Subfunction Structure Matters: A New Perspective on Local Optima Networks

Sarah L. Thomson
Edinburgh Napier University
Edinburgh, Scotland, UK
s.thomson4@napier.ac.uk

Michal W. Przewozniczek
Wroclaw University of Science and Technology
Wroclaw, Poland
michal.przewozniczek@pwr.edu.pl

Abstract

Local optima networks (LONs) capture fitness landscape information. They are typically constructed in a black-box manner; information about the problem structure is not utilised. This also applies to the *analysis* of LONs: knowledge about the problem, such as interaction between variables, is not considered. We challenge this status-quo with an alternative approach: we consider how LON analysis can be improved by incorporating subfunction-based information — this can either be known a-priori or learned during search. To this end, LONs are constructed for several benchmark pseudo-boolean problems using three approaches: firstly, the standard algorithm; a second algorithm which uses deterministic grey-box crossover; and a third algorithm which selects perturbations based on learned information about variable interactions. Metrics related to subfunction changes in a LON are proposed and compared with metrics from previous literature which capture other aspects of a LON. Incorporating problem structure in LON construction and analysing it can bring enriched insight into optimisation dynamics. Such information may be crucial to understanding the difficulty of solving a given problem with state-of-the-art linkage learning optimisers. In light of the results, we suggest incorporation of problem structure as an alternative paradigm in landscape analysis for problems with known or suspected subfunction structure.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

fitness landscape analysis; local optima networks; combinatorial optimization; visualization

ACM Reference Format:
Sarah L. Thomson and Michal W. Przewozniczek. 2025. Subfunction Structure Matters: A New Perspective on Local Optima Networks. In *Genetic and Evolutionary Computation Conference (GECCO '25), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3712256.3726426

1 Introduction

In recent years fitness landscape analysis has increased in popularity [16], probably because of expanding interest in the notion of explainability [44] for optimisation algorithms. Local optima networks (LONs) [21] are a landscape analysis tool which have been successfully used to help explain optimisation dynamics across several domains [28, 36, 37, 41]. Typically, LONs are visualised and metrics are computed to achieve these aims. Many metrics have been proposed: complex network measurements such as node degrees, clustering coefficients, and assortativity [21], metrics related to neutrality and plateaus at the level of local optima [17], to fractal dimension [31], and to the notion of *funnels* [22] — among others. Something which has not been incorporated in the analysis of LONs is knowledge of problem structure: in particular, the existence of subfunctions. Many optimisation problems can be represented in a subfunction-based manner. For instance, the k-bounded problems [40] can be represented as a sum of subfunctions which take no more than k arguments. The subfunction structure can be known in advance [7, 43] or learned during search [9]. Previous works have constructed LONs for k-bounded problems such as MAX3SAT [20] and NK landscapes [4]; however, the measurements taken from the LONs did not consider subfunction structure which may be crucial to understand the effectiveness (or its lack) of state-of-the-art gray-box mechanisms [33, 35] and optimizers using linkage learning [24, 25]. Our main contribution in this paper is to address this gap. We construct LONs for well-known k-bounded problems and analyse them in a new way. Novel visualisations and metrics are proposed and used to compare algorithms and problem types. We find that incorporating subfunction structure into LON analysis is promising for bringing insight into optimisation dynamics on k-bounded problems.

2 Preliminaries

2.1 Optimisation problems

All problems used in this study have objective functions which are subject to *maximisation*. We now describe each of the problems in turn.

Deceptive functions. We consider problems whose solutions are represented by binary vectors of size n: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. Deceptive functions [7] were proposed to support tools for modelling hard artificial problems for binary search spaces. They are frequently used as benchmarks in the research concerning Genetic Algorithms [10, 25, 29]. We consider standard and bimodal deceptive functions of unitation. The standard deceptive function of order k is defined as:

$$trap_k(u) = \begin{cases} k-u-1 & , u < k \\ k & , u = k \end{cases} \quad (1)$$

where u is the sum of binary values (so-called *unitation*) of the function's argument. The standard deceptive trap function has one

GECCO '25, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1465-8/2025/07
https://doi.org/10.1145/3712256.3726426

local optimum (for $u = 0$) and one global optimum (for $u = k$). It may be considered hard to solve because starting from the random solution, any local search algorithm that greedily flips a single bit will converge to the suboptimal solution with $u = 0$. Such a local search will find the optimal solution only if it is randomly hit or starts from the solution whose unitation is $u = k - 1$, and the first flipped bit is the only '0' in such a genotype.

The bimodal deceptive function is defined as [8]

$$bimTrap_k(u) = \begin{cases} k/2 - |u - k/2| - 1 & , u \neq k \wedge u \neq 0 \\ k/2 & , u = k \vee u = 0 \end{cases} \quad (2)$$

Bimodal deceptive functions have two optimal solutions (all '0's and all '1's) and $\binom{k/2}{k}$ or $\binom{k/2+1}{k}$ of locally optimal solutions for even and odd k, respectively.

Problem structure. Deceptive functions can be assembled to create larger problems, e.g., they can be concatenated. In such a problem, subfunctions do not share any arguments, and each subfunction can be optimised separately. However, they can form more sophisticated problems if they share variables. Then, the subfunctions will *overlap*. The example of overlapping problems are cycling traps [43], where each deceptive function shares o variables with its left and right neighbours. In *conforming overlapping* problems, the optimal solution of each subfunction is a part of the globally optimal solution. In *conflicting overlapping* problems, the optimal solution of each subfunction is not a part of the globally optimal solution. The differences between conflicting and conforming overlapping problems are discussed in subsequent sections. More information can be found in [15, 26].

NK Landscapes. were designed to model heavily overlapping problems. In NK Landscapes, each variable forms the subfunction argument set with k subsequent variables. Thus, every subfunction takes $k + 1$ variables and each variable is an argument of $k + 1$ subfunction.

MAX-SAT. The maximum satisfiability problem (MAXSAT) is a real-world problem in which we are to satisfy clauses that consist of logical variables that may overlap. Any MAXSAT can be reformulated to MAX3SAT, where each clause contains three variables [40]. We consider artificial MAX3SAT instances and the generator employed in [10], using the typical clause-to-variable ratio of $cr = 4.27$.

2.2 Local Optima Networks

LON nodes. We consider a solution to be a *local optimum* lo_i if its fitness according to a function f is better than the fitnesses within its neighbourhood N. In practice, the neighbourhood is often sampled rather than fully enumerated — which is the case in this work. Formally: $\forall neigh \in SN(lo_i) : f(lo_i) > f(neigh)$ (assuming maximisation, as is the case for this study) where $SN(lo_i)$ is the *sampled* neighbourhood and $neigh$ is a particular neighbour. In a LON, the nodes are a set of local optima according to the definition just defined.

LON edges. Two local optima lo_i and lo_j have a LON edge traced between them under the condition that lo_j can be reached from applying random perturbation to lo_i and then subsequently local search [this type of edge has been termed an *escape* edges [37] in previous LON literature], and additionally $f(lo_j) \geq f(lo_i)$ (maximisation). This last criterion renders the edges *monotonic* in nature because they record only non-deteriorating, directed connections between local optima [22]. Edges have weights: the frequency of transition between the two local optima. That is: lo_j was reached through perturbation applied to lo_i followed by local search. The set of edges is denoted by E.

Local optima network (LON). We can now define a LON = (LO, E), which comprises nodes $lo_i \in LO$ i.e. the local optima, and edges $e_{ij} \in E$ between pairs of nodes lo_i and lo_j with weight w_{ij} iff $w_{ij} > 0$.

3 Related Work

3.1 Local optima networks

The first ever study on LONs [21] used NK landscapes as the domain, but did not consider subfunction structure in the analysis. The same is true for some subsequent studies on NK landscapes and LONs [13, 30]. Although LONs are usually constructed in a 'black-box' manner [22, 37]; that is, with no problem structure information used — there have been a few works which use grey-box optimisation to construct LONs. Chicano *et al.* [4] used greybox iterated local search with recombination to optimise high-dimensional NK landscapes and construct LONs. The LONs were visualised but not subject to subfunction analysis. Ochoa *et al* [20] also used greybox operators to construct LONs for MAX3SAT, but the metrics were 'black-box' LON measurements such as the number of local optima plateaus. Two recent studies [3, 27] extract LONs using greybox operators for PUBOi and NKQ problems, respectively. Although subfunction structure is not considered in the LON analysis, the papers show a trend and interest in the community towards utilising problem structure in landscape analysis.

3.2 Variable dependencies in evolutionary optimisation

Gray-box optimisation uses the *a-priori* available knowledge about problem structure to leverage its effectiveness and efficiency [39]. This includes taking advantage of subfunction-based function representation that may utilise the *additive form* defined as [40]:

$$f(\boldsymbol{x}) = \sum_{s=1}^{S} f_s(\boldsymbol{x}_{I_s}), \quad (3)$$

where I_s are subsets of $\{1, ..., n\}$, which can overlap (i.e., do not have to be disjoint) and S is the number of these subsets.

The additive form is convenient to represent the k-bounded problems, i.e., the problems that can be represented by a sum of subfunctions where each subfunction does not take more than k arguments [40]. Consider a situation in which the additive form of such a problem is known, which is the case in gray-box optimisation, and we have evaluated solution $\boldsymbol{x_a}$. Thus, we have computed the values of all subfunctions. By $\boldsymbol{x_a^g}$, we denote solution $\boldsymbol{x_a}$ with gth gene flipped. To evaluate $\boldsymbol{x_a^g}$, we do not have to evaluate all subfunctions. Knowing the subfunction values for $\boldsymbol{x_a}$ is enough to compute only those subfunctions that take x_g as an argument. This phenomenon is employed by *partial evaluations* and decreases the

computational costs of optimisation significantly [2, 42]. Thus, the fewer subfunctions are modified by one move, the cheaper it is to evaluate.

In gray-box optimisation, it is typical to consider non-linear dependencies, i.e., variables x_g and x_h are dependent if [18]:

$$f(\boldsymbol{x}) + f(\boldsymbol{x}^{g,h}) \neq f(\boldsymbol{x}^{g}) + f(\boldsymbol{x}^{h}) \tag{4}$$

where by $\boldsymbol{x}^{g}$, $\boldsymbol{x}^{h}$, and $\boldsymbol{x}^{g,h}$, are solutions obtained from $\boldsymbol{x}$ by modifying gene g, gene h or both of them, respectively.

Frequently, the non-linear dependencies arise from decomposing the optimised function using the *Walsh decomposition* [12]

$$f(\boldsymbol{x}) = \sum_{i=0}^{2^n-1} w_i \varphi_i(\boldsymbol{x}) \tag{5}$$

where $w_i \in \mathbb{R}$ is the ith Walsh coefficient, $\varphi_i(\mathbf{x}) = (-1)^{\mathbf{i}^{\mathrm{T}}\mathbf{x}}$ generates a sign, and $\mathbf{i} \in \{0,1\}^n$ is the binary representation of index i.

If there exists at least one nonzero Walsh coefficient w_i such that the gth and hth elements of $\mathbf{i}$ are equal to one, then variables x_g and x_h are non-linearly dependent [33]. In gray-box optimisation, the knowledge about variable dependencies is frequently represented by the Variable Interaction Graph (VIG) [6] that is a square matrix. VIG entry equals one if two variables are dependent and zero, otherwise. Gray-box operators frequently utilise this structure to obtain variation masks.

The Partition Crossover (PX) [35] is a recombination operator. It uses VIG to exchange only those variables that should be processed jointly. To mix two individuals $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$, PX removes from VIG all dependencies for which at least one variable meets the condition $\boldsymbol{x_a}(i) = \boldsymbol{x_b}(i)$, where $\boldsymbol{x}(i)$ is the value of the ith variable in $\boldsymbol{x}$. Then, PX joins variables in clusters using the remaining VIG dependencies. Each cluster can be used for mixing $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$. Consider a PX offspring individual $\boldsymbol{x_a}' = \boldsymbol{x_a} + PX_{mask}(\boldsymbol{x_b})$ created by inserting genes from $\boldsymbol{x_b}$ marked by a PX mask into $\boldsymbol{x_a}$. The key feature of PX is that individuals $\boldsymbol{x_a}'$ and $\boldsymbol{x_b}'$ will refer only to those subfunction argument subsets that exist in $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$ even if subfunctions overlap. If VIG concerns the non-linear dependencies, then it is guaranteed that $f(\boldsymbol{x_a}') + f(\boldsymbol{x_b}') = f(\boldsymbol{x_a}) + f(\boldsymbol{x_b})$. The above features make PX particularly useful in optimizing the overlapping problems [24].

VIG-based perturbation (VIGbp) [33] is a gray-box operator using VIG to produce variation masks. It randomly chooses a gene x_r and considers all genes that are dependent on x_r concerning VIG. If the number of genes dependent on x_r is equal or lower than a user-defined parameter α, then all genes are included by a mask. Otherwise, we randomly choose α genes from the x_r-dependent genes set. VIGbp is employed by the Iterated Local Search (ILS), which flips all genes marked by a mask and performs a local search on such perturbed genotypes. ILS aims to improve solutions step by step by improving only one part of a solution at a time. VIGbp perturbs a set of overlapping subfunctions instead of spreading the perturbation all over the genotype. VIGbp was shown to be highly effective in solving overlapping problems [33].

4 Problem structure and the expected LON

This section shows and analyses an example of an artificial problems and how LONs that describe its features should look. The objective is to present the key intuitions that will be helpful while considering the experimental results.

Consider $f_{sep}(x_0, x_1, ..., x_8) = f_1(x_0, x_1, x_2) + f_2(x_3, x_4, x_5) + f_3(x_6, x_7, x_8)$, where each $f_s(\boldsymbol{x}_{I_s}) = trap_3(u(\boldsymbol{x}_{I_s}))$. Subfunctions in f_{sep} are separable, i.e., each of the argument sets (0,1,2), (3,4,5), and (6,7,8) can be optimised separately. If we start from a random solution, in most of the cases, we will converge to solution 000 000 000. If we use **dependency-aware** ILS, e.g., ILS using VIGbp [33], to optimise f_{sep}, then we will variate a variable subset that refers to one subfunction (a single I_s in Formula 3). Thus, local optima that are worth showing in LON can be divided into four groups:

- **Group 1** consists of a single solution: 000 000 000.
- **Group 2** consists of three solutions: 111 000 000, 000 111 000, and 000 000 111.
- **Group 3** consists of three solutions: 111 111 000, 111 000 111, and 000 111 111.
- **Group 4** consists of a single optimal solution: 111 111 111.

All solutions in Group 2 are one **subfunction modification** away from the solution in Group 1: that is, the value of exactly one subfunction [from the three separable subfunctions] is different. Every solution in Group 2 is one modification away from **some** of the solutions in Group 3. Finally, all solutions in Group 3 are one modification away from the optimal solution in Group 4. This example shows that we can start from any solution, greedily optimise it, and (if we know the problem structure) it is enough to optimise only one subfunction at a time to get to the optimal solution. Thus, the LON we wish to get is presented in Figure 1.

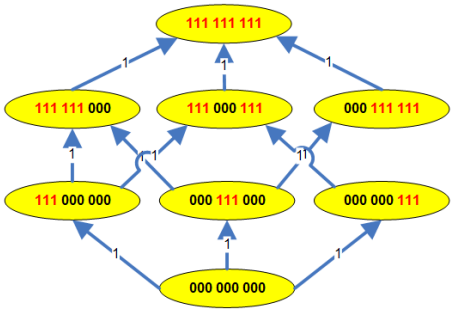


Figure 1: Decomposition-aware LON showing the number of modified subfunctions in each move

Consider an overlapping problem $f_{ovr}(x_0, ..., x_9) = f_1(x_0, ..., x_3) + f_2(x_3, ..., x_6) + f_3(x_6, ..., x_9)$, where each $f_s(\boldsymbol{x}_{I_s}) = bimTrap_4(u(\boldsymbol{x}_{I_s}))$. Each subfunction in f_{ovr} shares one or two variables with its neighbours. Thus, the modification of shared variables influences more than one subfunction. Optimal solutions to $bimTrap_4$ are 0000 and 1111. Locally optimal solutions contain two zeros and two ones, e.g., 0110 and 1001.

In Figure 2, we present a LON, which concerns two locally optimal solutions as its roots. The left LON part starts from the solution in which the arguments of each subfunction are 1001, while in the right part of a LON, the arguments of each subfunction are 0110. Same as in Figure 1, we consider the improvement of a single subfunction. The optimal solutions (built only from '1's or only from '0's) are located on the top of the two LON parts. In a jump from

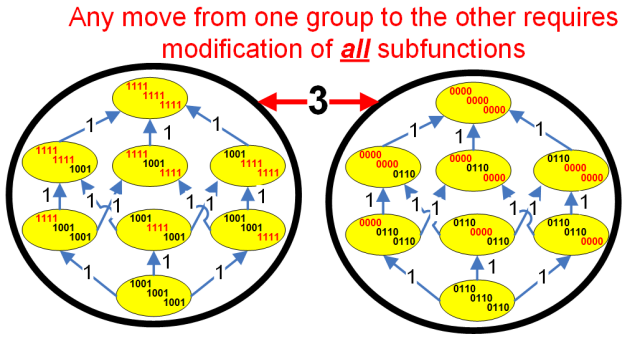


Figure 2: Decomposition-aware LON for overlapping problem

any solution on the left part of the LON to any solution on the right part of the LON, all subfunctions must be modified. Such a situation would not occur if the subfunction did not overlap. Thus, the overlaps have caused the creation of two separate attraction basins of the two global optima. Moving between these two attraction basins is hard because it requires the modification of all subfunctions. In the latter part of this work, we show that we can observe such separate attraction basins in the proposed decomposition-aware LONs obtained experimentally.

5 Methods

5.1 LON Construction

Traditional (black-box) ILS. Hereafter referred to as algorithm TRAD. In accordance with LON literature [17, 22, 32], we base the traditional construction algorithm on sampling using iterated local search (ILS). Several independent runs of ILS are executed; these follow cycles of perturbation followed by local search. In our case, the mutation operator is a single bitflip. Perturbation is three bitflips, and the local search is first-improvement — accepting strictly improving solutions. For the local optimum acceptance criterion, improving or equal local optima are accepted.

Deterministic recombination ILS. Hereafter referred to as algorithm PX. Instead of perturbation chaining together the local searches, partition crossover [as described in Section 3.2] is used. This algorithm is from a previous work [4] except we use ordinary first-improvement local search instead the specialised speedup hill-climber employed in that paper [they were optimising very high-dimensional problems]. The approach has been called deterministic recombination iterated local search (DRILS) in previous literature.

VIGp with FIHCwLL. Hereafter referred to as algorithm VIGP. Based on the approach in a previous work [34], this algorithm uses VIGp as described in Section 3.2 as the perturbation operator. We use first improvement hill climber with linkage learning (FIHCwLL) [25] to locally optimise solutions and learn dependencies simultaneously. FIHCwLL optimises a solution by flipping its genes. If a gene flip improves fitness, then it is preserved, or it is rejected. In each iteration, FIHCwLL considers all genes in a random order. Iterations are executed until at least one gene is modified during the preceding iteration. Fitness evaluations that are used for local optimisation are also used to perform variable dependency checks. Since we consider k-bounded problems that are additively decomposable, we use the non-linearity dependency check (Formula 4) that requires performing four fitness evaluations, i.e., $f(\boldsymbol{x})$, $f(\boldsymbol{x}^g)$, $f(\boldsymbol{x}^h)$, and $f(\boldsymbol{x}^{g,h})$. In FIHCwLL, three of them are the side-effect of the local search, and only one must be computed for linkage discovery purposes, which significantly reduces the cost of decomposition. FIHCwLL was inspired by [33].

5.2 LON metrics

Subfunction structure. We propose metrics relating to subfunction structure in LONs. These can be described as follows:

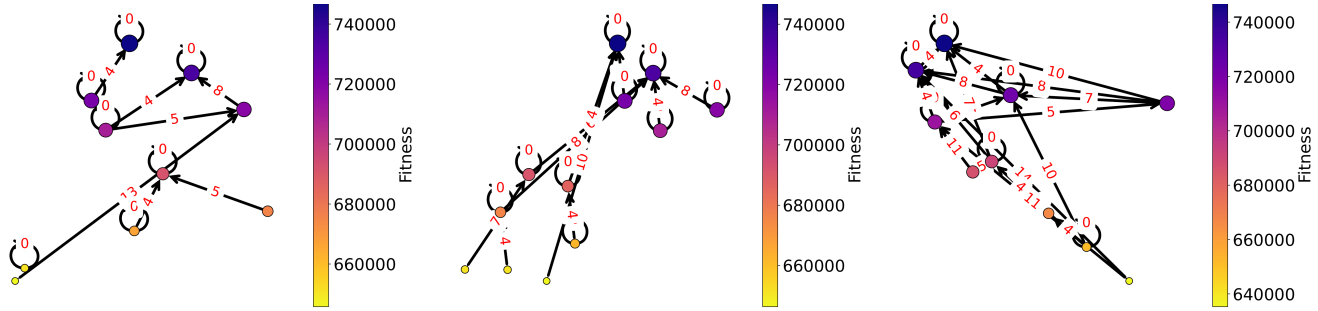
- number of subfunction changes associated with a LON directed edge between two local optima: that is, the number of subfunctions which changed when comparing the destination node to the source node
  - a LON typically contains several edges, so we consider the mean, median, and standard deviation of this metric across the network
- the number of positive subfunction changes in LON edges
  - for a decision problem such as MAX3SAT, this is the number of subfunctions which changed from unsatisfied in the source node to satisfied in the destination node
  - for an optimisation problem such as NK landscapes, this is the number of subfunctions which increased in value (assuming maximisation) from the source to the destination node
  - the mean, median, and standard deviation of this metric across a LON are considered
  - the number of negative subfunction changes in LON edges
    - this is the same as just described, except for negative changes to subfunction values

Previous metrics. In addition to the subfunction metrics, we also compute 14 LON metrics from previous literature: [14, 22, 32]. None of these aim at capturing subfunction structure, but we would nevertheless like to compare them and ascertain whether they capture different information. Two of the metrics relate to the number of local and global optima; two describe neutrality at the local optima level; nine are related to the notion of landscape *funnels* [basins of attraction at the local optima level]; and one considers the pagerank centrality of the global optimum.

6 Experimental Setup

6.1 Problem instance generation

We consider a set of well-known benchmarks that include NK-landscapes, MAX3SAT, and deceptive functions. All instances are of toy size: 15 bits for NK-landscapes and MAX3SAT, and between 15 and 18 bits for the deceptive functions due to varying subfunction structure setups which necessitate different problem sizes. Thus, it is easy to analyse the obtained LONs in terms of their expected and obtained features. For NK-landscape and MAX3SAT generation, we have adopted the software from [10] and generated 30 instances each. For the deceptive functions, there is no element of randomness in their construction. We consider 11 different problems of this type, with varying numbers of subfunctions and degrees of



(a) Algorithm TRAD **(b) Algorithm PX** **(c) Algorithm VIGP**

Figure 3: Local optima networks for an NK landscape of size 15 with $k = 2$. Node size, colour, and position on the y-axis is relative to fitness (maximisation). Position on the x-axis is obtained through multi-dimensional scaling of the binary solutions using Hamming distance

subfunction overlap. All instances used in this work are available in the supplemental material [see footnote 1].

6.2 LON Construction

For all three LON construction algorithms, there are 30 independent runs per problem instance. ILS runs terminate when there has not been a [strict] improvement to local optimum quality in 30 full cycles. These numbers are sufficient given the search space sizes under study, which comprise a maximum of 2^{18} = 262,144 solutions. LON construction and analysis are coded from scratch in Python.

6.3 Visualisation

The LON visualisations are implemented using the Python libraries NetworkX [11] and Matplotlib [1] For all network visualisations shown, the position of a node on the x-axis is the result of applying multi-dimensional scaling (MDS) to the local optima sample, using Hamming distance between the binary vectors as the measurement of distance. scikit-learn [23] is used for MDS and SciPy [38] for the distance computations.

7 Results

7.1 Network visualisation

Figure 3 shows local optima networks for an NK landscape instance constructed using the three different algorithms. Note that visualisations for all other algorithms and problems under study are available in the supplemental material [1]. In the Figure, each node is a local optimum and each edge is a directed transition between them. Node size, colour, and position on the y-axis are relative to fitness (maximisation). Position on the x-axis is the result of multi-dimensional scaling on the solution sample. The red numbers annotated onto edges capture the number of subfunctions which changed in the transition from the source local optimum to the destination local optimum.

Comparing the three sub-plots, we can immediately notice that there is a difference between Algorithm VIGP (Figure 3c) and the two others. While Algorithms TRAD and PX lend to rather sparse and simple networks, Algorithm VIGP leads to a denser view with more information. Notice that although all three reach the same fitness level, Algorithms TRAD identifies only a single edge leading towards the solution with this fitness. On the other hand, Algorithms PX and VIGP reveal three and four edges (respectively) directed towards the highest fitness level. Looking at the edge labels, it seems that these transitions require quite dramatic subfunction alterations in the solution: as many as 8-10 changes. It is interesting that these two algorithm were able to identify the dramatic changes which are needed to ascend.

[1] https://zenodo.org/records/14767046

7.2 LON subfunction metrics

Next we consider the distribution of metrics related to subfunction changes associated with edges of a LON. For every directed edge we assess which subfunctions changed in value; for those, we consider the number and *direction* of those changes: did the subfunction value increase (positive) or decrease (negative)? Figure 4 shows, for 30 MAX3SAT instances with a clause-to-variable ratio of 4.27: the mean, median, and standard deviation number of subfunction changes encoded in LON edges. Results are shown for the three construction algorithms, and indications of statistical difference betwween pairs of distributions are annotated, as described in the caption.

Looking across the plots in Figure 4, we can see that with respect to mean and median number of changes Algorithm VIGP appears to have significantly different distributions when compared to Algorithms TRAD and PX. From visual inspection of the boxes we can see that the Algorithm VIGP distributions are lower than those of the other two algorithms (that is, there are less subfunction changes in the directed edges between two local optima). Algorithms TRAD and PX have no significant difference in these two metrics. With respect to the standard deviation (the right-most plot), none of the algorithms differ significantly from one another.

Figure 5 has the same layout as Figure 4 but presents different metrics. Instead of the number of subfunction changes, these measurements consider the *direction*: that is, how many subfunctions a.k.a clauses changed from not satisfied to satisfied —- a positive change — and how many changed from satisfied to not satisfied — a negative change —- between the source and destination of a

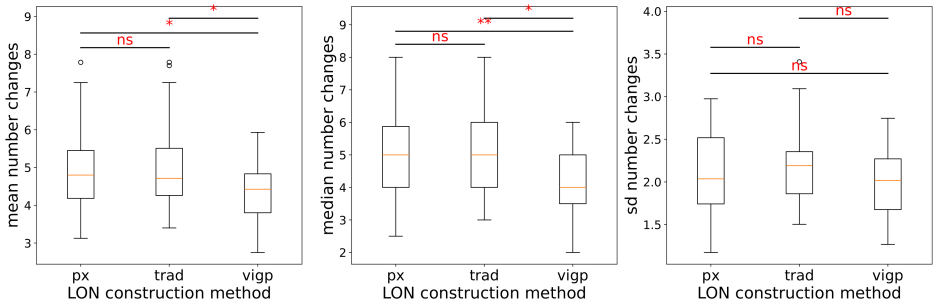


Figure 4: Distribution of LON edge subfunction metrics for 30 MAX3SAT instances [clauses-to-variable ratio 4.27]. Metrics consider the number of subfunctions which change in value between the start and end of LON edges [for improving edges only]. An indication of significant difference between pairs according to a Mann-Whitney test is annotated on the plots; ns means not significant; * means $p < 0.05$, ** means $p < 0.01$, * means $p < 0.001$**

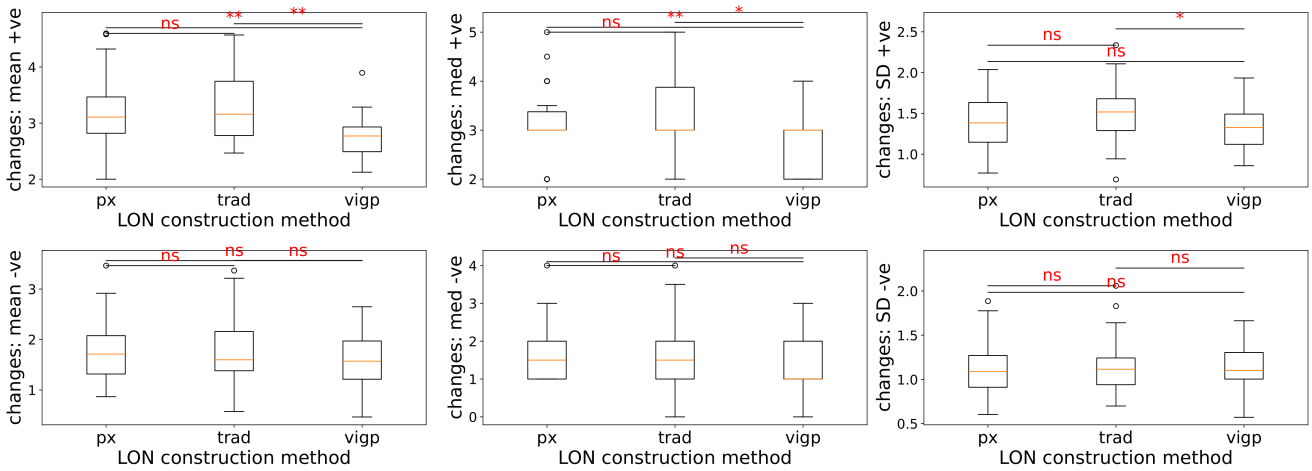


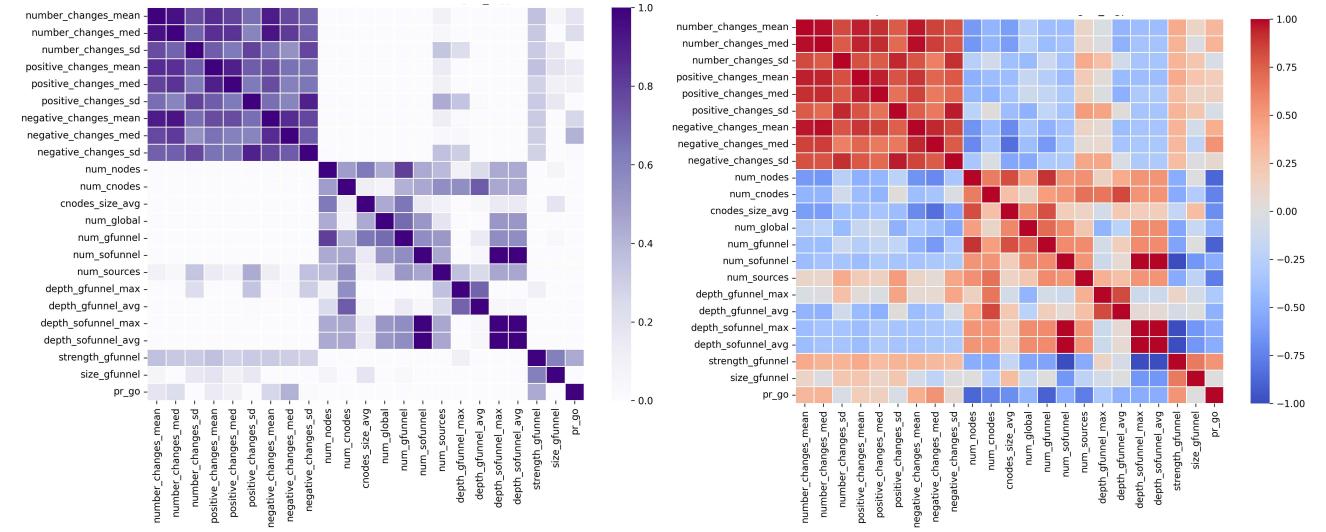
Figure 5: Distribution of LON edge subfunction metrics for 30 MAX3SAT instances [clauses-to-variable ratio 4.27]. The metrics consider the number of positive (+ve) and negative (-ve) changes between the start and end of LON edges [for improving edges only]. Indication of significant difference between pairs according to a Mann-Whitney test is annotated on the plots: ns means not significant; * means $p < 0.05$, ** means $p < 0.01$, * means $p < 0.001$**

LON edge. We notice that for the positive changes, there is significant difference between Algorithm VIGP and the two others with respect to the mean and median. Algorithm VIGP makes less positive changes; however, we can recall from Figure 4 that this algorithm makes less changes overall. Such an observation indicates that it makes smaller steps in terms of the number of the modified subfunctions. Such a feature may be considered important because VIGbp can be considered as making more precise steps. Therefore, VIGbp-generated LONs are more dense, and VIGbp visits more local optima than PX. Note that considering smaller variation masks that allow the improvement makes finding this improvement easier. Therefore, some studies focus on limiting the variation of mask sizes [5, 19]. There is no significant difference between the algorithms with respect to negative changes.

Plots for the other problems are available in the supplemental material [see footnote 1]. In the case of NK landscapes, Algorithm PX makes less subfunction changes than the other two algorithms, and this difference has statistical significance. Algorithms TRAD and VIGP make more positive changes than Algorithm PX — although this may be an artefact of the fact that they make more changes in general. In terms of negative subfunction changes, Algorithm VIGP makes more of them when compared with the other to algorithms [with indication of statistical significance]. The deceptive problem group, on the other hand, is not associated with significant differences between the three algorithms. This may be due to the low number of problems in the group [10, because one of the problems has LONs with no edges and therefore no edge subfunction metrics].

7.3 LON metric comparison

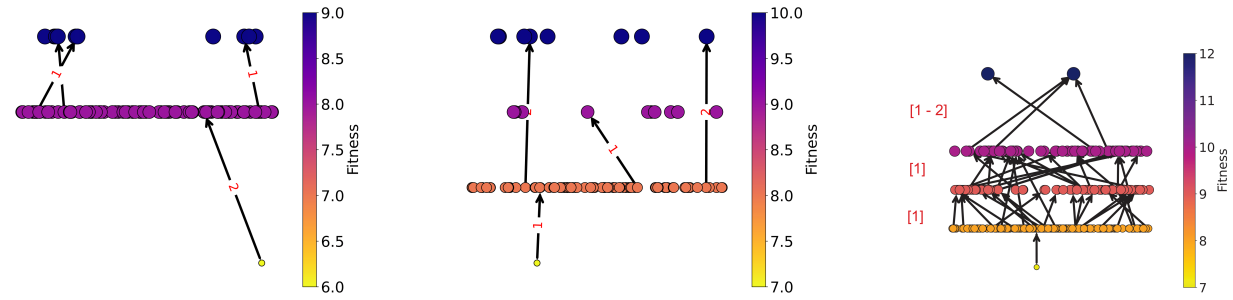
We would like to consider to what extent the LON subfunction metrics proposed here are similar to different, previously-proposed metrics for LONs. To this end, Figure 6 presents, for the deceptive



(a) Kendall-tau correlation

(b) Spearman correlation

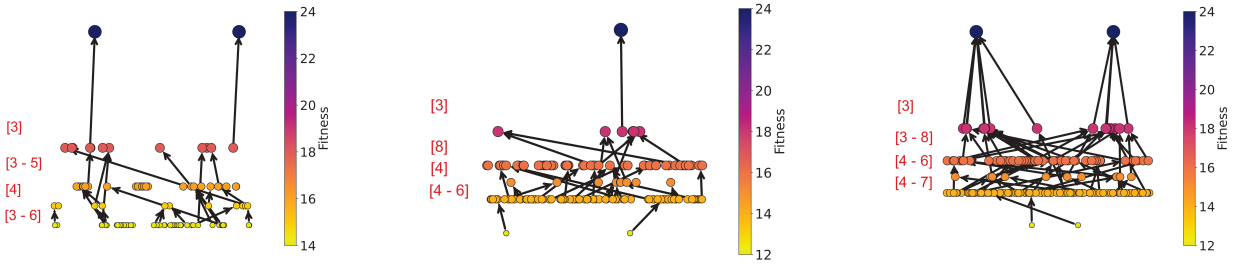
Figure 6: Similarity between LON metrics (algorithm VIGP) for deceptive functions. The proposed LON subfunction metric names contain the substring "changes"; all other metrics are from previous LON literature



(a) Subfunction overlap of 2; Algorithm TRAD

(b) Subfunction overlap of 2; Algorithm PX

(c) Subfunction overlap of 2; Algorithm VIGP



(d) Subfunction overlap of 4; Algorithm TRAD

(e) Subfunction overlap of 4; Algorithm PX

(f) Subfunction overlap of 4; Algorithm VIGP

Figure 7: Local optima networks for two deceptive problems with different degrees of subfunction overlap. Node size, colour, and position on the y-axis is relative to fitness (maximisation). Red text on an individual edge is the number of subfunction changes between the source and destination node. Red text in brackets indicates the range of subfunction changes needed to jump between the two fitness levels beside it. Position on the x-axis is obtained through multi-dimensional scaling of the binary solutions using Hamming distance

problem set, the Kendall-Tau correlation and also the Spearman correlation for pairs of LON measurements. LON metrics are named on each axis [the proposed subfunction metric names contain 'changes' and the colour in a square captures the similarity metric, as indicated in the colourbar.

Looking across Figure 6a, we can notice from looking along the rows associated with the subfunction metrics that while they have moderate-to-strong correlations with each other, they have only weak or very weak correlation to other LON metrics from the literature. The Spearman correlations in Figure 6b show that the subfunction metrics have moderate-to-strong correlations among themselves, and weak-to-moderate correlations with other LON metrics from the literature. In most cases this is a [weak] negative correlation, but there are a few metrics from previous works which show [weak] positive correlations with the subfunction measurements. Correlation matrices for the other problems and algorithms can be found in the supplemental material [footnote 1]; these showed similar trends to those just described.

7.4 Case study: subfunction overlap

To demonstrate the insights that can be gained using our approach, we consider a case study where we would like to analyse the difference in landscape structure when problems have a different degree of subfunction overlap. To this end, we choose two bimodal deceptive problems for comparison: one has subfunctions with a variable overlap of two, and the other has an overlap of four. LONs for both of them, constructed using the three algorithms, are presented in Figure 7. In these plots, red text on an individual edge represents the number of subfunction changes between the source and destination node. Red text in brackets indicates the range of subfunction changes needed to jump between the two fitness levels beside it.

Looking at the first row of plots, which relates to the problem with lower subfunction overlap, we can observe that only Algorithm VIGP found the global optimum fitness (which is 12). In fact, it identifies both of the global optima. The Algorithm TRAD and Algorithm PX LONs are very sparse — both of them have only four edges. On the other hand, Algorithm VIGP is a densely-connected network, with plenty of opportunities for search to transition between fitness levels. From the red text annotations, we can see that moving up one fitness level is associated with a low number of subfunction changes: typically one and maximum two. Thus, thanks to using smaller steps, it is easier for VIGP to traverse the network of local optima. This observation is coherent with the aforementioned research direction to eliminate dependencies irrelevant to optimisation and obtain shorter variation masks [5, 19]. Indeed, Algorithm VIGP can find a path to a global optimum from almost any initial state (e.g., Fig 7f). Oppositely, for TRAD- and PX-based LONS, the number of paths from a randomly chosen state to global optimum is low (e.g., Fig 7d) and 7d.

Looking now at the lower row of plots, which are for the instance with a higher degree of subfunction overlap. When compared to the problem with lower subfunction overlap, both Algorithm TRAD and Algorithm PX have an increased number of connections in Figure 7d and 7e. Comparing the top row of plots with the bottom row, we can notice that the problem with higher subfunction overlap has LONs with more dramatic subfunction changes between fitness levels. On the higher-overlap problem (Figures 7d-7f), all of the three algorithms reach at least one global optimum (fitness of 24). Algorithm PX reaches only one global optimum and there is a single connection towards it. Algorithms TRAD and VIGP reach both global optima, but we notice that the Algorithm VIGP LON contains several connections towards them, while the Algorithm TRAD LON has only two. Another interesting observation is that Algorithms PX and VIGP find connections which require eight subfunctions to change between the second and third-highest fitness levels. Algorithm TRAD did not find these options and this may be a reason for the sparse connectivity of its LON in relation to the global optima.

8 Conclusion

We have considered whether including problem structure information into local optima network (LON) construction and analysis can bring additional insights into optimisation dynamics. Three well-known k-bounded problem domains were included in the analysis: MAX3SAT, NK landscapes, and deceptive bimodal problems. We constructed LONs in different ways: the standard 'black box' approach of iterated local search (ILS) runs; an ILS algorithm using partition crossover which incorporated subfunction structure information a-priori; and an ILS algorithm which learns and uses subfunction structure during the search. The LONs were visualised in a new manner which conveys the number of subfunction changes between local optima. New metrics were proposed: these relate to the number and direction of subfunction changes between optima. These were compared across the three LON algorithm approaches and also compared against existing LON metrics from the literature.

The results showed considering subfunction structure in the analysis of LONs can increase the amount of insight gained into optimisation dynamics: for example, identifying how many subfunctions must change in order to ascend fitness levels. The proposed metrics were shown to contain different information to existing LON metrics from the literature. Lastly, a case study was presented. We used the new approach to compare landscape structure and algorithm trajectories between problems with differing degrees of subfunction overlap. Future work will consider the scalability of our approach, as well as comparing the LON subfunction structure analysis with an analysis done using only partial learned problem information. Code, data, and additional plots associated with this work are available in a dedicated Zenodo repository [see footnote 1].

Acknowledgements. The work of Michal W. Przewozniczek was supported by the Polish National Science Centre (NCN) under Grant 2022/45/B/ST6/04150.

References

[1] Niyazi Ari and Makhamadsulton Ustazhanov. 2014. Matplotlib in python. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*. IEEE, 1–6.

[2] Anton Bouter, Tanja Alderliesten, Arjan Bel, Cees Witteveen, and Peter A. N. Bosman. 2018. Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Kyoto, Japan) *(GECCO '18)*. Association for Computing Machinery, New York, NY, USA, 1199–1206. doi:10.1145/3205455.3205610

[3] Lorenzo Canonne, Bilel Derbel, Francisco Chicano, and Gabriela Ochoa. 2023. To Combine or not to Combine Graybox Crossover and Local Search?. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 257–265.

[4] Francisco Chicano, Darrell Whitley, Gabriela Ochoa, and Renato Tinós. 2017. Optimizing one million variable NK landscapes by hybridizing deterministic recombination and local search. In *Proceedings of the genetic and evolutionary computation conference*. 753–760.

[5] Francisco Chicano, Darrell Whitley, Gabriela Ochoa, and Renato Tinós. 2024. Generalizing and Unifying Gray-Box Combinatorial Optimization Operators. In *Parallel Problem Solving from Nature – PPSN XVIII: 18th International Conference, PPSN 2024, Hagenberg, Austria, September 14–18, 2024, Proceedings, Part I* (Hagenberg, Austria). Springer-Verlag, 52–67.

[6] Francisco Chicano, Darrell Whitley, and Andrew M Sutton. 2014. Efficient identification of improving moves in a ball for pseudo-boolean problems. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation*. 437–444.

[7] Kalyanmoy Deb and David E. Goldberg. 1993. Sufficient Conditions for Deceptive and Easy Binary Functions. *Ann. Math. Artif. Intell.* 10, 4 (1993), 385–408.

[8] Kalyanmoy Deb, Jeffrey Horn, and David E. Goldberg. 1993. Multimodal Deceptive Functions. *Complex Systems* 7, 2 (1993).

[9] Arkadiy Dushatskiy, Tanja Alderliesten, and Peter A. N. Bosman. 2021. A Novel Approach to Designing Surrogate-assisted Genetic Algorithms by Combining Efficient Learning of Walsh Coefficients and Dependencies. *ACM Trans. Evol. Learn. Optim.* 1, 2, Article 5 (July 2021), 23 pages. doi:10.1145/3453141

[10] Brian W. Goldman and William F. Punch. 2014. Parameter-less Population Pyramid. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (Vancouver, BC, Canada) *(GECCO '14)*. ACM, 785–792.

[11] Aric Hagberg and Drew Conway. 2020. Networkx: Network analysis with python. *URL: https://networkx. github. io* (2020).

[12] R. B. Heckendorn. 2002. Embedded Landscapes. *Evolutionary Computation* 10, 4 (2002), 345–369.

[13] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. 2016. Communities of local optima as funnels in fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. 325–331.

[14] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. 2018. PageRank centrality for performance prediction: the impact of the local optima network model. *Journal of Heuristics* 24 (2018), 243–264.

[15] Marcin Michal Komarnicki, Michal Witold Przewozniczek, Renato Tinós, and Xiaodong Li. 2024. Overlapping Cooperative Co-Evolution for Overlapping Large-Scale Global Optimization Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Melbourne, VIC, Australia) *(GECCO '24)*. Association for Computing Machinery, New York, NY, USA, 665–673. doi:10.1145/3638529.3654171

[16] Katherine Mary Malan. 2021. A survey of advances in landscape analysis for optimisation. *Algorithms* 14, 2 (2021), 40.

[17] Werner Mostert, Katherine M Malan, Gabriela Ochoa, and Andries P Engelbrecht. 2019. Insights into the feature selection problem using local optima networks. In *Evolutionary Computation in Combinatorial Optimization: 19th European Conference, EvoCOP 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24–26, 2019, Proceedings 19*. Springer, 147–162.

[18] M. Munetomo and D.E. Goldberg. 1999. A genetic algorithm using linkage identification by nonlinearity check. In *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*, Vol. 1. 595–600 vol.1. doi:10.1109/ICSMC.1999.814159

[19] Masaharu Munetomo and David E. Goldberg. 1999. Linkage identification by non-monotonicity detection for overlapping functions. *Evol. Comput.* 7, 4 (dec 1999), 377–398. doi:10.1162/evco.1999.7.4.377

[20] Gabriela Ochoa and Francisco Chicano. 2019. Local optima network analysis for MAX-SAT. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1430–1437.

[21] Gabriela Ochoa, Marco Tomassini, Sebástien Vérel, and Christian Darabos. 2008. A study of NK landscapes' basins and local optima networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. 555–562.

[22] Gabriela Ochoa, Nadarajen Veerapen, Fabio Daolio, and Marco Tomassini. 2017. Understanding phase transitions with local optima networks: number partitioning as a case study. In *Evolutionary Computation in Combinatorial Optimization: 17th European Conference, EvoCOP 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings 17*. Springer, 233–248.

[23] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[24] Michal W. Przewozniczek, Renato Tinós, Bartosz Frej, and Marcin M. Komarnicki. 2022. On Turning Black - into Dark Gray-Optimization with the Direct Empirical Linkage Discovery and Partition Crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 269–277. doi:10.1145/3512290.3528734

[25] Michal Witold Przewozniczek, Renato Tinós, and Marcin Michal Komarnicki. 2023. First Improvement Hill Climber with Linkage Learning – on Introducing Dark Gray-Box Optimization into Statistical Linkage Learning Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lisbon, Portugal) *(GECCO '23)*. ACM, 946–954.

[26] Y. Sun, X. Li, A. Ernst, and M. N. Omidvar. 2019. Decomposition for Large-scale Optimization Problems with Overlapping Components. In *Proc. IEEE Congr. Evol. Comput. (CEC)*. 326–333.

[27] Sara Tari, Gabriela Ochoa, Matthieu Basseur, and Sébastien Verel. 2023. On the Global Structure of PUBOi Fitness Landscapes. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. 247–250.

[28] Matheus C Teixeira and Gisele L Pappa. 2022. Understanding AutoML search spaces with local optima networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 449–457.

[29] Dirk Thierens and Peter A.N. Bosman. 2013. Hierarchical Problem Solving with the Linkage Tree Genetic Algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*. ACM, 877–884.

[30] Sarah L Thomson, Fabio Daolio, and Gabriela Ochoa. 2017. Comparing communities of optima with funnels in combinatorial fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 377–384.

[31] Sarah L Thomson, Gabriela Ochoa, and Sébastien Verel. 2022. The fractal geometry of fitness landscapes at the local optima level. *Natural Computing* 21, 2 (2022), 317–333.

[32] Sarah L Thomson, Nadarajen Veerapen, Gabriela Ochoa, and Daan van den Berg. 2023. Randomness in local optima network sampling. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. 2099–2107.

[33] Renato Tinós, Michal W. Przewozniczek, and Darrell Whitley. 2022. Iterated Local Search with Perturbation Based on Variables Interaction for Pseudo-Boolean Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. ACM, 296–304.

[34] Renato Tinós, Michal W Przewozniczek, and Darrell Whitley. 2022. Iterated local search with perturbation based on variables interaction for pseudo-boolean optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 296–304.

[35] Renato Tinós, Darrell Whitley, and Francisco Chicano. 2015. Partition Crossover for Pseudo-Boolean Optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII* (Aberystwyth, United Kingdom) *(FOGA '15)*. Association for Computing Machinery, New York, NY, USA, 137–149. doi:10.1145/2725494.2725497

[36] German Treimun-Costa, Elizabeth Montero, Gabriela Ochoa, and Nicolás Rojas-Morales. 2020. Modelling parameter configuration spaces with local optima networks. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 751–759.

[37] Sébastien Verel, Fabio Daolio, Gabriela Ochoa, and Marco Tomassini. 2012. Local optima networks with escape edges. In *Artificial Evolution: 10th International Conference, Evolution Artificielle, EA 2011, Angers, France, October 24-26, 2011, Revised Selected Papers 10*. Springer, 49–60.

[38] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.

[39] D. Whitley. 2019. Next generation genetic algorithms: a user's guide and tutorial. In *Handbook of Metaheuristics*. Springer, 245–274.

[40] Darrell Whitley, Hernan Aguirre, and Andrew Sutton. 2020. Understanding Transforms of Pseudo-Boolean Functions. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Cancún, Mexico) *(GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 760–768.

[41] Darrell Whitley and Gabriela Ochoa. 2022. Local optima organize into lattices under recombination: an example using the traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 757–765.

[42] L. Darrell Whitley, Francisco Chicano, and Brian W. Goldman. 2016. Gray Box Optimization for Mk Landscapes Nk Landscapes and Max-Ksat. *Evol. Comput.* 24, 3 (Sept. 2016), 491–519. doi:10.1162/EVCO_a_00184

[43] Tian-Li Yu, Kumara Sastry, and David E. Goldberg. 2005. Linkage Learning, Overlapping Building Blocks, and Systematic Strategy for Scalable Recombination. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05)*. ACM, 1217–1224.

[44] Ryan Zhou, Jaume Bacardit, Alexander EI Brownlee, Stefano Cagnoni, Martin Fyvie, Giovanni Iacca, John McCall, Niki van Stein, David J Walker, and Ting Hu. 2024. Evolutionary Computation and Explainable AI: A Roadmap to Understandable Intelligent Systems. *IEEE Transactions on Evolutionary Computation* (2024).