algorithms

Article

An Exploratory Landscape Analysis-Based Benchmark Suite

Ryan Dieter Lang [1,*] and Andries Petrus Engelbrecht [2]

[1] Computer Science Division, Stellenbosch University, Stellenbosch 7600, South Africa
[2] Department of Industrial Engineering and Computer Science Division, Stellenbosch University, Stellenbosch 7600, South Africa; engel@sun.ac.za
* Correspondence: langr@sun.ac.za

Abstract: The choice of which objective functions, or benchmark problems, should be used to test an optimization algorithm is a crucial part of the algorithm selection framework. Benchmark suites that are often used in the literature have been shown to exhibit poor coverage of the problem space. Exploratory landscape analysis can be used to quantify characteristics of objective functions. However, exploratory landscape analysis measures are based on samples of the objective function, and there is a lack of work on the appropriate choice of sample size needed to produce reliable measures. This study presents an approach to determine the minimum sample size needed to obtain robust exploratory landscape analysis measures. Based on reliable exploratory landscape analysis measures, a self-organizing feature map is used to cluster a comprehensive set of benchmark functions. From this, a benchmark suite that has better coverage of the single-objective, boundary-constrained problem space is proposed.

Keywords: exploratory landscape analysis; benchmarking; algorithm selection problem; sample size; single-objective boundary-constrained continuous optimization problems; black-box optimization

Citation: Lang, R.D.; Engelbrecht, A.P. An Exploratory Landscape Analysis-Based Benchmark Suite. *Algorithms* **2021**, *14*, 78. https://doi.org/10.3390/a14030078

Academic Editors: Mario A. Muñoz and Katherine Malan

Received: 30 November 2020
Accepted: 22 February 2021
Published: 27 February 2021

1. Introduction

The field of computational intelligence is inordinately powered by empirical analysis. Without theoretical derivations for the performance of optimization algorithms, one must compare said algorithms against one another by analyzing their performance on a collection of benchmark problems. The problem of selecting which algorithm to use is non-trivial and is not only limited to computational intelligence. The algorithm selection problem was formalized by Rice in 1976 [1]. Rice's framework defines four components, namely the problem space, the feature space, the algorithm space, and the performance measure space. The problem space contains all the possible problems that can exist for a particular problem type. The feature space consists of all possible measures that can describe the characteristics of problems found in the problem space. The algorithm space consists of all possible algorithms that can be used to solve the problems found in the problem space. Lastly, there is the performance space, which describes how well a particular algorithm solves the problems found in the problem space.

This study focuses on the case of single-objective, continuous-valued, boundary-constrained optimization problems.

The choice of benchmark problems to sample from the problem space has a direct impact on the relative performance of algorithms when they are compared with one another. To have a fair comparison of algorithm performance, the algorithms should be run on either (a) the same benchmark problems, or (b) benchmark problems that have similar characteristics. In the literature, it is common to make use of predefined benchmark suites to compare algorithm performance. The IEEE Congress on Evolutionary Computation (CEC) Special Sessions and Competitions on Real-Parameter Single-Objective Optimization [2–5] and the Genetic and Evolutionary Computation Conference (GECCO) Black-Box Optimization workshops [6] provide such benchmark suites. Furthermore, there are several other benchmark problems defined in the literature [7–10] that do not appear in the commonly used benchmark suites. Recent work has shown that the CEC and BBOB benchmark suites provide poor coverage of the problem space, from both the perspective of the performance space as well as the feature space. Garden and Engelbrecht [11] showed that the CEC 2005 and the BBOB benchmark suites have similar distributions for some fitness landscape analysis measures. Muñoz et al. [12] showed that the BBOB benchmark suite provides poor coverage of the problem space. Škvorc et al. [13] further showed that many of the CEC benchmark suites, as well as the BBOB benchmark suite, provide poor coverage of the problem space. Zhang and Halgamuge [14] and Christie et al. [15] both showed that benchmark functions in the literature are highly correlated from the perspective of algorithm performance. The optimal choice of benchmark functions to include in a benchmark suite is therefore still an open question.

Landscape analysis encompasses mathematical and statistical techniques used to quantify characteristics of an optimization problem, and therefore landscape analysis can be used to describe the feature space. Both fitness landscape analysis (FLA) [16] and exploratory landscape analysis (ELA) [17] can be used for this purpose. However, the subtle difference between the two techniques is that ELA measures are generally used in sets to relate to problem characteristics, whereas individual FLA measures can describe a problem characteristic independently. In the case of continuous-valued optimization problems, landscape analysis measures are calculated on a sample of values of the objective function. This is done since it is not computationally feasible to calculate all possible values of an objective function, since there are infinitely many such values in a continuous space. Furthermore, the algorithms used to generate the samples used in landscape analysis are stochastic. Consequently, landscape analysis measures can fluctuate due to the sampling method. It is, therefore, imperative that the landscape analysis measures produce results that do not fluctuate significantly, since this can affect decisions on which algorithm to select to solve a problem. If a landscape analysis measure does not have a large variance, it is deemed to be robust [18]. Landscape analysis measures must be robust, because if these measures are intended to be used as input to machine learning models for automated algorithm selection, reproducibility of the models is desirable. There is little focus in the literature on how to adequately choose the sample size for ELA. Kerschke et al. [19] noted that a sample size of $50 \times D$, where D is the dimensionality of the decision space, is sufficient to train a classifier on the BBOB benchmark suite to differentiate problem instances based on the number of funnels in the fitness landscape. However, this may not necessarily generalize to other problem characteristics.

The main contributions of this paper are (1) an approach to determine the sample size needed to produce robust measures from exploratory landscape analysis (ELA), and (2) a proposal of a single-objective boundary-constrained benchmark suite, using commonly found benchmark functions in the literature, that has better coverage of the single-objective, continuous-valued optimization problem space than benchmark suites that are currently used in the literature.

The remainder of this study is organized as follows: Section 2 contains an introduction to the concepts used within the study, Section 3 investigates the robustness of ELA measures and provides the sample size used for the remainder of the study, Section 4 contains the benchmark suite proposal. Finally, Section 5 concludes the study and discusses avenues for future work.

2. Background

This section contains an introduction to the concepts used in the remainder of this paper. First, benchmark problems and benchmark suites in single-objective, continuous-valued, boundary-constrained optimization are discussed. Then landscape analysis, in particular exploratory landscape analysis, and its relation to problem characterization is discussed. The suitability of the coverage of the problem space of currently used benchmark suites is discussed. Finally, a brief introduction to self-organizing feature maps is given.

2.1. Benchmark Functions

There are many commonly used single-objective boundary-constrained benchmark functions that have been defined in the literature. Jamil and Yang [7] provide a comprehensive overview of commonly used benchmark functions, and along with the Al-Roomi repository [9], the smoof R package [20], and the CIlib benchmark repository [21], these resources provide the benchmark function definitions that the majority of researchers use to test their algorithms. Although the basic benchmark functions are described in the literature, it is common for researchers to make use of benchmark suites, which contain a collection of these benchmark functions, to compare the performances of their algorithms. Both the GECCO and CEC conferences hold annual competitions in which such benchmark suites are defined.

The CEC benchmark suites are composed of three problem types: standard functions, hybrid functions, and composition functions. The standard functions make use of the standard function definitions of benchmark functions as they are described in the literature (called basic functions by the CEC organizers). However, input vectors are first rotated and shifted by predefined matrices and vectors. The hybrid functions are linear combinations of basic functions, where the input vector is divided up and each sub-vector becomes an input to a separate basic function. The composition functions are a weighted linear combination of both the basic, standard, and hybrid functions. Each year the benchmark suite used in the CEC competition changes. Despite this, there is an overlap in the functions used across the different benchmark suites. Nevertheless, the predefined rotation matrices and shift vectors differ for each competition and therefore the properties of the benchmarks can differ, despite having the same function definition. The study focuses on the CEC benchmark functions defined for the 2013, 2014, 2015, and 2017 competitions [2–5]. The CEC'13 functions are defined in 10, 30, and 50 dimensions. The CEC'14, CEC'15 and CEC'17 functions are defined in 10, 30, 50, and 100 dimensions. For all the CEC benchmark suites, the functions are defined within the hypercube $[-100, 100]^D$, where D is the dimensionality of the decision variable space. That is, the search space has a range of $[-100, 100]$ in each dimension.

The GECCO conference holds the Black-box Optimization Benchmarking (BBOB) workshop, in which several benchmarking suites are provided for different problem types. This study focuses on the BBOB benchmark suite [6] which contains 24 noiseless single-objective, boundary-constrained optimization problems. The BBOB benchmark suite has remained the same since its inception in 2009. In the BBOB benchmark suite, the optimization functions are categorized into five categories:

- Separable functions
- Functions with low or moderate conditioning
- Functions with high conditioning and unimodal
- Multi-modal functions with an adequate global structure
- Multi-modal functions with a weak global structure

The BBOB functions are defined in 2, 3, 5, 10, 20, and 40 dimensions. Furthermore, the functions are defined within the hypercube $[-5, 5]^D$, where D is the dimensionality of the decision variable space.

2.2. Landscape Analysis

In the context of the algorithm selection framework defined by Rice, the characteristics space can be defined using landscape analysis measures. Landscape analysis encompasses both fitness landscape analysis (FLA) [16] and exploratory landscape analysis (ELA) [17]. An individual FLA measure generally defines a single high-level characteristic of an optimization problem, whereas an individual ELA measure is generally associated with several high-level properties [17]. Mersmann et al. [22] defined eight high-level properties for optimization problems, namely

- Multi-modality, which refers to the number of local optima in the fitness landscape.
- Global structure, which refers to the underlying structure of a fitness landscape when removing local optima.
- Separability, which describes if an objective function can be decomposed into subproblems in which all the variables in each subproblem are independent of the variables in the other subproblems.
- Variable scaling, which describes the effect that scale has on the behavior of algorithms in different dimensions.
- Search space homogeneity, which describes the phase transitions between different areas of the fitness landscape, i.e., how the properties of the fitness landscape vary in different areas of the search space.
- Basin size homogeneity, which describes the differences in the sizes of the basins of attractions.
- Global to local optima contrast, which describes the difference in fitness values between local and global optima.
- Plateaus, which refers to areas of a fitness landscape in which the fitness values do not fluctuate significantly.

An individual FLA measure is interpretable in the sense that it can, on its own, describe a high-level property of a fitness landscape, as listed above. Individual ELA measures generally have less interpretability, and are designed to be used collectively and used as input for machine learning models. The distinction between ELA and FLA is not clear-cut, and often measures that were originally described as FLA measures are used in ELA.

Mersmann et al. [17] refer to groups of related ELA measures as feature sets. The flacco R package [23] provides an interface for calculating many landscape analysis measures in the literature. In this study, the following ELA feature sets from flacco are investigated:

- Dispersion (*disp*): Defined by Lunacek and Whitley [24], these measures describe the global structure of the objective function.
- Information content (*ic*): Defined by Muñoz et al. [25], these measures calculate the differences between points in the sampled fitness values to determine the ruggedness of the fitness landscape.
- Level-set (*ela_level*): Defined by Mersmann et al. [17], these measures split the initial sample into two groups, and then the performance of multiple classification algorithms is measured.
- Meta-model (*ela_meta*): Defined by Mersmann et al. [17], these measures determine how well the sampled fitness values fit linear and quadratic models.
- Nearest better clustering (*nbc*): Defined by Kerschke et al. [26], these measures calculate various statistics based on the comparison of the distances between the sample points' nearest neighbor and their nearest neighbor that has a better fitness value.
- Principal component analysis measures (*pca*): Defined by Kerschke and Trautmann [23], these measures perform principal component analysis on the sampled values in both the decision variable and fitness spaces.
- y-distribution features (*ela_distr*): Defined by Mersmann et al. [17], these measures describe the distribution of the fitness values obtained by the sampling algorithm.

These feature sets are chosen since they do not require any further objective function evaluations other than that of the initial sample. This simplifies the procedure needed to determine the point of robustness for the measures since the only variable with regards to the sampling is the size of the sample. In contrast, several FLA measures are calculated from samples generated by random walk algorithms. With random walk algorithms, there are two parameters, the number of points in the walk (sample) and the bound on the step size of the walk.

In addition, ELA measures that do not require further function evaluations are more likely to be feasible in practical usage of ELA for automated algorithm selection. This is because the computational costs associated with calculating function evaluations are generally significantly higher than the computational costs of calculating the ELA measures. Lastly, the ELA measures used in this study do not require knowledge of the objective function definition, which allows this study to generalize to black-box optimization problems.

In recent years, there has been work towards finding analyzing the effects of sample size on landscape analysis measures. Lang and Engelbrecht [18] studied the robustness of FLA measures, based on samples generated by random walks. Renau et al. [27] studied the effect robustness of ELA measures on the BBOB benchmark suite. However, both of these works do not specify at what sample size landscape analysis measures provide robust results.

Renau et al. [28] analyzed the effects of differing sampling strategies on the location and variance, or robustness, of ELA measures.

2.3. Coverage of the Problem Space

As noted by Bartz-Beielstein et al. [29] the quality of a benchmark suite can be evaluated using both the feature space and the performance space from the algorithm selection framework.

When analyzing the quality of a benchmark suite through the lens of the feature space, the characteristics of the benchmark problems are calculated, for example using landscape analysis. Then benchmark problems can be compared with one another through differences in the characteristics. Garden and Engelbrecht [11] calculated nine FLA measures on the CEC 2005 and BBOB benchmark suites and used a self-organizing feature map to project the problem space into a two-dimensional grid of 9-dimensional weight vectors. An analysis of the distributions of the FLA measures showed that those benchmark suites had poor representation for the measured characteristics. It was also shown that the functions within the benchmark suites are extremely similar. Muñoz et al. [12] calculated several ELA measures on the BBOB benchmark suite. It was shown that when using principal component analysis to project the space into two dimensions, the benchmark problems are highly similar and exhibit poor coverage of the problem space. Škvorc et al. [13] calculated ELA measures on several CEC benchmark suites, as well as the BBOB benchmark suite. Then, the t-SNE [30] dimensionality reduction algorithm is used to project the problem space into two dimensions. Even when using a more advanced dimensionality reduction technique compared to Muñoz et al., the benchmark functions were shown to have poor coverage of the problem space.

Analyzing the quality of a benchmark suite through the lens of the performance space entails running a collection of algorithm problems on a benchmark suite, and then comparing the performance metrics of algorithms against one another. Using various differential evolution algorithms, Christie et al. [15] , using fixed-budget performance measures showed that the BBOB benchmark suite has many highly correlated benchmark functions. Zhang and Halgamuge [14] analyzed many performance measures on a large collection of benchmark problems and showed that the CEC 2017 and BBOB benchmark suites have poor coverage of the problem space. Zhang and Halgamuge showed that the coverage of a collection of benchmark problems that commonly appear in the literature provides the largest coverage of the problem space.

2.4. Self-Organizing Feature Map

The self-organizing feature map (SOM) [31] is an artificial neural network-based nonlinear dimensionality reduction technique. The SOM can be used to map an m-dimensional continuous space onto an n-dimensional discrete grid, where typically $n = 2$. The SOM maintains the topological structure of vectors from the higher-dimensional space in the projected lower-dimensional space. Therefore, if two vectors are close in m-dimensional space, then they will be located close on the 2-dimensional grid.

A SOM grid consists of several codebook vectors, which represent where in the 2-dimensional grid space a particular node in the grid lies. The number of nodes in the grid is chosen by the practitioner and can have any rectangular shape. Engelbrecht [32] notes that the size of the grid should be at least equal to the number of independent patterns in the training set. Kohonen [33] notes that if one would like to find fine structures within the data, a large grid is needed. However, the computational costs of training a SOM increases with the number of nodes in the map.

With the unified distance matrix (U-matrix) [34], the SOM allows for visualization of high-dimensional space, and it can thus be used for the visualization of the distribution of benchmark functions in the problem space. Clustering algorithms can then be used on the U-matrix to determine which benchmark functions are similar. Additionally, the SOM can be used for exploratory data analysis. The distribution of the values for an individual dimension from the input space can be visualized by used component planes. The component planes are constructed by using a color scale range for the input parameter, or component, of interest. In this application of the SOM, this will indicate if a collection of benchmark functions is representative of a wide range of values for an ELA measure.

For a detailed discussion on the training of a SOM, the reader is directed to the literature [31–33].

3. Robustness of Exploratory Landscape Analysis Measures

This section discusses the need for robust exploratory landscape analysis measures and presents an approach to determine the sample size that results in robust exploratory landscape analysis measures. This approach is then applied to a large collection of benchmark functions, and the results are presented and analyzed. Finally, the choice of a sample size to use for ELA measures for the remainder of the study is determined.

3.1. Determining Robustness

The choice of the sample size for calculating ELA measures presents a trade-off between accuracy and computational costs. For smaller sample sizes, the computational cost will be low, but the accuracy of the resulting ELA measures is poor. For larger sample sizes, the computational costs will be high, and the accuracy of the resulting ELA measure will increase.

As noted by Muñoz et al. [35], an ELA measure $c(f,n)$ which is calculated on an objective function f from a sample size n is a random variable. Therefore, $c(f,n)$ has a probability distribution whose variance, σ^2, should converge to zero when n tends to infinity, otherwise σ^2 is dependent on f and n. Several independent runs of a measure $c(f,n)$ can be conducted to approximate the probability distribution. When the variance, σ^2, is small then $c(f,n)$ is said to be robust. Defining when the variance is small results in extra hyperparameters when using ELA, as a threshold needs to be defined for each ELA measure. Rather than defining an absolute threshold, this study makes use of a procedure that determines when the variance becomes small enough, relative to increasing sample sizes. This, coupled with the fact that the variance tends to zero as the sample size increases, allows one to determine the sample size needed to provide robust ELA measures.

To determine the sample size needed to produce a robust measure, non-parametric statistical tests are needed, since the $c(f,n)$ distributions are unknown and are unlikely to follow the normal distribution. In the literature, there are many hypothesis tests for equality of variance, also called homogeneity of variance, the most common being the F-test, which tests the hypothesis that two normal distributions have the same variance. The Levene test [36] is used to determine whether k samples have equal variances, and can be used as an alternative to the F-test when the population data does not follow the normal distribution. Consider k random samples, where the i-th sample has observations $x_{i1}, x_{i2}, \ldots, x_{in_i}$. Levene considers the absolute differences between each observation and its corresponding group mean, i.e., $d_{ij} = |x_{ij} - \bar{x}_{i.}|$, $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, n_i$, where

n_i is the number of observations in the i-th group, and $\bar{x}_{i.}$ is the sample mean for the i-th group. Then, the Levene test is defined as:

$$H_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2 \tag{1}$$

$$H_1 : \sigma_i^2 \neq \sigma_j^2 \text{for at least one pair} (i, j) \tag{2}$$

The Levene test statistic is then defined as:

$$L = \frac{N-k}{k-1} \frac{\sum_{i=1}^{k} n_i (\bar{d}_{i.} - \bar{d}_{..})^2}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} (d_{ij} - \bar{d}_{i.})^2} \tag{3}$$

where

$$\bar{d}_{i.} = \sum_{j=1}^{n_j} \frac{d_{ij}}{n_j} \qquad \bar{d}_{..} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \frac{d_{ij}}{N} \qquad N = \sum_{i=1}^{k} n_i \tag{4}$$

Levene transforms the population data by considering the absolute differences between each observation and its corresponding group mean, and therefore $d_{ij} = |x_{ij} - \bar{x}_{i.}|$, $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, n_i$, and $\bar{x}_{i.}$ is the sample mean for the i-th group in the above equations. Brown and Forsythe [37] proposed a modification to the Levene test which provides more robust results, in which the absolute differences between each observation and its corresponding group median is calculated. That is, $d_{ij} = |x_{ij} - \tilde{x}_{i.}|$, where $\tilde{x}_{i.}$ is the median of the i-th group.

However, it is only of interest whether the variance of the measures decreases as the sample size increases, and not if variances between sample sizes are equal or not. This is because a two-sided hypothesis test for the variance does not indicate if the variance of two samples is larger or smaller than the other. For this purpose, Levene trend tests [38] can be used to determine if there is a monotonic increasing or decreasing trend in a group of variances. As described in [39], such a hypothesis test can be set up as follows:

$$H_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2 \tag{5}$$

$$H_1 : \sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_k^2 \tag{6}$$

Then, all observations in a group i are assigned a score w_i, for each group $i = 1, \ldots, k$. Now, regress the transformed data, d_{ij}, on w_i and consider the regression slope

$$\hat{\beta} = \frac{\sum_{i=1}^{k} n_i (w_i - \bar{w})(\bar{d}_{i.} - \bar{d}_{..})}{\sum_{i=1}^{k} n_i (w_i - \bar{w})^2} \tag{7}$$

where

$$\bar{w} = \sum_{i=1}^{k} w_i \tag{8}$$

Under the null hypothesis, $\hat{\beta} = 0$ and the test statistic follows a t-distribution with $(N-1)$ degrees of freedom, where N is the total number of observations from all groups. Scores can be assigned as either linear or non-linear functions, which respectively allows testing for linear or non-linear trends in the variances. In this study, linear scores are investigated. That is, $w_i = 1 \forall i$.

The lawstat package [39] in R is used to perform the Levene trend test.

Now, to determine at what sample size a measure $c(f, n)$, for a particular objective function f, becomes robust, the following procedure is performed:

1. Choose the sample sizes $s = s_1, ..., s_M$ to be investigated.
2. For each sample size s_i, calculate the measure $c(f, s_i)$ for r independent runs.
3. Perform the Levene trend test on the above samples, for each pair of sample sizes, s_i and s_{i+1}. In this case, there $k = 2$ groups. Obtain the test statistic and p-value.

4. For each pair of sample sizes, if the resulting p-value is less than or equal to the predefined significance level, α, then the null hypothesis is rejected. This implies that it is likely that there is a monotonic decrease in the variance between the sample sizes. If the p-value is greater than α, then the null hypothesis cannot be rejected. It is then said that there is strong evidence that the variance between tequivalencyhe different sample sizes is equal.

When using the procedure described above, for a particular ELA measure, there are several possibilities with regards to the number of occurrences of p-values $< \alpha$:

1. Zero occurrences: This implies that there is no evidence that the variance is lower for any sample size. The smallest sample size is chosen as the point of robustness since there is no decrease in variance from increasing sample size.
2. One occurrence: The first sample size after the occurrence is chosen to be the point of robustness.
3. Two or more consecutive occurrences: The first sample size after the chain of consecutive occurrences is chosen as the point of robustness.
4. Two or more non-consecutive occurrences: The first sample size after the first chain of consecutive occurrences is chosen as the point of robustness.

Please note that when the null hypothesis is rejected for a pair of sample sizes, it implies that the variance of the larger sample size is statistically likely to be lower than the variance of the smaller sample size. Therefore, the larger sample size is chosen as the point of robustness.

Based on the observation of Muñoz et al. [35], the variance of a particular ELA measure tends to zero as the sample size increases. For the case of two or more non-consecutive occurrences of statistically significant pairs, Muñoz et al.'s observation implies that the first chain of statistically significant pairs is more likely to provide practically significant differences in variance than the second, or later, chain of statistically significant pairs. Therefore, the first sample size after the first chain of statistically significant pairs is chosen as the point of robustness.

3.2. Empirical Procedure

As noted in Section 2, there are several benchmark functions defined in the literature. In this study, the following benchmark functions are investigated:

- the BBOB benchmark suite, which contains 24 benchmark functions. This study focuses on only the first five instances of these functions, for a total of 120 benchmark functions;
- the CEC 2013 benchmark suite, which contains 28 benchmark functions [2];
- the CEC 2014 benchmark suite, which contains 30 benchmark functions [3];
- the CEC 2015 benchmark suite, which contains 15 benchmark functions [4];
- the CEC 2017 benchmark suite, which contains 29 benchmark functions [5]; and
- 118 miscellaneous benchmark functions obtained from various sources listed in Section 2.

Then, the ELA measures described in Section 2 are calculated for varying sample sizes for the 340 benchmark functions listed above. To calculate the ELA measures, the flacco library [23] is used. In particular, the following sample sizes are investigated: $50 \times D$, $100 \times D$, $200 \times D$, ..., $1000 \times D$, where D is the dimensionality of the decision variable space. The improved Latin hypercube sampling [19] algorithm is used to sample the points for the ELA measures. This study focuses on the case when $D = 10$ since it is the only dimensionality for which both the CEC and BBOB benchmark suites have been defined. Each of the investigated feature sets is calculated from the same generated sample, and therefore all features are calculated from the same sample. The ELA measures that are investigated are described in Section 2. The breakdown for the ELA measures is as follows: 16 dispersion measures, three y-distribution measures, 18 level-set measures, nine meta-model measures, five information content measures, five nearest better clustering measures, and eight principal component analysis measures, for a total of 64 measures. Each measure for all combinations of functions and sample sizes are calculated over 30 independent runs.

For this hypothesis test, the level of significance, α, is chosen as 5% a priori. Please note that the choice of the level of significance has a strong impact on the procedure for determining the point of robustness. If α is large, then it is likely that the Levene trend test will find statistically significant differences in the variance between the pairs of smaller sample sizes, and therefore the point of robustness will be a relatively small sample size. If α is small, then it is likely that the Levene trend test will either (i) find statistically significant differences in the variance between pairs of larger sample sizes, and the point of robustness will occur at larger sample sizes, or (ii) find no pairs of statistically significant differences in variances. To estimate the sampling distribution more accurately, bootstrapping is performed on the samples used as input to the Levene trend test, as described by Lim and Loh [40]. In the experiments, bootstrapping is performed with replacement, and the number of bootstrap samples is set to 10,000.

3.3. Results and Discussion

Figure 1 provides the distribution of the point of robustness for each of the investigated ELA measures on all the investigated benchmark functions.

Figure 1 indicates that most of the ELA measures have two dominating points of robustness, with peaks at sample sizes $50 \times D$ and $200 \times D$. The features that provide the lowest point of robustness are **pca.expl_var.cov_x**, **pca.expl_var.cov_init**, **pca.expl_var.cov_x**, **pca.expl_var.cor_init**, **ela_meta.lin_simple.coef.max_by_min**, and **ela_distr.number_of_peaks**, with large peaks at sample size $50 \times D$. The measures which have platykurtic distributions, i.e., wide and flat distributions, are the dispersion and level-set feature sets. These platykurtic distributions indicate that the number of sample sizes needed to produce robust ELA measures often differs for different benchmark functions.

Figure 1 also shows that measures in a particular feature set tend to have the same distribution for the point of robustness. This observation is most prominent for the dispersion feature set. These similar distributions may indicate that measures within a feature set are highly correlated.

Figure 2 contains the plots of the distribution of the point of robustness for all investigated benchmark suites. Figure 3 contains the plots of the distribution of the point of robustness for the combination of all investigated benchmark functions. These two figures combine the point of robustness across all investigated ELA measures for a particular benchmark suite. Thus, in Figure 1 the cynosure is the different ELA measures, and in Figures 2 and 3 the cynosure are the different benchmark suites.

Figure 2 indicates that the distribution of the point of robustness is roughly the same for the BBOB and CEC benchmark suites. These distributions appear to follow a negative binomial distribution, and this is validated with a goodness-of-fit test. Figure 2f contrasts the robustness results of the BBOB and CEC benchmark suites, and indicates that the miscellaneous functions generally have a point of robustness at $50 \times D$. It is hypothesized that the oscillation functions used in the BBOB and CEC benchmark suites induces more phase transitions in the fitness landscapes, whereas the collection of miscellaneous functions are not oscillated. However, further research is required to determine the true cause.

As seen in Figure 3, most benchmark functions provide robust ELA measures at sample sizes of $50 \times D$, $100 \times D$ and $200 \times D$. Since the different benchmark suites are defined for various search space dimensionalities, it is interesting to note that the distributions of the point of robustness do not change significantly between the benchmark suites. This implies that the improved Latin hypercube sampling algorithm is a good choice to generate samples for ELA, and is likely to provide good coverage of the function space, regardless of the size of the search space.

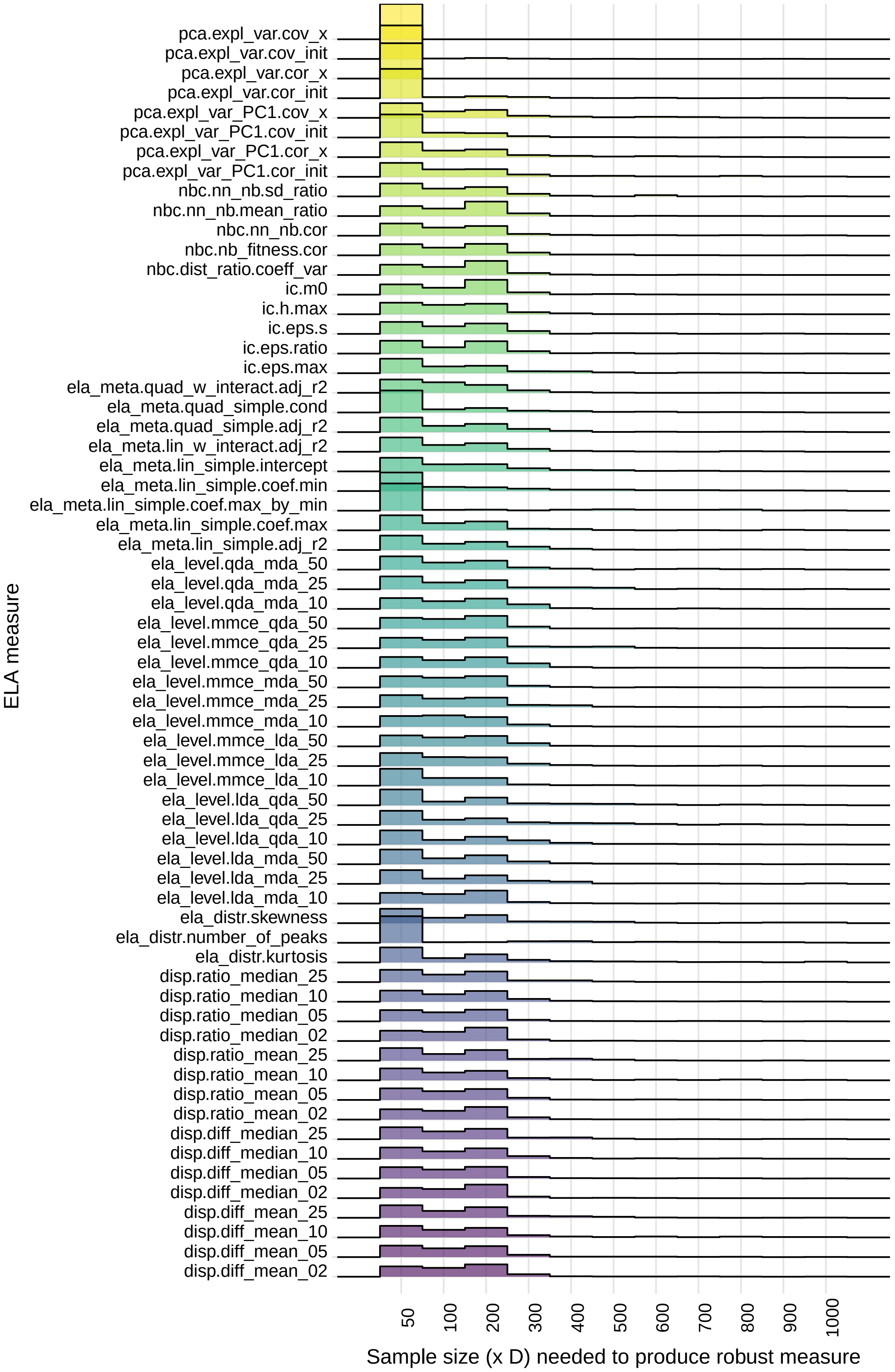
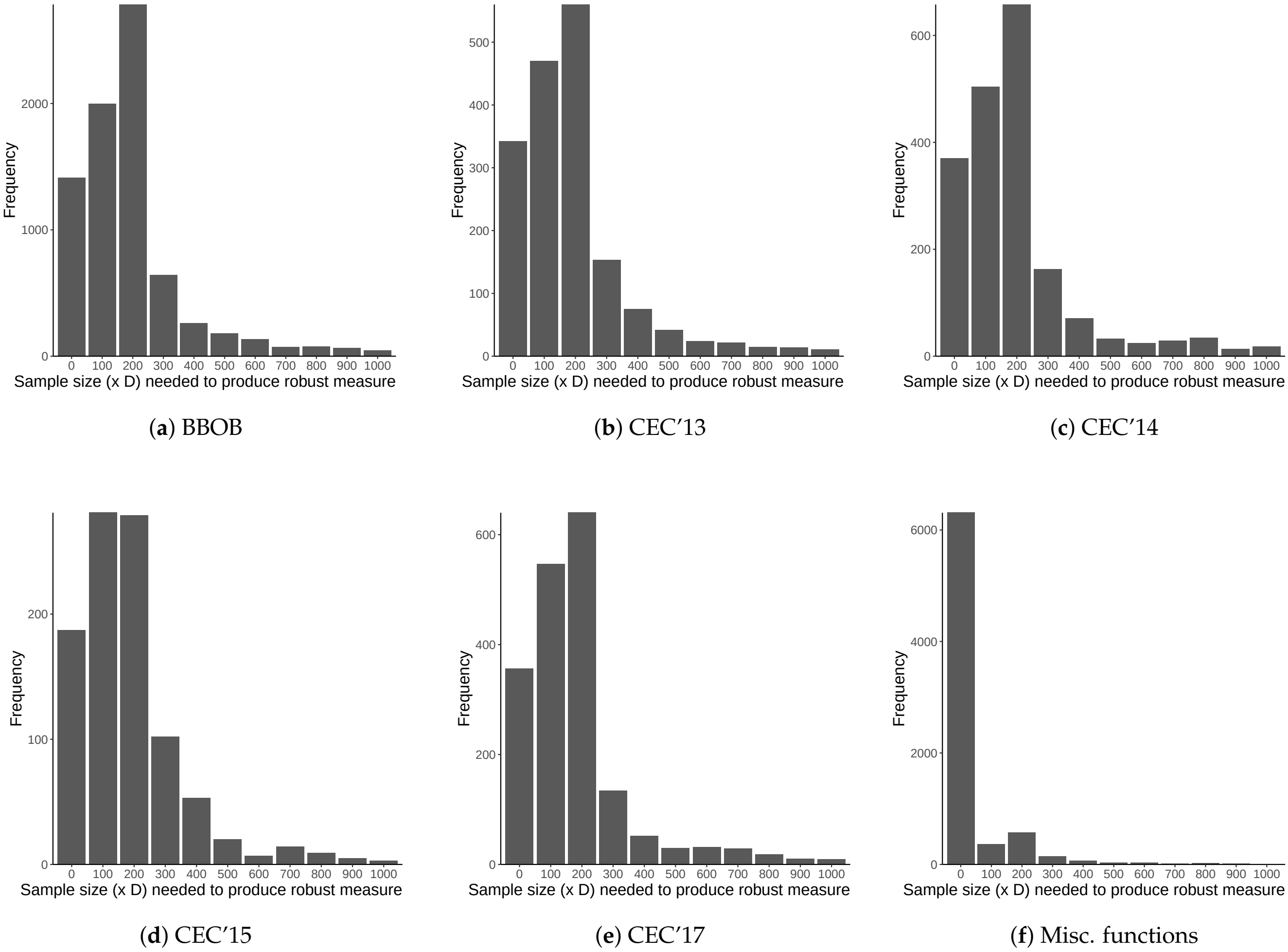


Figure 1. Distribution of the point of robustness for each of the investigated ELA measures across all benchmark functions in $D = 10$ dimensions.



(**a**) BBOB (**b**) CEC'13 (**c**) CEC'14

(**d**) CEC'15 (**e**) CEC'17 (**f**) Misc. functions

Figure 2. Plots of the distribution of the point of robustness for each of the investigated benchmark suites in $D = 10$ dimensions.

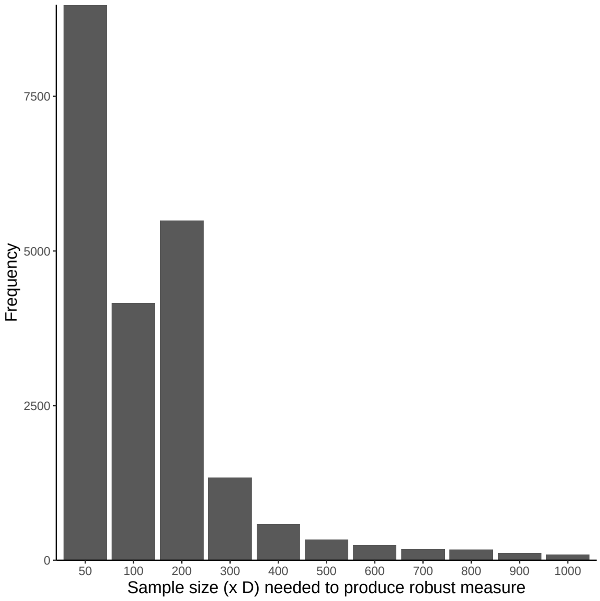


Figure 3. Plot of the combined distribution of the point of robustness of all investigate benchmark functions in $D = 10$ dimensions.

As noted above, an ELA measure $c(f, n)$ depends on both the function f and the sample size n. Since the procedure to determine the point of robustness for an ELA measure holds the function constant and varies the sample size, a summary statistic is needed to generalize the robustness for a particular ELA measure across a collection of functions. For this purpose, percentiles may be used. A percentile describes the percentage of observations that fall below a particular value. For example, the median is the 50th percentile. It implies that 50% of the observations in a data set lie above the median.

Table 1 contains the percentiles for the point of robustness over all the investigated benchmark functions.

Table 1. Percentiles for the point of robustness for all the investigated benchmark functions. The entries in the table represent sample sizes multiplied by D, where $D = 10$.

ELA Measure	10%	25%	50%	75%	90%	95%	99%	100%
disp.diff_mean_02	50	50	100	200	300	300	700	1000
disp.diff_mean_05	50	50	100	200	300	400	800	1000
disp.diff_mean_10	50	50	100	200	400	700	1000	1000
disp.diff_mean_25	50	50	100	200	300	500	700	1000
disp.diff_median_02	50	50	100	200	200	300	500	600
disp.diff_median_05	50	50	100	200	200	400	800	1000
disp.diff_median_10	50	50	100	200	300	600	900	1000
disp.diff_median_25	50	50	100	200	300	400	900	1000
disp.ratio_mean_02	50	50	100	200	300	300	700	1000
disp.ratio_mean_05	50	50	100	200	300	400	700	1000
disp.ratio_mean_10	50	50	100	200	400	700	1000	1000
disp.ratio_mean_25	50	50	100	200	400	400	600	900
disp.ratio_median_02	50	50	100	200	200	300	500	900
disp.ratio_median_05	50	50	100	200	200	400	800	1000
disp.ratio_median_10	50	50	100	200	300	500	900	1000
disp.ratio_median_25	50	50	100	200	300	400	900	1000
ela_distr.kurtosis	50	50	100	200	500	700	1000	1000
ela_distr.number_of_peaks	50	50	50	50	500	700	900	1000
ela_distr.skewness	50	50	100	200	500	800	1000	1000
ela_level.lda_mda_10	50	50	100	200	200	300	400	700
ela_level.lda_mda_25	50	50	100	200	400	600	1000	1000
ela_level.lda_mda_50	50	50	100	200	300	500	700	1000
ela_level.lda_qda_10	50	50	100	200	400	600	800	1000
ela_level.lda_qda_25	50	50	100	300	500	700	900	1000
ela_level.lda_qda_50	50	50	100	200	500	700	900	1000
ela_level.mmce_lda_10	50	50	100	200	200	400	700	1000
ela_level.mmce_lda_25	50	50	100	200	300	400	800	800
ela_level.mmce_lda_50	50	50	100	200	300	400	700	900
ela_level.mmce_mda_10	50	50	100	200	300	300	600	1000
ela_level.mmce_mda_25	50	50	100	200	300	400	900	1000
ela_level.mmce_mda_50	50	50	100	200	200	300	600	800
ela_level.mmce_qda_10	50	50	100	200	300	300	700	1000
ela_level.mmce_qda_25	50	50	100	200	400	500	800	1000
ela_level.mmce_qda_50	50	50	100	200	200	300	400	600
ela_level.qda_mda_10	50	50	100	200	300	400	700	900
ela_level.qda_mda_25	50	50	100	200	400	500	900	1000
ela_level.qda_mda_50	50	50	100	200	400	700	900	1000
ela_meta.lin_simple.adj_r2	50	50	100	200	300	500	900	1000
ela_meta.lin_simple.coef.max	50	50	100	200	300	400	900	1000
ela_meta.lin_simple.coef.max_by_min	50	50	50	50	600	700	900	1000
ela_meta.lin_simple.coef.min	50	50	50	300	600	700	1000	1000
ela_meta.lin_simple.intercept	50	50	100	200	400	500	800	900

Table 1. *Cont.*

ELA Measure	10%	25%	50%	75%	90%	95%	99%	100%
ela_meta.lin_w_interact.adj_r2	50	50	100	200	300	500	900	900
ela_meta.quad_simple.adj_r2	50	50	100	200	300	500	900	1000
ela_meta.quad_simple.cond	50	50	50	200	400	600	800	900
ela_meta.quad_w_interact.adj_r2	50	50	100	200	300	300	400	800
ic.eps.max	50	50	100	200	400	700	900	1000
ic.eps.ratio	50	50	100	200	300	300	700	1000
ic.eps.s	50	50	100	200	300	500	900	1000
ic.h.max	50	50	100	200	300	400	700	900
ic.m0	50	50	200	200	200	300	500	600
nbc.dist_ratio.coeff_var	50	50	100	200	200	300	500	800
nbc.nb_fitness.cor	50	50	100	200	300	400	600	1000
nbc.nn_nb.cor	50	50	100	200	400	600	1000	1000
nbc.nn_nb.mean_ratio	50	50	200	200	200	300	600	800
nbc.nn_nb.sd_ratio	50	50	100	200	300	600	800	1000
pca.expl_var_PC1.cor_init	50	50	100	200	400	700	900	1000
pca.expl_var_PC1.cor_x	50	50	100	200	400	600	800	900
pca.expl_var_PC1.cov_init	50	50	50	100	200	400	900	1000
pca.expl_var_PC1.cov_x	50	50	100	200	400	600	800	900
pca.expl_var.cor_init	50	50	50	50	200	300	900	1000
pca.expl_var.cor_x	50	50	50	50	50	50	50	50
pca.expl_var.cov_init	50	50	50	50	50	100	300	900
pca.expl_var.cov_x	50	50	50	50	50	50	50	50

To select the appropriate percentile, the practitioner should consider the sensitivity and consequences that the choice of the sample size will have on the later stages of the application of landscape analysis. As noted earlier, there is a trade-off between accuracy and computational costs when calculating ELA measures. For example, if ELA measures are used in automated algorithm selection, a practitioner may be satisfied with lower accuracy to keep computational costs down. The selection of a benchmark suite which is used to compare algorithms is a task that has significant effects on the algorithm selection problem. Therefore, in this case large computational costs from landscape analysis is acceptable so that a comprehensive, representative benchmark suite may be found. The larger the chosen percentile, the larger the sample size and the higher the computational costs and accuracy will be. For this purpose, the 95th percentile is chosen to determine which sample size will be used for the remainder of the study.

All ELA measures that belong to a particular feature set are calculated from the same sample. Additionally, several feature sets can be calculated from the same sample, as is the case with the feature sets used in this study. Using such a group of measures is advantageous since multiple measures can be calculated from the same sample, which consequently allows for more accurate characterization of a benchmark problem. However, as shown in Table 1, different measures provide robust results at different sample sizes. Since these measures are calculated from the same sample, the point of robustness of the group of measures should be defined as the largest sample size needed for any single measure within the collection of ELA measures.

When using the 95th percentile, Table 1 indicates that a sample size of 800 × D is the largest point of robustness for the whole collection of ELA measures. Therefore, for the remainder of the study, the ELA measures are calculated from a sample size of 800 × D.

4. Benchmark Suite Proposal

Now that the sample size that produces robust exploratory landscape analysis measures has been determined, the measures can be used reliably to characterize the single-objective, boundary-constrained problem space. A self-organizing feature map is used to project the problem space into two-dimensional grid representation, which allows for visualization of the distribution of the exploratory landscape analysis measures and clustering of the benchmark functions. First, the preprocessing of the data is discussed. Then the results of applying the self-organizing feature map to the benchmark function data are presented and analyzed. Finally, a benchmark suite that is representative of the problem space is proposed.

4.1. Preprocessing

To ensure the quality of the SOM, some preprocessing steps are needed:

1. Determine the sample size used to sample ELA measures. This was determined in the previous section as $800 \times D$.
2. Identify ELA measures that do not provide useful information, in other words, measures that are not expressive [27].
3. Identify ELA measures that are highly correlated to prevent multicollinearity.

As defined in [27], an ELA measure is defined as expressive if it can distinguish different benchmark problems. If the variance of an ELA measure is low, it implies that the measure is not expressive. When analyzing the distribution of the ELA measures, it is noted that **pca.expl_var.cov_x** and **pca.expl_var.cor_x** generate the same value across all investigated benchmark functions. All investigated benchmark functions generated only two different values for **pca.expl_var.cor_init**. Furthermore, the variance of both **pca.expl_var_PC1.cov_x** and **pca.expl_var_PC1.cor_x** are both 0.000012. Therefore, the above ELA measures are removed from all further analysis in this study, as they are unlikely to provide useful information to the SOM. It is noted that these non-expressive measures all have peaks, in their distribution of point of robustness, at a sample size of $50 \times D$, as illustrated in Figure 1.

Pearson's correlation coefficient can determine if a linear correlation between two ELA measures exists. However, a non-linear correlation relationship may exist between measures. In this case, Pearson's correlation will not be able to detect the dependency. The maximal information coefficient (MIC) [41] is a measure that captures the strength of the association between two random variables.

MIC has several useful properties:

- It produces values in between 0 and 1, with 0 indicating that there is no association between the two variables, and 1 indicating that the variables have a perfect noiseless relationship. This allows for easy interpretation of the MIC score.
- It captures a wide range of relationships, both functional and non-functional.
- It is symmetric, which implies that $MIC(X, Y) = MIC(Y, X)$.

There is no rule of thumb, as with Pearson's correlation, to determine thresholds of when associations are deemed to be strong. To be conservative in removing ELA measures, a threshold of 0.9 is selected. With a MIC score of larger than 0.9, an ELA measure can be said to explain a second ELA measure well, and therefore the second ELA measure is redundant.

The minerva R package [42] is used to calculate the MIC. To calculate the MIC, the median value for each ELA measure of all the functions used in Section 3 is taken. Then, the MIC is calculated for each pair of ELA measures.

To reduce the number of redundant ELA measures, groups of highly associated measures are found. This is done by creating a graph based on the MIC scores. Each ELA measure is represented as a node in this graph. If the MIC score is larger than 0.9, then an edge is added between the two ELA measures. Each group of correlated measures will then form an independent set. A representative ELA measure can be chosen from each group, and the remaining measures are not used as input for the SOM.

The above procedure resulted in six groups of highly associated ELA measures. The graph generated by the above procedure is shown in Figure 4. Figure A1 in Appendix A indicates the MIC scores between all investigated ELA measures.

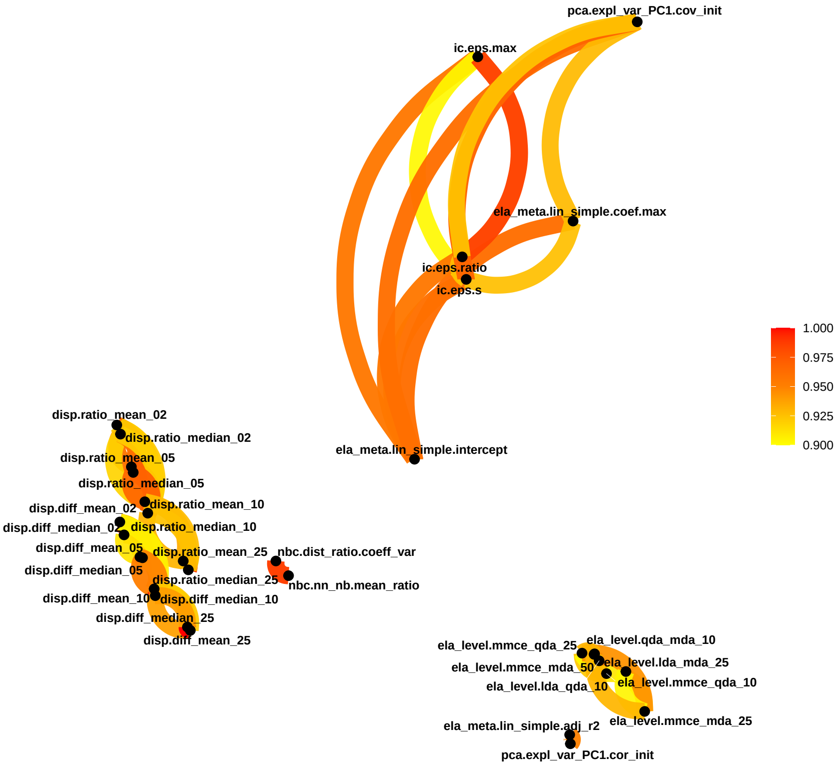


Figure 4. Plot of the graph based on the MIC scores. Nodes represent ELA measures. Edges represent a strong association between two ELA measures.

From each group, the representative ELA measure is chosen as the measure which has the smallest point of robustness. As noted earlier, in this study, the 95th percentile is chosen to summarize the point of robustness across functions. If there are ties in the smallest point of robustness, then the tie is broken arbitrarily. This ensures that the final set of ELA measures used as input to the SOM provides characteristics that are as reliable as possible.

Therefore, from the highly associated groups, the following ELA measures are kept: **disp.ratio_mean_02**, **disp.diff_mean_02**, **pca.expl_var_PC1.cor_init**, **pca.expl_var_PC1.cov_init**, **ela_level.lda_qda_10** and **nbc.dist_ratio.coeff_var**.

Thus, the final set of ELA measures that are used as input for the SOM are the following:

- **disp.ratio_mean_02**
- **disp.diff_mean_02**
- **ela_meta.lin_simple.coef.min**
- **ela_meta.lin_simple.coef.max_by_min**
- **ela_meta.lin_w_interact.adj_r2**
- **ela_meta.quad_simple.adj_r2**
- **ela_meta.quad_simple.cond**
- **ela_meta.quad_w_interact.adj_r2**
- **ela_level.mmce_lda_10**
- **ela_level.mmce_mda_10**
- **ela_level.lda_qda_10**
- **ela_level.lda_mda_10**
- **ela_level.mmce_lda_25**
- **ela_level.lda_qda_25**
- **ela_level.qda_mda_25**
- **ela_level.mmce_lda_50**
- **ela_level.mmce_qda_50**
- **ela_level.lda_qda_50**
- **ela_level.lda_mda_50**
- **ela_level.qda_mda_50**
- **ela_distr.skewness**
- **ela_distr.kurtosis**
- **ela_distr.number_of_peaks**
- **ic.h.max**
- **ic.m0**
- **nbc.nn_nb.sd_ratio**
- **nbc.nn_nb.cor**
- **nbc.dist_ratio.coeff_var**
- **nbc.nb_fitness.cor**
- **pca.expl_var.cov_init**
- **pca.expl_var_PC1.cov_init**
- **pca.expl_var_PC1.cor_init**

4.2. Self-Organizing Feature Map

The Kohonen R package [43] was used to generate the SOM models in this study. The set of measures listed in the previous section are used as input for the SOM. These measures are calculated for 30 independent runs on the 340 benchmark functions described in Section 3. This results in a dataset of 10,200 training patterns. As mentioned in Section 3, this study focuses on 10-dimensional benchmark functions. To prevent any of the ELA measures from dominating the training of the SOM, the input is normalized to the range $[0, 1]$.

The stochastic training rule [32] was used to train the SOM. For the stochastic training rule, the learning rate decreases linearly from 0.1 to 0.01 as a function of the training epoch. The SOM was trained for 100,000 epochs. The positions of the codebook vectors in the grid are updated using a Gaussian neighborhood function. Euclidean distances are used when determining the winning codebook vector for the SOM.

Several SOM models with differing grid sizes and dimensions were trained on the data. The U-matrix of the 75 by 75 grid provided the best visualization of the clustering structures [33]. Therefore, this map was chosen for the analysis.

Figure 5a contains the U-matrix for the trained SOM. From this figure, it can be seen that the grid size is sufficiently large to see the fine grain clustering structures. This figure suggests that there are many natural clusterings. However, there are also several codebook vectors which are topologically similar. This is highlighted by the large grouping of nodes in the bottom left and right corners of the U-matrix.

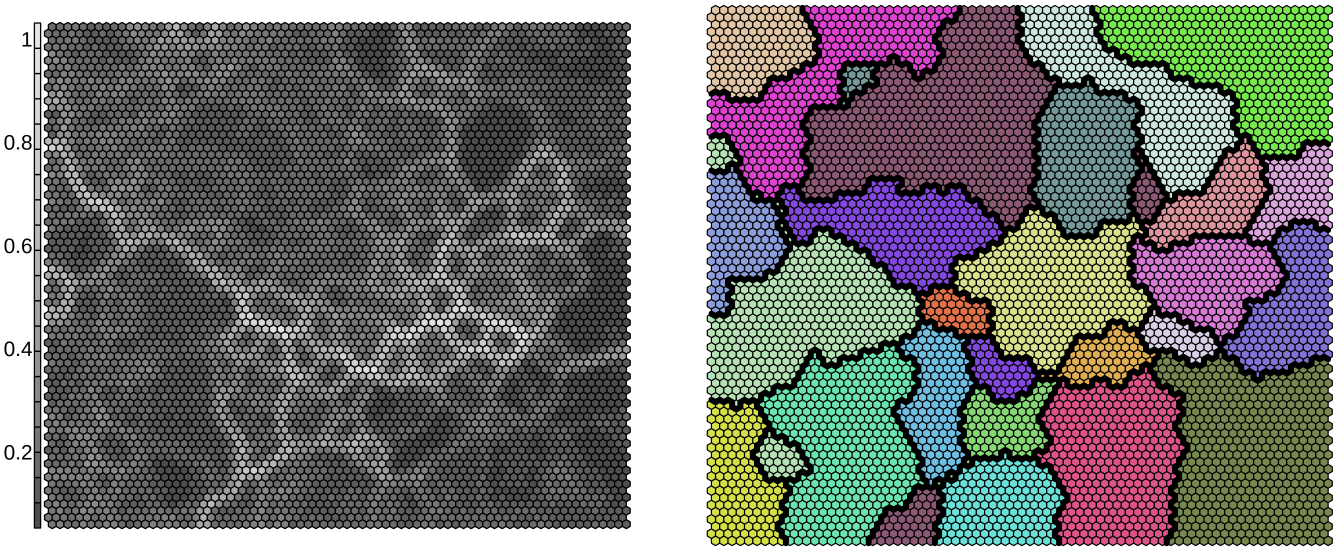
The benchmark suites from both the BBOB and CEC competitions typically have between 20 and 30 benchmark functions included. This is a reasonable number of benchmark problems, as this many functions are likely to provide a good representation of the problem space, provided that the functions are not similar. This number of functions is also not large enough for the computational costs to be excessive for the design and analysis of algorithms.

The NbClust R package [44] was used to cluster the benchmark functions, as well as to validate the number of clusters. To cluster the codebook vectors in the SOM, the Ward hierarchical clustering algorithm [45] was used. Ward's method works by minimizing the total within-cluster variance. The dissimilarity between the codebook vectors is calculated using Euclidean distance. Once the codebook vectors of the SOM were clustered, the validity of the clustering is quantified based on the Davies–Bouldin index [46]. The Davies–Bouldin index measures the cohesion of a cluster. A lower Davies–Bouldin score implies that the clusters are compact and well separated. Table 2 contains the corresponding Davies–Bouldin scores. From this table, it is observed that the optimal number of clusters is 24.

Table 2. Davies–Bouldin scores for differing number of clusters of the codebook vectors of the SOM, based on Ward's clustering method.

Number of Clusters	Davies–Bouldin Score
20	1.3568
21	1.3758
22	1.3566
23	1.3481
24	**1.3228**
25	1.3593
26	1.3350
27	1.3412
28	1.3726
29	1.3594
30	1.3498

Since all 30 independent runs of the ELA measures are used as input to the SOM, the quality of the clustering can be asserted by examining to which cluster each of the independent runs are assigned. Ideally, all independent runs for a particular function should appear in the same cluster. For the clustering in Figure 5b, six of the 340 benchmark functions have independent runs assigned to three different clusters. However, upon inspection, these functions that have not been uniquely assigned to the clusters have been assigned to clusters that are topological neighbors in the SOM grid. This is likely since there are a different benchmark functions that have similar ELA measures, which affects the clustering of the codebook vectors.



(**a**)

(**b**)

Figure 5. The U-Matrix and the clustering of the codebook vectors of the SOM, trained on all 30 independent runs of 243 benchmark functions. (**a**) U-Matrix; (**b**) Clustering of the codebook vectors.

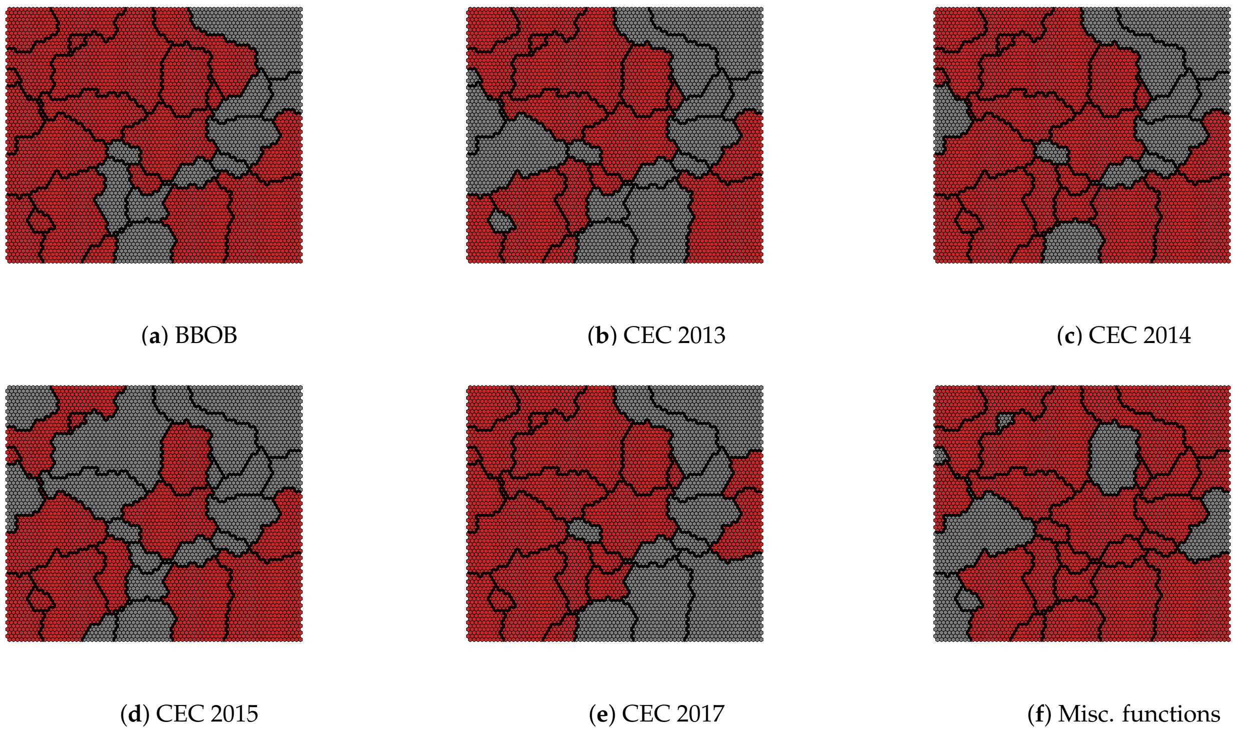
To illustrate the coverage of the problem space, Figure 6 illustrates for each benchmark suite which clusters contain functions from that benchmark suite. That is, if the cluster is gray then the benchmark suite does not contain a function in that cluster. If the cluster is red, then the benchmark suite contains a function in that cluster.

Figure 6a–e show that the commonly used benchmark suites in the literature do not cover large parts of the problem space. These figures validate the findings in [12,13] that the CEC and BBOB benchmark suites do not provide sufficient coverage of the problem space with respect to ELA characteristics.

Figure 6b–e show that the CEC benchmark suites have similar coverage of the problem space over the years. Please note that the BBOB benchmark suite and miscellaneous benchmark suite contain more than 100 functions each, and therefore it is inequitable to compare their coverage of the problem space with the coverage provided by the CEC benchmark suites. It is noted that the CEC 2014 benchmark suite provides the best coverage of the problem space out of the set of commonly used benchmark suites.

As seen in Figure 6a–f there are portions of the problem space that each collection of functions do not cover. Thus, the proposed benchmark suite will require benchmark problems from the multiple benchmark suites.

The SOM maintains the topological structure of the input space in the two-dimensional grid. This means that if the benchmark functions that are investigated in this study collectively have poor coverage of the problem space, then this will not be represented by the SOM's grid. In other words, the SOM does not indicate the positioning of the benchmark functions relative to the boundaries of the problem space. It is, therefore, possible that the investigated benchmark functions are tightly clustered in the greater, infinite problem space. However, to the best knowledge of the authors, all the benchmark functions that are found in the literature are included in this study. Consequently, the proposed benchmark suite will be the best representation of the presently understood problem space. If increasingly better coverage of the problem space is required, then new benchmark functions need to be created. This can be done either by an expert or through a benchmark problem generator.



(**a**) BBOB

(**b**) CEC 2013

(**c**) CEC 2014

(**d**) CEC 2015

(**e**) CEC 2017

(**f**) Misc. functions

Figure 6. Coverage of the problem space for each of the benchmark suites. Illustrated by highlighting in red which clusters contain functions from the particular benchmark suite.

Investigations into where new benchmark functions are required can be based on the component maps of the SOM, as can be seen in Figures A2–A4. It is remarked that the distribution of some ELA measures are highly skewed, which indicates that the collective set of benchmark functions has poor coverage of the problem space and that new benchmark problems are required to have better coverage of the problem space.

4.3. Selecting a Benchmark Suite

As mentioned above, a SOM can be clustered based on the distances between the codebook vectors. The U-matrix of a SOM represents how similar the codebook vectors are to one another and thus can be used to visualize the clustering structures of the SOM. To find a minimal set of functions to be representative of the landscape characteristics, one representative benchmark function is selected from each cluster. Some suggestions are presented on how to select a suitable representative. However, note that many combinations of benchmark functions can be chosen - resulting in many possible benchmark suites. This is congruous since the training patterns presented to the SOM are the vectors of ELA measures for the benchmark functions and hence functions that have similar landscape characteristics will be clustered together.

The results of the clustering allow for the benchmark suite to be used in situations such as competitions at conferences, or in creation of automated algorithm selection oracles, such as hyper-heuristics. Two sets of benchmark functions can be chosen as the training suite and the test suite, and the generalization abilities of algorithms can be observed. This will prevent overfitting algorithms on common benchmark functions and allow for better applicability for both unseen and real-world problems.

To select a benchmark function from a cluster, the following criteria are used:

- Functions from the miscellaneous group are preferable, as they do not require additional information such as rotation matrices and shift vectors, which is the case with the CEC and BBOB benchmark suites.
- Functions from the BBOB benchmark suite are preferred over functions from CEC benchmark suites, as there is a large amount of information, such as algorithm performance, for the BBOB benchmark suite.

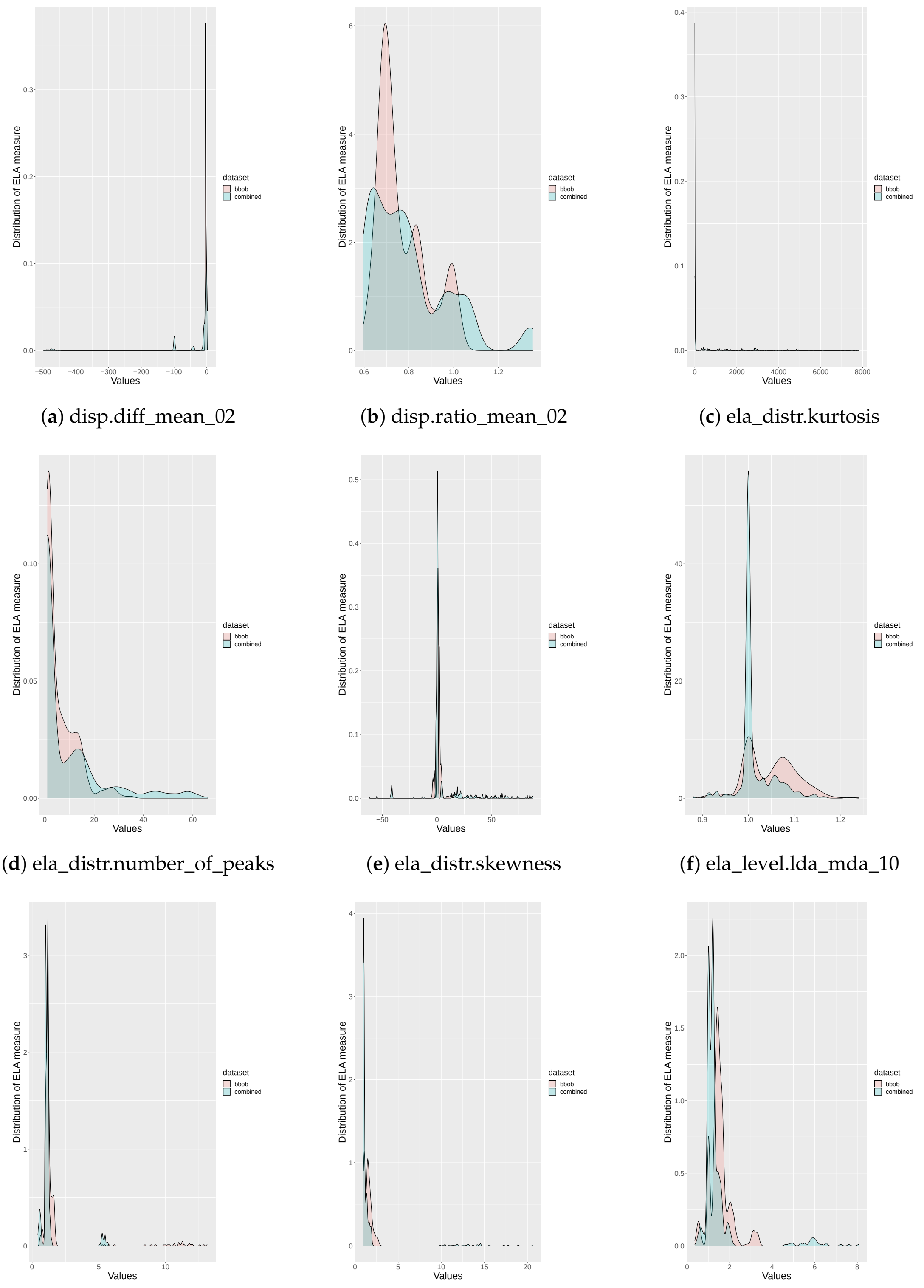
Using the above criteria, the proposed benchmark suite is composed of the following functions:

- Schwefel 1 [7]
- Ripple 25 [7]
- Exponential [7]
- Needle Eye [9]
- Step Function N. 3 [7]
- Generalized Giunta [7]
- Generalized Paviani [7]
- Brown [7]
- Cosine Mixture [7]
- Mishra 7 [7]
- Mishra 1 [7]
- Generalized Price 2 [7]
- Generalized Egg Crate [7]
- Rosenbrock [7]
- Pinter 2 [7]
- Qing [7]
- BBOB FID 2 IID 1 [6]
- BBOB FID 6 IID 1 [6]
- BBOB FID 16 IID 1 [6]
- BBOB FID 17 IID 2 [6]
- Generalized Drop-Wave [21]
- Bonyadi-Michalewicz [21]
- Discus [21]
- Elliptic [21]

This benchmark suite, along with the miscellaneous benchmark functions analyzed in this study is available online at [47].

Please note that since a benchmark function is chosen from every cluster in the SOM, this benchmark suite provides the best coverage over all landscape characteristics currently available in the literature. This claim is validated by analyzing the distributions of the ELA measures, as seen in Figures 7–10. These plots illustrate the differences between the distributions of the proposed benchmark suite's ELA measures and the first instances of the BBOB benchmark suite functions' ELA measures. In most of the plots, the distribution for the proposed benchmark suite has a wider spread than the distribution for the BBOB benchmark suite. This wider spread indicates that the proposed benchmark suite provides better coverage of individual ELA measures and collectively provides better coverage of the problem space. This is validated by calculating the standard deviation, which is a measure of spread, of the values for each of the ELA measures. The proposed benchmark suite has a larger standard deviation for 26 of the 32 ELA measures used as input to the SOM.

Please note that the distributions of several the ELA measures have extreme peaks. This corresponds to the equivalent component planes for the ELA measures in Figures A2–A4, which illustrate small portions of the SOM grid that have extreme values. These plots may indicate that the available benchmark problems in the literature have poor coverage for a particular ELA measure, or that the aforementioned ELA measures are not expressive.

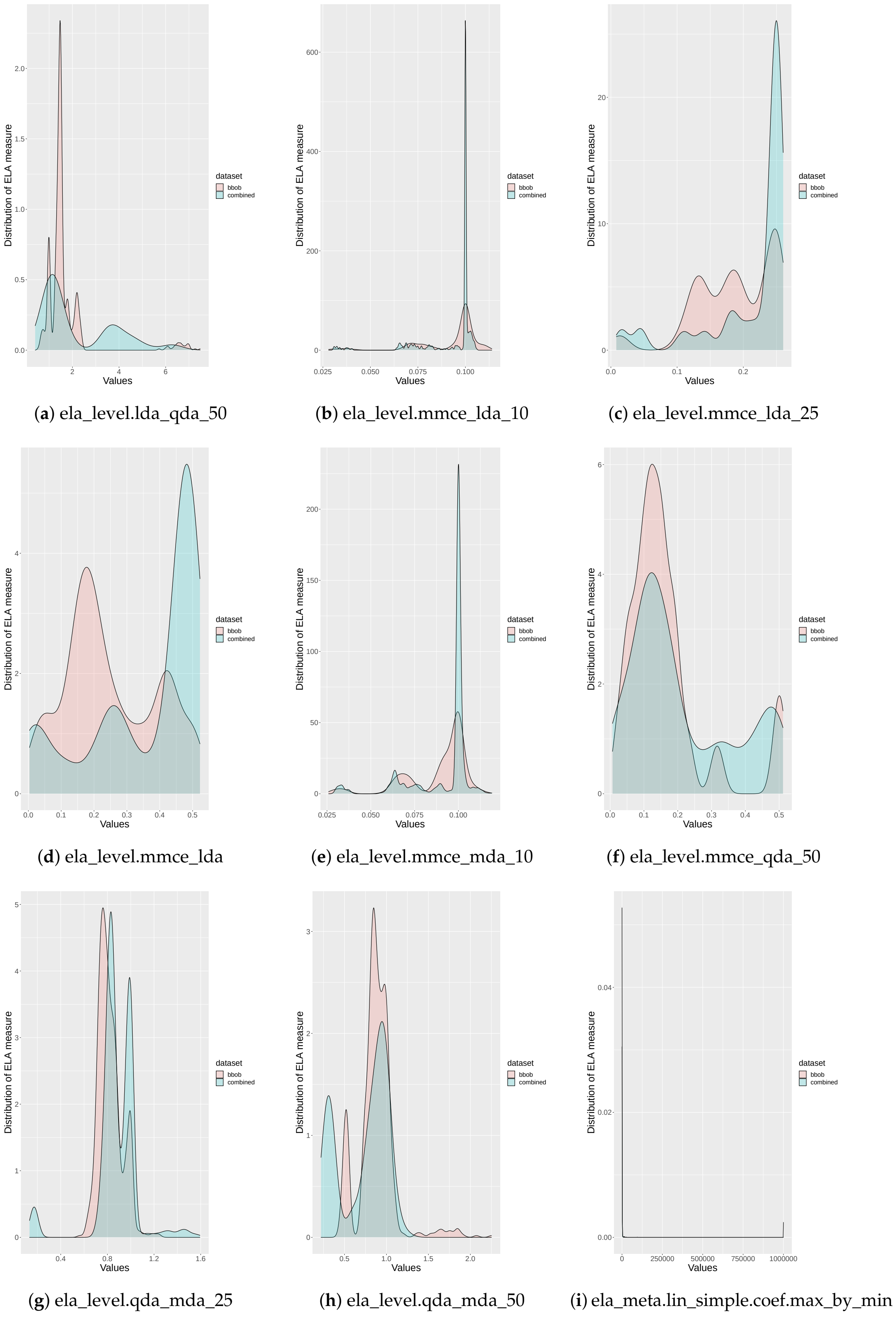


(**a**) disp.diff_mean_02 (**b**) disp.ratio_mean_02 (**c**) ela_distr.kurtosis

(**d**) ela_distr.number_of_peaks (**e**) ela_distr.skewness (**f**) ela_level.lda_mda_10

(**g**) ela_level.lda_mda_50 (**h**) ela_level.lda_qda_10 (**i**) ela_level.lda_qda_25

Figure 7. Comparison of the distributions of the BBOB benchmark suite and the proposed benchmark suite for all of the ELA measures used as input to the SOM.



(**a**) ela_level.lda_qda_50

(**b**) ela_level.mmce_lda_10

(**c**) ela_level.mmce_lda_25

(**d**) ela_level.mmce_lda

(**e**) ela_level.mmce_mda_10

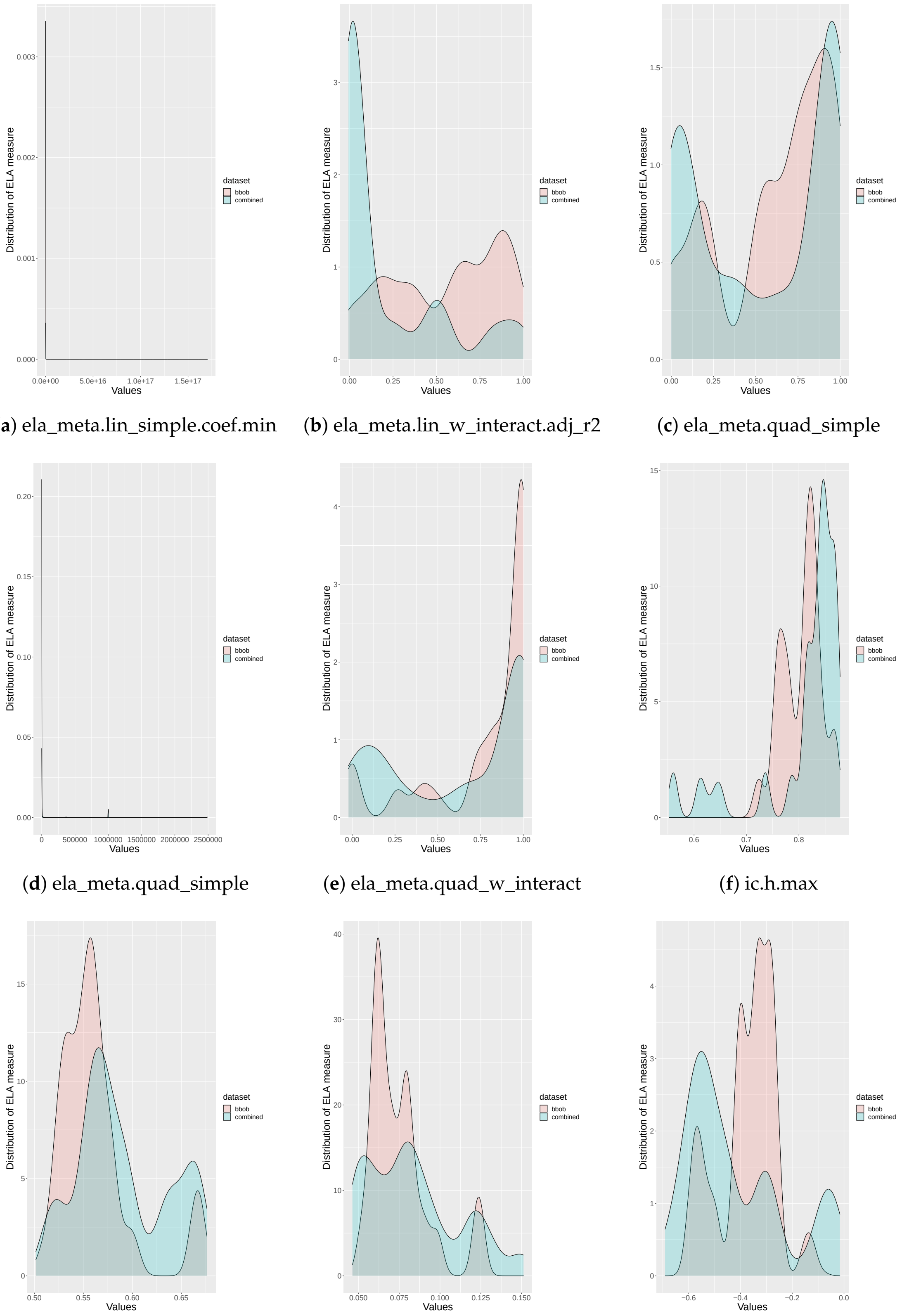
(**f**) ela_level.mmce_qda_50

(**g**) ela_level.qda_mda_25

(**h**) ela_level.qda_mda_50

(**i**) ela_meta.lin_simple.coef.max_by_min

Figure 8. Comparison of the distributions of the BBOB benchmark suite and the proposed benchmark suite for all of the ELA measures used as input to the SOM (continued).

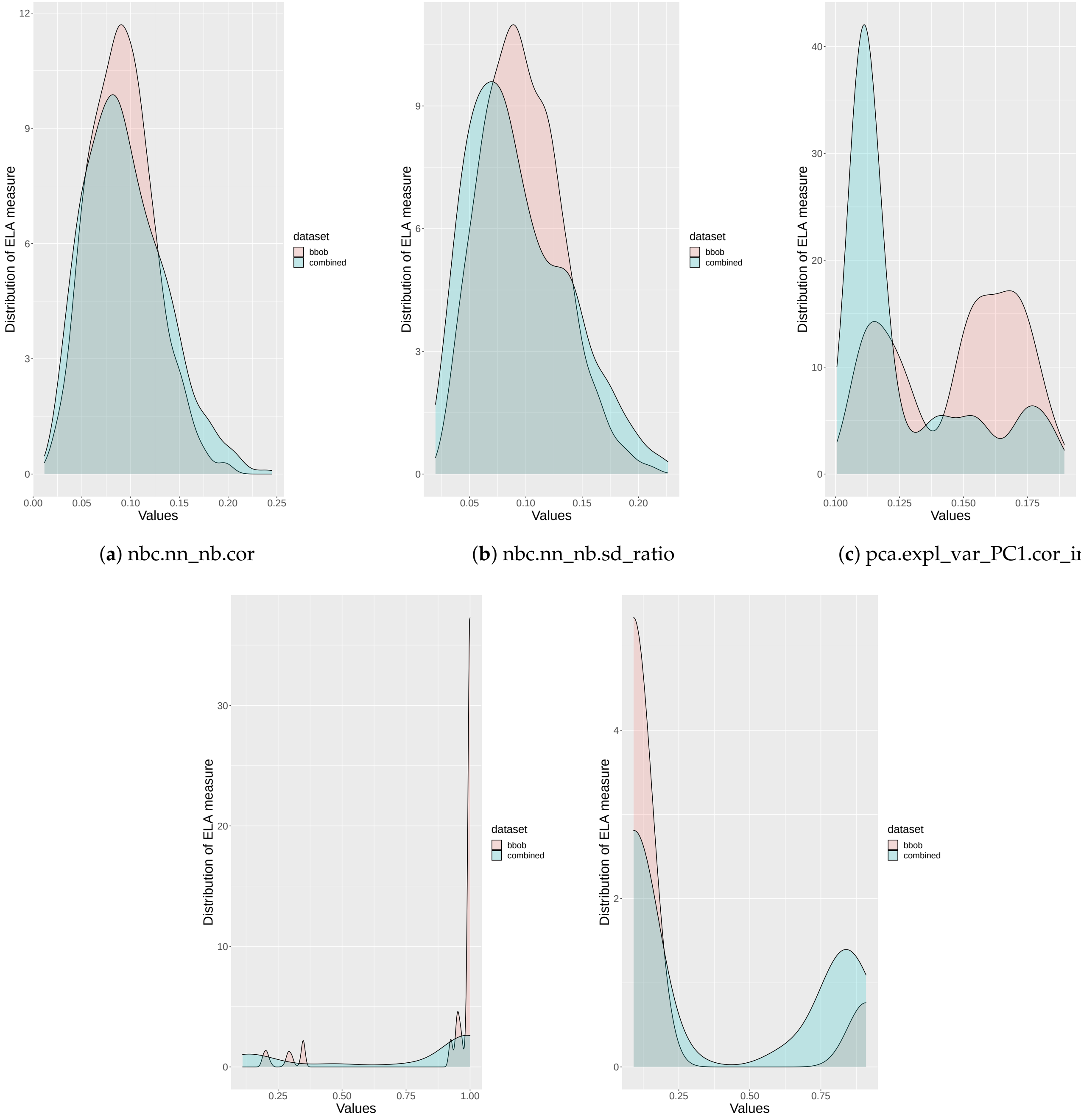


(**a**) ela_meta.lin_simple.coef.min (**b**) ela_meta.lin_w_interact.adj_r2 (**c**) ela_meta.quad_simple

(**d**) ela_meta.quad_simple (**e**) ela_meta.quad_w_interact (**f**) ic.h.max

(**g**) ic.m0 (**h**) nbc.dist_ratio.coeff_var (**i**) nbc.nb_fitness.cor

Figure 9. Comparison of the distributions of the BBOB benchmark suite and the proposed benchmark suite for all of the ELA measures used as input to the SOM (continued).



(**a**) nbc.nn_nb.cor

(**b**) nbc.nn_nb.sd_ratio

(**c**) pca.expl_var_PC1.cor_init

(**d**) pca.expl_var_PC1.cov_init

(**e**) pca.expl_var.cov_init

Figure 10. Comparison of the distributions of the BBOB benchmark suite and the proposed benchmark suite for all of the ELA measures used as input to the SOM (continued).

5. Conclusions and Future Work

The larger the sample size used in exploratory landscape analysis, the more accurate the characterization of the objective function. However, evaluation of an objective function is typically the most computationally expensive task in optimization. Thus, to minimize computational costs, the sample size should be chosen as small as possible, while providing accurate results. This study proposes an approach to determine the sample size needed to obtain robust exploratory landscape analysis measures.

For the purposes of creating a benchmark suite, a larger computational budget can be afforded than say, using exploratory landscape analysis for automated algorithm selection. The sample sized need for generating robust exploratory landscape analysis measures is generalized over benchmark functions that are commonly used in the literature. It was shown that a sample size of $800 \times D$, where $D = 10$, is sufficient to provide robust exploratory landscape analysis measures for 95% of benchmark problems.

This study focused on the case of 10-dimensional benchmark functions. A future step is to perform the analysis for different dimensions. This analysis will indicate the effect of dimensionality on the robustness of exploratory landscape analysis measures.

The case of 10-dimensional problems was chosen since it is the only dimensionality for which both the CEC and BBOB benchmark suites are defined. This is indicative of the issue that the benchmark suites found in the literature are not generalizable. Ideally a benchmark suite should provide benchmark functions which can be evaluated for any dimensionality of the decision variable space.

To propose a benchmark suite that provides better coverage of the problem space, a self-organizing feature map was trained on the robust exploratory landscape analysis measures of many benchmark functions. A benchmark suite is proposed by clustering the codebook vectors of the self-organizing feature map, and then selecting a single representative benchmark function from each cluster. The proposed benchmark suite contains 24 benchmark functions. This study showed that the coverage of the proposed benchmark suite is significantly better than the CEC and BBOB benchmark suites.

In this study, the exploratory landscape analysis measures that are used as input to the self-organizing feature map was calculated using a feature selection approach using the maximal information coefficient metric. Using maximal information coefficient allows one to find associations between the exploratory landscape analysis measures. Based on the maximal information coefficient scores, redundant exploratory landscape analysis measures were removed from further analysis. Future work could include a comparing the effects on the benchmark clustering of different feature selection techniques.

From the viewpoint of the algorithm selection problem, these findings are significant, since the generalizability of an algorithm's performance to unseen problems is greatly affected by the performance of the algorithm on test problems. Therefore, using the proposed benchmark suite is an important step in the algorithm selection problem and consequently automated algorithm selection.

As this study focused on the selection of a benchmark suite from the perspective of the problem characteristics space, an important future step is to investigate how this benchmark suite selection affects the performance space. It is hypothesized that the relative rankings of a collection of algorithms on, say the BBOB benchmark suite, versus the proposed benchmark suite will be notably different.

The link between artificial benchmark functions, which are analyzed in this study, and real-world optimization problems is unclear. However, if a benchmark suite provides better coverage of the artificial problem space, it is likely that the coverage of the real-world problem space will improve. Consequently, the proposed benchmark suite will aid in generalizability of algorithm performance to real-world problems.

This study focused on the case of single-objective, boundary-constrained optimization problems. As the exploratory landscape analysis measures used in this study do not require additional function evaluations, the presented procedures are generalizable to black-box optimization problems. The procedures described in this paper can be extended to propose benchmark suites for different problem categories, such as multi-objective or constrained optimization problems. To extend these procedures to other problem spaces, exploratory landscape analysis measures are needed to characterize such benchmark problems.

Author Contributions: Conceptualization, R.D.L. and A.P.E.; methodology, R.D.L.; software, R.D.L.; validation, R.D.L.; formal analysis, R.D.L.; investigation, R.D.L.; resources, R.D.L. and A.P.E.; data curation, R.D.L.; writing—original draft preparation, R.D.L.; writing—review and editing, A.P.E.; visualization, R.D.L.; supervision, A.P.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FLA	fitness landscape analysis
ELA	exploratory landscape analysis
LA	landscape analysis
SOM	Self-organizing feature map
GECCO	Genetic and Evolutionary Computation Conference
CEC	IEEE Congress on Evolutionary Computation
MIC	Maximal information coefficient

Appendix A. Associations between ELA Measures

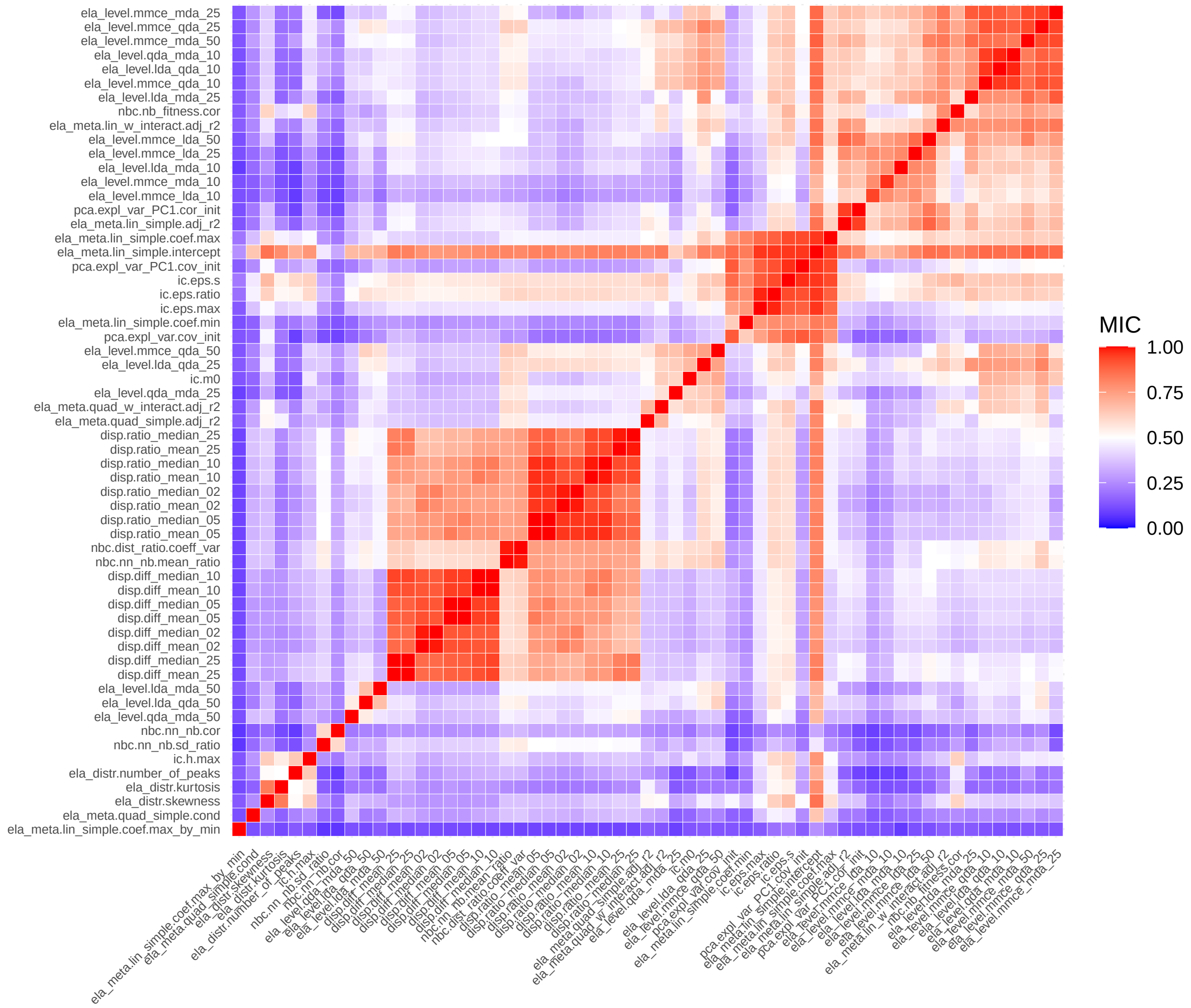
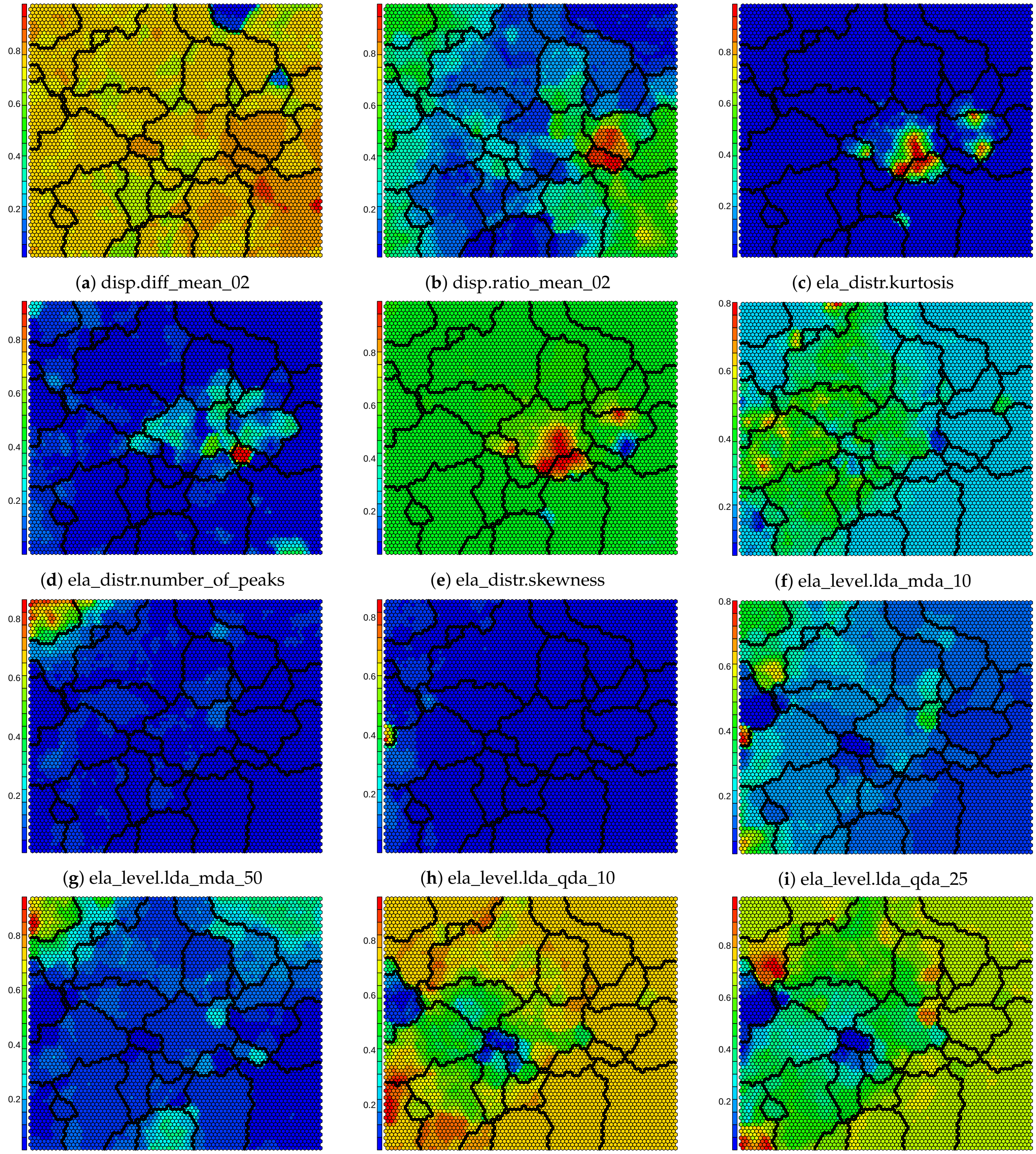


Figure A1. Plot of the maximal information coefficient scores between each of the investigated ELA measures.

Appendix B. Component Maps for the Self-Organizing Map



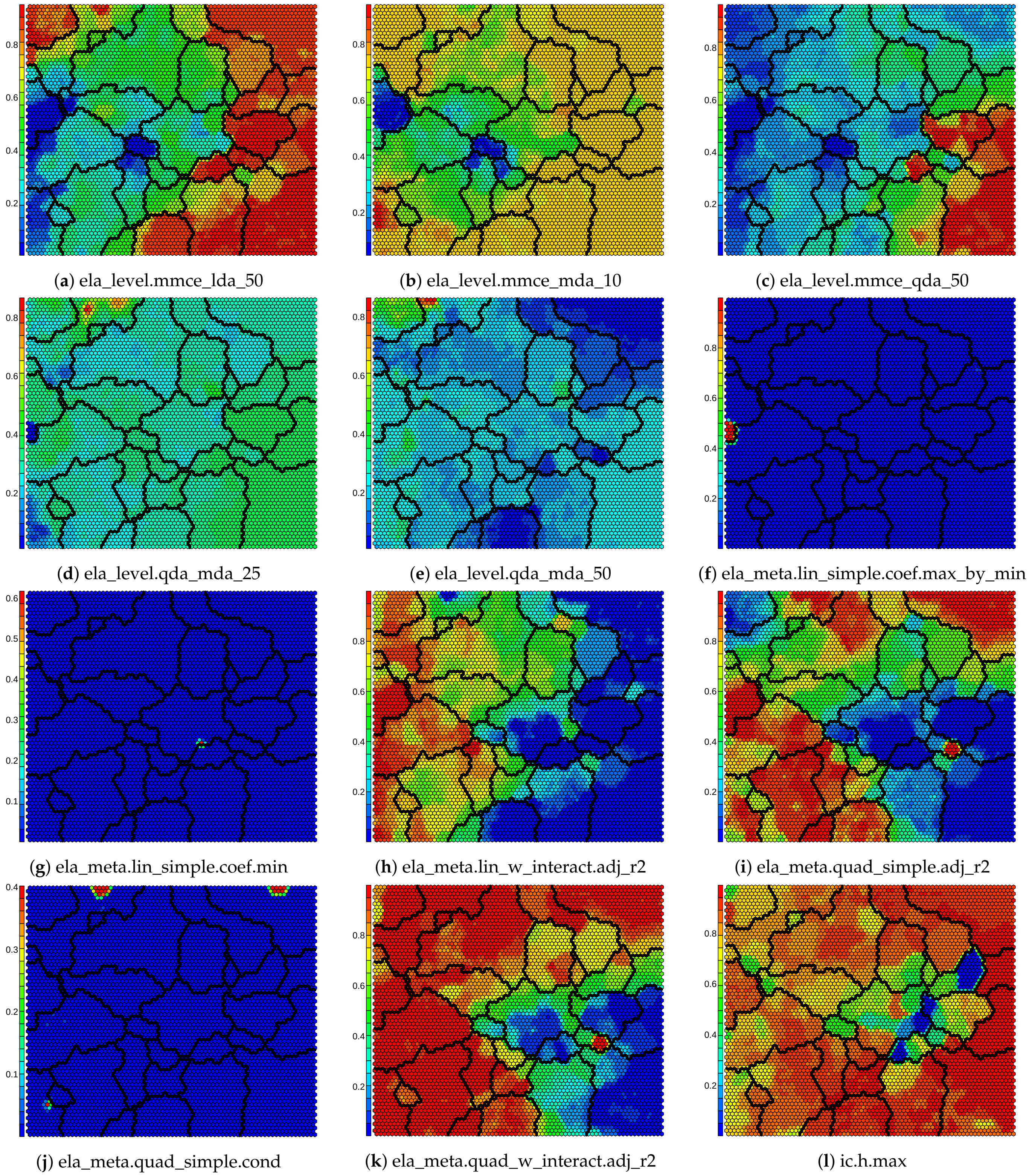
(**a**) disp.diff_mean_02 (**b**) disp.ratio_mean_02 (**c**) ela_distr.kurtosis

(**d**) ela_distr.number_of_peaks (**e**) ela_distr.skewness (**f**) ela_level.lda_mda_10

(**g**) ela_level.lda_mda_50 (**h**) ela_level.lda_qda_10 (**i**) ela_level.lda_qda_25

(**j**) ela_level.lda_qda_50 (**k**) ela_level.mmce_lda_10 (**l**) ela_level.mmce_lda_25

Figure A2. Normalized component maps from the clustered SOM.



(**a**) ela_level.mmce_lda_50

(**b**) ela_level.mmce_mda_10

(**c**) ela_level.mmce_qda_50

(**d**) ela_level.qda_mda_25

(**e**) ela_level.qda_mda_50

(**f**) ela_meta.lin_simple.coef.max_by_min

(**g**) ela_meta.lin_simple.coef.min

(**h**) ela_meta.lin_w_interact.adj_r2

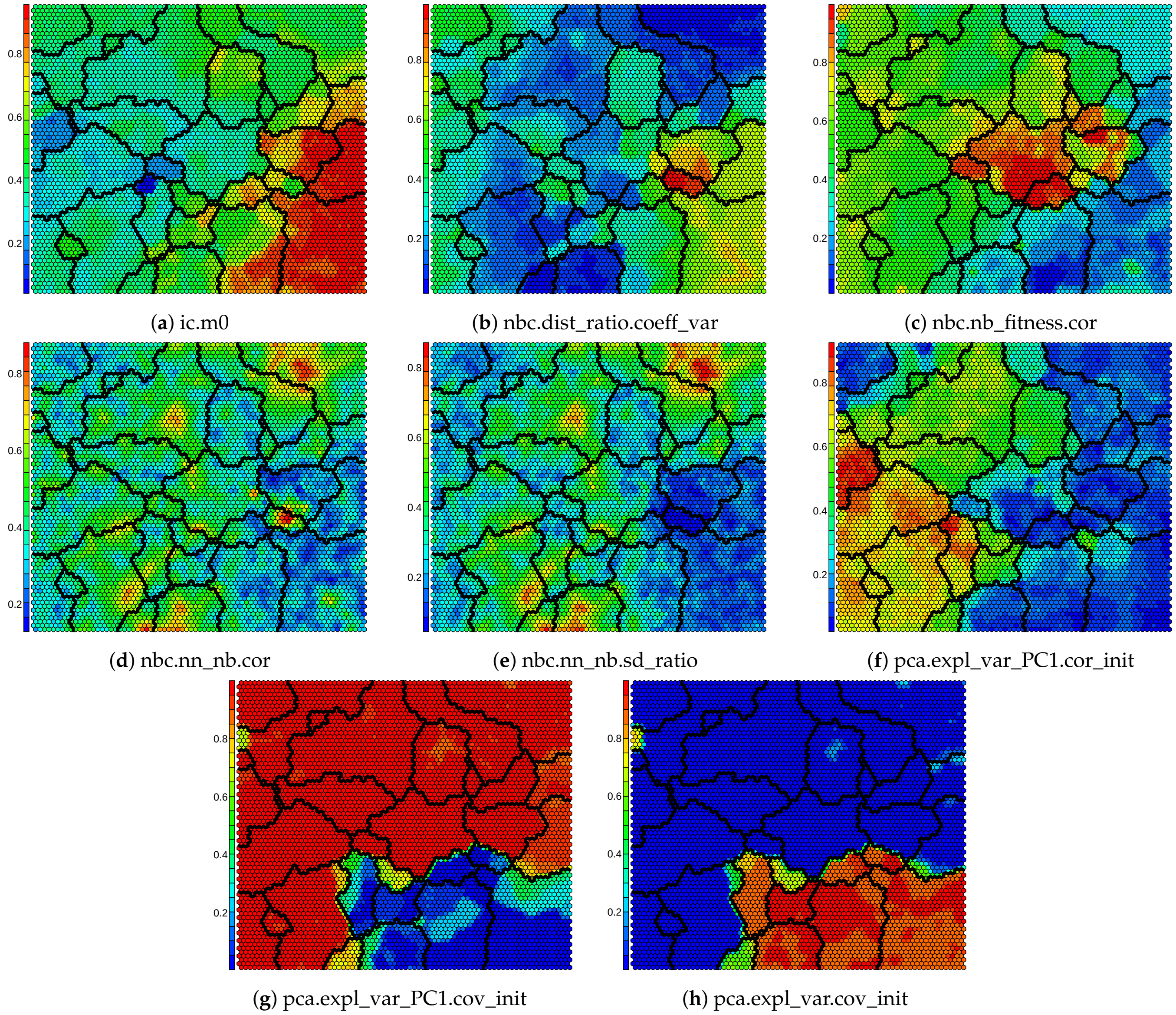
(**i**) ela_meta.quad_simple.adj_r2

(**j**) ela_meta.quad_simple.cond

(**k**) ela_meta.quad_w_interact.adj_r2

(**l**) ic.h.max

Figure A3. Normalized component maps from the clustered SOM (continued).



(**a**) ic.m0 (**b**) nbc.dist_ratio.coeff_var (**c**) nbc.nb_fitness.cor

(**d**) nbc.nn_nb.cor (**e**) nbc.nn_nb.sd_ratio (**f**) pca.expl_var_PC1.cor_init

(**g**) pca.expl_var_PC1.cov_init (**h**) pca.expl_var.cov_init

Figure A4. Normalized component maps from the clustered SOM (continued).

References

1. Rice, J.R. The Algorithm Selection Problem. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 1976; Volume 15, pp. 65–118. [CrossRef]
2. Liang, J.; Qu, B.; Suganthan, P.; Hernández-Díaz, A. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. 2013. Available online: https://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf (accessed on 20 February 2021).
3. Liang, J.; Qu, B.; Suganthan, P. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. 2013. Available online: https://bee22.com/resources/Liang%20CEC2014.pdf (accessed on 20 February 2021).
4. Liang, J.; Qu, B.; Suganthan, P.; Chen, Q. Problem Definitions and Evaluation Criteria for the Cec 2015 Competition on Learning-Based Real-Parameter Single Objective Optimization. 2014. Available online: https://al-roomi.org/multimedia/CEC_Database/CEC2015/RealParameterOptimization/LearningBasedOptimization/CEC2015_LearningBasedOptimization_TechnicalReport.pdf (accessed on 20 February 2021).
5. Wu, G.; Mallipeddi, R.; Suganthan, P. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016. Available online: https://www.researchgate.net/profile/Guohua-Wu-5/publication/317228117_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2017_Competition_and_Special_Session_on_Constrained_Single_Objective_Real-Parameter_Optimization/links/5982cdbaa6fdcc8b56f59104/Problem-Definitions-and-Evaluation-Criteria-for-the-CEC-2017-Competition-and-Special-Session-on-Constrained-Single-Objective-Real-Parameter-Optimization.pdf (accessed on 20 February 2021).

6. Hansen, N.; Finck, S.; Ros, R.; Auger, A. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions*; Research Report RR-6829; INRIA: Le Chesnay-Rocquencourt, France, 2009. Available online: https://hal.inria.fr/inria-00362633v2 (accessed on 26 February 2021).
7. Jamil, M.; Yang, X. A Literature Survey of Benchmark Functions For Global Optimization Problems. *arXiv* **2013**, arXiv:1308.4008.
8. Adorio, E.P.; Diliman, U. *Mvf-Multivariate Test Functions Library in C for Unconstrained Global Optimization*; GeoCities: Quezon City, Philippines, 2005; pp. 100–104.
9. Al-Roomi, A.R. *Unconstrained Single-Objective Benchmark Functions Repository*; 2015. Available online: https://www.al-roomi.org/component/content/article?id=175:generalized-rosenbrock-s-valley-banana-or-2nd-de-jong-s-function (accessed on 26 February 2021).
10. Hedar, A.R. *Test Functions for Unconstrained Global Optimization*; System Optimization Laboratory, Kyoto University: Kyoto, Japan, 25 May 2013. Available online: http://www-optima.amp.i.kyotou.ac.jp/member/student/hedar/Hedar_files/TestGO.htm (accessed on 26 February 2021).
11. Garden, R.W.; Engelbrecht, A.P. Analysis and classification of optimisation benchmark functions and benchmark suites. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1641–1649.
12. Muñoz, M.A.; Smith-Miles, K. Generating New Space-Filling Test Instances for Continuous Black-Box Optimization. *Evol. Comput.* **2020**, *28*, 379–404. [CrossRef]
13. Škvorc, U.; Eftimov, T.; Korošec, P. Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis. *Appl. Soft Comput.* **2020**, *90*, 106138. [CrossRef]
14. Zhang, Y.W.; Halgamuge, S.K. Similarity of Continuous Optimization Problems from the Algorithm Performance Perspective. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 2949–2957.
15. Christie, L.A.; Brownlee, A.E.I.; Woodward, J.R. Investigating Benchmark Correlations When Comparing Algorithms with Parameter Tuning. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'18, Kyoto, Japan, 15–19 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 209–210. [CrossRef]
16. Malan, K.M.; Engelbrecht, A.P. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Inf. Sci.* **2013**, *241*, 148–163. [CrossRef]
17. Mersmann, O.; Bischl, B.; Trautmann, H.; Preuss, M.; Weihs, C.; Rudolph, G. Exploratory landscape analysis. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 829–836.
18. Lang, R.; Engelbrecht, A. On the Robustness of Random Walks for Fitness Landscape Analysis. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 1898–1906.
19. Kerschke, P.; Preuss, M.; Wessing, S.; Trautmann, H. Low-Budget Exploratory Landscape Analysis on Multiple Peaks Models. In Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO'16, Denver, CO, USA, 20–24 July 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 229–236. [CrossRef]
20. Bossek, J. smoof: Single- and Multi-Objective Optimization Test Functions. *R J.* **2017**, *9*, 103. [CrossRef]
21. CIlib Benchmarks. Available online: https://github.com/ciren/benchmarks (accessed on 26 February 2021).
22. Mersmann, O.; Preuss, M.; Trautmann, H. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Krakov, Poland, 11–15 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 73–82.
23. Kerschke, P.; Trautmann, H. Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-package flacco. In *Applications in Statistical Computing—From Music Data Analysis to Industrial Quality Improvement*; Studies in Classification, Data Analysis, and Knowledge Organization; Bauer, N., Ickstadt, K., Lübke, K., Szepannek, G., Trautmann, H., Vichi, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; pp. 93–123; [CrossRef]
24. Lunacek, M.; Whitley, D. The dispersion metric and the CMA evolution strategy. In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Seattle, WA, USA, 8–12 July 2006, pp. 477–484.
25. Muñoz, M.A.; Kirley, M.; Halgamuge, S.K. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Trans. Evol. Comput.* **2014**, *19*, 74–87. [CrossRef]
26. Kerschke, P.; Preuss, M.; Wessing, S.; Trautmann, H. Detecting funnel structures by means of exploratory landscape analysis. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, 11–15 July 2015; pp. 265–272.
27. Renau, Q.; Dréo, J.; Doerr, C.; Doerr, B. Expressiveness and robustness of landscape features. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, 13–17 July 2019; pp. 2048–2051.
28. Renau, Q.; Doerr, C.; Dreo, J.; Doerr, B. Exploratory landscape analysis is strongly sensitive to the sampling strategy. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Leiden, The Netherlands, 5–9 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 139–153.
29. Bartz-Beielstein, T.; Doerr, C.; Bossek, J.; Chandrasekaran, S.; Eftimov, T.; Fischbach, A.; Kerschke, P.; Lopez-Ibanez, M.; Malan, K.M.; Moore, J.H.; et al. Benchmarking in Optimization: Best Practice and Open Issues. *arXiv* **2020**, arXiv:2007.03488. .
30. Maaten, L.v.d.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
31. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [CrossRef]
32. Engelbrecht, A.P. *Computational Intelligence: An Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2007.

33. Kohonen, T. Essentials of the self-organizing map. *Neural Netw.* **2013**, *37*, 52–65. [CrossRef]
34. Iivarinen, J.; Kohonen, T.; Kangas, J.; Kaski, S. Visualizing the clusters on the self-organizing map. In Proceedings of the Conference on Artificial Intelligence Research in Finland, Turku, Finland, 29–31August 1994; pp. 122–126.
35. Muñoz Acosta, M.A.; Kirley, M.; Smith-Miles, K. Analyzing randomness effects on the reliability of Landscape Analysis. *Nat. Comput.* **2020**. [CrossRef]
36. Levene, H. Contributions to probability and statistics. In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*; Stanford University Press: Redwood City, CA, USA, 1960; pp. 278–292.
37. Brown, M.B.; Forsythe, A.B. Robust tests for the equality of variances. *J. Am. Stat. Assoc.* **1974**, *69*, 364–367. [CrossRef]
38. Neuhäuser, M.; Hothorn, L.A. Parametric location-scale and scale trend tests based on Levene's transformation. *Comput. Stat. Data Anal.* **2000**, *33*, 189–200. [CrossRef]
39. Hui, W.; Gel, Y.; Gastwirth, J. lawstat: An R Package for Law, Public Policy and Biostatistics. *J. Stat. Softw. Artic.* **2008**, *28*, 1–26. [CrossRef]
40. Lim, T.S.; Loh, W.Y. A comparison of tests of equality of variances. *Comput. Stat. Data Anal.* **1996**, *22*, 287–301. [CrossRef]
41. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting Novel Associations in Large Data Sets. *Science* **2011**, *334*, 1518–1524. [CrossRef] [PubMed]
42. Albanese, D.; Filosi, M.; Visintainer, R.; Riccadonna, S.; Jurman, G.; Furlanello, C. Minerva and minepy: A C engine for the MINE suite and its R, Python and MATLAB wrappers. *Bioinformatics* **2013**, *29*, 407–408. [CrossRef]
43. Wehrens, R.; Buydens, L.M.C. Self- and Super-Organizing Maps in R: The kohonen Package. *J. Stat. Softw.* **2007**, *21*, 1–19. doi:10.18637/jss.v021.i05. [CrossRef]
44. Charrad, M.; Ghazzali, N.; Boiteau, V.; Niknafs, A. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *J. Stat. Softw.* **2014**, *61*, 1–36. [CrossRef]
45. Murtagh, F.; Legendre, P. Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion? *J. Classif.* **2014**, *31*, 274–295. [CrossRef]
46. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
47. Lang, R.D. Ela_benchmark. 2021.Available online: https://zenodo.org/record/4539080#.YDnclNwRVPY (accessed on 6 February 2021).