Project Number: 766186
Project Acronym: ECOLE
Project title: Experienced-based Computation: Learning to Optimize

Deliverable D2.3
Robustness & uncertainty modeling in experience-based optimization

Authors:
Sibghat Ullah, Thomas Bäck, and Hao Wang – Universiteit Leiden

Project Coordinator: Professor Xin Yao, University of Birmingham
Beneficiaries: Universiteit Leiden, Honda Research Institute Europe, NEC Laboratories Europe

H2020 MSCA-ITN
Date of the report: **31.03.2021**

Contents

Executive Summary

This document provides a concise report on the research invested and the scientific contributions made regarding the work package 2.3 in ECOLE. This work package deals with the issue of uncertainty and/or noise within the context of black-box design optimization. A systematic empirical comparison was performed in ECOLE to discuss the suitability of surrogate models for solving such problems in the face of uncertainty, in addition to the determination of training sample size. The findings suggest that in 32/36 test cases, surrogate-assisted optimization yields an optimal or suboptimal value of the original function, based on a 100 independent runs of the Sequential Least Square Programming algorithm for global optimization. The training sample size for these surrogate models was set to be linear in the search dimensions. Overall, the results demonstrate the superiority of Kriging, Support Vector Machines, and Polynomial regression as they achieve high modeling accuracy in 34/36 test cases, and the optimal point on the model landscape is close to the true optimum of the test function in these cases.

Major Achievements

Major scientific achievements regarding the work package 2.3 are presented below. In particular, short answers to some of the most important research questions – practical issues – are described:

Research Questions	Discussion
Is surrogate modeling applicable to solve robust optimization problems – problems affected by uncertainties and/or noise?	Our findings indicate the practical applicability of surrogate modeling for robust optimization (cf. Table VI).
How much training data is required to construct a (good) surrogate model of the objective function under uncertainty?	For optimization problems with small dimensionality, the training sample size can be a linear function of dimensionality (cf. Figs. (3-8)). For problems with high dimensionality, further research is needed.
Which modeling technique is best suited to solve the robust optimization problems?	Our findings indicate (and validate) Kriging, Polynomials and Support Vector Machines as the most promising modeling techniques in this context (cf. Tables (IV-VI)).
Does noise level significantly affect the quality of the surrogate models?	Our findings do not indicate that (cf. Figs. (3-8)). Note, however, that our research here does not involve complex problems, e.g., with high dimensionality, multi-objective cases, multiplicative noise. Further research is required to validate it on more complex cases.

1. Introduction

ECOLE aims at shortening the product-development cycle, reducing the resource consumption during the complete process, and creating more balanced and innovative products. One of the most important challenges ECOLE addresses the presence of uncertainties and/or noise within the system (or model of the system), for which optima are sought. The issue of handling uncertainties and/or noise in the design optimization process is important for two main reasons:

1) Uncertainties or noise can affect the objective landscape significantly [1]. Therefore, the theoretical optimum found by common optimization algorithms may not be optimal for practical applications where unexpected drift and changes can occur.
2) Uncertainties or noise can affect the accuracy and convergence speed of optimization algorithms [1] [2], thereby directly affecting the quality of the optimal solution.

This report summarizes the work and research invested in work package 2.3 which embroils the issue of efficiently solving optimization problems subject to uncertainty. In this report, we therefore summarize the following scientific findings and research outcomes pertaining to the work package 2.3:

- Literature study on the presence of uncertainty and/or noise in continuous black-box optimization (Sec. 2). We concisely summarize the sources of uncertainty and/or noise based on the existing work [1] [2] [3] [4] [5]. Sec. 2 is further associated with a summary of methodologies for modeling uncertainties, and practical scenarios for robust optimization (RO).
- Literature study on robust optimization (Sec. 3). We encapsulate the practical goal for robust optimization based on *robust counterpart approach* [3]. Two robust counter approaches – robust regularization based on the minimax principle, and composite robustness – are described.
- Empirical investigation of model-assisted robust optimization (Sec. 4). We provide a comprehensive empirical analysis on several practical issues for RO. In this section, we focus on efficiently solving the RO problems.

The following publication of the ECOLE project is contributing to this report:

S. Ullah, H. Wang, S. Menzel, B. Sendhoff and T. Bäck, "An Empirical Comparison of Meta-Modeling Techniques for Robust Design Optimization," in *IEEE Symposium Series on Computational Intelligence*, Xiamen, 2019.

2. Uncertainty & Noise in Optimization

Uncertainty is a recurring motif in design optimization. The classical view on black-box optimization does not account for uncertainty [4]. It is important to state that in our research, uncertainty and/or noise refer to the same concept – unexpected drifts and changes in the optimization setup. These unexpected drifts can be found in the design and environmental parameters, e.g., temperature, stiffness, and structural rigidity, as well as in the constraints and objectives [3]. Accounting for these uncertainties leads to the concept of robust design optimization (RDO), also called quality engineering [4]. Before discussing the RDO, we provide an overview of different sources of uncertainty and/or noise, various methodologies to deal with these uncertainties and/or noise, and different scenarios of optimization under uncertainty. An example of the black-box optimization loop of a system is provided in Fig. 1, where an optimizer is coupled with the system, or model of the system, for which an optimum is sought. The optimizer generates some candidate solution(s), which are fed to the system, and some quality score is received. Based on this score, the optimizer generates a new set of solutions, and repeats the same process. This loop is repeated until either a satisfactory solution has been found, or a predefined computational budget or termination criterion has been reached.

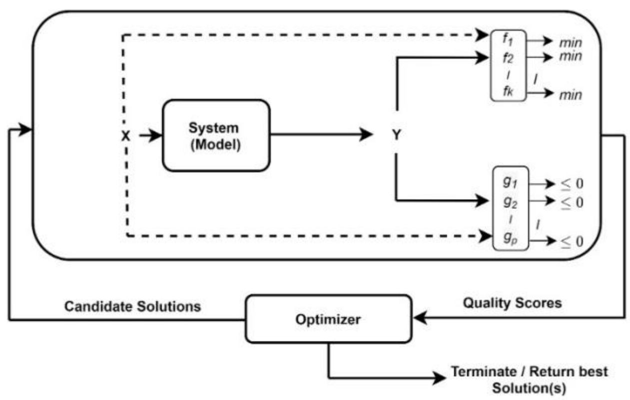

Figure 1. The general setup for black-box optimization.

2.1. Sources of Uncertainty & Noise

One of the ways to distinguish between different forms of uncertainty and/or noise in design optimization is by analyzing different parts of the system and looking for the sources of uncertainty and/or noise [2]. It is important to state that here the black-box design optimization problems are referred to the problems where only the access to evaluating the function value is provided, and any analytical properties, e.g., gradient/Hessian/convexity, are not available to the optimization process. From Fig.1, it can be argued that uncertainties and/or noise can arise in different parts of the optimization pipeline. This aspect is further highlighted in Fig. 2, where five common sources of uncertainty and/or noise are highlighted. These sources include:

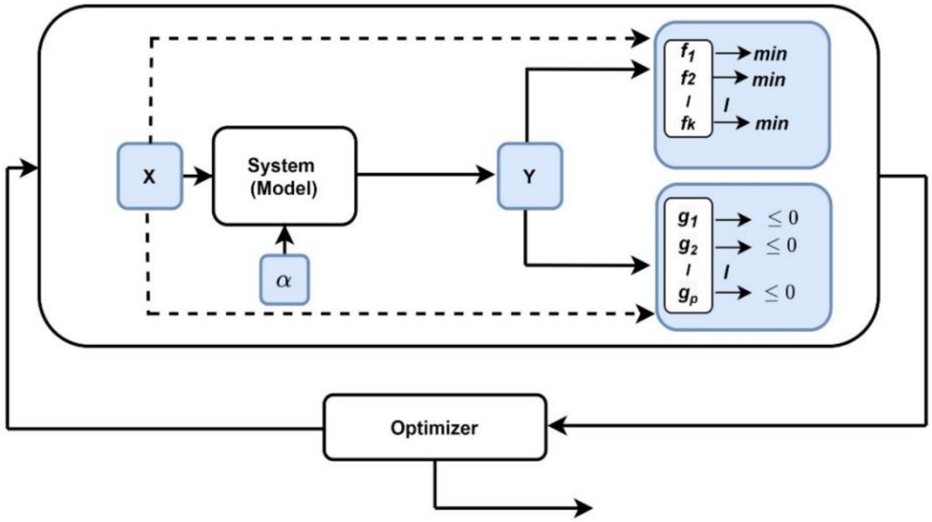

Figure 2. The general black-box optimization setup highlighting different parts of the system where uncertainties can arise.

A. Design variables $\boldsymbol{X}$
B. Environmental parameters $\boldsymbol{\alpha}$
C. Output $\boldsymbol{Y}$
D. Constraints $\boldsymbol{g_i}$
E. Objectives $\boldsymbol{f_i}$

Type A uncertainties occur since in the real-world realizations, the design variables can be controlled with limited precision only. Type B uncertainties occur due to the uncontrollable external factors, e.g., outside temperature, wind speed, which can influence the performance of a system, for which optima are sought. Note that in the classical black-box design optimization as characterized in Fig. 1, such factors are not included. Nonetheless, it is important to discuss that such external factors can influence the performance of an otherwise predictable or stable system [1]. Type C uncertainties are mostly caused due to the stochastic nature of the measurements and the evaluation of the system. Type D uncertainties characterize the inherent vagueness in the constraints while formulating a design optimization problem. Type E uncertainties emerge while having multiple conflicting objectives. As such, a natural tradeoff between different objectives exists.

2.2. Modeling Uncertainty & Noise

Another way to distinguish between different types of uncertainties and/or noise is to consider their nature. Here, it is also appropriate to explain the difference between uncertainty and noise. The difference between the uncertainty and noise is the same as the difference between epistemic and aleatory uncertainty [6]. Epistemic uncertainty refers to the uncertainty which is due to the lack of knowledge and/or data, and is, in principle, reducible. Noise, on the other hand, is the same as the aleatory uncertainty – uncertainty due to the inherent stochastic nature of a measurement/quantity. Note, however, that classifying different types of uncertainties in real-world scenarios is often not straightforward. Therefore, it is essential to classify different types of uncertainties based on the way they can be mathematically modeled within an optimization problem. To this end, one can distinguish between three different classes [1]:

1) Fuzzy
2) Deterministic
3) Probabilistic

Modeling uncertainties using *fuzzy logic* refers to formulating *fuzzy* statements about the possibility of the state of the uncertain variable. This is usually done with the help of *fuzzy* sets. Modeling uncertainties in a *deterministic* way refers to modeling the *crisp* possibility of whether a particular state of an uncertain variable is possible or not. This is usually done with the help of a *crisp* set. The last way of modeling uncertainty is the most common in design optimization problems. Here, a *probability* measure can be established on the frequency of event(s) that may happen. This is done with the help of *probability density functions* for continuous variables, and *probability mass functions* in the case of discrete variables. It is also logical to build an intuition here that aleatory uncertainty is naturally modeled in a probabilistic fashion. On the other hand, epistemic uncertainty can be mathematically modeled using all three approaches. A schematic view of different ways to distinguish uncertainties and/or noise is provided in Table I for further clarification.

Table I. A classification of different types of uncertainties based on their mathematical properties and conceptual distinction.

Conceptual Classification	**Mathematical Modeling**	**Mathematical Properties**
Epistemic (uncertainty)	Possibilistic	Uncertainty domain unknown, probabilities unknown
	Deterministic	Uncertainty domain known; probabilities unknown
Aleatory (noise)	Probabilistic	Uncertainty domain known; probabilities known

2.3. Scenarios of Uncertainty & Noise

Five different sources of uncertainty and/or noise, combined with three different ways to model them within the optimization problem, have been identified. This gives rise to 15 different scenarios that can be faced when solving real-world design optimization problems. Note, however, that some of these scenarios occur more often than others. As such, from a practical point of view, it is desirable to focus on those scenarios. Table II summarizes the sources (classes) of uncertainties alongside the mathematical ways to model them. From the discussion in this section, it is clear that there are at least five most common sources of uncertainty and/or in noise in the black-box design optimization pipeline. Furthermore, there are at least three different ways of mathematically modeling these uncertainties. A distinction between aleatory and epistemic uncertainty, and correspondingly, between noise and uncertainty [7] has also been made. Note, however, that within the scope of ECOLE, discussing all these manifestations of uncertainties is infeasible. Therefore, it is logical to choose a set of uncertainties which are most interesting to study from the practical point of view of design optimization. To this end, Type A and B uncertainties are selected which are the most common ones in design optimization. The reason for selecting these types of uncertainties include the limited machine precision, i.e., the implementation errors, production tolerances, and the change in environmental and operating conditions. For the remainder of this report, we will therefore discuss Type A and B uncertainties – uncertainties which are commonly found in the design and environmental parameters of a design optimization problem. Moreover, our way of modeling these uncertainties in ECOLE will be based on deterministic sets and probability distributions. Having discussed the different manifestations of uncertainty and/or noise, the next section provides an overview on RO.

Table II. Classification and categorization of different manifestations of uncertainty and/or noise.

Class	**Type**
A) Uncertainties and/or noise in the design-variables	Possibilistic
	Deterministic
	Probabilistic
B) Uncertainties and/or noise in the environmental parameters	Possibilistic
	Deterministic
	Probabilistic
C) Uncertainties and/or noise in the output	Possibilistic
	Deterministic
	Probabilistic
D) Vagueness in the constraints	Possibilistic
	Probabilistic
E) Preference uncertainty in the objectives	Possibilistic
	Deterministic

3. Robust Optimization

Various sources of uncertainty and/or noise within optimization problems have been analyzed in Sec 2. Different ways to mathematically model these uncertainties have also been described. Type A and B uncertainties have been identified as the most relevant ones to the project. This section shares the project's approach towards modeling these uncertainties in a deterministic or a probabilistic fashion. Based on this, it is possible to measure the impact of these uncertainties on the optimization problems, and accordingly adapt the optimization setup to minimize the effect of these uncertainties. This is known in the literature as the *Robust Counterpart Approach* (RCA) [3], which is discussed in Sec. 3.1.

3.1. Robust Counterpart Approach

Robust optimization (RO) is often referred to as the practice of optimization which deals with the unexpected drifts and changes in the optimization setup. For the uncertainties considered in the ECOLE project, two most important questions [2] need to be tackled:

- In what ways do these uncertainties affect the optimization algorithms and the practical applicability of the solution(s) found by the algorithms?
- How should the optimization algorithms be adapted to mitigate the effects of these uncertainties?

For handling these two questions, the most suitable approach turns out to be the replacement of the original optimization goal with the goal of RO. This approach is often referred to as the *Robust Counterpart Approach* (RCA), and is the most common approach in engineering applications. The goal of RO, however, depends on the source(s) of uncertainty and the way in which the uncertainty is mathematically modeled. In general, we can define the practical goal of RO [2] as:

"Given an optimization problem with uncertainty and/or noise, and given an optimization goal and a limited number of computational resources, the goal of robust optimization is to use these computational resources to find the best solution(s) despite uncertainty and/or noise, which are still optimal and useful in the face of uncertainties/noise."

Keeping the general goal of RO in mind, RCA emphasizes on two aspects. Firstly, modeling the uncertainties and/or noise in different parts of the system in a probabilistic or a deterministic fashion. Secondly, adapting the practical goal of optimization to minimize the effects of modeled uncertainties and/or noise in the first step. In the case of Type A and B uncertainties, this leads to two different concepts – the *worst-case robust optimization* based on the minimax principle, and the *composite robustness*. These two concepts model the uncertainties in deterministic and probabilistic fashion respectively, and thereon adapt the goal of optimization based on modeled uncertainties.

3.2. The Minimax Principle

In ECOLE, we emphasize on Type A and B uncertainties – uncertainties which are commonly found in the design and environmental parameters of a design optimization problem. The robustness taking into account the uncertainties of Type A and B is referred to as *Sensitivity Robustness* [1]. In ECOLE, we consider real-valued black-box optimization problems of the form $f: \boldsymbol{\mathcal{S}} \to \mathbb{R}, \boldsymbol{\mathcal{S}} \subseteq \mathbb{R}^{\boldsymbol{d}}$, where the so-called feasible region $\boldsymbol{\mathcal{S}}$ is specified by the inequality constraints $g_j(X) \leq 0\ (j \in \{1, \dots, J\})$, and the equality constraints $h_k(X) = 0\ (k \in \{1, \dots, K\})$, $\mathrm{X} \in \boldsymbol{\mathcal{S}}$. Without loss of generality, the objective function f is subject to minimization. The effect of additive noise $\delta_{\mathrm{X}} \in \mathbb{R}^d$ in design parameters of the objective function is formulated as:

$$\widehat{\boldsymbol{f}(\boldsymbol{x})} = \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{x}}), \qquad \textbf{(1)}$$

where $\widehat{f(x)}$ is the noisy counterpart of the objective function $f(x)$. The minimax principle [3] emphasizes on minimizing the worst output (highest value) of the noisy function $\widehat{f(x)}$. Given an objective function $f(x)$ to be minimized, this RCA first defines a neighborhood $\boldsymbol{N}_{\boldsymbol{\epsilon}}(\boldsymbol{x})$ of design $\boldsymbol{x}$ whose size is determined by the parameter $\epsilon > 0$. Then, an upper-bound of $f(x)$ is defined by taking into account the worst output (highest value) of design x including neighborhood $N_\epsilon(x)$. Finally, the goal of RO is defined as minimizing this upper bound [1] [3]. The minimum returned by using this strategy is called the "least upper-bound". Formally, the robust counterpart $R(x)$ based on this principle is defined as:

$$\boldsymbol{R}(\boldsymbol{x}) = \quad \mathbf{sup}\ \boldsymbol{f}(\boldsymbol{\xi}), \qquad \boldsymbol{where}\ \ \boldsymbol{\xi} \in \boldsymbol{N}_{\boldsymbol{\epsilon}}(\boldsymbol{x}) \qquad \textbf{(2)}$$

The goal of the RO then becomes to solve Eq. (2). This RCA is known as the *robust regularization* since ϵ acts as a regularization parameter, i.e., it defines the size of the neighborhood.

3.3. Composite Robustness

Different from the minimax principle, the designer can also optimize the expected output of a noisy function $\mathbb{E}[\widehat{f(x)}]$ whilst minimizing the dispersion, i.e., standard deviation $\sigma\,[\widehat{f(x)}]$, simultaneously [5]. In ECOLE, we refer to it as the *composite robustness*, same as in [5]. This robustness criterion, however, requires the noise δ_{X} to be specified in the form of a probability distribution. The expectation $\mathbb{E}[\widehat{f(x)}]$ and dispersion, i.e., standard deviation, $\sigma\,[\widehat{f(x)}]$ of the noisy objective function are combined at each design point x in $\boldsymbol{\mathcal{S}}$ to produce a robust scalar output. The RO goal thus becomes to find a design point x^* in $\boldsymbol{\mathcal{S}}$ which minimizes this scalar. This robust composition is formally defined in Eq. (3):

$$\boldsymbol{minimize}\ \boldsymbol{R}(\boldsymbol{x}) = \mathbb{E}[\widehat{\boldsymbol{f}(\boldsymbol{x})}] + \boldsymbol{\sigma}\,[\widehat{\boldsymbol{f}(\boldsymbol{x})}] \qquad \textbf{(3)}$$

where $\widehat{f(x)} = f(x + \delta_x)$ and $\delta_x \sim \mathcal{N}(0, \mathrm{s}^2)$. Notably, $\boldsymbol{s}$ denotes the standard deviation of the probability distribution of the noise, and depends on the range of design parameters and noise level chosen.

4. Surrogate-Assisted Robust Optimization

The idea of *surrogate-assisted robust optimization* (SARO) is to employ an empirical approximation model, called the surrogate model or a metamodel, for optimizing a noisy objective function $\widehat{\boldsymbol{f}(\boldsymbol{x})}$ such as the one presented in Eqs. (1-3) [5]. This section provides a brief introduction to surrogate modeling. It then shifts to discuss the basics of SARO. Finally, the results from ECOLE [8] are presented.

4.1. Surrogate-Assisted Optimization

The idea of surrogate modeling is to build an empirical approximation model $\widehat{\boldsymbol{f}(\boldsymbol{x})}$ of an objective function $\boldsymbol{f}(\boldsymbol{x})$. The approximation $\widehat{\boldsymbol{f}(\boldsymbol{x})}$ then acts as the surrogate model, also called the metamodel of $\boldsymbol{f}(\boldsymbol{x})$. This abstraction is useful in many situations. For instance, it simplifies the task to a great extent in simulation-based modeling and optimization [9]. It can also provide the designer with an opportunity to evaluate $\boldsymbol{f}(\boldsymbol{x})$ indirectly if the exact computation of $\boldsymbol{f}(\boldsymbol{x})$ is too costly and/or complex. Additionally, it can provide practically useful insights to the designer. Surrogate modeling first evaluates $\boldsymbol{f}(\boldsymbol{x})$ at several design points of interest $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N})$, e.g., using Latin hypercube sampling, Plackett-Burman, Box-Behnken design, and generates the data set of input and output pairs $-$ sampled design points and resulting function values. The data generated in this fashion is used to build a nonlinear regression model for approximating the objective function $\boldsymbol{f}(\boldsymbol{x})$. In principle, any regression-based machine learning [10] methodology, e.g., Kriging, Neural networks, Polynomials, may be used. Since no regression model can perfectly approximate the original function $\boldsymbol{f}(\boldsymbol{x})$ by means of a limited number of evaluations, the resulting model $\widehat{\boldsymbol{f}(\boldsymbol{x})}$ will be relatively inaccurate.

Although surrogate modeling provides useful abstraction for solving complex and/or costly optimization problems, it must be carefully employed. For instance, selecting the set of input points $\{(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N})\}$ to evaluate $\boldsymbol{f}(\boldsymbol{x})$ is not straightforward. To this end, the designer can take help from design of experiment (DoE) methodologies such as Latin hyper-cube sampling, factorial designs etc. Note also that the (training) sample size $\boldsymbol{N}$ to evaluate $\boldsymbol{f}(\boldsymbol{x})$ is another parameter that can be varied, without knowing a priori what the optimal setting is. This is since the designer is interested in finding a surrogate $\widehat{\boldsymbol{f}(\boldsymbol{x})}$ (almost) as good as the original function $\boldsymbol{f}(\boldsymbol{x})$ with minimum training data $-$ function evaluations. Consequently, in most applications, the designer must come to a compromise on the quality of the surrogate model and computational budget.

4.2. Empirical Comparison of Surrogate Models for Robust Optimization

It has been already established that surrogate models aim at replacing the often complex and/or costly function evaluations with the help of an empirical approximation model [11]. This model can be used in a variety of ways. In the context of global optimization, this model replaces the function evaluations within the optimization loop. Many different supervised learning techniques [12], e.g., Neural Networks, Kriging and Random Forest, have been applied as surrogate models. Note, however, that surrogate models were initially proposed to replace deterministic computer simulations. Consequently, their suitability in the context of optimization under uncertainty remains to be evaluated [5]. Earlier studies ignored the multivariate analysis of the performance of the surrogate models in the context of RO. In ECOLE [8], we performed a comprehensive quality assessment of the surrogate modeling techniques for the scenarios of RO. Our research involved the impact of dimensionality, problem landscape, noise level, robustness criterion and sample size. This section presents some of the most important details on the experimental setup and the associated results.

4.2.1. Test Problems

Six unconstrained, single-objective optimization problems were chosen in ECOLE. Each of these problems was uniquely identified based on the choice of the test function and dimensionality $D \in \{2,5,10\}$. The chosen test functions are known as Ackley, Branin, Sphere and Rastrigin. Among these test functions, Branin was only defined for $2D$, Sphere for $5D$, Rastrigin for $10D$ and Ackley was tested for all three dimensionality values. This led to six optimization problems. Additionally, each one of these problems was investigated on three different levels of additive noise $-$ $\{5, 10, 20\}$ % noise perturbation based on the nominal values of the design parameters $-$ and two robustness strategies $-$ robustness based on the minimax principle and the composite robustness. All six optimization problems are presented in Table III, including the box constraints and key landscape characteristics.

Table III. All six optimization problems with test functions, key landscape characteristics, dimensions, and box constraints.

Function	Landscape	Dimensionality	Bounds
Ackley	Multi-Modal	2	$x_i \in [-32.768, 32.768]$
Branin	Multi-Global	2	$x_1 \in [-5,10]$, $x_2 \in [0,15]$
Ackley	Multi-Modal	5	$x_i \in [-32.768, 32.768]$
Sphere	Isotropic	5	$x_i \in [-5,5]$
Ackley	Multi-Modal	10	$x_i \in [-32.768, 32.768]$
Rastrigin	Multi-Modal	10	$x_i \in [-5.12, 5.12]$

4.2.2. Noise Levels

In our research, three levels of additive noise δ_x were selected. The effect of additive noise in design parameters is now elaborated. Let $R = |u - l|$ be the absolute range of design parameters, where u and l serve as the upper and lower bounds of the box constraints. Further, let Z be the additive noise level. For the case of robust optimization based on the minimax principle, this means having a neighborhood of design point x whose scale ϵ is defined by the parameters range R and noise level Z. As an example, the Ackley function is commonly defined from $l = -32.768$ to $u = 32.768$, having an absolute range of $R = 65.536$. Considering the first noise level $-$ $Z = 5$ % in Eq. (2) $-$ this means the regularization parameter $\epsilon = Z * R = 0.05 * 65.536 = 3.2768$. In ECOLE, we subsumed that noise was symmetric. Hence, a neighborhood $[-3.2768, 3.2768]$ of each design point x for the Ackley function is constructed and Eq. (2) can be solved. Note also, that we classified all parameters as design ones. However, the real-world engineering applications can have additional environmental parameters which can be modeled in exactly the same way. For composite robustness in our research, we employed a normal distribution $\mathcal{N}(0, s^2)$, where the standard deviation is $s = Z * R/6$, and Z and R serve as the noise level and the absolute range of parameters.

4.2.3. Surrogate Models and Sample Size

The selected modeling techniques were Kriging, Support Vector Machines (SVMs), Radial Basis Function Network (RBFN), Random Forest (RF), K-Nearest Neighbors (KNNs) and Elastic-net with second order polynomial regression function (ELN), respectively. The surrogate models were evaluated on two criteria − modeling accuracy and the quality of the robust optimal solution. The former was measured using the so-called relative mean absolute error (RMAE):

$$\text{RMAE} = \frac{\mathbf{1}}{\boldsymbol{M}}\sum_{\boldsymbol{l=1}}^{\boldsymbol{M}} \mathbf{100} \cdot \left(\frac{|\boldsymbol{y_l} - \widehat{\boldsymbol{y}_l}|}{|\boldsymbol{y_l}|}\right) \tag{4}$$

where $\boldsymbol{M}$ denotes the size of the testing data set. Similarly, $\boldsymbol{y_l}$ and $\widehat{\boldsymbol{y}_l}$ correspond to the true and approximated/predicted function values. In all cases, the surrogate modeling techniques were trained on ten different training sample sizes as: $\boldsymbol{N} = \{\mathbf{5}, \mathbf{10}, \mathbf{15}, \mathbf{20}, \mathbf{25}, \mathbf{30}, \mathbf{35}, \mathbf{40}, \mathbf{45}, \mathbf{50}\} \times \boldsymbol{D}$, where $\boldsymbol{D}$ stands for the dimensionality. The sampled design points $\{(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})\}$ **and** $\{(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_M})\}$ to train and test the surrogate model were generated using the maximum-distance Latin hyper-cube sampling scheme. To achieve the best results in the model training, a detailed hyperparameter optimization (HPO) with cross validation was also performed.

4.2.4. Appraising the Surrogate Models

The criterion of RMAE in Eq. (4) helps to understand the modeling accuracy and computational tractability − accuracy vs computational budget − of the modeling techniques. In ECOLE, another criterion to evaluate the surrogate models based on the quality of the optimal robust solution was also utilized. For this criterion, however, the optimal values on the surrogate models $\widehat{\boldsymbol{R}(\boldsymbol{x})}$ and the robust optimization problem $\boldsymbol{R}(\boldsymbol{x})$ were found with the help of a benchmark optimization algorithm. To this end, Sequential Least Square Programming (SLSQP) was chosen as the benchmark. This second criterion helped to understand the reliability of the surrogate models in practical situations for RO. To evaluate the surrogate models on this criterion, each surrogate was first trained using HPO on a training sample of $\boldsymbol{N} = \mathbf{50} \times \boldsymbol{D}$, where $\boldsymbol{D}$ denotes the dimensionality of the problem. An optimization run with SLSQP was then performed on the trained surrogate to find the optimal values of $\boldsymbol{R}(\boldsymbol{x})$. This process was repeated for $\mathbf{100}$ times, and the mode of the group was chosen as the final optimal value of the $\boldsymbol{R}(\boldsymbol{x})$ using the surrogate model.

4.2.5. Results

Graphs depicting the accuracy of the surrogate models by varying the (training) sample size, evaluated on the basis of RMAE are presented in Figs. (3-8). Standard error (SE) for each RMAE computation is also presented in the graphs. Fig. 3 shares the RMAE values on Ackley $\mathbf{2D}$.

Likewise, Fig. 4 shows the RMAE values for Branin $\boldsymbol{2D}$. Fig. 5 depicts these values for Ackley $\boldsymbol{5D}$. Fig. 6 presents these results for Sphere $\boldsymbol{5D}$. Fig. 7 presents the RMAE values concerning Ackley $\boldsymbol{10D}$ and lastly, Fig. 8 shows the results on Rastrigin $\boldsymbol{10D}$. Note that for each case, the two best modeling techniques were selected based on the measure of the lowest RMAE, averaged over all values of training size $\boldsymbol{N}$. A Mann-Whitney U statistical test was then performed to find the overall best modeling technique for that case. The $\boldsymbol{p}$-values resulting from this test are presented in Table IV. In that table, the first column reads the optimization problem under consideration, the second column presents the robustness strategies $-$ RR for the robust regularization and RC for the robust composition $-$ the third column reports the noise-level, the fourth column describes the two best modeling techniques as indicated above, while the last column shows the $\boldsymbol{p}$-value resulting from the statistical test. The frequencies of the modeling techniques to achieve the highest accuracy for all problems are presented in Table V. Table V follows the same evaluation criterion $-$ lowest average RMAE.

The results concerning the optimality of $\boldsymbol{R}(\boldsymbol{x})$ for all thirty-six cases are presented in Table VI. In that table, the first column reports the optimization problem, the second column shows the robustness scheme (RR for the robust regularization and RC for the robust composition), the third column presents the noise level, the fourth column shows the optimal value of $\boldsymbol{R}(\boldsymbol{x})$, and all the next columns depict the optimal values of $\boldsymbol{R}(\boldsymbol{x})$ proposed by the surrogate models.

Observations from Figs. (3-8) suggest that robustness schemes and noise levels did not have much impact on the accuracy of the surrogate models since similar patterns $-$ RMAE curves – are detected across rows and columns. Furthermore, these figures show that the Radial Basis Function Network (RBFN) had the most variance in prediction. Likewise, these figures characterize that setting (linear) training size $\boldsymbol{N} = \boldsymbol{KD}$ resulted in good modeling accuracy. Note that $\boldsymbol{K} \in \{\boldsymbol{5}, \boldsymbol{10}, \boldsymbol{15}, \boldsymbol{20}, \boldsymbol{25}, \boldsymbol{30}, \boldsymbol{35}, \boldsymbol{40}, \boldsymbol{45}, \boldsymbol{50}\}$ is a scalar and $\boldsymbol{D}$ denotes the dimensionality of the problem. Hence, it can be argued that computational complexity of the surrogate models in our research was a linear function of $\boldsymbol{D}$. Table IV displays that Kriging, Support Vector Machines (SVM) and Polynomial regression (ELN) achieve high accuracy in most test cases. Kriging, in particular, performed well on Branin $\boldsymbol{2D}$ and Sphere $\boldsymbol{5D}$. Polynomial regression (ELN) performed well on all $\boldsymbol{5D}$ and $\boldsymbol{10D}$ cases whereas SVMs performed excellently in most cases. Table V argues that K-Nearest Neighbors (KNNs) and Random Forest (RF) do not achieve high accuracy when compared with the other modeling techniques. Finally, Table VI demonstrates that the surrogate models were able to find an optimal or suboptimal solution in most cases except Rastrigin $\boldsymbol{10D}$.

Table IV. All 36 test cases resulted from the combination of surrogate models, three noise levels, two robustness definitions and six optimization problems. In each test case, we pick the best two surrogate models in terms of average RMAE. Given the alternative hypothesis, the Mann-Whitney U statistical test is performed to check if the surrogate with highest average accuracy is significantly better than the alternate using $\alpha = 0.05$. *The resulting p-values are presented.*

Problem	**Robustness**	**Noise Level**	$\boldsymbol{H_a}$	$\boldsymbol{p}$**-value**
Ackley $\boldsymbol{2D}$	RR	**5** %	$\boldsymbol{KNN < SVM}$	$\boldsymbol{0.425}$
Ackley $\boldsymbol{2D}$	RR	**10** %	$\boldsymbol{KNN < SVM}$	$\boldsymbol{0.192}$
Ackley $\boldsymbol{2D}$	RR	**20** %	$\boldsymbol{KNN < SVM}$	$\boldsymbol{0.012}$

Ackley $2D$	RC	5 %	$SVM < KNN$	0.153
Ackley $2D$	RC	10 %	$SVM < KNN$	0.136
Ackley $2D$	RC	20 %	$SVM < KNN$	0.285
Branin $2D$	RR	5 %	$SVM < Krig$	0.395
Branin $2D$	RR	10 %	$Krig < SVM$	0.425
Branin $2D$	RR	20 %	$Krig < SVM$	0.153
Branin $2D$	RC	5 %	$Krig < SVM$	0.425
Branin $2D$	RC	10 %	$Krig < SVM$	0.425
Branin $2D$	RC	20 %	$Krig < KNN$	0.037
Ackley $5D$	RR	5 %	$SVM < ELN$	0.010
Ackley $5D$	RR	10 %	$SVM < ELN$	0.010
Ackley $5D$	RR	20 %	$SVM < ELN$	0.010
Ackley $5D$	RC	5 %	$SVM < ELN$	0.0001
Ackley $5D$	RC	10 %	$SVM < ELN$	0.0001
Ackley $5D$	RC	20 %	$SVM < ELN$	0.0001
Sphere $5D$	RR	5 %	$Krig < ELN$	0.0001
Sphere $5D$	RR	10 %	$Krig < ELN$	0.0001
Sphere $5D$	RR	20 %	$Krig < ELN$	0.0001
Sphere $5D$	RC	5 %	$Krig < ELN$	0.0001
Sphere $5D$	RC	10 %	$Krig < RBFN$	0.0001
Sphere $5D$	RC	20 %	$Krig < ELN$	0.002
Ackley $10D$	RR	5 %	$SVM < ELN$	0.012
Ackley $10D$	RR	10 %	$SVM < ELN$	0.008
Ackley $10D$	RR	20 %	$SVM < ELN$	0.015
Ackley $10D$	RC	5 %	$ELN < SVM$	0.285
Ackley $10D$	RC	10 %	$ELN < SVM$	0.260
Ackley $10D$	RC	20 %	$ELN < SVM$	0.12
Rastrigin $10D$	RR	5 %	$ELN < SVM$	0.06
Rastrigin $10D$	RR	10 %	$ELN < SVM$	0.06
Rastrigin $10D$	RR	20 %	$ELN < SVM$	0.236
Rastrigin $10D$	RC	5 %	$ELN < SVM$	0.037
Rastrigin $10D$	RC	10 %	$ELN < SVM$	0.192
Rastrigin $10D$	RC	20 %	$ELN < SVM$	0.052

Table V. The frequencies of surrogate-modeling techniques achieving highest accuracy (i.e., based on the value of lowest average RMAE) for all six optimization problems are presented.

Problem	Kriging	SVM	RBFN	KNN	RF	ELN
Ackley $2D$	0/6	4/6	0/6	2/6	0/6	0/6
Branin $2D$	5/6	1/6	0/6	0/6	0/6	0/6
Ackley $5D$	0/6	6/6	0/6	0/6	0/6	0/6
Sphere $5D$	6/6	0/6	0/6	0/6	0/6	0/6
Ackley $10D$	0/6	3/6	0/6	0/6	0/6	3/6

Rastrigin $10D$	$0/6$	$0/6$	$0/6$	$0/6$	$0/6$	$6/6$

Table VI. Final optimal function values for all thirty-six cases by the original model and all the surrogate models. The function values in the table represent the mode of 100 runs alongside the standard deviation, both rounded off to nearest integer representation. In each case, the surrogate models with most optimal function values are highlighted.

Problem	Rob..	NL	Orig..	Krig.	SVM	RBFN	KNN	RF	ELN
Ackley $2D$	RR	5 %	11 ± 3	14 ± 2	22 ± 3	13 ± 2	12 ± 3	14 ± 2	12 ± 3
Ackley $2D$	RR	10 %	16 ± 1	17 ± 1	22 ± 1	18 ± 1	17 ± 1	17 ± 1	16 ± 0
Ackley $2D$	RR	20 %	20 ± 0	21 ± 0	22 ± 0	21 ± 0	21 ± 0	21 ± 0	21 ± 0
Ackley $2D$	RC	5 %	5 ± 7	10 ± 2	22 ± 7	10 ± 2	6 ± 4	10 ± 2	5 ± 5
Ackley $2D$	RC	10 %	7 ± 6	10 ± 2	22 ± 7	11 ± 2	7 ± 4	11 ± 2	7 ± 0
Ackley $2D$	RC	20 %	10 ± 5	12 ± 2	22 ± 5	12 ± 2	11 ± 3	12 ± 2	10 ± 0
Branin $2D$	RR	5 %	3 ± 3	6 ± 64	9 ± 3	6 ± 64	6 ± 52	6 ± 64	17 ± 0
Branin $2D$	RR	10 %	9 ± 4	13 ± 83	16 ± 3	13 ± 83	13 ± 56	13 ± 83	25 ± 0
Branin $2D$	RR	20 %	20 ± 0	20 ± 129	20 ± 0	20 ± 146	20 ± 103	20 ± 146	106 ± 0
Branin $2D$	RC	5 %	1 ± 0	1 ± 52	3 ± 7	1 ± 51	2 ± 43	2 ± 52	12 ± 0
Branin $2D$	RC	10 %	1 ± 1	2 ± 54	4 ± 6	2 ± 54	2 ± 45	2 ± 54	13 ± 0
Branin $2D$	RC	20 %	3 ± 3	4 ± 59	6 ± 4	4 ± 45	3 ± 48	4 ± 59	15 ± 0
Ackley $5D$	RR	5 %	16 ± 1	16 ± 1	12 ± 1	17 ± 1	12 ± 2	17 ± 1	22 ± 0
Ackley $5D$	RR	10 %	14 ± 1	16 ± 1	20 ± 0	16 ± 1	14 ± 1	16 ± 1	22 ± 0
Ackley $5D$	RR	20 %	16 ± 1	18 ± 1	21 ± 0	18 ± 1	14 ± 1	18 ± 1	22 ± 0
Ackley $5D$	RC	5 %	5 ± 5	19 ± 1	12 ± 2	18 ± 1	14 ± 2	19 ± 1	22 ± 0

Ackley $\mathbf{5}\boldsymbol{D}$	RC	$\mathbf{10}$ %	$\mathbf{7 \pm 5}$	$\mathbf{19 \pm 1}$	$\mathbf{9 \pm 2}$	$\mathbf{18 \pm 1}$	$\mathbf{14 \pm 2}$	$\mathbf{19 \pm 1}$	$\mathbf{22 \pm 0}$
Ackley $\mathbf{5}\boldsymbol{D}$	RC	$\mathbf{20}$ %	$\mathbf{10 \pm 4}$	$\mathbf{19 \pm 1}$	$\mathbf{16 \pm 1}$	$\mathbf{22 \pm 3}$	$\mathbf{15 \pm 2}$	$\mathbf{19 \pm 1}$	$\mathbf{22 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RR	$\mathbf{5}$ %	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 7}$	$\mathbf{0 \pm 0}$	$\mathbf{1 \pm 8}$	$\mathbf{0 \pm 12}$	$\mathbf{7 \pm 19}$	$\mathbf{0 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RR	$\mathbf{10}$ %	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 8}$	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 7}$	$\mathbf{1 \pm 12}$	$\mathbf{5 \pm 22}$	$\mathbf{0 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RR	$\mathbf{20}$ %	$\mathbf{0 \pm 0}$	$\mathbf{0 \pm 9}$	$\mathbf{0 \pm 0}$	$\mathbf{1 \pm 16}$	$\mathbf{1 \pm 12}$	$\mathbf{8 \pm 31}$	$\mathbf{0 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RC	$\mathbf{5}$ %	$\mathbf{0 \pm 0}$	$\mathbf{5 \pm 16}$	$\mathbf{0 \pm 0}$	$\mathbf{9 \pm 15}$	$\mathbf{1 \pm 7}$	$\mathbf{9 \pm 18}$	$\mathbf{0 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RC	$\mathbf{10}$ %	$\mathbf{0 \pm 0}$	$\mathbf{9 \pm 16}$	$\mathbf{0 \pm 0}$	$\mathbf{7 \pm 17}$	$\mathbf{1 \pm 8}$	$\mathbf{10 \pm 18}$	$\mathbf{0 \pm 0}$
Sphere $\mathbf{5}\boldsymbol{D}$	RC	$\mathbf{20}$ %	$\mathbf{1 \pm 0}$	$\mathbf{10 \pm 18}$	$\mathbf{1 \pm 0}$	$\mathbf{10 \pm 17}$	$\mathbf{2 \pm 11}$	$\mathbf{11 \pm 18}$	$\mathbf{1 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{5}$ %	$\mathbf{18 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{8 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{8 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{10}$ %	$\mathbf{18 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{8 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{15 \pm 1}$	$\mathbf{20 \pm 0}$	$\mathbf{7 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{20}$ %	$\mathbf{19 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{12 \pm 0}$	$\mathbf{19 \pm 0}$	$\mathbf{17 \pm 1}$	$\mathbf{20 \pm 0}$	$\mathbf{22 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{5}$ %	$\mathbf{5 \pm 5}$	$\mathbf{20 \pm 0}$	$\mathbf{8 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{5 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{10}$ %	$\mathbf{7 \pm 5}$	$\mathbf{20 \pm 0}$	$\mathbf{9 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{20 \pm 0}$	$\mathbf{7 \pm 0}$
Ackley $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{20}$ %	$\mathbf{10 \pm 5}$	$\mathbf{20 \pm 0}$	$\mathbf{11 \pm 0}$	$\mathbf{21 \pm 0}$	$\mathbf{21 \pm 0}$	$\mathbf{21 \pm 0}$	$\mathbf{10 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{5}$ %	$\mathbf{918 \pm 27}$	$\mathbf{1021 \pm 33}$	$\mathbf{1013 \pm 0}$	$\mathbf{1154 \pm 28}$	$\mathbf{975 \pm 41}$	$\mathbf{1020 \pm 34}$	$\mathbf{993 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{10}$ %	$\mathbf{923 \pm 28}$	$\mathbf{1024 \pm 40}$	$\mathbf{987 \pm 0}$	$\mathbf{1083 \pm 20}$	$\mathbf{985 \pm 31}$	$\mathbf{1024 \pm 38}$	$\mathbf{996 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RR	$\mathbf{20}$ %	$\mathbf{919 \pm 40}$	$\mathbf{1038 \pm 54}$	$\mathbf{952 \pm 0}$	$\mathbf{1033 \pm 50}$	$\mathbf{975 \pm 31}$	$\mathbf{1047 \pm 53}$	$\mathbf{1032 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{5}$ %	$\mathbf{946 \pm 23}$	$\mathbf{1020 \pm 29}$	$\mathbf{1034 \pm 0}$	$\mathbf{1164 \pm 14}$	$\mathbf{988 \pm 31}$	$\mathbf{1020 \pm 29}$	$\mathbf{961 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{10}$ %	$\mathbf{995 \pm 9}$	$\mathbf{1042 \pm 26}$	$\mathbf{1005 \pm 0}$	$\mathbf{1134 \pm 27}$	$\mathbf{1013 \pm 20}$	$\mathbf{1041 \pm 27}$	$\mathbf{996 \pm 0}$
Rastrigin $\mathbf{10}\boldsymbol{D}$	RC	$\mathbf{20}$ %	$\mathbf{1003 \pm 12}$	$\mathbf{1058 \pm 24}$	$\mathbf{1045 \pm 0}$	$\mathbf{1041 \pm 23}$	$\mathbf{1023 \pm 25}$	$\mathbf{1064 \pm 25}$	$\mathbf{1055 \pm 0}$

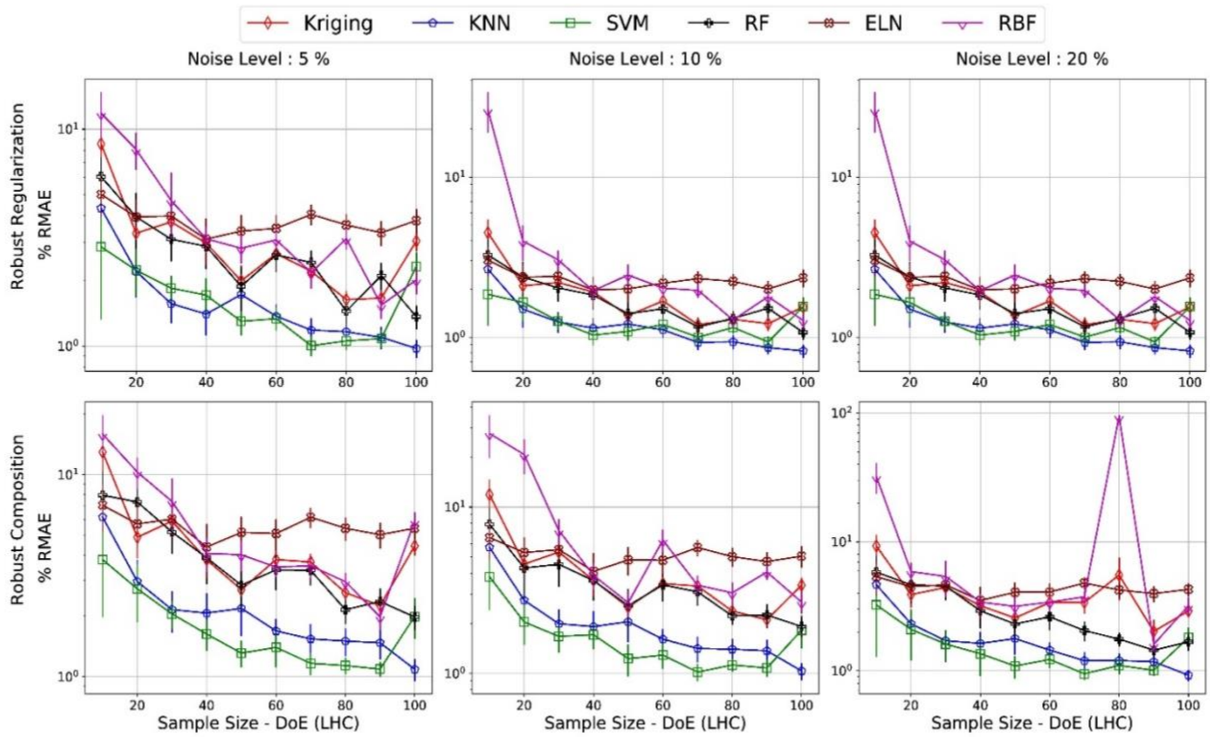

Figure 3. Modeling accuracy of surrogates on Ackley 2D with three noise levels, two robustness strategies and ten training sample sizes.

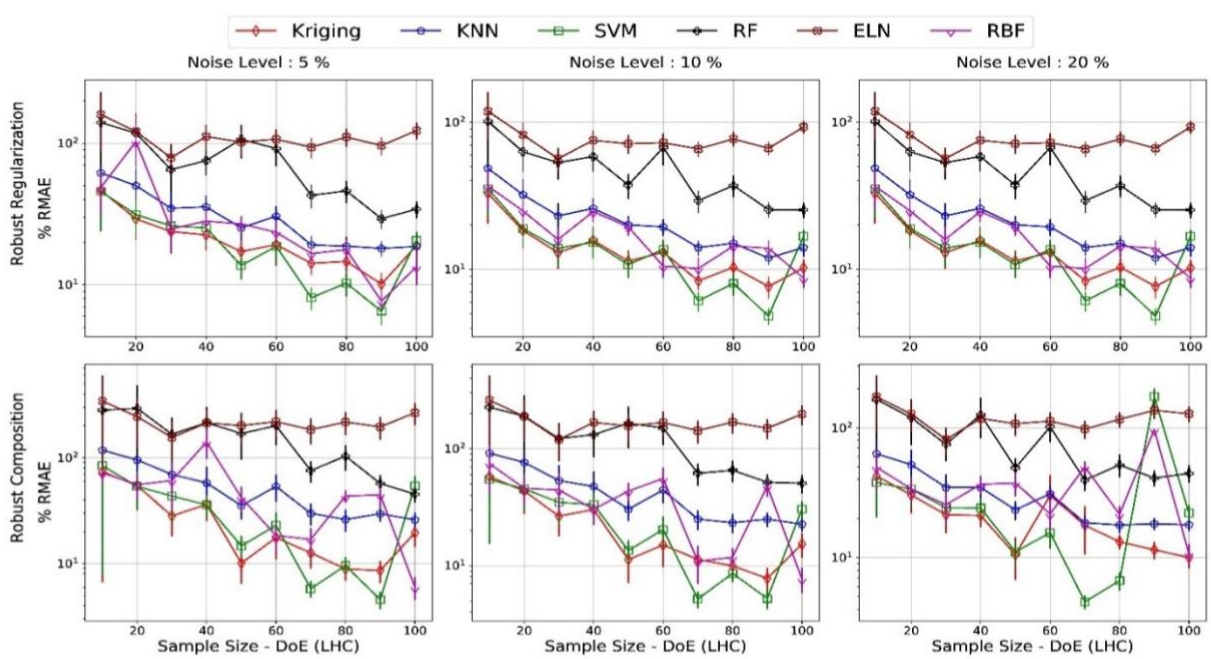

Figure 4. Modeling accuracy of surrogates on Branin 2D with three noise levels, two robustness strategies and ten training sample sizes.

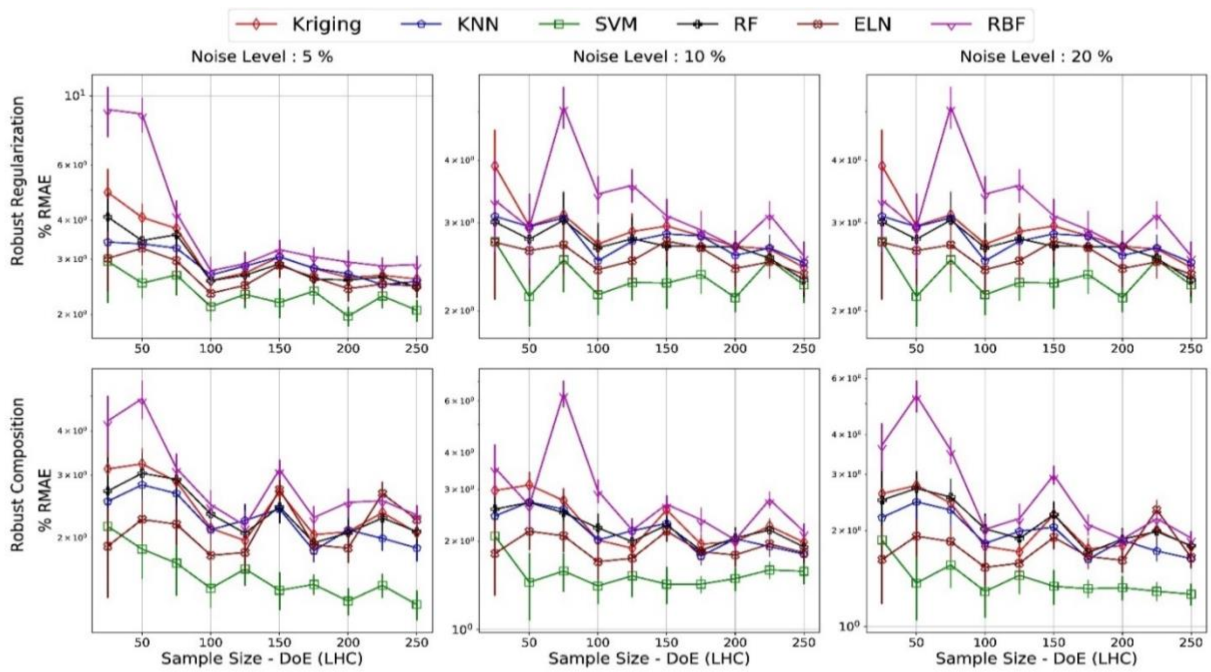

Figure 5. Modeling accuracy of surrogates on Ackley 5D with three noise levels, two robustness strategies and ten training sample sizes.

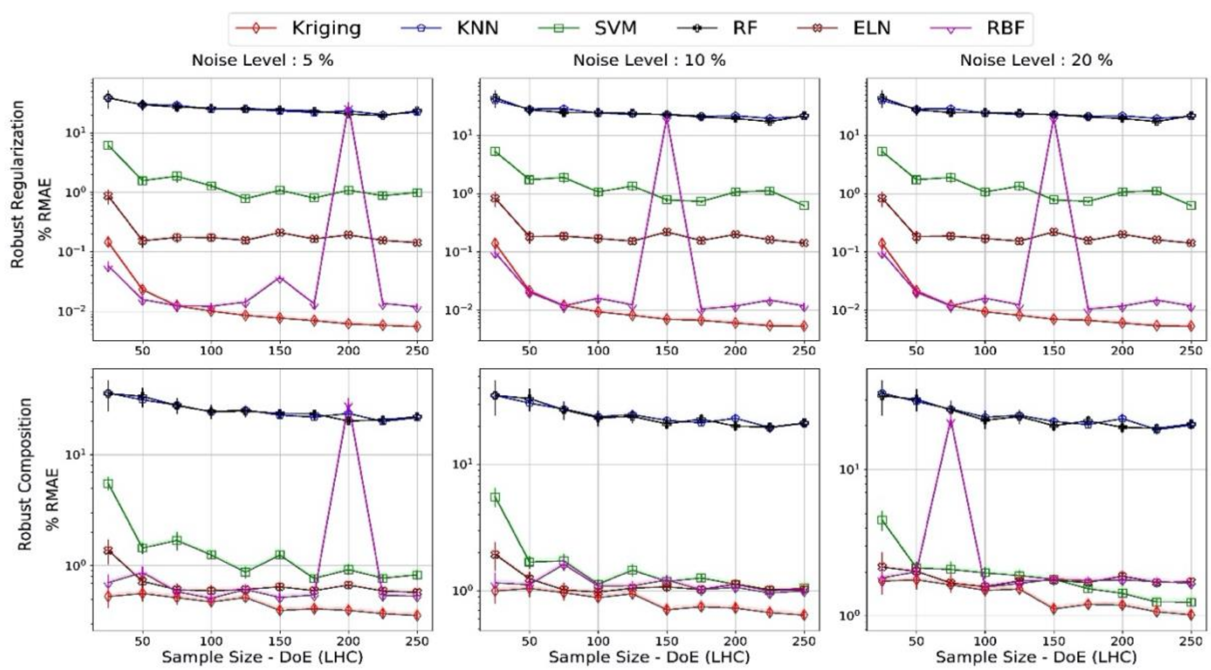

Figure 6. Modeling accuracy of surrogates on Sphere 5D with three noise levels, two robustness strategies and ten training sample sizes.

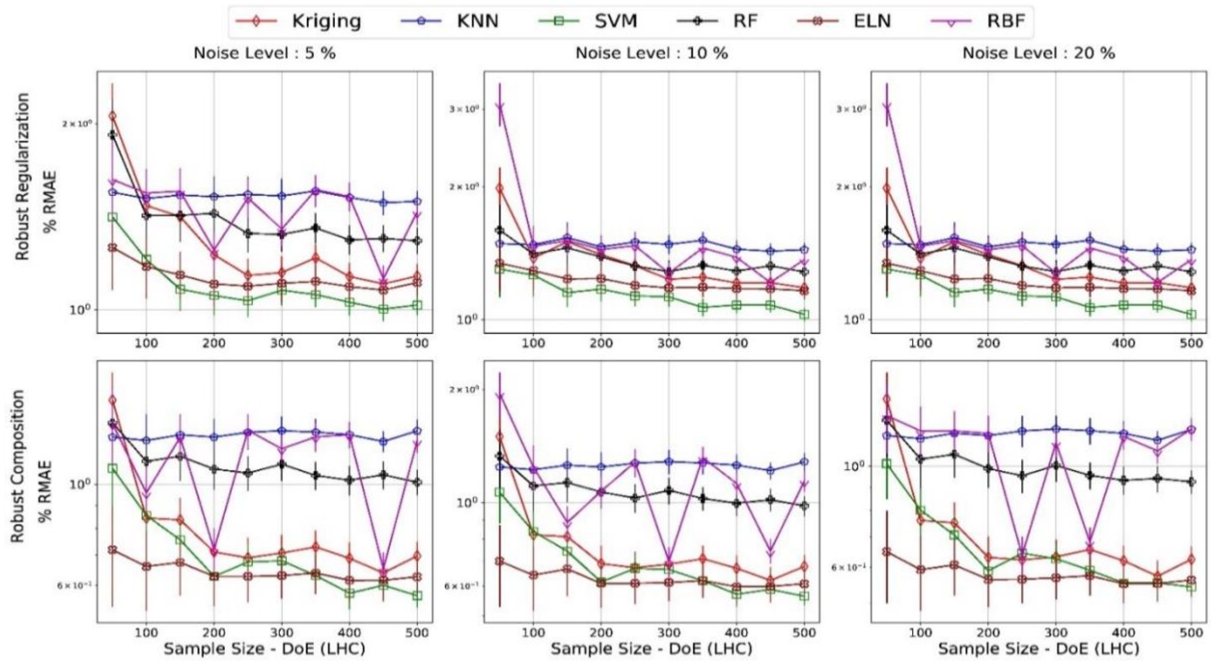

Figure 7. Modeling accuracy of surrogates on Ackley 10D with three noise levels, two robustness strategies and ten training sample sizes.

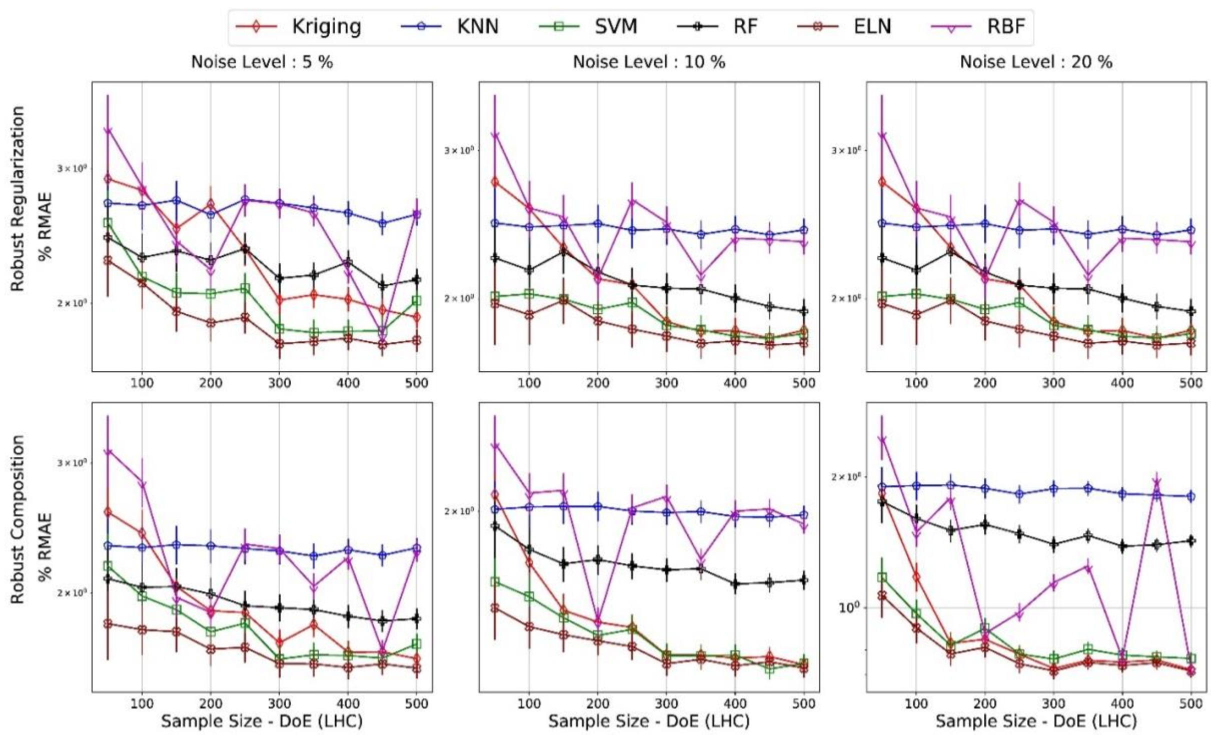

Figure 8. Modeling accuracy of surrogates on Rastrigin 10D with three noise levels, two robustness strategies and ten training sample sizes.

5. Summary & Outlook

This deliverable report focuses on the scientific achievements regarding the work package 2.3 in ECOLE, namely, robustness & uncertainty modelling in experience-based optimization. The importance of handling uncertainty and/or noise within the context of design optimization is palpable since uncertainty and/or noise can greatly determine the practical applicability of the optimal solution(s). Furthermore, uncertainty and/or noise can significantly affect the objective landscape. Sec. 2 of this report provides an overview on different sources of uncertainty and/or noise within the context of black-box design optimization. It further delineates three common ways to mathematically model the uncertainties and/or noise. The final part of Sec. 2 shares different scenarios of optimization under uncertainty, and highlights the cases which are most relevant to the project. Sec. 3 of this report focuses on the RCA, in particular, the robustness based on the minimax principle and the composite robustness. Sec. 4 focuses on SARO, and shares the experimental design and key results from our research in ECOLE.

Major findings in Sec. 4 suggest the usefulness of surrogate modeling for robust design optimization. This is due to the fact that in 8/36 cases reported in Table VI, at least one surrogate achieves the optimal function value $-$ based on the mode of 100 runs and if tied, based on standard deviation. Additionally, in 12/36 cases, at least one surrogate finds a better local optimum of the function following the similar evaluation criteria. In the majority of the remaining cases, at least one surrogate achieves suboptimal function. From the findings in Sec. 4, in particular Tables IV, V and Figs. (3-8), we argue that Kriging, SVMs and ELN provide a high modeling accuracy even with limited training data in most cases. Additionally, these techniques find optimal or suboptimal function values in most cases as presented in Table VI. Our findings are a further validation of Kriging and Polynomial models as two of the most reliable modeling techniques. Our research also suggests SVMs as a promising and competitive modeling technique since it provides the highest modeling accuracy in most cases, i.e., lowest average RMAE in 14/36 as presented in Table V, and proposes suboptimal solutions in most $\boldsymbol{5D}$ and $\boldsymbol{10D}$ cases as illustrated in Table VI.

Future investigations are necessary to validate these findings on more complex problems such as the ones with multiple objectives, higher dimensionalities, asymmetric noise and real-world engineering case studies, e.g., crashworthiness, target shape design optimization. Furthermore, it is important to model multiplicative noise and define other robustness strategies characterizing the practical nature of RO. An important and fundamental scientific question to answer is the cost of robustness. Similarly, a natural extension of the famous Efficient Global Optimization (EGO) framework to the cases with uncertainty and/or noise is desirable. Finally, there is a dire need to produce tutorial based cohesive work for designers to adopt the surrogate models in practical cases of RO.

Bibliography

[1] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering,* vol. 196, no. 33-34, pp. 3190-3218, 2007.

[2] J. W. Kruisselbrink, Evolution strategies for robust optimization, Leiden: Leiden University, 2012.

[3] A. Ben-Tal, L. E. Ghaoui and A. Nemirovski, Robust optimization, Princeton University Press, 2009.

[4] G. Taguchi and M. Phadke, "Quality engineering through design optimization," in *Quality Control, Robust Design, and the Taguchi Method*, Boston, Springer, 1989, pp. 77-96.

[5] F. Jurecka, Robust design optimization based on metamodeling techniques, Munich: Technical University of Munich, 2007.

[6] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?," *Structural Safety*, pp. 105-112, 2009.

[7] H. Griethe and H. Schumann., "Visualizing uncertainty for improved decision making," in *4th International Conference on Business Informatics Research*, 2005.

[8] S. Ullah, H. Wang, S. Menzel, B. Sendhoff and T. Bäck, "An Empirical Comparison of Meta-Modeling Techniques for Robust Design Optimization," in *IEEE Symposium Series on Computational Intelligence*, Xiamen, 2019.

[9] Y. Carson and A. Maria, "Simulation optimization: methods and applications," in *29th conference on Winter simulation*, 1997.

[10] C. M. Bishop, Pattern recognition and machine learning, springer, 2006.

[11] A. Forrester, A. Sobester and A. Keane., Engineering design via surrogate modelling: a practical guide, John Wiley & Sons, 2008.

[12] T. Hastie, R. Tibshirani and J. Friedman, The elements of statistical learning: data mining, inference, and prediction, pringer Science & Business Media, 2009.