2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Analysis and Classification of Optimisation Benchmark Functions and Benchmark Suites

Robert W. Garden
Department of Computer Science
University of Pretoria
South Africa
Email: robgarden@gmail.com

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
South Africa
Email: engel@cs.up.ac.za

***Abstract*—New and existing optimisation algorithms are often compared by evaluating their performance on a benchmark suite. This set of functions aims to evaluate the algorithm across a range of problems and serves as a baseline measurement of how the algorithm may perform on real-world problems. It is important that the functions serve as a good representative of commonly occurring problems. In order to select functions that will make up the benchmark suite, the characteristics and relationships among the functions must be known. This paper characterises the landscapes of two commonly used benchmark suites, and uses these landscape characteristics to obtain a high level view of the current state of benchmark functions. This is done by using a self-organising feature map to cluster and analyse functions based on landscape characteristics. It is found that while there are numerous functions that cover a wide range of characteristics, there are characteristics that are under represented, or not even covered at all. Furthermore, it is discovered that common benchmark suites are composed of functions which are highly similar according to the measured characteristics.**

I. Introduction

WHEN a new optimisation algorithm or a variation of an existing algorithm is proposed, the performance of the algorithm is commonly measured on a range of known benchmark problems. These problems provides a baseline measurement of the performance of the algorithm and allows the algorithm to be compared to some other algorithm(s) in the same manner.

It is therefore important that the set of problems (the *benchmark suite*) includes a sufficient number of problems that cover a wide variety of characteristics, such that the algorithm(s) can be evaluated under diverse conditions. The challenge becomes how problems should be selected that will form an unbiased set on which to test existing and future algorithms.

Benchmark functions can be viewed as *function landscapes* on which a search algorithm aims to locate an optimal position. These landscapes contain topograhical features that can be identified and characterised. It is these features that allow categorisation of different functions without requiring an understanding of the underlying mathematical properties.

The aim of this paper is not to propose a single benchmark suite that covers all possible landscape characteristics, but rather to provide an overview and categorisation of existing functions in order to gain an understanding of the differences and similarities between them. In this way, existing benchmark suites can be scrutinised, and new benchmark suites can be composed with greater coverage of landscape characteristics.

The remainder of this paper is organised such that the reader is first introduced to benchmark functions and the ways in which their characteristics can be measured. It is then shown how exploratory data analysis (statistical methods and *self-organising feature maps*) can be used to provide a high level view of existing benchmark suites. Section II provides background on the function landscape characteristics used in this paper. Following this, existing function characterisation metrics are described. Lastly, self-organising feature maps (SOMs) are discussed since they form an integral part of the data analysis approach. Section III outlines the experimental approach that was followed. Section IV provides and discusses the results of the data analysis process. Finally, Section V concludes the paper with a summary of the findings as well as suggestions for further research.

II. Background

The main focus of this paper is an analysis of boundary constrained, continuous-valued optimization functions used to analyse the performance of optimization algorithms. This section provides an overview of measures used to characterise fitness function landscapes of boundary constrained functions, with possibly unknown optima. Most of the these metrics require that points are sampled using *random-walks* through the search space. Random-walk algorithms are beyond the scope of this paper and the interested reader is referred to Malan and Engelbrecht [1].

While reading this section, refer to Table I for an interpretation of the output of each metric.

1) Separability: Separability (additive) refers to the concept that a function can be reformulated as the sum of one or more functions. Suppose F is a function of I variables $\{x_1, x_2, \ldots, x_I\}$. Then, F is separable if there exist functions $f_1, f_2, \ldots, f_I$ that each accept one variable such that

$$F(x_1, x_2, \ldots, x_I) = f_1(x_1) + f_2(x_2) + \ldots + f_I(x_I)$$

The notion of separability can be used as a classifier of benchmark functions. Certain algorithms perform optimisation on a per-dimension basis and therefore separable functions are easier to optimise than non-separable ones for such functions.

2) First Entropic Measure: The first entropic measure (FEM), proposed by Vassilev *et al.* [2], is a measure of ruggedness in the landscape. In this case, ruggedness refers to the number and distribution of local optima. Entropy refers to the uncertainty involved in sampling using some method. Therefore, the fitness values of a number of points sampled using a random walk on a function landscape can give an entropic estimate of the ruggedness of the landscape.

In order to calculate this entropy measure, the fitness values of the landscape need to be encoded in some way. A random walk through the landscape yields a sample of points and corresponding fitness values that can be grouped as distinct three-point objects. Each three-point object is a point on the path together with its adjacent neighbours. A three-point object on the landscape is categorised as being either *neutral* (equal fitness), *smooth* (increasing/decreasing fitness), or *rugged* (both increasing and decreasing fitness). The entropic measure of ruggedness is calculated as the average ruggedness of all three-point objects.

Since fitness values are continuous, a margin of error, ϵ, is used to calculate if two fitness values are 'equal' or not. Because the fitness range of a benchmark function may be unknown, this margin of error is computed such that the end result is a measurement which yields the maximum number of rugged objects. The reader is referred to Malan and Engelbrecht [1], [3] for a more detailed explanation.

Two ruggedness measures are used in this study: $\text{FEM}_{0.01}$ (micro ruggedness) refers to ruggedness measured at step sizes of 1% of the function's domain. $\text{FEM}_{0.1}$ (macro ruggedness) refers to ruggedness measured at steps sizes of 10% of the domain.

3) Dispersion: Many functions exhibit an underlying unimodal structure in the fitness landscape when viewed over a larger domain. This structure can said to be a *funnel*: a global basin shape that consists of clustered local optima [4]. Lunacek and Whitley [5] introduced the *dispersion metric* (DM) as a way of measuring the global topology of a function landscape. In this case, the dispersion of a sample of points refers to how spread out the points are. The metric is calculated by comparing the overall dispersion of points to a subset of points which have superior fitness values. Low dispersion indicates the presence of a global single funnel (unimodal), while a high dispersion indicates the presence of multiple funnels (globally multimodal).

Malan and Engelbrecht [6] proposed an extension to the dispersion metric where points in the sample are normalised to allow comparison of functions with different domains.

4) Gradient: A random walk through a function landscape starting at position $\mathbf{x}(t)$ with T steps results in a sample of $T+1$ points, $\mathbf{x}(t), \mathbf{x}(t+1), \ldots, \mathbf{x}(T)$. The gradient between two points can then be computed as

$$g(t) = \frac{f(\mathbf{x}(t+1)) - f(\mathbf{x}(t))}{\mathbf{x}(t+1) - \mathbf{x}(t)}$$

Therefore a walk of T steps produces T gradient values [6]. These values are used to quantify both the average gradient of the landscape, i.e.

$$\text{G}_{avg} = \frac{\sum_{t=1}^{T} |g(t)|}{T}$$

as well as the standard deviation of gradient values, i.e.

$$\text{G}_{dev} = \sqrt{\frac{\sum_{t=1}^{T} (G_{avg} - |g(t)|)^2}{T}}$$

If the walk through the search space provides a good sample of neighbouring points, the G_{avg} measure will be a good estimate of the average gradient between points in the entire search space. The G_{dev} measure gives an indication of how much the gradient between two points differs from the average gradient. Therefore, a low G_{dev} indicates that the G_{avg} is a reliable estimator of the gradient. On the other hand a high G_{dev} indicates that there are extreme areas on the landscape that have sudden 'cliffs' or 'valleys' where the gradient differs from the rest of the landscape.

5) Fitness-Distance Correlation: The fitness-distance correlation (FDC) metric was proposed by Jones and Forest [7] as a way of measuring the performance of algorithms with known optima. It is a correlation of the fitness of a solution and its distance to the nearest optimum. For a landscape to be easy to search, the fitness values should become better as solutions approach an optimum.

Malan [8] proposed an extension to FDC to measure functions with unknown optima. This new measure, FDC_s, requires a sample of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and corresponding fitness values $S = \{s_1, \ldots, s_n\}$, where the fittest point is denoted as x^*. The Euclidean distance between every x_i and x^* is calculated and denoted as $Dist^* = \{d_1^*, \ldots, d_n^*\}$. The metric is then defined as:

$$\text{FDC}_s = \frac{\sum_{i=1}^{n} (s_i - \overline{s})(d_i^* - \overline{d^*})}{\sqrt{\sum_{i=1}^{n} (s_i - \overline{s})^2} \sqrt{\sum_{i=1}^{n} (d_i^* - \overline{d^*})^2}}$$

where $\overline{s}$ is the mean of S, and $\overline{d^*}$ is the mean of $Dist^*$.

6) Information Landscape: Borenstein and Poli [9] introduced the information landscape metric which uses a matrix of all pairwise comparisons between the fitness values of points in a random sample. Consider X points in a sample. The information matrix has $|X| \times |X|$ entries $m_{ij} = t(\mathbf{x}_i, \mathbf{x}_j)$, where

$$t(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 & \text{if } f(\mathbf{x}_i) < f(\mathbf{x}_j) \\ 0.5 & \text{if } f(\mathbf{x}_i) = f(\mathbf{x}_j) \\ 0 & \text{if } f(\mathbf{x}_i) > f(\mathbf{x}_j) \end{cases}$$

for minimisation problems.

Since the matrix contains a full pairwise comparison of points, duplicate values occur. Therefore, only a subset of the matrix is needed to describe the information landscape. Diagonal entries are always 0.5 and can be ignored. Similarly, the matrix is symmetrical about the diagonal, and the lower half can also be ignored. Lastly, the row and column of the optimal solution can be ignored since it is known that this is always one.

The information matrix can then be reduced to a vector that contains only the relevant entries in the matrix, i.e. $V = (v_1, v_2, \ldots, v_n)$. A number of calculations can be performed on this vector. Borenstein and Poli proposed that the distance between this vector and the information landscape vector of an 'optimal' landscape, such as Sphere, be measured.

In order to compare two information landscapes, it is necessary that both landscapes are aligned with respect to their global optimum, which may be unknown. Malan [8] proposed an extension (IL$_{ns}$) to this method where the Sphere function is shifted such that its global optimum is the same as the best solution in the sample. The distance between the two vectors then gives an indication of the searchability of the landscape, with low values indicating high (easy) searchability, and high values indicating low (hard) searchability.

7) Fitness Cloud Index: Veral *et al.* [10] introduced the concept of a fitness cloud as a way of visualising the evolvability of an evolutionary search algorithm. The fitness cloud is a plot of parent and offspring points, showing the relationship between them. The offspring in this case are created via some evolutionary operator such as crossover or as a result of an iteration of a particle swarm. Malan [8] proposed a fitness cloud index metric that makes use of cognitive-only and social-only PSO update equations. These two metrics give a measure of the evolvability of a function landscape with respect to local and global search heuristics.

To do this, an initial sample of parent points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is generated. Two PSO particle update iterations are performed in order to generate offspring points. Two iterations are needed as particles have an initial velocity of zero. The resulting offspring $X' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_n\}$ are compared to the parent population X. The final metric then is calculated as the ratio of offspring that have an improved fitness value compared to their respective parent.

TABLE I
FUNCTION METRIC RANGES AND THEIR INTERPRETATIONS

Metric	**Range and Interpretation**
Separability	{0, 1}: where 0 indicates a non-separable function and 1 indicates a separable function
FEM	[0, 1]: where 0 indicates a flat landscape and 1 indicates maximal ruggedness
DM	$[-disp_D, \sqrt{D} - disp_D]$: where $disp_D$ is the dispersion of a large uniform random sample of a D-dimensional space normalised to [0, 1] in all dimensions. A positive value for DM indicates the presence of multiple funnels.
G_{avg}	A positive real number, where a higher value indicates higher average gradients.
G_{dev}	A positive real number, where a higher value indicates higher deviations from average gradients.
FDC$_s$	$[-1, 1]$: For a minimisation problem, 1 indicates the highest measure of searchability (perfect correlation between fitness values and distance to the fittest solution).
IL$_{ns}$	[0, 1]: A value of 0 indicates maximum searchability (no difference from the reference landscape vector vr).
FCI$_{\text{cog}}$ FCI$_{\text{soc}}$	[0, 1]: indicating the proportion of fitness improving solutions after two PSO updates.

A. Self-Organising Feature Maps

This section describes self-organising feature maps in order for the reader to understand how they were used in the experimental procedure.

Kohonen developed the self-organising feature map (SOM) [11] using the human cerebral-cortex as motivation. A SOM is a method of scaling an I-dimensional input space to a discrete output space, usually in the form of a two-dimensional grid. The SOM can therefore be considered as a compression algorithm which attempts to approximate the distribution of the input space while maintaining its topological characteristics. That is, if two vectors are similar in input space, they will be similar in the map representation.

The first step in the SOM process is to decide on the number and structure of elements (neurons) in the grid. The grid is usually square but can be any rectangular shape. The number of neurons is ideally the same as the number of unique training patterns. Each neuron on the map is assigned an I dimensional weight vector $\mathbf{w}_k$, where I is the number of attributes in the dataset. There are many weight initialisation schemes and the reader is referred to Engelbrecht [12] for a more detailed explanation.

Once weights have been initialised, the network of neurons is trained competitively. For each neuron and training pattern $\mathbf{z}_p$, the weight vector is updated as:

$$\mathbf{w}_{kj}(t+1) = \mathbf{w}_{kj}(t) + h_{mn,kj}(t)[\mathbf{z}_p - \mathbf{w}_{kj}(t)]$$

where mn is the index of the winning neuron; the neuron that best matches the current input pattern. The winning neuron is computed as the shortest Euclidean distance between the weight vector and the input pattern.

The function $h_{mn,kj}(t)$ is the neighbourhood function. It is used to define an area around the winning neuron that will be affected by weight updates. Neurons outside of this neighbourhood will have their weights updated negligibly, or not at all.

The training process continues until a 'good' map has been constructed. The quantisation error is generally used as an indicator of the map's accuracy. The end-result is a two-dimensional structure that represents the entire dataset of I-dimensional data patterns while maintaining the topographical relationship between them.

In the context of this work, the SOM is trained using data gathered from the various function metrics. Analysis of the relationships between function characterstics is performed, and additionally, by labelling neurons using functions from benchmark suites, it is possible to observe the distribution of these functions across all charactersitics.

III. Experimental Approach

This section provides an overview of the experimental process followed. Section III-A summarizes the functions used, the function metrics are listed in Section III-B, and the experimental procedure is given in Section III-C.

A. Benchmark Functions Dataset

1) Functions: Seventy-six boundary constrained benchmark functions are included in the function dataset. These include a wide variety of separable and non-separable functions and functions of other characteristics. The majority of functions are defined and used in 10 dimensions, however there are functions that are only defined for low dimensions (1, 2, 5) and these are used accordingly.

The full list of unconstrained benchmark functions can be found in the Computational Intelligence Library (CIlib) [13].

2) Suites: In addition to the above functions, two commonly used benchmark suites are included in the function dataset. The CEC 2005 [14] benchmark suite contains 25 shifted and rotated functions. All functions in this suite are tested in 10 dimensions. The BBOB 2009 [15] benchmark suite contains 24 asymmetric and conditioned functions which are again used in 10 dimensions.

This results in a total of 125 benchmark functions in the function dataset.

B. Function Metrics

All function metrics described in Section II were used in the experimental procedure. Each metric made use of 1000 sampled points, regardless of the domain and dimension of the function. The number of sampled points is roughly 2^D, which was found in [16] to be the number of points which yielded a similar distribution to uniform sampling. The way in which the points were sampled is listed below in Table II.

TABLE II
FUNCTION METRICS SAMPLING METHODS

Metric	Sampling Method
FEM	Progress Random Walk
DM	Uniform
G_{avg}, G_{dev}	Manhattan Random Walk
FDC_s	Uniform
IL_{ns}	Uniform
FCI_{cog}, FCI_{soc}	Uniform

C. Experimental Procedure

For each benchmark function and function metric combination, the metric was calculated 30 times, due to the stochastic nature of the sampling methods. Mean and standard deviations were calculated over the 30 runs. The final mean values were then used to calculate distributions as well as to serve as input to the SOM.

The SOM was trained using 125 neurons, the same as the number of functions in the dataset. Each input vector corresponds to one function, where the input vector consists of the nine values described in Table I. Data values have not been normalised.

IV. Experimental Results

The results and findings are discussed in this section. This is done in two parts: Section IV-A considers an analysis with respect to frequency distributions of the different function metrics, while Section IV-B performs an exploratory analysis using a SOM.

A. Analysis Based on Metric Distributions

Fig. 1 illustrates the distribution of the recorded functions metrics.

The dataset is fairly evenly distributed with regards to separability. Separable functions account for 40% of the dataset and non-separable functions account for 60%.

With regards to landscape micro ruggedness, $\text{FEM}_{0.01}$, functions are roughly distributed around 0.5, indicating that many function landscapes contain both flat and highly rugged areas. However, more functions are classified as being highly rugged (≥ 0.7) than being flat (< 0.3). This indicates that more functions which contain flat or neutral landscapes in small areas (1% of the landscape) are needed.

Macro ruggedness, $\text{FEM}_{0.1}$, shows a similar trend. However, the middle of the distribution is at 0.6. Indeed there are more functions with high ruggedness and there are no functions with very large (10% of the landscape) neutral areas.

The dispersion metric, DM, highlights that there are extremely few multi-funnelled landscapes. The dataset has only 11 (8.8%) functions characterised as being multi-funnelled. This means that when selecting a function to be used in a benchmark suite, one does not have access to a large set of multi-funnelled functions.

Both the average gradient, G_{avg}, and standard deviation of gradients, G_{dev}, are low. There are, however, several outliers with very high gradients or with gradients that vary greatly. This indicates that functions with medium-high gradients are needed. The functions with high gradients and high gradient variations can be useful when constructing benchmark suites to test gradient-based optimisation methods as these may pose significant difficulties to these algorithms.

Fitness-distance correlation, FDC, is slightly skewed to the left, with a concentration (mean) around 0.37, indicating that there are more functions with low searchability. Overall, fitness-distance correlation is the metric that is the most evenly distributed, meaning that across the entire range of used functions there exist functions that range from low searchability to high searchability.

The information landscape metric, IL_{ns}, is skewed to the right, with a concentration around 0.35. No functions have a value greater than 0.6 and few functions have a value less than 0.1. This indicates that there are few (or no) functions that are as searchable as the Sphere function and that there are no functions that differ substantially from the Sphere function, a limitation of benchmark suites. It is important the functions that have a maximum negative searchability, i.e. an IL_{ns} value of one, are defined and included in benchmark suites.

The cognitive fitness-cloud index, FCI_{cog}, exhibited expected results. Only one function performed poorly with respect to the fitness values of offspring and parents. That is, only one function had offspring with poorer fitness than their respective parents.

The social fitness-cloud index, $\mathrm{FCI}_{\mathrm{cog}}$, behaved similarly to the cognitive model, except that there are more functions with greater proportion of worse offspring. This distribution is still greatly skewed to the right, meaning that there are many functions that, when only globally searched, have 90% searchability.

B. Exploratory Analysis using Self-Organising Feature Maps

1) Observations: Fig. 2 and 3 provide component maps for the different metrics, obtained from a SOM trained on the landscape characteristics of the 125 functions. The SOM gives a visual overview of the entire set of benchmark functions. From these components, it is possible to see how the different landscape characteristics relate to one another. There are many observations that can be made, of which the most prominent are described below.

Considering ruggedness (FEM), functions with micro ruggedness generally have similar macro ruggedness properties. However, there are functions with high micro ruggedness, but low macro ruggedness. When comparing the ruggedness component planes to separability, it is observed that high ruggedness corresponds to non-separability, except for a distinct cluster of high micro ruggedness functions who are all separable. This is a limitation of benchmark suites having functions with similar micro and macro ruggedness characteristics, as well as lacking separable functions with high ruggedness.

Dispersion, DM, has low values across the entire map except for a small cluster of functions at the bottom of the map which have high dispersion. What is interesting is that these functions are a mixture of separable and non-separable functions and these are functions which have a mixture of micro ruggedness but all have low macro ruggedness. This is another limitation of benchmark suites lacking multi-funnelled functions with high macro ruggedness. Also, the functions with high dispersion have low gradients, low fitness-distance correlations, and high information landscape values (bad searchability). It would be good to be able to find functions that have positive dispersion (multi-funnels) that are also varied with regards to the other characteristics such as ruggedness and gradient.

Gradient values, G_{avg} and G_{dev}, are low in most parts of the map except in the top right cluster. In this cluster we see that functions have both high average gradients as well as high variations in gradient. However, there is a group of functions which have higher deviations in gradient while still maintaining a low average gradient. It is important to have functions which may have sudden gradients but with underlying flat landscapes. Another observation worth mentioning is that all functions with high gradients are non-separable. However, the opposite is not true, there are non-separable functions with low gradients.

Fitness-distance correlation (FDC) and information landscape maps are almost exact inverses. This is to be expected as the information landscape describes negative searchability with respect to the Sphere function whereas FDC describes positive searchability (and the best FDC landscape is Sphere). Note that areas with high FDC values (good searchability) correspond with areas that have low dispersion (single funnelled). The converse is also true, areas with worst searchability are multi-funnelled landscapes. It is therefore important to find more multi-funnelled landscapes and thus increase the number of functions with poor FDC values.

The cognitive fitness-cloud index, $\mathrm{FCI}_{\mathrm{cog}}$, has high values across the map except in the top right region. It is exactly this region that has high gradients. This makes intuitive sense because the cognitive-only PSO is a hill climber and thus may overshoot optima on functions with high gradients and narrow basins. The social fitness-cloud index, $\mathrm{FCI}_{\mathrm{soc}}$, has higher values in those regions with high gradients. An interesting observation is that the social FCI has one very low region which corresponds to the exact same area of macro ruggedness. This is again intuitive as areas with very low ruggedness may not yield enough information to guide a global search process.

2) CEC 2005 and BBOB 2009: Fig. 2 and 3 respectively illustrate how the CEC 2005 and BBOB 2009 benchmark functions map to the different clusters shown on the component planes.

With reference to Fig. 2, the lower-left region of the map contains no functions from the CEC 2005 benchmark suite. This means that for the CEC 2005 benchmark suite there are

- very few separable functions, which may be disadvantageous to certain algorithms;
- no functions with low micro and macro ruggedness, i.e. neutral functions;
- many functions with high micro and macro ruggedness;
- no multi-funnelled functions and very few functions with low dispersion;
- very few functions with low gradients (the majority are medium-high gradients);
- few functions with low or high fitness-distance correlation (most lie in the medium range);
- most functions have medium cognitive (hill-climbing) properties;
- no functions that have low hill-climbing properties;
- no functions that have low global-search properties (although such functions are difficult to find); and
- 10 functions with very similar characteristics (they are contained within a single cluster).

Fig. 3 shows that the BBOB 2009 functions are more spread out over the map than the CEC 2005 functions. However, there are still areas on the map which are sparsely populated. This means that for the BBOB 2009 benchmark suite there are

- very few separable functions;
- very few (only one) functions with very low (neutral) micro and macro ruggedness;
- very few functions with high micro and macro ruggedness;

- no multi-funnelled functions; however there are more functions with low dispersions when compared to the CEC 2005 suite;
- more functions with low gradients than with high gradients;
- very few (only one) functions with negative FDC searchability; and
- most functions that have both high hill-climbing and high global-search properties.

Based on the points above, the CEC 2005 and BBOB 2009 benchmark suites should include more functions with low ruggedness in order to provide neutral landscapes. Additionally, both benchmark suites should include multi-funnelled functions in order to have landscapes with underlying multimodal structures. The CEC 2005 suite requires functions with low gradients whereas the BBOB 2009 suites requires additional functions with medium and high gradients. Finally, both benchmark suites are lacking functions with low hill-climbing properties and these should be included in order to evaluate algorithms that rely on local search methods.

V. Conclusion

Benchmark functions are commonly used to compare the performance of optimisation algorithms. These benchmark functions can be considered to be a function landscape on which the entities of an optimisation algorithm search to find optimal solutions. In this regard, the topographical features of a function landscape play a crucial role in determining the performance of new and existing algorithms.

This paper analysed a wide range of individual benchmark functions and functions taken from two common benchmark suites. The topographical features of the function landscapes have been characterised according various function metrics. After obtaining characteristics for all functions in the dataset, the functions were analysed by observing the distributions of values of the individual metrics. Additionally, an unsupervised learning approach, namely a self-organising feature map, was used to further identify areas of characteristics and the relationships between these characteristics.

The CEC 2005 and BBOB 2009 benchmark suites were labelled on to the SOM in order to identify landscape characteristics that are under represented by these benchmark suites. It has been found that both these suites are lacking functions in certain areas of the characteristic space. In particular, the CEC 2005 benchmark suite contains 10 functions that have similar characteristics (they are contained within a single cluster). By using this information it is possible to see how benchmark suites have been composed and to identify their limitations.

An important finding of this study is that, of the used functions, there are currently characteristics for which benchmark suites lack representative functions. For example, there are very few functions that are multi-funnelled. It is important to then find functions with these characteristics such that a diverse function portfolio can be maintained and utilised.

This study is by no means an exhaustive survey of all current benchmark functions. Possible further research includes

- using metrics to characterise constrained functions;
- observing how the relationships between functions change at varying dimensions;
- performing additional statistical and clustering methods to find further relationships;
- testing the hypothesis that if an algorithm performs well on a function, that it will perform similarly on functions with similar characteristics (in the same cluster);
- using function characteristics to compose new benchmark suites that aim to fully cover the range of available function types.

References

[1] K. M. Malan and A. P. Engelbrecht, "Quantifying Ruggedness of Continuous Landscapes using Entropy," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009*, 2009, pp. 1440–1447.

[2] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Smoothness, Ruggedness and Neutrality of Fitness Landscapes: from Theory to Application," in *Advances in Evolutionary Computing: Theory and applications*. Springer-Verlag New York, Inc., 2003, pp. 3–44.

[3] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Inf. Sci.*, vol. 241, pp. 148–163, 2013.

[4] A. M. Sutton, D. Whitley, M. Lunacek, and A. Howe, "PSO and multi-funnel landscapes: how cooperation might limit exploration," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2006, pp. 75–82.

[5] M. Lunacek and D. Whitley, "The dispersion metric and the CMA evolution strategy," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2006, pp. 477–484.

[6] K. M. Malan and A. P. Engelbrecht, "Ruggedness, funnels and gradients in fitness landscapes and the effect on pso performance," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013*, 2013, p. To appear.

[7] T. Jones and S. Forrest, "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms," in *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1995, pp. 184–192.

[8] K. M. Malan, "Characterising continuous optimisation problems for particle swarm optimisation performance prediction," Ph.D. dissertation, University of Pretoria, 2014.

[9] Y. Borenstein and R. Poli, "Information landscapes," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. New York, NY, USA: ACM Press, 2005, pp. 1515–1522.

[10] S. Verel, P. Collard, and M. Clergue, "Where are bottlenecks in NK fitness landscapes?" in *The 2003 Congress on Evolutionary Computation, CEC '03*, vol. 1, 2003, pp. 273–280.

[11] T. Kohonen, *Self-organizing maps*. Springer, 2001, vol. 30.

[12] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Ltd, 2005.

[13] CIRG, "Continuous unconstrained functions," https://github.com/cilib/cilib/tree/next/library/src/main/java/net/sourceforge/cilib/functions/continuous/unconstrained, 2013, [Online; accessed 28-September-2013].

[14] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," *KanGAL Report*, vol. 2005005, 2005.

[15] N. Hansen, S. Finck, R. Ros, A. Auger *et al.*, "Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions," 2009.

[16] K. M. Malan and A. P. Engelbrecht, "A progressive random walk algorithm for sampling continuous fitness landscapes," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014*, 2014, p. To appear.

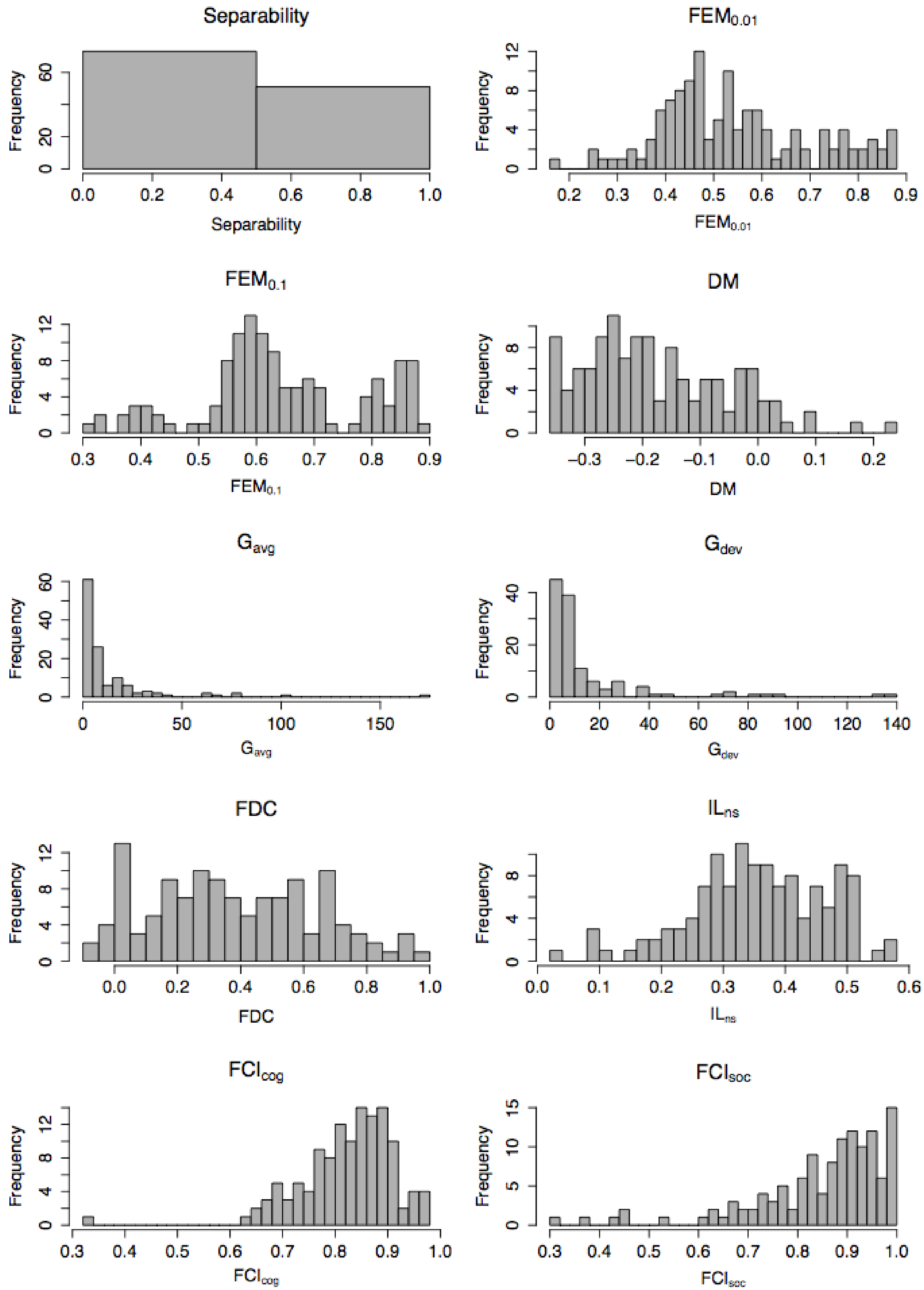

Fig. 1. Distributions of all benchmark functions according to specific function metrics

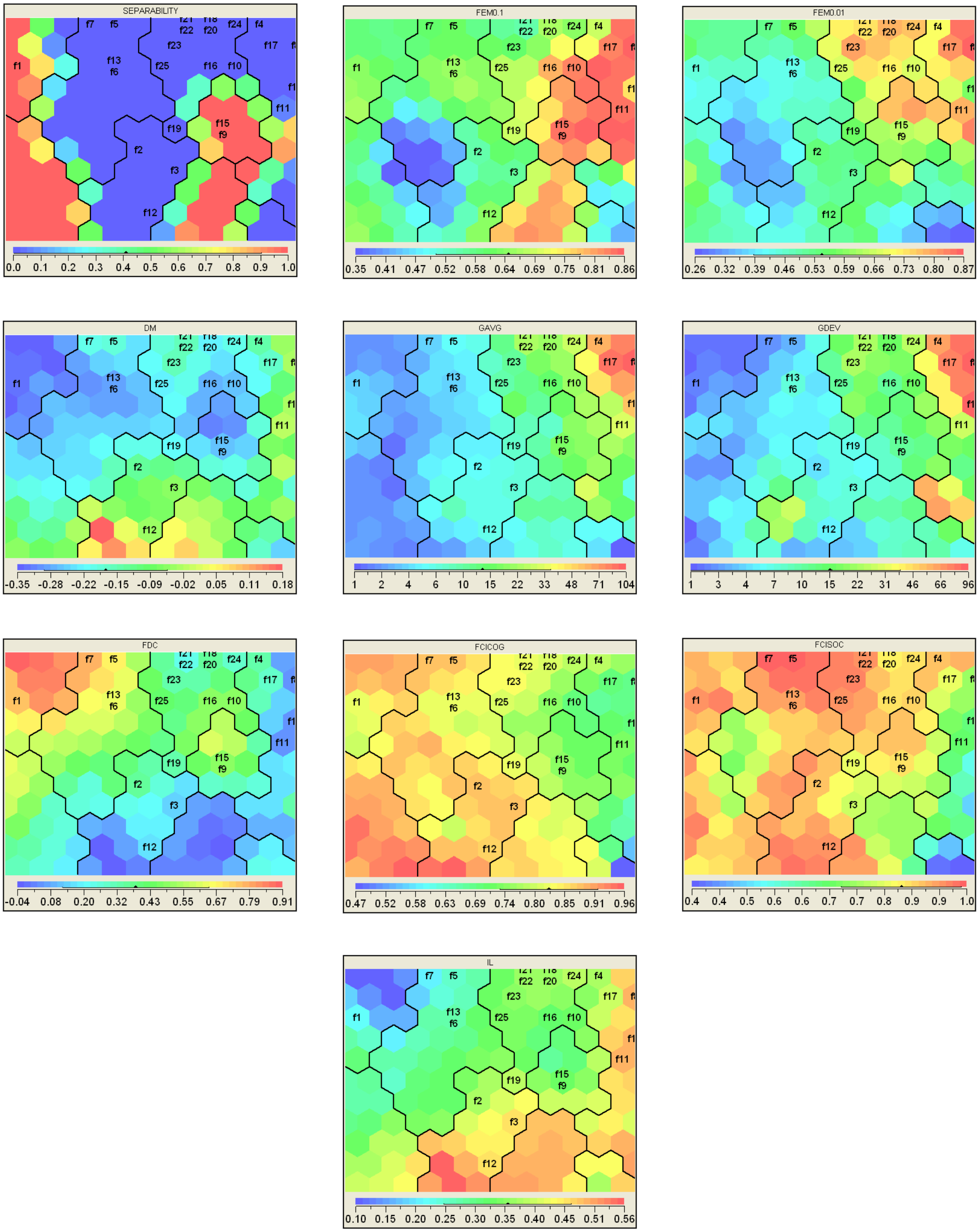

Fig. 2. SOM component planes with CEC 2005 functions labelled

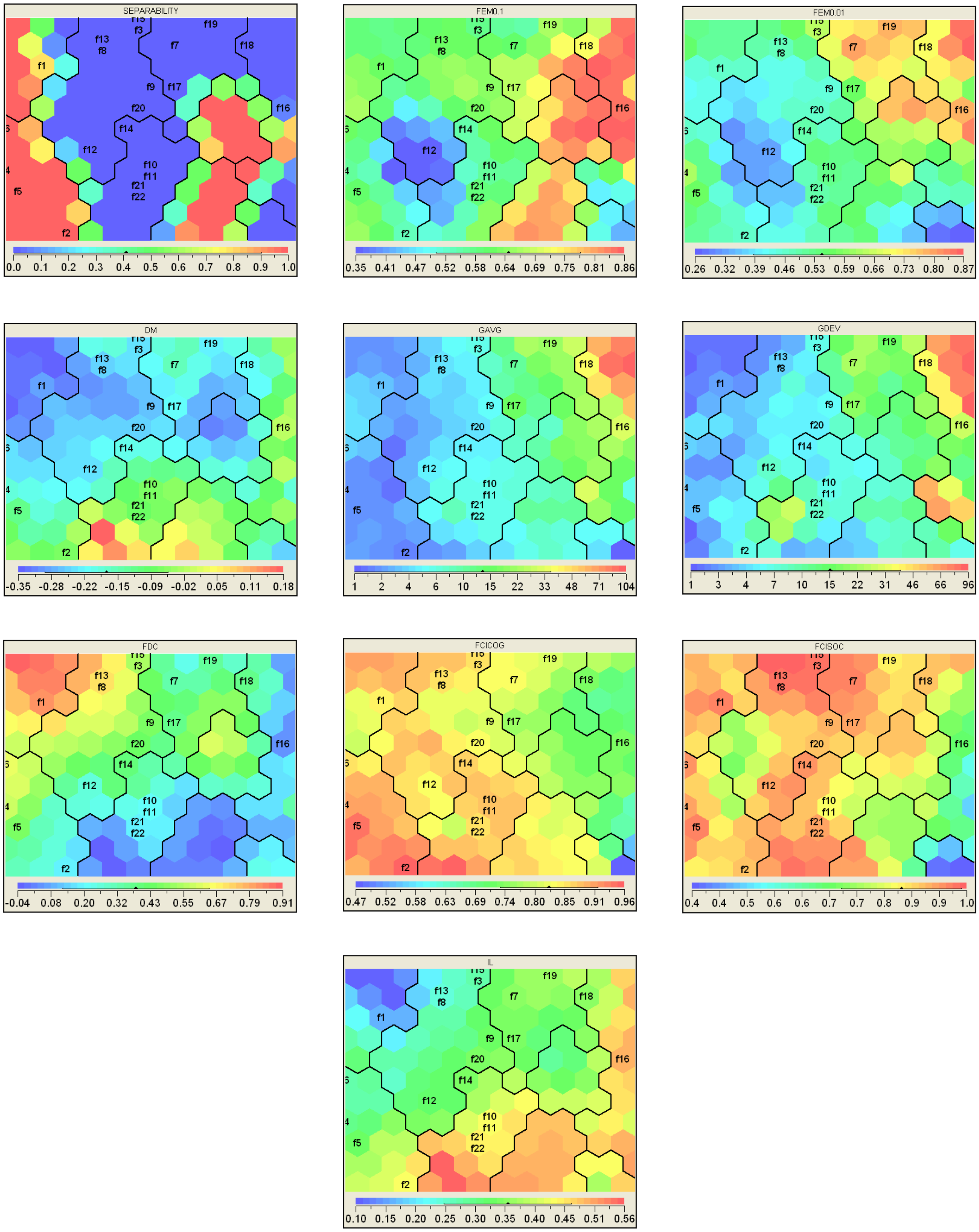

Fig. 3. SOM component planes with BBOB 2009 functions labelled