Synthetic vs. Real-World Continuous Landscapes: A Local Optima Networks View

Marco A. Contreras-Cruz[1], Gabriela Ochoa[2(✉)], and Juan P. Ramirez-Paredes[1]

[1] Electronics Engineering Department, University of Guanajuato, Salamanca, Mexico
{ma.contrerascruz,jpi.ramirez}@ugto.mx
[2] Computing Science and Mathematics, University of Stirling, Stirling, UK
gabriela.ochoa@stir.ac.uk

Abstract. Local optima networks (LONs) are a useful tool to analyse and visualise the global structure of fitness landscapes. The main goal of our study is to use LONs to contrast the global structure of synthetic benchmark functions against those of real-world continuous optimisation problems of similar dimensions. We selected two real-world problems, namely, an engineering design problem and a machine learning problem. Our results indicate striking differences in the global structure of synthetic vs real-world problems. The real-world problems studied were easier to solve than the synthetic ones, and our analysis reveals why; they have easier to traverse global structures with fewer nodes and edges, no sub-optimal funnels, higher neutrality and multiple global optima with shorter trajectories towards them.

Keywords: Local optima networks · Funnel structures · Fitness landscapes · Real-world optimisation problems

1 Introduction

The structure of fitness landscapes is known to relate to the performance of optimisation algorithms. Several tools and metrics have been proposed to characterise fitness landscapes [12], however, few of them deal with the landscapes' global structure. Local optima networks (LONs) help to fill this gap, by providing information about the number, distribution, and connectivity patterns of local optima [14]. LONs were inspired by work in theoretical/computational chemistry, where the structure of energy landscapes (derived from the atomic interactions in clusters and molecules) is modelled as a graph [5]. The central idea of LONs is to compress the search space into a graph where nodes are local optima and edges are possible transitions between optima with a given search operator. Several topological network features can be extracted from LONs [14]; recent work have also studied the landscapes' funnel structure [15,16]. The notion of funnels comes from the study of energy landscapes; according to Doye *et al.*

B. Filipič et al. (Eds.): BIOMA 2020, LNCS 12438, pp. 3–16, 2020.
https://doi.org/10.1007/978-3-030-63710-1_1

[6] a funnel "is a region of configuration space that can be described in terms of a set of downhill pathways that converge on a single low-energy structure or a set of closely related low-energy structures." Funnels are important global structure features, a single global funnel will facilitate optimisation, but the presence of sub-optimal funnels will hinder it.

Network-based models of fitness landscapes have only recently been applied to continuous optimisation problems outside the realm of molecular energy minimisation, examples are [1,2,17]. Our work extends these results in order to contrast the global (funnel) structure of synthetic vs. real-world continuous optimisation problems, we also develop some aspects of the methodology. Our main contributions are to:

- Apply the LON model to real-world continuous optimisation problems (including engineering design and machine learning) and contrast their structure against the structure of synthetic benchmark functions.
- Contrast two initialisation methods, uniform and Latin hypercube, when sampling the fitness landscapes to extract the LON models.
- Explore the effect of increasing the problem dimension on the global (funnel) structure of the underlying landscapes.

2 Definitions

We start by formalising the notions of fitness landscapes and local optimum, before defining the local optima network models considered in our study. We use two models to discover the global structure of continuous optimisation problems: the Monotonic LON model, and the Compressed Monotonic LON model, introduced in [16].

Fitness Landscape. In the context of continuous optimisation, a fitness landscape is a triplet $(\mathbf{X}, N, f)$ where $\mathbf{X} \in \mathbb{R}^n$ is the set of all real-valued solutions of n dimensions, *i.e.*, the search space; N is a function that assigns to every solution $\mathbf{x} \in \mathbf{X}$ a set of neighbors $N(\mathbf{x})$; and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the fitness function. A potential solution $\mathbf{x}$ is denoted as vector $\mathbf{x} = (x_1., x_2, \ldots, x_n)$, and the neighbourhood is based on hypercubes. Formally, the neighbourhood of a candidate solution $\mathbf{x}_k$ is defined as, $\mathbf{x}_j \in N(\mathbf{x}_k) \leftrightarrow |x_{ki} - x_{ji}| < s_i$, $i = \{1, \ldots, n\}$ where $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ is a vector that represents the size of the neighbourhood in all dimensions.

Local Optimum. A solution $\mathbf{x}^* \in \mathbf{X}$ such that $\forall \mathbf{x} \in N(\mathbf{x}^*)$, $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Monotonic LON Model. The monotonic LON is the directed graph $MLON = (L, ME)$ where the nodes L are local optima and the edges $ME \subset E$ are monotonic perturbation edges.

Monotonic Perturbation Edges. There is a monotonic perturbation edge from a local optimum $\mathbf{l}_1$ to a local optimum $\mathbf{l}_2$, if $\mathbf{l}_2$ can be obtained from a random perturbation of $\mathbf{l}_1$ followed by a local minimisation process, and $f(\mathbf{l}_2) \leq f(\mathbf{l}_1)$. The edge is called monotonic because the transition between two local optima is non-deteriorating. In this model, the edges are weighted with the number of times a transition between two local optima occurred.

Compressed Monotonic LON Model. This is a coarser model that compresses connected local optima at the same fitness into single nodes. The purpose is to facilitate modelling landscapes with neutrality.

Compressed Monotonic LON. It is the directed graph $CMLON = (CL, CE)$ where the nodes are compressed local optima CL, and the edges $CE \subset ME$ are aggregated from the monotonic edge set ME by summing up the edge weights.

Compressed Local Optimum. A compressed local optimum is a single node that represents a set of connected nodes in the MLON model with the same fitness value.

Monotonic Sequence. A monotonic sequence is a path of connected local optima where their fitness values are always decreasing. Every monotonic sequence has a natural end, which represent a funnel bottom, also called sink in graph theory.

Funnel. We can characterise funnels in the CMLON as all the monotonic sequences ending at the same compressed local optimum (funnel bottom or sink).

3 Methodology

Our methodology for sampling and constructing the networks is based on the *basin-hopping* (BH) algorithm, proposed in the context of computational chemistry [18]. BH is an iterative algorithm, where each iteration is composed of a random perturbation of a candidate solution, followed by a local minimisation process and an acceptance test. In this study, we adopted a variant of the BH algorithm called monotonic basin-hopping (MBH) proposed in [10] (see in Algorithm 1), where the acceptance criterion considers only improving solutions.

To construct the network models a number of runs are conducted. Each run on a given problem instance produces a search trajectory, which is recorded as a set of nodes (local minima) and edges (consecutive transitions), and stored in the sets L, and ME, respectively. Note that different runs can in principle traverse the same nodes and edges, even if they start from different initialisation points. The MLON network is constructed in a post-processing stage where the trajectories generated by a fixed number of runs (100 in our implementation) are aggregated to contain only unique nodes and edges.

We compared two techniques to generate the initial candidate solutions ($\mathbf{x}_0$ in Algorithm 1) during the sampling process: uniform and Latin hypercube sampling. The uniform sampling generates initial solutions following a uniform distribution within the problem bounds, whereas the Latin hypercube sampling (LHS) technique partitions the search space into equally probable bins and positions the samples at each axis-aligned hyperplane.

Algorithm 1. Monotonic basin-hopping sampling.

Require: search space $\mathbf{X} \in \mathbb{R}^n$, fitness function $f(\mathbf{X})$, step size p
Initial random solution $\mathbf{x}_0 \in \mathbf{X}$
$\mathbf{x} \leftarrow \text{LocalMinimization}(\mathbf{x}_0)$
$L \leftarrow \{\mathbf{x}\}$
repeat
  $\mathbf{x}' \leftarrow \text{Perturbation}(\mathbf{x}, p)$
  $\mathbf{y} \leftarrow \text{LocalMinimization}(\mathbf{x}')$
  if $f(\mathbf{y}) \leq f(\mathbf{x})$ **then**
    $L \leftarrow L \cup \{\mathbf{y}\}$
    $ME \leftarrow ME \cup (\mathbf{x}, \mathbf{y})$
    $\mathbf{x} \leftarrow \mathbf{y}$
  end if
until Stopping criterion is not reached
return L, ME

As the local minimiser we used the Limited-memory Broyden-Fletc.her-Goldfarb-Shanno algorithm (L-BFGS) [13], which is an extension of BFGS, a well-known quasi-Newton algorithm. L-BFGS is scalable to higher dimensions because it avoids storing a fully dense approximation of the Hessian matrix.

The step size p has to be appropriately selected for each problem. If p is too small, the candidate solution can be easily trapped in a low quality local optimum and unable to escape from its basin of attraction; whereas if p is too large, the candidate solution can move drastically to regions of worse quality, degenerating into random search and losing the progress attained, especially if the search has already reached relatively low fitness solutions. In order to select p, we followed the suggestion in [10], which indicates to vary p until roughly half of the steps attempted escape the starting basin of attraction. In continuous space, there is a precision issue to decide if two solutions correspond to the same local optimum. In this work, we used a position threshold ϵ that depends on the optimisation problem. Two solutions represent the same local optimum if the absolute difference between each of their components is less than ϵ.

Once the models are constructed, we can extract different metrics to bring insight into the search difficulty and the global structure of the studied landscapes. Table 1 describes the metrics used in this study. The network metrics were gathered from the CMLON model. Note that when there is little neutrality in the search space, the number of nodes in the CMLON model is close to the number of local optima in the underlying landscape, whereas when neutrality is

high, the number of nodes in the CMLON is smaller than the number of underlying landscape optima. The ratio between the number of compressed optima over the total number of local optima is captured by the *neutral* metric described in Table 1.

Table 1. Performance and network metrics.

Performance metrics	
Success	Proportion of runs that reached the global minimum.
Deviation	Mean deviation from the global minimum.
Network metrics	
Nodes	Number of nodes in the CMLON (compressed local optima).
Funnels	Number of sinks (CMLON nodes without outgoing edges).
Neutral	Proportion of CMLON nodes to the number of local optima.
Strength	Normalized incoming strength of the globally optimal sinks.

4 Experimental Setup

4.1 Synthetic Functions

We consider three classical test functions, modified Ackley (Ackley 4), Griewank, and Schwefel 2.26, which are all differentiable, separable, scalable, and multimodal, but are known to differ in their global structures. Figure 1 shows 3D visualisations of the functions with two variables. The Ackley 4 function [7] has two global minima close to the origin, presents a single pronounced funnel toward both minima, and evaluates within the range $[-35, 35]$. The Griewank function has a single funnel structure, many local optima almost at the same fitness level than the single global optimum (located at the origin), and evaluates in the range $[-600, 600]$. The Schwefel 2.26 function has multiple funnels, a single global optimum located far from the origin, at $f(\mathbf{x}^*) = f(420.9687, \ldots, 420.9687)$, and evaluates in the range $[-500, 500]$. For each function, we considered 3 instances with $n = \{3, 5, 8\}$.

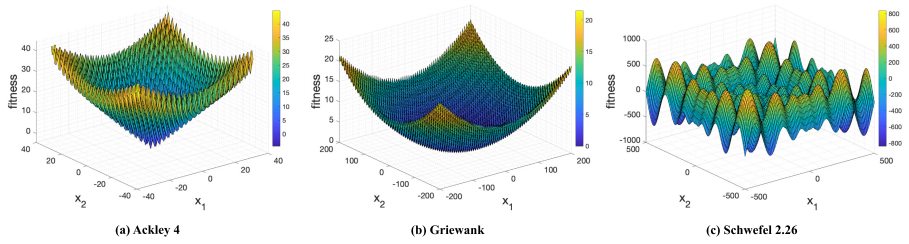

Fig. 1. 3D visualization of the selected synthetic benchmark functions with two variables.

4.2 Real-World Functions

Engineering Design Problem. We selected the radar pulse modulation problem named "spread spectrum radar polly phase code design problem", as described and implemented in the CEC2011 "Competition on Testing Evolutionary Algorithms on Real-World Optimisation Problems" [4]. This problem can be scaled to n dimensions and belongs to the class of continuous min-max global optimisation problems. In both problems, the global optima are known. We selected 3 instances with $n = \{3, 5, 8\}$, to allow a closer comparison with the synthetic functions. In what follows, we call this the *Radar* function.

Machine Learning Problem. We used the sum of squares clustering problem from the "Machine Learning and Data Analysis" problem set [9]. The problem is to determine the position of k cluster centres in order to minimise the sum of squared distances between each data point (in a dataset) and its nearest cluster center. Let us denote $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ as the set of centers where $\mathbf{c}_j \in \mathbb{R}^q$; and $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ as the dataset, with $\mathbf{d}_i \in \mathbb{R}^q$. A candidate solution is represented as a vector $\mathbf{x} \in \mathbb{R}^{qk}$ that concatenates the centers coordinates. Several test instances can be generated, changing the value of k and the dataset. We selected the *Ruspini* dataset that contains 75 observations in a 2-dimensional space ($\mathbf{d}_i \in \mathbb{R}^2$). Since the observations are in a 2-dimensional space, and the problem representation concatenates the cluster centres, instances can only be generated with an even dimension. In order to closely compare with the other benchmark functions, we selected 3 instances with $n = \{4, 6, 8\}$ (corresponding respectively to $k = \{2, 3, 4\}$). In what follows, we call this the *Clustering* function.

4.3 Parameter Settings and Experiments

To extract the networks, 100 runs of the monotonic basin-hoping sampling algorithm (Algorithm 1) were conducted on each function with a stopping criterion based on a predefined number of cycles. The number of cycles was determined experimentally for each problem, by observing the convergence behavior of 30 runs. Table 2 summarises the parameters used for each function.

Table 2. Parameter settings for each function.

Function	Dimensions	Step size	Cycles	ϵ
Ackley 4	3, 5, 8	1.63	300	10^{-2}
Griewank	3, 5, 8	3.60	200	10^{-2}
Schwefel 2.26	3, 5, 8	151.00	4000	10^{-2}
Radar	3, 5, 8	1.27	100	10^{-1}
Clustering	4, 6, 8	48.64	100	1.0

All the experiments were conducted in an Intel Core i7 computer, with a processor running at 3.5 GHz and 16 GB of RAM. We used the basin-hopping implementation provided in the Python package Scipy [8]. For constructing models, computing the metrics and visualising the networks we used the igraph package [3] with the R statistical language. For visualising the networks, we used force-directed graph layout algorithms as provided in igraph.

5 Results

5.1 Synthetic Functions

Performance and Network Metrics. Figure 2 reports the performance and network metrics described in Table 1 for the synthetic functions with $n = \{3, 5, 8\}$. Results are shown for both initialisation methods, which are indicated using the subscript *uni* for the uniform initialisation and *lhs* for the hypercube sampling.

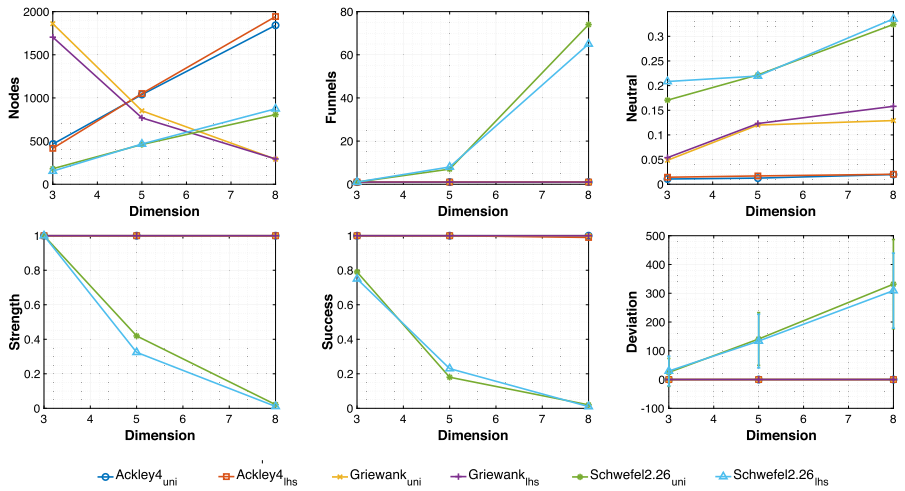

Fig. 2. Network and performance metrics on the synthetic functions. For each function, the subscripts *uni* and *lhs* denote uniform initialisation and hypercube sampling, respectively.

Let us first consider the performance metrics (Success and Deviation). The Ackley 4 and Griewank functions are easy to solve by the MBH algorithm, regardless of the initialisation method used. However, the MBH algorithm shows a reduction in the success rate in the Schwefel 2.26 function as the problem dimension increases. For $n = 3$, the uniform initialisation showed a higher success rate than the success rate of the LHS. However, with $n = 5$, it is the other way around, supporting that LHS might be beneficial for high dimension problems.

The problem becomes very hard to solve with $n = 8$, where MBH with both initialisation methods failed to find the global optimum in most runs. We can also verify the difficulty of the Schwefel 2.26 function, looking at how the deviation from the optimum increases with n.

With respect to the network metrics, the number of nodes in the CMLON model increases with the problem dimension for Ackley 4 and Schwefel 2.26. For the Griewank function, the trend is reversed, the number of nodes decreases when the problem dimension increases. These results are aligned with the earlier findings by Locatelli et al. [11], where the Griewank function was found to behave closer to its convex quadratic function as n increases. Theoretically, the number of local optima in the Griewank function increases exponentially with n due to its oscillatory non-convex part; however, local optimisers (such as L-BFGS) can only capture a lower number of optima because the convex quadratic function is more pronounced than the oscillatory non-convex component. The number of funnels indicates that Ackley 4 and Griewank feature a single funnel structure for all the studied problem dimensions; which helps to explain that these functions are not difficult to solve. For the Schwefel 2.26 function, as the dimension increases, some sub-optimal funnels appear, which helps to explain the deterioration in the MBH performance. The neutral metric represents the proportion of connected local optima with the same fitness value. In an analysis per function, we can observe that Schwefel 2.26 shows higher neutrality than the other functions. Finally, the strength metric measures the incoming weighted degree of the global optimum node. We can appreciate visually that this metric correlates well with the success rate. For Ackley 4 and Griewank, the strength is the highest possible value (1.0), and it does not seem to be affected by n. For Schwefel 2.26, an increase in n represents a reduction in strength.

Network Visualisation. Visualisation is a useful tool to get insight into the structure of networks. Figure 3 shows the CMLONs for the synthetic functions with $n = \{3, 5, 8\}$; the plots captions indicate the success rate for each instance. Due to space restrictions, we only show networks generated with uniform initialisation. However, both initialisation methods produced similar visual results. The graphs decorations highlight relevant features of the search dynamics; node sizes are proportional to their incoming strength, and edges to their weight, i.e. the number of times a transition between two nodes occurred in the sampling process. Pink nodes correspond to global funnels, and blue nodes to sub-optimal funnels. The global optima is highlighted with bright red, while the sub-optimal funnel bottoms with dark blue nodes. The Ackley 4 function shows a single funnel structure for all dimensions, with the number of nodes increasing with n. The Griewank function also shows a single funnel for all dimensions; however, the number of nodes decreases with n as discussed above, and also due to the higher neutrality observed in this function. For the Schwefel 2.26 function, sub-optimal funnels (blue nodes) start to emerge as n increases, which explains the success rate deteriorating with n.

To illustrate the advantage of modelling the search space with LONs, we used 3D visualisations of the networks to gain insight into the landscapes funnel structure. Figure 4 shows 3D networks of the synthetic functions with $n = 5$, where the fitness values are added as the third dimension to the 2D graph layouts. We can observe that Ackley 4 shows a deep funnel towards a single global optimum. Griewank also shows a single funnel, but with a flatter structure; we can see the several local optima are located almost at the same fitness level than the global optimum. Schwefel 2.26 shows multiple funnels, where the sub-optimal funnel bottoms (dark blue nodes) are at different fitness levels. Contrasting the standard 3D visualisation, only possible for functions with $n = 2$ (Fig. 1), with the network visualisations for $n = 5$ (Fig. 4), we can observe that the overall shape of the functions global structure is maintained. The networks

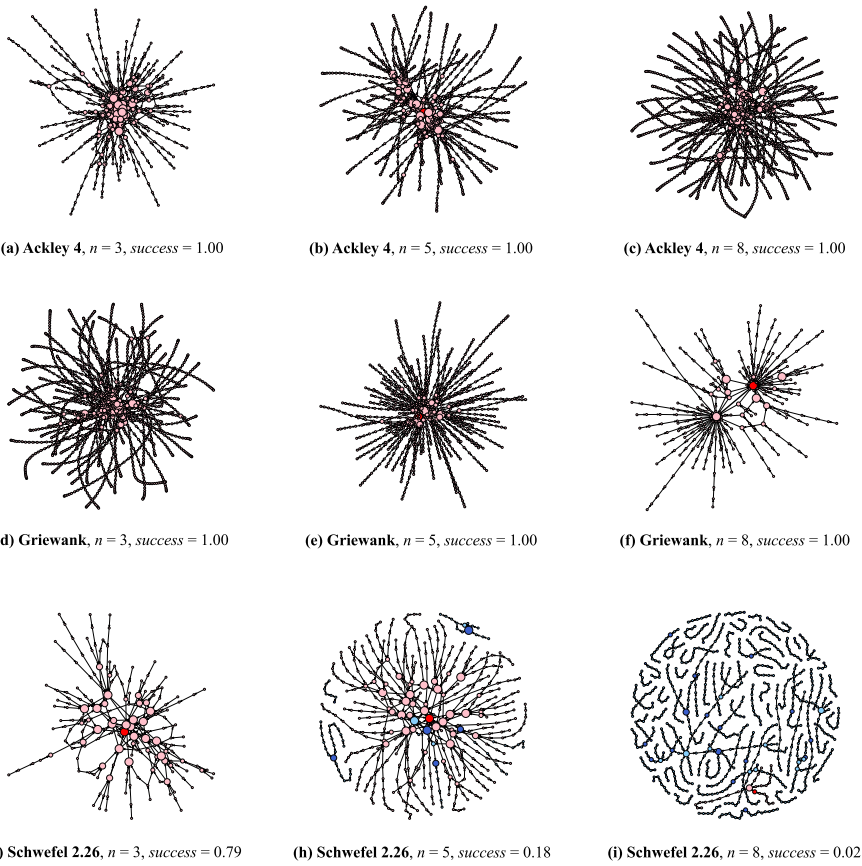

Fig. 3. CMLON visualisations for the synthetic instances with $n = \{3, 5, 8\}$. Node sizes are proportional to their incoming strength. Pink nodes belong to the funnel containing the global optimum (red node), while blue nodes belong to sub-optimal funnels (bottoms coloured in a dark blue). (Color figure online)

visualisations, therefore, offer a novel way of visualising the structure of functions of higher dimensions.

5.2 Real-World Problems

Performance and Network Metrics. Figure 5 reports the performance and network metrics described in Table 1 for the real-world functions with $n = \{3, 5, 8\}$ for the radar function and $n = \{4, 6, 8\}$ for the clustering function.

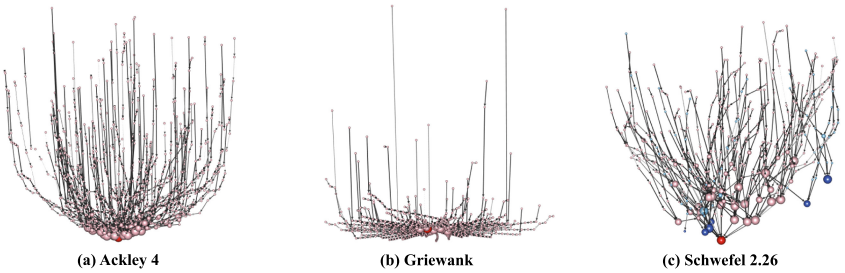

Fig. 4. 3D visualisation of the CMLON networks for the three synthetic functions with $n = 5$. The fitness value represents the third dimension. Nodes sizes are proportional to their incoming strength. Pink nodes belong to the funnel containing the global optimum (red node), while blue nodes belong to sub-optimal funnels (bottoms coloured in a dark blue). (Color figure online)

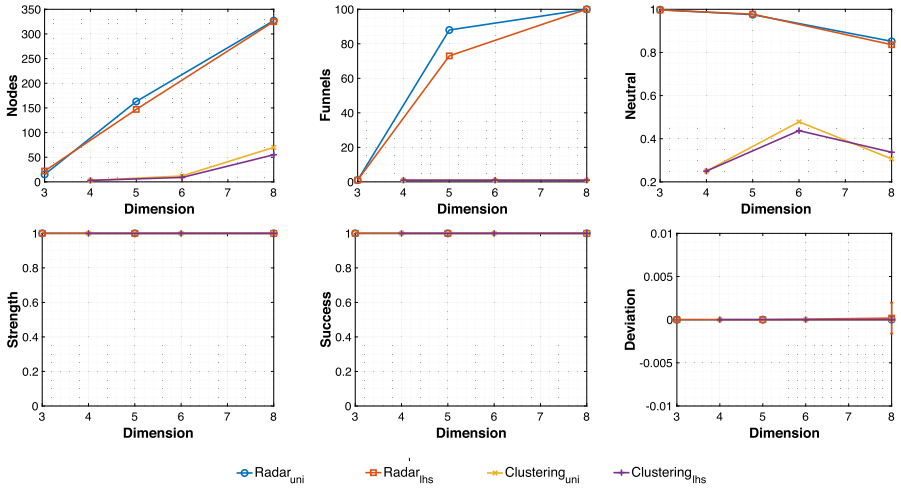

Fig. 5. Network and performance metrics on the real-world functions. For each function, the subscripts *uni* and *lhs* denote uniform initialization and hypercube sampling, respectively.

Results are shown for both initialisation methods, which are indicated using the subscript *uni* for the uniform initialisation and *lhs* for the hypercube sampling. The performance metrics (Success and Deviation) indicate that the real-world problems are easy to solve for all the dimensions and initialisation methods considered. The strength network metric (1.0 in all cases) clearly correlates with the performance metrics, supporting that the studied problems are easy to solve. Looking at the other network metrics, we can see that the number nodes increases with the dimension for both problems, and it is lower for the clustering problem. When contrasting with the synthetic functions (Fig. 2), we can see that the real-world problems produced much lower number of nodes. For the clustering function, a single global funnel is observed in all dimensions. The number of funnels increases with the dimension for the radar function, but all these funnels are global, there are no sub-optimal funnels in the studied real-world problems. This is consistent with the good performance of MBH in these problems. The differences between the two sampling methods are not marked for most metrics, with uniform sampling showing higher values for some of them. The neutral metric clearly indicates higher values for the real-world problems as compared with the synthetic functions (Fig. 2). Specially the radar function shows very high neutrality. Finally, another marked difference we found between the synthetic and real-world functions was the total number of different global optima. The synthetic functions revealed a very low number of global optima in all the studied dimensions (two for the Ackley 4 function, and one for the Griewank and Schwefel 2.26 functions) whereas the real-world functions produced a larger number of global optima. For example, for the uniform sampling, the number of global optima for the radar function was: 7014, 6477, and 1934, for $n = 3$, 5, and 8, respectively; whereas for the clustering function the number of global optima was 2, 6 and 24, for $n = 4$, 6, and 8, respectively.

Network Visualisation. Figure 6 shows the CMLONs for the real-world functions. For consistency with Fig. 3, the captions also indicate the success rate, which was 1.0 in all cases. We can observe that the networks for the real-world problems (Fig. 6) are much smaller (in terms of the number of nodes and edges) than the synthetic functions (Fig. 3); their overall structure is also different. The radar function with $n = 3$ has several thousands of global optima (7014 in our sampling process), however they are all connected and compressed into a single node in the CMLON model (red node in Fig. 6(a)). The CMLON in this case shows a single easy to traverse funnel structure, where one or two hops (edges) are sufficient to reach a global optimum. As n increases (Fig. 6(b) and (c)) the number of funnel sinks (nodes without outgoing edges) increases, but they are all global sinks (red nodes). Moreover, from none to three edges are sufficient to reach a global optimum (no edges means that the first optimum attained is a global optimum), whereas more steps are required to reach a global optimum for the easy to solve synthetic functions (Fig. 3(a)–(f)). The clustering function (Fig. 6(d)–(f)), shows small networks with a single global funnel and short paths

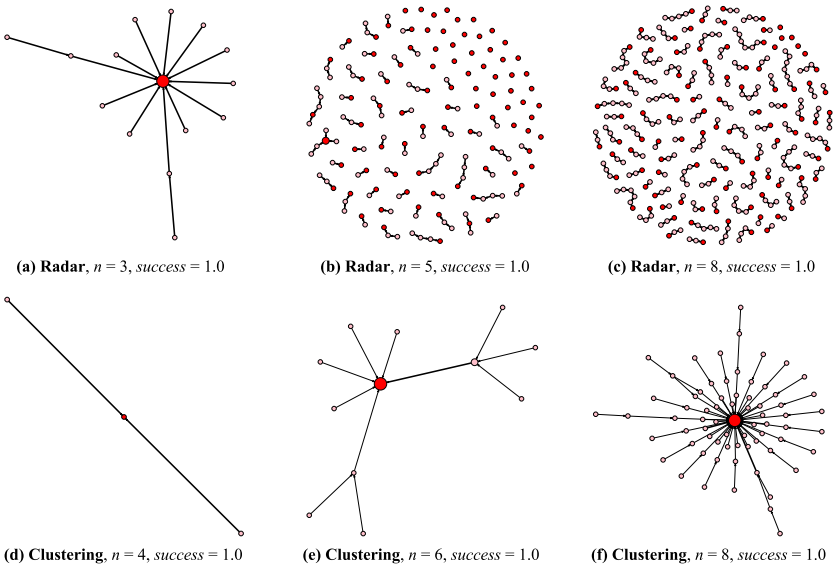

Fig. 6. CMLON visualisations for the real-world problems with different dimensions.

lengths to attain the global optimum. The networks for this problem increase in size (number of nodes and edges) with increasing problem dimension.

6 Conclusions

We extracted and analysed basin-hoping network models from synthetic and real-world continuous optimisation problems, and explored the effect of alternative initialisation methods as well as the problem dimension. Our main goal was to analyse and contrast the global (funnel) structure of the studied landscapes. The number of funnels tend to increase with the problem dimension, however, only one of the studied synthetic functions, Schwefel 2.26, showed sub-optimal funnels. We found striking differences between the synthetic and the real-world functions. Surprisingly, the real-world functions were easier to solve than the synthetic functions and our network analysis and visualisation revealed why this is the case.

The real-world problems produced much smaller network models (in terms of nodes and edges) and optimal funnels with shallow depth; a few hops were sufficient to reach a global optimum. The real-world problems also have many global optima, as opposed to one or two as observed in the synthetic functions, and much higher levels of neutrality measured as the proportion of connected local optima at the same fitness level. These findings indicate that designing and improving algorithms according to their performance on synthetic functions may be misleading. Our approach offers a novel way of visualising and analysing the global structure of continuous functions with more than two variables. Future

work will explore higher dimensions and additional real-world problems. The new set of metrics extracted from network models can also be used to guide parameter tuning and automatic algorithm selection in real-world engineering and machine learning optimisation problems.

Acknowledgment. Marco A. Contreras-Cruz thanks to the National Council of Science and Technology (CONACYT) for the scholarship with identification number 568675/302121. He also thanks to the Office of Research and Graduate Programs (DAIP) of the University of Guanajuato and the CONACYT, for the financial support during his research visit to the University of Stirling (from June to December 2019).

References

1. Adair, J., Ochoa, G., Malan, K.M.: Local optima networks for continuous fitness landscapes. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1407–1414. ACM (2019)
2. Ballard, A.J., et al.: Energy landscapes for machine learning. Phys. Chem. Chem. Phys. **19**(20), 12585–12603 (2017)
3. Csardi, G., Nepusz, T.: The igraph software package for complex network research. Int. J. Complex Syst. **1965**, 1–9 (2006). http://igraph.org
4. Das, S., Suganthan, P.N.: Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, pp. 341–359. Jadavpur University, Nanyang Technological University, Kolkata (2010)
5. Doye, J.P.: Network topology of a potential energy landscape: a static scale-free network. Phys. Rev. Lett. **88**(23), 238701 (2002)
6. Doye, J.P., Miller, M.A., Wales, D.J.: The double-funnel energy landscape of the 38-atom Lennard-Jones cluster. J. Chem. Phys. **110**(14), 6896–6906 (1999)
7. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimization problems. arXiv preprint arXiv:1308.4008 (2013)
8. Jones, E., Oliphant, T., Peterson, P.: SciPy: open source scientific tools for Python (2001). http://www.scipy.org/
9. Kerschke, P., Gallagher, M., Preuss, M., Teytaud, O.: The machine learning and data analysis (MLDA) problem set, v1 (2019). https://www.wi.uni-muenster.de/sites/wi/files/users/kerschke/gecco2019/gecco2019_umlop_mlda.pdf
10. Leary, R.H.: Global optimization on funneling landscapes. J. Glob. Optim. **18**(4), 367–383 (2000)
11. Locatelli, M.: A note on the Griewank test function. J. Glob. Optim. **25**(2), 169–174 (2003)
12. Malan, K., Engelbrecht, A.P.: A survey of techniques for characterising fitness landscapes and some possible ways forward. Inform. Sci. **241**, 148–163 (2013)
13. Nocedal, J., Wright, S.: Numerical Optimization. Springer Science & Business Media, New York (2006)
14. Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A study of NK landscapes, basins and local optima networks. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 555–562. ACM (2008)
15. Ochoa, G., Veerapen, N.: Mapping the global structure of TSP fitness landscapes. J. Heuristics **24**(3), 265–294 (2017). https://doi.org/10.1007/s10732-017-9334-0

16. Ochoa, G., Veerapen, N., Daolio, F., Tomassini, M.: Understanding phase transitions with local optima networks: number partitioning as a case study. In: Hu, B., López-Ibáñez, M. (eds.) EvoCOP 2017. LNCS, vol. 10197, pp. 233–248. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55453-2_16
17. Vinkó, T., Gelle, K.: Basin-hopping networks of continuous global optimization problems. Cent. Eur. J. Oper. Res. **25**(4), 985–1006 (2017)
18. Wales, D.J., Doye, J.P.: Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J. Phys. Chem. A **101**(28), 5111–5116 (1997)