Towards Population-based Fitness Landscape Analysis Using Local Optima Networks

Melike D. Karatas
M.D.Karatas@exeter.ac.uk
Department of Computer Science
University of Exeter
United Kingdom

Ozgur E. Akman
O.E.Akman@exeter.ac.uk
Department of Mathematics
University of Exeter
United Kingdom

Jonathan E. Fieldsend
J.E.Fieldsend@exeter.ac.uk
Department of Computer Science
University of Exeter
United Kingdom

ABSTRACT

A fitness landscape describes the interaction of a search domain, a cost function on designs drawn from the domain, and a neighbourhood function defining the adjacency of designs — induced by the optimisation method used. Fitness landscapes can be represented in a compact form as Local Optima Networks (LONs). Although research has been conducted on LONs in continuous domains, the majority of work has focused on combinatorial landscapes. LONs are often used to understand the landscape encountered by population-based search heuristics, but are usually constructed via point-based search. This paper proposes the first construction of LONs by a population-based algorithm, applied to continuous optimisation problems. We construct LONs for three benchmark functions with well-known global structure using the widely used Nelder-Mead downhill simplex algorithm, and contrast these to the LONs from a point-based approach. We also investigate the sensitivity of the LON visualisation to the downhill simplex algorithm's hyperparameters, by varying the initial step size of the simplex and the step size for connectivity of optima. Our results suggest that large initial simplex sizes fragment the landscape structure, and exclude some local optima from the fitness landscape.

CCS CONCEPTS

• **Theory of computation** → **Design and analysis of algorithms**;
• **Computing methodologies** → *Continuous space search.*

KEYWORDS

Fitness Landscapes, Local Optima Networks, Population-based Optimisation Algorithms, Continuous Optimisation, Nelder-Mead Downhill Simplex Algorithm

ACM Reference Format:
Melike D. Karatas, Ozgur E. Akman, and Jonathan E. Fieldsend. 2021. Towards Population-based Fitness Landscape Analysis Using Local Optima Networks. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449726.3463170

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463170

1 INTRODUCTION

The fitness landscape of a problem represents the search space structure of an optimisation problem induced by a search algorithm. It is defined by the candidate solutions of the problem together with their fitness values and a neighbourhood structure that designates the connectivity among these candidate solutions. Formally, it is defined as a triple (X, N, f), where X is a set of potential solutions, *i.e.* the search space, $N : X \rightarrow 2^X$ is a neighborhood structure that assigns to every $\mathbf{x} \in X$ a set of neighbors $N(\mathbf{x})$, and $f : X \rightarrow \mathbb{R}$ is the fitness function [28]. In the evolutionary computation field, heuristic search methods are commonly used to search for the point in the search space which, without loss of generality, minimises the response of the fitness function. The Local Optima Network (LON) model was proposed in [24] as a weighted and directed graph visualisation tool that compacts the search landscape by representing only local optima as network nodes (compressing other locations in the design space).

While local optima constitute the nodes of this network, graph edges account for possible transitions between the nodes (search). This technique facilitates the visualisation of the local optima structure, making it easy to see their number, distribution and connectivity in the underlying landscape. Traditionally, LON construction has been derived from a point-based view of search and neighbourhoods — however, this potentially limits the insight that can be gained for landscapes induced by population-based search, which is an underlying property of evolution-based optimisation heuristics (*e.g.* genetic algorithms). Recently, the search trajectory networks were introduced as a way to visualise and analyse population-based algorithms [23]. However, this is an approach to representing the search progress, rather than visualising the global structure and the distribution of local optima in the search space. As a first step toward population-based LONs, we derive and investigate the LONs generated via the well-known Nelder-Mead downhill simplex algorithm [22], which is a population based search, but whose performance is deterministic. We also investigate the interplay between the algorithm hyperparameters (*e.g.* the initial simplex hypervolume) and the optimisation task. To explore the visualisations generated with our approach, we consider a set of benchmark functions with well-known landscape connectivity patterns (global structure) that have been explored in previous studies on constructing LONs for continuous search domains [1]. We also exploit previous work on visualising funnel structures through monotonic LONs [12, 25].

Our results suggest that the landscape's global structure strongly varies with the perturbation step-size — which determines the initial simplex (hyper)volume. Additionally, the LONs generated via the Nelder-Mead optimisation are noticeably different from

those created using point-based gradient search (the Monotonic Sequence Basin-Hopping Algorithm [16]), as used in recent work on LONs for continuous domains [1].

The paper proceeds as follows. In Section 2 we set out the fundamental definitions to understand the standard LON construction. Section 3 details our approach for population-based LON construction using the Nelder-Mead downhill simplex algorithm. Section 4 details the configuration of our experiments, with Section 5 providing analysis and visualisations of the point-based and population-based search LONs. The work concludes with Section 6, which outlines some potential future directions of the work, building on the results presented here.

2 DEFINITIONS & ALGORITHMS

We now present the definitions needed to understand the construction of LONs, and the basic algorithm for population-based LON construction using the Nelder-Mead downhill simplex method.

2.1 Formal Definitions

Fitness Landscape. A triple (X, N, f) where X is a set of potential solutions, *i.e.* the search space, $N : X \rightarrow 2^X$ is a neighborhood structure that assigns to every $\mathbf{x} \in X$ a set of neighbors $N(\mathbf{x})$, and $f : X \rightarrow \mathbb{R}$ is the fitness function. $\mathbf{x}$ has n elements.

Local Optima (L). The set of solutions of an optimisation problem, whose fitness is superior to all other solutions in their neighborhood. For example, in a minimisation problem, L is a local optimum if its fitness value is not worse than all other solutions in its neighborhood: $\forall x \in N(L) : f(x) \geq f(L)$, where $N(L)$ is the neighbourhood of L.

Edge (E). Search transitions among local optima are represented by directed and weighted edges. The weight w_{ij} of an edge from an optimum L_i to an optimum L_j represents the probability of the transition.

Basin of attraction. The basin of attraction B_i of a local optimum L_i in the search space X, is the set $B_i = \{\mathbf{x} \in X \mid optimiser(\mathbf{x}) = L_i\}$. The cardinality $|B_i|$ of this set gives the size of the basin of attraction of L_i.

Local Optima Network (LON). A directed and weighted graph $LON = (L, E)$, which compacts a fitness landscape by taking local optima in the search space as nodes and connecting these nodes with edges based on their transition as a result of search operators.

Monotonic Edges (ME). Transition from a local optimum into the basin of a non-deteriorating local optimum after a perturbation on the search operator.

Monotonic Local Optima Network (M-LON). $M\text{-}LON = (L, ME)$ is the graph comprised only of monotonic edges, *i.e.* where the nodes are local optima and and there is an edge $e_{ij} \in ME$, with weight w_{ij} between two nodes L_i and L_j if $f(L_i) \geq f(L_j)$.

Compressed Local Optima (CL). Optima obtained by merging sets of local optima with the same fitness value into a single node.

Compressed Edges (CE). The edges between two compressed local optima, which are generated by summing up the weights of the corresponding monotonic edges in M-LON for those optima.

Compressed Monotonic Local Optima Network (CM-LON). The graph $CM\text{-}LON = (CL, CE)$ where nodes are compressed local optima CL and edges are compressed edges CE, as defined above.

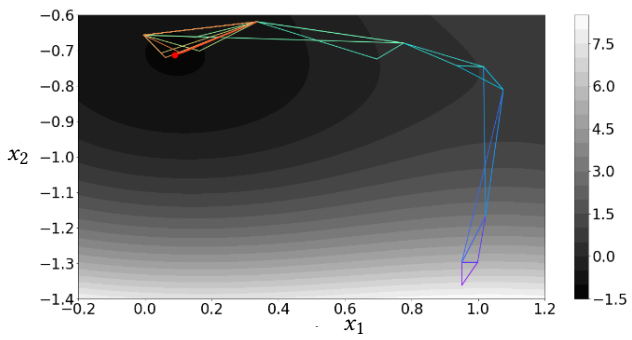


Figure 1: An example search trace, ending at a global optimum, on the Six-hump camel back function using the Nelder-Mead downhill simplex algorithm to minimise fitness. It starts with a simplex on the bottom right of the plot converging to a global optimum at (0.0898,-0.7126). The colourbar indicates fitness values, with regions of low fitness (high function values) coloured white and regions of high fitness (low function values) coloured black. The simplex colours indicate search progress, with increasing time represented by a shift from purple to red.

Funnels: Sets of local optima for which each element of the set terminates at the same local optimum (funnel sink). These can be viewed as basins of attraction at the local optima level.

2.2 Nelder-Mead Downhill Simplex Search

The Nelder-Mead downhill simplex algorithm is a widely-used direct search method for multidimensional continuous problems [22]. It uses the value of the objective function only, without requiring gradient information, and hence its implementation is straightforward in practice, especially when the problem function is non-differentiable or its derivative is difficult to compute. However, the method is not guaranteed to return the global optimum. We employ a particular algorithm variant in this paper [10]. Other variants of the Nelder-Mead downhill simplex method exist in the literature e.g., Geometric Nelder Mead [21], constrained optimization based on the simplex [19, 29], as well as hybrids with other evolutionary algorithms to improve convergence rate, *e.g. NM-PSO* [8].

The basic Nelder-Mead algorithm starts with a set of $(n + 1)$ points taken as vertices of a simplex in $\mathbb{R}^n$, *e.g.*, it is a triangle in $\mathbb{R}^2$, a tetrahedron in $\mathbb{R}^3$, and so on. Figure 1 is an example of how the simplex converges to a single optimum in the restricted region of the 2-dimensional six-hump camel benchmark function search space (see Table 1).

The Nelder-Mead algorithm converges quickly to optimal solutions in a deterministic fashion, given the initial simplex configuration and the hyperparameter settings [29]. Each iteration of the algorithm starts with a population, P, of size $(n + 1)$. A selection operator is then applied to eliminate one individual, thereby effectively choosing the best n individuals after sorting the individuals based upon their fitness. We refer the reader to [15] for a tie-breaking rule in sorting process. The discarded individual is replaced by a new individual added to the population through a

Table 1: Characteristics of the continuous benchmark problems used in the paper. All are minimisation functions (note that Six-hump camel is therefore inverted from its usual maximisation formulation).

Benchmark	Six-hump camel	Ackley	Rastrigin
Function	$\left(4 - 2x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + (4x_2^2 - 4)x_2^2$	$-20\exp[-0.2(\sqrt{0.5(x_1)^2 + (x_2)^2})]$ $-\exp[0.5(\cos(2\pi x_1) + \cos(2\pi x_2))] + e + 20$	$10n + \sum_i (x_i^2 - 10\cos(2\pi x_i))$
Boundaries	$x_1 \in [-3, 3]$, $x_2 \in [-2, 2]$	$x_i \in [-32.768, 32.768]$	$x_i \in [-5.12, 5.12]$
# local optima	6	several local minima	several local minima
# global optima	2	1	1

reflection operator, which is given in (1):

$$\mathbf{x}_{reflection} = \mathbf{x}_{centre} + \alpha(\mathbf{x}_{centre} - P_{worst}) \quad (1)$$

This process includes finding the mean, $\mathbf{x}_{centre}$, of the best n individuals and a new individual is then added to the population, $\mathbf{x}_{reflection}$, by reflecting the discarded individual, P_{worst} through this mean, with weight given by the reflection parameter α. A common choice for the value of this parameter is $\alpha = 1$ [15]. Other implementations of the Nelder-Mead method have been proposed with adaptive parameters depending upon the dimension of the problem, see *e.g.* [10]. If $\mathbf{x}_{reflection}$ is a better solution than the best individual in the initial population, P_{best}, search is expanded in this promising direction by the expansion parameter γ in equation (2) (a standard choice is $\gamma = 2$):

$$\mathbf{x}_{expansion} = \mathbf{x}_{centre} + \gamma(\mathbf{x}_{reflection} - \mathbf{x}_{centre}) \quad (2)$$

The resulting new individual $\mathbf{x}_{expansion}$ is accepted in the population if it is a better solution than the current P_{best}. If $\mathbf{x}_{expansion}$ is a worse solution than the second worst solution in the population P_{second_worst}, the search operator to be used is *contraction*, which contracts the hypervolume by adjusting the parameter β given in (3):

$$\mathbf{x}_{contraction} = \mathbf{x}_{centre} + \beta(P_{worst} - \mathbf{x}_{centre}) \quad (3)$$

This parameter is usually taken to be $\beta = \frac{1}{2}$. If $\mathbf{x}_{contraction}$ is also worse than P_{second_worst}, the next attempt is to shrink all the points in the simplex towards the best solution at hand, P_{best}, by the shrink coefficient δ given in (4):

$$\mathbf{x}_{shrinkage} = P_{best} + \delta(\mathbf{x}_{shrinkage} - P_{best}) \quad (4)$$

The choice for this parameter is usually $\delta = \frac{1}{2}$. The simplex is gradually reduced while better points are found primarily inside the simplex. The search stops when the points converge on an optimum, when a minimum difference between evaluations is observed, or when a maximum number of function evaluations are performed.

The choice of the initial simplex is a crucial aspect of Nelder-Mead performance, *i.e.* if all vertices reside in the same hyperplane, search will not explore all dimensions of the search domain [32]. We now analyse the landscape induced by the Nelder-Mead algorithm on several well-known benchmark problems and conduct an experimental sensitivity analysis of the landscape to the initial simplex size.

3 POPULATION-BASED LON CONSTRUCTION

Population-based algorithms employ a set of points to search the design space in parallel. They have an advantage over single point-based search optimisers, which can be more susceptible to specific properties of the landscape structure. One of those advantages is that population-based approaches have the ability to search different parts of the fitness landscape simultaneously, thus mitigating search bias in the initial position or sampling process. They also provide a good balance between exploration and exploitation on the search landscape [26]. While multiple points can explore diverse regions, population members tend to densify or converge to each other in promising regions. Research has also shown that population-based algorithms are better at handling high-dimensional optimisation problems compared to single point-based approaches [4].

Many population-based metaheuristics exist in the literature [11]. These include genetic algorithms [20], ant colony optimization (ACO) [7], particle swarm optimization (PSO) [14] and artificial immune systems [9], to name but a few. As an initial step towards constructing LONs for these types of heuristic population-based search, we set out the process by which LONs may be generated using the widely used Nelder-Mead algorithm. This optimiser sits usefully on the spectrum between point-based and population-based approaches, as although it employs a population of designs in its search, the cardinality of the set is predetermined (as $n+1$), and it is deterministic in terms of the operations and population evolution, which makes reasoning about the neighbourhood tractable. To the best of the authors' knowledge, our study presents the first population-based fitness landscape analysis using LONs.

The pseudocode given in Algorithm 1 presents the steps required to extract the nodes and basin sizes of LONs when employing the Nelder-Mead method. After detecting local optima and their corresponding basin sizes, we construct connectivity between optima by defining a neighbourhood of simplices anchored at the optima. This is accomplished via pivoting a simplex with step size $p^\star$ through all combinations of additive and subtractive axis-aligned steps, whilst keeping the anchor point unchanged. This mimics the initialisation of a downhill search, where an initial point is chosen, and n copies are made, with an additive perturbation on a single design element in turn, to generate the first simplex. By choosing all additive and subtractive perturbation combinations possible, and applying them sequentially one design value at a time (*i.e.* changing one element of the anchor vector at a time to create the n other simplex vertices), all neighbouring simplices cover the same hypervolume, and the number of neighbours (simplex populations) is deterministic, though scales poorly with n (specifically 2^n). Each of the neighbouring simplices are used to initialise downhill searches, with edges (and corresponding weights) derived from their terminating optima. Algorithm 2 details this process, perturbing from the previously found optima in Algorithm 1.

Data: Dimension number, n, iteration number, m, initial simplex size parameter, p
Result: Approximated optima, L, basins of attraction, B
$L := \emptyset$;
$B := \emptyset$;
counter := 0 ;
while counter $< m$ **do**
  Start with a random initial vector of size $(1 \times n)$, assigned to the first simplex vertex $\mathbf{s}_0$.
  Create the remaining simplex vertices $S = \{\mathbf{s}_i\}_{i=0}^{n}$ based on $\mathbf{s}_0$, with the following:
  $i := 1$; // Index into next simplex vertex
  while $i < n + 1$ **do**
    $\mathbf{s}_i := \mathbf{s}_0$; // Copy base solution
    if $s_{0,i-1} \neq 0$ **then**
      $s_{i,i-1} := s_{0,i-1} + p \times s_{0,i-1}$;
    else
      $s_{i,i-1} := 0.00025$; // As $0 \times p = 0$, an offset used
    end
    $i := i + 1$; // Move on to next simplex vertex
  end
  $S = \{\mathbf{s}_i\}_{i=0}^{n}$;
  $\mathbf{x}_k$:= Nelder-Mead(S);
  if $\mathbf{x}_k \notin L$ **then**
    $L := L \cup \mathbf{x}_k$;
    $B_{\mathbf{x}_k} := S$;
  else
    $B_{\mathbf{x}_k} := B_{\mathbf{x}_k} \cup S$;
  end
  counter := counter + 1;
end
Merge all basins whose optima values are within some threshold distances in design space, and eliminate corresponding duplicate from L;

Algorithm 1: Nelder-Mead algorithm for approximating the local optima and basin size.

Data: Approximated optima, L, basins of attraction, B, dimension number, n, perturbation step-size from local optima, $p^\star$
Result: Edge weight list, w
perm := $(\{-1, 1\})_{|2^n|,|n|}$; // Initialise 2^n simplex masks
foreach $\mathbf{x} \in L$ **do**
  for $k \leftarrow 0$ **to** $2^n - 1$ **do**
    Create the simplex with a vertex on $\mathbf{x}$ following the below process :
    $\mathbf{s}_0 := \mathbf{x}$; // Anchor simplex at optima
    $i := 1$; // Index into next simplex vertex
    while $i < n + 1$ **do**
      $\mathbf{s}_i := \mathbf{s}_0$; // Copy anchor solution
      for $j \leftarrow 0$ **to** $n - 1$ **do**
        if $s_{i,j} \neq 0$ **then**
          $s_{i,j} := s_{0,j} + p^\star \times perm[k, j] \times s_{0,j}$;
        else
          $s_{i,j} := 0.00025$;
        end
      end
      $i := i + 1$; // Move on to next simplex vertex
    end
    $S = \{\mathbf{s}_i\}_{i=0}^{n}$;
    $\mathbf{x}_k$:= Nelder-Mead(S);
    if $\mathbf{x}_k \in L$ **then**
      $w_{\mathbf{x},\mathbf{x}_k} := w_{\mathbf{x},\mathbf{x}_k} + 1$; // Increment weight
    else
      $w_{\mathbf{x},\mathbf{x}_k} := 1$; // Initialise weight
    end
  end
end
$w := w/2^n$; // Convert counts to probabilities

Algorithm 2: Edge generation between local optima using the Nelder-Mead downhill simplex algorithm.

This construction has the advantage of having a countable number of neighbours within a continuous design space, and additionally, since one of the vertices of the initial simplex is on the initial local optima, search will only terminate at *another* optima if it possesses a higher quality. As such, our edge definition for Nelder-Mead corresponds to the Monotonic Edge definition in Section 2. Of course, for large n such a neighbourhood of populations could be sampled to give an approximate edge weight if enumerating them is infeasible, with the approximation error derived from the proportion of the 2^n neighbouring populations sampled.

4 EXPERIMENTAL SETTINGS

We generate our LONs by initially sampling the continuous design space. Specifically, here we complete 100 runs of Algorithm 1 to generate the local optima and corresponding basin sizes using the Nelder-Mead implementation provided in [13].

Initial simplex generation. The initial simplex is constructed by first choosing a starting point from the space through Latin Hypercube Sampling [18] and then adding the step size parameter, p, to each design dimension individually in turn, generating the other n vertices of the initial simplex. To prevent the generation of illegal simplex vertices, the domain sampled by LHS was limited to $[\mathbf{s}_{min}, \mathbf{s}_{max} - p]$, where $\mathbf{s}_{min}$ and $\mathbf{s}_{max}$ define the lower and upper bounds of the box constraints for the problem. The budget of function evaluations was set to 20000 in each run.

Sampling & convergence. Two solutions obtained through different algorithm runs in a continuous space are regarded as the same local optimum if they are close enough in the design space. In this study, the threshold for local optimum location value was set to 10^{-5} in all dimensions following the same approach in [1].

Benchmark Functions. Our experiments used three well-known continuous benchmark problems [17]: the Six-hump camel, Ackley and Rastrigin functions [27]. Detailed information about the benchmarks used is provided in Table 1.

4.1 Parameters

Initial simplex step size (p). This is the parameter used in the initial local optima and basin size search (*cf.* Algorithm 1). Typically the value of this parameter is fixed at $p = 0.05$. Here, we conducted sensitivity analysis to examine the effect of varying this parameter on the resulting landscapes. Sensitivity studies of the algorithm parameters have been conducted in previous studies, *e.g.* [8, 31], showing that a large initial simplex size allows more flexibility to move in the design space and hence convergence to the true optimum, when the value of p is 0.5 or greater.

Perturbation step size for connectivity ($p^\star$). This parameter determines the size of the perturbation used to define the simplex neighbourhood, when defining edges (*cf.* Algorithm 2). This parameter was also varied to quantify its effect on the search landscape (though $p \leq p^\star$ was ensured).

4.2 Metrics

One of the advantages of visualising landscapes through LONs is that it allows us to use complex network metrics [30]. The metrics used in this paper are the following:

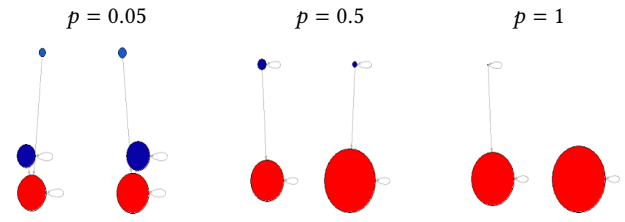
- The number of optima found, $\bar{n}$
- The mean density of the network, $\overline{D}$ (which shows the ratio of actual edges in the network to all possible edges in the network)
- The mean incoming degree, $\overline{d}_{in}$ (the total number of incoming edges in the graph)
- The mean degree centrality, $\overline{d}_{centrality}$, (a measure for finding the importance of a node in a network)
- The mean degree assortivity coefficient, $\overline{r}$, which is the Pearson correlation coefficient of degree between pairs of linked nodes (which quantifies the tendency of nodes being connected to similar nodes in a complex network)

5 RESULTS

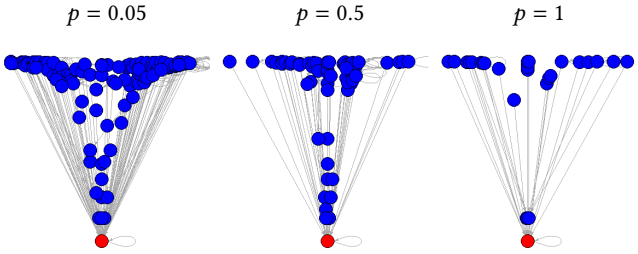
The experiments in this paper have been conducted to understand: (i) the sensitivity of the landscape to the initial perturbation size, as quantified by the parameter p; (ii) the sensitivity of the connectivity between local optima to the perturbation size, as quantified by the parameter $p^\star$; (iii) the global structure of the benchmark functions we have extracted in terms of the corresponding funnels; (iv) the differences between a single point construction, *i.e.* Basin-Hopping using L-BFGS-B [1], and our population-based approach using Nelder-Mead on some well-understood continuous benchmark problems. Code is available at http://pop-project.ex.ac.uk/.

5.1 Sensitivity to the initial simplex size

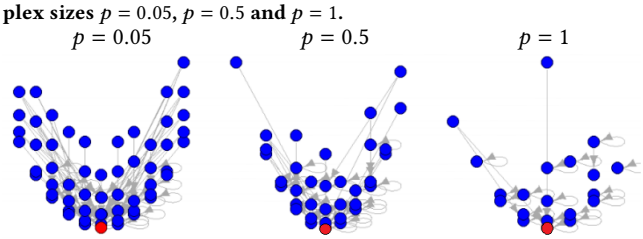
To see the effect on the landscape of varying the initial simplex size, we have used three different step sizes: $p = 0.05$, $p = 0.5$ and $p = 1$. The visualisations are presented in Figure 2. The Six-hump camel back benchmark function has six optima, two of which are global optima. As can be seen in Figure 2(a), with the default initial simplex step size $p = 0.05$, all local optima appear in the corresponding LON. However, increasing p leads to two of the local optima disappearing from the induced landscape (and the basins of attraction of the remaining two non-global optima being significantly reduced). Similarly, Figures 2(b) and 2(c) show a similar pattern for the Ackley and Rastrigin benchmark problems, with larger p values leading to more local optima being excluded from the LON. In particular, increasing p from 0.05 to 1 in the Rastrigin function leads to more than half the local optima being lost.



(a) Six-hump camel benchmark function with two variables and initial simplex sizes $p = 0.05$, $p = 0.5$ and $p = 1$.



(b) Ackley benchmark function with two variables and initial simplex sizes $p = 0.05$, $p = 0.5$ and $p = 1$.

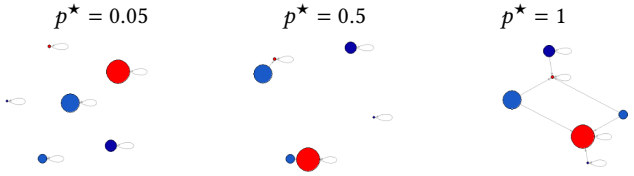


(c) Rastrigin benchmark function with two variables and initial simplex sizes $p = 0.05$, $p = 0.5$ and $p = 1$.

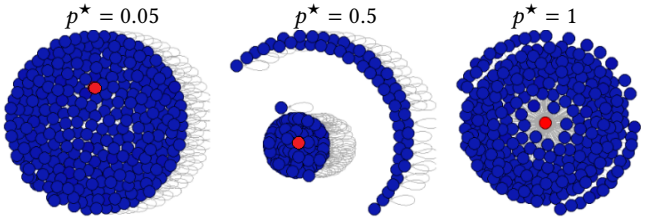
Figure 2: Local optima networks for the three benchmark problems as a function of the initial simplex size p. Red nodes indicate global optima, while blue nodes show local optima. The size of each node is proportional to the size of its basin of attraction. Edges are weighted based on the probabilities of transition among optima.

5.2 Sensitivity to the perturbation step-size

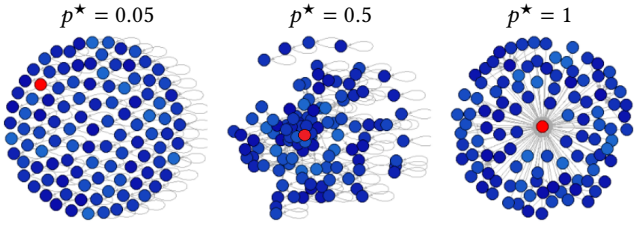
The effect of the connectivity perturbation size $p^\star$ was assessed by increasing its value from 0.05 to 1. It can be seen from Figure 3(a) that for the Six-hump camel benchmark function, there is no connectivity between optima for $p^\star = 0.05$. As $p^\star$ is increased, however, connections become established between local and global optima, and when $p^\star = 1$, the edges enable the search to jump from a given local optimum to another with a better fitness value consistently. In Figure 3(b), it can be seen that for the Ackley benchmark function, local optima only have self-directed edges when



(a) Six-hump camel benchmark function with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.



(b) Ackley benchmark function with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.

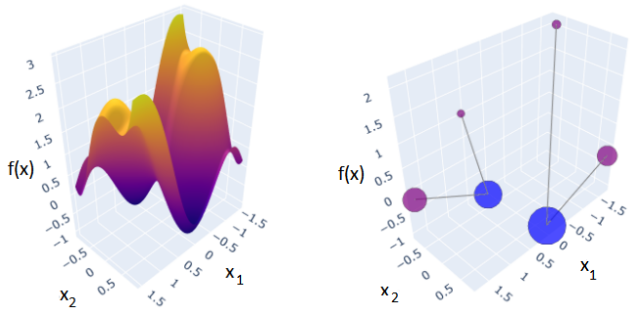


(c) Rastrigin benchmark function with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.

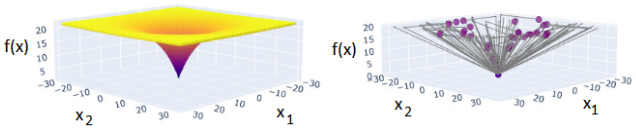
Figure 3: Local optima networks for the three benchmark problems as a function of the connectivity perturbation size $p^\star$. Red nodes indicate global optima, while blue nodes show local optima. The size of each node is proportional to the size of its basin of attraction. Edges are weighted based on the probabilities of transition among optima.

$p^\star = 0.05$. Increasing $p^\star$ to 0.5 leads to the local optima becoming clustered, and when $p^\star = 1$, jumps from the local optima to the global optimum (in the centre) become possible. Similarly, Figure 3(c) shows that as $p^\star$ is increased from 0.05 to 1 for the Rastrigin function, whilst some local optima progressively connect to other local optima, the majority of local optima appear to become connected to the global optimum.

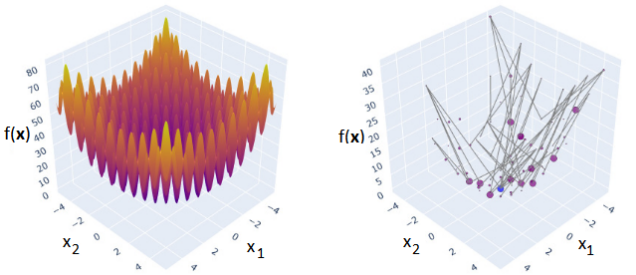
Figure 4 shows 3-D surfaces of the three benchmark functions in 2 dimensions, together with the corresponding LONs obtained for connectivity perturbation sizes of $p^\star = 0.57$ (Six-hump camel), $p^\star = 0.42$ (Ackley) and $p^\star = 0.13$ (Rastrigin), with the initial simplex size p fixed at its default value of 0.05 in each case. These connectivity perturbation sizes were chosen by a sampling process where 50% of search initialised with simplices anchored at the local optima could escape the original basin (estimated from the average of 100 runs), mimicking the procedure in [1] for specifying the neighbourhood size for perturbed starting locations of L-BGFS-B.



(a) Fitness landscape and LON for the Six-hump camel benchmark function with two variables, $p^\star = 0.57$ and $p = 0.05$.



(b) Fitness landscape and LON for the Ackley benchmark function with two variables, $p^\star = 0.42$ and $p = 0.05$.

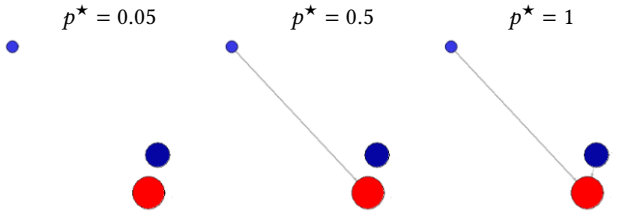


(c) Fitness landscape and LON for the Rastrigin benchmark function with two variables, $p^\star = 0.13$ and $p = 0.05$.

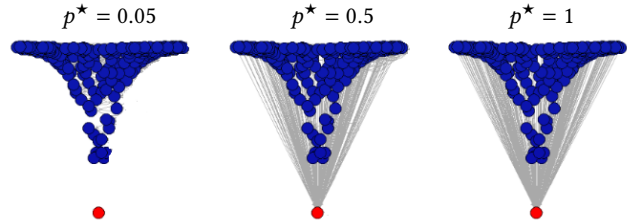
Figure 4: Fitness landscapes and corresponding local optima networks for the three benchmark functions. Blue nodes indicate global optima, while purple nodes show local optima. The size of each node is proportional to the size of its basin of attraction.

5.3 Funnel Structures

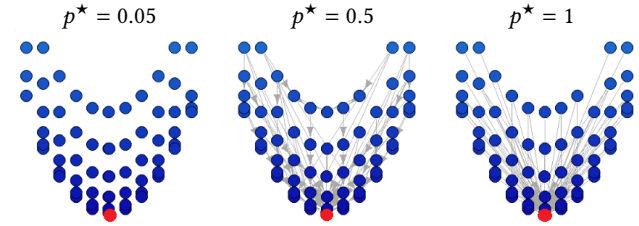
The funnel structures were extracted and are shown in Figure 5. The number of local optima in Figure 5(a) for the Six-hump camel benchmark function were combined into 3 local optima, which have the same fitness. In this case we can see that when the connectivity perturbation step size is chosen to be 1 the chained search always reaches the global optimum. For the Ackley function, Figure 5(b) shows that no edges are obtained with $p^\star = 0.05$, although some edges are occurred when $p^\star$ is increased to 0.5. Finally, with $p^\star = 1$, all of the edges leaving local optima eventually terminate at the global optimum. Figure 5(c) demonstrates that for the Rastrigin benchmark, increasing $p^\star$ yields a similar smooth progression between a graph with no edges to a graph in which all local optima connections end up with the global optimum.



(a) Six-hump camel back benchmark with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.



(b) Ackley benchmark function with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.



(c) Rastrigin benchmark function with two variables and connectivity perturbation sizes $p^\star = 0.05$, $p^\star = 0.5$ and $p^\star = 1$.

Figure 5: Funnel structures derived from the local optima networks of the three benchmarks, as a function of the connectivity perturbation size $p^\star$, with a fixed initial simplex size $p = 0.05$. Red nodes indicate global optima, while blue nodes show local optima. Node size is proportional to the size of its basin of attraction. Edges are weighted based on the probabilities of transition among optima.

5.4 Comparison with Basin-Hopping using L-BFGS-B

Network metrics were applied to compare the LONs extracted using a single point Basin-Hopping [1] algorithm with our population-based Nelder-Mead downhill simplex method, for the three benchmark problems. Although the sampling procedures used by the algorithms are different, we started with the same 100 randomly-selected initial points to extract the nodes in both cases. For the Basin-Hopping (BH) optimiser, the perturbation step sizes used were: 0.43 (Six-hump camel), 0.45 (Ackley) and Rastrigin (0.47). For the Nelder-Mead (NM) algorithm, the initial simplex size p was fixed at 0.05 for all benchmarks, while the connectivity perturbation sizes were as follows: $p^\star = 0.65$ (Six-hump camel), $p^\star = 0.52$ (Ackley) and $p^\star = 0.24$ (Rastrigin). Basin-Hopping uses an iterative sampling method to check connectivity, *i.e.* the run ends when no better optima are found after 1000 iterations. Nelder-Mead based edge generation runs the downhill optimiser from 2^n neighbouring simplices; there is no checking step in this case because the vertices of a neighbouring simplex is always anchored at the current local optimum, so the returned solution is never worse in quality. As summarised in Table 2, the results show that the iterative sampling procedure used for edge construction in Basin-Hopping generates a larger number of local optima. By contrast, Nelder-Mead generates fewer optima; for example, the algorithm generates at most four different local optima during edge search in the $n = 2$ design space case. Futhermore, Nelder-Mead constructs much denser networks than Basin-Hopping, as evidenced by the larger mean network densities, D.

Figure 6 compares the qualities of optima obtained using NM and BH when the algorithms are initialised from the same points, prior to edge generation. For NM, the initial simplex size was fixed at the default value $p = 0.05$. It can be seen from Figure 6(a) that for the Six-hump camel function, NM found global optima 25% more often than BH. However, BH has a greater concentration of points below the diagonal in the correlation plot, indicating that on average it finds better quality optima than NM in this case. For the Ackley benchmark, we see that when NM finds the global optimum, BH tends to find lower quality optima that are clustered (see left-hand side of Figure 6(b)). By contrast, when BH finds the global optimum, the distribution of optima quality for NM is spread fairly uniformly (see lower side of Figure 6(b)). Figure 6(c) shows that BH located more global optima than NM for the Rastrigin benchmark. This may be because whilst $p = 0.05$ is a reasonable perturbation size for the 6-Hump camel benchmark, which has few local optima, it is fairly small for the Rastrigin function, and hence does not enable sufficient exploration of the space. Finally, for the Six-hump camel function, Figure 7 plots the basin sizes of six optima found after 10000 iterations of the algorithms, initialised from 100 randomly sampled points in the design space. NM hits the global optima more often than BH, as indicated by the larger basin sizes, shown in red on the lower side of the diagonal in the plot. Moreover, BH found larger basin sizes for lower quality optima, shown in yellow on the upper side of the diagonal in the plot.

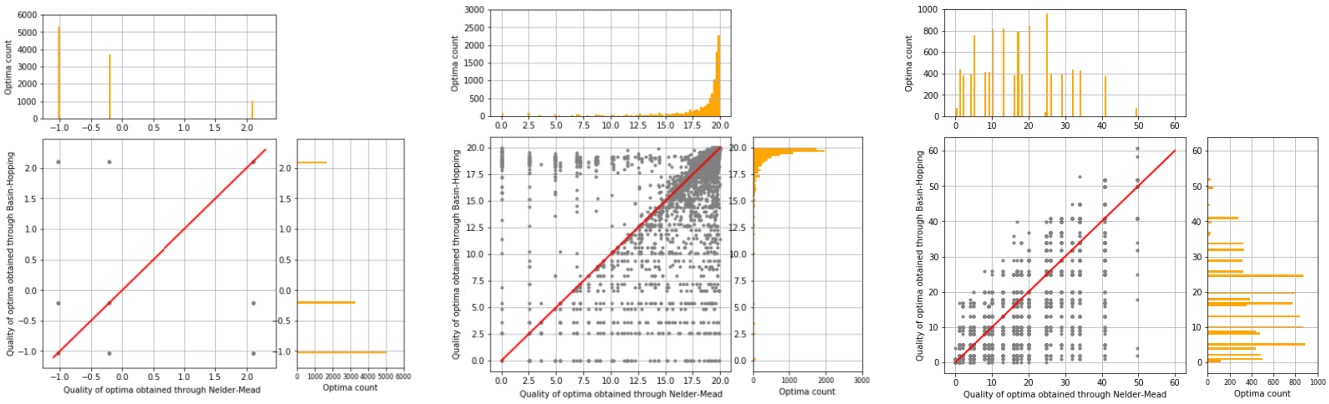
6 CONCLUSIONS & FUTURE WORK

This paper presents a new method for extracting local optima networks (LONs) from continuous fitness functions. LON construction was performed with a widely-used population-based optimisation algorithm: the Nelder-Mead downhill simplex method. This approach enabled us to compare the fitness landscapes generated using our population-based algorithm to those generated using the more established point-based approach. To facilitate this comparison, we applied our algorithm to a set of continuous optimisation benchmarks used previously in point-based studies [1].

Additionally, to gain insights into how optimisation hyperparameters affect the induced continuous fitness landscapes, we conducted computational experiments to assess how the LONs generated by our Nelder-Mead method were modified by changes to the size of the initial simplex and the size of the simplices used to generate edges. Our results indicate that the latter crucially effects the connectivity between local optima, with jumps from lower quality

Table 2: Network properties. Values are averages over 30 random instances, standard deviations are shown as subscripts. $\overline{n}$ represents the mean number of nodes, $\overline{D}$ is the mean density of the network, $\overline{d}_{in}$ is the mean incoming degree, $\overline{d}_{cent}$ is the mean degree centrality and $\overline{r}$ represent the mean degree assortivity coefficient.

Benchmark	Ackley (3-D)		Rastrigin (3-D)		Ackley (5-D)		Rastrigin (5-D)	
	BH	NM	BH	NM	BH	NM	BH	NM
$\overline{n}$	$2655_{29.376}$	$428_{13.873}$	$1235.33_{97.991}$	$444_{18.261}$	$2821_{8.405}$	$2116_{30.370}$	$10962_{3690.374}$	$3070_{57.852}$
$\overline{D}$	$0.0008_{0.000}$	$0.1047_{0.000}$	$0.001_{0.000}$	$0.003_{0.000}$	$0.0001_{0.000}$	$0.0004_{0.000}$	$3.41e^{-5}_{0.000}$	$0.0002_{0.000}$
$\overline{d}_{in}$	$0.543_{0.001}$	$0.689_{0.268}$	$1.763_{0.008}$	$1.461_{0.114}$	$0.533_{0.002}$	$1.001_{0.000}$	$0.437_{0.011}$	$0.659_{0.486}$
$\overline{d}_{cent}$	$0.0128_{0.001}$	$0.019_{0.003}$	$0.030_{0.002}$	$0.031_{0.001}$	$0.0131_{0.000}$	$0.015_{0.001}$	$0.004_{0.001}$	$0.009_{0.001}$
$\overline{r}$	$0.109_{6.205}$	$0.162_{0.090}$	$-0.266_{0.029}$	$-0.162_{0.063}$	0_{0}	$0.016_{0.003}$	$-0.017_{0.008}$	$-0.075_{0.118}$



(a) Six-hump camel function. **(b) Ackley function.** **(c) Rastrigin function.**

Figure 6: (a-c) Optima qualities obtained from the same 1000 initial points for the test problems considered.

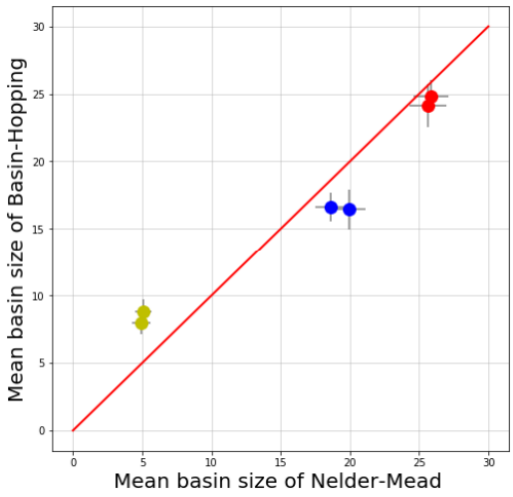


Figure 7: Mean basin sizes computed using Nelder-Mead and Basin Hopping for the Six-hump camel benchmark obtained after 10000 iterations from the same 100 initial points given to both algorithms. Horizontal and vertical error bars indicate the standard deviation for NM and BH, respectively. High and low quality optima represented by red and yellow points, respectively. There are two red points, representing two global optima with the same fitness value.

optima to higher quality optima becoming more probable as the perturbation step size is increased. Our findings also showed that the initial simplex size affects the convergence of the optimiser, with larger values increasing the basin of attraction of the global optima. This was consistent with previous studies [31, 32].

Future research includes using LON metrics to quantify how well LONs derived from the Nelder-Mead algorithm can predict the performance of population-based evolutionary optimisers (*e.g.* genetic algorithms) compared with point-based methods, such as iterated local search [5]. In addition, we will aim to extend the analysis performed here to a broader class of benchmark problems, including higher-dimensional functions, in order to assess the generalisability and scalability of our population-based method. We would also like to investigate the stability of LONs for both point-based and population-based approaches. In our limited investigation here, population-based construction tended to produce fewer basins, implying fewer samples may be needed to generate stable LONs. Finally, real-world applications will also be investigated, focusing on computational biology models of gene regulatory networks [2, 3], for which previous work has demonstrated the potential utility of landscape analysis in understanding optimiser performance [6].

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/N017846/1].

REFERENCES

[1] J. Adair, G. Ochoa, and K. M. Malan. 2019. Local optima networks for continuous fitness landscapes. In *Proc. GECCO 2019*. 1407–1414.

[2] O. E. Akman and J. E. Fieldsend. 2020. Multi-objective optimisation of gene regulatory networks: Insights from a Boolean circadian clock model. In *Proc. BICOB 2020*, Vol. 70. 149–162.

[3] O. E. Akman, S. Watterson, A. Parton, N. Binns, A. J. Millar, and P. Ghazal. 2012. Digital clocks: simple Boolean models can quantitatively describe circadian systems. *J. R. Soc. Interface* 9, 74 (2012), 2365–2382.

[4] Z. Beheshti and S. M. Shamsuddin. 2013. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl* 5, 1 (2013), 1–35.

[5] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini. 2012. Local optima networks and the performance of iterated local search. In *Proc. GECCO 2012*. 369–376.

[6] K. Doherty, K. Alyahya, O. E. Akman, and J. E. Fieldsend. 2017. Optimisation and landscape analysis of computational biology models: a case study. In *Proc. GECCO 2017*. 1644–1651.

[7] M. Dorigo, V. Maniezzo, and A. Colorni. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Syst.* 26, 1 (1996), 29–41.

[8] S-K. S. Fan and E. Zahara. 2007. A hybrid simplex search and particle swarm optimization for unconstrained optimization. *Eur. J. Oper. Res.* 181, 2 (2007), 527–548.

[9] J. D. Farmer, N. H. Packard, and A. S. Perelson. 1986. The immune system, adaptation, and machine learning. *Physica D* 22, 1-3 (1986), 187–204.

[10] F. Gao and L. Han. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.* 51, 1 (2012), 259–277.

[11] F. W. Glover and G. A. Kochenberger. 2006. *Handbook of Metaheuristics*. Vol. 57. Springer Science & Business Media.

[12] S. Herrmann, G. Ochoa, and F. Rothlauf. 2016. Communities of local optima as funnels in fitness landscapes. In *Proc. GECCO 2016*. 325–331.

[13] E. Jones, T. Oliphant, P. Peterson, et al. 2001. SciPy: Open source scientific tools for Python. (2001).

[14] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Proc. ICNN 1995*, Vol. 4. IEEE, 1942–1948.

[15] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. 1998. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. Optim.* 9, 1 (1998), 112–147.

[16] R. H. Leary. 2000. Global optimization on funneling landscapes. *J. Glob. Optim.* 18, 4 (2000), 367–383.

[17] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China. 2013. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. *Gene* 7, 33 (2013), 8.

[18] M. D. McKay, R. J. Beckman, and W. J. Conover. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 42, 1 (2000), 55–61.

[19] V. K. Mehta and B. Dasgupta. 2012. A constrained optimization algorithm based on the simplex search method. *Eng. Optim.* 44, 5 (2012), 537–550.

[20] M. Mitchell. 1998. *An Introduction to Genetic Algorithms*. MIT press.

[21] A. Moraglio and C. G. Johnson. 2010. Geometric generalization of the Nelder-Mead algorithm. In *Proc. EvoCOP 2010*. 190–201.

[22] J. A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Comput. J.* 7, 4 (1965), 308–313.

[23] G. Ochoa, K. M. Malan, and C. Blum. 2020. Search trajectory networks of population-based algorithms in continuous spaces. In *EvoApplications 2020*. 70–85.

[24] G. Ochoa, M. Tomassini, S. Vérel, and C. Darabos. 2008. A study of NK landscapes' basins and local optima networks. In *Proc. GECCO 2008*. 555–562.

[25] G. Ochoa, N. Veerapen, F. Daolio, and M. Tomassini. 2017. Understanding phase transitions with local optima networks: number partitioning as a case study. In *Proc. EvoCOP 2017*. 233–248.

[26] A. Prügel-Bennett. 2010. Benefits of a population: Five mechanisms that advantage population-based algorithms. *IEEE Trans. Evol. Comput.* 14, 4 (2010), 500–517.

[27] L. A. Rastrigin. 1974. Systems of extremal control. *Nauka, Moscow* (1974).

[28] P. F. Stadler. 2002. Fitness landscapes. In *Biological Evolution and Statistical Physics*. Springer, 183–204.

[29] T. Takahama and S. Sakai. 2005. Constrained optimization by applying the/spl alpha/constrained method to the nonlinear simplex method with mutations. *IEEE Trans. Evol. Comput.* 9, 5 (2005), 437–451.

[30] M. Tomassini, S. Vérel, and G. Ochoa. 2008. Complex network analysis of combinatorial spaces: The N-K landscape case. *Phys. Rev. E* 78, 6 (2008), 066114.

[31] P. C. Wang and T. E. Shoup. 2011. Parameter sensitivity study of the Nelder–Mead simplex method. *Adv. Eng. Softw.* 42, 7 (2011), 529–533.

[32] S. Wessing. 2019. Proper initialization is crucial for the Nelder–Mead simplex search. *Optim. Lett.* 13, 4 (2019), 847–856.