Learning the characteristics of engineering optimization problems with applications in automotive crash

Long, F.X.; Stein, B. van; Frenzel, M.; Krause, P.; Gitterle, M.; Bäck, T.H.W.; Fieldsend, J.E.

Learning the Characteristics of Engineering Optimization Problems with Applications in Automotive Crash

Fu Xing Long
fu-xing.long@bmw.de
BMW Group
Munich, Germany

Bas van Stein
b.van.stein@liacs.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Moritz Frenzel
moritz.frenzel@bmw.de
BMW Group
Munich, Germany

Peter Krause
krause@divis-gmbh.de
divis intelligent solutions GmbH
Dortmund, Germany

Markus Gitterle
markus.gitterle@hm.edu
University of Applied Sciences
Munich, Germany

Thomas Bäck
t.h.w.baeck@liacs.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

Oftentimes the characteristics of real-world engineering optimization problems are not well understood. In this paper, we introduce an approach for characterizing highly nonlinear and Finite Element (FE) simulation-based engineering optimization problems, focusing on ten representative problem instances from automotive crashworthiness optimization. By computing characteristic Exploratory Landscape Analysis (ELA) features, we show that these ten crashworthiness problem instances exhibit landscape features different from classical optimization benchmark test suites, such as the widely-used Black-Box Optimization Benchmarking (BBOB) problem set. Using clustering approaches, we demonstrate that these ten problem instances are clearly distinct from the BBOB test functions. Further analysis of the crashworthiness problem instances reveal that, as far as ELA concerns, they are most similar to a class of artificially generated functions. We identify such artificially generated functions and propose to use them as scalable and fast-to-evaluate representatives of the real-world problems. Such artificially generated functions could be used for the automated design of an optimization algorithm for specific real-world problem classes.

CCS CONCEPTS

• **Computing methodologies** → **Uncertainty quantification**; **Continuous space search**.

KEYWORDS

automotive crashworthiness, black-box optimization, exploratory landscape analysis, artificially generated functions, hierarchical clustering

ACM Reference Format:

Fu Xing Long, Bas van Stein, Moritz Frenzel, Peter Krause, Markus Gitterle, and Thomas Bäck. 2022. Learning the Characteristics of Engineering Optimization Problems with Applications in Automotive Crash. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3512290.3528712

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07.
https://doi.org/10.1145/3512290.3528712

1 INTRODUCTION

When dealing with black-box optimization problems, identifying and selecting the most time- and resource-efficient algorithm for a specific problem is a key challenge. This task is also known as the algorithm selection problem (ASP) [42]. In Evolutionary Computation, recent works focus on landscape-aware automated algorithm selection based on Machine Learning approaches [1, 7, 17, 18, 24, 38]. In these approaches, the so-called landscape features, which quantify different landscape characteristics of a given problem instance, are used to predict the performance of an optimization algorithm on the problem instance. In other words, the performance of an optimization algorithm on an unseen problem instance can be roughly estimated, once the problem landscape characteristics have been identified. This is beneficial, as landscape analysis can provide additional insights for explaining the effectiveness of an algorithm across different problem instances [44]. Complete reviews of this topic can be found in [21, 31, 35].

To the best of our knowledge, previous works on ASP were mainly based on academic benchmark functions, such as the Black-Box Optimization Benchmarking (BBOB) test set in [1, 17, 21]. On the other hand, little work has been attempted to investigate algorithm selection in the field of real-world expensive black-box optimization. An attractive application example is the crashworthiness optimization in automotive industry. For instance, the structural design of a car must provide sufficient protection to passengers in the event of a crash, while fulfilling other requirements at the same time, such as durability and weight [8]. As car design is getting ever more sophisticated, crashworthiness optimization is notoriously challenging and tedious. Furthermore, a Finite Element (FE) simulation of a highly nonlinear crash is usually computationally expensive, which severely limits the function evaluation budget affordable for optimization. While most work in the last years focused on developing different surrogate-based optimization methods for automotive crashworthiness problems [9–12, 19, 27–29, 43, 47], a proper understanding of the problem characteristics is still lacking.

Our long-term vision is to develop an automated optimization pipeline for automotive crashworthiness optimization problems, including algorithm selection and configuration. We aim to construct a generic pipeline that can be easily transferred and applied on other highly nonlinear expensive black-box optimization problems. To achieve that, we argue that a proper understanding of the problem characteristics of automotive crashworthiness optimization problem instances is essential in the first step. In this paper, we propose an approach for characterizing automotive crashworthiness optimization problem instances based on landscape characteristics with Exploratory Landscape Analysis (ELA). In this context, we are addressing the following research questions.

(1) What are the typical landscape characteristics of automotive crashworthiness optimization problems?
(2) Is there any similarity in landscape characteristics between crashworthiness optimization problems and academic benchmark functions, e.g., the BBOB functions?
(3) To which extent can the BBOB functions be used as representatives of the crashworthiness optimization problem class in terms of similarity in landscape characteristics? If none of the BBOB functions turn out to be sufficiently representing them, are there other types of test function, which could serve as a good approximation?

We argue that extensive insights into the objective function characteristics is helpful for developing an efficient black-box optimizer that is tuned for a particular problem class, in our case, automotive crashworthiness optimization. In this study, we focus on automotive side crash as a representative of this problem class, where the battery cells installed in an electric car must be additionally protected from crash impact. Furthermore, we consider only unconstrained single-objective optimization problems to minimize the problem complexity.

The remainder of this paper is structured as follow: Section 2 briefly introduces the concepts of ELA. This is followed by the description of our approach in Section 3 and an overview on the experimental setup in Section 4. Our experimental results are presented and discussed in Section 5. Lastly, conclusions and outlook are provided in Section 6.

2 EXPLORATORY LANDSCAPE ANALYSIS

One way of characterizing the complexity of continuous optimization problems is through the so-called high-level features defined by experts, including multi-modality, global structure, separability, variable scaling, search space homogeneity, basin size homogeneity, global to local optima contrast and plateaus [33]. To facilitate an automated workflow, six classes of low-level features were introduced in ELA to numerically quantify the landscape characteristics of an unknown optimization problem, consisting of y-distribution, level set, meta-model, local search, curvature and convexity [32]. Each feature class contains a set of features, which can be cheaply computed. For the feature computation, a data set of some sample points $\mathcal{X}$ and the corresponding function values $f : \mathcal{X} \to \mathbb{R}$ are required as inputs, with an assumption that the search space is continuous, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$, where d represents the number of design variables.

Apart from handling ASP tasks, ELA has been applied to understand the optimization landscape of neural architecture search tasks [51], to classify the BBOB problems [41], to analyze the problem space of different benchmark problem sets [52], to investigate whether the BBOB test function set can represent hyper-parameter tuning problems [5] and to study multi-objective optimization problems [23]. As far as we are aware, no previous work is related to the application of ELA on automotive crashworthiness optimization problems.

3 METHODOLOGY

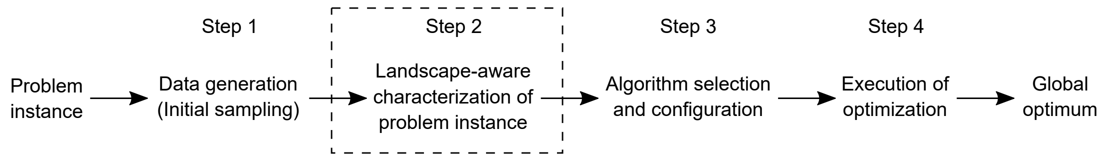
The general concept of our automated optimization pipeline is visualized in Figure 1(a) and the workflow proposed for characterizing problem instances based on ELA is presented in Figure 1(b). Within the framework of this paper, we refer mainly to the latter (Figure 1(b)). We feed different problem instances into our pipeline as input data, for which Design of Experiments (DoE) have been previously set up with numerous FE simulation runs executed in parallel to exploit the power of parallel computing. This so-called one-shot optimization approach [2, 3] is the classical strategy in automotive crashworthiness optimization. In one-shot optimization, optimization (e.g., with surrogate-based methods) is carried out based on an initial set of fixed sample points in a DoE, without evaluating new sample points in the process.

To characterize an automotive crashworthiness optimization problem instance, we design our pipeline based on two crucial aspects, including computation of ELA features on the problem instance and problem characterization based on the corresponding ELA features. That is, the ELA features computed on the crashworthiness problem instance will be compared with those computed on some established academic benchmark functions, namely, the BBOB functions. Subsequently, we can characterize the crashworthiness problem instance by identifying the BBOB function(s) with similar ELA features. In our study, we consider the noise-free BBOB problem set from the BBOB 2009 workshop [15] (e.g., available in the analysis tool IOHprofiler [6] and the benchmarking platform COCO [14]), consisting of altogether 24 real-parameter single-objective benchmark functions of different complexity, such as multi-modality, separability, global structure, etc. In fact, the BBOB functions have been commonly taken as test suites in benchmarking experiments for continuous optimization [13]. The four central sections of the pipeline (codes are available at https://github.com/fx-long/CEOELA) are described in detail in the following.

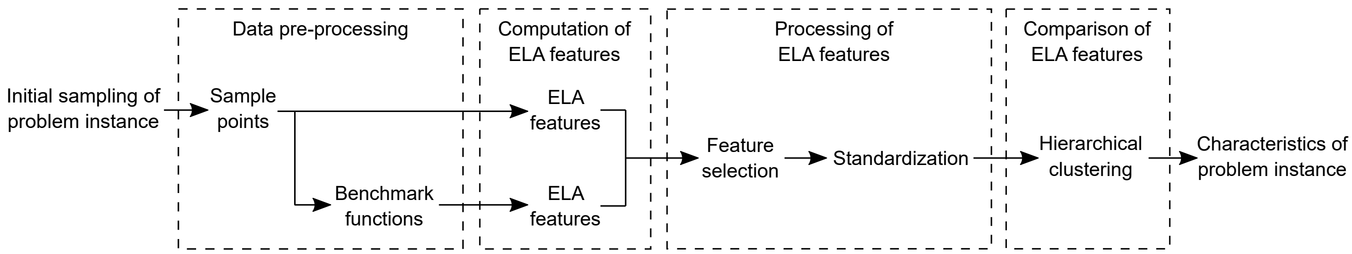
Data pre-processing. Firstly, the FE simulation input data are pre-processed, where incomplete data with missing result (e.g., due to interrupted FE simulation runs) are filtered out. Next, the design spaces of the problem instances are re-scaled to the domain $[-5, 5]$ with Equation 1, corresponding to the domain of the BBOB functions, where the global optimum of BBOB functions is located [15].

$$x_{new} = \frac{x_{orig} - a_{min}}{a_{max} - a_{min}} \cdot (b_{max} - b_{min}) + b_{min}\,, \qquad (1)$$

where x_{orig} and x_{new} are the design variables before and after scaling, a_{min} and a_{max} are the original minimum and maximum scale range, and b_{min} and b_{max} are the minimum and maximum scale range after re-scaling. Re-scaling is necessary to facilitate the comparison of ELA features between crashworthiness problem instances and the BBOB functions. Moreover, since ELA features are highly sensitive to sample size and sampling strategy [39, 40], the identical sample points (and thus the identical dimensions) of the



(a) Preliminary concept of our automated optimization pipeline for real-world highly nonlinear black-box optimization problems. The general idea is to capture the landscape characteristics of a problem instance and use this information for selecting and configuring an optimal optimization algorithm. The approach that we propose in this paper for characterizing problem instances based on landscape characteristics is labelled as Step 2 in the pipeline (outlined).



(b) Detailed visualization of the workflow of Step 2 in Figure 1(a), consisting of four central sections as marked with boxes. Generally, we characterize a black-box optimization problem instance by comparing its ELA features with those of benchmark functions, e.g., the BBOB functions.

Figure 1: Overview of (a) optimization pipeline and (b) approach proposed for characterizing engineering black-box optimization problems.

automotive crashworthiness problem instances are used to compute the ELA features of the BBOB functions for a fair comparison.

Computation of ELA features. For the ELA feature computation, we integrate the R-package **flacco** [26] into our pipeline. Other than the six classical ELA feature classes mentioned in Section 2, additional complementary feature classes were introduced and included in this package, such as dispersion, nearest better clustering (NBC), principal component analysis (PCA), linear model, Information Content of Fitness Sequences (ICoFiS), etc. [22, 25, 30, 34]. Among the more than 300 ELA features available, we only consider ELA features that can be computed

- without knowing the mathematical expression of the objective function f, and
- without the need to discretize the design spaces into blocks.

Moreover, we neglect features concerning the computational costs of each feature class, as they do not provide useful information on the problem landscape. In total 68 ELA features (from eight feature classes) are separately computed on the crashworthiness problem instances and the BBOB functions, as provided in Table 1. In cases where a feature computation fails (e.g., when the sample size is too small for computing the level set features), it will be skipped. Consequently, less ELA features will be computed for such a problem instance.

Processing of ELA features. Due to the fact that many of the ELA features are redundant [40, 52], we consider only subset of all ELA features computed for clustering. For this purpose, we carry out feature selection in the following two steps, which can be easily implemented in our automated pipeline.

(1) We first remove ELA features with zero-variance, that is, ELA features with a constant value across all functions.
(2) Next, we remove highly correlated ELA features based on Pearson's correlation coefficient. In our study, we consider a Pearson's correlation coefficient greater than 0.95 as highly correlated. For each highly correlated feature pair, the feature that has a higher mean correlation with other features is removed.

Lastly, the remaining ELA features are standardized (by removing mean and scaling to unit variance) with the intent to improve the distance-based clustering results.

Comparison of ELA features. To estimate the similarity between automotive crashworthiness problem instances and the BBOB fucntions, we measure the differences in their ELA feature values. We measure the pairwise distance between problems in terms of ELA features and then cluster them into groups accordingly based on the agglomerative hierarchical clustering approach [36]. With this approach, problems are clustered together in a bottom up fashion, starting with each problem as its own cluster and progressively merging clusters together until one large cluster is left, consisting of all problems. The Euclidean distance is chosen as the proximity metric between problems, where problems with higher similarity have a smaller Euclidean distance. Moreover, we consider the Ward's method [20] as linkage criterion for the cluster merging strategy, that is, by minimizing the within-cluster variance. Principally, clusters are selected for merging based on the smallest possible increase in the within-cluster sum of squared error (proportional to Euclidean distance) after merging. By analyzing the clustering results, the automotive crashworthiness problem instances can be characterized based on their neighbouring BBOB functions within the same cluster.

While the application of the distance-based hierarchical clustering approach in high dimensional problems is limited, because

Table 1: Brief descriptions of the eight ELA feature classes considered in this study (with the respective labels in flacco for feature classes and ELA features) [25].

Feature class	Description	ELA feature

y-distribution	Distribution of function values.
	Feature class labelled as `ela_distr.*`
	3 features
		`skewness`
		`kurtosis`
		`number_of_peaks`

Level set	Measure the performance of different classification methods based on function value thresholds.
	Feature class labelled as `ela_level.*`
	18 features
		`mmce_lda_{10,25,50}`
		`mmce_qda_{10,25,50}`
		`mmce_mda_{10,25,50}`
		`lda_qda_{10,25,50}`
		`lda_mda_{10,25,50}`
		`qda_mda_{10,25,50}`

Meta-model	Fitting quality of linear and quadratic models with and without interactions.
	Feature class labelled as `ela_meta.*`
	9 features
		`lin_simple.{adj_r2,intercept}`
		`lin_simple.coef.{min,max,max_by_min}`
		`lin_w_interact.adj_r2`
		`quad_simple.{adj_r2,cond}`
		`quad_w_interact.adj_r2`

Dispersion	Comparison of dispersion between initial sample points and subsets of points based on function value thresholds.
	Feature class labelled as `disp.*`
	16 features
		`ratio_mean_{02,05,10,25}`
		`ratio_median_{02,05,10,25}`
		`diff_mean_{02,05,10,25}`
		`diff_median_{02,05,10,25}`

NBC	Comparison of distance between all sample points towards nearest points and nearest points with better function value.
	Feature class labelled as `nbc.*`
	5 features
		`nn_nb.{sd_ratio,mean_ratio,cor}`
		`dist_ratio.coeff_var`
		`nb._fitness.cor`

PCA	Information based on PCA on initial sample points. Feature class labelled as `pca.*` 8 features	`expl_var.{cov_x,cor_x,cov_init,cor_init}` `expl_var_PC1.{cov_x,cor_x,cov_init,cor_init}`

Linear model	Measure the average coefficient vectors across multiple linear models.
	Feature class labelled as `limo.*`
	4 features
		`avg_length.{reg,norm}`
		`length.mean`
		`ratio.mean`

ICoFiS	Measure of smoothness, ruggedness and neutrality of the landscape through random walk.
	Feature class labelled as `ic.*`
	5 features
		`h.max`
		`eps.{s,max,ratio}`
		`m0`

distances between clusters in high dimensional spaces become relatively uniform and hence the notion of nearest neighbour is meaningless [46], we argue that the approach is still applicable for our middle-range dimensional problems (clustering based on a maximum possible of 68 ELA features).

4 EXPERIMENTAL SETUP

In our study, we focus on crashworthiness optimization of a rocker panel w.r.t. side crash against a pole, as shown in Figure 2. All the FE simulation data that we analyze were generated during several recent development projects by BMW, a German premium automobile manufacturer. The explicit FE simulations were solved with the commercial code LS-DYNA [4]. As summarized in Table 2, four rocker panels with similar design (D1-D4) and three different load cases, where the side pole was positioned at different locations (P1-P3) as shown in Figure 2(b), were investigated. The design variables were the thicknesses of different components of the rocker panels. In total, we consider ten representative problem instances, where the DoE of each problem instance was generated with the sampling method Modified Extensible Lattice Sequence (MELS) available in the commercial tool HyperStudy [16]. Basically, MELS is a sequential lattice space-filling DoE approach developed based on Sobol' sequences [45]. During the development projects, side crashes at the pole positions P1 and P3 for the rocker panel design D1 were considered by crash experts as non-critical and were therefore not investigated.

Table 2: Summary of ten automotive crashworthiness optimization problem instances, consisting of four rocker panel designs and three side pole positions.

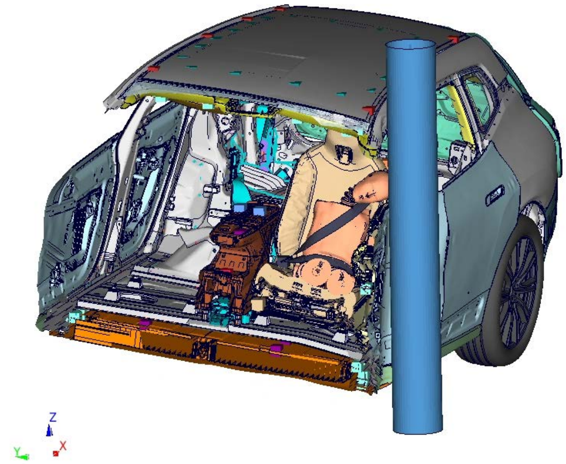
Problem instance	Rocker panel design	Pole position	Design variables	Sample size
1	D1	P2	18	530
2	D2	P1	22	59
3	D2	P2	22	309
4	D2	P3	22	309
5	D3	P1	16	150
6	D3	P2	16	102
7	D3	P3	16	132
8	D4	P1	20	329
9	D4	P2	20	330
10	D4	P3	20	333

During these development projects, the quality of a particular rocker panel design was evaluated by quantifying its structural crashworthiness through the following four objectives.

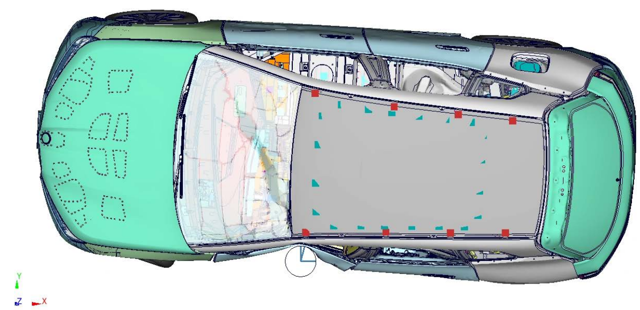
(1) Maximum force (F_{max}): Maximum impact force during side crash.
(2) Intrusion (*Intr*): Magnitude of inward structural deformation of the rocker panel into the battery cell compartment.
(3) Energy absorption (*EA*): The amount of kinetic energy absorbed during side crash.
(4) Rotation (*Rot*): Rotational deformation of rocker panel during side crash. This metric is introduced by crash experts internally to measure the average vertical deformation of FE nodes between inner and outer side of rocker panel.

All four objectives were measured and quantified as scalar FE outputs. Nonetheless, depending on the purposes of each development project, not all four objectives were always considered and therefore not all of them are available for our study. In our research, we separately analyze each of the objectives available and end up with a total of 30 (out of the maximum of 40 potentially available ones) single-objective optimization problem instances.

In our experiments, we feed the data sets of the ten automotive crashworthiness optimization problem instances into our pipeline and independently compute the aforementioned 68 ELA features. For each crashworthiness problem instance, we consider the mean ELA feature values computed based on a bootstrapping strategy to minimize the effects of random sampling in ELA, using a bootstrap size of 80% of the initial sample size and 30 repetitions. For the same reason, we consider the ELA feature values averaged across the first 20 instances (the global optimum is randomly shifted by rotating and translating the original problem [15]) for each of the 24 BBOB functions. Precisely, for all BBOB functions, we compute the ELA features on the first 20 instances based on the full sample points of each crashworthiness problem instance.



(a) Cross-section of the deformed FE model during a side crash. To protect passengers and battery cells, the crash impact energy must be sufficiently absorbed through plastic deformation of different components, such as rocker panel.



(b) Depending on the investigation purposes, the side pole can be positioned at different locations, that is, alongside the car body with slight offsets.

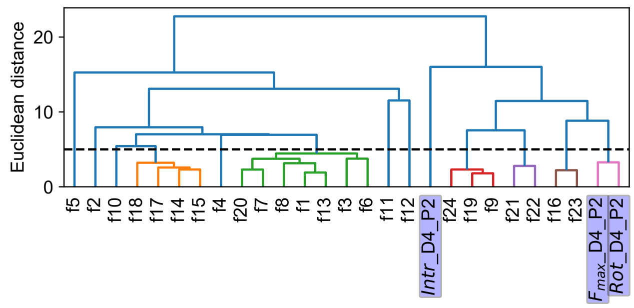
Figure 2: An example of FE model developed for investigating automotive side crash against a pole.

5 RESULTS

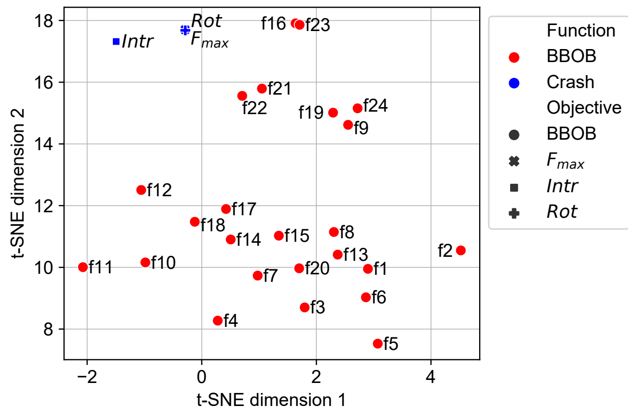
In this paper, we focus on analyzing results of the crashworthiness problem instance with design D4 and pole position P2 (in short D4_P2) as reference, because it has middle-range dimensions and sample size among all problem instances. Due to the limited space, the remaining results not included in this paper are made available in the repository https://zenodo.org/record/6424633.

5.1 BBOB functions

For the crashworthiness problem instance D4_P2, a final set of 35 ELA features are used for the hierarchical clustering, as visualized in Figure 3(a). In general, we observe that the crashworthiness objectives are separated from the BBOB functions, especially the intrusion objective. The fact that the maximum force and rotation objective are clustered in the same group as the BBOB functions f16 (Weierstrass function) and f23 (Katsuura function)



(a) Clustering pattern for 24 BBOB functions (labelled from f1 to f24) and crashworthiness objectives. The crashworthiness objectives are highlighted in blue color. The Euclidean distance of 5 is marked as reference.



(b) Projection of the high-dimensional ELA feature spaces onto a two-dimensional space through t-SNE visualization for 24 BBOB functions and crashworthiness objectives.

Figure 3: Clustering results based on ELA features between the BBOB functions and the crashworthiness problem instance D4_P2.

suggest that these objectives could have similar landscape properties, e.g., highly rugged and repetitive landscape. Additionally, we use the t-SNE approach [50] to visualize the high-dimensional ELA feature spaces on a two-dimensional space for an easier interpretation, as shown in Figure 3(b). For this purpose, we employ the `sklearn.manifold.TSNE` package [37], using 5,000 maximum number of iterations to allow convergence and lowering perplexity (metric for effective number of nearest neighbours) to 10 due to low data density (24 BBOB functions + 3 objectives = 27 functions). We recognize that the crashworthiness objectives are generally far away from the BBOB functions in the ELA feature spaces, indicating that the ELA features between them could be different.

To have a better understanding on the clustering results, we delve into examining the ELA feature values, which are presented in Figure 4. Based on visual inspection, we notice that several ELA features show clear differences in feature values between the BBOB functions and crashworthiness objectives, particularly `pca.expl_var_PC1.cor_x`, `pca.expl_var_PC1.cor_init` as well as `ela_meta.lin_w_interact.adj_r2`, which might explain the separation observed in the clustering pattern.

Since similar clustering patterns can be observed from the remaining crashworthiness problem instances, we suspect that the BBOB functions are different from our nonlinear automotive crashworthiness problem instances in terms of landscape characteristics. Consequently, the BBOB functions might be not sufficiently representative for our crashworthiness problem instances. In other words, the BBOB problem set seems to be inadequate in characterizing our crashworthiness problem classes.

5.2 Artificially generated functions

We further our research by shifting our focus towards test function generators that can create artificial test functions with similar landscape characteristics as to our automotive crashworthiness problem instances. Inspired by [53], our experimental testings show that the function generator in [49] has great potential in creating test functions similar to our crashworthiness problems. Following this argument, we re-implement the function generator [48] in Python, which was originally developed in Matlab, and integrate it into our pipeline in Figure 1(b) as benchmark functions with the following minor modifications:

- Discard a generated function as invalid, if any of the following conditions is fulfilled.
  (1) Invalid value in f, e.g., missing or infinity.
  (2) Extremely large value $|f| > 10^8$.
  (3) Extremely small value $|f| < 10^{-8}$.
  (4) Variance of $f < 1.0$, to avoid (on rare occasion) a constant function due to rounding, e.g., values f are rounded off to a single integer.

Generally, the function generator works in a tree-based fashion, that is, by constructing and extending a tree (function expression) with leaf nodes (mathematical operands and operators randomly selected from a predefined pool) [49].

We perform another investigation in a similar fashion by applying the pipeline on the previously introduced ten automotive crashworthiness problem instances and clustering with 1,000 artificial functions. Here, we use the same sample points of crashworthiness problem instances (bootstrap size of 80% of the initial sample size and 30 repetitions) to compute the ELA features of both artificial functions and crashworthiness problem instances. Unlike our previous investigation, we use the design spaces of crashworthiness problem instances for the generation of artificial functions and hence re-scaling to domain [-5,5] before ELA features computation is not required. Similarly, we consider the crashworthiness problem instance D4_P2 as our reference.

As visualized in Figure 5, we can observe that several artificial functions are clustered in the same groups as the crashworthiness objectives and with a smaller Euclidean distance as compared to the BBOB functions. In fact, these artificial functions are close to the crashworthiness objectives in the ELA feature spaces, as shown in Figure 6, indicating that they both have similar ELA feature values. As we can observe similar clustering patterns in most of the remaining crashworthiness problem instances, we believe that we can find artificial functions, which are sufficiently representative for our crashworthiness problem instances. In cases, where no

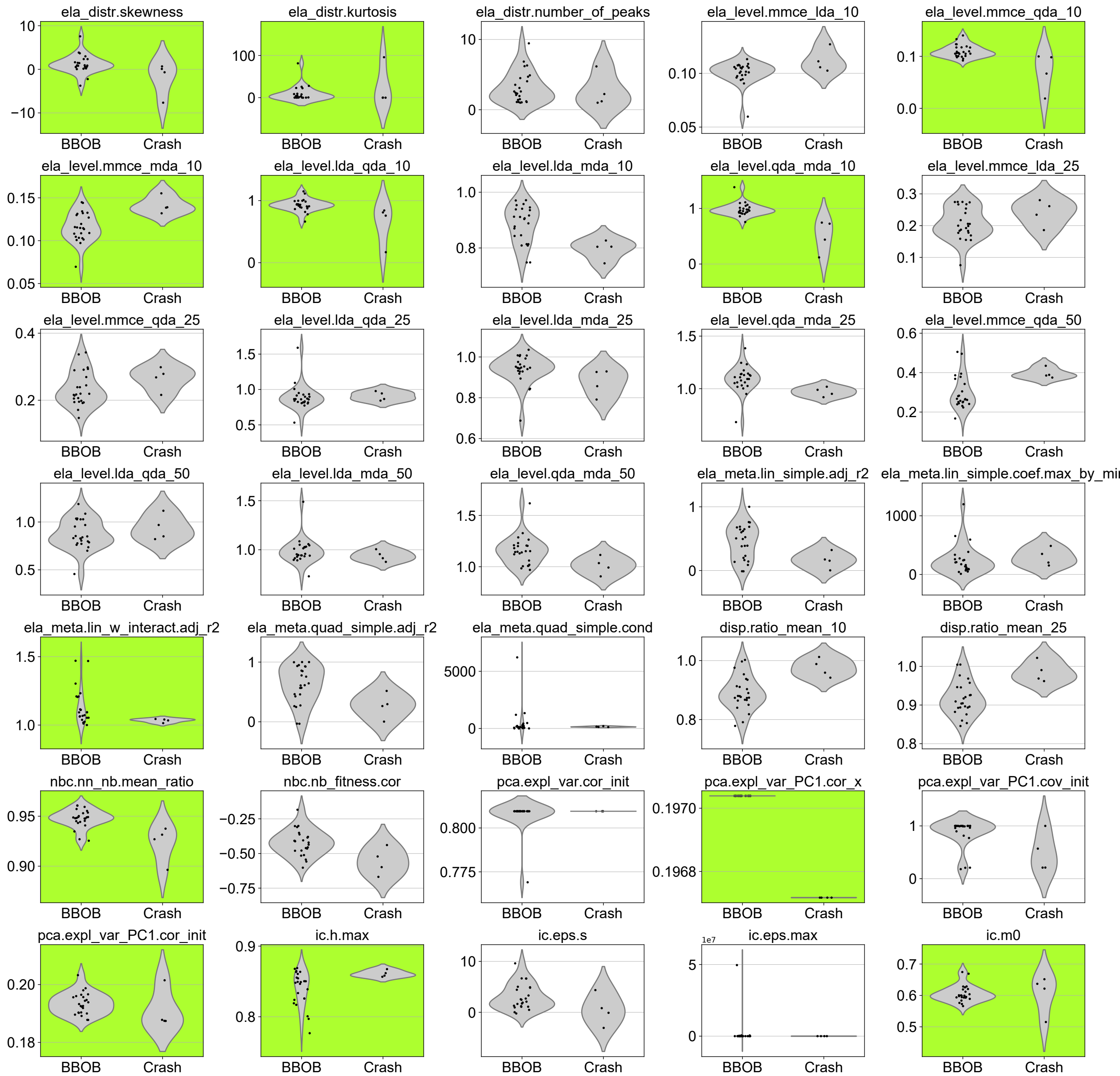


Figure 4: Distribution of the 35 ELA features for the 24 BBOB functions and three crashworthiness objectives of the problem instance D4_P2. Moreover, ELA features with clear differences between them are manually highlighted in green color based on observation.

(close) neighbouring artificial functions can be identified (e.g., rotation objective in D4_P1), we suspect that either (a) such a similar function is not included, which can be easily solved by increasing the test function set with more artificial functions, or (b) creating such a similar artificial function is simply not possible with the current function generator, which is a limitation of our approach and further improvements are necessary.

6 CONCLUSIONS AND OUTLOOK

In this paper, we propose an approach for characterizing the problem class of highly nonlinear engineering optimization problems. For this purpose, we develop an automated pipeline based on ten real-world representative automotive crashworthiness optimization problem instances, which can be conveniently transferred and applied on other similar problems. We focus on crashworthiness

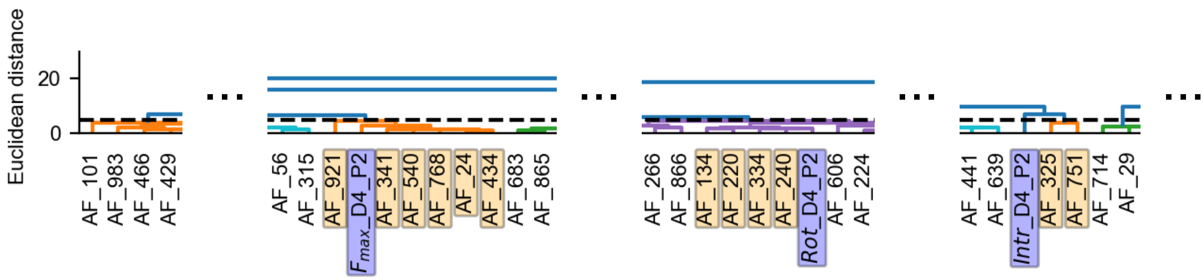


Figure 5: Clustering pattern for 1,000 artificial functions (labelled from AF_1 to AF_1000) and crashworthiness objectives of D4_P2. Only relevant sections of the clustering are shown due to the limited space. The crashworthiness objectives are highlighted in blue color, while the closely clustered artificial functions in orange color. The Euclidean distance of 5 is marked as reference.

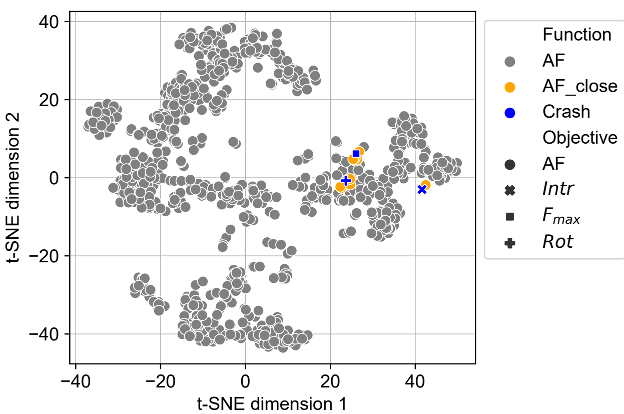


Figure 6: Projection of the high-dimensional ELA feature spaces onto a two-dimensional space through t-SNE visualization for 1,000 artificial functions and crashworthiness objectives of D4_P2. Here, we use the default value 30 for perplexity. (AF: Artificial function; AF_close: Artificial function clustered closely to crashworthiness objectives as in Figure 5).

optimization of rocker panel designs (between 16 and 22 design variables) for automotive side crash based on FE simulations. In our approach, we characterize the crashworthiness problem instances by comparing their landscape properties or ELA features with those of some well-established academic benchmark functions, such as the BBOB problem set, through hierarchical clustering.

By analyzing the ELA features, we can have a better understanding on the characteristics of our crashworthiness problem instances. Our results show that all the crashworthiness problem instances are separated from the BBOB test functions. Following this argument, the BBOB problem set is inappropriate for characterizing our crashworthiness problem instances, which belong to a distinguishable problem category. Consequently, we continue our investigation with an artificial function generator, which is capable in creating test functions with similar landscape properties as our crashworthiness problem instances. We suspect that, an artificial function with similar landscape characteristics can always be identified for all real-world problem instances, provided that (1) the function set is sufficiently large and (2) the problem complexity is well covered by the function generator.

In our future work, we intend to develop an automated design of optimization algorithms (Figure 1(a)) for real-world engineering optimization problems by exploiting the artificial functions with similar landscape characteristics. We propose to consider such artificial functions as scalable and fast-to-evaluate representatives of real-world problem instances. With this approach, we can improve the overall optimization efficiency by specifically designing and fine-tuning optimization algorithms for real-world problem classes. Ideally, we would like to extend our approach for multi-objective and constrained optimization problems.

Currently, the complexity of artificial functions is mostly limited by the predefined pool of mathematical operands and operators. An interesting idea for future work could be extending the diversity of artificial functions that can be generated, e.g., jump functions, to characterize a broader spectrum of optimization problems.

Furthermore, we would like to raise our concerns regarding the effectiveness of ELA features in quantifying the similarities between function landscapes, since ELA features are manually engineered by experts and thus might be biased in feature computation. Accordingly, another research outlook could be exploring other alternatives for feature computation in an unbiased way.

Moreover, we believe that there is still room for improvement in our feature selection approach. For future work, we plan to investigate other methods, which can potentially select optimal feature set, to accurately characterize optimization problems of different complexity.

ACKNOWLEDGMENTS

The contribution of this paper was written as part of the joint project newAIDE under the consortium leadership of BMW AG with the partners Altair Engineering GmbH, divis intelligent solutions GmbH, MSC Software GmbH, Technical University of Munich, TWT GmbH. The project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag.

REFERENCES

[1] Bernd Bischl, Olaf Mersmann, Heike Trautmann, and Mike Preuß. 2012. Algorithm Selection Based on Exploratory Landscape Analysis and Cost-Sensitive Learning. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO '12)*. Association for Computing Machinery, New York, NY, USA, 313–320. https://doi.org/10.1145/2330163.2330209

[2] Jakob Bossek, Carola Doerr, Pascal Kerschke, Aneta Neumann, and Frank Neumann. 2020. Evolving Sampling Strategies for One-Shot Optimization Tasks. In *Parallel Problem Solving from Nature – PPSN XVI*, Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann (Eds.), Vol. 12269. Springer International Publishing, Cham, 111–124. https://doi.org/10.1007/978-3-030-58112-1_8

[3] Olivier Bousquet, Sylvain Gelly, Karol Kurach, Olivier Teytaud, and Damien Vincent. 2017. Critical Hyper-Parameters: No Random, No Cry. arXiv:1706.03200 [cs.LG]

[4] Livermore Software Technology Corporation. 2019. LS-DYNA Theory Manual. Retrieved December 28, 2021 from https://ftp.lstc.com/anonymous/outgoing/jday/manuals/DRAFT_Theory.pdf

[5] Carola Doerr, Johann Dreo, and Pascal Kerschke. 2019. Making a Case for (Hyper-)Parameter Tuning as Benchmark Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 1755–1764. https://doi.org/10.1145/3319619.3326857

[6] Carola Doerr, Hao Wang, Furong Ye, Sander van Rijn, and Thomas Bäck. 2018. IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics. *arXiv e-prints:1810.05281* (2018). arXiv:1810.05281 https://arxiv.org/abs/1810.05281

[7] Johann Dréo, Carola Doerr, and Yann Semet. 2019. Coupling the Design of Benchmark with Algorithm in Landscape-Aware Solver Design. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 1419–1420. https://doi.org/10.1145/3319619.3326821

[8] Fabian Duddeck. 2008. Multidisciplinary optimization of car bodies. *Structural and Multidisciplinary Optimization* 35, 4 (2008), 375–389. https://doi.org/10.1007/s00158-007-0130-6

[9] H. Fang, Masoud Rais-Rohani, Z. Liu, and Mark Horstemeyer. 2005. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Computers & Structures* 83, 25-26 (2005), 2121–2136. https://doi.org/10.1016/j.compstruc.2005.02.025

[10] Jianguang Fang, Guangyong Sun, Na Qiu, Nam-Ho Kim, and Qing Li. 2017. On design optimization for structural crashworthiness and its state of the art. *Structural and Multidisciplinary Optimization* 55, 3 (2017), 1091–1119. https://doi.org/10.1007/s00158-016-1579-y

[11] Jimmy Forsberg and Larsgunnar Nilsson. 2006. Evaluation of response surface methodologies used in crashworthiness optimization. *International Journal of Impact Engineering* 32, 5 (2006), 759–777. https://doi.org/10.1016/j.ijimpeng.2005.01.007

[12] Karim Hamza and Mohamed Shalaby. 2014. A framework for parallelized efficient global optimization with application to vehicle crashworthiness optimization. *Engineering Optimization* 46, 9 (2014), 1200–1221. https://doi.org/10.1080/0305215X.2013.827672

[13] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 1689–1696. https://doi.org/10.1145/1830761.1830790

[14] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144. https://doi.org/10.1080/10556788.2020.1808977

[15] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. 2009. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions.* Research Report RR-6829. INRIA. https://hal.inria.fr/inria-00362633

[16] Altair Engineering Inc. 2022. Altair HyperStudy. Retrieved January 15, 2022 from https://www.altair.com/hyperstudy/

[17] Anja Jankovic and Carola Doerr. 2020. Landscape-Aware Fixed-Budget Performance Regression and Algorithm Selection for Modular CMA-ES Variants. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 841–849. https://doi.org/10.1145/3377930.3390183

[18] Anja Jankovic, Gorjan Popovski, Tome Eftimov, and Carola Doerr. 2021. *The Impact of Hyper-Parameter Tuning for Landscape-Aware Performance Regression and Algorithm Selection.* Association for Computing Machinery, New York, NY, USA, 687–696. https://doi.org/10.1145/3449639.3459406

[19] Tomas Jansson, Larsgunnar Nilsson, and Marcus Redhe. 2003. Using surrogate models and response surfaces in structural optimization – with application to crashworthiness design and sheet metal forming. *Structural and Multidisciplinary Optimization* 25, 2 (2003), 129–140. https://doi.org/10.1007/s00158-002-0279-y

[20] Joe H. Ward Jr. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (1963), 236–244. https://doi.org/10.1080/01621459.1963.10500845

[21] Pascal Kerschke, Holger Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation* 27, 1 (2019), 3–45. https://doi.org/10.1162/evco_a_00242

[22] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. 2015. Detecting Funnel Structures by Means of Exploratory Landscape Analysis. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO '15)*. Association for Computing Machinery, New York, NY, USA, 265–272. https://doi.org/10.1145/2739480.2754642

[23] Pascal Kerschke and Heike Trautmann. 2016. The R-Package FLACCO for exploratory landscape analysis with applications to multi-objective optimization problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 5262–5269. https://doi.org/10.1109/CEC.2016.7748359

[24] Pascal Kerschke and Heike Trautmann. 2019. Automated Algorithm Selection on Continuous Black-Box Problems by Combining Exploratory Landscape Analysis and Machine Learning. *Evolutionary Computation* 27, 1 (2019), 99–127. https://doi.org/10.1162/evco_a_00236

[25] Pascal Kerschke and Heike Trautmann. 2019. *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package Flacco.* Springer International Publishing, Cham, 93–123. https://doi.org/10.1007/978-3-030-25147-5_7

[26] Pascal Kerschke and Heike Trautmann. 2019. flacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems. https://github.com/kerschke/flacco Last accessed 15 January 2022.

[27] Morteza Kiani, Imtiaz Gandikota, A. Parrish, Keiichi Motoyama, and Masoud Rais-Rohani. 2013. Surrogate-based optimisation of automotive structures under multiple crash and vibration design criteria. *International Journal of Crashworthiness* 18, 5 (2013), 473–482. https://doi.org/10.1080/13588265.2013.805294

[28] Hasan Kurtaran, Azim Eskandarian, D. Marzougui, and N. Bedewi. 2002. Crashworthiness design optimization using successive response surface approximations. *Computational Mechanics* 29, 4 (2002), 409–421. https://doi.org/10.1007/s00466-002-0351-x

[29] Xingtao Liao, Qing Li, Xujing Yang, Wei Li, and Weigang Zhang. 2008. A two-stage multi-objective optimisation of vehicle crashworthiness under frontal impact. *International Journal of Crashworthiness* 13, 3 (2008), 279–288. https://doi.org/10.1080/13588260801933659

[30] Monte Lunacek and Darrell Whitley. 2006. The Dispersion Metric and the CMA Evolution Strategy. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*. Association for Computing Machinery, New York, NY, USA, 477–484. https://doi.org/10.1145/1143997.1144085

[31] Katherine Mary Malan. 2021. A Survey of Advances in Landscape Analysis for Optimisation. *Algorithms* 14, 2 (2021), 40. https://doi.org/10.3390/a14020040

[32] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. 2011. Exploratory Landscape Analysis. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. Association for Computing Machinery, New York, NY, USA, 829–836. https://doi.org/10.1145/2001576.2001690

[33] Olaf Mersmann, Mike Preuss, and Heike Trautmann. 2010. Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis. In *Parallel Problem Solving from Nature, PPSN XI*, Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 73–82. https://doi.org/10.1007/978-3-642-15844-5_8

[34] Mario Andrés Muñoz, Michael Kirley, and Saman K. Halgamuge. 2015. Exploratory Landscape Analysis of Continuous Space Optimization Problems Using Information Content. *IEEE Transactions on Evolutionary Computation* 19, 1 (2015), 74–87. https://doi.org/10.1109/TEVC.2014.2302006

[35] Mario Andrés Muñoz, Yuan Sun, Michael Kirley, and Saman K. Halgamuge. 2015. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences* 317 (2015), 224–245. https://doi.org/10.1016/j.ins.2015.05.010

[36] Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 1 (2012), 86–97. https://doi.org/10.1002/widm.53

[37] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

[38] Maxim Pikalov and Vladimir Mironovich. 2021. *Automated Parameter Choice with Exploratory Landscape Analysis and Machine Learning.* Association for Computing Machinery, New York, NY, USA, 1982–1985. https://doi.org/10.1145/3449726.3463213

[39] Quentin Renau, Carola Doerr, Johann Dreo, and Benjamin Doerr. 2020. Exploratory Landscape Analysis is Strongly Sensitive to the Sampling Strategy. In *Parallel Problem Solving from Nature – PPSN XVI*, Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann (Eds.). Springer International Publishing, Cham, 139–153. https://doi.org/10.1007/978-3-030-58115-2_10

[40] Quentin Renau, Johann Dreo, Carola Doerr, and Benjamin Doerr. 2019. Expressiveness and Robustness of Landscape Features. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 2048–2051. https://doi.org/10.1145/3319619.3326913

[41] Quentin Renau, Johann Dreo, Carola Doerr, and Benjamin Doerr. 2021. Towards Explainable Exploratory Landscape Analysis: Extreme Feature Selection for Classifying BBOB Functions. In *Applications of Evolutionary Computation*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.). Springer International Publishing, Cham, 17–33. https://doi.org/10.1007/978-3-030-72699-7_2

[42] John R. Rice. 1976. The Algorithm Selection Problem. Advances in Computers, Vol. 15. Elsevier, 65–118. https://doi.org/10.1016/S0065-2458(08)60520-3

[43] Lei Shi, Ren-Jye Yang, and Ping Zhu. 2012. A method for selecting surrogate models in crashworthiness optimization. *Structural and Multidisciplinary Optimization* 46, 2 (2012), 159–170. https://doi.org/10.1007/s00158-012-0760-1

[44] David Simoncini, Sophie Barbe, Thomas Schiex, and Sébastien Verel. 2018. Fitness Landscape Analysis around the Optimum in Computational Protein Design. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 355–362. https://doi.org/10.1145/3205455.3205626

[45] Il'ya Meerovich Sobol'. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *U. S. S. R. Comput. Math. and Math. Phys.* 7, 4 (1967), 86–112. https://doi.org/10.1016/0041-5553(67)90144-9

[46] Michael Steinbach, Levent Ertöz, and Vipin Kumar. 2004. *The Challenges of Clustering High Dimensional Data*. Vol. 213. Springer Berlin Heidelberg, Berlin, Heidelberg, 273–309. https://doi.org/10.1007/978-3-662-08968-2_16

[47] Guangyong Sun, Ye Tian, Ruoyu Wang, Jianguang Fang, and Qing Li. 2020. Parallelized multiobjective efficient global optimization algorithm and its applications. *Structural and Multidisciplinary Optimization* 61, 2 (2020), 763–786. https://doi.org/10.1007/s00158-019-02417-1

[48] Ye Tian, Shichen Peng, Xingyi Zhang, Tobias Rodemann, Kay Chen Tan, and Yaochu Jin. 2020. Algorithm-Recommendation. https://github.com/BIMK/Algorithm-Recommendation Last accessed 15 January 2022.

[49] Ye Tian, Shichen Peng, Xingyi Zhang, Tobias Rodemann, Kay Chen Tan, and Yaochu Jin. 2020. A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks. *IEEE Transactions on Artificial Intelligence* 1, 1 (2020), 5–18. https://doi.org/10.1109/TAI.2020.3022339

[50] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[51] Bas van Stein, Hao Wang, and Thomas Bäck. 2020. Neural Network Design: Learning from Neural Architecture Search. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1341–1349. https://doi.org/10.1109/SSCI47803.2020.9308394

[52] Urban Škvorc, Tome Eftimov, and Peter Korošec. 2020. Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. *Applied Soft Computing* 90 (2020), 106138. https://doi.org/10.1016/j.asoc.2020.106138

[53] Urban Škvorc, Tome Eftimov, and Peter Korošec. 2021. *A Complementarity Analysis of the COCO Benchmark Problems and Artificially Generated Problems*. Association for Computing Machinery, New York, NY, USA, 215–216. https://doi.org/10.1145/3449726.3459585