Landscape Analysis Based vs. Domain-Specific Optimization for Engineering Design Applications: A Clear Case

Roy de Winter
LIACS, Leiden University
Leiden, The Netherlands
r.de.winter@liacs.leidenuniv.nl

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

Andre Thomaser
BMW Group
Munich, Germany
andre.thomaser@bmw.de

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

Niki van Stein
LIACS, Leiden University
Leiden, The Netherlands
n.van.stein@liacs.leidenuniv.nl

Anna V. Kononova
LIACS, Leiden University
Leiden, The Netherlands
a.kononova@liacs.leidenuniv.nl

***Abstract*—Traditional methods in black-box optimization often prescribe general-purpose algorithms for broad problem categories, which overlooks the significant variability in optimization landscapes that can exist even within a single domain. We address this gap by demonstrating the efficacy of landscape analysis techniques in guiding the choice and tuning of optimization algorithms for distinct landscape features. This study focuses on two real-world problem domains characterized by black-box simulation-based optimization: automotive control system calibration (anti-lock braking system) and automotive crashworthiness optimization. Through comparative analysis of computer experiments in these domains, landscape analysis techniques are employed for both visualization and explanation purposes. The results reveal substantial variations in landscape features across different instances within the same problem domain. Consequently, we advocate for a paradigm shift towards learning and applying optimizers tailored to specific landscape feature spaces rather than applying a one-size-fits-all approach to an entire problem domain. This research not only enhances the understanding of landscape variability in optimization problems but also paves the way for more efficient and effective optimization processes in complex, real-world scenarios.**

***Index Terms*—Landscape Analysis, Algorithm Selection, Black Box Optimization, Engineering Optimization.**

I. Introduction

In engineering optimization applications, the objective function is often expensive, e.g. computed based on the output generated by running costly and/or time-consuming simulations. For instance, structural optimization or control system calibration in the automotive industry are two typical examples. Given a resource-constrained real-world setting, in terms of the number of available commercial simulator licenses and the maximally available wall-clock time for running an optimization, practitioners would ideally prefer a domain-specific optimization algorithm selection approach ("for a structural mechanics problem, use optimization algorithm A") over an instance-based optimization algorithm selection ("for this particular instance of a structural mechanics problem, use optimization algorithm B"). Commercially available engineering design optimization tools, such as[1] `Optimus` (Noesis Solutions), `modeFRONTIER` (ESTECO), `Hyperstudy` (Altair), `LS-OPT` (DYNAmore), `optiSLang` (Ansys), and `HEEDS` (Siemens), typically offer a domain-specific approach.

The reason for preferring a domain-specific optimization algorithm selection [1]–[3] is that an instance-based algorithm selector necessarily requires some features of the problem instance to be computed. This requires the execution of several simulation runs, e.g., for evaluating an initial Design of Experiments (DoE), based on which the instance-specific features can be computed. When dealing with algorithm selection problems in the continuous optimization domain, the so-called landscape analysis can be used to obtain such features [4], which can be exploited to select the best-performing algorithm [5].

In this paper, two completely different engineering design optimization domains are investigated, namely the designs of vehicle dynamics control systems and vehicle crashworthiness, to prove our hypotheses empirically: *An instance-based optimization algorithm selection approach is needed as the diversity of landscape features across problem instances from the same engineering design domain is high.* In fact, our empirical results imply that diversity within a single engineering design domain can be as high as across different domains. To phrase this differently, one cannot assume an optimization algorithm that performs well on one instance of an engineering design domain to also work well on another instance. Therefore, there is no single best solver for a single domain, instead an instance-based algorithm selector is required: No-free-lunch principle [6] and the focused no-free-lunch principle [7] also hold within a specific engineering design problem domain.

[1]This list is incomplete and serves as representative examples only.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00148

II. EXPLORATORY LANDSCAPE ANALYSIS

The complexity of continuous optimization problems can be understood by analyzing high-level properties, such as multimodality, global structure, and separability. To automate this analysis, exploratory landscape analysis (ELA) techniques were developed [8], [9], which quantify six fundamental landscape properties, namely distribution of objective values, level sets, meta-models, local searches, curvature and convexity. Subsequently, more features were added to these six primary ELA categories, including dispersion, nearest better clustering (NBC), principal component analysis (PCA), linear models and information content of fitness sequences (ICoFiS) [10].

These ELA features have proven effective for classifying functions used in the black-box optimization benchmarking (BBOB) suite. In brief, the computation of ELA features requires a DoE with a set of samples $\mathbf{X} = \{x_1, \cdots, x_n\}$ evaluated on a specific objective function f, i.e., $f : \mathbb{R}^d \to \mathbb{R}$, where $x_i \in \mathbb{R}^d$, n is the sample size, and d is the dimensionality. However, it is important to note that the effectiveness of ELA features can depend on the DoE sample size and sampling strategy [11]. Previously, ELA has been utilized in various domains beyond algorithm selection problems, including vehicle dynamics control [12], [13], neural architecture search [14], automotive car crash optimization [15], [16], and selecting optimal hyperparameters in Bayesian optimization algorithms [17]. While ELA has already been in the literature successfully, it still struggles with a few challenges, including but not limited to (i) potential bias in hand-crafted features, (ii) decision making in feature selection, and (iii) dependency on sampling size and strategy. There are several alternatives to ELA features, such as DoE2Vec [18] and Deep-ELA [19], which use deep-learning techniques to automatically learn latent features from landscape samples. These latent features are, however, less explainable and understandable than classical ELA features and are therefore left out of the scope of this research.

III. ENGINEERING PROBLEMS

In this work, the exploratory landscape features of two different engineering problem domains are studied. First, the landscape features of Anti-lock Braking Systems of cars are explored, secondly, the crashworthiness of automative design problems is analyzed.

TABLE I
VEHICLE TIRE AND LOADING SETTINGS

Function	Tires	Vehicle load
y_1	High performance	Partially loaded
y_2	Medium performance	Partially loaded
y_3	Under performance	Partially loaded
y_4	High performance	Fully loaded
y_5	High performance	Little loaded

A. Design of Vehicle Dynamics Control Systems

Control systems for vehicle dynamics, such as the anti-lock braking system (ABS) [20], have significantly improved vehicle safety and handling within the automotive sector. ABS prevents wheels from locking during braking events by modulating brake pressure to maintain brake slip in a desirable range. This action shortens the braking distance and enables the driver to retain steering control during urgent braking situations. The effectiveness of these systems depends on precise calibration of their parameters to cater to various driving conditions and vehicle configurations.

A benchmark test commonly used to assess a vehicle's braking capability is the emergency straight-line full-stop braking test with the ABS fully activated [21]. This test is divided into three distinct phases:

1: The vehicle is accelerated to a speed of 103.5 km/h.
2: The vehicle's speed is maintained at a steady 103 km/h without any acceleration or deceleration.
3: The brakes are applied and held until the vehicle comes to a complete stop.

The calculation of braking distance is determined by integrating the vehicle's longitudinal velocity over the time interval from the moment the vehicle's speed is 100 km/h (v_s) at time t_s until it comes to a halt at 0 km/h (v_e) at time t_e. To ensure accuracy and avoid the influence of any initial disturbances, the initial deceleration phase from 103 km/h to 100 km/h is excluded from the braking distance computation. In accordance with ISO 21994:2007 [21], the prescribed methodology for measuring braking distance involves a series of ten individual valid measurements. The objective y, which represents the braking distance to be minimized, is defined as follows:

$$y = \frac{1}{10} \sum_{k=1}^{10} \int_{t_s}^{t_e} v_k(t)\, dt. \tag{1}$$

1) ABS Benchmarking Dataset: We use a dataset with braking distances for five different vehicle settings from [22], where a setting consists of a vehicle load and a tire (Table I). The two ABS parameters x_1 and x_2 have the defined bounds: $x_1 \in [-5, 6]$ and $x_2 \in [-5, 4]$. For x_1 and x_2 a set of values D_i with a resolution of 0.1 as equal distance between the values is provided. Thus, there are 10 101 possible combinations for these two ABS parameters and the corresponding braking distance per vehicle setting. In order to apply Exploratory Landscape Analysis, in the following we consider the problem as *quasi-continuous*, and values are rounded to the nearest given data point.

Within the used data set the braking distances of a particular vehicle setting y_i are specified as the distance in meters to the corresponding optimal breaking distance. Figure 1 shows the distribution of the 10 101 data points for each real-world problem y_i. Note that minimizing the braking distance for each vehicle setting serves as one objective, and a different parameter set (x_1, x_2) is optimal for each vehicle setting.

B. Crashworthiness of Automotive Design

Our second engineering domain is crashworthiness optimization in the automotive industry, which is a classical example of black-box optimization problem with expensive

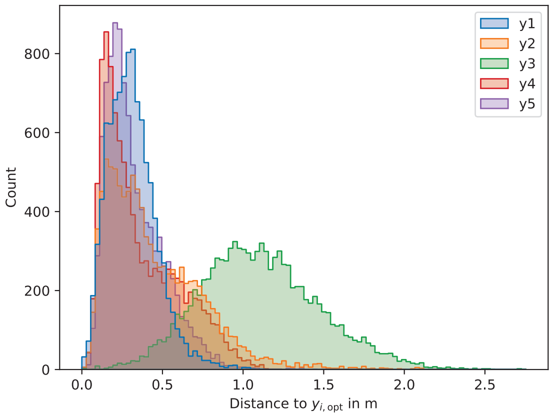

Fig. 1. Distribution of distances to the optimal braking distance for the 10 101 ABS parameter combinations, per vehicle setting y_i from Table I. The data was published in [22].

function evaluation, e.g., requiring finite element (FE) simulation runs. Due to increasingly strict road safety regulations by authorities, vehicle design is getting more challenging and tedious. In automotive crashworthiness optimization, the primary target is to identify an optimal vehicle design that not only can sufficiently protect passengers in the event of a crash but also fulfill other requirements, such as low weight and production costs [23]. The optimization problem is typically formulated as a minimization of the vehicle weight subject to constraints, e.g., peak impact force or magnitude of vehicle deformation.

The focus of this work is on the FE simulation-based crashworthiness optimization of the vehicle body, also known as body-in-white, which belongs to the early phase of vehicle design. Classically, automotive crash optimization problems are solved with the response surface method, where a response surface is constructed based on a preferably large DoE to approximate the true function and to predict the global optimum, i.e., the best vehicle design [24]. Following this, we analyze crash problem instances using the DoE samples generated by a German premium automobile manufacturer in previous vehicle development projects, where the problem instances were mainly different in terms of crash scenarios, vehicle models, and load cases, e.g., different pole positions for side crash, as shown in Figure 2. On average, each FE simulation required around 20 computation hours on high-performance computing clusters, using the commercial solver `LS-DYNA` [25]. Consequently, an enumeration of the whole design space is computationally infeasible for crash optimization, unlike in (simplified) vehicle dynamic problems. Nonetheless, to ensure a reliable computation of ELA features, only crash problem instances with a DoE sample size of at least $10d$ are considered. Altogether, 13 crash problem instances consisting of seven side crashes, two rear crashes, and four front crashes are available for our analysis, with

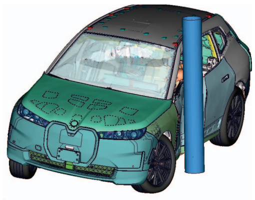

Fig. 2. An example of vehicle side crash using FE simulation, where the pole can be positioned at a different location depending on the optimization problem definition.

dimensionalities between 8 and 22, where the design variables are the thicknesses of different vehicle components.

Vehicle designs were evaluated based on the following five types of *crash functions*, which can be quantified as scalar FE simulation outcomes:

1: Mass (*M*): Component weight, affecting the vehicle weight and manufacturing costs;
2: Intrusion (*Intr*): Magnitude of inward structural deformation, which might cause passenger injuries and damage crucial components, e.g., the battery in an electric vehicle;
3: Maximum force (F_{max}): Peak impact force during crash;
4: Energy absorption (*EA*): The total amount of kinetic energy absorbed during the crash;
5: Rotation (*Rot*): Rotational deformation of components during crash.

IV. Experimental Setup

To test our hypothesis; *an instance-based optimization algorithm selection approach is needed because the diversity of landscape features across problem instances from the same engineering design domain is large*, an experiment is set up. In the experiment, the ELA features of the two engineering problems are computed and compared to the ELA features of well-known academic benchmark functions. The benchmark problems selected to compare with are the black-box optimization benchmarking (BBOB) testsuite problems. The BBOB suite consists of 24 continuous and noiseless functions of different optimization landscape complexity and was introduced to facilitate the development and performance evaluation of optimization algorithms [26]. The ELA features of BBOB functions are considered as references in scaling and transforming the engineering problems for a fair investigation of the landscape characteristics.

The ELA features of both engineering domains are analyzed sequentially by executing the following three steps:

a) Re-scaling: The design space is rescaled to $[-5, 5]^d$, which is the typical search domain considered in the BBOB suite. Furthermore, the objective values are normalized by using min-max scaling to minimize potential inherent bias in ELA feature computation [27].

b) ELA feature computation: The ELA features are computed for each problem instance using the `pflacco` package [28]. In this work, 49 ELA features are computed based on a DoE without requiring additional function evaluations

TABLE II
SUMMARY OF 49 ELA FEATURES CONSIDERED IN THIS WORK

Feature class	ELA feature
y-distribution (3 features)	skewness kurtosis number_of_peaks
Level set (9 features)	mmce_lda_{10,25,50} mmce_qda_{10,25,50} lda_qda_{10,25,50}
Meta-model (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 (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 (5 features)	nn_nb.{sd_ratio,mean_ratio,cor} dist_ratio.coeff_var nb._fitness.cor
PCA (2 features)	expl_var_PC1.{cov_init,cor_init}
ICoFiS (5 features)	h.max eps.{s,max,ratio} m0

(Table II). It is to be noted, that if a feature computation fails, e.g., due to small DoE sample size, this feature will be skipped.

For the vehicle dynamics control systems, the DoE sample size is $10\,101$ and all ELA features can be calculated. The DoE sample size of some crash problem instances is only $(10d)$, which is too small for reliable ELA feature computation. Therefore, the ELA features on the crash problem instances are calculated in a bootstrapping manner. Meaning, the ELA feature computation is repeated for 20 times using only 80% of the original DoE samples that are randomly selected in each repetition, and eventually, the mean feature values are used in further analysis. Similarly, the ELA features are computed of the BBOB functions using the same DoE samples $\mathbf{X}$ as the problem instance and averaged across the first five BBOB instances.

c) Dimension reduction: To visualize the distribution of engineering problem instances within the ELA feature space, the high-dimensional ELA feature vectors are projected onto a 2-dimensional space using a PCA approach. Another advantage of using PCA is that ELA features that are highly correlated can be excluded [29]. Before the PCA transformation, the ELA features are normalized using min-max scaling, to ensure that all ELA features are within a similar range. We perform both steps here first on the ELA features of BBOB functions, and then on the engineering problem instances, using identical scaling and transformation steps.

V. RESULTS

The distribution of engineering problem instances in the ELA feature space is visualized in Figure 3 based on the first

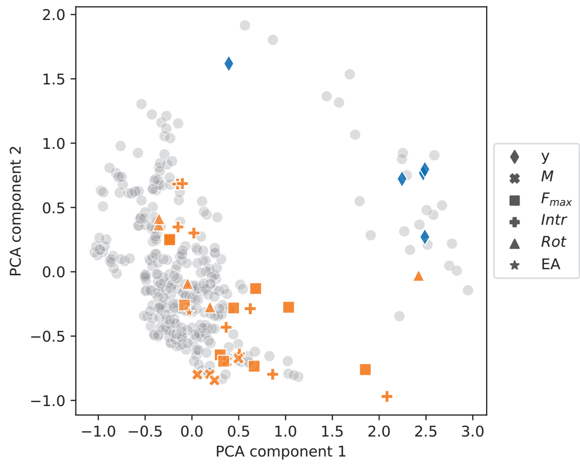

Fig. 3. Visualization of the ELA feature space for the five ABS (blue color), 30 crash (orange color), and 336 BBOB functions (altogether 24 BBOB $\times$ 14 problem instances; gray color) using the first two PCA components.

two PCA components. Remarkably, a clear separation between the vehicle dynamic and crash problems can be observed in the ELA feature space, indicating that both domains are indeed different in terms of landscape characteristics. Beyond that, within the same engineering domain, the problem instances are scattered across the ELA feature space as well, where the diversity can be as large as across different domains.

To have a deeper understanding on the separation between engineering problem instances, each computed ELA feature is indivudually inspected in box plots, as shown in Figure 4. The level set features are not shown here, because the feature computation fails in many of the problem instances. Based on visual inspection, we notice that many of the ELA features are indeed different between the two engineering domains. For an unbiased investigation, the ELA feature distributions are compared using the two-sample Kolmogorov-Smirnov (KS) test [30], with the null hypothesis that *the distribution of ELA features is similar*.

VI. CONCLUSIONS AND FUTURE WORK

On closely related instance sets of two optimization problems defined in two different engineering domains, a high diversity of landscape properties is observed, as exemplified by the distributions of individual ELA features of problem instances as well as their visualization after performing dimensionality reduction. Such diversity can in practice lead to large differences in performance across a portfolio of optimization algorithms on these instances [3], [31]. Therefore, using performance data from the optimizers executed on a limited set of problem instances to subsequently select the best optimizer for a new unseen problem instance within the same engineering domain is risky as instance diversity within a domain can be as high as across different domains. Even within one engineering domain, algorithm selection is

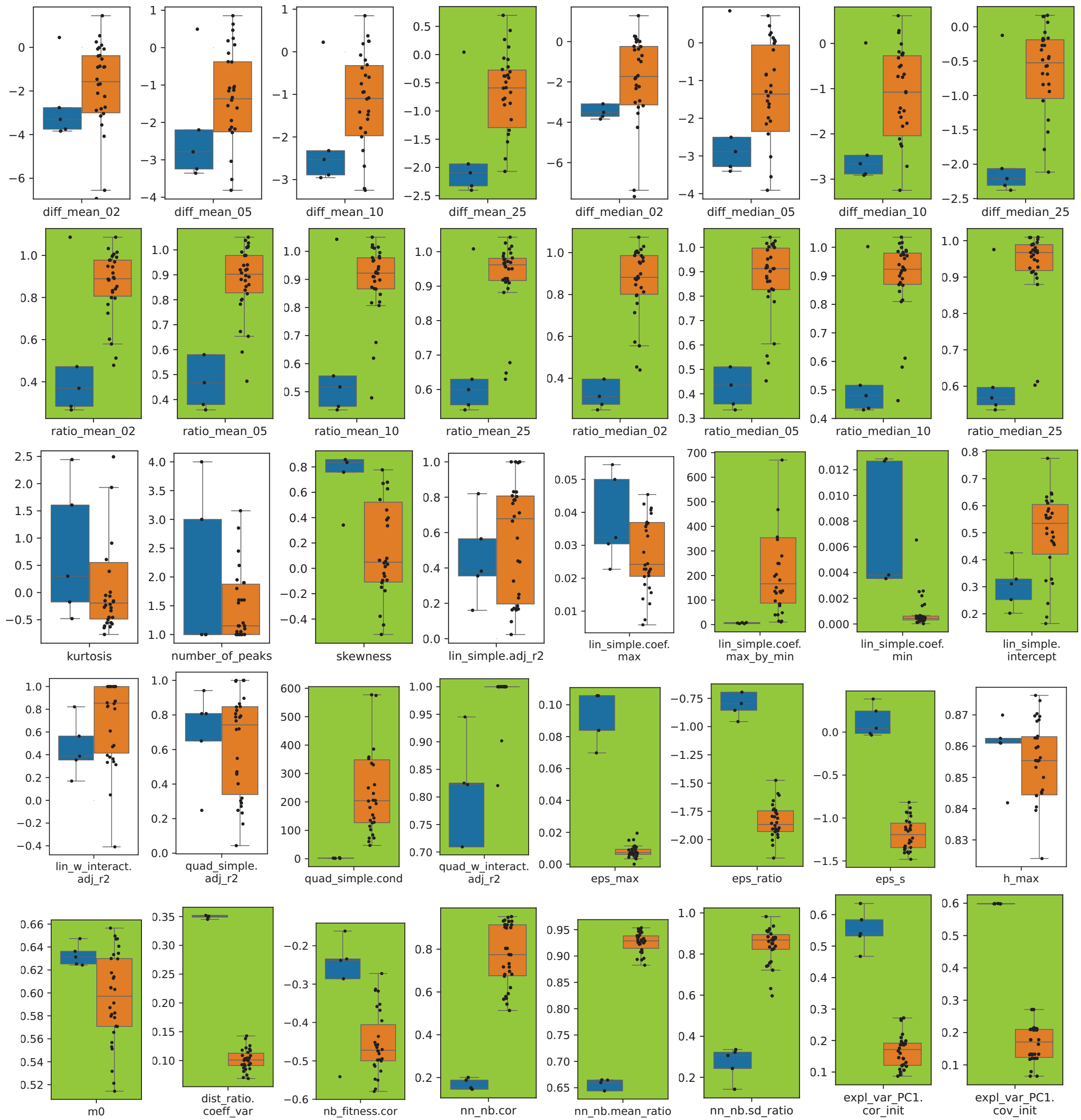

Fig. 4. Comparison of 40 ELA features in raw feature values between the five ABS (blue color) and 30 crash functions (orange color). We highlight an ELA feature with green color, if the KS-test null hypothesis *distribution of feature values between the ABS and crash functions are similar* is rejected with 95% confidence.

not possible across traditionally defined problem instances. We assume that such conclusions apply well beyond the two considered engineering domains. However, an open question remains regarding how to define problem classes. In conclusion, the case of these engineering applications clearly shows the diversity of problem instances within a single domain, implying that selecting the best optimization algorithm should be based on the specific problem instance characteristics rather than the engineering domain. The authors believe that exploratory landscape analysis features can help guide the user select the best algorithm, respectively on the engineering domain.

Acknowledgment: This paper was written as part of the joint project newAIDE. 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] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection: Survey and perspectives," *Evolutionary computation*, vol. 27, no. 1, pp. 3–45, 2019.

[2] P. Kerschke and H. Trautmann, "Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning," *Evolutionary Computation*, vol. 27, no. 1, p. 99–127, 2019.

[3] R. P. Prager, M. V. Seiler, H. Trautmann, and P. Kerschke, "Automated algorithm selection in single-objective continuous optimization: a comparative study of deep learning and landscape analysis methods," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2022, pp. 3–17.

[4] K. M. Malan, "A survey of advances in landscape analysis for optimisation," *Algorithms*, vol. 14, no. 2, p. 40, 2021.

[5] M. A. Muñoz, M. Kirley, and S. K. Halgamuge, "The algorithm selection problem on the continuous optimization domain," in *Computational Intelligence in Intelligent Data Analysis*, C. Moewes and A. Nürnberger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–89.

[6] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[7] D. Whitley and J. Rowe, "Focused no free lunch theorems," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 811–818.

[8] O. Mersmann, M. Preuss, and H. Trautmann, "Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis," in *Parallel Problem Solving from Nature, PPSN XI*, ser. Lecture Notes in Computer Science, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Berlin: Springer, 2010, pp. 73–82.

[9] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory Landscape Analysis," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. ACM Conferences, P. L. Lanzi, Ed. New York, NY, USA: ACM, 2011, pp. 829–836.

[10] P. Kerschke and H. Trautmann, *Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-Package Flacco*, ser. Studies in Classification, Data Analysis, and Knowledge Organization. Cham: Springer International Publishing, 2019, pp. 93–123.

[11] Q. Renau, C. Doerr, J. Dreo, and B. Doerr, "Exploratory Landscape Analysis is Strongly Sensitive to the Sampling Strategy," in *Parallel Problem Solving from Nature - PPSN XVI*, ser. Lecture Notes in Computer Science, T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, Eds. Cham: Springer International Publishing, 2020, vol. 12270, pp. 139–153.

[12] A. Thomaser, A. V. Kononova, M.-E. Vogt, and T. Bäck, "One-Shot Optimization for Vehicle Dynamics Control Systems: Towards Benchmarking and Exploratory Landscape Analysis," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 2036–2045.

[13] A. Thomaser, M.-E. Vogt, A. V. Kononova, and T. Bäck, "Transfer of Multi-objectively Tuned CMA-ES Parameters to a Vehicle Dynamics Problem," in *Evolutionary Multi-Criterion Optimization*, M. Emmerich, A. Deutz, H. Wang, A. V. Kononova, B. Naujoks, K. Li, K. Miettinen, and I. Yevseyeva, Eds. Cham: Springer Nature Switzerland, 2023, pp. 546–560.

[14] B. van Stein, H. Wang, and T. Bäck, "Neural network design: learning from neural architecture search," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 1341–1349.

[15] F. X. Long, D. Vermetten, A. V. Kononova, R. Kalkreuth, K. Yang, T. Bäck, and N. van Stein, "Challenges of ela-guided function evolution using genetic programming," in *Proceedings of the 15th International Joint Conference on Computational Intelligence - ECTA*, INSTICC. SciTePress, 2023, pp. 119–130, arXiv:abs/2305.15245.

[16] F. X. Long, B. van Stein, M. Frenzel, P. Krause, M. Gitterle, and T. Bäck, "Learning the Characteristics of Engineering Optimization Problems with Applications in Automotive Crash," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022.

[17] C. Benjamins, A. Jankovic, E. Raponi, K. van der Blom, M. Lindauer, and C. Doerr, "Towards automated design of bayesian optimization via exploratory landscape analysis," *arXiv preprint arXiv:2211.09678*, 2022.

[18] B. Van Stein, F. X. Long, M. A. Frenzel, P. Krause, M. Gitterle, and T. Bäck, "Doe2vec: Representation learning for exploratory landscape analysis," 2022.

[19] M. V. Seiler, P. Kerschke, and H. Trautmann, "Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems," *arXiv preprint arXiv:2401.01192*, 2024.

[20] H.-J. Koch-Dücker and U. Papert, "Antilock braking system," in *Brakes, Brake Control and Driver Assistance Systems*, ser. Bosch professional automotive information, K. Reif, Ed. Wiesbaden: Springer Vieweg, 2014, pp. 74–93.

[21] International Organization for Standardization, "ISO 21994:2007 - Passenger cars - Stopping distance at straight-line braking with ABS - Open-loop test method," 2007.

[22] A. Thomaser, M.-E. Vogt, T. Bäck, and A. Kononova, "Real-World Optimization Benchmark from Vehicle Dynamics: Specification of Problems in 2D and Methodology for Transferring (Meta-)Optimized Algorithm Parameters," in *Proceedings of the 15th International Joint Conference on Computational Intelligence*. SCITEPRESS - Science and Technology Publications, 2023, pp. 31–40.

[23] F. Duddeck, "Multidisciplinary optimization of car bodies," *Structural and Multidisciplinary Optimization*, vol. 35, no. 4, pp. 375–389, 2008.

[24] J. Fang, G. Sun, N. Qiu, N.-H. Kim, and Q. Li, "On design optimization for structural crashworthiness and its state of the art," *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 1091–1119, 2017.

[25] Livermore Software Technology Corporation, "Ls-dyna theory manual," Jul. 2019. [Online]. Available: https://ftp.lstc.com/anonymous/outgoing/jday/manuals/DRAFT_Theory.pdf

[26] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions," INRIA, Research Report RR-6829, 2009. [Online]. Available: https://hal.inria.fr/inria-00362633

[27] R. P. Prager and H. Trautmann, "Nullifying the inherent bias of non-invariant exploratory landscape analysis features," in *Applications of Evolutionary Computation: 26th European Conference, EvoApplications 2023, Held as Part of EvoStar 2023, Brno, Czech Republic, April 12–14, 2023, Proceedings*. Springer, 2023, pp. 411–425.

[28] ——, "Pflacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems in Python," *Evolutionary Computation*, pp. 1–25, 07 2023.

[29] Q. Renau, J. Dreo, C. Doerr, and B. Doerr, "Expressiveness and robustness of landscape features," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2048–2051.

[30] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[31] N. van Stein, D. Vermetten, A. V. Kononova, and T. Bäck, "Explainable benchmarking for iterative optimization heuristics," *arXiv preprint arXiv:2401.17842*, 2024.