12], each input point in the training set can be an eligible knot location. As the size of the training set increases, the number of eligible knot locations increases. MARS tends to interpolate the input points and loses its flexibility as the number of eligible knots increases. Interpolating a set of limited input points early leads to a false assessment of function behavior and multiple local optima for highly noisy functions, as described in §2.2 and in Koc and Iyigun [32]. This may cause difficulties for surrogate optimization. In addition, when a large number of eligible knots are selected, multicollinearity can occur between basis functions with knots that are close to each other.

Eligible knot selection techniques are developed to mitigate local variance and multicollinearity issues [32]. One of the most common techniques selects evenly spaced knot locations within the range of the input points [33,34,35,36,37]. Friedman [12] proposes a minimum span (MinSpan) approach to minimize the local variability. In the MinSpan approach, for each independent variable, a local search around its current knot location is designed to reduce the number of eligible knot locations. Miyata and Shen [38] presents a simulated annealing approach to choose eligible knot locations. Koc and Iyigun [32] develops a mapping strategy by transforming the original input points into a network of nodes through a nonlinear mapping. The nodes in the mapped network act as a reference for choosing the new eligible knot locations. Traditional knot selection approaches are mainly designed to fit the entire approximation over the whole space, which is different from what is best for surrogate optimization. Moreover, Müller and Shoemaker [17] conducted a comprehensive study on the influence of ensemble surrogate models and sampling strategies, and they showed that using MARS as a surrogate for optimization was inferior to using both RBF and GP. These facts motivate modifying MARS to be more effective for surrogate optimization. To the best of our knowledge, there exists no other method for MARS in surrogate optimization. In this work, we propose a partitioning-based approach using classification and regression trees to tailor MARS for surrogate optimization.

2.7 Classification and Regression Trees

Classification and Regression Trees (CART) use a non-parametric decision tree as a nonlinear predictive modeling method [11]. The CART algorithm recursively partitions the input variable space into two smaller sub-regions. The goal is to find partitions that minimize the sum of squared error, SSE, of the output in the resulting sub-regions. Greedy recursive binary splitting approach is performed to identify an optimal splitting variable and an optimal cutpoint that leads to the greatest possible reduction in SSE. The splitting process continues until a stopping criterion is reached; for instance, CART may continue until each node has fewer than some minimum number of observations, or the depth of the tree reaches a maximum. The final partitions after the CART algorithm stops are known as leaves or terminal nodes. As a result of the binary recursive splitting approach, the input points should have similar function values within one terminal node, and there should be a significant change in function values between different partitions. The predicted output is the average output of the input points within each terminal
node. In §3.1, we elaborate on our proposed eligible knot location approach for TK-MARS using CART.

2.8 Exploration-Exploitation Pareto Approach
As we discussed in Algorithm 1, sampling input points in Step 5 is critically important to find promising points to evaluate in the black-box function and avoid unnecessary expensive evaluations. In this research, we apply an Exploration-Exploitation Pareto Approach (EEPA) [39, 40, 41, 42, 43] for sampling in Step 5. Let \( P \subset R \), where \( P \) is a set of new candidate points, and \( R \) is a fixed pool of randomly generated candidate points. To determine a set of new input points from \( R \) to be evaluated, EEPA creates a Pareto frontier on the predicted function value using the surrogate model, as one dimension, and the distance of the candidate points from the already evaluated input points, as the second dimension. The first dimension exploits near-optimal areas, and the second dimension explores undiscovered regions of the input space. In particular, for each input point \( x \in R \), the exploration metric is \( \delta(x) = \min_{\tilde{x} \in I} \| x - \tilde{x} \| \), and the exploitation metric is \( \hat{f}(x) \). The first metric should be maximized, while the second should be minimized. The non-dominated Pareto set is given by \( F = \{ x \in R | \exists \tilde{x} \in R, \hat{f}(\tilde{x}) \leq \hat{f}(x), \delta(\tilde{x}) \geq \delta(x) \} \).

There might be candidate points on the Pareto set that are close to each other. To find the final set of candidate points from the Pareto set, EEPA first selects the point with the minimum predicted function value, then it applies a maximin function on the distance of the Pareto set points from the evaluated data points, to maximize the exploration further. Dickson [39] shows that EEPA outperforms pure exploration and exploitation methods. For more details, we refer to the pseudocode of EEPA in Appendix A.

2.9 Related Literature
Various surrogate models have been used in the BBO literature. A few of them employ non-interpolating models, such as MARS [39, 44, 45] and polynomial regression [46]. Despite the aforementioned shortcomings of interpolation methods, such as Kriging [15, 18, 42, 47, 48, 49] and RBF [43, 50, 51, 52, 53, 54, 55, 56], their use within BBO is far more prevalent.

In addition, these methods consider every variable as an important input to the output, which is a weakness in the surrogate optimization literature. A regularized RBF has been developed in the PYSOT toolbox [57] to ensure a well-conditioned system for RBF approximation. However, its ability to screen important variables in surrogate optimization has not been discussed in the literature. By contrast, Crino and Brown [45] demonstrate that MARS is capable of screening and reducing input variables using the parsimonious nature of MARS.

In some expensive computer simulations, there is inherent noise associated with the system. Most of the existing methods cannot handle uncertainty [46, 58, 59, 60]. While the majority of surrogate optimization literature ignores uncertainty, some research has taken it into account [18, 29, 41, 43, 49, 61]. Huang et al. [49] develops a Kriging-based surrogate optimization approach and applies it to low-dimensional test problems that include a low-level of noise. The computational effort of fitting Kriging increases for higher-dimensional problems. The proposed method for highly fluctuating functions under higher noise levels requires further investigation. Picheny, Wagner, and Ginsbourger [62] adds Gaussian noise with a fixed independent variance to the output of low-dimensional optimization test problems. The
results show the relative poor modeling performance of Kriging. A large part of the variability that cannot be explained by the model is due to the observational noise during optimization. Jakobsson et al. [29] and Picheny et al. [18] apply RBF and Kriging based surrogate optimization methods on low-dimensional test problems under low levels of noise. Costas et al. [44] shows MARS is preferable in real-world applications due to slope discontinuities and uncertainties, even though they do not study the effect of noise.

A few other studies in surrogate optimization concentrate on sampling. The main approach to select a new candidate point to sample is to solve an optimization sub-problem that is subject to exploration and exploitation constraints [16, 46, 63, 64, 65]. In these researches, a weighted score of response surface predictions and a distance metric is minimized to choose the next sample point. Müller et al. [56, 66] applied a stochastic sampling approach by perturbing the variables of a BKS with a perturbation probability distribution. The Pareto-based candidate point sampling approach has recently begun [39, 40, 41, 42, 43].

2.10 Contributions

In this research, we propose a new flexible, parsimonious, and non-interpolating surrogate model called TK-MARS, which can identify important input variables. We specifically design TK-MARS for surrogate optimization of black-box functions using a partitioning technique. Besides, we propose a smart replication strategy to mitigate the uncertainty associated with the black-box output. To evaluate the performance of surrogate optimization under uncertainty, we develop a new metric, called the Maximal True Function Area Under the Curve. We demonstrate the effectiveness of the proposed TK-MARS model and the smart replication approach using various complex global optimization test functions.

3 Technical Description

In this section, we describe the aforementioned contributions to surrogate optimization.

3.1 Tree-Knot Multivariate Adaptive Regression Splines

As we have discussed so far, in practice, there are unimportant input variables that are unknown a priori in black-box systems. Consequently, we would like to use MARS as a surrogate model in Step 4 of Algorithm 1 since it can identify important input variables and screen unimportant ones. However, the original version of MARS is not customized for optimization. In this section, we develop Tree-Knot MARS (TK-MARS), a new version of MARS that is more efficient for surrogate optimization. More precisely, TK-MARS has fewer eligible knot locations but ones that are more promising for optimization. Moreover, developing a MARS model that uses fewer eligible knot locations avoids interpolation causing oscillations for highly noisy systems, as discussed in [22].

The motivation behind the new approach is to provide eligible knots around potential optimum input points. The proposed TK-MARS uses CART as a partitioning technique to capture the function structure and identify potentially near-optimal knot locations in each partition.
Before giving a detailed description of TK-MARS, consider the following high-level example with a data set generated from the function $f(x) = \sin(x) + \epsilon$, where $\epsilon$ is a Gaussian noise term with a mean of 0, as in Figure 2(a). Consequently, $E[f(x)] = \sin(x)$. TK-MARS first uses CART to split the input points into four partitions, as in Figure 2(b). The solid vertical lines in Figure 2(b) represent the boundaries of the partitions, and the horizontal lines show the predicted average function value in each partition. Note that CART partitions the data based on the split where there is a significant change in the underlying function $f$, i.e., points with a similar objective value (closer to a peak or closer to a valley) fall into a single partition. As a result, the peaks and valleys of $f$ occur near the centers of the two middle partitions in the example as in Figure 2. As we discussed in §2.7, the final partitions obtained based on the CART binary recursive splitting algorithm using a stopping criterion are terminal nodes. Next, TK-MARS identifies the centroid of each terminal node and selects the closest input points to the centroids as eligible knot locations. The selected knot locations are represented by the dashed vertical lines as in Figure 2(c). Since these centroids tend to be close to the peaks and valleys where optima lie, selecting points near them as eligible knot locations facilitates optimization. Given the new set of eligible knot locations, TK-MARS approximates $f$, as $\hat{f}$, which is represented by the dotted function in Figure 2(d). The selected knot locations are still represented by the dashed vertical lines in Figure 2(d). The surrogate model $\hat{f}$ bends at near-optimal locations.

Specifically, let $I$ be the set of input points, let $x_j$ be the input variable in dimension $j, j = 1, \ldots, d$, and let $V$ be the set of terminal nodes from CART. For each terminal node $v \in V$, let $K_v \subset \{x^i | i = 1, \ldots, N\}$ be the set of observations in terminal node $v$. Depending on the number of terminal nodes $|V|$, the $N$ number of points in $I$ is partitioned into $|V|$ subsets. Let $c^j_v$ be the $j$th dimension of the centroid in terminal node $v$. Let $x^k_{jv}$ be the $j$th dimension of the $k$th observation in terminal $v$. Let $t^j_v$ be the index of the nearest input point to the centroid in terminal node $v$ for dimension $j$ and $t$ be the set of eligible knot locations, $\mathcal{T} = \{t^j_v, \forall v \in V, j = 1, \ldots, d\}$. Algorithm 2 presents the TK-MARS framework. As shown in Algorithm 2, CART partitions the data set based upon the function structure. TK-MARS then calculates the centroid of each terminal node $v \in V$ by Equation 6.
Algorithm 2 TK-MARS

input: \((I, F)\)

1: Fit CART on \((I, F)\)
2: \(t = \emptyset\)
3: for \(v \in V\) do
4: for \(j = 1\) to \(d\) do
5: \(c^v_j = \frac{1}{|K_v|} \sum_{k=1}^{|K_v|} x^j_k\)
6: \(t^v_j \in \arg\min_{k=1, \ldots, |K_v|} |x^j_k - c^v_j|\)
7: \(T = T \cup t^v_j\)
8: end for
9: end for
10: \(\hat{f}(x) = MARS((I, F), T)\)
11: return \(\hat{f}(x)\)

\[ c^v_j = \frac{1}{|K_v|} \sum_{k=1}^{|K_v|} x^j_k \quad \forall v \in V, j = 1, \ldots, d. \] (5)

MARS considers a set of univariate knots in each dimension, so TK-MARS determines the input points near the centroids in each dimension \(j = 1, \ldots, d\) of each terminal node \(v \in V\). Specifically, Equation (6) describes the index \(t^v_j\) of the input variable that is closest to the centroid of each terminal node \(v \in V\) in each dimension \(j = 1, \ldots, d\).

\[ t^v_j \in \arg\min_{k=1, \ldots, |K_v|} |x^j_k - c^v_j| \quad \forall v \in V, j = 1, \ldots, d. \] (6)

When there are ties for the nearest input points, \(t^v_j\) is the smallest index. Finally, we fit a MARS model \(\hat{f}\) using the previously evaluated data set \((I, F)\) with the potential knot locations \(T\) to predict the output of the black-box function. Note that TK-MARS uses \(|V|\) potential knots in each dimension \(j = 1, \ldots, d\).

In addition to assisting with developing the surrogate \(\hat{f}\) (Step 4 of Algorithm 1) using TK-MARS, the centroids \(C = \{c^v| v \in C\}\) can assist in the sampling of new candidate points \(P\) (Step 5). As discussed in \[2.8\] and Appendix \[A\] EEPA uses a large fixed pool, \(R\), from which it determines a Pareto set, \(F\), from which it samples a set of new candidate points, \(P\). However, EEPA and the quality of a best known solution found using EEPA depends on the fixed pool \(R\). \[20\] As shown in Fig. \[2\] good candidate solutions are often near centroids, consequently, in this research, we dynamically augment the pool \(R\) with the centroids; specifically \(R = R \cup C\).

3.2 Smart-Replication

We relax the assumption that the black-box function is deterministic in this section. Consequently, the black-box function output in Step 4 includes uncertainty. Specifically, \(\tilde{f}(x) = f(x) + \varepsilon\), where \(\tilde{f}(x)\) is the output of the black-box system, and \(\varepsilon\) is a random variable that follows an unknown probability distribution with a mean of \(0\) and a variance of \(\sigma^2\). This implies that each time a candidate point is evaluated a different output is obtained. Nonetheless, the goal is still to minimize the true objective function \(f(x)\), Equation (1).

Since there is uncertainty associated with the black-box output, a single evaluation of a candidate point might not be representative of the true output value.
Consequently, the deterministic approach, which we also refer to as No-Replication, may not be adequate to handle the uncertainty and, therefore, mislead the optimization process. Replicating the same input point multiple times and taking the mean of the outputs provides the most efficient estimator of the true objective value of \( f \) at the input point. Let \( x^i \) be the \( i \)th input point in the set of evaluated input points \( I \). Let \( \varepsilon^i_1, \varepsilon^i_2, \ldots, \varepsilon^i_r \) be a random sample of size \( r^i \) of the noise term \( \hat{f}(x^i) - f(x^i) \). We assume that the sample \( \varepsilon^i_1, \varepsilon^i_2, \ldots, \varepsilon^i_r \) is independent and identically distributed. Consequently, we can estimate \( f(x^i) \) as the sampled mean \( \bar{f}(x^i) \) using Equation (7):

\[
\bar{f}(x^i) = f(x^i) + \frac{\sum_{k=1}^{r^i} \varepsilon^i_k}{r^i}, \quad \forall x^i \in I.
\]

(7)

\( \bar{f}(x^i) \) in Equation (7) follows an unknown distribution, with \( f(x^i) \) mean and \( \frac{s^i}{\sqrt{r^i}} \) variance, based on the central limit theorem. If \( r^i = r, \forall x^i \in I \), we refer to this approach as Fixed-Replication, where we evaluate input points a fixed number of times, \( r \). For noisy black-box systems, we no longer have a best known solution because we do not have the true objective value of \( f(x) \). Consequently, we refer to a solution \( x \) with the best \( \bar{f}(x) \) as the best sampled mean solution (BSMS) in Step [8] of Algorithm [4].

To avoid unnecessary expensive evaluations in Step [8] of Algorithm [4] we propose a novel strategy called Smart-Replication, which replicates not all the candidate points but only the promising points for optimization based on confidence intervals around the sampled means. Let \( x^o \) be a BSMS, and let \( x^i \in P \) be a candidate point, and let \( r^o \) denote the number of replications for \( x^o \). Suppose \( r^o, r^i \geq 2 \). Then, we are \( 100(1 - \alpha)\% \) confident that \( f(x^i) \) is within the interval \( \bar{f}(x^i) \pm t_{\alpha/2, r^i-1} \frac{s^i}{\sqrt{r^i}} \), where \( s^i \) the sample variance of \( \bar{f}(x^i) \), which is given by the Equation (8), \( t_{\alpha/2} \) is the critical value of the student’s t-distribution, and \( \alpha \) is the significance level. We denote the sample variance of \( \bar{f}(x^o) \) by \( s^o \).

\[
s^i = \sqrt{\frac{1}{r^i - 1} \sum_{k=1}^{r^i} (f(x^i) + \varepsilon^i_k - \bar{f}(x^i))^2}.
\]

(8)

Let \( CI_{low}^o \) and \( CI_{up}^o \) be the lower and upper limits of the aforementioned confidence interval of \( f(x^i) \). Consequently, we can stop replicating a candidate point \( x^i \) if \( CI_{low}^o \geq CI_{up}^o \), i.e., if the lower bound of the objective value for point \( x^i \) is smaller than the upper bound of the best sampled mean solution, BSMS, there is no need to replicate \( x^i \).

Figure [3] presents an example of the Smart-Replication approach. The black bars show the confidence intervals for the true objective value, \( f \), at each input point in \( I \). The dashed horizontal line is at the \( CI_{up}^o \) value. The promising points are the candidate points that have confidence intervals that overlap with that of the current BSMS, i.e., \( CI_{low}^o < CI_{up}^o \). Consequently, the circled points are replicated further.

Since black-box function evaluations are expensive, for a given candidate point \( x^i \), Smart-Replication limits the number of replications \( r^i \) to a maximum number of replications denoted as \( r_{\text{max}} \). Specifically, Algorithm [3] shows the proposed surrogate optimization approach, which uses Smart-Replication (Algorithm [4]) for evaluation, Step [6].
Algorithm 3 Proposed Surrogate Optimization Approach
1: $I = \{x^1, \ldots, x^N\}$, a set of $N$ input points in $D$, selected with a DOE method
2: $F = \{ f(x^i) | x^i \in I \}$
3: while Termination criteria is not satisfied do
4: Surrogate: $\hat{f}(x) = \text{TK-MARS}(I, F)$
5: Sampling: $P = \text{EEPA}(R, I, \hat{f}(x))$
6: Evaluation: Smart-Replication($I, F, P$)
7: Best Sampled Mean Solution (BSMS): $x^o \in \text{argmin}_{x \in I} \bar{f}(x)$
8: end while
9: return $x^* = x^o$

Algorithm 4 Smart-Replication
input: ($I, F, P$)
1: $F_P = \{ \tilde{f}(x^i) | x^i \in P \}$
2: $I = I \cup P; F = F \cup F_P$
3: $CI_{low}^i = -\infty, CI_{up}^i = +\infty, r^i = 1, s^i = 1, \forall i \in I$
4: for $i = 1$ to $|I|$ do
5: while $CI_{low}^i < CI_{up}^i \& r^i \leq r_{\text{max}}$ do
6: Evaluation: $F = F \cup \{ \tilde{f}(x^i) \}$
7: $r^i = r^i + 1$
8: $\bar{f}(x^i) = f(x^i) + \sum_{k=1}^{r^i} \frac{\varepsilon_k^i}{r^i}$
9: $s^i = \sqrt{\frac{1}{r^i - 1} \sum_{k=1}^{r^i} (f(x^i) + \varepsilon_k^i - \bar{f}(x^i))^2}$
10: $CI_{low}^i = \bar{f}(x^i) - t_{\frac{\alpha}{2}} \frac{s^i}{\sqrt{r^i}}$
11: $CI_{up}^i = \bar{f}(x^i) + t_{\frac{\alpha}{2}} \frac{s^i}{\sqrt{r^i}}$
12: end while
13: $x^o \in \text{argmin}_{x \in I} \bar{f}(x)$
14: $CI_{up}^o = \bar{f}(x^o) + t_{\frac{\alpha}{2}} \frac{s^o}{\sqrt{r_o}}$
15: end for

Fig. 3: Confidence interval threshold mechanism of the Smart-Replication Approach

Smart-Replication is similar to the No-Replication approach when the uncertainty level is low and similar to the Fixed-Replication approach when the uncertainty level is high. Specifically, for a high noise level, the variance is larger, so more replications are required, and most candidate points are replicated up to the maximum number of replications $r_{\text{max}}$. Consequently, Smart-Replication adjusts its behaviour based on the uncertainty level of the system.
3.3 Performance Metrics for Surrogate Optimization

Since black-box function evaluations are expensive, the number of such evaluations performed before finding an optimal solution is a common metric to test a surrogate optimization algorithm. However, obtaining a global optimum cannot be guaranteed. A metric that can quantify the improvement in the true BSMS objective value over fewer black-box function evaluations is more appropriate in the context of BBO. Area Under the Curve has been used in black-box optimization to measure how quickly an algorithm improves solution quality compared to other algorithms [67]. AUC is also commonly used in other optimization domains [68] and in statistics to compare models with the two metrics sensitivity and specificity [69,70]. Specifically, consider an executed surrogate optimization algorithm, and let \( I \) be the total number of black-box function evaluations conducted during the execution. The Area Under the Curve (AUC) is given by definition 1.

**Definition 1 (Area Under the Curve – AUC)** Let \( x^{oi} \) be a BSMS found after \( i \) black-box function evaluations, and let \( f(x^{oi}) \) be the true objective value of the BSMS. Let \( f^{min} \) be the optimal objective value of the optimization problem (1)-(2), and let \( f^{max} = \max\{f(x^{oi}) | \forall i = 1, \ldots, I\} \). The normalized objective value of BSMS is:

\[
\hat{f}(x^{oi}) = \frac{f(x^{oi}) - f^{min}}{f^{max} - f^{min}}, \quad \forall i = 1, \ldots, I.
\]

The AUC is given by

\[
AUC = \sum_{i=1}^{I} \hat{f}(x^{oi}). \quad (9)
\]

AUC measures the performance of a surrogate optimization algorithm well in a deterministic environment with no noise (i.e., \( \sigma^2 = 0 \)), since the objective value of the BSMS monotonically decreases over each black-box function evaluation. However, this monotonicity may not hold for black-box systems with uncertainty, as the true objective value of the BSMS may oscillate. Even though oscillations in early iterations are tolerable, we would like stable behavior towards the end of the algorithm execution. Consequently, we propose a metric for black-box systems with uncertainty that penalizes instability by considering the maximum true objective value of all BSMS found at the current and subsequent black-box function evaluations of the algorithm. Specifically, we define the Maximal True Function Area Under the Curve (MTFAUC) in definition 2.

**Definition 2 (Maximal True Function Area Under the Curve – MTFAUC)** Let \( \hat{j}(i) \) be the index of the black-box function evaluation of the maximum true objective value among all BSMS found in function evaluations after function evaluation \( i - 1 \); that is,

\[
\hat{j}(i) \in \arg\max_{j=i, \ldots, I} \hat{f}(x^{oj}). \quad (10)
\]

The MTFAUC is given by

\[
MTFAUC = \sum_{i=1}^{I} \frac{\hat{f}(x^{\hat{j}(i)}) + \hat{f}(x^{\hat{j}(i-1)})}{2}. \quad (11)
\]
MTFAUC penalizes the instability of the true objective value of BSMS, by using \( j(i) \) to consider subsequent oscillations. Hence, MTFAUC increases if the true objective value oscillates. A surrogate optimization algorithm that is less sensitive to uncertainty has a lower MTFAUC value than one that is more sensitive. MTFAUC is equal to AUC in a deterministic environment.

Figure 4 shows the difference between AUC and MTFAUC. Figure 4 (a) demonstrates a deterministic test case in which the AUC curve is non-increasing. As we can observe in Figure 4 (b), the AUC curve of the true function is no longer non-increasing since there are uncertainties associated with the sampled function values. It is worth mentioning that the AUC in the stochastic case in (b) is 0.35, which is smaller than that in (a) with 0.37. However, we can observe that the final BSMS reported by (b) has a higher true objective value than solutions found earlier in the surrogate optimization process. The oscillations are caused by solutions with relatively good sampled mean objective value but not a good true objective value. Our proposed MTFAUC penalizes the unexpected oscillations caused by the noisy objective values and is a new non-increasing curve. Figure 4 (c) demonstrate the same test case in Figure 4 (b). While the AUC true function is calculated with the blue area under the curve, MTFAUC penalizes the area painted in red and blue. Note that the BSMS in the final steps is critical as it is the output of the surrogate optimization framework. A robust surrogate would have fewer and shorter oscillations in the final iterations, which results in a lower overall MTFAUC value. Note that this cannot be captured by mean and variance, as those are affected by oscillations at the beginning and end equally.

![MTFAUC vs. AUC](image)

Indeed the AUC and MTFAUC evaluation metrics are designed to evaluate the performance of different algorithms across many different types of cases. AUC and MTFAUC are designed for system testing purposes, and they are not intended to be used in a real-world engineering problem. The proposed measures are used after a full execution of an algorithm and are not involved as a convergence criterion inside the algorithm. For example, if we execute two different algorithms for the same number of iterations, then we can use MTFAUC to compare their performance. Even when the global minimum is unknown, in the deterministic case, AUC can be applied directly to evaluate the performance of algorithms with the true objective minimum substituted by the objective value of the minimum BKS.
of all the algorithm executions being evaluated. Under uncertainty, we would collect sampled response values for each data point over all the algorithm executions and apply MTFAUC using the mean sampled response values in place of the true function value.

4 Experimental Results

We evaluate the performance of our proposed approach on global optimization test functions that have proven to be challenging for BBO \cite{71, 72}. Table 2 describes the characteristics of the selected test functions, and $d$ denotes the number of variables. In this research, we first assume that the black-box function is deterministic, and later we relax this assumption in \S 4.2. We use MTFAUC throughout the experiments for performance evaluation. Recall MTFAUC is equal to AUC for deterministic cases.

We evaluate the performance of surrogate optimization using TK-MARS, RBF interpolant, non-interpolating RBF (nonRBF), and non-interpolating GP (nonGP) as surrogates. We fixed the budget to $B = 1000$ function evaluations. The benchmark test functions presented in Table 2 are used along with the cross-validated hyperparameters presented in Table 1 for different surrogates. For nonGP implementation, we use the GP package in Matlab.\footnote{https://www.mathworks.com/help/stats/gaussian-process-regression-models.html} We use Bayesian hyperparameter optimization (which is an option in the Gaussian Processes package in Matlab) to tune the GP hyperparameters using cross-validation. We perform this hyperparameter tuning once on the first iteration of our surrogate optimization algorithm, and once after the algorithm completes 500 function evaluations. In other iterations, we fit the nonGP model using the same set of hyperparameters as in the previous iteration. This is a common strategy in Bayesian optimization literature \cite{73, 74, 75}. Moreover, in other preliminary experiments, we saw no significant difference between tuning the GP hyperparameters on every iteration and tuning them on just these two iterations. We refer to the surrogate optimization with different surrogates as the name of surrogates in the plots to save space. For example, we mark the surrogate optimization algorithm with RBF as “RBF”.

Table 1: Hyperparameter settings of different surrogates

| Model       | Hyperparameters                                      |
|-------------|------------------------------------------------------|
| TK-MARS:    | CART: minsplit=20 maxdepth=30                        |
|             | MARS: $M_{\text{max}} = \left\lfloor \frac{2n+3}{n+1} \right\rfloor$ $L_m = 1$ |
| RBF:        | kernel = MQ $\omega = 2$ $\eta = 0.0001$            |
| nonRB:      | kernel = MQ                                           |
| nonGP:      | Bayesian $\sigma, \sigma_f \in \mathbb{R}$           |

2. \footnote{https://www.mathworks.com/help/stats/gaussian-process-regression-models.html}
Table 2: Test Functions Definition

| Function | Formulation | Range       | Global Min |
|----------|-------------|-------------|------------|
| Rosenbrock | $f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | [-5,10] | $f(x^*) = 0$, $x^* = (1, \ldots, 1)$ |
| Rastrigin | $f(x) = 10d + \sum_{i=1}^{d-1} [1 + 10 \cos(2\pi x_i)]$ | [-5.12,5.12] | $f(x^*) = 0$, $x^* = (0, \ldots, 0)$ |
| Levy | $f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} [w_i - 1] [1 + 10 \sin^2(\pi w_i + 1)]$ where, $w_i = 1 + \frac{x_i - 1}{4}$ | [-10,10] | $f(x^*) = 0$, $x^* = (1, \ldots, 1)$ |
| Ackley | $f(x_0 \cdots x_n) = -20 \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x_i)) + 20 + e$ | [-32.768, 32.768] | $f(x^*) = 0$, $x^* = (1, \ldots, 1)$ |
| Zakharov | $f(x) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5 x_i)^2$ | [-5,10] | $f(x^*) = 0$, $x^* = (0, \ldots, 0)$ |

4.1 Comparison of MARS and TK-MARS

First, we evaluate the proposed TK-MARS versus original MARS with evenly spaced knot locations in the context of surrogate optimization. The number of eligible knot locations, $T$, has to be preset for original MARS. We examine a surrogate optimization algorithm that uses original MARS with $T = 10$, $T = 20$, and $T = 50$. Since TK-MARS sets $T$ based on the number of terminal nodes, $|V|$, to ensure a fair comparison between the two approaches, we consider $T = |V|$ for original MARS, as well. It should be noted that as the number of knots increases, the computation time of MARS increases, even though the accuracy may increase. We initialize $I$ with $N = 31$ points selected with a LHD for $d = 30$ independent input variables. Figures 5(a)-(e) show the average true objective value of the BSMS versus the number of black-box function evaluations for each of the test functions listed in Table 2. The results are presented for each surrogate optimization algorithm using the aforementioned versions of MARS as well as our proposed TK-MARS over 30 different executions. Note that using TK-MARS as a surrogate within a surrogate optimization algorithm outperforms that using the original MARS with different number of knots for the various test functions. The green triangles corresponds to TK-MARS, and the faster convergence leads to smaller AUC.

Next, we compare the performance of our surrogate optimization using TK-MARS versus deterministic RBF (“RBF”), non-interpolating RBF (“nonRBF”), and non-interpolating GP for noisy observations (“nonGP”) on a new class of test functions that include unimportant input variables. Specifically, the new test functions only consider a fraction of the input variables, $fiv$ for the function evaluation. In particular, we select the first $fiv\%$ of the variables for the evaluation. For example, if $fiv = 0.5$ and $d = 30$ only $\{x_1, \ldots, x_{15}\}$ are considered for the function evaluation.

Figure 6 shows the average MTFAUC of 30 different executions for each test function with four different levels of $fiv$, 0.25, 0.5, 0.75, and 1.0. The error bar shows the standard deviation of the average MTFAUC of 30 different executions.

Note that MTFAUC is higher when fewer variables are important, suggesting that the surrogates struggle to determine which input variables are important.
Fig. 5: TK-MARS vs. MARS with different number of eligible knot locations when few of them are. The benefit of using TK-MARS is notable for the Rosenbrock, Rastrigin, and Levy functions. This difference is likely because MARS is parsimonious, while others use all the input variables in their functional form. We do not observe the same pattern for Zakharov function since the plate-shaped valley of Zakharov function makes the surrogate fitting extremely difficult for metamodels \cite{76}. For the Ackley function, the difference is not visible in the plot because of the higher range of MTFAUC for nonRBF and nonGP, however, there is 2\% difference between the performance of TK-MARS compared with RBF when $f_{iv} = 0.25$.

To demonstrate the performance of TK-MARS in correctly detecting the important variables, Figure\[\text{7a}\] presents the variables selected by TK-MARS for Rosenbrock function, at the final iteration of surrogate optimization. The bars show the percentage of a variable being selected in 10 different executions (different pools). The green horizontal bar indicates the range of the important variables considered for the function evaluation, and the red horizontal bar shows the unimportant range of the variables. As we can observe, in Figure\[\text{7a}\], TK-MARS correctly selects all the important variables (except 30) in all 10 executions. In Figure\[\text{7b}\], TK-MARS in the majority of cases selects all the important variables, which are $x_1, \ldots, x_{22}$, and do not prefer the unimportant variables $x_{23}, \ldots, x_{30}$. One can confirm the same pattern in Figure\[\text{7c}\] and Figure\[\text{7d}\]. Although TK-MARS selects few unimportant variables, because the variable selection of TK-MARS also depends on the input-output of the training dataset, TK-MARS involves the significant variables for its basis function construction.
Fig. 6: TK-MARS vs. other surrogate models across different $f_{iv}$

Fig. 7: Percentage of variables selected by TK-MARS across 10 different executions

While the performance of surrogate optimization is usually based upon the number of expensive function evaluations, it is worth comparing the non-evaluation time of the algorithm considering different surrogate models. In our experiments, for Rosenbrock function with $d = 30$, RBF requires an average of 0.06 seconds,
Table 3: Replication parameters and levels

| Problem Parameters | Levels |
|-------------------|--------|
| Test function     | Rosenbrock, Rastrigin, Levy, Ackley, Zakharov |
| Dimension \((d)\) | 30     |
| Fraction of important variables \((fiv)\) | 0.50 |
| Noise level \((np)\) | 0, 0.05, 0.10, 0.25 |

| Algorithm Parameters | Levels |
|----------------------|--------|
| Initial pool size \((N)\) | \(d + 1\) |
| DOE method in Step 1 of Algorithm 1 | LHD |
| EEPA number of candidate points \((N^*\) in Appendix A) | 3 |
| Replication type | No-Replication, Fixed-Replication, Smart-Replication |
| Replication number \((r\) or \(r_{\text{max}}\)) | 5, 10 |
| Surrogate Model | RBF, TK-MARS |

original MARS and TK-MARS use 0.04 seconds, non-interpolating RBF needs 0.07 seconds, and non-interpolating GP uses 0.22 seconds. Although TK-MARS requires negligible time more than original MARS for CART to select the candidate knot locations, this step is completed in 0.006 seconds. Moreover, the total computation time of the No-Replication algorithm with using TK-MARS, MARS with \(k = 20\), and \(k = 50\) on Rosenbrock function is 1148 secs, 927 secs, and 1346 secs, respectively. Consequently, the difference in the computation times of TK-MARS and original MARS is unlikely to be a concern to a user given how well TK-MARS outperforms MARS in terms of AUC as shown in Figure 5. The experiments were performed on a 2.3 GHz Core i9 machine with 16 GB DDR4 memory.

4.2 Optimization Under Uncertainty

To evaluate the performance of the proposed approach under uncertainty, we assume that the unknown noise associated with the black-box outcome follows a Gaussian distribution and simulate the noise accordingly. As a result, we add a Gaussian noise to the function values, i.e., \(\tilde{f}(x) = f(x) + \varepsilon\), where \(\varepsilon \sim N(0, np * \sigma_0)\), \(np \in (0, 1)\) is the noise percentage level, and \(\sigma_0\) is given by \(\sigma_0 = \max\{f(x^1), \ldots, f(x^N)\} - \min\{f(x^1), \ldots, f(x^N)\}\).

In a preliminary analysis, we designed a set of experiments using an Orthogonal Array, \([77, 78]\), on the full list of algorithm and problem parameters (Table 5 in the Appendix) and performed an ANOVA (Table 6 in Appendix B) to focus on the significant parameters, which are given in Table 3. The Replication number is \(r\) when using Fixed-Replication and \(r_{\text{max}}\) for Smart-Replication. We refer to No-Replication as \(\text{Norep}\), Fixed-Replication with \(r\) replications as \(\text{Fixedrep}_r\), and Smart-Replication with \(r_{\text{max}}\) as \(\text{Smartrep}_{r_{\text{max}}}\), throughout the experiments.

Figures 9(a)-(t) show the average performance of No-Replication, Fixed-Replication, and Smart-Replication for 30 executions on the test functions in Table 2. Each row represents a test function, and each column corresponds to a surrogate model. Tables 7, 8, and 9 in Appendix B, show corresponding MTFAUC values. The results indicate that when no noise is present, interpolating is a good choice for surrogate fitting. When the objective function is noisy, interpolating may lead to a poor approximation and misleads the optimization. Note that No-Replication outperforms the replication approaches in some cases when we use non-interpolating surrogates. That is due to the fact that non-interpolating surrogates minimize
the average error across all data points, which collectively reduces the impact of the noise. However, we realized non-interpolating models can capture the noise without replication when the objective function is not highly complex (highly oscillated). In such situations, where “predictability” is high, the model can use the value of other points to cancel out the noise for a specific point. To further clarify this, let us consider the toy example in Figure 8. In Figure 8 (a), the underlying objective function is linear. We have highlighted a red star point, for which the function value is highly affected by the noise (dashed line). One can observe that minimizing the error for all the points, the non-interpolating model (linear regression) could cancel out the effect of the red star point.

This however, is not the case for “unpredictable” and highly-oscillating functions. That simply is because, due to the high oscillations, other points (slightly far from the star noisy point) cannot be used for estimating the correct value of the noisy point. We highlighted this in the toy example shown in Figure 8 (b) (Ackley function estimated by (non-interpolating) quadratic model). As one can confirm, assuming that the function had a peak (valley) at the noisy point, the existing data points cannot predict the depth of the peak. In such situations, replications are needed to cancel out the impact of noise. In Figure 8 (c), we added the replications for the star noisy point, the points shown as triangles. We can observe the non-interpolating model has a better estimation of the true function at the red star point, and was more effective in mitigating the noise effect of that point.

In summary, while non-interpolating methods may perform well for non-complex noisy environments, they fail to work well for complex noisy functions without replications. We observed a similar behavior in our experiments where no-replication failed to work well for complex functions, such as Zakharov, Ackley, and Rastrigin.

For a deeper evaluation of the performance of No-Replication and replication strategies using different surrogates, the robustness and the quality of BSMS must be assessed. Figure 10 show box-plots of the MTFAUC for 30 executions at different noise levels using different approaches on the Rosenbrock test function. We observe that the interquartile range (the difference between the 25th and 75th percentiles, i.e., the height of the boxes) for Smart-Replication is smaller especially when the noise level is high. This suggests that Smart-Replication is more robust. Although Fixedrep, 10 has also short interquartile range in higher levels of noise in some cases, the overall average MTFAUC is larger in lower noise levels than that of Smart-Replication. This is because the Fixed-Replication strategy makes unnecessary functional evaluations. Fixedrep, 10 outperforms Smartrep when we use nonGP as a surrogate. This suggests that nonGP needs more replications to
Fig. 9: Average MTFAUC across different noise levels
handle noise, and we perhaps need to increase $r_{\text{max}}$ of $\text{Smartrep}$ to observe its benefit. Note that as the noise level increases, $\text{Smartrep}$, 10 is more competitive with $\text{Norep}$ especially when we use the interpolating RBF model.

Figure 11 presents the box-plots of the true objective value of BSMS after 1000 black-box function evaluations for the same 30 executions on the Rosenbrock test function as those in Figure 10. We observe that $\text{Smartrep}$ is competitive with $\text{Norep}$ in the lower levels of noise after 1000 black-box function evaluations. $\text{Smartrep}$, 10 has the smallest interquartile range with the highest level of noise, indicating its robustness. The box-plots for the other test functions are provided in Appendix B.

It should be noted that while $\text{Fixedrep}$ and $\text{Norep}$ may have a competitive performance or outperformance compared to $\text{Smartrep}$ in some situations, the level of uncertainty of the black-box system is unknown apriori, $\text{Smartrep}$ may adapt to the unknown noise level to have the same performance of $\text{Norep}$ in the lower noise level and $\text{Fixedrep}$ in the higher noise level.
5 Conclusions

In this paper, we developed a partitioning-based MARS model called TK-MARS, which is specifically designed for surrogate optimization of black-box functions. Moreover, we designed a smart replication approach to mitigate the impact of uncertainty associated with the black-box output. We demonstrated the performance of the proposed approaches using complex global optimization test functions. The performance of a surrogate optimization algorithm when using TK-MARS was compared with that when using original MARS. We observed that TK-MARS outperforms original MARS in the context of surrogate optimization. We also showed that TK-MARS is capable of correctly detecting the important variables.

No-Replication outperforms other approaches overall in a deterministic environment, but in a noisy environment, Smart-Replication yields more robust results. The robustness and the quality of the final optimum solution found through Smart-Replication is competitive with that using no replications in environments with low levels of noise and using a fixed number of replications in highly noisy environ-
ments. Moreover, a surrogate optimization algorithm with an interpolating surrogate, such as RBF, needs more replication than the one with a non-interpolating surrogate, such as TK-MARS.

The results indicate that surrogate optimization using TK-MARS outperforms that of interpolating and non-interpolating RBF and non-interpolating GP in most of the cases, which are the prevalent surrogate models used in the literature. It is worth mentioning that non-interpolating RBF model proposed by Jakobsson in [29] is competitive with TK-MARS in a few instances. Non-interpolating surrogate models still need replications in the presence of noise when the underlying function is complex.

We consider using real datasets for evaluating our approaches as part of our future work. Our future work also includes investigating other feature selection techniques in the surrogate optimization framework to identify unimportant variables of a black-box function.

6 Acknowledgement

The work was supported by the National Science Foundation Award CMMI–1434401.

References

1. D. Wilson, S. Rodrigues, C. Segura, I. Loshchilov, F. Hutter, G. L. Buenfil, A. Kheiri, E. Keedwell, M. Ocampo-Pineda, E. Özcan, et al., “Evolutionary computation for wind farm layout optimization,” Renewable energy, vol. 126, pp. 681–691, 2018.
2. J. Su, J. Wu, P. Cheng, and J. Chen, “Autonomous vehicle control through the dynamics and controller learning,” IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 5650–5657, 2018.
3. A.-T. Nguyen, S. Reiter, and P. Rigo, “A review on simulation-based optimization methods applied to building performance analysis,” Applied Energy, vol. 113, pp. 1043–1058, 2014.
4. K. Hamza and K. Saitou, “Vehicle crashworthiness design via a surrogate model ensemble and a co-evolutionary genetic algorithm,” in ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 899–907, American Society of Mechanical Engineers, 2005.
5. E. Pyzer-Knapp, “Bayesian optimization for accelerated drug discovery,” IBM Journal of Research and Development, vol. 62, no. 6, pp. 2–1, 2018.
6. R.-R. Griffiths and J. M. Hernández-Lobato, “Constrained bayesian optimization for automatic chemical design,” arXiv preprint arXiv:1709.05501, 2017.
7. L. Farkas, D. Vandeputte, D. Vandepitte, and W. Deamet, “Fuzzy finite element analysis based on reanalysis technique,” Structural Safety, vol. 32, no. 6, pp. 442–448, 2010.
8. L. Gu, “A comparison of polynomial based regression models in vehicle safety analysis,” ASME Paper No. DAC-21063, 2001.
9. D. G. Krige, A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige. PhD thesis, University of the Witwatersrand, 1951.
10. M. J. Powell, “Radial basis functionn for multivariable interpolation: A review,” in IMA Conference on Algorithms for the Approximation of Functions ans Data, pp. 143–167, RMCS, 1985.
11. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and regression trees. CRC press, 1984.
12. J. H. Friedman, “Multivariate adaptive regression splines,” The annals of statistics, pp. 1–67, 1991.
13. M. Papadrakakis, N. D. Lagaros, and Y. Tsompanakis, “Structural optimization using evolution strategies and neural networks,” Computer methods in applied mechanics and engineering, vol. 156, no. 1-4, pp. 309–333, 1998.
14. S. M. Clarke, J. H. Griebsch, and T. W. Simpson, “Analysis of support vector regression for approximation of complex engineering analyses,” *Journal of mechanical design*, vol. 127, no. 6, pp. 1077–1087, 2005.
15. D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.
16. R. G. Regis and C. A. Shoemaker, “Constrained global optimization of expensive black box functions using radial basis functions,” *Journal of Global optimization*, vol. 31, no. 1, pp. 153–171, 2005.
17. J. Müller and C. A. Shoemaker, “Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems,” *Journal of Global Optimization*, vol. 60, no. 2, pp. 123–144, 2014.
18. V. Picheny, D. Ginsbourger, Y. Richet, and G. Caplin, “Quantile-based optimization of noisy computer experiments with tunable precision,” *Technometrics*, vol. 55, no. 1, pp. 2–13, 2013.
19. M. Feurer and F. Hutter, “Hyperparameter optimization,” in *Automated Machine Learning*, pp. 3–33, Springer, Cham, 2019.
20. V. Pilla, “Robust airline fleet assignment,” 2007.
21. M. C. Kennedy and A. O’Hagan, “Bayesian calibration of computer models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.
22. I. M. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.
23. I. M. Sobol’, “Uniformly distributed sequences with an additional uniform property,” *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 5, pp. 236–242, 1976.
24. A. S. Hedayat, N. J. A. Sloane, and J. Stufken, *Orthogonal arrays: theory and applications*, Springer Science & Business Media, 2012.
25. J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*, vol. 4. Irwin Chicago, 1996.
26. J. T. Hwang and J. R. Martins, “A fast-prediction surrogate model for large datasets,” *Aerospace Science and Technology*, vol. 75, pp. 74–87, 2018.
27. J. A. Caballero and I. E. Grossmann, “An algorithm for the use of surrogate models in modular flowsheet optimization,” *AIChE Journal*, vol. 54, no. 10, pp. 2633–2650, 2008.
28. G. B. Wright, “Radial basis function interpolation: numerical and analytical developments,” 2003.
29. S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski, “A method for simulation based optimization using radial basis functions,” *Optimization and Engineering*, vol. 11, no. 4, pp. 501–532, 2010.
30. C. E. Rasmussen and C. Williams, “Gaussian processes for machine learning, vol. 1,” 2006.
31. K. Craig, N. Stander, D. Dooge, and S. Varadappa, “Automotive crashworthiness design using response surface-based variable screening and optimization,” *Engineering Computations*, vol. 22, no. 1, pp. 38–61, 2005.
32. E. K. Koc and C. Iyigun, “Restructuring forward step of mars algorithm using a new knot selection procedure based on a mapping approach,” *Journal of Global Optimization*, vol. 60, no. 1, pp. 79–102, 2014.
33. V. C. Chen, D. Ruppert, and C. A. Shoemaker, “Applying experimental design and regression splines to high-dimensional continuous-state stochastic dynamic programming,” *Operations Research*, vol. 47, no. 1, pp. 38–53, 1999.
34. D. Martinez, *Variants of multivariate adaptive regression splines (mars): Convex vs. non-convex, piecewise-linear vs. smooth and sequential algorithms*. PhD thesis, Ph. D. thesis, The University of Texas at Arlington, 2013.
35. Y. Cao, H. Lin, T. Z. Wu, and Y. Yu, “Penalized spline estimation for functional coefficient regression models,” *Computational statistics & data analysis*, vol. 54, no. 4, pp. 891–905, 2010.
36. J. Z. Huang and H. Shen, “Functional coefficient regression models for non-linear time series: A polynomial spline approach,” *Scandinavian journal of statistics*, vol. 31, no. 4, pp. 515–534, 2004.
37. Q. Song and L. Yang, “Oracally efficient spline smoothing of nonlinear additive autoregression models with simultaneous confidence band,” *Journal of Multivariate Analysis*, vol. 101, no. 9, pp. 2008–2025, 2010.
38. S. Miyata and X. Shen, “Free-knot splines and adaptive knot selection,” *Journal of the Japan Statistical Society*, vol. 35, no. 2, pp. 303–324, 2005.
39. J. F. Dickson, *An exploration and exploitation pareto approach to surrogate optimization*, The University of Texas at Arlington, 2014.
40. B. Bischi, S. Wessing, N. Bauer, K. Friedrichs, and C. Weihs, “MoI-mbo: Multiobjective infill for parallel model-based optimization,” in *International Conference on Learning and Intelligent Optimization*, pp. 173–186, Springer, 2014.
41. T. Kritikakierne, “Global optimization of computationally expensive blackbox problems using radial basis functions,” 2014.
42. T. Krityakierne, T. Akhtar, and C. A. Shoemaker, “Sop: parallel surrogate global optimization with pareto center selection for computationally expensive single objective problems,” *Journal of Global Optimization*, vol. 66, no. 3, pp. 417–437, 2016.
43. D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, “Global optimization of stochastic black-box systems via sequential kriging meta-models,” *Journal of global optimization*, vol. 34, no. 3, pp. 441–466, 2006.
44. C. Liu, Z. Wan, Y. Liu, X. Li, and D. Liu, “Trust-region based adaptive radial basis function algorithm for global optimization of expensive constrained black-box problems,” *Applied Soft Computing*, vol. 105, p. 107235, 2021.
45. R. G. Regis, “Multi-objective constrained black-box optimization using radial basis function surrogates,” *Journal of computational science*, vol. 16, pp. 140–155, 2016.
46. R. G. Regis, “Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions,” *Computers & Operations Research*, vol. 38, no. 5, pp. 837–853, 2011.
47. R. G. Regis, “Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points,” *Engineering Optimization*, vol. 46, no. 2, pp. 218–243, 2014.
48. R. G. Regis, “Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.
49. J. Müller, “Miso: mixed-integer surrogate optimization framework,” *Optimization and Engineering*, vol. 17, no. 1, pp. 177–203, 2016.
50. D. Eriksson, D. Bündel, and C. A. Shoemaker, “pysot and poap: An event-driven asynchronous framework for surrogate optimization,” *arXiv preprint arXiv:1908.00420*, 2019.
51. E. Davis and M. Ierapetritou, “Adaptive optimisation of noisy black-box functions inherent in microscopic models,” *Computers & Chemical Engineering*, vol. 31, no. 5, pp. 466–476, 2007.
60. J.-B. Grill, M. Valko, and R. Munos, “Black-box optimization of noisy functions with unknown smoothness,” in Advances in Neural Information Processing Systems, pp. 667–675, 2015.
61. Z. Wang and M. Ierapetritou, “A novel surrogate-based optimization method for black-box simulation with heteroscedastic noise,” Industrial & Engineering Chemistry Research, vol. 56, no. 38, pp. 10720–10732, 2017.
62. V. Picheny, T. Wagner, and D. Ginsbourger, “A benchmark of kriging-based infill criteria for noisy optimization,” Structural and Multidisciplinary Optimization, vol. 48, no. 3, pp. 607–626, 2013.
63. Z. Li, Z. Dong, Z. Liang, and Z. Ding, “Surrogate-based distributed optimisation for expensive black-box functions,” Automatica, vol. 125, p. 109407, 2021.
64. J. Knowles, “Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” IEEE Transactions on Evolutionary Computation, vol. 10, no. 1, pp. 50–66, 2006.
65. J. Mühler and J. D. Woodbury, “Gosac: global optimization with surrogate approximation of constraints,” Journal of Global Optimization, vol. 69, no. 1, pp. 117–136, 2017.
66. J. Mühler, R. Paudel, C. Shoemaker, J. Woodbury, Y. Wang, and N. Mahowald, “Ch 4 parameter estimation in clm4. 5bgc using surrogate global optimization,” Geoscientific Model Development, vol. 8, no. 10, pp. 3285–3310, 2015.
67. I. Dewancker, M. McCourt, S. Clark, P. Hayes, A. Johnson, and G. Ke, “A stratified analysis of bayesian optimization methods,” arXiv preprint arXiv:1603.09441, 2016.
68. A. Memarian, J. M. Rosenberger, S. P. Mattingly, J. C. Williams, and H. Hashemi, “An optimization-based traffic diversion model during construction closures,” Computer-Aided Civil and Infrastructure Engineering, vol. 34, no. 12, pp. 1087–1099, 2019.
69. J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve,” Radiology, vol. 143, no. 1, pp. 29–36, 1982.
70. E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, “Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach,” Biometrics, pp. 837–845, 1988.
71. S. Surjanovic and D. Bingham, “Virtual library of simulation experiments: Test functions and datasets,” Retrieved April 25, 2017, from http://www.sfu.ca/~ssurjano.
72. N. Hansen, S. Finck, R. Ros, and A. Auger, Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. PhD thesis, INRIA, 2009.
73. S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. Shiomi, “Designing nanostructures for phonon transport via bayesian optimization,” Physical Review X, vol. 7, no. 2, p. 021024, 2017.
74. H. Song, J. J. Thiagarajan, P. Sattigeri, and A. Spanias, “Optimizing kernel machines using deep learning,” IEEE transactions on neural networks and learning systems, vol. 29, no. 11, pp. 5528–5540, 2018.
75. M. S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jolicic, and J. Clune, “A deep active learning system for species identification and counting in camera trap images,” Methods in Ecology and Evolution, vol. 12, no. 1, pp. 185–191, 2021.
76. I. Tsoukalas, P. Kossieris, A. Efstratiadis, and C. Makropoulos, “Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget,” Environmental Modelling & Software, vol. 77, pp. 122–142, 2016.
77. R. C. Bose and K. A. Bush, “Orthogonal arrays of strength two and three,” The Annals of Mathematical Statistics, pp. 508–524, 1952.
78. R. L. Plackett and J. P. Burman, “The design of optimum multifactorial experiments,” Biometrika, vol. 33, no. 4, pp. 305–325, 1946.
APPENDIX

A Exploration-Exploitation Pareto Approach (EEPA)

Dickson [39] proposed an exploration-exploitation Pareto approach for sampling in surrogate optimization. Algorithm 5 presents the EEPA pseudocode. EEPA starts with a set of input points $R$ representing the solution space. For each input point in $R$, it calculates the minimum distance, $\delta(x)$, from the already evaluated set of input points $I$. Next, it determines a non-dominated set of input points $F$ according to $\delta(x)$ and the predicted output values $\hat{f}(x)$ of the input points in $R$. To eliminate the close input points in $F$, EEPA applies a maximin exploration measure to $F$ in order to select the most diverse candidates. The sampled input points are stored in $P$. EEPA limits the size of $P$ to a maximum $K'$ in each iteration due to the expensive black-box evaluation.

Algorithm 5 EEPA, [39]

input: $I$, $\hat{f}$, $R$

1: $\delta(x) = \min_{\tilde{x} \in I} \| x - \tilde{x} \|$, $\forall x \in R$
2: $F = \{ x \in R \mid \exists \tilde{x} \in R, \hat{f}(\tilde{x}) \leq \hat{f}(x), \delta(\tilde{x}) \geq \delta(x) \}$
3: $P = \{ \}; k' = 1$
4: $x' \in \arg\min \{ \hat{f}(x) | x \in F \}$
5: $P = P \cup x'$
6: while $k' \leq K'$ and $P$ do
7: $\delta'(x) = \min_{x \in I, P} \| x - \tilde{x} \|$, $\forall x \in F$
8: $x \in \arg\max \{ \delta'(x) | x \in P \}$
9: $P = P \cup \{ x \}$
10: $k' = k' + 1$
11: end while
12: return $P$
B Additional Experimental Results

This section provides experimental results in addition to those in Section 4.

Table 4 shows the preliminary results of using RBF with different kernels in surrogate optimization for a different fraction of important variables. The values in the table denote the MTFAUC measure and the total number of function evaluations before the optimum solution is found, respectively.

Table 6 shows the ANOVA table of our preliminary analysis based on an orthogonal array design considering different parameters, as listed in Table 5. We fixed the unimportant variables to their most significant level to study the impact of noise on different approaches, in a full factorial setting.

Tables 7, 8, and 9 show the average and variance of the performance of No-Replication, Fixed-Replication, and Smart-Replication for 30 different executions on the test functions listed in Table 2. The highlighted cells show the best method for each test function under different uncertainty levels.

Figure 12 presents the box-plots of the MTFAUC for 30 executions using different approaches on the Rastrigin test function. We can observe that Fixedrep, 10 has the smallest interquartile range and is the most robust approach to uncertainty. The reason is likely due to the highly fluctuating behavior of the Rastrigin function with several local optima combined with the noise added to the function, which makes an optimum hard to obtain.

Figure 13 shows the box-plots of the true objective value of BSMS after 1000 black-box function evaluations for the same 30 executions on the Rastrigin test function. We observe the same pattern for the quality of the final BSMS after 1000 evaluations. Smartrep, 10 finds a slightly better BSMS across the noise levels with relative robustness. Note that Smartrep, 10 performance is independent of the surrogate model used. For example, we can observe that Fixedrep, 10 has higher BSMS value for nonRBF and nonGP at higher levels of noise, which is not the case for Smartrep, 10.

Figure 14 displays the box-plots of the MTFAUC for 30 executions using different approaches on the Levy test function. We note that the interquartile range for Smartrep, 10 is comparatively small across different levels of noise. Note that in this instance, nonGP needs full replication, and the performance of Fixedrep, 10 outperforms Smartrep, 10 when we use nonGP as a surrogate.

Figure 15 shows the box-plots of the true objective value of BSMS after 1000 black-box function evaluations for the same 30 executions on the Levy test function. Smartrep, 10 is the most robust approach to the different noise levels.

Figure 16 presents the box-plots of the MTFAUC for 30 executions using different approaches on the Ackley test function. We observe that Smartrep, 10 has the highest robustness to uncertainty.

Figure 17 shows the box-plots of the true objective value of BSMS after 1000 black-box function evaluations for the same 30 executions. On average, Smartrep, 10 finds better BSMS across different noise levels.

Figure 18 presents the box-plots of the MTFAUC for 30 executions using different approaches on the Zakharov test function. The performance of Smartrep, 10 is noticeable at the higher noise levels. Although the performance of Fixedrep, 10 is competitive to some extent, it conducts unnecessary black-box function evaluations in a deterministic environment.

Figure 19 shows the box-plots of the true objective value of the BSMS after 1000 black-box function evaluations for the same 30 executions on the Zakharov test function. We verified that Smartrep, 10 has the smallest interquartile range overall in higher levels of noise compared with Fixedrep, 10, but at the lower levels of noise, it has larger range compared with Fixedrep, 10.
Fig. 12: Box-plots of MTFAUC of surrogate optimization on the Rastrigin function
Fig. 13: Box-plots of the true objective value of the BSMS after 1000 black-box function evaluations of surrogate optimization on the Rastrigin function
High-dimensional Black-box Optimization Under Uncertainty

Fig. 14: Box-plots of MTFAUC of surrogate optimization on the Levy function
Fig. 15: BBox-plots of the true objective value of the BSMS after 1000 black-box function evaluations of surrogate optimization on the Levy function.
Fig. 16: Box-plots of MTFAUC of surrogate optimization on the Ackley function
Fig. 17: Box-plots of the true objective value of the BSMS after 1000 black-box function evaluations of surrogate optimization on the Ackley function.
Fig. 18: Box-plots of MTFAUC of surrogate optimization on the Zakharov function

(a) Noise=0

(b) Noise=0.05

(c) Noise=0.1

(d) Noise=0.25
Fig. 19: Box-plots of the true objective value of the BSMS after 1000 black-box function evaluations of surrogate optimization on the Zakharov function.
### Table 4: RBF Results

| Function | Multi-Quadric | Gaussian | Thin Plate Spline |
|----------|---------------|----------|------------------|
| fiv = 1  | fiv = 0.75    | fiv = 0.5 | fiv = 0.25       |
| Rosenbrock | 0.0087 | 47 | 0.0077 | 46 | 0.0126 | 52 | 0.0203 | 131 |
| Rastrigin  | 0.0087 | 47 | 0.0087 | 47 | 0.0145 | 53 | 0.0285 | 107 |
| Sphere     | 0.0087 | 47 | 0.0087 | 47 | 0.0079 | 47 | 0.0203 | 59 |
| Levy       | 0.0145 | 53 | 0.0145 | 53 | 0.0283 | 59 | 0.0386 | 83 |
| fiv = 0.75 | fiv = 0.5    | fiv = 0.25 |
| Rosenbrock | 0.7376 | 895 | 0.7156 | 895 | 0.4293 | 895 | 0.2035 | 896 |
| Rastrigin  | 0.0087 | 47 | 0.0087 | 47 | 0.0087 | 47 | 0.0087 | 47 |
| Sphere     | 0.0083 | 47 | 0.0074 | 47 | 0.0087 | 47 | 0.0145 | 53 |
| Levy       | 0.0038 | 40 | 0.0035 | 40 | 0.0079 | 40 | 0.0035 | 40 |
| fiv = 0.5  | fiv = 0.25   |
| Rosenbrock | 0.0183 | 62 | 0.0219 | 65 | 0.0287 | 69 | 0.0434 | 309 |
| Rastrigin  | 0.0241 | 63 | 0.0319 | 71 | 0.0550 | 95 | 0.3004 | 307 |
| Sphere     | 0.0197 | 58 | 0.0183 | 57 | 0.0283 | 59 | 0.0376 | 77 |
| Levy       | 0.0241 | 63 | 0.0261 | 65 | 0.0425 | 82 | 0.0531 | 93 |
| fiv = 0.25 |
| Rosenbrock | 0.0225 | 62 | 0.0219 | 65 | 0.0361 | 78 | 0.0245 | 171 |
| Rastrigin  | 0.0245 | 61 | 0.0261 | 65 | 0.0280 | 68 | 0.0705 | 111 |
| Sphere     | 0.0241 | 63 | 0.0241 | 63 | 0.0193 | 58 | 0.0319 | 71 |
| Levy       | 0.0277 | 75 | 0.0424 | 83 | 0.0338 | 73 | 0.0666 | 107 |

### Table 5: Parameters and levels for OA design in Preliminary Analysis

| Problem Parameters | levels |
|--------------------|--------|
| Test function      | Rosenbrock, Rastrigin, Levy |
| Dimension          | 10, 20, 30 |
| Fraction of important variables | 0.25, 0.50, 0.75, 1 |
| Noise level (%)    | 5, 10, 25 |

| Algorithm Parameters | levels |
|----------------------|--------|
| Initial pool size    | d + 1, 2(d + 1) |
| DOE method           | LHD, sobol |
| EEPA distance        | Euclidean, Cosine |
| EEPA # candidates    | 3, 6 |
| Replication type     | fixed_rep, smart_rep |
| Replication #        | 5, 10 |
| Model                | RBF_avrg, TK-MARS_avrg, TK-MARS_keepall |
### Table 6: Preliminary ANOVA table

| Estimate  | Std. Error | t value | Pr(>|t|) |
|-----------|------------|---------|----------|
| (Intercept)| 0.250806   | 0.086891| 2.886    | 0.005626 ** |
| fiv=0.75  | 0.07965    | 0.056382| 1.413    | 0.163596    |
| fiv=0.50  | 0.207822   | 0.056382| 3.686    | 0.000537 ***|
| fiv=0.25  | 0.132775   | 0.056382| 2.355    | 0.022408 *  |
| model=TK-MARS_avg | -0.04463 | 0.048828 | -0.914   | 0.364833    |
| model=TK-MARS_keepall | 0.113446 | 0.048828 | 2.323    | 0.02404 *   |
| noise level=10% | 0.004939 | 0.048828 | 0.101    | 0.91982     |
| noise level=25% | 0.009702 | 0.048828 | 0.199    | 0.843259    |
| smart_rep | -0.09262   | 0.039868| -2.323   | 0.02404 *   |
| Rastrigin | 0.344011   | 0.048828| 7.045    | 3.81E-09 ***|
| Levy      | 0.045359   | 0.048828| 0.929    | 0.357128    |
| dimension=20 | 0.07677   | 0.048828| 1.509    | 0.137262    |
| dimension=30 | 0.051991  | 0.048828| 1.046    | 0.300151    |
| Replication=10 | 0.146338 | 0.048828| 2.997    | 0.004142 ** |
| poolSize=2(d+1) | -0.01711 | 0.048828 | -0.429   | 0.669465    |
| DOE=Sobol | 0.090609   | 0.039868| 2.323    | 0.02404 *   |
| EEPA distance=Cosine | -0.01711 | 0.048828 | -0.429   | 0.669465    |
| EEPA number of candidates=6 | -0.02138 | 0.039868 | -0.536   | 0.594059    |

Signif. Codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

### Table 7: Average and variance of MTFAUC of different test functions and surrogates at different noise levels

|                      | Rosenbrock | Rastrigin |
|----------------------|------------|-----------|
|                      | RBF        | TK-MARS   | nonRBF    | nonGP    | RBF        | TK-MARS   | nonRBF    | nonGP    |
| noise=0              | 0.25,0.04  | 0.17,0.03 | 0.21,0.03 | 0.76,0.09| 0.93,0.03  | 0.89,0.07 | 0.93,0.04 | 0.95,0.03|
| Fixedrep,5           | 0.42,0.08  | 0.40,0.07 | 0.52,0.09 | 0.99,0.02| 1.00,0.01  | 0.97,0.04 | 1.00,0.01 | 1.00,0.00|
| Fixedrep,10          | 0.62,0.09  | 0.78,0.07 | 1.00,0.00 | 1.00,0.00| 0.99,0.03  | 1.00,0.01 | 1.00,0.01 | 1.00,0.01|
| Smartrep,5           | 0.32,0.06  | 0.32,0.04 | 0.31,0.05 | 0.92,0.09| 0.97,0.03  | 0.92,0.06 | 0.97,0.03 | 0.99,0.01|
| Smartrep,10          | 0.32,0.06  | 0.32,0.04 | 0.31,0.05 | 0.91,0.09| 0.97,0.03  | 0.92,0.06 | 0.97,0.03 | 0.99,0.02|
| noise=0.05           |            |           |           |           |            |           |           |           |
| Fixedrep,5           | 0.29,0.09  | 0.20,0.08 | 0.24,0.06 | 0.64,0.11| 0.89,0.04  | 0.83,0.07 | 0.89,0.05 | 0.91,0.04|
| Fixedrep,10          | 0.54,0.07  | 0.58,0.07 | 0.68,0.08 | 0.96,0.08| 0.99,0.01  | 0.98,0.02 | 1.00,0.01 | 1.00,0.01|
| Smartrep,5           | 0.31,0.07  | 0.26,0.03 | 0.32,0.05 | 0.72,0.14| 0.95,0.04  | 0.88,0.06 | 0.97,0.03 | 0.97,0.03|
| Smartrep,10          | 0.33,0.06  | 0.29,0.05 | 0.35,0.05 | 0.72,0.14| 0.95,0.04  | 0.89,0.05 | 0.96,0.03 | 0.97,0.03|
| noise=0.1            |            |           |           |           |            |           |           |           |
| Fixedrep,5           | 0.43,0.23  | 0.27,0.10 | 0.26,0.13 | 0.62,0.19| 0.87,0.06  | 0.84,0.07 | 0.86,0.07 | 0.89,0.06|
| Fixedrep,10          | 0.63,0.11  | 0.72,0.10 | 0.72,0.10 | 0.97,0.07| 0.99,0.02  | 0.99,0.02 | 0.99,0.01 | 1.00,0.01|
| Smartrep,5           | 0.43,0.11  | 0.43,0.10 | 0.41,0.08 | 0.82,0.11| 0.90,0.05  | 0.87,0.05 | 0.91,0.06 | 0.94,0.05|
| Smartrep,10          | 0.48,0.11  | 0.49,0.09 | 0.49,0.08 | 0.89,0.11| 0.95,0.05  | 0.93,0.05 | 0.96,0.04 | 0.98,0.03|
| noise=0.25           |            |           |           |           |            |           |           |           |
| Fixedrep,5           | 0.62,0.25  | 0.55,0.21 | 0.48,0.20 | 0.85,0.15| 1.00,0.00  | 1.00,0.00 | 1.00,0.00 | 1.00,0.00|
| Fixedrep,10          | 0.63,0.18  | 0.74,0.18 | 0.67,0.16 | 0.99,0.03| 0.99,0.02  | 1.00,0.01 | 1.00,0.00 | 1.00,0.00|
| Smartrep,5           | 0.65,0.20  | 0.74,0.15 | 0.60,0.12 | 0.97,0.07| 1.00,0.00  | 1.00,0.00 | 1.00,0.00 | 1.00,0.00|
| Smartrep,10          | 0.55,0.08  | 0.65,0.11 | 0.67,0.12 | 0.93,0.11| 1.00,0.00  | 0.99,0.04 | 1.00,0.01 | 1.00,0.01|
Table 8: Average and variance of MTFAUC of different test functions and surrogates at different noise levels

|         | Levy         |         | Ackley        |         |
|---------|--------------|---------|---------------|---------|
|         | RBF | TK-MARS | nonRBF | nonGP | RBF | TK-MARS | nonRBF | nonGP |
| noise=0 |     |         |         |       |     |         |         |       |
| Norep   | 0.25,0.05   | 0.17,0.03 | 0.32,0.06 | 0.34,0.04 | 0.43,0.08 | 0.40,0.05 | 0.93,0.05 |
| Fixedrep,5 | 0.54,0.11 | 0.55,0.07 | 0.84,0.07 | 0.58,0.06 | 0.61,0.09 | 0.75,0.06 | 1.00,0.00 |
| Fixedrep,10 | 0.68,0.08 | 0.86,0.07 | 0.95,0.05 | 0.78,0.05 | 0.80,0.08 | 0.94,0.03 | 1.00,0.00 |
| Smartrep,5 | 0.37,0.09 | 0.30,0.05 | 0.80,0.14 | 0.43,0.04 | 0.48,0.08 | 0.54,0.07 | 0.99,0.00 |
| Smartrep,10 | 0.37,0.09 | 0.30,0.05 | 0.80,0.14 | 0.43,0.04 | 0.48,0.08 | 0.99,0.01 | 0.99,0.01 |
| noise=0.05 |     |         |         |       |     |         |         |       |
| Norep   | 0.30,0.10   | 0.22,0.04 | 0.36,0.07 | 0.33,0.02 | 0.43,0.05 | 0.39,0.03 | 0.93,0.04 |
| Fixedrep,5 | 0.38,0.08 | 0.52,0.06 | 0.72,0.13 | 0.52,0.04 | 0.60,0.07 | 0.67,0.05 | 1.00,0.00 |
| Fixedrep,10 | 0.56,0.07 | 0.80,0.08 | 0.89,0.09 | 0.71,0.03 | 0.80,0.06 | 0.89,0.03 | 1.00,0.00 |
| Smartrep,5 | 0.31,0.09 | 0.34,0.05 | 0.73,0.11 | 0.41,0.03 | 0.48,0.04 | 0.52,0.04 | 0.99,0.00 |
| Smartrep,10 | 0.31,0.08 | 0.37,0.04 | 0.70,0.11 | 0.43,0.03 | 0.52,0.06 | 0.99,0.01 | 0.99,0.00 |
| noise=0.1 |     |         |         |       |     |         |         |       |
| Norep   | 0.44,0.11   | 0.32,0.09 | 0.47,0.07 | 0.36,0.02 | 0.54,0.16 | 0.40,0.05 | 0.93,0.04 |
| Fixedrep,5 | 0.46,0.07 | 0.56,0.05 | 0.79,0.10 | 0.58,0.07 | 0.63,0.09 | 0.75,0.08 | 1.00,0.00 |
| Fixedrep,10 | 0.63,0.07 | 0.83,0.08 | 0.98,0.03 | 0.77,0.06 | 0.83,0.09 | 0.94,0.03 | 1.00,0.00 |
| Smartrep,5 | 0.41,0.11 | 0.42,0.05 | 0.80,0.10 | 0.46,0.08 | 0.56,0.14 | 0.54,0.07 | 1.00,0.00 |
| Smartrep,10 | 0.45,0.10 | 0.47,0.06 | 0.80,0.09 | 0.48,0.08 | 0.52,0.07 | 1.00,0.00 | 1.00,0.00 |
| noise=0.25 |     |         |         |       |     |         |         |       |
| Norep   | 0.57,0.15   | 0.41,0.10 | 0.55,0.10 | 0.38,0.05 | 0.51,0.10 | 0.44,0.05 | 0.94,0.04 |
| Fixedrep,5 | 0.56,0.10 | 0.64,0.07 | 0.88,0.09 | 0.56,0.06 | 0.65,0.09 | 0.73,0.07 | 1.00,0.00 |
| Fixedrep,10 | 0.68,0.09 | 0.89,0.07 | 0.98,0.04 | 0.73,0.05 | 0.82,0.07 | 0.92,0.03 | 1.00,0.00 |
| Smartrep,5 | 0.52,0.14 | 0.52,0.06 | 0.85,0.10 | 0.50,0.05 | 0.61,0.11 | 0.65,0.07 | 1.00,0.00 |
| Smartrep,10 | 0.52,0.12 | 0.59,0.07 | 0.85,0.10 | 0.57,0.05 | 0.67,0.11 | 1.00,0.00 | 1.00,0.00 |
Table 9: Average and variance of MTFAUC of different test functions and surrogates at different noise levels

|                | Zakharov | RBF       | TK-MARS   | nonRBF    | nonGP     |
|----------------|----------|-----------|-----------|-----------|-----------|
|                |          | noise=0   |           |           |           |
| Norep          | 0.04,0.00| 0.04,0.01 | 0.04,0.00 | 0.04,0.01 |           |
| Fixedrep,5     | 0.18,0.02| 0.19,0.03 | 0.18,0.02 | 0.20,0.03 |           |
| Fixedrep,10    | 0.36,0.04| 0.39,0.06 | 0.36,0.04 | 0.39,0.05 |           |
| Smartrep,5     | 0.07,0.01| 0.08,0.01 | 0.07,0.01 | 0.08,0.01 |           |
| Smartrep,10    | 0.07,0.01| 0.08,0.01 | 0.07,0.01 | 0.08,0.01 |           |
|                | noise=0.05 |         |           |           |           |
| Norep          | 0.84,0.26| 0.79,0.30 | 0.75,0.36 | 0.85,0.24 |           |
| Fixedrep,5     | 0.54,0.33| 0.55,0.28 | 0.47,0.28 | 0.91,0.22 |           |
| Fixedrep,10    | 0.57,0.23| 0.68,0.22 | 0.56,0.17 | 0.92,0.17 |           |
| Smartrep,5     | 0.52,0.34| 0.55,0.25 | 0.47,0.30 | 0.60,0.34 |           |
| Smartrep,10    | 0.50,0.23| 0.64,0.25 | 0.50,0.24 | 0.58,0.24 |           |
|                | noise=0.1 |         |           |           |           |
| Norep          | 0.59,0.37| 0.59,0.35 | 0.54,0.37 | 0.59,0.37 |           |
| Fixedrep,5     | 0.58,0.32| 0.68,0.31 | 0.56,0.28 | 0.73,0.28 |           |
| Fixedrep,10    | 0.78,0.21| 0.80,0.22 | 0.75,0.24 | 0.76,0.25 |           |
| Smartrep,5     | 0.56,0.34| 0.62,0.28 | 0.57,0.30 | 0.68,0.32 |           |
| Smartrep,10    | 0.72,0.25| 0.76,0.26 | 0.71,0.26 | 0.81,0.28 |           |
|                | noise=0.25 |         |           |           |           |
| Norep          | 0.71,0.30| 0.74,0.31 | 0.71,0.32 | 0.69,0.34 |           |
| Fixedrep,5     | 0.51,0.28| 0.58,0.29 | 0.49,0.25 | 0.73,0.26 |           |
| Fixedrep,10    | 0.60,0.23| 0.64,0.21 | 0.63,0.24 | 0.79,0.22 |           |
| Smartrep,5     | 0.49,0.27| 0.57,0.28 | 0.48,0.24 | 0.56,0.25 |           |
| Smartrep,10    | 0.58,0.26| 0.60,0.21 | 0.59,0.27 | 0.61,0.25 |           |