Natural Computing (2022) 21:131–154
https://doi.org/10.1007/s11047-021-09847-1

Analyzing randomness effects on the reliability of exploratory landscape analysis

Mario Andrés Muñoz[1] · Michael Kirley[2] · Kate Smith-Miles[1]

Accepted: 27 January 2021 / Published online: 12 February 2021

Abstract

The inherent difficulty of solving a continuous, static, bound-constrained and single-objective black-box optimization problem depends on the characteristics of the problem's fitness landscape and the algorithm being used. Exploratory landscape analysis (ELA) uses numerical features generated via a sampling process of the search space to describe such characteristics. Despite their success in a number of applications, these features have limitations related with the computational costs associated with generating accurate results. Consequently, only approximations are available in practice which may be unreliable, leading to systemic errors. The overarching aim of this paper is to evaluate the reliability of five well-known ELA feature sets across multiple dimensions and sample sizes. For this purpose, we propose a comprehensive experimental methodology combining exploratory and statistical validation stages, which uses resampling techniques to minimize the sampling cost, and statistical significance tests to identify strengths and weaknesses of individual features. The data resulting from the methodology is collected and made available in the LEarning and OPtimization Archive of Research Data v1.0. The results show that instances of the same function can have feature values that are significantly different; hence, non-generalizable across instances, due to the effects produced by the boundary constraints. In addition, some landscape features under evaluation are highly volatile, and strongly susceptible to changes in sample size. Finally, the results show evidence of a curse of modality, meaning that the sample size should increase with the number of local optima.

Keywords Black-box optimization · Bound-constrained optimization · Continuous optimization · Exploratory landscape analysis · Single-objective optimization

1 Introduction

The expectation that a good solution to a continuous, static, bound-constrained and single-objective black-box optimization problem can be found reasonably fast by a given algorithm depends on the characteristics of the problem's fitness landscape (Sala and Müller 2020). The 'hardness' of such problems is related to landscape characteristics such as modality, smoothness and variable separability, and the way that such characteristics are exploited by an algorithm. Therefore, by quantifying the landscape characteristics we can potentially identify which algorithm works 'best' on a given problem. *Exploratory Landscape Analysis* (ELA) is an umbrella term for a range of sample-based methods for measuring the landscape characteristics of a problem, which generate one or more numerical results describing particular characteristics, called *features*. ELA has been successfully used for identifying strengths and weaknesses of algorithms (Malan and Engelbrecht 2014), automatic algorithm selection (Bischl et al. 2012a; Kerschke and Trautmann 2019a), per instance algorithm configuration (Belkhir et al. 2016a, 2017) and generation (Miranda et al. 2017) methods, and benchmark construction techniques (Muñoz and Smith-Miles 2020).

However, ELA features have limitations worth acknowledging if landmines are to be avoided. For example, their assumptions are valid on simple instances, where

✉ Mario Andrés Muñoz
munoz.m@unimelb.edu.au

[1] School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia

[2] School of Computer and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia

it is easier to isolate a specific characteristic. New information cannot be added seamlessly without incurring a significant bias or computational cost (Muñoz et al. 2012). Identifying features that correlate with algorithm performance is difficult, time consuming and not always intuitive (Alissa et al. 2019), with some being more predictive than others. Since we currently lack a good understanding of which features are the most relevant (Renau et al. 2019), a new feature may be needed each time a different algorithm is being considered, or a characteristic is found to be influential. Moreover, some features are hard to interpret, limiting the provision of meaningful explanations and restricting their usage as inputs to machine learning-based selection or configuration models. If none of the algorithms under consideration are recommended for a problem using algorithm selection methods (Smith-Miles et al. 2014), and we need to design a new algorithm, our ability to gain required insights from the features is severely impaired. Finally, but not least importantly, as the dimension of the instance increases, so too does the computational cost of the feature calculations measured in terms of function evaluations (Müller and Sbalzarini 2011; He et al. 2007). Therefore, the cost of calculating accurate features can be greater than the cost of running an algorithm to solve the problem (Beck and Freuder 2004; He et al. 2007), and only approximate features can be calculated in practice (He et al. 2007). Although approximated features can be used for automatic algorithm selection and configuration (Kerschke et al. 2016; Belkhir et al. 2016a, 2017; Kerschke and Trautmann 2019a), they are random variables whose distribution depends on the function instance, the sample size, and even the sample generator (Renau et al. 2020). Hence, it is necessary to have a sufficiently large sample such that the median of the distribution has converged.

In order to avoid the kinds of systemic errors introduced by uncertainties in approximated features, and to ensure the simplicity and efficiency of systems relying on such features, we propose some reliability criteria. An approximated feature is *reliable* if it: (a) produces useful information, (b) is free of vulnerabilities that could lead to inaccurate interpretations, (c) has low variance, (d) is different between functions, (e) is stable across instances of the same function generated through translations or rotations, (f) converges quickly, and (g) is uncorrelated with other features. Unfortunately, there is limited literature exploring the reliability of approximated features in detail, specifically the strengths and weaknesses of the various types of ELA features. Muñoz and Smith-Miles (2015) explored the effect that translations had on the features, when the cost function is bound-constrained, demonstrating that translations led to phase transitions; hence, providing evidence of non-generality of the features across instances. Renau et al. (2019) explored the robustness of features against the random sampling process, the number of sample points, and the expressiveness in terms of their ability to discriminate between problems. Focusing on a fixed dimension of five and seven feature sets, they determined that most features are not robust against the sampling method, and are similar for several function pairs. Saleem et al. (2019) proposed a method to evaluate features based on a ranking of similar problems and Analysis of Variance (ANOVA), which does not require machine learning models or confounding experimental factors. Focusing on 12 features, four benchmark sets in two- and five-dimensions, and four one-dimensional *transformed* functions, they identified that not a single feature is capable of identifying all the landscape characteristics but some features prove valuable in capturing certain characteristics of the landscape. Moreover, they emphasize the necessity to examine the variability of a feature with different sample sizes, as some can be estimated with small sizes, while others not. Finally, Škvorc et al. (2020) used ELA to develop a generalized method for visualizing a set of arbitrary optimization functions, focusing on two- and ten-dimensional functions. By applying feature selection, they showed that many features are redundant and most are non-invariant to simple transformations such as scaling and shifting.

In this paper, we advance the concepts studied in these previous works, by exploring the reliability of five well-known ELA feature sets on a larger set of dimensions, across multiple sample sizes. For this purpose, we propose a systematic experimental methodology combing exploratory and statistical validation stages, which uses statistical significance tests and resampling techniques to minimize the computational costs, both in collection time and storage space. Assuming that it is a random variable, a feature with a high-variance distribution may exhibit extreme changes across experiments due to the sample points collected, making it unreliable. Moreover, estimating the distribution allows us to determine whether the difference in value between instances or functions is statistically significant. The data resulting from the proposed methodology, applied to the *Comparing Continuous Optimizers* (COCO) noiseless benchmarks, is collected and made available in the LEarning and OPtimization Archive of Research Data (LEOPARD) v1.0 (Muñoz 2020)—another contribution of this work. The results demonstrate that some features are highly volatile for particular functions and sample sizes. In addition, a feature may have significantly different values between two instances of a function due to the bounds of the input space. This implies that the results from an instance should not be generalized across all instances. Finally, the results show evidence of a *curse of modality*, which means that the sample size should increase with the number of local optima in the function.

This paper is organized as follows. Section 2 presents the details of the five feature sets under evaluation: fitness distance correlation (Jones and Forrest 1995), dispersion (Lunacek and Whitley 2006), fitness distribution analysis (Mersmann et al. 2011; Marin 2012), model fitting (Mersmann et al. 2011), and information significance (Seo and Moon 2007). Section 3 presents the proposed experimental methodology, and describes the data contained in LEOPARD. Results are presented in Sect. 4. The implications of the findings are presented in Sect. 5. The paper finalizes with the conclusions in Sect. 6.

2 Exploratory landscape analysis

Exploratory landscape analysis are methods used to measure specific problem landscape characteristics thought to be related to the difficulty of the underlying optimization task, i.e., modality, smoothness, global structure, variable scaling and separability (Muñoz et al. 2015b), through feature sets. Figure 1 maps the feature sets evaluated in this paper to the landscape characteristics they measure. Before presenting the technical details of each set, we define our notation. Without loss of generality for maximization, a continuous, static, bound-constrained and single-objective optimization problem is a function to be minimized, $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^D$ is the *input space*, $\mathcal{Y} \subset \mathbb{R}$ is the *output space*, and $D \in \mathbb{N}^*$ is the *dimensionality* of the problem. A *sample point* $\mathbf{x} \in \mathcal{X}$ is a D-dimensional vector, and $y \in \mathcal{Y}$ is the sample point's *cost* or *fitness*. The set of *global optima* is defined as $\{\mathbf{x}_O \in \mathcal{X} : \forall \mathbf{x} \in \mathcal{X}, f(\mathbf{x}) \geq f(\mathbf{x}_O)\}$. Let $\mathbf{X} \subset \mathcal{X}$ be an *input sample* of size n, and the *output sample*, $\mathbf{Y} \subset \mathcal{Y}$, be the result of evaluating $\mathbf{X}$ in f.

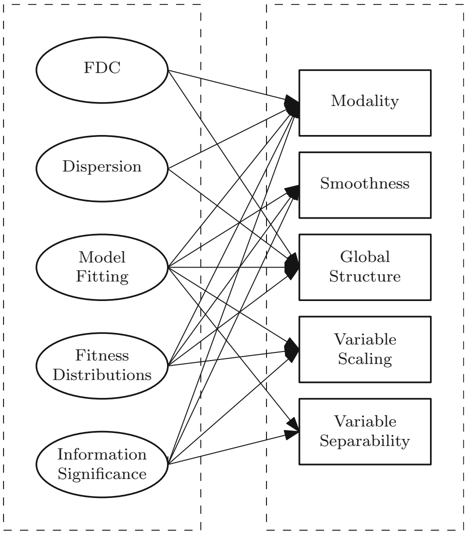


Fig. 1 A mapping of the feature sets (left) to landscape characteristics (right) used in this study. Here, five different sets are used to quantify five different landscape characteristics. It should be noted that the features must be used in combination to paint a clear picture of the underlying problem landscape

Fitness distance correlation (Jones and Forrest 1995) measures the relationship between the location in the input space and the fitness value. It aims to identify whether a landscape is unimodal or multimodal, and whether it has a strong global structure or not. It is defined as the Pearson correlation between the fitness value y_i and the Euclidean distance, d_i, between $\mathbf{x}_o$ and $\mathbf{x}_i$ with $\mathbf{x}_o$ approximated by the best point from the sample (Müller and Sbalzarini 2011), as follows:

$$FDC = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{y_i - \bar{y}}{\hat{\sigma}_y} \right) \left(\frac{d_i - \bar{d}}{\hat{\sigma}_d} \right) \quad (1)$$

where $\bar{y}$ and $\bar{d}$ are the mean fitness and the mean distance between $\mathbf{x}_o$ and $\mathbf{x}_i$ respectively, $\hat{\sigma}_y$ and $\hat{\sigma}_d$ are the sample standard deviation of the fitness and the distance respectively. Arguably, *FDC* is invariant to shifts and rotations on the input space, because they are *global isometries* of the Euclidean space, i.e., shifts and rotations do not affect the distance between sample points. *Dispersion* (Lunacek and Whitley 2006), like *FDC*, is a measure of the relationship between the location of the samples in the input space and the fitness value. It is defined as the normalized average Euclidean distance between the $qn, q < 1$ fittest sample points. The assumption behind this feature is that points taken from well correlated landscapes have similar fitness if they are close to each other. The feature is normalized over the diagonal of the input space as follows:

$$DISP_q = \frac{1}{\|\mathbf{x}_{\max} - \mathbf{x}_{\min}\|} \frac{1}{qn} \sum_{i=1}^{qn} \sum_{j=1}^{qn} d(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$

where $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ are the upper and lower bounds of the input space respectively, and q is the truncation level between the $[0, 1]$ range, which we set to $q = \{0.01, 0.02, 0.05, 0.07, 0.10, 0.20, 0.50\}$. As it is the case with *FDC*, it can be proven that $DISP_q$ is theoretically invariant to shifts and rotations on the input space, because they are global isometries of the Euclidean space.

Fitness distributions is a powerful approach to assess the complexity (or hardness) of a problem, as the fitness probability distribution is independent from the representation of the input space (Rosé et al. 1996). According to Mersmann et al. (2011), the level of smoothness and the global structure of the function can be characterized through the *skewness* and *kurtosis*, which are the standardized third and fourth moments of the distribution. Skewness is a measure of the symmetry of the distribution, with a negative value indicates that the distribution's tail is heavier on the left side, where as a positive value indicates that the tail is heavier on the right side. On the other hand, kurtosis measures the propensity of the distribution to produce outliers without identifying aspects of the peak. The sample skewness, γ, and kurtosis, κ, are both calculated using their unbiased estimators, which in turn are derived from expansions of the unbiased estimators of the third and fourth *cumulants*, $\{k_3, k_4\}$ and the standard deviation, k_2, as follows:

$$\gamma \approx \frac{k_3}{k_2^{3/2}} = \frac{\sqrt{n(n-1)}}{n-2}\,\frac{\frac{1}{n}\sum_{i=1}^{n}\Delta y_i^3}{\left(\frac{1}{n}\sum_{i=1}^{n}\Delta y_i^3\right)^{3/2}} \tag{3}$$

$$\kappa \approx \frac{k_4}{k_2^4} = \frac{n-1}{(n-2)(n-3)} \left((n+1)\,\frac{\frac{1}{n}\sum_{i=1}^{n}\Delta y_i^4}{\left(\frac{1}{n}\sum_{i=1}^{n}\Delta y_i^4\right)^2} - 3(n-1)\right) + 3 \tag{4}$$

where $\Delta y_i = y_i - \bar{y}$. Moreover, it might be of use to quantify the complexity of the distribution instead of characterizing the shape, through the *entropy*, $H(\mathbf{Y})$ (Marin 2012), which we estimate using the *kd*-tree partition method (Stowell and Plumbley 2009).

Model fitting is another approach that can also be used to analyze the modality and the global structure of the landscape using linear or quadratic regression (Mersmann et al. 2011). Model fitting can be thought of as measuring the distance between a reference problem and the problem under analysis (Graff and Poli 2010). In addition, included interaction terms in the fitted model to add information about linear variable dependencies, which it is equivalent to measure linear variable dependencies (Rochet et al. 1996; Davidor 1991). In total, we train four models: linear model without interactions, *L*, linear model with interactions, *LI*, quadratic model without interactions, *Q*, and quadratic model with interactions *QI* (Mersmann et al. 2011).

As an example, assume that we are fitting a two-dimensional quadratic regression model with variable interactions, i.e., $y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{1,i} + \beta_3 x_{1,i} x_{2,i} + \beta_4 x_{1,i}^2 + \beta_5 x_{2,i}^2$, where $\boldsymbol{\beta} = \{\beta_0, \beta_1, \ldots, \beta_5\}$ are the estimated regression coefficients and $\mathbf{x}_i = \{x_{1,i}, x_{2,i}\}$ is a sample point. We fit the model using least squares, such that $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{XY}$. The fit of this model is measured using the *Adjusted Coefficient of Determination*, $\bar{R}^2$, which is calculated using Eq. (5) where $\hat{y}_i$ is the estimated fitness, $\bar{y}$ is the mean of the observed fitness, and $|\boldsymbol{\beta}|$ is the cardinality of the estimated coefficients. In our example, $|\boldsymbol{\beta}| = 6$.

$$\bar{R}^2 = 1 + \frac{|\boldsymbol{\beta}|}{n - |\boldsymbol{\beta}| - 1}\,\frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{5}$$

In addition, to estimate the level of variable scaling we measure the minimum and maximum of the absolute value of the linear model coefficients, i.e., $\min(|\beta_L|)$ and $\max(|\beta_L|)$, which indicate the most extreme scales of the variables; and the ratio between the minimum and the maximum absolute values of the quadratic term coefficients in the quadratic model, i.e., $\min\left(|\beta_Q|\right)/\max\left(|\beta_Q|\right)$, which is an indicator of the conditioning of the function (Mersmann et al. 2011). *Information significance* is a method used to measure variable dependency or *Epistasis* (Seo and Moon 2007), which is an important characteristic, as a careful analysis of variable interactions suggests that a given problem might be broken down into simpler problems of lower dimensionality (Pošík 2005; Mersmann et al. 2010). Most methods proposed for Epistasis, assume linear interactions between variables (Davidor 1991; Naudts et al. 1997; Fonlupt et al. 1998; Rochet et al. 1998). In contrast, information significance measures non-linear variable dependencies based on mutual information, expressing dependency as the joint probability of a group of variables relative to the probability of each variable. To use this method, let $\mathcal{V} = \{1, \ldots, D\}$ be a set of variable indexes, where D is the dimension of the problem, $v \in \mathcal{V}$ is the index of one of such variables, and $V \subset \mathcal{V}$ is a combination of such variables. The information significance of a variable combination V, $\xi(V)$, is the uncertainty coefficient calculated using Eq. (6), where $\hat{H}(\mathcal{Y})$ is the estimated information entropy and $\hat{I}(\mathcal{X}_V; \mathcal{Y}) = \hat{H}(\mathcal{X}_V) + \hat{H}(Y) - \hat{H}(\mathcal{X}_V, Y)$ is the estimated mutual information.

$$\xi(V) = \frac{\hat{I}(\mathcal{X}_V; \mathcal{Y})}{\hat{H}(\mathcal{Y})} \tag{6}$$

To summarize the results, we calculate the mean information significance, $\xi^{(k)}$, of order k using Eq. (7) (Seo and Moon 2007), where $k = |V|$.

$$\xi^{(k)} = \frac{1}{\binom{D}{k}} \sum_{V \subset \mathcal{V}, |V| = k} \xi(V). \tag{7}$$

The method is dependent on $\hat{H}(\mathcal{Y})$, which we estimate using the *kd*-trees (Stowell and Plumbley 2009). However, the result is not bounded between $[0, 1]$ as it is for binary spaces. Because the number of possible variable combinations increases with D, we only calculate the information significance of first, $\zeta^{(1)}$, second, $\zeta^{(2)}$ and D-th, $\zeta^{(D)}$ orders.

We conclude this brief review of ELA feature sets with an acknowledgment that other sets have been proposed in the literature (Kerschke and Trautmann 2019b). However, all the feature sets described above are *cheap* i.e., they share the advantage that any sample of size n is required to calculate them all; hence, reducing the overall computational cost of our experiments (Muñoz et al. 2015b). By definition, and as experimentally demonstrated by Škvorc et al. (2020), these features are expected to be invariant to scaling or shifting of the function by a constant, as their values are calculated relative to the average cost that includes such constants.

3 Experimental methodology

To investigate the reliability of the features, we propose a two-stage experimental methodology spanning exploratory and statistical analysis stages. Table 1 provides a high-level overview of this methodology, by listing specific questions related to performance characteristics of the features considered and a brief outline of the approach used to answer the questions.

The aim of the *exploratory* validation stage is to demonstrate that the feature fulfills its stated objectives. We are particularly interested in identifying vulnerabilities, in other words, we wish to answer questions such as “when is the feature likely to fail or have similar results to other features?”, and “how easy it is to produce an inaccurate interpretation?”.

The aim of the *statistical* validation stage is to assess the reliability of the feature under varying experimental conditions. Our main premise is that a feature $c_k(f, n)$ for a given function instance f calculated from a sample of size n is a random variable, with randomness being originated by the sampling or feature calculation procedures and not the function itself. Therefore, it can be also defined as $c_k(f, n) \equiv T((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n))$, where $(\mathbf{x}_i, y_i), i = 1, \ldots, n$ are also random variables. As such, c_k has a probability distribution whose variance, σ_k^2, should converge to zero when $n \rightarrow \infty$, otherwise σ_k^2 is dependent on both f and n. The statistical validation focuses on three aspects:

1. Measuring the magnitude of σ_k^2. A high value indicates that the feature is unreliable, as it changes across experiments for a particular function due to sampling.
2. Testing the significance of the difference in feature’s values across functions, instances or sample sizes. Differences between two features that are not statistically significant indicate that the features may converge to the same value.
3. Detecting linear correlations between features. Strong correlation, either positive or negative, implies that the features could be equivalent.

The MATLAB implementation of the *Comparing Continuous Optimizers* (COCO) noiseless benchmarks v13.09 (Hansen et al. 2014) are used in our experiments. The motivation for this choice of benchmark problems revolves around practical advantages. For example, there is a wealth of data collected about the performance of a large set of search algorithms, and there are established conventions on the number of dimensions and the limits on number of function evaluations. In addition, the software implementation of the benchmark set generates instances by translating and rotating the function in the input and output spaces. Let $f_{k,l}$ be an instance $k = 1, \ldots, 15$ of the test functions $l = 1, \ldots, 24$. The functions in the COCO set are scalable with the dimension D, which we set to $D = \{2, 5, 10, 20\}$. Ideally, if the features are to be used for algorithm selection, their additional computational cost should be a fraction of the budget allocated to a single search algorithm, which is usually bounded within COCO at $10^4 \times D$ function evaluations (Hansen et al. 2014).

Table 1 A summary of the experimental methodology used to establish the reliability of landscape features. Answers to the seven key questions provides important insight into the reliability of particular landscape characteristics

Stage	Key question	Approach
I. Exploratory	Is the feature fulfilling its stated objectives?	Demonstrate that the feature can produce useful information
	Is the feature free of major vulnerabilities?	Find counter examples to highlight potential issues
II. Statistical	Is the feature non-volatile?	Calculate the feature’s variance
	Is the feature different between functions?	Test the significance of the differences across functions
	Is the feature stable across instances of the same function?	Test the significance of the differences across instances
	Does the feature converge quickly?	Test the significance of the differences across sample sizes
	Is the feature uncorrelated with other features?	Check for correlations between features

Belkhir et al. (2016b) and Kerschke et al. (2016) use sample sizes of $30 \times D$ and $50 \times D$ respectively, which are close to the population size of an evolutionary algorithm. However, Belkhir et al. (2016b) establishes that $30 \times D$ produces poor approximations of the feature values, although they can be improved by training and resampling a surrogate model. Nevertheless, their experiments also demonstrate that most features for the COCO benchmark set level-off between $10^2 \times D$ and $10^3 \times D$. These results are supported by Škvorc et al. (2020), who determined that for $D = 2$ a sample size of at least $200 \times D$ was necessary to guarantee convergence.

Given this evidence and to balance the cost of our computations, we set the lower bound for the sample size at $10^2 \times D$, corresponding to 1% of the budget, and the upper bound at $10^3 \times D$, corresponding to 10% of the budget, which we consider to be reasonable sample sizes. We divided the range between $10^2 \times D$ and $10^3 \times D$ into five equally sized intervals in base-10 logarithmic scale, with the objective of producing a geometric progression analogous to the progression in D. As a result, we have five samples for each dimension. The samples have sizes n equal to $\{100, 178, 316, 562, 1000\} \times D$ points, where each one is roughly 80% larger than the previous. We generate the input samples, $\mathbf{X}$, using MATLAB's Latin Hypercube Sampling (LHS) function `lhsdesign` with default parameters.

An output sample, $\mathbf{Y}_{k,l} \subset \mathcal{Y}$, is the result of evaluating $\mathbf{X}$ in one of the first 15 instances of the 24 functions from the COCO benchmark. This data generation procedure guarantees that the differences observed on the features depend on the output sample and not on the input sample; hence, it eliminates a source of uncertainty. As D increases, the size of the $\mathbf{X}$ is relatively smaller with respect to the size of $\mathcal{X}$. This is an unavoidable limitation due to the curse of dimensionality.

A *trial set*, $\mathbf{Z}_{k,l}$, is the combination of input and output samples; therefore, a trial set of n points has $D + 1$ variables. For example, a trial set for a two-dimensional function has three variables: the two input variables, $[x_{1,i}, x_{2,i}]$, and the output variable, y_i. With one trial set for each instance of each function, at each dimension and for each sample size, we have a total of 7200 trial sets to analyze $(24 \times 15 \times 4 \times 5 = 7200)$.

To calculate the variance, we estimate the *empirical probability distribution* of each feature, $\widehat{pr}(c_k)$. There are multiple approaches to this problem. For example, to take multiple, independent trial sets per function (Renau et al. 2019), which would guarantee the most accurate estimator of $\widehat{pr}(c_k)$. However, this comes at the substantial computational of cost collecting the function responses and calculating the features, particularly as n and D increase. Another approach is to train a surrogate model with a small sample, and then resample from the model (Belkhir et al. 2016b), which would provide a low computational cost estimate of $\widehat{pr}(c_k)$. However, the resample also includes assumptions generated by the surrogate that may not correspond to the actual function. Moreover, parametric assumptions cannot be made, and there is no guarantee of fulfilling asymptotic convergence. Therefore, we use *bootstrapping* (Efron and Tibshirani 1993) to estimate $\widehat{pr}(c_k)$, which is a type of resampling method that uses the empirical distribution to learn about the population distribution as follows: Let $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}, \mathbf{z}_i = (\mathbf{x}_i, y_i), i = 1, \ldots, n$ be a sample of n independent and identically distributed random variables drawn from the distribution $pr(\mathbf{Z})$. We can consider $c_k = T(pr(\mathbf{Z}))$ to be a *summary statistic* of $pr(\mathbf{Z})$ for example the mean, the standard deviation, or in our case an ELA feature. For each $j = 1, \ldots, N$, let $\mathbf{Z}^*_j$ be a *bootstrap sample*, which is a new set of n points independently sampled with replacement from $\mathbf{Z}$. From each one of the N bootstrap samples, we calculate a *bootstrap statistic*, $\hat{c}^*_{k,j}$ also in our case a feature. The set of N bootstrap statistics is used to find the *bootstrap distribution*, $\widehat{pr}^*(c_k)$. We estimate the variance of the bootstrap distribution, $\hat{\sigma}^2(\hat{c_k})$, as follows:

$$\hat{\sigma}^2(c_k) = \frac{1}{N}\sum_{j=1}^{N}\left(c^*_{k,j}\right)^2 - \left(\frac{1}{N}\sum_{j=1}^{N} c^*_{k,j}\right)^2 \tag{8}$$

where $N = 2000$. In other words, from each sample of size n we create N resamples with replacement, also of size n, from which we estimate all the features under examination.

To identify whether the difference in a feature is statistically significant, we employ the Wilcoxon rank-sign test at the 95% significance level. The null hypothesis is that the median of the two distributions are equal. Ideally, the difference should be significant when the features for two different problems are compared. The opposite indicates that the feature does not provide useful information because it converges to the same value regardless of the function. The difference should not be significant when the features are from two instances of the same problem, unless the feature captures variable dependencies; or when the feature is calculated with two samples of different sizes, which indicates that the feature starts to converge at small sample sizes. To illustrate the importance of significance testing, Fig. 2 shows a kernel estimate of the probability density function of $\{FDC, \gamma(\mathbf{Y})\}$ for the first five instances of the f_1 function from COCO, which were generated through translation. Although the features have different mean values, their probability distribution overlap, implying that the difference may not be significant.

To mitigate false rejection errors, we use Benjamini and Yekuteli's method (Benjamini and Yekutieli 2001) to correct the p-values resulting from the tests over each

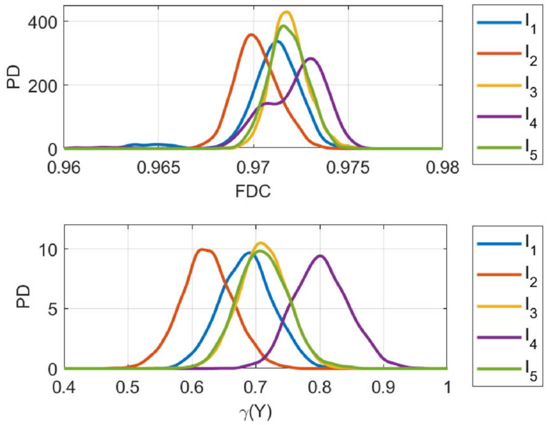


Fig. 2 Kernel estimate of the probability density function for two related features: *FDC* and $\gamma(\mathbf{Y})$ for the first five instances of the f_1 function from the COCO benchmarks functions. The overlap between probability density functions indicates that the difference in their mean values could be insignificant

family, setting the *False Discovery Rate*, *FDR*, at 5%. If the method rejects some hypotheses using conventionally low alpha levels such as 5%, we can be confident that the number of false discoveries is much smaller than the number of correct rejections (Groppe et al. 2011). We summarize the results by counting the percentage of tests which are statistically significant for each function at each dimension. The methodology assumes that the features fulfill the requirements for bootstrapping; $\widehat{pr}^*(c_k)$ is a good estimate of $\widehat{pr}(c_k)$; and c_k is not biased, which means that the median of $\widehat{pr}^*(c_k)$ converges to the real value of c_k.

Bootstrapping has limitations due to resampling with replacement. Although the size of $\mathbf{Z}^*$ is the same than $\mathbf{Z}$, asymptotically only 63.2% of the points are unique in $\mathbf{Z}^*$ (Bischl et al. 2012b). Therefore, some areas may be better represented in the sample than others. We assume that this effect is ameliorated by having a large value of N. Moreover, a LHS is a type of stratified sampling, which may not fulfill the independence assumption within bootstrapping. The Appendix demonstrates that this assumption holds experimentally, and there is no practical difference on the results between taking N uniformly distributed random samples of n points and bootstrapping N times a LHS of n points. Nevertheless, it should be acknowledged that a bootstrapped LHS is strictly no longer a LHS; hence, we can expect its variance reduction properties to be affected (Stein 1987). Combining bootstrapping with LHS or pseudo-random sampling methods is not uncommon on system simulation experiments where data could be expensive to obtain (Storlie et al. 2009; Tian et al. 2014), as it reduces the cost of collecting and storing data. For example, storing 2000 points of a 2D function in double precision require approximately 48kB of memory $(2000 \times 3 \times 8)$. Replicating this experiment 2000 times would require approximately 96MB of memory in total, whereas generating 2000 bootstrap indexes stored as 16-bit unsigned integers would require 8MB $(2000 \times 2000 \times 2)$ plus the original 48kB of memory for the data. This corresponds to approximately 8% of the storage space, without accounting any saving due to compression.

Finally, we calculate the Pearson correlation between each of the features. We follow a rule of thumb that considers a correlation between 0.7 and 0.9 to be high, and between 0.9 and 1.0 to be very high (Hinkle et al. 2003). With a dataset of 7200 different values of each feature, there is enough evidence to identify whether one feature is correlated with another.

The experimental data, consisting of the input and output samples, indexes of the bootstrap samples, and feature values, their bootstrapped values and summary statistics of $\widehat{pr}(c_k)$, such as the feature variances, is collected in the LEarning and OPtimization Archive of Research Data (LEOPARD) v1.0 (Muñoz 2020) as compressed CSV files. All data was originally generated using MATLAB 2012b.

4 Results

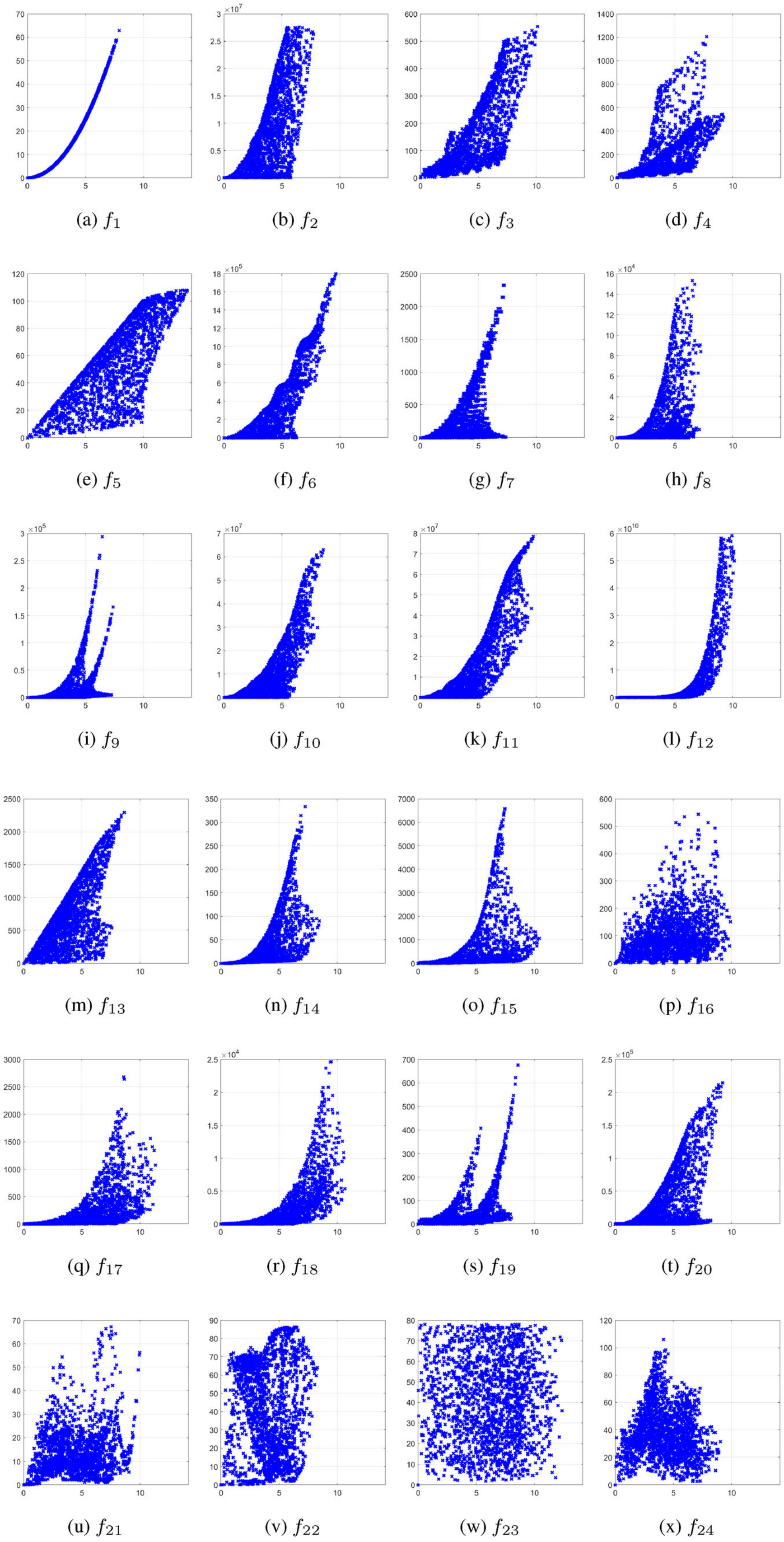
We present the results from our experiments for both the exploratory and statistical validation stages. In the exploratory validation stage, the overarching aim was to identify whether a landscape feature fulfills its stated objectives. We were particularly interested in determining whether the feature generates similar results to other features, and importantly, how features can be misinterpreted. To highlight the vulnerabilities, a series of unique experiments were identified and examined using selected test functions from the COCO benchmarks. We describe the details of the experiment before presenting the results in Sect. 4.1. In the statistical validation stage, quantifying reliability of features was the primary goal. That is, the aim was to identify the comparative advantages and disadvantages of each feature, by analyzing the relationships between them, and the statistical significance of their differences. We present the results of the statistical stage in Sect. 4.2.

4.1 Exploratory validation stage

4.1.1 Fitness distance correlation

We focus our exploratory validation of *FDC* on determining whether two instances of the same function may have similar values of *FDC*; hence, its interpretation provides clues about the complexity of the function. We collected a sample of $D \times 10^3$ points that is then evaluated on

Fig. 3 Fitness-distance clouds for the first instance of the functions from the COCO benchmark at $D = 2$. The horizontal axis represents the distance from the estimated global optimum, whereas the vertical axis represents the fitness



(a) f_1 (b) f_2 (c) f_3 (d) f_4

(e) f_5 (f) f_6 (g) f_7 (h) f_8

(i) f_9 (j) f_{10} (k) f_{11} (l) f_{12}

(m) f_{13} (n) f_{14} (o) f_{15} (p) f_{16}

(q) f_{17} (r) f_{18} (s) f_{19} (t) f_{20}

(u) f_{21} (v) f_{22} (w) f_{23} (x) f_{24}

the first instance from each one of the functions from the COCO benchmark at $D = 2$. The results are illustrated in Fig. 3 using scatter plots of the fitness, y, on the vertical axis against the distance d between $\mathbf{x}_o$ and $\mathbf{x}_i$ on the horizontal axis. We call this representation *Fitness-Distance cloud*.

For most functions, the fitness increases as the distance from the estimated global optimum increases, except for functions $\{f_{21}, f_{22}, f_{23}, f_{24}\}$. This trend is the most evident for f_1. We can infer that $FDC \approx 1$ for f_1, and that $FDC = 1$ for an inverse conical function, which is not included in the benchmark set. We observe that linear correlation is a poor summary of the complex shape that the Fitness-Distance cloud might have. Therefore, it might be inappropriate to provide an interpretation of *FDC*, such as the one by Müller and Sbalzarini (2011).

We evaluate the sample extracted above on the first 15 instances of ten selected functions from the COCO benchmark at $D = 2$, two from each one of the groups defined by (Hansen et al. 2011a). These functions are $\{f_1, f_4, f_6, f_9, f_{10}, f_{13}, f_{15}, f_{19}, f_{23}, f_{24}\}$. We calculated the *FDC* for these functions, and we classify the functions into five categories depending on its *FDC* value and the rule of thumb by Hinkle et al. (2003) for the interpretation of the Pearson correlation. The categories are: (1) Very high $[0.9, 1.0]$, (2) High $[0.7, 0.9]$, (3) Moderate $[0.5, 0.7]$, (4) Low $[0.3, 0.5]$ and (5) Little $[-0.3, 0.3]$. The minimum and maximum value of *FDC* among the first 15 instances, and the category to which they belong are presented in Table 2. The distributions of the values are presented as box-plots in Fig. 4.

Table 2 Minimum and maximum values of the *FDC* for the first 15 instances of ten selected functions from the COCO benchmark, and their classification according to the rule of thumb proposed by Hinkle et al. (2003)

	min	Class	max	Class
f_1	0.969	(1)	0.973	(1)
f_4	0.159	(5)	0.717	(2)
f_6	− 0.290	(5)	0.904	(1)
f_9	0.316	(4)	0.519	(3)
f_{10}	0.295	(5)	0.863	(2)
f_{13}	0.288	(5)	0.900	(1)
f_{15}	0.368	(4)	0.865	(2)
f_{19}	0.216	(5)	0.604	(3)
f_{23}	− 0.036	(5)	0.030	(5)
f_{24}	− 0.143	(5)	0.733	(2)

Under the same experimental conditions, two instances of the same function can be classified in opposite classes

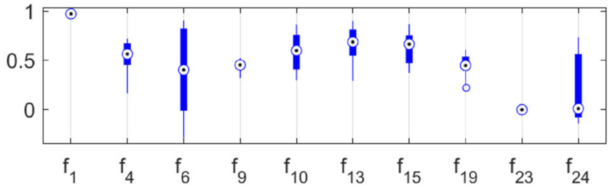


Fig. 4 Distribution of values of the *FDC* for the first 15 instances of ten selected functions from the COCO benchmark. The distribution demonstrates that the values from Table 2 do not correspond to extreme outliers

Table 2 and Fig. 4 suggest that the instances of the same function can be classified into different categories, implying that a function cannot be categorized correctly using the results from a single instance. For example, the Attractive Sector (f_6) and the Sharp Ridge (f_{13}) functions have at least one instance with little correlation and one with high correlation. There are functions, such as the Sphere (f_1) and the Katsuura (f_{23}) functions, whose categories do not change over the instances under analysis. In summary and answering the questions for Stage I from Table 1, *FDC* is a limited summary of the shape of the Fitness-Distance cloud, resulting on instances of the same function having *FDC* values that range from little to high correlation. Therefore, *FDC* is a reliable measure for the smoothest and ruggedest functions, but it is deceptive for those in between.

4.1.2 Dispersion

We focus our exploratory validation of $DISP_q$ on determining the influence of the truncation level, q, on the value of $DISP_q$. We expect that the difference between functions decreases as q increases. We extracted a $D = 10$ sample of 10^4 points. Then, we calculated the value of $DISP_q$ for the first instance of five functions from the COCO benchmark at $D = 10$. The functions are $\{f_1, f_9, f_{10}, f_{15}, f_{23}\}$, one from each one of the groups defined by (Hansen et al. 2011a). Figure 5 shows the results, illustrating the relationship between $\log_{10}(q)$ and the value of $DISP_q$. In addition, the figure shows a linear fit between $\log_{10}(q)$ and $DISP_q$.

Since the same input sample was used for all functions and the average distance between points is 0.401, then $DISP_q$ at $q = 1.00$ should converge also to 0.401. The figure shows that the linear fit does converge towards 0.401 for the five functions. In addition, this value is close to the theoretical convergence bound of $DISP_q$, which is equal to $1/\sqrt{6}$ (Morgan and Gallagher 2014). For function f_{23} the values of $DISP_q$ are close to the theoretical convergence level regardless of q. This result suggests that the function is either very rugged or neutral.

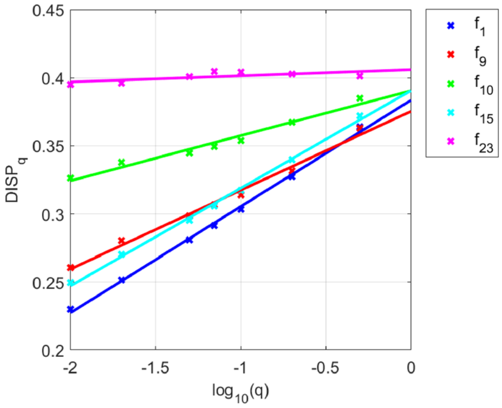


Fig. 5 Relationship between the truncation level and the value of the Dispersion for the first instance of five functions from the COCO benchmark at $D = 10$. On the horizontal axis is the truncation level, represented as $\log_{10}(q)$, whereas on the vertical axis is the Dispersion value. The figure also shows a linear fit of the relationship, which converges to 0.401 for $\log_{10}(q) = 0$, a close value to the theoretical convergence bound of $DISP_q$. Moreover, the highest discrimination between functions is achieved with $q = 0.01$, and may not be necessary to calculate $DISP_q$ at other values of q

The linear fit suggests that it is likely that the $DISP_q$ values are highly correlated. To verify this observation, we calculate the values of $DISP_q$ at $q = \{0.01, 0.02, 0.05, 0.07, 0.10, 0.20, 0.50\}$ for the first 15 instances at $\{2, 5, 10, 20\}$ dimensions and with a sample size of $\{100, 178, 316, 562, 1000\} \times D$ points. Then, we calculated the Pearson correlation between all the values of $DISP_q$. The results are presented in Table 3.

The results in the table show that the values of $DISP_q$ are almost perfectly correlated, i.e., $\rho_{x,y} \approx 1$, for most values of q. This suggest that it may not be necessary to calculate $DISP_q$ at different values of q, and perhaps only one level is sufficient. Since the difference in the values of $DISP_q$ between functions is inversely proportional to q, then the highest discrimination between functions is achieved with $q = 0.01$. This level has the additional advantage of being the least computational expensive value to calculate. In addition, we consider it adequate to calculate the value of $DISP_q$ for $q = 1.00$ to verify whether the average distance converges to the theoretical limit of $1/\sqrt{6}$.

Table 3 Pearson correlation between the values of $DISP_q$ for $q = \{0.01, \ldots, 0.50\}$

	0.02	0.05	0.07	0.10	0.20	0.50
0.01	0.98	0.95	0.93	0.92	0.90	**0.85**
0.02		0.98	0.97	0.96	0.94	**0.88**
0.05			1.00	0.99	0.97	0.91
0.07				1.00	0.98	0.92
0.10					0.99	0.93
0.20						0.97

In boldface are those values for which the correlation is high, i.e., $0.7 \leq \rho_{x,y} < 0.9$, while the remaining values are very high, i.e., $0.9 \leq \rho_{x,y} \leq 1.0$. The results show that all of the levels of $DISP_q$ are highly correlated, suggesting that only one level should be calculated

We shift the focus of the exploratory evaluation to determining whether two instances of the same function may have similar values of $DISP_{0.01}$. We calculated $DISP_{0.01}$ for the first 15 instances of the same ten functions that we analyzed for the *FDC*. The minimum and maximum value of $DISP_{0.01}$ from the 15 instances are presented in Table 4. The table includes the *relative difference* between the minimum and maximum value, *RD*, which is expressed as a percentage of the maximum. The *RD* help us to analyze the sensitivity of $DISP_{0.01}$ to translational shifts and orthogonal rotations of a function. Moreover, the distributions of the values are presented as box-plots in Fig. 6.

Table 4 and Fig. 6 show that, while the ranges for the value of $DISP_{0.01}$ can be small for some instances of a function under these experimental conditions, they can have large relative variations, i.e., over 10% for all functions and close to 70% for some functions. It is not yet clear whether $DISP_{0.01}$ converges to the same value between instances as $n \to \infty$, i.e., the *RD* for all functions will converge to zero. We find that the values of $DISP_{0.01}$ cannot be generalized between instances under these experimental conditions. In summary and answering the questions for Stage I from Table 1, $DISP_q$ works best with a small value of q, providing the highest level of

Table 4 Minimum and maximum values of the $DISP_{0.01}$ for the first 15 instances of ten functions from the COCO benchmark, and their relative difference

	min	max	RD (%)
f_1	0.044	0.056	21.4
f_4	0.083	0.121	31.2
f_6	0.055	0.077	28.0
f_9	0.106	0.165	35.9
f_{10}	0.158	0.320	50.5
f_{13}	0.071	0.221	67.8
f_{15}	0.109	0.153	29.0
f_{19}	0.173	0.210	17.7
f_{23}	0.342	0.399	14.4
f_{24}	0.217	0.262	17.4

Under the same experimental conditions, the difference between the minimum and the maximum exceeds 10% of the maximum and can be close to 70%

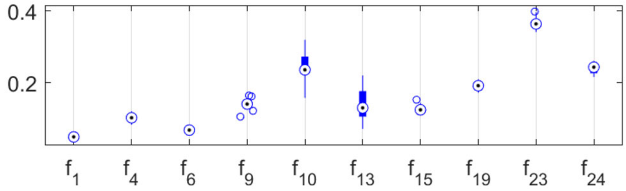


Fig. 6 Distribution of values of the $DISP_{0.01}$ for the first 15 instances of ten selected functions from the COCO benchmark. The distributions demonstrate that f_{10} and f_{13} have wider distributions, which corresponds to large *RD* values

discrimination between functions, while high values of q are vulnerable to quick convergence to the same value.

4.1.3 Fitness distributions

We focus our exploratory validation of the Fitness Distributions on determining whether two instances of the same function may have similar distributions. We obtain a sample of $D \times 10^3$ points, which is evaluated on the first instance from each one of the functions from the COCO benchmark at $D = 2$. Then, we estimated the Fitness Distribution using a kernel density estimator with a Gaussian kernel and bandwidth $h = 1.06\hat{\sigma}p^{-1/5}$, where $\hat{\sigma}$ is the sample standard deviation. The distributions are illustrated in Fig. 7, which shows that most of the problems have a left skewed distribution, except for f_5, f_{22} and f_{23}. This may explain why fast (significant) improvements are often made in early stages evolutionary optimization process; however, it becomes increasingly difficult to make minimal improvements afterwards. On the other hand, functions like f_{22} and f_{23}, which have a distribution close to uniform and no exploitable global structure, are difficult because it is equally probable to find both fit and unfit points. This does not apply for f_5 as it has an exploitable gradient. Perhaps, the COCO benchmark set lacks functions with a right skewed distribution, similar to the needle-in-a-haystack function or other deceptive functions, for which it would be difficult to make progress at the start of the search or convergence to poor local optima is frequent; hence, indicating whether the search can explore the space and find those uncommon regions.

To further explore the fitness distributions, we estimated the entropy, $H(\mathbf{Y})$, skewness, $\gamma(\mathbf{Y})$, and kurtosis, $\kappa(\mathbf{Y})$, for the first 15 instances of the same ten functions that we analyzed for the *FDC*. The minimum and maximum value and *RD* for each feature between the 15 instances are presented in Table 5 whereas the distributions of the values are presented as box-plots in Fig. 8. The values of $\gamma(\mathbf{Y})$ suggest that most of the functions have left skewed distributions excepting f_{23}, which is almost non-skewed, i.e., $\gamma(\mathbf{Y}) \approx 0$. The values of $\kappa(\mathbf{Y})$ suggest that most of the functions are *leptokurtic*, i.e., $\kappa(\mathbf{Y}) > 3$, which means that the fitness distribution has a "fat" tail that slowly converges to zero, indicating a higher propensity to generate outliers. Some strongly leptokurtic distributions, i.e., $\kappa(\mathbf{Y}) > 10$, are associated with high variable scaling functions, such as $\{f_6, f_9\}$. On the other hand, only f_{23} is *platykurtic*, i.e., $\kappa(\mathbf{Y}) < 3$, which means that the distribution has a "thin" tail with less propensity to generate outliers. Following Hansen et al. (2011a) classification of the functions, we observe different values of the features within the groups. For example, $\{f_{10}, f_{13}\}$ are described as unimodal functions with high conditioning. The range of both $\gamma(\mathbf{Y})$ and $\kappa(\mathbf{Y})$ for these two functions overlap; however, this is not the case for the range of $H(\mathbf{Y})$ indicating that the constraints present in f_{13} affect this feature's ability to detect ill-conditioning.

The difference in the value of $H(\mathbf{Y})$ between instances of a function does not exceed 10% excepting $\{f_1, f_4, f_6, f_{15}\}$. In some cases, i.e., $\{f_9, f_{19}, f_{23}, f_{24}\}$, it is close to 1%. For $\gamma(\mathbf{Y})$ and $\kappa(\mathbf{Y})$, the differences exceed 40% for most cases. As it is the case for $DISP_{0.01}$, the results suggest that these features may not be generalized between instances of a function. In summary and answering the questions for Stage I from Table 1, $H(\mathbf{Y})$ is a more reliable indicator of similarities between instances of the same function, while also having a clear relationship with ill-conditioning (Muñoz et al. 2015a). However, $\gamma(\mathbf{Y})$ and $\kappa(\mathbf{Y})$ can swing widely between instances, and lack a clear interpretation, reducing the interpretability of the results.

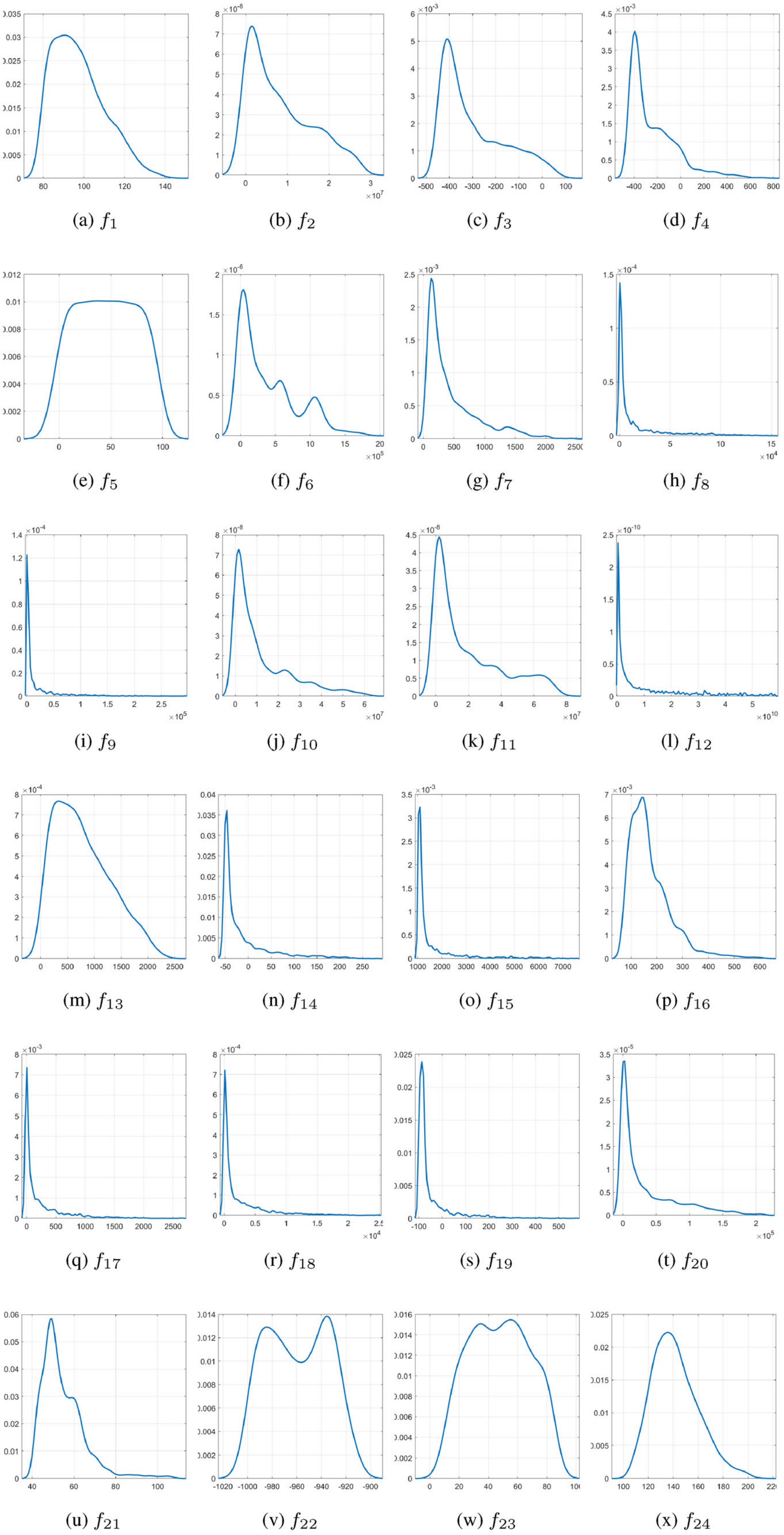
4.1.4 Model fitting

We focus our exploratory validation of the Model Fitting method on determining the sensitivity of the adjusted coefficient of determination, $\bar{R}^2$, to translational shifts of a function. For this purpose, we fitted linear and quadratic regression models to two instances of the Rastrigin function in one dimension (Eq. 9) bounded between $[-5, 5]$. We generate the shifts by setting the value of δ_i to $\{-5.0, 2.5\}$. Figure 9 illustrates the results.

$$y = 10 \cdot (2 - \cos(2\pi(x + \delta_i))) + (x + \delta_i)^2 \qquad (9)$$

Fig. 9a shows that the linear fit of the model is affected by the bounds of the function. We calculated $\bar{R}^2$ for both instances. The result for Instance 1 is 0.889 and for Instance 2 is 0.660, and their difference is ≈ 0.228. These results suggest that $\bar{R}^2$ of a linear fit converges to 1 when the optimum is closer to the bounds than to the center of the space, for some simple functions such as the Sphere. Therefore, we may have different values of $\bar{R}^2$ for different instances of the same function.

Fig. 7 Kernel density estimator of the $\mathcal{Y}$-distribution for the first instance of each of the 24 functions from the COCO benchmark at $D = 2$. For most of the functions, the distribution is left skewed



(a) f_1 (b) f_2 (c) f_3 (d) f_4

(e) f_5 (f) f_6 (g) f_7 (h) f_8

(i) f_9 (j) f_{10} (k) f_{11} (l) f_{12}

(m) f_{13} (n) f_{14} (o) f_{15} (p) f_{16}

(q) f_{17} (r) f_{18} (s) f_{19} (t) f_{20}

(u) f_{21} (v) f_{22} (w) f_{23} (x) f_{24}

Table 5 Minimum and maximum values of the entropy, $H(\mathbf{Y})$, skewness, $\gamma(\mathbf{Y})$, and kurtosis, $\kappa(\mathbf{Y})$ for the first 15 instances of ten functions from the COCO benchmark, and their relative difference

	$H(\mathbf{Y})$			$\gamma(\mathbf{Y})$			$\kappa(\mathbf{Y})$		
	min	max	*RD* (%)	min	max	*RD* (%)	min	max	*RD* (%)
f_1	3.661	4.641	21.1	0.451	0.804	43.9	2.475	3.109	20.4
f_4	5.778	7.106	18.7	0.841	1.999	57.9	2.601	7.531	65.5
f_6	7.097	13.151	46.0	0.703	5.478	87.2	2.130	37.591	94.3
f_9	10.045	10.151	1.0	1.904	3.547	46.3	6.132	18.567	67.0
f_{10}	16.659	17.757	6.2	0.906	1.878	51.8	2.779	6.281	55.8
f_{13}	6.986	7.679	9.0	0.144	1.047	86.3	1.975	3.713	46.8
f_{15}	5.809	7.562	23.2	0.920	4.175	78.0	2.795	26.022	89.3
f_{19}	4.701	4.767	1.4	1.835	3.512	47.7	5.915	18.161	67.4
f_{23}	4.277	4.298	0.5	− 0.033	0.109	130.0	1.912	1.992	4.0
f_{24}	4.228	4.278	1.2	0.536	0.665	19.4	2.938	3.377	13.0

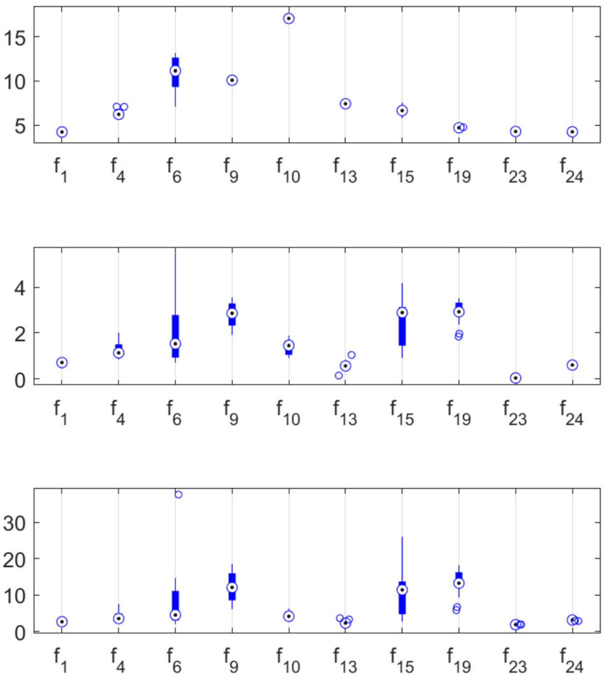


Fig. 8 Distribution of values of the (top) entropy, $H(\mathbf{Y})$, (middle) skewness, $\gamma(\mathbf{Y})$, and (bottom) kurtosis, $\kappa(\mathbf{Y})$ for the first 15 instances of ten functions from the COCO benchmark. The distribution demonstrates that $H(\mathbf{Y})$ have tighter ranges, whereas $\gamma(\mathbf{Y})$ and $\kappa(\mathbf{Y})$ can swing widely between instances, reducing the interpretability of the results

The results are different for the quadratic model, which are illustrated in Fig. 9. We calculated $\bar{R}^2$ for both instances. The result for Instance 1 is 0.947 and for Instance 2 is 0.841. The instance with the highest $\bar{R}^2$ is also the one with the optimum close to the bounds; however, the difference between both values is smaller compared to the result from the linear model. The quadratic model has more degrees of freedom than the linear model, i.e., the cardinality of the estimated regression coefficients, $|\boldsymbol{\beta}|$, is larger. Hence, the quadratic model may overfit to the data, which results in different functions having seemingly similar values of $\bar{R}^2$.

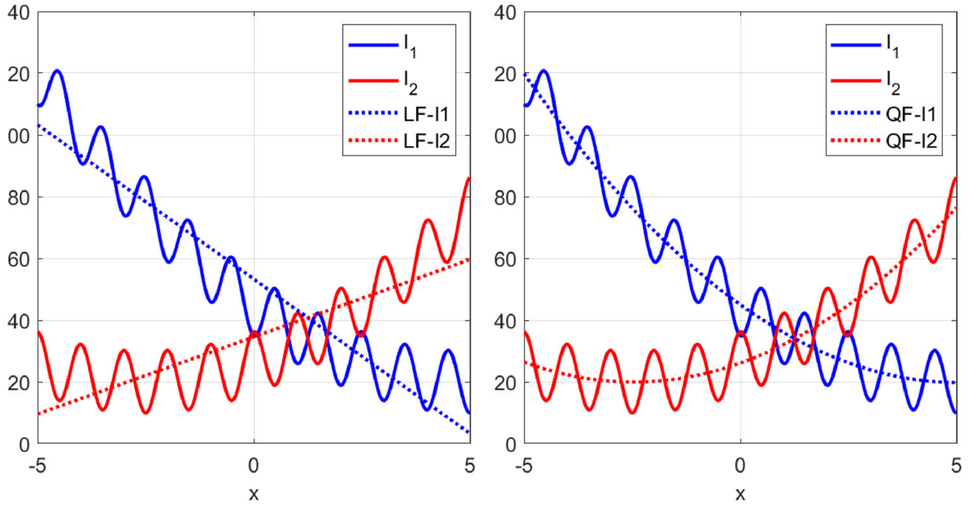
We calculated $\bar{R}^2$ using four models for the first 15 instances of the same ten functions that were used in the analysis for *FDC*. The four models considered are: linear model without interactions, *L*, linear model with interactions, *LI*, quadratic model without interactions, *Q*, and quadratic model with interactions *QI*. Table 6 shows the minimum and maximum value of $\bar{R}_2$ from the 15 instances and *RD* and Fig. 10 presents the distributions of the values as box-plots.

By comparing the results from the linear models in Table 6a with the quadratic models in Table 6b, we observe a gradual decrease on *RD* as the degrees of freedom increase. The only exception being f_{23}, which has similar differences for all models. However, because the minimum value of $\bar{R}^2$ is zero for this function, *RD* is not the best indicator of sensitivity of $\bar{R}^2$. In summary and answering the questions for Stage I from Table 1, $\bar{R}^2_{QI}$ is the most reliable indicator of similarities between instances, that includes both linear and quadratic effects, and it is the least vulnerable to wrong interpretations based on translations of the function.

4.1.5 Information significance

We focus our exploratory validation of the Information Significance on determining whether it is a suitable feature of variable dependency. For this purpose, we examine the effects of an orthogonal rotation in the input space has on the Information Significance. Rotations are a simple way to generate variable dependencies. The three functions under evaluation are the Sphere (Eq. 10), Ellipsoidal (Eq. 11) and

Fig. 9 Linear and quadratic models for the Rastrigin function in one dimension. Instance 1 (I1) is displaced in $\mathcal{X}$ to the right, with its optimum at $x = 5.0$, while Instance 2 (I2) is displaced in $\mathcal{X}$ to the left, with its optimum at $x = -2.5$. The figure illustrates how the linear fit may be different between instances, whereas the quadratic fit may be similar



(a) Linear models

(b) Quadratic models

Table 6 Values of $\bar{R}^2$ of four models for the first 15 instances of ten selected functions from the COCO benchmark at $D = 2$, using a sample of 2000 points

	$\bar{R}^2_L$			$\bar{R}^2_{LI}$		
	min	max	*RD* (%)	min	max	*RD* (%)
(a) Linear models						
f_1	0.012	0.887	98.6	0.012	0.887	98.6
f_4	0.153	0.852	82.0	0.153	0.852	82.0
f_6	0.227	0.906	74.9	0.397	0.911	56.4
f_9	0.091	0.220	58.6	0.248	0.574	56.8
f_{10}	0.000	0.821	100.0	0.187	0.930	79.9
f_{13}	0.062	0.922	93.3	0.127	0.949	86.6
f_{15}	0.091	0.866	89.5	0.514	0.952	46.0
f_{19}	0.091	0.228	60.1	0.235	0.553	57.5
f_{23}	0.000	0.002	100.0	0.000	0.002	100.0
f_{24}	0.356	0.392	9.2	0.367	0.402	8.7
(b) Quadratic models						
f_1	1.000	1.000	0.0	1.000	1.000	0.0
f_4	0.776	0.990	21.6	0.776	0.990	21.6
f_6	0.573	0.994	42.4	0.378	0.993	61.9
f_9	0.853	0.881	3.2	0.381	0.791	51.8
f_{10}	0.992	0.998	0.6	0.298	0.993	70.0
f_{13}	0.917	0.992	7.6	0.258	0.976	73.6
f_{15}	0.774	0.992	22.0	0.362	0.945	61.7
f_{19}	0.839	0.864	2.9	0.378	0.849	55.5
f_{23}	0.000	0.003	100.0	0.000	0.002	100.0
f_{24}	0.588	0.621	5.3	0.583	0.606	3.8

Rastrigin (Eq. 12) functions at $D = 2$, where $\mathbf{z} = \mathbf{Rx}$. The matrix $\mathbf{R}$ controls the rotation and it is calculated using Eq. (13), where ω is a rotation angle in radians within the $[0, \pi]$ range. The scale vector for the Ellipsoidal function, $\mathbf{C}$, is equal to $[1\ 1000]$. The results are illustrated in Fig. 11.

$$y = \sum_{i=1}^{2} z_i^2 \tag{10}$$

$$y = \sum_{i=1}^{2} C_i z_i^2 \tag{11}$$

$$y = 10 \cdot \left(2 - \sum_{i=1}^{2} \cos(2\pi z_i)\right) + \sum_{i=1}^{2} z_i^2 \tag{12}$$

$$\mathbf{R} = \begin{bmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{bmatrix} \tag{13}$$

Fig. 11a is a top view of the Sphere function surface and Fig. 11b is a line plot of $\xi(x_1)$, $\xi(x_2)$, $\xi^{(1)}$, $\xi^{(2)}$ and $\sigma_{\xi}^{(1)}$, which is the standard deviation of the information significance of order 1. The Sphere function is rotational invariant, which means that ideally $\xi(x_1) = \xi(x_2) = \xi^{(1)}$, $\xi^{(2)} > \xi^{(1)}$ and $\sigma_{\xi}^{(1)} = 0$ for any value of ω within the range. The results in Fig. 11b are consistent with the ideal results. There is a small difference between the values of $\xi(x_1)$ and $\xi(x_2)$, which may be attributed to numerical rounding errors.

Fig. 11c is a top view of the Ellipsoidal function surface and Fig. 11d is a line plot of the features. Unlike the Sphere function, the Ellipsoidal function is not rotational invariant. Therefore, the values of $\xi(x_1)$ and $\xi(x_2)$ are equal at $\frac{\pi}{4}$ radians and have a phase difference of $\frac{\pi}{2}$ radians. At $\left\{0, \frac{\pi}{2}, \pi\right\}$ radians, $\xi^{(1)}$ is maximum, which is less than the maximum values of $\xi(x_1)$ or $\xi(x_2)$. In addition, $\xi^{(2)} \geq \xi^{(1)}$ when ω is between $[0.127, 1.428]$ and $[1.777, 3.015]$

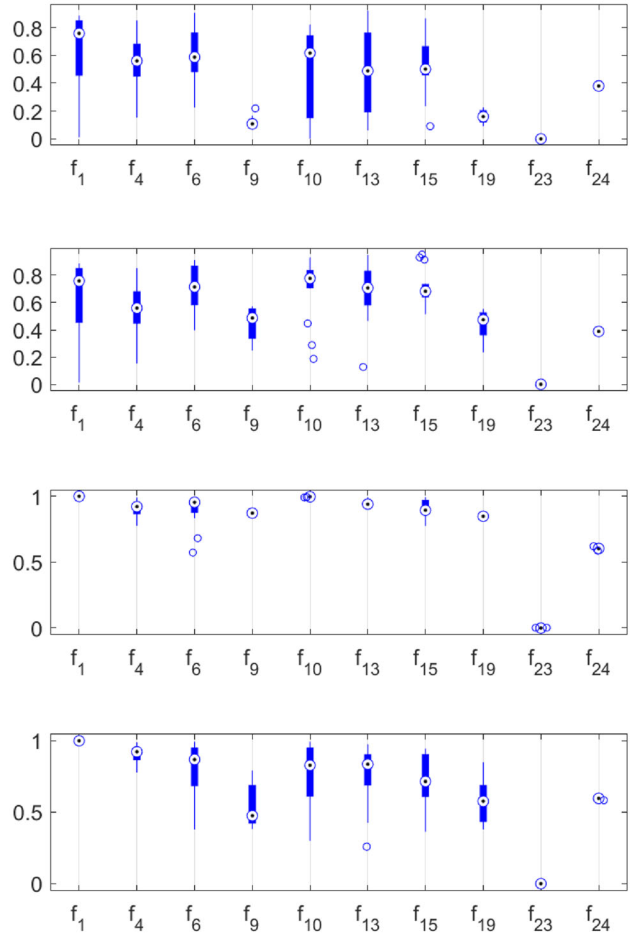


Fig. 10 Distribution of values of $\bar{R}^2$ of the (top) linear model without interactions, *L*, (second) linear with interactions, *LI*, (third) quadratic without interactions, *Q*, (bottom) and quadratic with interactions *QI* models for the first 15 instances of ten selected functions from the COCO benchmark. The distribution demonstrates that the most reliable feature is $\bar{R}^2_{QI}$, as it has the least variability, includes both linear and quadratic effects, and it is the least vulnerable to wrong interpretations based on translations of the function

radians. Perhaps the variables can be separated when ω is not within these ranges.

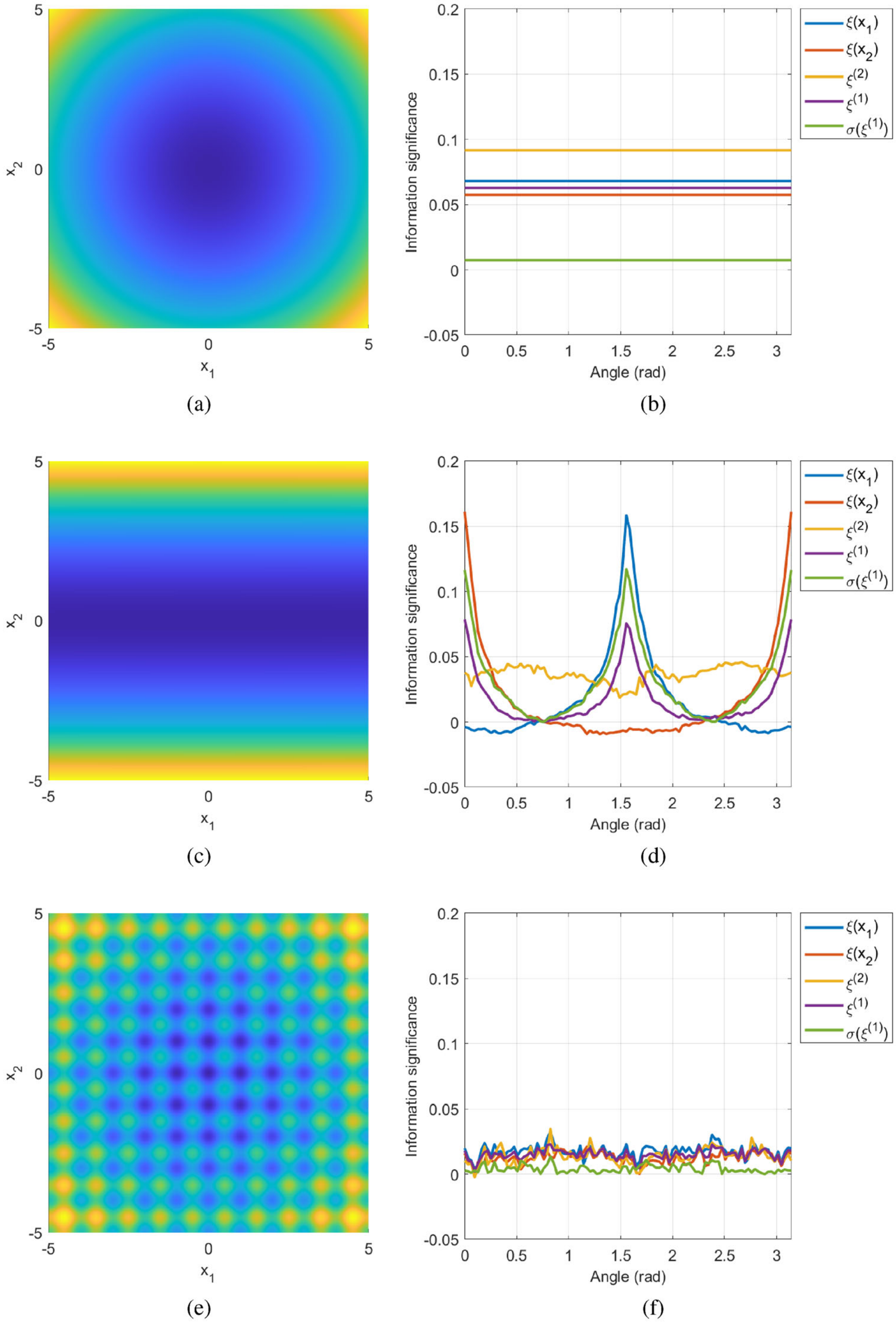
The Information Significance appears to generate correct measurements of the variable dependencies on the Ellipsoidal function. However, this was not the case for the Rastrigin function. Figure 11e is a top view of this function surface and Fig. 11f is a line plot of the features. An appropriate pattern is not evident in Fig. 11f, implying that the Information Significance has lost its ability to identify variable dependencies due to the high modality of the function.

To further explore this observation, we calculated the values of $\xi^{(1)}$ and $\xi^{(2)}$ for the first 15 instances of the same 10 functions from the COCO benchmark at $D = 2$ that we analyzed for the *FDC*. To obtain a sample of $n = 2000$ points, we followed the procedure described in Sect. 3. Table 7 shows the minimum and maximum value of the features from the 15 instances and their relative difference, *RD* and Fig. 12 presents the distributions of the values as box-plots.

For the Sphere function, f_1, the three features in Table 7 have *RD* above 30%. This seems counter-intuitive because f_1 is rotational and translation invariant; hence, we expected that $RD \approx 0.0\%$ for all features. However, this can be explained by the differences on the experiment above and the COCO benchmark. For the former, the optimum value is fixed at $(0, 0)$, whereas for the latter the optimum is randomly displaced within a bounded space, where sections of the function would be present in some instances but absent in others. In other words, for these features, a translational shift over two or more variables creates dependencies between them. For the Rotated Ellipsoidal function, f_{10}, whose instances are created by rotation, have *RD* above 90% for $\xi(x_1)$ as expected. Highly multimodal functions, such as the Katsuura, f_{23}, and Lunacek bi-Rastrigin, f_{24}, functions, have feature whose values are approximately zero. As the values for f_{23} are above 90%; *RD* is not the best indicator of the sensitivity of these features to translational shifts and orthogonal rotations.

We use the mean and the standard deviation to summarize the results for the Information Significances of first and second orders. Figure 13 shows two scatter plots where each mark represents the value of $\xi(V)$ for $|V| = \{1, 2\}$, while the boxes represent the range of $\xi^{(k)} \pm \sigma_{\xi}^{(k)}$, and the center line is equal to $\xi^{(k)}$. We analyze the first five instances of the Rotated Ellipsoidal function, f_{10}, from the COCO benchmark at $D = 10$. We use a sample of 10^4 points.

Fig. 13 shows that for Instances $\{1, 2\}$, there are three variables with values of $\xi(V)$, $|V| = 1$ above average, whereas the remaining variables have values of $\xi(V)$ below average. This implies that these three variables may have higher independence. On the other hand, Instance 5 has variables with values of $\xi(V)$ close to average, which implies that all the variables may be dependent from each other. We observe similar patterns in both figures. Combinations of the independent variables in Instances $\{1, 2\}$ have values of $\xi(V), |V| = 2$ above average. Instance 5 has variable combinations with values of $\xi(V)$ close to the average. The analysis of $\xi^{(k)}$ provide similar conclusions. High values of $\xi^{(k)}$ imply that most of the variables are independent, such as those for Instances $\{3, 4\}$. In summary and answering the questions for Stage I from Table 1, Information Significance features not only capture variable dependencies produced by rotations, but also by translational shifts within a bounded space and other non-linearities. Therefore, as it confounds multiple effects, their



(a) (b)

(c) (d)

(e) (f)

Fig. 11 Effect that a rotation in the input space has on the Information Significance features. **a**, **c**, **e** are top views of the surfaces for the Sphere, Ellipsoidal and Rastrigin functions respectively, whereas **b**, **e**, **f** show the value of features against the rotation angle. The features capture variable dependencies only for smooth functions, whose optimal value has not been displaced, such as the Sphere and Ellipsoidal functions

interpretation as variable independence measures becomes limited.

4.2 Statistical validation stage

The results from Sect. 4.1 suggest that translational shifts and orthogonal rotations significantly impact on the value of most features; hence, the conclusions drawn from several instances of the same function may be contradictory. However, it is not yet clear if the difference between features is statistically significant or not. In addition, these results do not show if there are relationships between the different landscape features. In this section, we present the results of the statistical stage.

Table 7 Values of $\xi^{(1)}$ and $\xi^{(2)}$ for the first 15 instances of ten selected functions from the COCO benchmark at $D = 2$, using a sample of 2000 points

	$\xi^{(1)}$			$\xi^{(2)}$		
	min	max	*RD* (%)	min	max	*RD* (%)
f_1	0.076	0.115	34.4	0.131	0.203	35.5
f_4	0.063	0.098	36.2	0.075	0.123	38.7
f_6	0.011	0.076	86.0	0.048	0.100	52.5
f_9	0.018	0.041	56.8	0.037	0.067	44.5
f_{10}	0.002	0.042	95.1	0.027	0.046	41.2
f_{13}	0.010	0.085	87.7	0.065	0.136	52.0
f_{15}	0.019	0.081	76.5	0.074	0.128	42.4
f_{19}	0.018	0.065	72.3	0.048	0.081	41.3
f_{23}	− 0.006	− 0.003	142.3	− 0.012	− 0.005	136.0
f_{24}	0.022	0.027	18.7	0.043	0.053	18.9

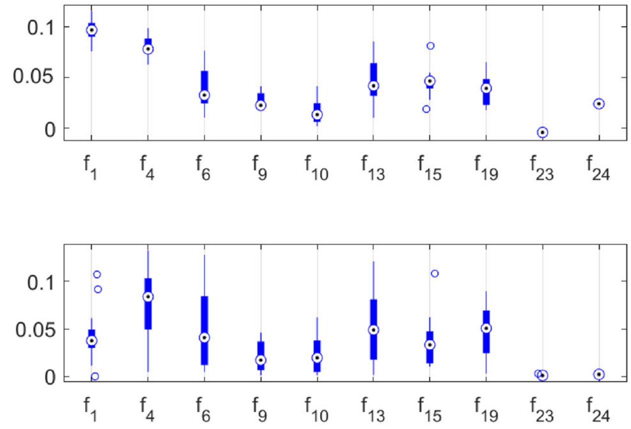


Fig. 12 Distribution of values of (top) $\xi^{(1)}$ and (bottom) $\xi^{(2)}$ for the first 15 instances of ten selected functions from the COCO benchmark. The distributions show that these features capture variable dependencies produced by translational shifts within a bounded space, confounding multiple effects, their interpretation as variable independence measures becomes limited

4.2.1 Magnitude of the bootstrap variance

We estimated the variance of the features using bootstrapping and Eq. (8). A high variance indicates that a feature may change for a particular function due to the sample points collected. As such, the magnitude of the variance should be reported. Figure 14 shows the average variance for the $\max(|\beta_L|)$ feature over the the first 15 instances for the Sphere, f_1, and Bent Cigar, f_{15}, functions from the COCO benchmark set. Each line on the graph represents a dimension of the function. On the horizontal axis is the sample size. In both figures, the variance decreases as the sample size increases, albeit not monotonically. However, if we focus on the vertical axis, we notice the difference on magnitude of the variances. The figure shows that for f_1, the average variance for $\max(|\beta_L|)$ is in the $[10^{-3}, 10^{-1}]$ range, whereas for f_{12}, the average variance is in the $[10^{15}, 10^{18}]$ range. This is a large difference in orders of magnitude. Considering that the variance quantifies the volatility of the feature, this figure implies that for f_{12} the value of $\max(|\beta_L|)$ might change several orders of magnitude depending on the instance.

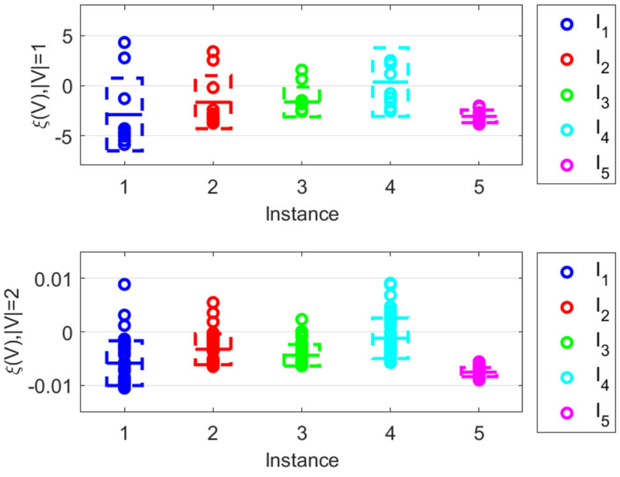


Fig. 13 Value of the information significance of first and second orders for the first five instances of the Rotated Ellipsoidal function, f_{10}, from the COCO benchmark at $D = 10$. Each mark represents the value of $\xi(V)$ for $|V| = \{1, 2\}$, while the boxes represent the range of $\xi^{(k)} \pm \sigma_{\xi}^{(k)}$, and the center line is equal to $\xi^{(k)}$. Variables or combinations above the average indicate higher independence, while values clustered together indicate dependencies

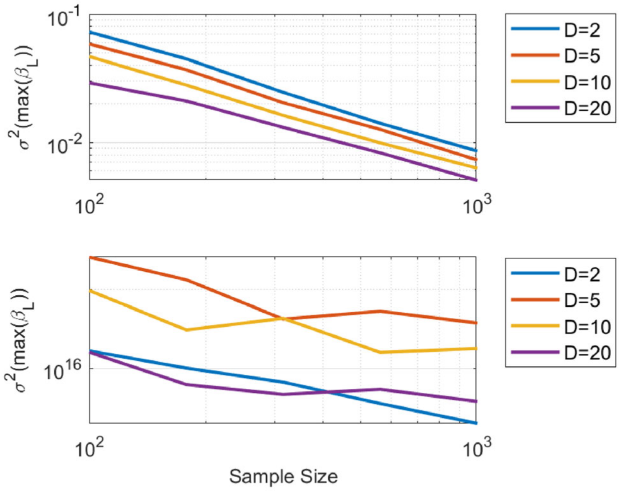


Fig. 14 Average variance for $\max(|\beta_L|)$ feature for the Sphere (top) and Bent Cigar (bottom) functions against the sample size. Both the horizontal and vertical axis are on a logarithmic scale. The figure shows how the variance of a feature might be large or small depending on function

Because $\max(|\beta_L|)$ is a variable scaling feature, and f_{12} has a condition number, which is the ratio between the highest and lowest scaled variables, close to 10^6, it explains the range of the feature.

We calculate the Quartile Coefficient of Dispersion, *QCD*, to summarize the variance of each feature at each sample size. The *QCD* is a normalized measure of variation, defined as the ratio between the median and the interquartile range of the statistic. Figure 15 illustrates the results. The horizontal axis represents the *QCD* in a base-10 logarithmic scale. Large values of *QCD* indicate that the feature often has large variances. Therefore, for the same function but two different samples of the same size, the feature values may be different by several orders of magnitude. We call these features *volatile*, and unreliable under the definitions described in Table 1. Both $\min(|\beta_L|)$ and $\max(|\beta_L|)$ are volatile, matching the results from Fig. 14.

4.2.2 Statistical significance of the difference between features

Table 8 shows the results of the tests of statistical significance between functions, instances and sample sizes for the landscape features. The number between parenthesis indicates the size of the family of tests for which the False Discovery Rate (*FDR*) was adjusted using the method by Benjamini and Yekutieli (2001). The table shows that all the features have values that are statistically different between functions. These results imply that it is unlikely that for two different functions the features will converge to the same value. Hence, these features discriminate between functions.

Between function instances for the landscape features, Table 8 shows that the number of tests for which H_0 was rejected is close to 100% for almost all the features. Therefore, the features have values that are statistically different between instances, implying that the features for two different instances may not converge to the same value. Hence, the features also discriminate between instances of the same function. Given this result, we examine which functions have at least one pair of instances, at any sample size, for which less than 50% of the adjusted tests was rejected, i.e., which functions had instances with half of its features being statistically similar. We encounter that for all dimensions, functions $\{f_1, \ldots, f_{15}\}$ have at least two similar instances. With the exception of $\{f_3, f_4\}$, these functions are unimodal, with various degrees of conditioning, allowing features such as R^2_{QI} to converge. This also highlights the difficulty of finding similar instances as modality increases, which can be explained by the larger space where the global optimum can be located, minimizing the chance that the features converge.

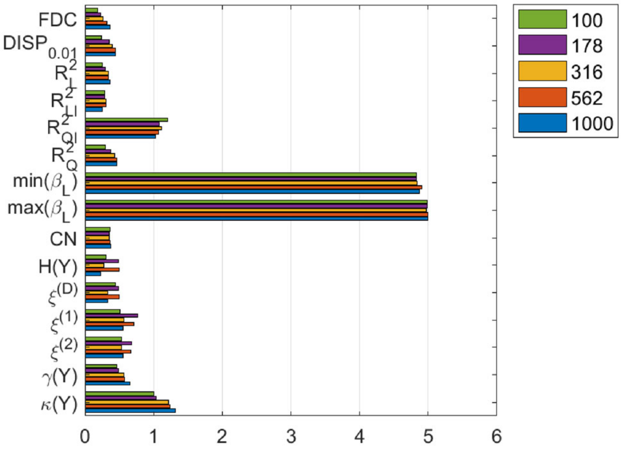


Fig. 15 Quartile coefficient of dispersion (*QCD*) of the bootstrapped variance. The horizontal axis is in a base-10 logarithmic scale. Both $\min(|\beta_L|)$ and $\max(|\beta_L|)$ have very large values of *QCD*, which indicates that these two feature are volatile

Between sample sizes for the landscape features, Table 8 shows that, against our expectation and similarly to the instance results, the number of tests for with H_0 was rejected is close to 90% for almost all the features. This result implies that the features do not converge during the early steps of the optimization process. As was the case with the instance tests, we examine which functions have at least one pair of sample sizes, for which less than 50% of the adjusted tests were rejected. We find that f_5 at each dimension have at least an instance for which there is no significant difference between sample sizes; for f_2 an instance was found with no significant differences for $D = 2$ between $\{316, 562, 1000\} \times D$, and for $D = 5$ between $\{562, 1000\} \times D$; and for f_{12} at $D = 2$ between $\{562, 1000\} \times D$. These results match those by Saleem et al. (2019), confirming that sample size effects need to be considered when features are estimated, as they not only depend on the problem landscape and feature definition, but also the sample size.

In summary, and answering the questions for Stage II from Table 1, the features under study discriminate between functions and instances by providing significantly different results between them. Moreover, the results do not converge within the sample sizes under study, as they produced significantly different results between them. This implies that features obtained using different sample sizes are incompatible, even if they are taken from different functions or instances.

4.2.3 Correlation between features

Table 9 shows the Pearson correlation, $\rho_{x,y}$, between the landscape features and the dimension of the problem, *D*.

Table 8 Percentage of rejected tests at the 5% confidence level between function, instances and sample sizes for the landscape features

D	Function (20700)				Instances (12600)				Sample Size (3600)			
	2 (%)	5 (%)	10 (%)	20 (%)	2 (%)	5 (%)	10 (%)	20 (%)	2 (%)	5 (%)	10 (%)	20 (%)
FDC	98.5	98.8	98.9	98.9	95.9	96.3	95.6	95.2	93.8	94.9	95.1	94.6
$DISP_{0.01}$	96.0	97.7	98.2	98.8	89.7	93.0	93.9	94.2	95.8	96.9	97.7	98.2
R^2_L	99.6	99.7	99.7	99.8	93.2	93.3	93.5	93.7	82.2	88.9	90.6	91.4
R^2_{LI}	99.6	99.8	99.8	99.8	93.3	93.7	93.5	93.4	84.8	90.7	92.1	94.6
R^2_{QI}	99.5	99.4	99.6	99.5	88.9	89.6	89.6	89.5	82.7	86.5	88.3	89.6
R^2_Q	99.4	99.4	99.4	99.5	89.5	89.6	89.3	89.6	81.3	84.7	86.5	87.2
$\min(\beta_L)$	99.5	99.5	99.4	99.3	93.6	89.3	85.6	81.7	90.8	89.7	88.0	86.0
$\max(\beta_L)$	99.8	99.9	99.9	100.0	93.7	93.7	93.5	93.5	80.6	87.0	90.8	90.8
CN	97.9	97.8	97.8	98.0	89.5	85.7	85.2	82.4	86.8	85.3	84.6	83.6
$H(\mathbf{Y})$	99.8	99.9	100.0	100.0	94.8	96.7	97.7	98.1	95.8	96.4	96.1	97.3
$\xi^{(D)}$	98.4	98.8	99.5	99.6	93.8	95.9	97.0	97.0	99.0	98.3	96.2	96.5
$\xi^{(1)}$	98.7	98.2	98.6	99.0	93.2	93.6	95.0	94.5	98.4	98.3	96.6	97.0
$\xi^{(2)}$	98.4	98.8	99.2	99.4	93.8	95.4	96.0	96.4	99.0	98.2	97.1	96.3
$\gamma(\mathbf{Y})$	99.4	99.7	99.8	99.7	96.9	97.4	97.0	97.3	85.5	91.7	93.5	95.5
$\kappa(\mathbf{Y})$	99.4	99.4	99.4	99.5	95.9	97.3	97.4	96.8	86.9	91.5	94.0	95.1

Ideally, comparisons between functions should have a high number of rejections, whereas between instances and sample sizes should have low number of rejections

Table 9 Pearson correlation ($\rho_{x,y}$) between the dimension of the problem, D, and the landscape features. In boldface are those features with high correlation, i.e., $\rho_{x,y} \geq 0.7$

	$\rho_{x,y}$		$\rho_{x,y}$		$\rho_{x,y}$
FDC	− 0.181	$\vec{R}^2_Q$	− 0.067	$\xi^{(D)}$	− 0.346
$DISP_{0.01}$	**0.714**	$\min(\beta_L)$	− 0.033	$\xi^{(1)}$	− 0.457
$\vec{R}^2_L$	− 0.023	$\max(\beta_L)$	0.022	$\xi^{(2)}$	− 0.449
$\vec{R}^2_{LI}$	0.033	CN	− 0.120	$\gamma(\mathbf{Y})$	0.028
$\vec{R}^2_{QI}$	− 0.014	$H(\mathbf{Y})$	0.059	$\kappa(\mathbf{Y})$	0.084

Only $DISP_{0.01}$ has a high correlation with D, implying they could be equivalent, i.e., they provide similar information

We boldfaced the value of the correlation of $DISP_{0.01}$ because it is the only feature with high positive correlation, i.e., $\rho_{x,y} \geq 0.7$ (Hinkle et al. 2003). In Sect. 4.1, we explained that the average distance between points converges to $1/\sqrt{6}$ as n increases (Morgan and Gallagher 2014). This also explains why $DISP_{0.01}$ is correlated with D implying that they could be equivalent.

Table 10 shows the value of $\rho_{x,y}$ between the landscape features. We observe that FDC has high correlation with $\vec{R}^2_Q$, the quadratic model without interactions. These two features are related because FDC is based on the Euclidean distance of all points to the approximated global optimum, $\hat{\mathbf{x}}_o$, which can be thought of as the square root of a quadratic relationship of the points. In addition, the $\vec{R}^2$ for all models have high correlations. There are two possible reasons for this result. First, that the function is simple enough that a linear or a quadratic model have a good fit. For example, the Linear Slope function, f_5, has a $\vec{R}^2 = 1$ regardless of the model, whereas the Sphere function, f_1, has values of $\vec{R}^2_{QI} = \vec{R}^2_Q = 1$. Second, that the function is so complex that a linear or a quadratic model are not good enough; hence, the fit of the model does not improve by increasing the degrees of freedom.

The mean Information Significances of first, $\xi^{(1)}$, and D-th, $\xi^{(D)}$, orders are moderately correlated between each other, and highly correlated with the mean information significance of second order, $\xi^{(2)}$. In addition, $\xi^{(D)} = \xi^{(2)}$ for any two-dimensional function, which means that a quarter of the data analyzed of each feature are equal. This implies that perhaps $\xi^{(1)}$ captures most of the information that $\xi^{(2)}$ and $\xi^{(D)}$ may contain, and it may be sufficient to only calculate $\xi^{(1)}$.

The skewness, $\gamma(\mathbf{Y})$, and kurtosis, $\kappa(\mathbf{Y})$, of the fitness distribution are highly correlated, implying that the higher the skewness the "fatter" the tail of the distribution. Using the evidence on Fig. 7 and Table 5, we conclude that those functions with high $\gamma(\mathbf{Y})$ and $\kappa(\mathbf{Y})$ are likely to have high

Table 10 Pearson correlation ($\rho_{x,y}$) between the landscape features combinations

	$DISP_{0.01}$	$\bar{R}^2_L$	$\bar{R}^2_{LI}$	$\bar{R}^2_{QI}$	$\bar{R}^2_Q$	$\min(\beta_L)$	$\max(\beta_L)$
FDC	− 0.469	0.694	0.586	0.698	**0.782**	− 0.011	− 0.050
$DISP_{0.01}$		− 0.236	− 0.169	− 0.308	− 0.364	− 0.046	− 0.001
$\bar{R}^2_L$			**0.859**	**0.730**	**0.841**	− 0.032	− 0.059
$\bar{R}^2_{LI}$				**0.839**	0.680	− 0.009	− 0.028
$\bar{R}^2_{QI}$					**0.870**	− 0.030	− 0.066
$\bar{R}^2_Q$						− 0.052	− 0.094
$\min(\beta_L)$							0.584
$\max(\beta_L)$							

	CN	$H(\mathbf{Y})$	$\xi^{(D)}$	$\xi^{(1)}$	$\xi^{(2)}$	$\gamma(\mathbf{Y})$	$\kappa(\mathbf{Y})$
FDC	0.253	0.076	0.216	0.377	0.358	− 0.130	− 0.110
$DISP_{0.01}$	− 0.242	0.055	− 0.320	− 0.552	− 0.508	− 0.078	0.041
$\bar{R}^2_L$	− 0.137	0.227	0.229	0.367	0.344	− 0.105	− 0.102
$\bar{R}^2_{LI}$	− 0.162	0.361	0.202	0.260	0.287	0.025	− 0.073
$\bar{R}^2_{QI}$	− 0.013	0.390	0.265	0.341	0.369	− 0.020	− 0.124
$\bar{R}^2_Q$	0.014	0.252	0.284	0.432	0.412	− 0.142	− 0.149
$\min(\beta_L)$	− 0.019	0.175	0.010	− 0.003	0.004	0.148	0.062
$\max(\beta_L)$	− 0.062	0.241	0.002	− 0.014	− 0.007	0.309	0.186
CN		− 0.312	0.022	0.041	0.076	− 0.131	− 0.059
$H(\mathbf{Y})$			0.240	0.090	0.184	0.419	0.197
$\xi^{(D)}$				0.624	**0.877**	0.179	− 0.048
$\xi^{(1)}$					**0.872**	0.046	− 0.056
$\xi^{(2)}$						0.121	− 0.056
$\gamma(\mathbf{Y})$							**0.760**

In boldface are those features with $\rho_{x,y} \geq 0.7$

variable scaling. In summary and answering the questions for Stage II from Table 1, the studied feature set can be summarized into four reliable features, $\left\{\bar{R}^2_{QI}, CN, \xi^{(1)}, H(\mathbf{Y})\right\}$, that are weakly correlated, have low volatility and clear interpretations.

5 Discussion

We have identified the strengths and weaknesses of five feature sets using a well-structured, experimental methodology focused on both exploratory and statistical validation stages which was summarized in seven key questions in Table 1. The answers to these questions are condensed in Table 11. Importantly, we have demonstrated that the results of an ELA feature for one instance cannot be generalized over all the instances of a function for all the features under analysis, even if the feature is theoretically invariant to translational shifts and orthogonal rotations, such as *FDC*. This effect is due to the bounds on the input space. When an instance is generated by translating or rotating a function, there are sections of the function present in some instances and absent in others. This *boundary effect* is evident in the features extracted through model fitting, where two instances of the same function produced values at the opposite bounds of the feature range. Therefore, it is important to consider the impact of this boundary effect on the theoretical analysis of landscape features, as it implies that one function instance is insufficient to describe a function with certainty when bounds are considered. This in itself is not a practical limitation of a given feature, as demonstrated in previous work (Belkhir et al. 2016a; Muñoz and Smith-Miles 2017; Kerschke and Trautmann 2019a), some features are good predictors of performance for some algorithms and not for others. Therefore, invariant features are useful for invariant algorithms, such as CMA-ES, while variant features are useful for variant algorithms, such as Classic PSO (Hansen et al. 2011b).

The results show the importance of stating the experimental conditions for the evaluation of the feature. Due to the fact that landscape features are approximate values, the

Table 11 A summary of the results from applying the experimental methodology proposed in Sect. 3

	Useful?	Vulnerable?	Low variance?	Different between			Uncorrelated?
				Functions?	Instances?	Sample Size?	
FDC	✔	Medium ruggedness	✔	✔			
$DISP_{0.01}$	✔	Convergence to same value	✔	✔			
R^2_L	✔	Translations	✔	✔			
R^2_{LI}	✔	Translations	✔	✔			
R^2_{QI}	✔		✔	✔			
R^2_Q	✔	Translations	✔	✔			✔
$\min(\beta_L)$	✔			✔			✔
$\max(\beta_L)$	✔			✔			✔
CN	✔		✔	✔			✔
$H(\mathbf{Y})$	✔		✔	✔			
$\xi^{(D)}$	✔	Translations	✔	✔			
$\xi^{(1)}$	✔	Translations	✔	✔			
$\xi^{(2)}$	✔	Translations	✔	✔			
$\gamma(\mathbf{Y})$	✔	Any transformation	✔	✔			✔
$\kappa(\mathbf{Y})$	✔	Any transformation	✔	✔			

Check marks indicate positive answers to each question

results generated from the features should be reported along with a measure of their volatility, such as the variance. We observed in Fig. 15 that some features, such as $\min(|\beta_L|)$ and $\max(|\beta_L|)$, are volatile on average, where others, such as $\bar{R}^2_{QI}$, are nonvolatile. The volatility of the feature helps to anticipate whether the changes in the feature may affect further analysis stages. We can expect that a volatile feature might affect the results of a subsequent analysis stage, if the feature is important in that stage's context.

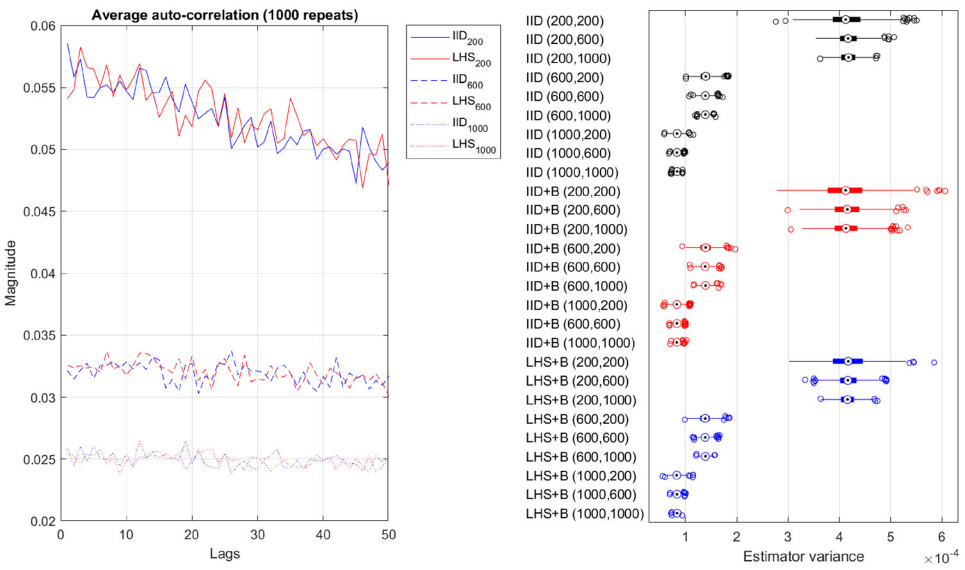
A larger sample size, n, leads to a more accurate feature, but also leads to a more accurate estimation of the variance. Ideally, the variance should converge to zero when $n \rightarrow \infty$, otherwise it is dependent on f and n, as mentioned in Sect. 3. During the experimental analysis, we found that the variance of a feature, such as *FDC*, decreases when n increases for some functions and increases for other functions. Analyzing the change of the variance due to the change in the value of n may provide additional evidence about the complexity of a function.

Modality has a similar effect on the sample size as the dimension of the problem. For example, consider the one-dimensional function $y = \sin(2\pi\varphi x)$, where φ controls the number of local optima in the landscape. Following the Nyquist-Shannon sampling theorem, we would require at least 2φ points to know the exact number of local optima. Therefore, if φ increases, the number of points required also increases. This *curse of modality* is present even in low dimensional functions. It affects estimations of the local optima and any feature based on the distance between an estimated optimum and any other points. For example, the value of $DISP_{0.01}$ for the Katsuura function, f_{23}, which converges towards $1/\sqrt{6}$, as illustrated in Fig. 5.

The combination of an exploratory and statistical analysis described in this paper is subject to two practical limitations worthy of further discussion. First, it can be argued that the sample size is relatively limited as the dimension of the function increases. To obtain a more precise feature value for complex functions, it may be adequate to evaluate larger sample sizes, for example, $D \times 10^4$ points. However, memory and computational time requirements pose practical constraints. For example, the calculation of a distance matrix, for a 20 dimensional sample of 20×10^4 points at double precision, requires a memory space of approximately 320GB, which often exceeds the memory available in a current workstation. Moreover, our choice of sample generator may also induce bias on the results, as sampling strategies have a demonstrated effect of convergence of the features depending on their space-covering characteristics (Renau et al. 2020; Crombecq et al. 2011).

Second, it can be argued that a more systematic approach should be employed to test the effect of instantiation on the features. Since the search space expands with dimension, the 15 random instances generated with the COCO software are more sparse in the space as the

Fig. 16 Validation of the assumptions behind our experimental methodology. **a** Average magnitude of the auto-correlation with lags in the $[1,\ 50]$ range, for data drawn from the $[0,\ 1]$ interval. **b** Distribution of the variance of the mean from N samples of n points, $IID(n,N)$, bootstrapping N times a sample of n points, $IID+B(n,N)$, and bootstrapping N times a LHS of n points, $LHS+B(n,N)$. The results confirm that there is no practical difference between taking N uniformly distributed random samples and bootstrapping N times a single LHS



(a) (b)

dimension increases. This is another expression of the curse of dimensionality. Ideally, there should be at least a linear increment on the total instances with the dimension; however, this is also subject to memory constraints.

6 Conclusions

In this paper, we have proposed a robust experimental methodology that considered the landscape features as random variables, allowing us to identify comparative advantages and disadvantages of each feature. Moreover, we evaluated the sensitivity of the features to the sample size and to shifts and rotations of the function.

We have drawn three conclusions from these experiments: First, all landscape features should be reported along with a measure of their volatility. Second, the value of a feature from one instance cannot be generalized over all the instances of a function, even if the feature is theoretically invariant to translational shifts and orthogonal rotations. Third, modality has a similar effect on the sample size as the dimension of the problem, e.g., as the number of local optima increases the size of the sample should also increase. By only focusing on the value of the feature, without considering its distribution, we may not have obtained these valuable insights.

Given that the features studied in this paper are sensitive to function transformations, our current research focuses on developing methods that help us better explain these changes, on both the features and the location of functions in an *instance space* (Muñoz and Smith-Miles 2015, 2017). Moreover, some statistical tests used in our experiments can be used to determine the reliability of the features in other fields where feature-based approaches to algorithm selection are becoming popular, such as forecasting (Kang et al. 2017).

Appendix: Validation of the assumptions behind the experimental methodology

Our experimental methodology makes assumptions that can be summarized in two questions: (a) Since Latin Hyper-cube Samplin (LHS) is a type of stratified sampling, is the independence assumption still valid? (b) What are the differences between multiple uniformly distributed random samples, bootstrapping a single uniformly distributed random sample, and bootstrapping a single LHS, when calculating the variance of an estimate? To answer these questions, we have carried out two simple experiments that demonstrate that there is no practical difference on the results between taking multiple uniformly distributed random samples and bootstrapping a LHS. On the first experiment, we address the independence assumption, by calculating the magnitude of the auto-correlation with lags in the $[1,\ 50]$ range, for data drawn from the $[0,\ 1]$ interval. For this assumption to hold for LHS, the magnitudes of the auto-correlation should follow the same trend that for a uniformly distributed random sample and be close to zero, indicating that it is not possible to estimate the value of one point from another. We repeat this experiment 1000 times and average the results, which are presented in Fig. 16a for samples with $\{200, 600, 1000\}$ points. Other than the descending trend for a sample of 200 points, which can be explained by the decrease in points in the sample for which the auto-correlation can be calculated, the results demonstrate that the independence assumption holds for a LHS in practice.

On the second experiment, we address the second question by estimating the variance of the mean from these three different sampling regimes, using data drawn from the $[0,\ 1]$ interval. On the first one, called $IID(n, N)$, we took N uniformly distributed random samples of n points. On the second one, called $IID + B(n, N)$, we took one uniformly distributed sample of n points and bootstrapped it N times. On the third one, called $LHS + B(n, N)$, we took one LHS of n points and bootstrapped it N times. Each sampling regime produced N mean estimates, from which the variance is calculated. The experiments are repeated 1000 times for all the combinations of $\{n, N\} = \{200, 600, 1000\}$. The results are shown in Fig. 16b as box-plots, which demonstrate that there is no practical difference between taking N uniformly distributed random samples and bootstrapping N times a single LHS.

Acknowledgements We express our gratitude to the two reviewers and the guest editors for their thorough and valuable suggestions, which significantly improved this paper. We also acknowledge Saman K. Halgamuge for his feedback on earlier versions on this work.

Funding Funding was provided by the Australian Research Council through the Australian Laureate Fellowship FL140100012, and The University of Melbourne through MIRS/MIFRS scholarships.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Alissa M, Sim K, Hart E (2019) Algorithm selection using deep learning without feature extraction. In: GECCO'19. ACM Press. https://doi.org/10.1145/3321707.3321845

Beck J, Freuder E (2004) Simple rules for low-knowledge algorithm selection. In: CPAIOR '04, LNCS, vol 3011. Springer, pp 50–64. https://doi.org/10.1007/978-3-540-24664-0_4

Belkhir N, Dréo J, Savéant P, Schoenauer M (2016a) Feature based algorithm configuration: A case study with differential evolution. In: Parallel problem solving from nature—PPSN XIV. Springer, pp 156–166. https://doi.org/10.1007/978-3-319-45823-6_15

Belkhir N, Dréo J, Savéant P, Schoenauer M (2016b) Surrogate assisted feature computation for continuous problems. In: Sellmann M, Vanschoren J, Festa P (eds) Learning and intelligent optimization. Springer, Berlin, pp 17–31

Belkhir N, Dréo J, Savéant P, Schoenauer M (2017) Per instance algorithm configuration of CMA-ES with limited budget. In: Proceedings of the genetic and evolutionary computation conference. ACM. https://doi.org/10.1145/3071178.3071343

Benjamini Y, Yekutieli D (2001) The control of the false discovery rate in multiple testing under dependency. Ann Stat 29(4):1165–1188

Bischl B, Mersmann O, Trautmann H, PreußM (2012a) Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: GECCO '12. ACM, pp 313–320. https://doi.org/10.1145/2330163.2330209

Bischl B, Mersmann O, Trautmann H, Weihs C (2012b) Resampling methods for meta-model validation with recommendations for evolutionary computation. Evol Comput 20(2):249–275

Crombecq K, Laermans E, Dhaene T (2011) Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. Eur J Oper Res 214(3):683–696. https://doi.org/10.1016/j.ejor.2011.05.032

Davidor Y (1991) Epistasis variance: a viewpoint on GA-hardness. In: Rawlins G (ed) FOGA I. Morgan Kauffmann, Burlington, pp 23–35

Efron B, Tibshirani R (1993) An introduction to the bootstrap. Chapman & Hall, London

Fonlupt C, Robilliard D, Preux P (1998) A bit-wise epistasis measure for binary search spaces. PPSN V LNCS 1498:47–56. https://doi.org/10.1007/BFb0056848

Graff M, Poli R (2010) Practical performance models of algorithms in evolutionary program induction and other domains. Artif Intell 174:1254–1276. https://doi.org/10.1016/j.artint.2010.07.005

Groppe D, Urbach T, Kutas M (2011) Mass univariate analysis of event-related brain potentials/fields I: a critical tutorial review. Psychophysiology 48(12):1711–1725. https://doi.org/10.1111/j.1469-8986.2011.01273.x

Hansen N, Auger A, Ros R, Finck S, Pošík P (2011a) Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: GECCO '11, pp 1689–1696. https://doi.org/10.1145/1830761.1830790

Hansen N, Ros R, Mauny N, Schoenauer M, Auger A (2011b) Impacts of invariance in search: when CMA-ES and PSO face ill-conditioned and non-separable problems. Appl Soft Comput 11(8):5755–5769. https://doi.org/10.1016/j.asoc.2011.03.001

Hansen N, Auger A, Finck S, Ros R (2014) Real-parameter black-box optimization benchmarking BBOB-2010: experimental setup. Tech. Rep. RR-7215, INRIA. http://coco.lri.fr/downloads/download15.02/bbobdocexperiment.pdf

He J, Reeves C, Witt C, Yao X (2007) A note on problem difficulty measures in black-box optimization: classification, realizations and predictability. Evol Comput 15(4):435–443. https://doi.org/10.1162/evco.2007.15.4.435

Hinkle D, Wiersma W, Jurs S (2003) Applied statistics for the behavioral sciences. Houghton Mifflin, Boston

Jones T, Forrest S (1995) Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of the sixth international conference on genetic algorithms. Morgan Kaufmann Publishers Inc., pp 184–192

Kang Y, Hyndman R, Smith-Miles K (2017) Visualising forecasting algorithm performance using time series instance spaces. Int J Forecast 33(2):345–358. https://doi.org/10.1016/j.ijforecast.2016.09.004

Kerschke P, Trautmann H (2019a) Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. Evol Comput 27(1):99–127. https://doi.org/10.1162/evco_a_00236

Kerschke P, Trautmann H (2019b) Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package flacco. In: Bauer N, Ickstadt K, Lübke K, Szepannek G, Trautmann H, Vichi M (eds) Applications in statistical computing—from music data analysis to industrial quality improvement, studies in classification, data analysis, and knowledge organization. Springer, Berlin, pp 93–123. https://doi.org/10.1007/978-3-030-25147-5_7

Kerschke P, PreußM, Wessing S, Trautmann H (2016) Low-budget exploratory landscape analysis on multiple peaks models. In: GECCO '16. ACM, New York, pp 229–236. https://doi.org/10.1145/2908812.2908845

Lunacek M, Whitley D (2006) The dispersion metric and the CMA evolution strategy. In: GECCO '06. ACM, New York, pp 477–484. https://doi.org/10.1145/1143997.1144085

Malan K, Engelbrecht A (2014) Characterising the searchability of continuous optimisation problems for PSO. Swarm Intell 8(4):1–28. https://doi.org/10.1007/s11721-014-0099-x

Marin J (2012) How landscape ruggedness influences the performance of real-coded algorithms: a comparative study. Soft Comput 16(4):683–698. https://doi.org/10.1007/s00500-011-0781-5

Mersmann O, PreußM, Trautmann H (2010) Benchmarking evolutionary algorithms: towards exploratory landscape analysis. In: PPSN XI. LNCS, vol 6238. Springer, pp 73–82. https://doi.org/10.1007/978-3-642-15844-5_8

Mersmann O, Bischl B, Trautmann H, PreußM, Weihs C, Rudolph G (2011) Exploratory landscape analysis. In: GECCO '11. ACM, pp 829–836. https://doi.org/10.1145/2001576.2001690

Miranda P, Prudêncio R, Pappa G (2017) H3ad: a hybrid hyperheuristic for algorithm design. Inf Sci 414:340–354. https://doi.org/10.1016/j.ins.2017.05.029

Morgan R, Gallagher M (2014) Sampling techniques and distance metrics in high dimensional continuous landscape analysis: limitations and improvements. IEEE Trans Evol Comput 18(3):456–461. https://doi.org/10.1109/TEVC.2013.2281521

Muñoz M (2020) LEOPARD: LEarning and OPtimization Archive of Research Data, version 1.0. https://doi.org/10.6084/m9.figshare.c.5106758

Muñoz M, Smith-Miles K (2015) Effects of function translation and dimensionality reduction on landscape analysis. In: IEEE CEC '15, pp 1336–1342. https://doi.org/10.1109/CEC.2015.7257043

Muñoz M, Smith-Miles K (2017) Performance analysis of continuous black-box optimization algorithms via footprints in instance space. Evol Comput 25(4):529–554. https://doi.org/10.1162/EVCO_a_00194

Muñoz M, Smith-Miles K (2020) Generating new space-filling test instances for continuous black-box optimization. Evol Comput 28(3):379–404. https://doi.org/10.1162/evco_a_00262

Muñoz M, Kirley M, Halgamuge S (2012) Landscape characterization of numerical optimization problems using biased scattered data. In: IEEE CEC '12, pp 1–8. https://doi.org/10.1109/CEC.2012.6256490

Muñoz M, Kirley M, Halgamuge S (2015a) Exploratory landscape analysis of continuous space optimization problems using information content. IEEE Trans Evol Comput 19(1):74–87. https://doi.org/10.1109/TEVC.2014.2302006

Muñoz M, Sun Y, Kirley M, Halgamuge S (2015b) Algorithm selection for black-box continuous optimization problems: a survey on methods and challenges. Inf Sci 317:224–245. https://doi.org/10.1016/j.ins.2015.05.010

Müller C, Sbalzarini I (2011) Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In: Applications of evolutionary computation. LNCS, vol 6624. Springer, pp 294–303. https://doi.org/10.1007/978-3-642-20525-5_30

Naudts B, Suys D, Verschoren A (1997) Epistasis as a basic concept in formal landscape analysis. In: Bäck T (ed) Proceedings of the 7th international conference on genetic algorithms. Morgan Kaufmann, pp 65–72

Pošík P (2005) On the utility of linear transformations for population-based optimization algorithms. IFAC Proc Vol 38(1):281–286. https://doi.org/10.3182/20050703-6-CZ-1902.01125**(16th IFAC World Congress)**

Renau Q, Dreo J, Doerr C, Doerr B (2019) Expressiveness and robustness of landscape features. In: GECCO'19. ACM Press. https://doi.org/10.1145/3319619.3326913

Renau Q, Doerr C, Dreo J, Doerr B (2020) Exploratory landscape analysis is strongly sensitive to the sampling strategy. In: Bäck T, Preuss M, Deutz A, Wang H, Doerr C, Emmerich M, Trautmann H (eds) Parallel problem solving from nature—PPSN XVI. Springer, Cham, pp 139–153

Rochet S, Slimane M, Venturini G (1996) Epistasis for real encoding in genetic algorithms. In: Australian and New Zealand conference on intelligent information systems, pp 268–271. https://doi.org/10.1109/ANZIIS.1996.573954

Rochet S, Venturini G, Slimane M, El Kharoubi E (1998) A critical and empirical study of epistasis measures for predicting GA performances: a summary. In: Third European conference on artificial evolution, pp 275–285. https://doi.org/10.1007/BFb0026607

Rosé H, Ebeling W, Asselmeyer T (1996) The density of states—a measure of the difficulty of optimisation problems. In: PPSN IV, LNCS, vol 1141. Springer, pp 208–217. https://doi.org/10.1007/3-540-61723-X_985

Sala R, Müller R (2020) Benchmarking for metaheuristic black-box optimization: perspectives and open challenges. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp 1–8

Saleem S, Gallagher M, Wood I (2019) Direct feature evaluation in black-box optimization using problem transformations. Evol Comput 27(1):75–98. https://doi.org/10.1162/evco_a_00247

Seo D, Moon B (2007) An information-theoretic analysis on the interactions of variables in combinatorial optimization problems. Evol Comput 15(2):169–198. https://doi.org/10.1162/evco.2007.15.2.169

Škvorc U, Eftimov T, Korošec P (2020) Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. Appl Soft Comput 90:106138. https://doi.org/10.1016/j.asoc.2020.106138

Smith-Miles K, Baatar D, Wreford B, Lewis R (2014) Towards objective measures of algorithm performance across instance space. Comput Oper Res 45:12–24. https://doi.org/10.1016/j.cor.2013.11.015

Stein M (1987) Large sample properties of simulations using latin hypercube sampling. Technometrics 29(2):143–151. https://doi.org/10.1080/00401706.1987.10488205

Storlie CB, Swiler LP, Helton JC, Sallaberry CJ (2009) Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. Reliab Eng Syst Saf 94(11):1735–1763. https://doi.org/10.1016/j.ress.2009.05.007

Stowell D, Plumbley M (2009) Fast multidimensional entropy estimation by k-d partitioning. IEEE Signal Process Lett 16(6):537–540. https://doi.org/10.1109/LSP.2009.2017346

Tian W, Song J, Li Z, de Wilde P (2014) Bootstrap techniques for sensitivity analysis and model selection in building thermal performance analysis. Appl Energy 135:320–328. https://doi.org/10.1016/j.apenergy.2014.08.110

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.