



Universiteit
Leiden

The Netherlands

Model-assisted robust optimization for continuous black-box problems

Ullah, S.

Citation

Ullah, S. (2023, September 27). *Model-assisted robust optimization for continuous black-box problems*. Retrieved from <https://hdl.handle.net/1887/3642009>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/3642009>

Note: To cite this publication please use the final published version (if applicable).

Engineering Applications

So far in this thesis, we have focused on two different but related research streams. The first one (Chapter 3) deals with the applicability of surrogate modeling to find robust solutions. The manifestation of surrogate modeling focused in this research stream is based on “one-shot optimization” strategy (Ullah et al., 2019). In this research stream, we emphasize on the fundamental questions regarding the applicability of surrogate modeling in robust optimization, and the related difficulties thereof, e.g., high dimensionality (Ullah et al., 2020a).

The second research stream (Chapters 4 and 5) emphasizes on the applicability of the Bayesian optimization algorithm, and the related difficulties thereof (Ullah et al., 2021). As part of the second research stream, we made an attempt to find a suitable robustness criterion in practical scenarios with regards to the associated computational cost (Ullah et al., 2022). We are now interested in benchmarking our earlier findings for both research streams on a real-world engineering application.

To this end, we consider a benchmark engineering case study based on the design of car hood frames. The associated data set contains over 10,000 3D mesh geometries for variants of car hood frames. This data set is generated through an automated, industry-grade Computer Aided Design (CAD) workflow, described in (Ramnath et al., 2022), and further benchmarked in (Wollstadt et al., 2022). The data set provides realistic designs of car hood frames, which were validated by experts with respect to realism, manufacturability, variability, and performance. Each geometry is described by a subset of design variables, such as cut-outs and ribs on the hood frame as well as their properties, for example, rib location and height, or cut-out location.

Starting in Section 6.1, we provide an overview on the case study with the most important details, such as the description of the data set, the data pre-processing, and the targeted tasks. In Section 6.2, we apply the data set to validate our findings for the first research stream. This is followed by benchmarking the performance of the Bayesian optimization algorithm on the data set. Lastly, we provide a short summary and discussion on the results.

6.1 Car Hood Design

We consider a case study inspired from a real-world design optimization scenario (Ramnath et al., 2022), where the aim is to optimize the design of a car hood frame with respect to three performance metrics. These performance metrics are the “Geometry Mass (kg)”, “Directional Deformation Maximum (mm)”, and “Equivalent Stress Maximum (MPa)”. Each geometry is represented as a surface mesh (STL file), and is described by a subset of 38 design variables, such as “Rear Rib Depth”, “Rear Rib Offset”, “Pocket Offset”, and “Front Curve Height” among others.

The data set in this context was generated using a feature-based modeling approach (Ramnath et al., 2019). In the context of automotive car hoods, features describe components that contribute to desirable properties of the design, e.g., ribs to add stiffness during driving or impact, or cut-outs and pockets to reduce overall weight. Real car hood designs were simplified by removing features that were irrelevant for the hood’s performance. Remaining features were created independent of the base surface to allow for the generation of a sufficiently large variety of hoods by combining features and feature patterns with a set of 100 base geometries. Features were parameterized and generated using an automated workflow in computer-aided three-dimensional interactive application (CATIA) v5 (König and Wintermantel, 2004). It is important to note that some parametrizations led to invalid geometries, such that in total 10,478 unique hood geometry files were generated. CAD models were converted to watertight STL surface meshes in the STL format (Ramnath et al., 2022).

For each car hood, structural mechanics performance values were simulated using finite element analysis (FEA) (Szabó and Babuška, 2021). FEA was performed for a hood lift load case under driving conditions, which is an important structural requirement when designing car hood frames (Vyas et al., 2020). The obtained

6. ENGINEERING APPLICATIONS



Figure 6.1: Example of variability in car hood designs in the data set. The variability is due to the introduction of features, e.g., pockets, cut outs, and ribs.

performance values are “Maximum Equivalent Stress (MPa)” and “Maximum Directional Deformation (mm)”. Additionally the “Geometry Mass (kg)” is provided for each design. FEA was performed using a standardized setup over all geometries to allow for automated generation of simulation results. Fig. 6.1 illustrates the variability in geometries considered in our case study.

6.1.1 Data Set

We start with the data set provided by Ramnath et al. (Ramnath et al., 2022), which includes geometries, i.e., surface meshes, for 10,070 different designs for car hood frames. Each geometry contains values of the design variables, e.g., “Rear Rib Depth”, used to run the FEA simulation for assessing the performance of the corresponding design with respect to structural mechanics indicators, e.g., “Maximum Equivalent Stress (MPa)” and “Maximum Directional Deformation (mm)”. Note that the entire data set contains 38 unique design variables, but each geometry is accompanied with a subset of these variables. The average number of design variables per geometry is found to be 14, whereas the maximum number of design variables is found to be 27.

6.1.2 Data Wrangling

Since the geometries in the data set have irregular and asynchronous design schema¹, our first task is to identify the most common design variables. For this purpose, we count the frequency of each design variable in the entire data set, and select the top five most commonly appearing design variables. They are “Rib Depth”, “Rear Rib Width”, “Rear Rib Offset”, “Rear Rib Depth”, and “Rear Rib

¹Irregular in this context refers to the fact that geometries are provided with varying number of design variables. Asynchronous refers to the fact that not all design variables are present in each geometry (Ullah et al., 2020b)

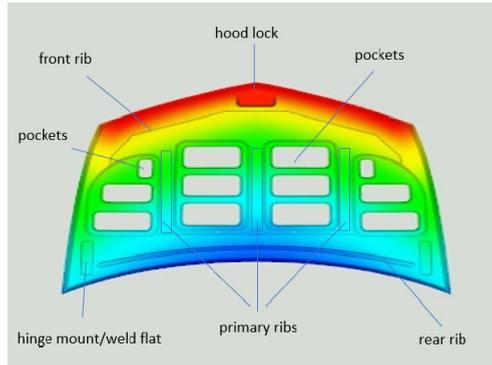


Figure 6.2: Some of the most commonly appearing features considered in the optimization of car hood design. The design variables considered in our study indicate the properties for these features, e.g., “Rear Rib Depth” and “Pocket Offset”.

End Point Y”. We then scan the entire data set to identify designs where these five variables appear together. As a result, we are left with 1176 geometries where these five variables appear together. We then extend this data set by scanning these 1176 designs so as to search for other design variables, which may be present in all of these 1176 geometries. This increases the number of design variables to 18.

We are interested in benchmarking our previous findings on a real-world engineering case study. For this purpose, we have to formulate optimization scenarios with three settings of dimensionality as: $\mathcal{D} = \{2, 5, 10\}$. This means we have to select two, five, and ten design variables among the set of available design variables. For this purpose, we construct a benchmark Kriging surrogate model (Rasmussen and Williams, 2006) with all 18 variables and 1176 training instances, for all three performance indicators: Mass (kg), Deformation (mm), and Stress (MPa). Then, we remove each one of the 18 variables in the model, and see the potential impact on the accuracy of the model. Based on this, we rank all 18 variables for all three performance indicators, and select the top two, five, and ten variables that have the most significant effect on the model accuracy (Fan, 2007). These variables for all three performance indicators are presented in Table 6.1. Furthermore, we present an example of some of the most important features for car hood designs in Fig. 6.2 for further clarification.

6. ENGINEERING APPLICATIONS

Table 6.1: A summary of the selected design variables and tasks to be performed. “Performance Indicators” indicates the three output variables (tasks), which are to be minimized in optimization. “Variables” indicates the design variables which are included for a particular choice of task and dimensionality (based on the data wrangling discussed earlier). The abbreviations for these design variables are presented in Table 6.2.

Performance Indicators	Dimensionality	Variables
Mass (kg)	2	“RRO”, “RRD”
	5	“RRO”, “RRD”
		“P1O”, “P2O”
		“P3O”
	10	“RRO”, “RRD”
		“P1O”, “P2O”
		“P3O”, “RREPY”
“P2R”, “P3R”		
“P4O”, “SRW”		
Deformation (mm)	2	“RRO”, “RCH”
	5	“RRO”, “RCH”
		“RREPY”, “P4O”
		“RRW”
	10	“RRO”, “RCH”
		“PREPY”, “P4O”
		“RRW”, “1SRL”
“P2O”, “P3O”		
“ARW”, “P3R”		
Stress (MPa)	2	“RRO”, “ARW”
	5	“RRO”, “ARW”
		“RCH”, “P4O”
		“SRW”
	10	“RRO”, “ARW”
		“RCH”, “P4O”
		“SRW”, “P2O”
“P3O”, “RRD”		
“1SRL”, “2SRL”		

Table 6.2: Abbreviations of the 18 design variables discussed in Section 6.1.2. Some of these variables are presented in Table 6.1 to formulate the optimization tasks with three settings of the dimensionality.

Design Variable	Abbreviation	Range
“1SRL”	“1st Subsidiary Rib Length”	[140, 200]
“2SRL”	“2nd Subsidiary Rib Length”	[120, 240]
“ARW”	“Angled Rib Width”	[30, 180]
“MRW”	“Middle Rib Width”	[40, 200]
“P1O”	“Pocket1 Offset”	[0, 10]
“P1R”	“Pocket1 Radius”	[18, 35]
“P2O”	“Pocket2 Offset”	[0, 10]
“P2R”	“Pocket2 Radius”	[18, 45]
“P3O”	“Pocket3 Offset”	[0, 10]
“P3R”	“Pocket3 Radius”	[13, 45]
“P4O”	“Pocket4 Offset”	[0, 10]
“RCH”	“Rear Curve Height”	[20, 120]
“RRD”	“Rear Rib Depth”	[8, 14]
“RREPY”	“Rear Rib End Point Y”	[400, 620]
“RRO”	“Rear Rib Offset”	[-50, 10]
“RRW”	“Rear Rib Width”	[20, 30]
“RD”	“Rib Depth”	[15, 30]
“SRW”	“Subsidiary Rib Width”	[25, 50]

6.1.3 Tasks

We are interesting in design optimization scenarios with three performance indicators illustrated in Table 6.1. For each one of the indicators, we consider three settings of dimensionality as described earlier. This gives rise to a total of 9 optimization tasks. Furthermore, we consider two different goals of robust optimization for these 9 tasks. These two goals emphasize on benchmarking “one-shot optimization” strategy (Chapter 3) and Bayesian optimization algorithm (Chapter 4) (Ullah et al., 2019, 2021).

It is pertinent to mention that we try to maintain the same experimental setup, wherever possible, for these two goals, as discussed previously in the thesis (Chapter 3 and Chapter 4), to account for fairness. Nonetheless, the nature of the

6. ENGINEERING APPLICATIONS

industrial data, as well as the realism, manufacturability, and variability associated with the data generation process means we might have to compromise on some settings of our previous experimental setups. This would be explained in further details in experimental setup, wherever applicable.

6.2 One-shot Optimization

We begin with the goal of benchmarking the “one-shot optimization” strategy based on our data set. This refers to the fact that we consider a surrogate model, which after construction, is directly utilized by a benchmark numerical optimization algorithm, e.g., L-BFGS-B (Wright et al., 1999), without any adaptive sampling, i.e., updating the surrogate model (Ta’asan et al., 1992). This strategy has been explained in detail in Chapter 2 (cf. Fig. 2.2). This goal has two objectives: evaluating the surrogate model based on the modeling accuracy, and the quality of the optimal solution obtained from surrogate modeling. Experimental setup as well as results for both of these manifestations are provided in the following.

6.2.1 Experimental Setup

In the context of modeling accuracy, our aim is to determine which of the popular modeling techniques (Bishop, 2007) is most suitable to model the objective function effectively. In this experimental setup, we consider five modeling techniques: Kriging, Polynomial (degree=2), K Nearest-Neighbour (KNN), Random Forest (RF), and Support Vector Machines (SVMs). Furthermore, for each one of these techniques, we consider ten different sample sizes as: $K \times D$, where D refers to the corresponding setting of the dimensionality, and $K \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. This gives rise to a total of 450 test cases, owing to the unique combinations of 9 optimization tasks, 10 different sample sizes, and 5 modeling techniques. For each one of these cases, we measure the modeling accuracy according to the RMAE criterion introduced earlier (cf. Eq. (3.11)).

In terms of data pre-processing, we first extract the information about the quantity of interest, e.g., Mass (kg), as per the task, and the corresponding design variables, e.g., “RRD”, according to the setting of the dimensionality. We then identify the duplicates in the newly formed data set and remove them. Following this, we construct a design for each setting of the sample size according to the LHS scheme (Montgomery, 2017). This, however, poses a practical problem, since we

do not have direct access to the FEA simulation, but rather a pre-computed evaluation of the FEA simulation for the corresponding quantity of interest. Therefore, for each location in the LHS design, we look for the nearest pre-computed evaluation available. The nearest evaluation is identified based on cosine similarity of the design variables. In this way, we sample the search space according to the LHS scheme based on the nearest available point. Note, however, that, this might also give rise to duplicates, since a pre-computed evaluation could be nearest to more than one location retrieved from LHS. In this case, we do not allow a duplicate, and rather select the second nearest point available point, from the data set. After the generation of the training data for a particular choice of sample size, we look for testing data points in the remaining data set. These data points are randomly selected based on a size, which is half of the training data size. We then feed the corresponding training and testing data set to all five modeling techniques, and report the RMAE.

In the context of the quality of the optimal solutions, we consider 180 test cases, owing to the combinations of 5 modeling techniques described above, 9 optimization tasks discussed, 2 noise levels, and 2 robustness formulations. An important thing to note in this context is that all design variables take integer values. Therefore, we employ the *Mixed-Integer Surrogate Models*, where-ever possible, to find robust solutions, similar to the approach by (Garrido-Merchán and Hernández-Lobato, 2020)¹. The two noise levels in this context characterize 0.5 and 1 % (max) deviation in the nominal values of the design variables as: $\mathcal{L} = \{0.005, 0.01\}$, whereas the two robustness formulation considered are MMR and CR.

For the deterministic setting of the uncertainty, i.e., MMR, the compact uncertainty set U is defined as: $U = [-(L \times R), (L \times R)]$, where $L \in \mathcal{L}$ denotes the choice of the noise level, and R serves as the absolute range of the search variables provided in Table 6.2. Note that in this context, the uncertainty set U is a subset of integer values: $U \subseteq \mathbb{Z}$, since all design variables take integer values². For

¹Only the Kriging and Polynomials are transformed to “Mixed-Integer Surrogate Models” in this context since current implementations do not allow other modeling techniques to be extended.

²It is not difficult to verify that the internal optimization loop of the MMR in this context refers to the complete enumeration over a full factorial design of all unique noise combinations. Hence, the noise levels in this experimental setup are significantly reduced to be 0.5 and 1 % respectively, as opposed to the 5 and 10 % (and sometimes even 20 %), discussed previously in the thesis. Increasing the noise levels to 5 and 10 % makes solving the problem infeasible since the size of the full factorial design increases rapidly with each new level.

6. ENGINEERING APPLICATIONS

the probabilistic setting of the uncertainty, i.e., CR, the uncertainty is modeled according to a discrete uniform probability distribution: $\Delta_{\mathbf{x}} \sim \mathcal{U}(a, b)$, where the boundaries a and b are defined similar to the boundaries of the the set U in the deterministic case.

The sample size is set to be $50 \times D$, and the resulting surrogate model each time is directly utilized to find robust solution according to the noise level and the robustness formulation chosen. For the purpose of data generation, we utilize the same procedure applied for modeling accuracy. Moreover, the numerical optimization algorithm employed to find robust solution is L-BFGS-B (Morales and Nocedal, 2011).

6.2.2 Results

Graphs showing the quality of the surrogate models, based on the criterion of RMAE (lower is better), and by varying the training sample size, are presented in Figs. 6.3 – 6.5. Each figure contains three subplots corresponding to three settings of the dimensionality, whereas each subplot indicates the RMAE values for five modeling techniques and ten sample sizes, i.e., S1 – S10. In particular, Fig. 6.3 illustrates the quality in this context for predicting “Mass (kg)”. Fig. 6.4 shows the quality regarding “Deformation (mm)”, whereas Fig. 6.5 indicates the quality for “Stress (MPa)”.

Similar to RMAE, the difference in the quality of the optimal solution, obtained from OSO strategy, to that of the baseline (\mathcal{DQ} cf. Eq. (3.12)), is presented in Figs. 6.6 – 6.8. Similar to modeling accuracy, each figure in this context also contains three subplots corresponding to three settings of the dimensionality, whereas each subplot indicates the \mathcal{DQ} values (lower is better) for five modeling techniques, two noise levels, i.e., NL 1 and 2, and two robustness formulations, i.e., MMR and CR. Note that in this context, \mathcal{DQ} values are computed based on objective function values. The baseline values are computed by solving the corresponding optimization problem – with corresponding settings of task, noise level, robustness formulation, and modeling technique – with a baseline surrogate model, which is trained with all 18 design variables for all car hood designs available. In the following, we report the major findings from our investigation.

- **Sample Size**

6.2 One-shot Optimization

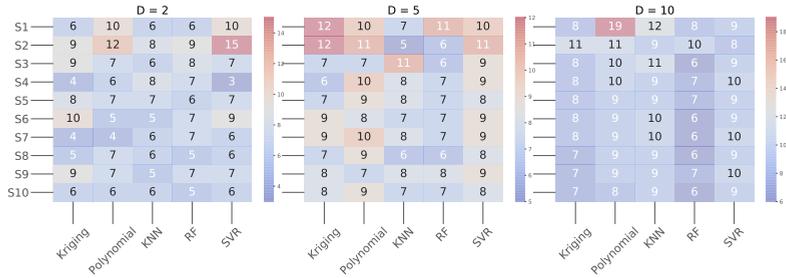


Figure 6.3: Quality of the surrogate models for all five modeling techniques, and ten sample sizes, i.e., S1 – S10, evaluated based on the criterion of RMAE (lower is better). The surrogate models are trained to predict “Mass (kg)” of the car hood designs.

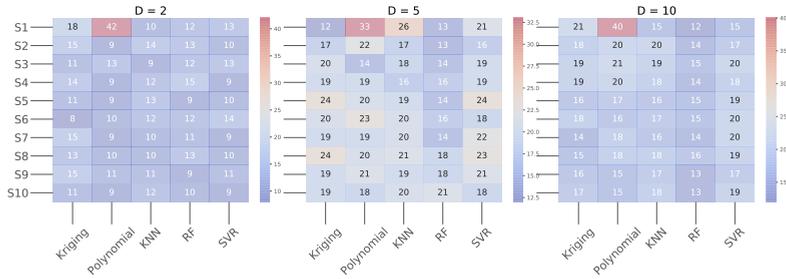


Figure 6.4: Quality of the surrogate models for all five modeling techniques, and ten sample sizes, i.e., S1 – S10, evaluated based on the criterion of RMAE (lower is better). The surrogate models are trained to predict “Maximum Directional Deformation (mm)” of the car hood designs.

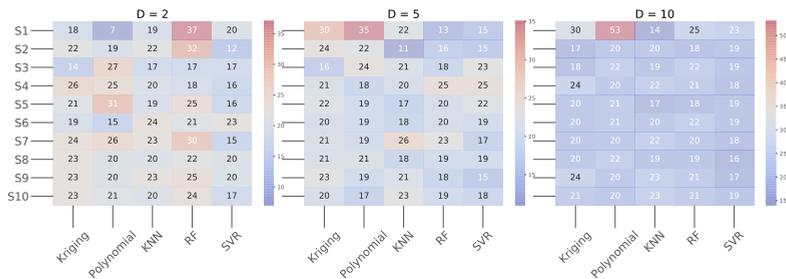


Figure 6.5: Quality of the surrogate models for all five modeling techniques, and ten sample sizes, i.e., S1 – S10, evaluated based on the criterion of RMAE (lower is better). The surrogate models are trained to predict “Maximum Equivalent Stress (MPa)” for the car hood designs.

6. ENGINEERING APPLICATIONS

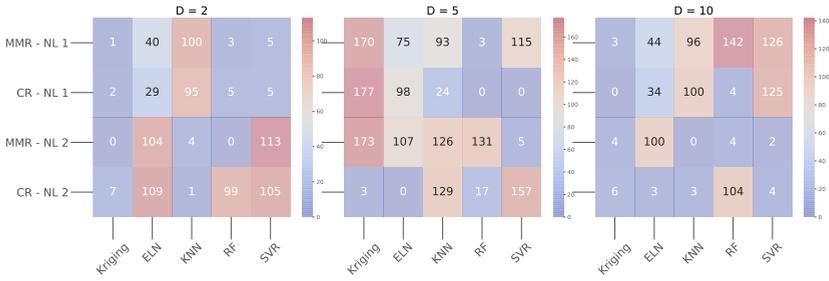


Figure 6.6: Quality of the surrogate models for all five modeling techniques, and for all four cases based on the combinations of two noise levels, i.e., NL 1 and 2, and two robustness formulations, i.e., MMR and CR. The quality is measured according to the criterion of \mathcal{DQ} (lower is better), introduced in Chapter 3. The goal of optimization in this context is to minimize the “Mass (kg)” of the geometry.

An analysis of the averaged results in Figs. 6.3 – 6.5 w.r.t. sample size indicates that we can achieve a good approximation quality with reasonable sample size in most test scenarios. Increasing the sample size does not strictly increase the approximation quality of the surrogate model. However, the surrogate models with the highest number of training points, i.e., S10, usually produce one of the best averaged results. In a loosely speaking manner, our observations re-affirm the generally employed heuristic in model-assisted optimization, which states that the initial sample size can be set linearly in terms of dimensionality (Forrester et al., 2008; Jurecka, 2007).

- **Modeling Technique**

An analysis of the averaged results in Figs. 6.3 – 6.5 w.r.t. modeling techniques indicates that, in general, all five modeling techniques, produce good approximations, for most test scenarios. RF produces the best averaged result in terms of the first and second tasks, i.e., “Mass (kg)” and “Deformation (mm)”, whereas KNN performs best in terms of the third task, i.e., “Stress (MPa)”. This gives us a new perspective of considering RF and KNN as well, when modeling the real-world complex objective functions.

In terms of the averaged results in Figs. 6.6 – 6.8 w.r.t. modeling techniques, we find that the optimal solutions obtained from RF achieve the highest quality for the first task, whereas Kriging produces the best solutions in terms of the second task. Optimal solutions obtained from KNN perform

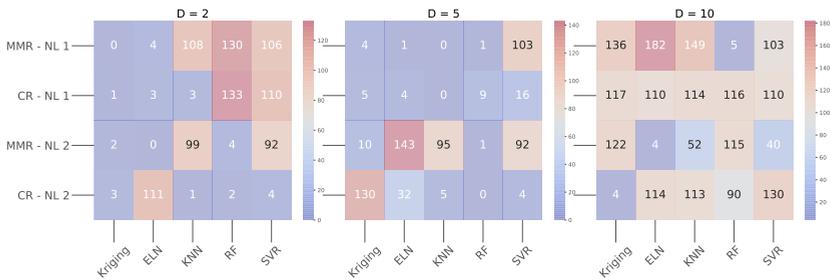


Figure 6.7: Quality of the surrogate models for all five modeling techniques, and for all four cases based on the combinations of two noise levels, i.e., NL 1 and 2, and two robustness formulations, i.e., MMR and CR. The quality is measured according to the criterion of \mathcal{DQ} (lower is better). The goal of optimization in this context is to minimize the “Maximum Directional Deformation (mm)” of the car hood design.

the best in terms of the third task. Overall, in terms of the quality of the optimal solutions, we conclude Kriging produces excellent results in most test scenarios. An important thing to note here is that we do not achieve the higher quality expected from polynomial surrogates.

- **Applicability**

Based on the overall performance of the surrogate models in terms of modeling accuracy, and quality of the robust optimal solutions, we deem surrogate modeling to be applicable for efficiently solving optimization problems under uncertainty. This is due to the fact that in most cases, the quality of the approximation obtained from Kriging, SVM, RF and KNN is good enough to employ a surrogate to find robust solution. The quality of the optimal solutions in most cases is also satisfactory, since the optimal function value found on the model surface is close to the baseline/ground truth in most cases.

6.3 Bayesian Optimization

We are interested in benchmarking the performance of the Bayesian optimization algorithm (cf. Alg. 1), which is based on the “sequential model-based optimization” framework, to find the optimal solutions in an efficient manner (Jones et al., 1998). In Chapter 4, we extended the Bayesian optimization algorithm to the robust

6. ENGINEERING APPLICATIONS

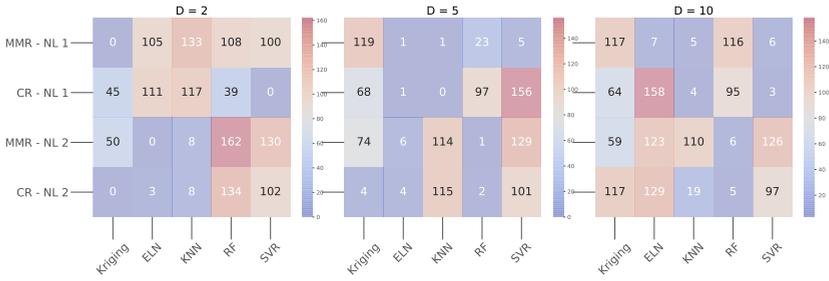


Figure 6.8: Quality of the surrogate models for all five modeling techniques, and for all four cases based on the combinations of two noise levels, i.e., NL 1 and 2, and two robustness formulations, i.e., MMR and CR. The quality is measured according to the criterion of \mathcal{DQ} (lower is better). The goal of optimization in this context is to minimize the “Maximum Equivalent Stress (MPa)” of the car hood design.

scenario, in order to efficiently find the robust solutions (Ullah et al., 2021). We now emphasize on the performance of the Bayesian optimization algorithm, as well as the choice of the sampling infill criterion. To this end, we study three sampling infill criterion: LCB, EIC, and MGFI, which have been introduced earlier (cf. Chapter 4) in the thesis.

6.3.1 Experimental Setup

We start with 9 optimization tasks, introduced earlier in this Chapter. These 9 optimization tasks refer to the minimization of three structural mechanics performance indicators: “Mass (kg)”, “Maximum Directional Deformation (mm)”, and “Maximum Equivalent Stress (MPa)”, each for three settings of the dimensionality as: $\mathcal{D} = \{2, 5, 10\}$. Furthermore, we consider two levels of additive noise as: $\mathcal{L} = \{0.005, 0.01\}$, and two robustness formulations: MMR and CR, respectively. In addition, we consider three sampling infill criteria for the Bayesian optimization algorithm: LCB, EIC, and MGFI, respectively. This gives rise to a total of 108 test cases, owing to the unique combinations of 9 optimization tasks, 2 noise levels, 2 robustness formulations and 3 sampling infill criteria.

In our study, the size of the initial training data is set to be $5 \times D$, where $D \in \mathcal{D}$ denotes the corresponding setting of the dimensionality. Likewise, the maximum number of iterations for Bayesian optimization is set to be $30 \times D$. Note that our Kriging surrogate is based on the “absolute exponential” kernel (Rasmussen and Williams, 2006), and we standardize the function responses: $\mathbf{y} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots,$

$f(\mathbf{x}_N)]^\top$, before constructing the Kriging surrogate \mathcal{K}_f . Furthermore, the Kriging surrogate is based on the implementation of (Garrido-Merchán and Hernández-Lobato, 2020), which transforms the model to handle variables that take integer values. The hyper-parameters β and t for LCB and MGFI are set similar to the setup described in Section 4.3.

For the parallel execution of Bayesian optimization for each of the 108 test cases considered, we utilize Das-5 (Bal et al., 2016), where each standard node has a dual 8-core 2.4 GHz (Intel Haswell E5-2630-v3) cpu configuration and 64 GB memory. We implement our experiments in python 3.7.0 with the help of scikit-learn module (Pedregosa et al., 2011). The performance assessment of the robust solutions in our experiments is based on 15 independent runs \mathcal{R} of the Bayesian optimization algorithm for each of the 108 test cases considered. Note that for each trial, i.e., the unique combination of the independent run and the test case, we ensure the same configuration of hardware and software to account for fairness. Furthermore, in each trial, we measure the cpu time for all iterations of the algorithm to measure the efficiency.

After the successful parallel execution of all trials, we evaluate the performance of our robust solutions based on quality difference \mathcal{DQ} from the baseline (cf. Eq. (3.12))). Note that \mathcal{DQ} in this case is based on the space of objective function values¹. After this, we perform six different analyses to answer the questions outlined earlier. The first two type of analyses are referred to as the fixed cpu time analysis, and the fixed iteration analysis respectively. In fixed cpu time analysis, we fix 50 different settings of the cpu time, and report the best \mathcal{DQ} (the lowest) for each trial. The \mathcal{DQ} in this context is averaged over all 50 settings of the cpu time. For fixed iteration analysis, we fix 30 different settings of the iterations (checkpoints) to report the best \mathcal{DQ} (the lowest) for each trial. The \mathcal{DQ} in this context is also averaged over all 30 checkpoints.

After fixed cpu time and fixed iteration analysis, we perform a fixed target analysis. The fixed target analysis is also based on two different settings: by fixing a target \mathcal{DQ} and reporting the cpu time as well as the number of iterations taken to reach that target. We fix ten different settings for the target in this context, and the corresponding cpu time and iterations are averaged over these target values. Note that each target describes the minimum desirable quality threshold of the robust

¹In this study, we do not divide \mathcal{DQ} with the number of independent runs \mathcal{R} as Eq. (3.12) suggests, but rather report all trials.

6. ENGINEERING APPLICATIONS

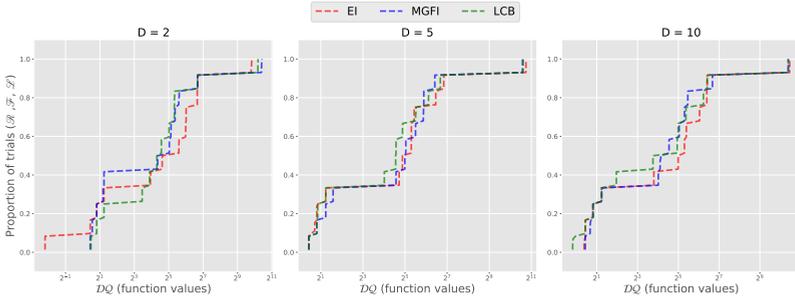


Figure 6.9: Fixed cpu time analysis for car hood design optimization. Each subplot contains 3 Ecdf curves based on 3 infill criteria discussed. Each Ecdf curve is based on 180 data points owing to the combinations of 6 optimization tasks \mathcal{F} (3 optimization scenarios and 2 robustness criteria), 2 noise levels \mathcal{L} , and 15 independent runs \mathcal{R} .

solution. If such a quality is never achieved, we report the penalized cpu time and penalized number of iterations respectively. The penalized cpu time is set to be $D \times T_{\max}$, whereas penalized number of iterations is set to be $D \times N_{\max}$. Here D is the corresponding setting of the dimensionality, and N_{\max} and T_{\max} indicate the maximum number of iterations of the BO algorithm and the cpu time taken to execute it. After the fixed budget and fixed target analyses, we also report the average cpu time per iteration for the BO algorithm. In addition, we also report T_{\max} : the accumulated cpu time at the last iteration of the BO algorithm, for each trial.

6.3.2 Results

The results originating from this are shown in Figs. 6.9 – 6.13. Each of these figures contains the graphs for a particular type of analysis. In particular, Fig. 6.9 shares the results based on a fixed cpu time analysis. The figure contains 3 different subplots corresponding to 3 different settings of the dimensionality. Each subplot shares the empirical cumulative distribution function (ecdf) of DQ for 3 different sampling infill criteria considered.

Likewise, Fig. 6.10 shares the ecdf plots corresponding to fixed iteration analysis, whereas the analysis based on fixed targets is presented in Fig. 6.11. The average cpu time per iteration of the BO algorithm to find robust solutions is presented

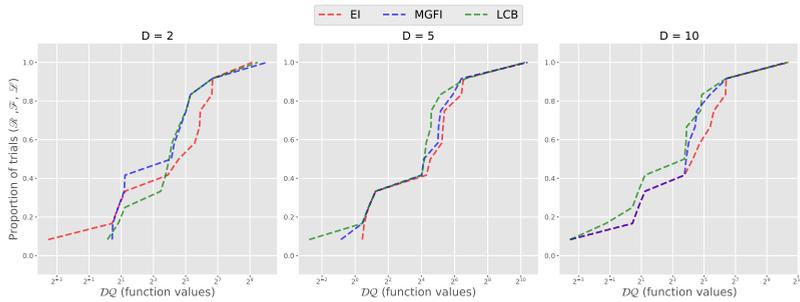


Figure 6.10: Fixed Iteration analysis. Each subplot contains 3 Ecdf curves based on 3 infill criteria discussed. Each Ecdf curve is based on 180 data points owing to the combinations of 6 optimization tasks \mathcal{F} (due to 2 optimization scenarios and 2 robustness criteria), 2 noise levels \mathcal{L} , and 15 independent runs \mathcal{R} .

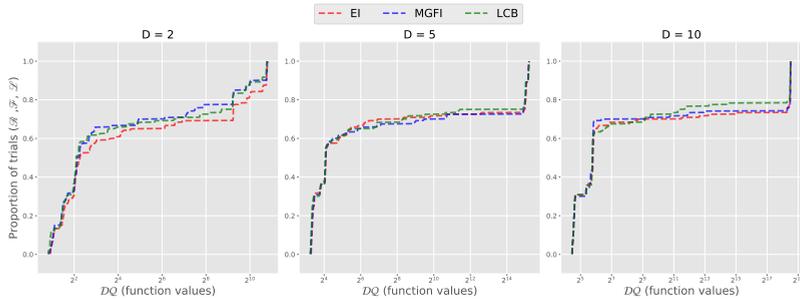


Figure 6.11: Fixed Target analysis. Each subplot contains 3 Ecdf curves based on 3 infill criteria discussed. Each Ecdf curve is based on 180 data points owing to the combinations of 6 optimization tasks \mathcal{F} (3 optimization scenarios and 2 robustness criteria), 2 noise levels \mathcal{L} , and 15 independent runs \mathcal{R} .

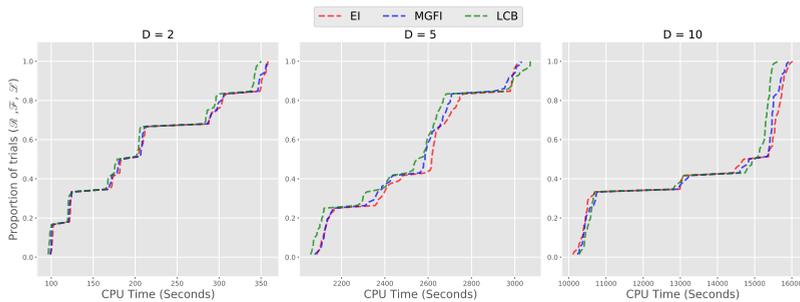


Figure 6.12: Average cpu time per iteration for the BO algorithm. Each subplot contains 3 Ecdf curves based on 3 infill criteria discussed.

6. ENGINEERING APPLICATIONS

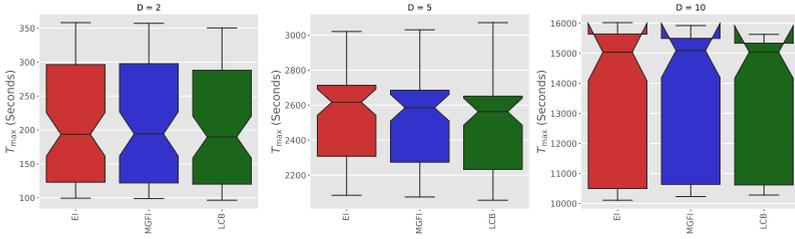


Figure 6.13: Maximum accumulated cpu time: T_{\max} for each trial, for each setting of the dimensionality.

in Fig. 6.12. Lastly, we present the maximum accumulated cpu time: T_{\max} , for each trial in the form of box plots in Fig. 6.13.

In the following, we report the major findings of these results.

- **Applicability of the Bayesian Optimization**

Based on the results presented in Figs. 6.9 – 6.10, we deem Bayesian optimization as a promising heuristic to efficiently find robust solutions in practical scenarios. This is due to the fact that the empirical success rate of the Bayesian optimization algorithm is satisfactory. For instance, if we cut-off the DQ values at 8, the empirical success rate is between 35-40 %.

- **Factors with Significant Influence**

Based on the results presented in Figs. 6.9 – 6.11, we find that dimensionality affects the quality of the robust solutions. Because of the dimensionality, the computational budget, i.e., whether measured in cpu time or number of iterations, also affects the quality of the robust solutions in a significant manner. For instance, in Fig. 6.9, we see that the empirical success measured at 2^4 seconds (cpu time) is more than 40 % for trials belonging to two-dimensional problems. On the other hand, the empirical success rate drops to under 35 % when dimensionality is increased (for five and ten-dimensional cases).

- **Impact of Infill Criterion**

In the context of fixed budget analyses, the performance of LCB_{eff} and $\mathcal{M}_{\text{eff}}(\mathbf{x}; t)$ is superior to that of the $\mathbb{E}[\mathcal{I}_{\text{eff}}(\mathbf{x})]$. In the context of fixed target analysis, we do not observe a significant difference in the performance for most trials. Hence, we cannot find a clear winner in this case.

- **Infill Criterion for Practical Scenarios**

For choosing a sampling infill criteria for practical scenarios, we emphasize on the average running cpu time per iteration (ARCTPI), as well as the maximum cpu time required for an independent run: T_{\max} , in addition to the fixed budget and fixed target analyses. In the context of ARCTPI, i.e., Fig. 6.12, we deem $\mathcal{M}_{\text{eff}}(\mathbf{x}; t)$ and LCB_{eff} performing better than $\mathbb{E}[\mathcal{I}_{\text{eff}}(\mathbf{x})]$. In the context of T_{\max} , i.e., we deem LCB_{eff} performing superior to its competitors for two and five-dimensional problems. Combining the performance in the context of fixed budget analyses, fixed target analysis, ARCTPI, and T_{\max} , we deem LCB_{eff} and $\mathcal{M}_{\text{eff}}(\mathbf{x}; t)$ as suitable sampling infill criteria.

6.4 Summary and Discussion

In this chapter, we benchmarked the applicability of surrogate modeling on a real-world engineering case study. To this end, we considered a benchmark engineering case study based on the design of car hood frames. The associated data set contains over 10,000 3D mesh geometries for variants of card hood frames. This data set was generated through an automated, industry-grade CAD workflow, described in (Ramnath et al., 2019), and further benchmarked in (Wollstadt et al., 2022). The data set provided realistic designs of car hood frames, which were validated by experts with respect to realism, manufacturability, variability, and performance.

Based on this data set, we focused on two goals, which emphasized on benchmarking the performance of “one-shot optimization” strategy (Ta’asan et al., 1992) and the Bayesian optimization algorithm (Jones et al., 1998) for finding robust solutions. Our findings validate the performance of Kriging (Morales and Nocedal, 2011) as one of the most important modeling techniques in surrogate modeling. Furthermore, we observed the promising nature of ensemble methods, i.e., Random Forest, to effectively model the objective function in practical scenarios. We also validated the commonly-employed heuristic of utilizing a linear sample size to construct the model (Jurecka, 2007). Finally, in this context, we were satisfied with the quality of the optimal solutions obtained from surrogate modeling.

In the context of Bayesian optimization (Jones et al., 1998), we validated the impact of dimensionality, and consequently, the computational budget, on the performance of the algorithm. Furthermore, We validated the performance of the

6. ENGINEERING APPLICATIONS

“Moment-Generating Function of the Improvement” as an effective sampling infill criterion in Bayesian optimization, in addition to the “Lower Confidence Bound”. However, we could not validate the highly competitive nature of the “Expected-Improvement” Criterion. We believe this is due to the fact that the our design variables and noise settings take integer (rather than continuous) values¹.

It is important to note that our study has certain limitations. Ideally, we should have constructed the surrogate models (in both cases) from the continuous (latent) variables, derived from the 3D point cloud auto-encoders (Wollstadt et al., 2022), which in turn could have been constructed from the car hood geometries. This, however, would have given rise to further difficulties, since such design variables are generally not interpretable. Furthermore, defining the bounds and the constraints for such latent variables is a difficult task, i.e., again, due to the fact that they are not interpretable in the nominal sense. This, in turn, would also have meant that we cannot specify uncertainty and noise since that requires a precise understanding of the bounds of the design variables. Combining these points, we believe our methodology in the experimental setups makes more sense from a practical point of view, since it validates some of our earlier findings, and offers us a new perspective to learn from.

¹We believe the performance of the ensemble methods is also excellent due to the same reason, i.e., integer values for design and noise variables

