Tunnelling Crossover Networks

Gabriela Ochoa
Department of Computing Science and Mathematics
University of Stirling
Stirling, FK9 4LA, Scotland
goc@cs.stir.ac.uk

Francisco Chicano
E.T.S. Ingeniería Informática
University of Málaga
Andalucía Tech, Spain
chicano@lcc.uma.es

Renato Tinós
Department of Computing and Mathematics
University of São Paulo
Ribeirão Preto, SP, Brazil
rtinos@ffclrp.usp.br

Darrell Whitley
Department of Computer Science
Colorado State University
Fort Collins, CO, USA
whitley@cs.colostate.edu

ABSTRACT

Local optima networks are a recent model of fitness landscapes. They compress the landscape by representing local optima as nodes, and search transitions among them as edges. Previous local optima networks considered transitions based on mutation; this study looks instead at transitions based on deterministic recombination. We define and analyse networks based on the recently proposed partition crossover for k-bounded pseudo-Boolean functions, using NKq landscapes as a case study. Partition crossover was initially proposed for the travelling salesman problem, where it was found to "tunnel" between local optima, i.e., jump from local optimum to local optimum. Our network analysis shows that this also happens for NK landscapes: local optima are densely connected via partition crossover. We found marked differences between the adjacent and random interaction NK models. Surprisingly, with the random model, instances have a lower number of local optima on average, but their networks are more sparse and decompose into several clusters. There is also large variability in the size and pattern of connectivity of instances coming from the same landscape parameter values. These network features offer new insight informing why some instances are harder to solve than others.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

Keywords

Local Optima Network; Partition Crossover; Hill Climbing; Local Search; Fitness Landscapes; NK-landscapes

GECCO '15, July 11–15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754657

1. INTRODUCTION

The number and distribution of local optima are fundamental features of fitness landscapes that can impact the performance of heuristic optimisation methods. In his seminal work [7], F. Glover states: "*Perhaps the most conspicuous limitation of a heuristic method for problems involving discrete alternatives is the ability to become trapped at a local optimum*".

Local Optima Networks (LONs) are a recent network-based model of combinatorial fitness landscapes capturing the structure and topology of local optima [12, 13, 16, 17]. They represent the local optima of the underlying optimisation problem as vertices; and as edges, possible transitions among them using a given search operator. Local optima networks were inspired by work in physics where *energy surfaces* are modelled as complex networks [6]. Modelling landscapes as networks introduces a new set of metrics to analyse computational search spaces and the possibility of visualising them [9, 10]. Previous work on local optima networks considered transitions among optima based on mutation operators; both binary and permutation spaces have been studied [5, 13]. The main contribution of this article is to instead consider transitions based on recombination.

Specifically, we consider the partition crossover recently proposed for k-bounded pseudo-Boolean functions (i.e., real-valued functions over binary strings where the interactions between variables is bounded by a constant k) [15]. NK landscapes, MAX-kSAT, and several graph optimisation problems, such as MAX-CUT, are examples of these functions. This class of functions is relevant to evolutionary computation, combinatorial optimisation, and machine learning.

Partition Crossover (PX) was first introduced for the travelling Salesman Problem (TSP) [19]. The operator exploits the fact that two "parent" solutions can be decomposed into a number of partial solutions or sub-tours (i.e., tours compressing a subset of the cities). Each partial solution is linearly independent from the other partial solutions, which means that each partial solution can be evaluated independently. For a specific pairing of parents, partial solutions are also modular so that one partial solution from one parent can replace a partial solution from another parent so as to always yield a feasible offspring.

Whitley et al. [19] have also shown that when partition crossover is used to recombine parent solutions that are local optima, the offspring are also local optima with a high probability (around 80% for TSP instances). There are two reasons for this. (i) PX generates offspring that are guaranteed to be "piecewise" locally optimal: no single partial solution that is inherited form a parent can be improved by local search, because every partial solution has already been optimized by local search, and each partial solution is inherited as a unit during recombination. Thus, the only way that local search can improve an offspring is by discovering an improving move that alters two or more partial solutions. This rarely happens, which may be explained by a more subtle secondary factor. (ii) PX is respectful and transmits edges from the parent solutions. *Transmission* means all of the edges in the offspring are inherited from the parents, while *respect* that all shared edges in the parents are also inherited by the offspring. In the TSP it is the *edges* that are inherited from the parents that impacts the evaluation of the offspring. Thus, if the parents are locally optimal and have a very good evaluation, the offspring will also likely have a very good evaluation, which increases the likelihood that the offspring will also be locally optimal. In a recent paper, Tinós, Whitley and Chicano [15] have shown that a similar decomposition exists for NK landscapes into partial solutions that are modular and which can be independently evaluated. They also demonstrated the effectiveness of PX, and again showed that when it is applied to parents that are locally optimal, the offspring are always locally optimal in a restricted hyperplane subspace, and are usually local optimal in the full search space as well.

This paper defines and analyses local optima networks where the transitions among optima are based on the application of PX over binary spaces such as NK landscapes, but the results may be extended to MAX-kSAT and other pseudo-Boolean problems. More specifically, the primary contributions of the paper are the following:

1. Definition of the mathematical object XLON (crossover local optima networks) with a case study considering PX.
2. Implementation of an effective algorithm for constructing the networks following recent developments which exploit the structure of k-bounded additive fitness functions, leading to fast deterministic hill-climbing [3].
3. Statistical analysis and visualisation of the extracted networks.
4. Empirical evidence indicating that the number of local optima is not necessarily a reliable indicator of search difficulty in combinatorial optimisation.

2. NK LANDSCAPES

Our study considers the NKq ('quantized' NK) family of landscapes [11], one of several models adding tunnable neutrality to the standard Kauffman's NK model [8].

NK landscapes are defined as a real stochastic function f on binary strings $x \in \mathbb{B}^N$, $f : \mathbb{B}^N \rightarrow \mathbb{R}$. The value of K determines how many other bits in the string interact with each bit x_i. These bit interactions can be expressed as sub-functions, f_i, where each sub-function takes $K+1$ bits as input. These K interacting bits can either be the nearest neighbours (adjacent model) or selected uniformly at random (random model). The fitness (evaluation) function is given by:

$$f(x) = \sum_{i=1}^{N} f_i(x|_{mask_i}), \quad (1)$$

where $mask_i$ selects the $k = K+1$ bits that will be accessed by sub-function f_i. In practice, the mask can be dropped in as much as it is inherent in the definition of sub-function f_i. In the standard model, the codomain values for each sub-function f_i are randomly generated real numbers from $[0, 1]$. In NKq landscapes, the fitness contributions are instead integers drawn from the range $[0, q)$. This allows solutions with the same fitness (i.e., landscapes with neutrality). Our study considers a value of $q = 100$, which produces discrete landscapes with a very low degree of neutrality. Future work will study partition crossover local optima networks for neutral landscapes of the type that are common in MAX-kSAT problems.

It should also be noted that while NK landscapes are the focus of the current study, the methods used here and the partition crossover operator can be applied to all pseudo-Boolean functions with closed form evaluation functions. Just as every SAT problem can be expressed as a MAX-kSAT problem, every pseudo-Boolean function with an algebraic evaluation function can be expressed as a quadratic pseudo-Boolean function [1]. Thus, it should be noted that NK landscapes where $K = 1$ are also quadratic pseudo-Boolean functions (i.e., $k = 2$). Future work should explore what happens to the fitness landscape when NK landscapes with larger values of K are converted into quadratic pseudo-Boolean functions where $k = 2$ ($K = 1$).

3. PARTITION CROSSOVER (PX)

Under partition crossover, the evaluation of the offspring can be directly obtained from partial evaluations of components found in the parents. For NK landscapes, components are sets of differing bits in the parent solutions with interactions within the sets but no interactions among sets. If the differing bits found in the two parents can be partitioned into q components, partition crossover can be used to find the best of 2^q possible offspring. This is done at $O(N)$ cost per recombination. If the parents are locally optimal, any offspring must be locally optimal in a (more restricted) hyperplane subspace.

Given a subset of variables z of f, we would like to find a partition π of z such that f can be additively decomposed into $|\pi|$ sub-functions g_i for $i = 1$ to $|\pi|$, where each g_i depends on the variables in the subset π_i but does not depend on the variables in $z - \pi_i$.

We will use the *Variable Interaction Graph* (VIG) proposed by Chicano et. al. [3], where the vertices of the graph are the N variables, and two variables (vertices) are connected by an edge if they appear together in the same sub-function f_i. Note that we can potentially remove additional edges from the Variable Interaction Graph if the variable interactions do not result in a nonlinear interaction; this can be determined by taking the Walsh transforms of the sub-functions: if i and j appear together in any Walsh coefficient that is non-zero, edge $e(v_i, v_j)$ is added to the VIG, otherwise it can be removed. However, in most cases, vari-

ables that appear together in some sub-function also have a nonlinear interaction.

We are interested in how partition crossover can be used to partition the VIG.

PROPOSITION 1 ([15]). *For any pseudo-Boolean function with VIG G, V the set of variables of f and z a subset of V. The most fine grain partition π of z that additively decomposes f is given by:*

$$f(x) = c + \sum_{i=1}^{|\pi|} g_i(x|_{mask_i}), \tag{2}$$

where $\pi_i \subseteq mask_i$, $(z - \pi_i) \cap mask_i = \emptyset$ and the constant c does not depend on any variable in z, is the partition induced by the connected components in the subgraph $G[z]$.

Proposition 1 defines the most fine grain way to additively decompose f when we are interested in separating the variables in a subset z. In partition crossover the set of variables z in which we are interested is the set of differing variables in the two parent solutions. Thus, instead of the VIG of the function f, it will be useful in the following to work with a reduced graph: the *recombination graph*. Prior to applying partition crossover it is necessary to construct the VIG, but the same VIG is common to all recombinations for the same objective function.

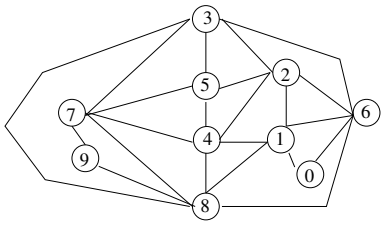
DEFINITION 1 (RECOMBINATION GRAPH [15]). *Let $f : \mathbb{B}^N \to \mathbb{R}$ be a pseudo-Boolean function with Variable Interaction Graph G and $x, y \in \mathbb{B}^N$ two solutions. We define the recombination graph of function f for solutions x and y as $G[x \oplus y]$ (where $\oplus$ denotes the exclusive OR bitwise operation), that is the subgraph of G composed of the variables in which x and y differ.*

The first step of the operator is to transfer all common bits to the offspring. Next, using the two parents x and y, determine the recombination graph and use Breadth First Search to find the connected components of the recombination graph. These connected components form a partition of the set of variables in $x \oplus y$ (the differing bits in the parent solutions).

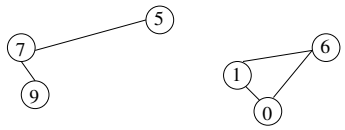
Having determined the partition of variables in $x \oplus y$, test the two assignments from the two parents. Since this only includes non-shared assignments to variables, the solutions from the two parents will be complements. Proposition 1 ensures that we can write f as a linear sum of sub-functions g_i where each one depends on the variables in one component π_i and not the others. Select the assignment in each connected component that results in the best evaluation for the corresponding sub-function g_i.

THEOREM 1. **The Partition Crossover Theorem [15]:** *For any k-bounded pseudo-Boolean function f, if the recombination graph of f for solutions x and y contains q connected components, then partition crossover returns the best of 2^q solutions; including the parent solutions.*

We will use the following sub-functions to illustrate the VIG and the construction of a recombination graph. In this case, $N = 10$ and K=2 (k=3). The sub-functions are as follows:



The Variable Interaction Graph



The decomposed Recombination Graph

Figure 1: The upper graph is the Variable Interaction Graph (VIG) of the sample function. When recombining the solutions 0000000000 and 1100011101, the vertices and edges associated with shared variables 2, 3, 4, 8 are deleted to yield the recombination graph which is shown below the VIG. If the recombination graph can be partitioned into q connected components, then recombination is guaranteed to return the best of 2^q possible offspring. In this example, there are only two components.

$f_0(x_0, x_1, x_6)$ $f_1(x_1, x_4, x_8)$ $f_2(x_2, x_3, x_5)$ $f_3(x_3, x_2, x_6)$
$f_4(x_4, x_2, x_1)$ $f_5(x_5, x_7, x_4)$ $f_6(x_6, x_8, x_1)$ $f_7(x_7, x_3, x_5)$
$f_8(x_8, x_7, x_3)$ $f_9(x_9, x_7, x_8)$

Each function yields a set of 4 nonlinear Walsh coefficients. Assume all of the Walsh coefficients are non-zero. This means that every pair of variables that appear together in some sub-function are connected by an edge in the VIG. This set of sub-functions results in the VIG shown in Figure 1.

Assume we wish to recombine two solutions 0000000000 and 1100011101; the variables are numbered from 0 to 9 from left to right. These two solutions reside in the hyperplane subspace **000***0* because these zero bits are shared in common by the parents. If the parents are locally optimal solutions, then any offspring are provably locally optimal in the reduced hyperplane subspace (but may or may not be locally optimal in the full search space).

From the VIG, we will generate the recombination graph by deleting the variables that share a common value assignment. The edges connected to the variables with common values are also deleted. We next ask if deleting these variables and edges "partitions" the recombination graph. In our example, the variables with shared values are 2, 3, 4 and 8. The variables that have different assignments are 0, 1, 5, 6, 7 and 9. The recombination graph is partitioned into two independent subgraphs, as illustrated in the lower graph in Figure 1. This means a new evaluation function can be constructed:

$$g(x) = c + g_1(x_5, x_7, x_9) + g_2(x_0, x_1, x_6)$$

For any string that resides in the subspace **000***0*, $f(x) = g(x)$, therefore $g(x)$ can be used to evaluate any offspring produced from the parent strings using partition crossover. Furthermore, in the example shown here, all offspring that are produced by partition crossover must also be locally optimal for the sub-functions g_1 and g_2.

In general, for every subgraph in the recombination graph, there will exist a sub-function g_i and every offspring generated by partition crossover will be locally optimal with respect to every sub-function g_i. In most cases, the offspring will also be locally optimal in the full search space. If the offspring is not a local optima in the full search space, the offspring can only be improved by first flipping one of the bits that was shared in common by the parents (meaning that one cannot flip any bit evaluated by a sub-functions g_i to obtain an improvement.)

4. CROSSOVER NETWORKS (XLON)

In order to produce the crossover networks, PX is applied to solutions that are locally optimal. To find a useful recombination graph, two parent solutions must share a significant number of bits in common. Random solutions tend to have fewer bits in common compared to parents that are local optima. Partition crossover is both deterministic and greedy. Given two parents, it deterministically generates the recombination graph, and then greedily returns the best possible partial solution for every sub-function g_i. In some cases, the best possible offspring is one of the parents.

In the XLON model, the nodes will be the set of all local optima in the landscape with respect to a given neighbourhood. As neighbourhoods, we will consider the Hamming distance 1 and 2 neighbourhoods (i.e., 1-flip and 2-flip). Directed edges will go from parents to their offspring using partition crossover. Formal definitions and the algorithmic procedure to construct the networks are detailed below.

4.1 Definitions

A fitness landscape [14] is a triplet $(S, \mathcal{N}, f)$ where S is a set of potential solutions i.e., a search space, $\mathcal{N} : S \longrightarrow 2^S$, a neighbourhood structure, is a function that assigns to every $s \in S$ a set of neighbours $\mathcal{N}(s)$, and $f : S \longrightarrow \mathbb{R}$ is a fitness function that can be pictured as the *height* of the corresponding solutions.

In our study, the search space is $\mathbb{B}^N$, i.e. the space of binary strings of length N, so its size is 2^N. As neighbourhoods, we consider the Hamming distance 1 and 2 neighbourhoods, that is, the set of all solutions at most Hamming distance 1 or 2, respectively from the current solution.

Vertices (V): Is the set of local optima in the search space. A local optimum, which is taken to be a maximum here, is a solution s' such that $\forall s \in \mathcal{N}$, $f(s) \leq f(s')$. Local optima are exhaustively obtained with Algorithm 1.

PX Edges (E_{PX}): Is the set of directed edges between parents and offspring after partition crossover. Specifically, there is an edge $e_{x,z}$ between local optima x and z in V if x is member of a crossover pair producing z. Either z is a local optimum directly after applying crossover, or it is the result of improving the offspring with a next improvement hill-climber. Multiple edges are possible between a pair of vertices (x, z) when z is produced by more than one crossover pair involving x. Partition crossover is not always possible, that is, for some pairs of local optima no new offspring is produced and the operator returns the best parent. This can happen for two reasons. (i) There is no partition of the VIG (Figure 1) for a pair of parents. (ii) Recombination is possible, but one of the two parents is better than any offspring. These unsuccessful pairings are not captured in the network-based model; there are no self-loops in the network.

Crossover Local Optima Network ($XLON$): Is the graph $XLON = G(V, E_{PX})$ where nodes are the local optima V, and edges E_{PX} link parents to offspring after partition crossover. The network is directed, without self-loops, and can have multiple edges.

4.2 Construction

The construction of the crossover networks proceeds in two steps: (i) all the vertices (i.e., the set V of local optima) are identified, (ii) all the pairs of local optima are recombined with partition crossover. That is, if $nv = |V|$ is the number of vertices, all the $nv \times (nv - 1)/2$ pairs are considered. If the resulting offspring after crossover is not one of the parents, then two directed edges are added to the network, both ending in the resulting local optimum and starting from each of the parents. Most successful crossover pairs produce local optima (between 70% and 90% in our study, as can be seen in rows named p_{px} in Table 2). To further increase the number of local optima an efficient deterministic next-ascent hill climber is started from the non-optima offspring.

4.2.1 Local optima identification

In order to efficiently identify all the local optima in the search space we use the concept of *Score* as defined in [3]. We will use $S_i(x)$ to denote the change in the objective function when we move from solution x to the one obtained from x by flipping the i-th bit. Formally:

$$S_i(x) = f(x \oplus 1_i) - f(x), \tag{3}$$

where $\oplus$ denotes the exclusive OR bitwise operation and 1_i denotes a binary string with 1 in the i-th bit and 0 in the other bits. A solution x will be a local optimum if there is no Score $S_i(x)$ greater than zero (we are maximising). Algorithm 1 describes this procedure.

Algorithm 1 Identification of local optima (V)

Output: V (set of local optima)
1: $V \leftarrow \emptyset$
2: **for** $x \in \mathbb{B}^n$ **do**
3: **if** $S_i(x) \leq 0$ for all $1 \leq i \leq n$ **then**
4: $V \leftarrow V \cup \{x\}$
5: **end if**
6: **end for**

Two aspects of the implementation of Algorithm 1 make it very efficient. First, the Scores are computed from scratch only for the first solution, and thereafter, they are stored in a vector and updated incrementally. Second, solutions are explored following a Gray code order; the move from one solution to the next in the sequence flips only a single bit. Under this scenario, we can use the algorithms proposed by Chicano *et al.* [3], Whitley and Chen [18] or Chen *et al.* [2] to update the Score vector in constant time for pseudo-Boolean problems with bounded epistasis. Thus, the complexity of running Algorithm 1 in an NK landscape is $O(2^N)$, proportional to the size of the search space. It is worth mentioning that these speed ups allow the exhaustive exploration of

larger landscapes than was possible before. Our study considers N values up to 30, while a maximum of $N = 18$ was considered in previous mutation based local optima network studies for NK landscapes [13, 17].

4.2.2 Network construction

Once all the local optima are identified, the edges are computed following Algorithm 2, which uses the partition crossover proposed by Tinós *et al.* [15], and the next improvement deterministic hill-climber proposed by Chicano *et al.* [3]. Two local optima x and z are connected by an edge $e_{x,z}$ if z can be obtained from x by applying partition crossover to x and another solution y and then running the hill-climber to reach z.

Algorithm 2 Network construction

Input: V
Output: $XLON = G(V, E_{PX})$
1: **for** $\{x, y\} \subseteq V$ **do** {All pairs of local optima}
2: $w \leftarrow$ PartitionCrossover(x,y)
3: $z \leftarrow$ HillClimber (w)
4: **if** $z \neq x$ and $z \neq y$ **then**
5: $E_{PX} \leftarrow E_{PX} \cup \{(x, z), (y, z)\}$
6: **end if**
7: **end for**

5. NETWORK ANALYSIS

Our study considers both random and adjacent NKq landscapes. Random NKq landscapes are NP-hard problems. However, the adjacent NK landscapes can be solved to optimality in polynomial time. In the adjacent NK landscapes, two adjacent sub-functions, f_i and f_{i+1} share $k - 1 = K$ variables.

NKq landscapes were generated for values of $q = 100, K \in \{2, 3\}$, and $N \in \{20, 25, 30\}$. In addition, Hamming distance neighbourhoods of size $r \in \{1, 2\}$ were considered. When $r = 1$ the neighbourhood is the standard bit flip neighbourhood. Up to two bits are flipped when $r = 2$, yielding the Hamming distance 2 neighbourhood. Since the local optima networks with $r = 2$ were small and very sparsely connected, most of the network analysis is done for $r = 1$. $N = 20$ still allows easy visual inspection, and 30 is (approximately) the maximum size for which the networks can be extracted in reasonable time. K values of 2 and 3 produced networks which are not too sparse and have a small number of connected components. Larger values of K produced sparsely connected networks, specially for the random model, which indicates that finding successful partition crossover pairs (i.e., pairs producing offspring different than their parents) is harder as K increases.

A first approach to analyse the structure of a network is often to visualise it. Software for analysing and visualising networks is currently available in various languages and environments. Here we use the R statistical language together with the igraph package [4]. Figure 2 illustrates two examples of local optima networks for the studied NKq landscapes with $r = 1$ and adjacent (left plot) and random interaction (right plot). Nodes are local optima, their area is proportional to fitness and the global optimum is highlighted in red (darker color). Edges link parents to offspring after partition crossover. Dark (black) edges indicate those cases where the result of crossover was not a local optimum and thus required improvement with hill-climbing. Notice that this only happens in few cases. Our statistical analysis revealed than 70% to 90% of the edges represent local optima obtained directly after partition crossover, with the percentage being higher for the adjacent model (rows named p_{px} in Table 2).

The networks in Figure 2 illustrate a marked difference we observed between the random and adjacent NK models. Networks for the adjacent model have on average a larger number of nodes under the Hamming distance 1 neighbourhood ($r = 1$), are densely connected and feature a single or a very small number of connected components. On the other hand, networks for the random model have on average fewer nodes, tend to be sparser, and decompose into several clusters.

Visualisation is only useful for small networks, we therefore turn to the statistical analysis of larger networks. Since the mid-1990s, there has been an explosion of interest in networks and network-based approaches to the modelling and analysis of complex systems. Among the many features and metrics that can be collected from complex networks [9], we selected only a few after some preliminary experiments, including those that better distinguish among the landscape classes, and intuitively may impact the dynamic of search (Table 1).

Table 1: Selected metrics.

nv	Number of vertices (local optima).
ne	Number of edges.
$\bar{f}$	Average fitness of local optima.
nc	Number of connected components (or clusters).
$\bar{\delta}$	Average in-degree (incoming edges).
p_{px}	Proportion of edges where the offspring was a local optimum immediately after PX.

For each landscape parameter set 30 instances were generated (with different random seeds) and their recombination local optima networks extracted following the procedure described in Section 4.2. Table 2 summarises the network metrics as mean and standard deviation values ($\bar{X}_\sigma$) on the select sets of landscape parameters (i.e., $N \in \{20, 25, 30\}$, $K \in \{2, 3\}$, and $q = 100$) and the two interaction models: adjacent and random. The first half of the table shows the landscapes for $K = 2$, while the bottom half those with $K = 3$. The top part of each half of Table 2 shows the metrics for a Hamming neighbourhood of radius $r = 1$, while the bottom part for $r = 2$. For $r = 2$ only the 3 most basic network metrics, namely number of nodes and edges (nv, ne) and the average fitness of local optima, are shown as these are small and sparse networks.

As expected, for a fixed K the number of local optima (nv) increases steadily with increasing N. Similarly, for a fixed N, the number of local optima increases rapidly with increasing K. What is surprising is that the number of local optima is consistently (i.e., for all landscape parameters) larger for the adjacent model as compared to the random model for a Hamming distance 1 neighbourhood. To the best of our knowledge this has not been reported before.

The box plots in Figure 3 show the distribution of the number of local optima across the 30 instances for a selected landscape class, namely $N = 20, K = 3$. Each plot reports the two models (adjacent and random). The top plot refers

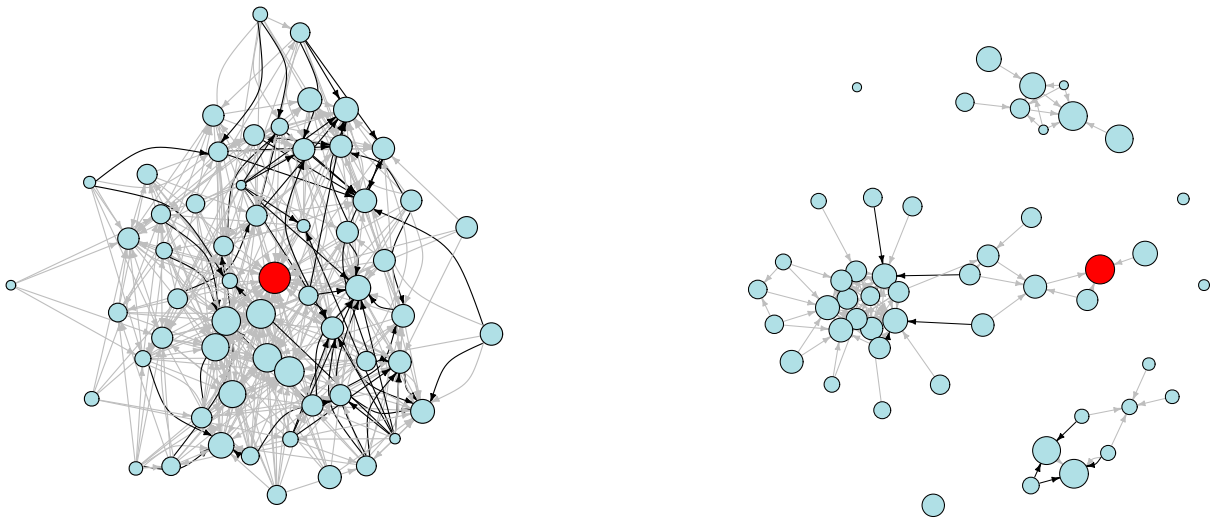


Figure 2: PX local optima networks for two selected instances with values $N = 20$, $K = 2$, $q = 100$. Nodes are local optima and edges connect parents to offspring after partition crossover (black edges indicate PX followed by hill-climbing). Vertex area is proportional to their fitness and the global optimum is highlighted in red (darker color). Left: Adjacent model, the network has 60 nodes and features a single connected component. Right: Random model, the network has 50 nodes (local optima) and features 7 connected components, 4 of which are isolated nodes.

to the Hamming distance 1 neighbourhoods ($r = 1$) and the bottom plot to distance 2 ($r = 2$). The number of local optima is substantially lower for $r = 2$, but their average fitness is higher as can be seen in Table 2 (rows $\bar{f}$ and $\bar{f}_{r=2}$).

The number of local optima is often considered to correlate with problem difficulty. However, this is not necessarily the case as our results indicate. The number of local optima is higher for the adjacent model and the distance 1 neighbourhood (Figure 3, top plot). But we know that: (i) the adjacent NK landscapes admit a polynomial time solution; (ii) empirical results indicate that a simple GA using partition crossover can easily find the global optimum for adjacent NK landscapes [15]. This is also confirmed by the experiments reported in section 6. Furthermore, when a Hamming distance 2 neighbourhood ($r = 2$) is used on exactly the same problems, there are more local optima for the random NK landscapes when $r = 2$ (Figure 3, bottom plot). Thus, while the adjacent NK landscapes yield more local optima for $r = 1$, most of these are "weak" local optima and disappear when $r = 2$.

The number of edges also clearly separates the adjacent from the random model. Networks are more densely connected for the adjacent model, which indicates a larger number of successful crossover pairs in this case. For $r = 2$, networks are sparsely connected (have a low number of edges), specially for the random model. The remaining 3 network features, number of components (nc), average in-degree ($\bar{\delta}$) and proportion of crossover pairs producing local optima (p_{px}), were only computed for $r = 1$. These metrics also clearly separate the two landscape models.

On one hand, the adjacent model shows an average number of components very close to one or even exactly one (i.e., a single connected component) for the larger networks ($N \in \{25, 30\}, K = 3$), and a noticeable higher average in-degree ($\bar{\delta}$), indicating that each node is produced by a larger number of crossover pairs. For both values of K the proportion of edges directly producing local optima after crossover (p_{px}) is higher. On the other hand, the random model reveals a much higher number of connected components, close to 100 on average for the landscapes with $K = 3$; and the in-degrees ($\bar{\delta}$) are around an order of magnitude lower as compared to those for the respective adjacent landscapes. For both values of K the proportion of edges directly producing local optima after crossover (p_{px}) is lower.

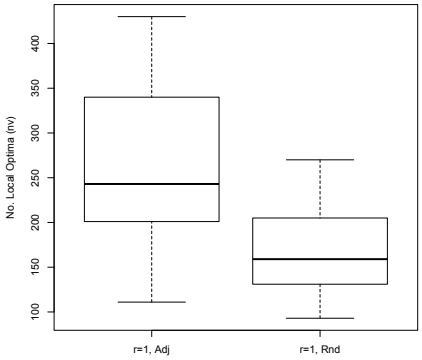
Our analysis suggests that the adjacent model is easier to solve by a PX-based GA as compared to the random model, despite the former producing a larger number of local optima on average. We hypothesise that search difficulty for a PX-based algorithm on the random model is related to the decomposition of the network into clusters and the low average in-degree.

6. ALGORITHM PERFORMANCE

An empirical study was conducted on the largest instances used in the network analysis, i.e., with $N = 30$, the two values of $K \in \{2, 3\}$ and the two landscape models. The hybrid genetic algorithm using partition crossover implemented in [15] was used. This implementation applies bit-flip local search with first improvement to random solutions in order to generate the initial population. Elitism is used to preserve

Table 2: Network metrics (as described in Table 1) for both random and adjacent NKq landscapes with $N \in \{20, 25, 30\}$ and $q = 100$. Top: $K = 2$. Bottom: $K = 3$. For each set of landscape parameters, results are mean and standard deviations ($\bar{X}_\sigma$) of 30 networks produced with different random seeds. Unless otherwise stated metrics refer to local optima networks with a Hamming distance neighbourhood $r = 1$.

$K = 2$						
	$N = 20$		$N = 25$		$N = 30$	
	Adjacent	*Random*	*Adjacent*	*Random*	*Adjacent*	*Random*
nv	$69.87_{32.61}$	$58.30_{27.30}$	$216.40_{120.86}$	$166.53_{98.53}$	$705.87_{545.48}$	$448.53_{254.65}$
ne	$688.83_{556.39}$	$179.73_{206.19}$	$6869.47_{7291.94}$	$1950.50_{2473.02}$	$80106.23_{139938.45}$	$12683.70_{13913.89}$
$\bar{f}$	$1363.88_{47.95}$	$1352.47_{46.60}$	$1701.84_{55.19}$	$1680.77_{72.68}$	$2045.08_{58.09}$	$2033.94_{82.76}$
nc	$1.47_{0.68}$	$8.40_{5.61}$	$1.60_{0.50}$	$4.90_{3.12}$	$1.43_{0.50}$	$2.57_{2.08}$
$\bar{\delta}$	$8.36_{3.54}$	$2.46_{1.73}$	$25.24_{12.30}$	$8.64_{6.26}$	$75.53_{51.18}$	$21.37_{13.80}$
p_{px}	$0.87_{0.08}$	$0.81_{0.13}$	$0.90_{0.07}$	$0.80_{0.11}$	$0.88_{0.08}$	$0.77_{0.08}$
$nv_{r=2}$	$6.20_{3.43}$	$6.97_{2.86}$	$10.57_{7.99}$	$13.60_{6.91}$	$14.70_{10.63}$	$23.57_{15.01}$
$ne_{r=2}$	$5.00_{6.90}$	$0.87_{2.40}$	$23.57_{43.88}$	$4.60_{7.39}$	$47.43_{63.44}$	$21.23_{34.04}$
$\bar{f}_{r=2}$	$1450.73_{68.80}$	$1419.58_{59.02}$	$1805.54_{69.20}$	$1764.67_{78.27}$	$2173.37_{62.41}$	$2129.01_{91.14}$
$K = 3$						
nv	$265.20_{86.40}$	$171.533_{50.78}$	$1105.97_{505.38}$	$648.33_{172.17}$	$4386.40_{2274.61}$	$2452.20_{803.55}$
ne	$3730.63_{2088.81}$	$154.80_{146.32}$	$77707.70_{69844.47}$	$2176.80_{2519.51}$	$1321295.27_{1298347.89}$	$27912.10_{28985.72}$
$\bar{f}$	$1346.84_{36.88}$	$1341.50_{39.92}$	$1676.01_{43.18}$	$1670.23_{41.44}$	$2004.21_{44.46}$	$2002.88_{44.95}$
nc	$1.90_{0.10}$	$88.50_{24.95}$	$1.17_{0.38}$	$111.53_{55.92}$	$1.10_{0.31}$	$83.37_{58.24}$
$\bar{\delta}$	$12.96_{3.88}$	$0.80_{0.61}$	$60.13_{23.23}$	$3.10_{2.65}$	$245.87_{111.32}$	$10.42_{7.21}$
p_{px}	$0.81_{0.07}$	$0.74_{0.12}$	$0.80_{0.06}$	$0.69_{0.09}$	$0.79_{0.05}$	$0.68_{0.07}$
$nv_{r=2}$	$15.90_{5.78}$	$23.10_{7.05}$	$36.80_{15.74}$	$53.67_{13.49}$	$67.87_{39.60}$	$128.67_{46.54}$
$ne_{r=2}$	$11.83_{9.63}$	$0.30_{1.02}$	$120.20_{93.54}$	$8.80_{17.23}$	$514.33_{523.32}$	$57.07_{146.34}$
$\bar{f}_{r=2}$	$1450.55_{47.64}$	$1411.43_{46.77}$	$1795.34_{55.11}$	$1761.92_{42.94}$	$2154.87_{55.59}$	$2111.62_{52.77}$



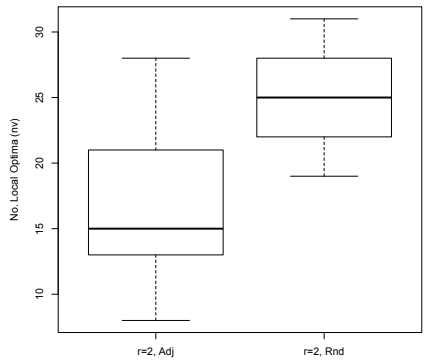


Figure 3: Distribution of the number of local optima for NKq landscapes with $N = 20, K = 3$, and Hamming distance neighborhoods $r \in \{1, 2\}$

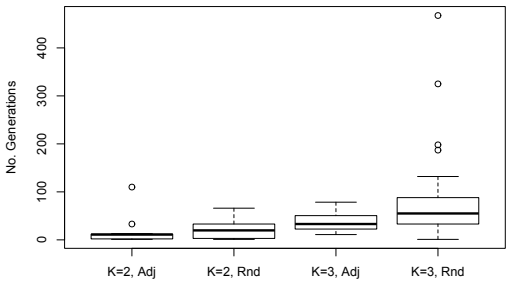


Figure 4: Distribution of the number of generations to find the global optima for a hybrid GA ruling on NKq landscapes with $N = 30$, $K \in \{2, 3\}$ and the two epistatic models.

the current best solution, while binary tournament selection is used to select parents for crossover and mutation. The population size was set to 10 individuals, given the small search spaces.

Instances are small and the global optimum was quickly found in all of them. Therefore, performance differences among landscape classes were not easy to assess. Figure 4 shows box-plots with the distribution of the number of generations before finding the global optimum when running the hybrid GA on the four landscape classes indicated. The values summarise the medians of 50 runs on each of the 30 instances for each landscape class. The number of generations to find the optimum is shorter for the adjacent model as compared to the random model (for both values of K), despite the former having a larger number of local optima.

7. CONCLUSIONS

We defined and analysed local optima networks with edges based on the recently proposed partition crossover for pseudo-Boolean optimisation, using the NKq landscape as a case study. Our network analysis revealed that for fixed N and K parameters and the 1-flip neighbourhood, the adjacent epistatic model produces much larger networks (i.e., with a larger number of local optima) as compared with the random model. It is well known that the adjacent model is easier to solve, which indicates that the number of local optima in a landscape is not necessarily a reliable indicator of difficulty. Our study of networks with Hamming distance 2 neighbourhood ($r = 2$) revealed that this reverses, i.e., the number of local optima for the random model is higher than for the adjacent model. Thus, while the adjacent NK landscapes yield more local optima for $r = 1$, most of these are "weak" local optima and disappear when $r = 2$.

The empirical results also show that the crossover networks for the random model are more sparsely connected than for the adjacent model. This means partition crossover does not as easily move between all local optima on random NK landscapes.

Future work will study crossover local optima networks for larger instances (by adequate sampling of the search space) and other combinatorial optimisation problems. The ultimate goal is to have a deeper understanding of the topological features of combinatorial search spaces over different search operators, which can help to predict performance and improve the design of heuristic search algorithms.

Acknowledgments

G. Ochoa acknowledges funding from the Engineering and Physical Sciences Research Council, UK (EPSRC) grant number EP/J017515. F. Chicano acknowledges funding from project number 8.06/5.47.4142 in collaboration with the VSB Technical University of Ostrava, UMA/FEDER FC14-TIC36, Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech and the Spanish Ministry of Education (grant CAS12/00274).

8. REFERENCES

[1] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155 – 225, 2002.

[2] W. Chen, D. Whitley, D. Hains, and A. Howe. Second order partial derivatives for NK-landscapes. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 503–510, New York, NY, USA, 2013. ACM.

[3] F. Chicano, D. Whitley, and A. M. Sutton. Efficient identification of improving moves in a ball for pseudo-boolean problems. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 437–444. ACM, 2014.

[4] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[5] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of the quadratic assignment problem. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010*, pages 1–8. IEEE Press, 2010.

[6] J. P. K. Doye. The network topology of a potential energy landscape: a static scale-free network. *Physical Review Letter*, 88:238701, 2002.

[7] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533 – 549, 1986.

[8] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.

[9] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[10] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2010.

[11] M. E. J. Newman and R. Engelhardt. Effect of neutral selection on the evolution of molecular species. *Proc. R. Soc. London B*, pages 1333–1338, 1998.

[12] G. Ochoa, M. Tomassini, S. Verel, and C. Darabos. A study of NK landscapes' basins and local optima networks. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*, pages 555–562. ACM, 2008.

[13] G. Ochoa, S. Verel, F. Daolio, and M. Tomassini. Local optima networks: A new model of combinatorial fitness landscapes. In H. Richter and A. Engelbrecht, editors, *Recent Advances in the Theory and Application of Fitness Landscapes*, volume 6 of *Emergence, Complexity and Computation*, pages 233–262. Springer Berlin Heidelberg, 2014.

[14] P. Stadler. Fitness landscapes. *Biological evolution and statistical physics*, pages 183–204, 2002.

[15] R. Tinós, D. Whitley, and F. Chicano. Partition crossover for pseudo-boolean optimization. In *Proceedings of FOGA*, 2015. (to appear).

[16] M. Tomassini, S. Verel, and G. Ochoa. Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Revview E*, 78(6):066114, 2008.

[17] S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of NK landscapes with neutrality. *IEEE Transactions on Evolutionary Computation*, 15(6):783–797, 2011.

[18] D. Whitley and W. Chen. Constant time steepest descent local search with lookahead for NK-landscapes and MAX-kSAT. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1357–1364, 2012.

[19] D. Whitley and A. Hains, D.and Howe. Tunneling between optima: partition crossover for the TSP. In *Proc. of GECCO'09*, pages 915–922, 2009.