CEJOR manuscript No.
(will be inserted by the editor)

Basin Hopping Networks of Continuous Global Optimization Problems

Tamás Vinkó · **Kitti Gelle**

Received: date / Accepted: date

Abstract Characterization of optimization problems with respect to their solvability is one of the focal points of many research projects in the field of global optimization. Our study contributes to these efforts with the usage of the computational and mathematical tools of network science. Given an optimization problem, a network formed by all the minima found by an optimization method can be constructed. In this paper we use the Basin Hopping method on well-known benchmarking problems and investigate the resulting networks using several measures.

Keywords benchmarking · network science · continuous global optimization · Basin Hopping

1 Introduction

The task of box-constrained global optimization (GO) is to find the solution to the problem

$$\min_{x \in S} f(x), \tag{1}$$

where $f : S \subset \mathbb{R}^n \to \mathbb{R}$ is a continuous function and S is a box. The vast literature of GO contains several proposed algorithms for solving (1), and it is a question of high interest how these algorithms perform on different problems. To this end, several benchmarking techniques have already been proposed (see, e.g. [7,20,24,29]). Our method complements these works with the help of the emerging field of network science [25]. The proposed methodology follows the core idea of the early work of Stillinger and Weber [31], in which potential energy landscapes of atom clusters were

T. Vinkó, K. Gelle
University of Szeged, Institute of Informatics
H-6720 Szeged, Árpád tér 2, Hungary
Tel.: +36-62-546 193
Fax: +36-62-546 397
E-mail: tvinko@inf.u-szeged.hu

formed into graphs. This is done in a way that the landscapes can be divided into basins of attraction surrounding each locally minimal energy level. This approach was later applied in the analysis of network topology of small Lennard-Jones clusters [8]. In that paper, the so-called *inherent structure network* (ISN) was built in which vertices correspond to the minima and the edges link those minima which are directly connected by a transition state. The same idea can be used for combinatorial optimization problems [30,34]. We give here a possible extension of these ideas to the space of continuous optimization problems[1], under the assumption that the optimization method used is Basin Hopping (BH). BH is a primary heuristic method which could be considered as the basis of many elaborate heuristic-based global optimization algorithms.

Once the network representation G of a global optimization problem P is constructed, similarly to the above mentioned ISN, many interesting graph metrics and measures of G can be calculated which can shed a light on several detailed characteristics of P. The important questions we aim at answering in this paper are the following:

- What kind of graph representations can be constructed for continuous global optimization problems?
- Practically, how difficult is it to find these graphs?
- From the network science literature, what are the interesting and relevant measures and what are the interpretations of them in the context of continuous global optimization?
- Given the networks and their measures, how can these be meaningfully applied together on (well-known) optimization problems and what are their implications?

In the following we first give an overview of the methodology producing the graph models. Then, we discuss several graph metrics and measures together with their interpretation in the context of continuous global optimization problems. This is followed by numerical experiments in which some benchmark optimization problems from the literature are investigated. Details on the network models of the tested functions are given, which we believe give further contributions to the understanding of why some problems are easy or hard for a particularly efficient optimization scheme called Basin Hopping.

2 Methodology

2.1 Network representation of optimization problems

Interestingly, an early paper of Locatelli [17] and the recent book of Locatelli and Schoen [18] already contain the idea of the (possible) construction of the network representing a continuous global optimization problem. In the following, using the terminology from [18], we give the necessary definitions of the graph construction.

[1] Note that the optimization problem (1) can also be extended to have constraints, although in the experimental part of our paper we will investigate only box-constrained problems of form (1)

First of all, we assume that a local search procedure $\mathscr{L}(\cdot)$ is available which, given a starting point y returns a locally optimal solution z of f characterized by $\|x-z\| \leq \varepsilon \implies f(z) \leq f(x) \quad (\forall x \in S)$. We associate a neighborhood structure $\mathscr{N}(\cdot)$ to each point in the search space S: for a given point $x \in S$, $\mathscr{N}(x)$ contains those points of S which we get by perturbation of x and subsequently starting a local optimization method from the perturbed point. Practically, the structure $\mathscr{N}$ depends on the underlying local optimization algorithm used to solve the global optimization problem (1). The *Local Optima Network* $G(V,E)$ can be defined in the following way. First of all, it is assumed that $\mathscr{L}(x) = x$ if x is a local minimizer point of f.

- The set V of vertices are the local minimizer points of f:

$$V = \{y \in S \ : \ \exists x \in S, y = \mathscr{L}(x)\}.$$

  Note that we need to assume that $|V| < \infty$.
- The set E of edges is defined as

$$E = \{(x,y) \in V \times V \mid \exists z \in \mathscr{N}(x) : \mathscr{L}(z) = y \text{ and } x \neq y\}.$$

Remark that the elements of set E are *directed*. Similarly to [18], a *monotonic* graph $G_m(V,E_m)$ can also be defined with the edge set

$$E_m = \{(x,y) \in V \times V \mid \exists z \in \mathscr{N}(x) : \mathscr{L}(z) = y \text{ and } f(y) \leq f(x) \text{ and } x \neq y\}.$$

We say that a local minimizer y is a *neighbor* of another local minimizer x iff $(x,y) \in E$. Note that in $G_m(V,E_m)$ all nodes with no outgoing arcs are locally optimal solution of (1).

We will also use the concept of the *adjacency matrix* A of a graph G in the later notations, which is defined as

$$A_{ij} = \begin{cases} 1 & \text{if } \ (i,j) \in E(G), \ \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we define the *natural* Local Optima Network (NLON). In this representation, two nodes are connected if they are separated by a critical point (i.e. a stationary point where the Hessian has a single negative eigenvalue [21]). Separation of two local minima x_1 and x_2 means that starting a gradient descent local search $\mathscr{L}$ from a point which is given by arbitrarily small perturbation of the critical point can lead to either x_1 or x_2.

Illustration. As an illustrative example, NLON of the classical, two dimensional Six Hump Camel Back (SHCB) global optimization problem is shown on Figure 1. This problem has 6 local optima among which two of them are global optima (shown as larger (blue) nodes). The labels on the nodes represent the two dimensional coordinates of the corresponding local optima. Size of the nodes are proportional to their degree.

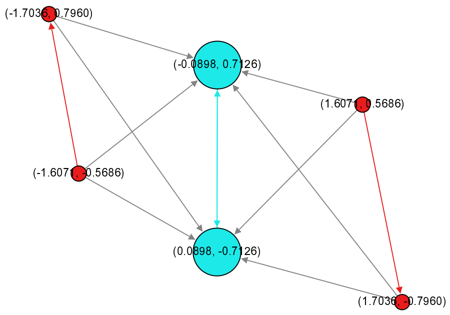


Fig. 1: The natural Local Optima Network of the SHCB global optimization problem

2.2 Basin Hopping method

The Basin Hopping (BH) method is a metaheuristic, which proved to be very efficient in solving global optimization problems [14, 18, 36]. Using the terminology of [18] the high level description is given in Algorithm 1. In the following, we refer to the lines of Algorithm 1 to give a detailed description. It is assumed that a uniform pseudorandom generator $\mathscr{U}(\cdot)$ is provided and the input is a continuous global optimization problem of form (1). In Line 1 a starting point y is generated uniformly at random in the search space S. Using a local search procedure $\mathscr{L}$ a local minimizer point x is found in Line 2. Line 4 selects a new starting point from the global neighborhood (to be defined later) of x. In order to do so, we let d be an n-dimensional Gaussian$(0,1)$ random vector with $\|d\| = 1$ (e.g. d is a random direction), and r_2 be a positive fixed step size. The new starting point z is generated as being $x + r_2 d$. In Line 5 a local search is performed starting from z and its result is stored as x (a local minimizer point). Line 7 selects a new starting point z from the local neighborhood of x. This is done by sampling a uniformly random point over $S \cap B[x', r_1]$, where $B[x, r]$ is a box centered at x and having half-edge length $r > 0$. We start a local search from z and its result is stored as y (Line 8). In Line 9 we check whether y is a better solution than x (being 'better' is to be defined later). In Lines 12 and 13 we check whether the local and global stopping criteria are satisfied, respectively. The algorithm returns with the local minimizer point x and the corresponding function value $f(x)$ in Line 14.

The conditional statement in Line 9 requires the procedure *IsAcceptable*(x, y) to be given. This procedure can be implemented in different ways, the most common approaches are as follows:

Monotonic: the procedure *IsAcceptable*(x, y) returns whether $f(y) < f(x)$.
Generic: the procedure *IsAcceptable*(x, y) returns whether

$$\mathscr{U}[0,1] \leq \exp(-(f(y) - f(x))/T),$$

where T is a nonnegative parameter (called *temperature* in the literature), which iteratively gets decreased during the execution of Algorithm 1. Note that this version of the algorithm occasionally accepts non-improving local solutions as well.

Algorithm 1 Basin Hopping method

```
 1: $y := \mathscr{U}(S)$;
 2: $x := \mathscr{L}(y)$;
 3: repeat
 4:     $z := \mathscr{U}(\mathscr{N}_g(x))$;
 5:     $x := \mathscr{L}(z)$;
 6:     repeat
 7:         $z := \mathscr{U}(\mathscr{N}_\ell(x))$;
 8:         $y := \mathscr{L}(z)$;
 9:         if $IsAcceptable(x, y)$ then
10:             $x := y$;
11:         end if
12:     until local stopping rule is not satisfied
13: until global stopping rule is not satisfied
14: return $x, f(x)$
```

Furthermore, there are two procedures in Algorithm 1, namely $\mathscr{N}_g(\cdot)$ and $\mathscr{N}_\ell(\cdot)$, which needed to be defined in detail. These procedures correspond to the local search at Level 3 and Level 2, respectively, of the multi level optimization approach of Locatelli [17]. We employ the scheme from [17], where the neighbors of a local minimum x_0 are all the local minima whose basins of attraction have a nonempty intersection with the box $B[x_0, r] \cap S$. Here $B[x_0, r] := [x_0 - r\mathbf{1}, x_0 + r\mathbf{1}]$, with half-edge length $r > 0$ and centered at x_0 (and $\mathbf{1}$ is the vector whose components are all equal to 1). As this definition depends on the parameter r (which appears to be either r_1 or r_2 in Algorithm 1) an adaptive scheme can be used which iteratively updates its value – for full details see [17].

2.3 Building the Basin Hopping Network

In order to build the local optima network for a particular optimization problem we applied an optimization scheme based on the BH method. Using the same terminology as in Section 2.2 the high level description is given in Algorithm 2.

In the following, we refer to the lines of Algorithm 2 to give a detailed description. The algorithm starts with an empty graph G_w, which iteratively gets expanded if new nodes and edges are found. In Line 1 a starting point y is generated uniformly at random in the search space S. The first node x of the graph G_w is found in Line 2. Line 4 selects a new starting point from the global neighborhood of x using the same technique in Algorithm 1. In Line 5 a local search is performed starting from z and its result x (as a local minimizer point) is added to the set of vertices. Note that it is possible that the local search finds a solution which has already been found earlier. In a computer implementation using floating-point arithmetic, one needs to apply ε-tolerance here, e.g. to check if $||x - \tilde{x}||_2 < \varepsilon$ for any $\tilde{x} \in V$ and prescribed $\varepsilon > 0$. Thus it is not given that the set V gets expanded in each iteration. In Line 7 we store the previously found local solution x in a temporary variable x'. This will be needed to construct new edges of the graph G_w. Line 8 selects a new starting point y from the local neighborhood of x', similarly to Line 7 Algorithm 1. What is done in Line 9 is that we start a new local search from y, and its result x is added to the set of nodes V,

Algorithm 2 Basin Hopping Network builder algorithm

```
Require: Global optimization problem $P$
 1: $y := \mathscr{U}(S)$;
 2: $x := \mathscr{L}(y)$; $V := \{x\}$;
 3: repeat
 4:     $z := \mathscr{U}(\mathscr{N}_g(x))$;
 5:     $x := \mathscr{L}(z)$; $V := V \cup \{x\}$;
 6:     repeat
 7:         $x' := x$;
 8:         $y := \mathscr{U}(\mathscr{N}_\ell(x))$;
 9:         $x := \mathscr{L}(y)$; $V := V \cup \{x\}$; $E := E \cup (x', x)$
10:     until local stopping rule is not satisfied
11: until global stopping rule is not satisfied
12: return $G_w(V, E)$
```

as well as the edge (x', x) to the set of edges E. In Line 10 and 11 we check whether the local and global stopping criteria are satisfied, respectively.

It is important to note that the output graph of Algorithm 2 is usually an approximation of the natural Local Optima Network of the input problem P. This is due to the fact that finding the natural LON is a computationally intractable task, especially for higher dimensions. Moreover, a computer implementation is based on floating-point numbers, thus checking if a new node is found can only be done with pre-defined and fixed precision only.

The efficiency of Algorithm 2 highly depends on the parameters r_1, r_2, on the stopping criteria used in Line 10 and 11, and on the local search procedure $\mathscr{L}$. The algorithm needs to find *all* local minima, thus it is usually better to let it run for longer time while allowing a larger number of iterations. According to our experiments, this usually leads to an output graph that has all the local minima of the optimization problem but with more edges than the natural LON. This means that, depending on $\mathscr{L}$, nodes which are not neighbors of each other in the natural LON get connected by an edge in the Basin Hopping Network. Thus, post-processing is necessary, which needs a slight modification of Algorithm 2 in the following way. When a potentially new edge is added to the graph in Line 9 we count how many times this edge has been found already. In this way, each edge in the resulting graph has a weight. The post-processing procedure then iterates through the list of edges and removes those ones whose weight is below a certain threshold. This threshold is chosen to be the P-th percentile calculated by the nearest rank method. In the numerical examples (see Section 4) we experimented with different values of P. Note that a similar procedure was proposed in [6].

Illustration. A possible Basin Hopping Network of the two dimensional Six Hump Camel Back function is shown in Figure 2. Note the differences between Figures 1 and 2.

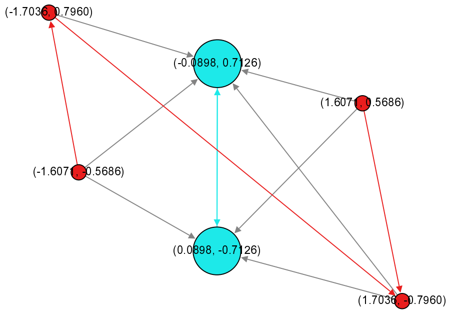


Fig. 2: A Basin Hopping Network of the SHCB global optimization problem

3 Graph measures

In the following we give a list of relevant graph measures, taken from network science literature, together with their interpretations in the context of LONs.

Size of the network. This measure is defined as the number of nodes, i.e. $|V|$. Clearly, this represents the number of local minima. As it has been argued, e.g., in [17] a higher number of minima does *not* imply that the problem at hand is more difficult to solve.

Neighborhood of a node. Besides the size of the network, this is also a critical feature to be found by Algorithm 2, as these two provide the basis for the following measures which are to capture the structural characteristics of the corresponding network. Put it differently, if Algorithm 2 is not able to find the correct network representation of the investigated global optimization problem P, then the measures listed in this section can lead to incorrect claims on P. The neighborhood set of node $i \in V$ in graph $G(V,E)$ is denoted by $N_i(G)$.

Path and shortest path. These are important definitions for further measures. The series of nodes $x = x_0, x_1, \dots, x_k = y$, where x_i is adjacent to x_{i+1}, is called a *walk* between the nodes x and y. If $x_i \neq x_j$ $(\forall i, j)$, then it is called a *path*. The *path length* is k. Given all paths between nodes x and y, a *shortest path* is a path with fewest edges. Shortest paths are usually not unique between two nodes. Note that most of the heuristic based global optimization methods basically do random walks on paths in a specific underlying graph. If the method is of monotonic type (like Monotonic Basin Hopping [36] or Differential evolution [32]) then it walks on G_m. Some methods, like Simulated Annealing [13], allow steps towards non-improving solutions, thus they walk on graph G.

Average path length. This is defined as the average value of all shortest paths in the network, denoted by ℓ. Networks with low average path length are called *small worlds*. More specifically, in small world networks the average path length grows proportionally to $\log(|V|)$. Intuitively, the small world property is a desirable feature in graphs corresponding to global optimization problems.

Diameter. The size of the longest of all shortest paths is called diameter, and it is denoted by D. This gives a worst-case scenario regarding the number of jumps that have to be taken to reach the global optimum. Similar to the average path length, the smaller the diameter is, the better it is.

Clustering coefficient. It measures the average probability that two neighbors of a node are themselves neighbors of each other. Formally, the *local clustering coefficient* of node i is

$$C_i = \frac{|\{(x,y) \in E \ : \ x,y \in N_i\}|}{k_i(k_i-1)},$$

where $k_i = |N_i|$. The definition of global clustering coefficient is based on triplets. A *triplet* consists of three nodes that are connected by either two (open triplet) or three (closed triplet) undirected ties. The *global clustering coefficient* C is the number of closed triplets over the total number of triplets (both open and closed).

Note that small world networks tend to have high clustering coefficient. Intuitively, networks with high C value correspond to easier to solve global optimization problems.

Node degree. The neighborhood structure $\mathscr{N}$ can be quantified. This gives the definition of node degree, which is the number of edges adjacent to a node. In our case, this measures the number of adjacent local optima. Since our graphs are directed, we have indegree and outdegree for a given node. Formally, the *outdegree* is a function $d^+ : V \to \mathbb{N}_0$ which for a node x gives $d^+(x) = |\{y \in V : (x,y) \in E\}|$. The *indegree* is defined as $d^-(x) = |\{y \in V : (y,x) \in E\}|$. Nodes with degree that greatly exceeds the average degree in the graph are called *hubs*. It is known that high degree nodes are easier to be found by random walks [25]. Hence, if the global optimum vertex is a hub, then a heuristic method can perform well on the problem.

Average degree. This measure is the ratio $\frac{1}{|V|}\sum_{x\in V} d(x)$, where $d(x)$ is either the indegree and outdegree (the average is the same value in both cases); and it is denoted by $\langle k \rangle$.

Degree distribution. This measure is defined as the probability distribution of all degrees in the graph. Formally, p_k is the fraction of nodes with degree k:

$$p_k = \frac{|\{x \in V : d(x) = k|}{|V|},$$

where $d(x)$ can be indegree or outdegree, or the sum of the two (i.e. the graph is made undirected). Degree distributions have two categories of particular interest: (*i*) *random networks* (also called Erdős-Rényi graphs [9]) have binomial distribution of degree k:

$$p_k = \binom{|V|-1}{k} p^k (1-p)^{|V|-1-k},$$

where p is the probability that two nodes are connected; and (*ii*) *scale-free networks* [2], which follow a power law distribution of the form $p_k \sim k^{-\alpha}$, where α is a parameter typically in the range $2 < \alpha < 3$.

The degree distribution is an important global measure of a network. Both random and scale-free networks have advantages and disadvantages. These networks tend to have small clustering coefficients and short average path length. By definition, scale-free networks contain a few hubs with high degree and lots of nodes with low degree. In contrast, random networks contain very similar nodes.

Community structure. It can be informally defined as a partition of vertices into groups in such a way that nodes are more connected within a group and sparsely connected between different groups [28]. Let H be a subgraph of G including node i. If the graph is directed, then define

$$k_i^{in}(H) := N_i(H), \text{ and } k_i^{out}(H) = N_i(G) \setminus N_i(H).$$

Moreover, $k_i(H) := k_i^{in}(H) + k_i^{out}(H)$. Now, one can define a subgraph H as a community in a *strong sense*, which is the case when $k_i^{in}(H) > k_i^{out}(H)$ holds $\forall i \in V(H)$; and also in a *weak sense*, when $\sum_{i \in H} k_i^{in}(H) > \sum_{i \in H} k_i^{out}(H)$. The number of communities we find in a network is denoted by K. Note that most of the community detection algorithms treat the graph as undirected. A high number of communities in G does not necessary imply a hard-to-solve optimization problem. However, if the problem is multimodal and the local minima are located in different communities then the Monotonic Basin Hopping method can have difficulties to find the global minimum.

Modularity. This quantity, denoted by Q, measures the fraction of the edges in the network that connect vertices of the same type (i. e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices [27]. Formally,

$$Q = \sum_i (e_{ii} - a_i^2),$$

where e_{ij} is the fraction of edges with one end vertices in community i and the other in community j, and a_i is the fraction of ends of edges that are attached to vertices in community i. Modularity intends to measure the strength of the community structure in a graph.

Betweenness centrality. This measure gives a local score to vertices by measuring the extent to which a vertex lies on paths between other vertices [11]. Mathematically, let n_{st}^i be the number of shortest paths from s to t that pass through i, and define g_{st} as the total number of shortest paths from s to t. Then the betweenness centrality (BC) of vertex i is $\sum_{st} \frac{n_{st}^i}{g_{st}}$. BC is usually calculated on undirected graphs. Since a global optimization method does not necessarily take shortest paths on G, a variant called Random Walk BC will instead be investigated in Section 4.

PageRank. This local measure is used on directed graphs, where the score of a vertex is derived from the scores of its network neighbors and it is proportional to their centrality divided by their out-degree. Formally, we need to calculate the vector $D(D-\alpha A)^{-1}\mathbf{1}$, where A is the adjacency matrix of the graph G_m, D is a diagonal matrix with elements $D_{ii} = \max\{d^+(i), 1\}$, $\mathbf{1}$ is again the vector whose components are all equal to 1 and α is a damping parameter (default $\alpha = 0.85$). PageRank was originally designed as an algorithm to rank web pages [4] and essentially the score it gives to a page reflects the chance that the random surfer will land on that page by clicking on a link. In the context of global optimization, higher PageRank score means higher chance to be found by the Monotonic Basin Hopping algorithm, which performs random walks on the directed network representing the optimization problem to be solved.

4 Numerical results

In this section we demonstrate the usage and implications of the analysis of the Basin Hopping Networks of global optimization problems. For this purpose, two well-known benchmarking problems have been selected from the literature which we discuss in Section 4.1 and 4.2 in full details. Further test functions are also analyzed in Section 4.3. We are interested to see if the global and local measures listed in Section 3 are able to characterize the solvability of the problems.

The implementation of Algorithm 2 was done in AMPL [10], which allows to use a very general class of objective functions and a large selection of local optimizer methods. In our tests we used MINOS [22] as local optimizer $\mathscr{L}$. The parameters were:

- the local stopping rule (in Line 10) was: 10000 iterations;
- the global stopping rule (in Line 11) was: 50 iterations;
- the parameter γ (see [17] for details) was set to 0.5;
- and the values of P in the post-processing were starting from 20 up to 70 with increment 5.

In order to compute the measures listed in Section 3, we used the `igraph` package in R and the `NetworkX` package in Python. Modularity Q and number of communities K were calculated with the method called Multi Level [3], which is based on local optimization of the modularity measure around a node.

As we have already discussed in Section 2.3, the output of the implemented procedure for a given global optimization problem is a set of graphs. These graphs are then used for two types of analysis.

- First, we need to select one of them, which gives the BHN representation of the problem. The selection of this graph is done in the following way. It is assumed that the global optimization problem is continuous, hence the BHN representation must be a connected graph. Furthermore, as a general rule, we select that connected graph which corresponds to a P value at which the diameter of the graph gets increased in case of choosing a larger P value. This is motivated by aiming at getting such BHN which is close to the natural LON of the problem.

If the diameter of the graph gets increased then it is an implication that we just removed a significant amount of edges than before. On the other hand, if the diameter does not change by removing edges, that means we have removed edges from the short ones from all shortest paths (i.e. we have removed unrealistic huge jumps between nodes which are far away from each other in the natural LON). The graph which represents the optimization problem can then be analyzed using the measures from Section 3.

- Secondly, the series of graphs can be considered as results of a certain edge-deleting procedure. This way the *robustness* of the graphs can be measured with respect to a particular metric called *random walk betweenness centrality* (RWBC) [26]. RWBC is a local measure, a particular variation of the betweenness centrality (see Section 3). It is based on random walks, counting how often a node is traversed by a random walk between two other nodes. Calculation of RWBC values are done on the vertices of graph G using the edge weights obtained by executing Algorithm 2, i.e., where we count how many times this edge has been found already. In particular, we essentially associate a relative quantity to the node corresponding to the global optimum and thus it can be seen and compared how it relates to the other nodes' RWBC values.

4.1 Griewank function

The first test function we study is proposed by Griewank [12] and it has the form

$$Griewank_n(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

Usually the search space used in the literature is $x_i \in [-600, 600], (i = 1, \ldots, n)$. However, as this function has a huge amount of local minima we restrict the search space to a much smaller one: $x \in [-28, 28]^n$. This restriction results in a smaller network, whose size can be justified by the literature [5].

The $Griewank_n$ function, independently from its dimension n, has exactly one global minimizer point with value 0, located at the origin. Although the number of its local minima is growing exponentially with n, the locations of these minima follow a regular pattern. This makes the corresponding network of simple form. Namely, in $n = 2$ it is a regular lattice, whose structure remains the same in higher dimensions as well.

Table 1: Network properties of Griewank graphs

graph	size	$\langle k \rangle$	ℓ	D	C	Q	K
G $(n = 2, P = 30)$	123	7.4796	4.7419	12	0.4810	0.6152	7
G_m $(n = 2, P = 30)$	123	3.7642	3.7609	11	0.4629	0.6179	7
G $(n = 3, P = 45)$	1359	6.8286	8.7206	20	0.1551	0.7019	12
G_m $(n = 3, P = 55)$	1359	2.8182	5.9961	17	0.0330	0.7180	13

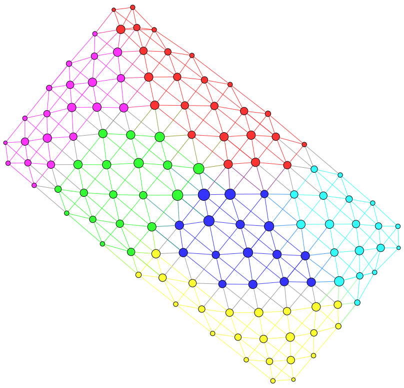


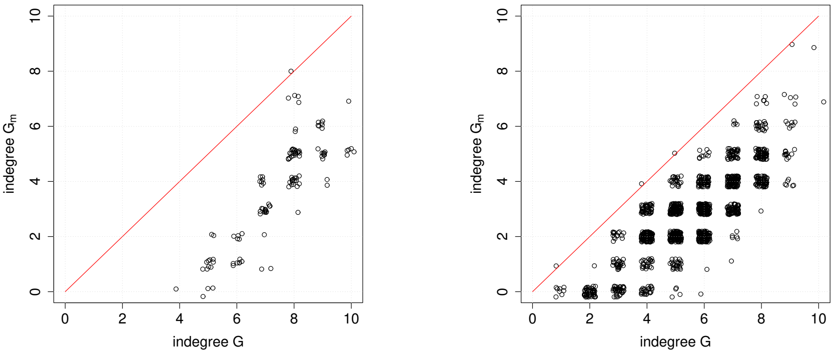
Fig. 3: A BSN of Griewank$_2$ function. Colors represent community structure, size of a node corresponds to its PageRank value

Graph measures. The summary of the graph measures are listed in Table 1. Note that the sizes of the networks reported here are in accordance with the (estimated) number of local optima reported in [5] if the search space is restricted to $[-28,28]^n$. We chose to study this test function first, mainly because of its regular structure, which is well illustrated on Figure 3. As we can see, almost all the nodes (apart from those at the edge) have the same degree, so this graph is a typical example of the Erdős-Rényi random networks (see Section 3).

It can be immediately noticed that the BHNs have relatively large diameters. This indicates that an optimization method needs to take a large number of iteration steps to guarantee success. This fact is already known from the literature, see, e.g. [16]. It is worth mentioning here that although these graphs have large modularity values, which implicates the presence of communities in the network, their nodes are very similar to each other with respect to their degree. Thus high Q values are misleading in these cases. We can also notice that the clustering coefficient C is much smaller for $n = 3$ than for $n = 2$, which should also be treated with care. In fact the simple reason for this is the BSN we found for $n = 3$ is incomplete compared to the natural LON representation. As we have already discussed, finding the natural LON representation of an optimization problem is practically impossible in general. Still, it can be constructed easily for the Griewank problem given its regular structure.

Concluding the analysis with the graph measures we can say that they do not give us any particular insights about the Griewank test problems.

Degree investigation. For investigating the degree distribution of the BHNs we propose the usage of a scatter plot on which the degree of the vertex of the undirected graph and the in-degree of the same vertex of the directed graph can be compared. This kind of visualization gives a very interesting landscape of the problem's local optima. Figure 4 shows the corresponding plots for the Griewank test function. By definition, no points can be above the red line. Note that in both cases the point representing the global optimum (which must be on the red line) is at the top right corner



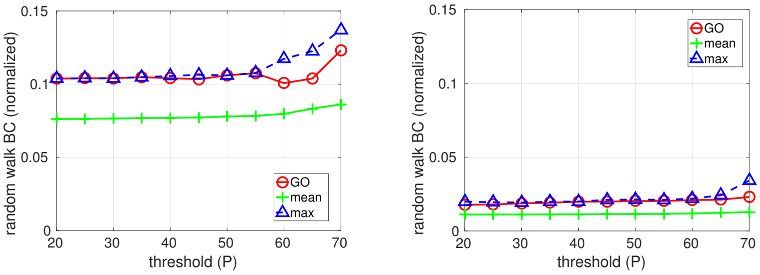
(a) $n = 2$

(b) $n = 3$

Fig. 4: Degree investigation of Griewank networks; the points are jittered for better visibility

of the figure and the other points are beneath. This implies that the Monotonic Basin Hopping method has a much better chance to find the global optimizer point than the Generic BH method in which steps towards non-improving solutions are allowed.

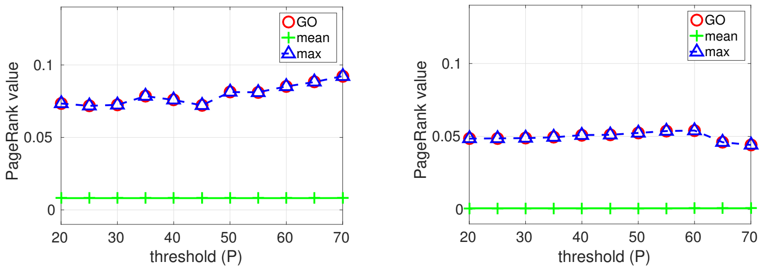
Robustness of BHNs. Using the graph sequences we obtained from Algorithm 2 we calculated the random walk betweenness centrality (RWBC) values. The results of these experiments are shown on Figure 5. Note that a higher P value means a sparser graph, thus higher P values correspond to such runnings of the Basin Hopping method where the number of iterations are relatively small (compared to those represented by lower P values). For both cases the RWBC value of the global optimum is higher than the nodes' average RWBC value. We can also see that for many P values the global optimum vertex has the highest RWBC value, especially for low P values. Clearly, nodes with high RWBC values are easier to be found by random walks. Thus, we can conclude that finding the global optimum by Basin Hopping using the general



(a) $n = 2$

(b) $n = 3$

Fig. 5: Random walk betweenness centralities of Griewank networks



(a) $n = 2$ (b) $n = 3$

Fig. 6: PageRank values of Griewank networks. Note that the global optimum vertex has the highest PageRank score.

approach is not hopeless, it is only a matter of allowing large numbers of iterations. On the other hand, it is also indicated by these figures that the RBWS values do not really change for lower P values, thus, by only letting the BH search run for a longer time does not guarantee success in global optimization.

Turning now our attention to the *monotonic* network representations, we have already seen in Figure 3 that due to the special structure of the Griewank functions the global optimum node has the highest PageRank score. Figure 6 shows the calculated values for the different P levels together with the mean PageRank scores. Note that the PageRank value of the global optimum is the highest, hence there are overlaps on the figures. It is clearly advised that using the BH method for solving the Griewank problems should be done using the Monotonic approach.

4.2 Schwefel

Another test problem we study is the Schwefel function which is defined as follows:

$$Schwefel_n(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \qquad x_i \in [-500, 500].$$

This problem differs from the previous one in a sense that it has exponentially growing number of local minimizer points whose values are very close to the global optimum and, more importantly, they are located at different regions of the search domain. Thus, this function is considered as a hard problem instance for global optimization methods.

Graph measures. The properties of the BHNs we found for the Schwefel problems are listed in Table 2. Comparing the different quantities to the ones we obtained for the Griewank functions, we can immediately see the differences everywhere. First of all, the Schwefel networks have very small diameter as well as small average path lengths. This means that the BH method can discover the entire network in reasonable time. However, it must be emphasized that this is true for the BH using the General

Table 2: Network properties of Schwefel graphs (directed graph)

name	size	$\langle k\rangle$	ℓ	D	C	Q	K
$G\ (n=2,P=50)$	64	14.9688	2.0761	4	0.5712	0.3679	4
$G_m\ (n=2,P=45)$	64	7.8281	2.0447	5	0.5478	0.4039	4
$G\ (n=3,P=30)$	502	17.4522	3.4073	7	0.3877	0.5345	6
$G_m\ (n=3,P=40)$	492	7.849593	3.6651	10	0.3655	0.5501	7

approach. The modularity values are not that high compared to those of the Griewank networks. Still, the community structure is clearly there in these Schwefel networks, as it is even shown on Figure 7. Note that the vertices representing the local optima are moved to the periphery for better visibility. We can see here a very interesting fact, namely that 3 out of 4 local optimizer points are in different communities. This is certainly an indication that the Schwefel functions are difficult problems for global optimization methods. In particular, applying the Monotonic approach for BH search is not advised in this case.

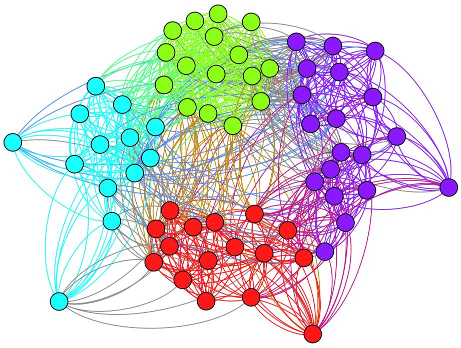
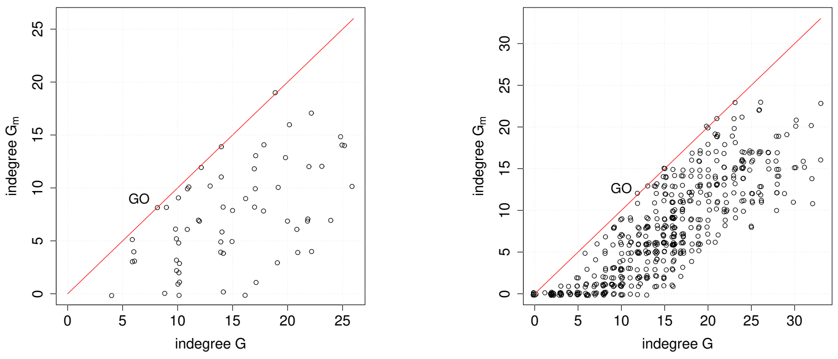


Fig. 7: A BSN of Schwefel$_2$ function; colors represent community structure

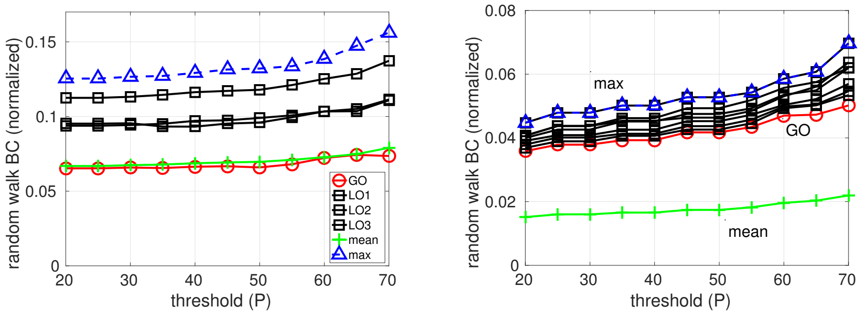
Degree investigation. Figure 8 shows the degree investigation of the Schwefel problems. In order to understand what makes this problem difficult to be solved (at least for BH) we note that the point representing the global optimum is always the one which has the lowest degree, i.e., it is the bottom left point on the red line, indicated by a label 'GO'. In particular, for $n = 3$, where the number of local optima is 8, there are many vertices having larger degree than that of the global optimum vertex and hence they are having higher probabilities to be found by random walk. Hence, this is another evidence for indicating the usefulness of applying the Generic BH approach for the Schwefel problems.

Robustness of BHNs. Finally, we have calculated the RWBC and PageRank scores for the series of Schwefel networks. Figure 9 shows the undirected case, thus it corresponds to the Generic Basin Hopping. We can immediately see that in these cases



(a) $n = 2$ (b) $n = 3$

Fig. 8: Degree investigation of Schwefel networks; points are jittered for better visibility

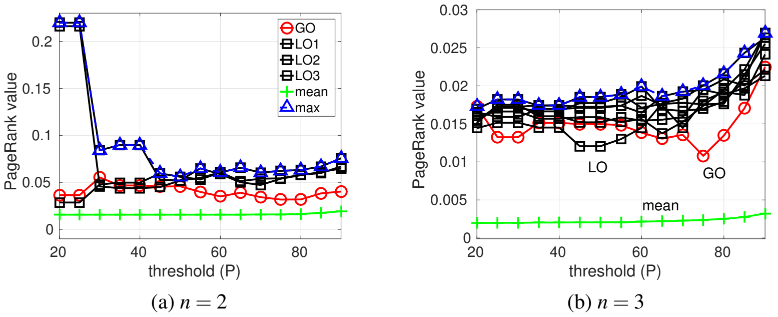


(a) $n = 2$ (b) $n = 3$

Fig. 9: Random walk betweenness centralities of Schwefel networks; black lines with square markers represent local optima

the global optimum vertex has lower value that those representing the local minima. Moreover, the node having the maximum RWBC score is a different one. For small P values (representing longer runs of the optimizer method) and $n = 3$, interestingly, the differences between the GO and the local minima are vanishing. However, this is not the case for $n = 2$. Though this does not imply that finding the global optimum of the Schwefel function is easier for higher dimension, it only indicates that for higher dimension the probabilities of finding any local minima (including the global one) are roughly equal. Hence, the advice here is to use the Generic Basin Hopping, which can more easily escape from local minimizer points compared to the Monotonic approach.

Regarding PageRank values on the directed networks, we obtain a completely different result, see Figure 10. In this case we include networks for higher P values,



(a) $n = 2$

(b) $n = 3$

Fig. 10: PageRank values of Schwefel networks; black lines with square markers represent local optima. Note the different scales on the y-axes.

which represent shorter BH runs. Although all the local optima have higher score than the average, the global optimum node ranks lower than the other optima. For large P values all of them are below the maximum score. When the P value is low, i.e., when the BH algorithm is allowed to take larger amount of iterations, the global optimum vertex has the highest PageRank score. The reason for this is very simple: being stuck in a local optimum by the Monotonic Basin Hopping, the only vertex to which we can jump is the global optimum node. Due to the recursive definition of PageRank, the global optimum node becomes the vertex of highest rank. Note that this happens when letting the MBH algorithm run for exceptionally long time.

4.3 Further test functions

In this section we show the analysis of further global optimization test functions. These functions are also extensively used as benchmarks in the GO literature, hence we do not give here the full definitions, only the references: Ackely [1], Levy8 [15], Rastrigin [35], and Sinusoidal [37]. As for the Griewank and Schwefel problems, the 2 and 3 dimensional versions of these additional functions were investigated. The results of the network measures are shown in Table 3.

We start with the discussion on Levy8. These functions have the smallest number of local minima, the smallest average path length and diameter, large clustering coefficients and the smallest number of communities. The degree investigation of Levy8 graphs are shown on Figure 11. For $n = 2$ the global optimizer node has the highest indegree in G_m and there is only one node which has higher indegree in G. Similar trend can be noticed for $n = 3$. We conclude that the Levy8 functions are the most simple ones for MBH. These indicators are in lines with the experiments done in [19] using MBH.

The Ackely and Rastrigin problems are similar to the already analyzed Griewank problem with respect to their landscape, their corresponding BH networks show rather regular grid structure. On the other hand, as we can see from the graph measures, the Ackely and Rastrigin functions have less number of nodes, larger average

Table 3: Network properties of additional test functions

name	size	$\langle k \rangle$	ℓ	D	C	Q	K
Levy8 $(n=2, P=40)$	47	13.0426	1.9172	4	0.5917	0.2035	4
Levy8$_m$ $(n=2, P=70)$	45	3.8222	1.8422	4	0.4386	0.3217	4
Levy8 $(n=3, P=35)$	97	9.4124	2.4099	5	0.4728	0.2612	5
Levy8$_m$ $(n=3, P=50)$	78	4.3333	2.1189	5	0.4353	0.3928	4
Ackely $(n=2, P=30)$	111	14.5225	2.4985	6	0.5988	0.2103	5
Ackley$_m$ $(n=2, P=20)$	109	7.1927	2.3597	7	0.5766	0.3638	6
Ackley $(n=3, P=30)$	358	13.9469	3.0894	7	0.4427	0.2452	5
Ackley$_m$ $(n=3, P=30)$	356	7.5365	2.7845	9	0.3928	0.4361	6
Rastrigin $(n=2, P=20)$	118	21.6102	2.2024	6	0.5933	0.1704	4
Rastrigin$_m$ $(n=2, P=30)$	116	10.0086	2.0473	6	0.5394	0.2714	5
Rastrigin $(n=3, P=65)$	335	13.2298	2.8954	8	0.3728	0.2548	10
Rastrigin$_m$ $(n=3, P=60)$	351	9.3988	3.0543	14	0.3717	0.2905	9
Sinusoidal $(n=2, P=25)$	178	22.6348	2.3764	6	0.5455	0.2010	5
Sinusoidal$_m$ $(n=2, P=25)$	167	10.3353	2.5047	6	0.4918	0.3723	6
Sinusoidal $(n=3, P=65)$	912	12.2983	3.9024	12	0.3365	0.3557	7
Sinusoidal$_m$ $(n=3, P=60)$	946	7.7833	3.1646	10	0.2892	0.4276	10

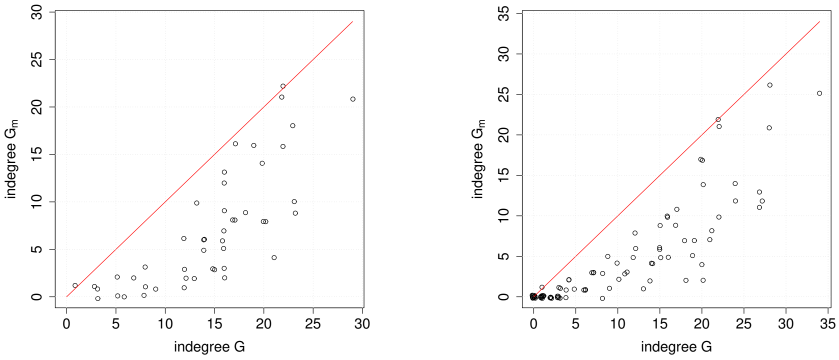
degree, smaller average path length and diameter compared to Griewank. The degree investigation figures of Ackely functions (see Figure 12) are similar to Griewank in the sense that there are only a few nodes which have higher degree than the global minimizer. In line with the experiments done in [19] using MBH, Rastrigin functions are slightly more difficult to solve, which can also be demonstrated by the degree investigation, see Figure 13. We conclude that these test problems can be solved easier than the Griewank problem.

Finally, the Sinusoidal test problem has the largest number of nodes. This simple fact does not make it difficult to solve. As it can be seen in Figure 14, especially for $n = 3$, the global minimizer node has the highest degree.

5 Conclusions

Basin Hopping Networks are interesting representations of global optimization problems. Using the rich set of measures and metrics from network science lots of properties can be analyzed regarding the solvability of continuous problems by the fundamental heuristic method Basin Hopping. In this paper we have investigated some well-known benchmark problems, hence our contribution here can be regarded as '*telling classical optimization stories in the language of network science*'. It needs to be emphasized that we did not want to solve the optimization problems but to analyze their structural properties. Hence, we proposed and successfully applied a graph building scheme which, in order to discover how the heuristic BH method performs its search, results in a series of (weighted) networks representing possible outcomes of BH run with different parameter setups.

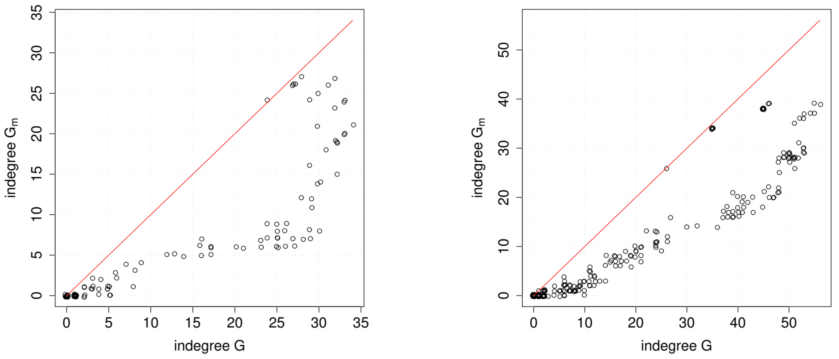
As future works we can outline two main directions. Based on the results shown in this paper, it is worth dealing with the development of an extension of the Basin Hopping method. That version would work as follows. During its run the algorithm



(a) $n = 2$

(b) $n = 3$

Fig. 11: Degree investigation of Levy8 networks; the points are jittered for better visibility

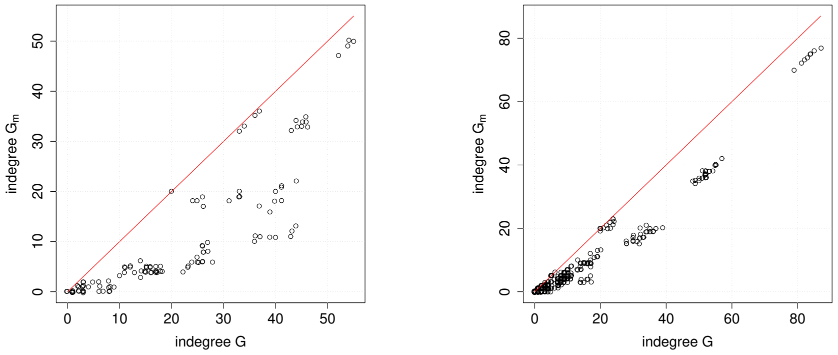


(a) $n = 2$

(b) $n = 3$

Fig. 12: Degree investigation of Ackley networks; the points are jittered for better visibility

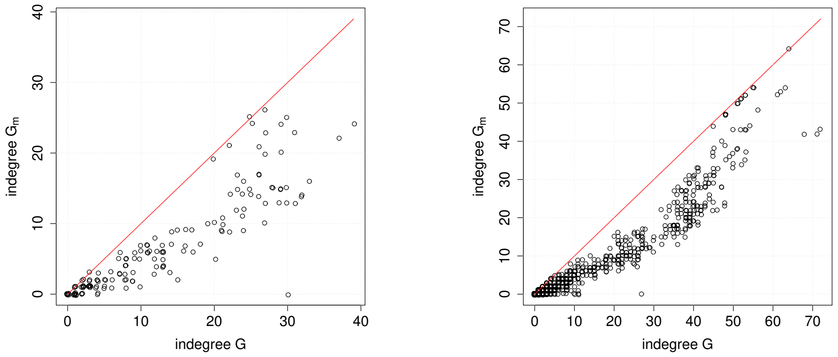
would build up the BHN representation of the global optimization problem. Using that network it would adaptively change its parameters (local stopping rule, direction of search, length of the jumps, acceptance criterion, etc) according to the characteristics of the BHN. For example, if it detects strong community structure in the network then the algorithm should make bigger jumps in the search space to discover further details. This and further techniques might result in a Basin Hopping approach which, although for a price of larger computational cost, would give higher level of guarantee that the best solution found is the real global minimum. This has particular relevance in case of multimodal optimization.



(a) $n = 2$

(b) $n = 3$

Fig. 13: Degree investigation of Rastrigin networks; the points are jittered for better visibility



(a) $n = 2$

(b) $n = 3$

Fig. 14: Degree investigation of Sinusoidal networks; the points are jittered for better visibility

Another line of research is to discover such network representations of global optimization problems which correspond to other optimization methods. Although many heuristic methods share similarities to BH, it would be interesting to see and compare the different graphs and develop benchmarking methodologies based on network science.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. T. Vinkó was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences.

References

1. Ackley D, A connectionist machine for genetic hillclimbing. Vol. 28. Springer Science & Business Media, 2012.
2. Albert R, Barabási A-L, Statistical mechanics of complex networks, Reviews of Modern Physics 74:47–97 (2002)
3. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, 10, P10008, 2008
4. Brin S, Page L, The anatomy of a large-scale hypertextual Web search engine, Computer Networks, **30**:107–117 1998
5. Cho H, Olivera F, Guikema SD, A derivation of the number of minima of the Griewank function, Applied Mathematics and Computation, 204, 694–701(2008).
6. Daolio F, Tomassini M, Vérel S, Ochoa G, Communities of minima in local optima networks of combinatorial spaces, Physica A: Statistical Mechanics and its Applications, 390, 1684–1694 (2011)
7. Dolan ED, Moré JJ, Benchmarking optimization software with performance profiles, Mathematical Programming, 91:201–213 2002
8. Doye JPK, The network topology of a potential energy landscape: A static scale-free network, Phys. Rev. Lett. 88, 238701 (2002)
9. Erdős P, Rényi A, On Random Graphs, Publicationes Mathematicae 6:290–297 (1959)
10. Fourer R, Kernighan BW, AMPL: A Modeling Language for Mathematical Programming, Duxbury Press, 2002.
11. Freeman L, A set of measures of centrality based on betweenness, Sociometry **40**:35–41 1977
12. Griewank AO, Generalized descent for global optimization, Journal of Optimization Theory and Applications 34, 11–39 1981
13. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
14. Leary RH, Global optimization on funneling landscapes, Journal of Global Optimization **18**:367–83 2000
15. Levy AV, Montalvo A, Gomez S, and Calderon A, Topics in Global Optimization, Lecture Notes in Mathematics, No. 909, Springer-Verlag, Berlin, 1981
16. Locatelli M, A Note on the Griewank Test Function, Journal of Global Optimization **25**:169–174, 2003
17. Locatelli M, On the Multilevel Structure of Global Optimization Problems, Computational Optimization and Applications, 30, 5–22 (2005)
18. Locatelli M, Schoen F, Global Optimization: Theory, Algorithms and Applications. SIAM-MOS, Philadelphia (PA), USA, (2013)
19. Locatelli M, Maischberger M, Schoen F, Differential evolution methods based on local searches. Computers & Operations Research, **43**:169–180, 2014
20. Mittelmann H, Benchmarks for Optimization software `http://plato.asu.edu/bench.html`
21. More JJ, Munson TS, Computing mountain passes, Preprint ANL/MCS-P957- 0502 2002
22. Murtagh BA, Saunders MA, MINOS 5.51 User's Guide, Technical Report SOL 83-20R, 2003
23. Neumaier A, Complete search in continuous global optimization and constraint satisfaction, Acta Numerica 13(2004), 271–369.
24. Neumaier A, Shcherbina O, Huyer W, Vinkó T, A comparison of complete global optimization solvers, Mathematical Programming 103(2005), 335–356.
25. Newman MEJ, Networks: An Introduction, Oxford University Press, 2010
26. Newman MEJ, A measure of betweenness centrality based on random walks, Social Networks **27**:39–54 2005.
27. Newman MEJ, Modularity and community structure in networks, PNAS 103:8577–8696 2006
28. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D, Defining and identifying communities in networks, PNAS 101 (9) 2658–2663 2004.
29. Rios LM, Sahinidis NV, Derivative-free optimization: A review of algorithms and comparison of software implementations, Journal of Global Optimization, 56(2013), 1247–1293.
30. Scala A, Amaral L, Barthélémy M, Small-world networks and the conformation space of lattice polymer chains, Europhys. Lett., **55**:594–600 2001
31. Stillinger FH, Weber TA, Packing structures and transitions in liquids and solids, Science 225 (1984), 983–989.

32. Storn R, Price K, Differential evolution - a simple and efficient heuristic for global optimisation over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.
33. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report, 2005005, 2005.
34. Tomassini M, Verel S, Ochoa G, Complex-network analysis of combinatorial spaces: The NK landscape case. Physical Review E 78.6 (2008): 066114.
35. Torn A, Zilinskas A, Global optimization. Springer-Verlag New York, Inc., 1989.
36. Wales DJ, Doye JPK, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 Atoms, J. Phys. Chem. A, 101, 5111–5116 1997
37. Zabinsky ZB, Smith RL, Pure adaptive search in global optimization. Mathematical Programming, **53**:323–338 1992