

Saddles and Barrier in Landscapes of Generalized Search Operators

Christoph Flamm^a, Ivo L. Hofacker^b, Bärbel M. R. Stadler^b,
and Peter F. Stadler^{c,a,d}

^a*Department of Theoretical Chemistry
University of Vienna, Währingerstraße 17, A-1090 Wien, Austria*

^b*Max Planck Institute for Mathematics in the Sciences
Inselstrasse 22-26, D-04103 Leipzig, Germany*

^c*Bioinformatics Group, Department of Computer Science, and Interdisciplinary
Center for Bioinformatics (IZBI), University of Leipzig,
Härtelstraße 16-18, D-04107 Leipzig, Germany
Tel: ++49 341 97-16691 Fax: ++49 341 97-16709*

^d*Santa Fe Institute,
1399 Hyde Park Rd., Santa Fe, NM 87501, USA*

Abstract

Barrier trees are a convenient way of representing the structure of complex combinatorial landscapes over graphs. Here we generalize the concept of barrier trees to landscapes defined over general multi-parent search operators based on a suitable notion of topological connectedness that depends explicitly on the search operator. We show that in the case of recombination spaces, path-connectedness coincides with connectedness as defined by the mutation operator alone. In contrast, topological connectedness is more general and depends on the details of the recombination operators as well. Barrier trees can be meaningfully defined for both concepts of connectedness.

Key words: Recombination, Genetic Algorithm, Fitness Landscape, Barrier Tree, Generalized Topology, Connectedness, Path Connectedness

1 Introduction

The concept of energy landscapes has proven to be of fundamental relevance in investigations of complex disordered systems, from simple spin glass models

to biopolymer folding. Barrier trees [30,16,12,14,15] provide a convenient condensed representation of the discrete landscapes such as the energy landscapes of biopolymer folding and the fitness landscapes of complex combinatorial optimization problems. Barrier trees encapsulate information of mutual reachability of local optima and the energy/fitness barriers that separate them. The concept easily generalizes to PO-set-valued landscapes, which arise naturally in multi-objective optimization [40]. In most studies it has however been restricted to mutation (single parent) as the search operator. In [21,22], barrier trees are used for studying heuristic optimization algorithms including genetic algorithms. In this work the barrier trees are built relative to the Hamming (bit-flip) neighborhood, i.e., without regard to the structure of the underlying search operator.

In the case of multi-parent search operators it is not obvious how barrier trees should be defined in such a way that the structure of the search spaces that is induced by the search operators is faithfully represented. In a series of papers [38,36,4,42] we have explored how generalized topology can be used to describe the search spaces underlying evolutionary processes with recombination and chemical reaction networks. In the latter, the educts and products of chemical reactions take on the roles of the parents and off-springs, respectively. In this contribution we first demonstrate that search spaces of combinatorial optimization problems inherit a definite topological structure from the collection of search operators that is used by a particular algorithm, such as Simulated Annealing, a Genetic Algorithm, or Genetic Programming. We then show that this topological structure implies a natural concept of connectedness.

The connectedness of subsets in a search space is a property that, intuitively, should have a close relation to properties of reachability or accessibility. Such notions, however, lie at the heart of theories that explain the performance of heuristic optimization procedures on value landscapes. The simplest example are the “cycles” in the theory of Simulated Annealing [2], which in essence can be understood as the connected components of a subset of search space on which the cost function has values better than a given threshold η . We will see that, in conjunction with the cost function, connectedness of subsets then defines a structure of basins and barriers that generalizes the notion of barrier trees from graphs to spaces induced by arbitrary search operators. We finally give a brief example demonstrating that such a type of landscape analysis is indeed computationally feasible at least for certain Genetic Algorithms.

2 Search Operators and Generalized Topology

A (combinatorial) optimization problem is usually specified in terms of a set X of configuration and a cost function $f : X \rightarrow \mathbf{R}$, where \mathbf{R} is an ordered set. In

the case of multi-objective optimization [11] we have to admit partially ordered value sets \mathbf{R} [40]. A large class of heuristic algorithms, including Simulated Annealing, Genetic Algorithms, Evolutionary Strategies, or Genetic Programming, attempt to find optimal solutions by moving through the set X and evaluating the cost function at different points $x \in X$. This search procedure imposes an implicit mathematical structure on the set X that determines how points or, more generally, subsets are mutually accessible. In a more biologically inspired setting, this *search space* is uniquely determined by the genetic operators at work: mutation, recombination, genome rearrangements, and so on.

2.1 Mutation and Move Sets

In the case of point mutations and constant length sequences, the situation is straight forward. Naturally, sequences that differ by a single mutation are *neighbors* in “sequence space” [32,13]. The sequence space can thus be represented as a graph, also known as Hamming graphs or generalized hypercube. The Hamming distance, d_H , counts the number of positions at which two sequences differ.

Move sets are by no means restricted to mutating letters in fixed length string representations. Another examples that are commonly used in an evolutionary optimization context are permutation operators (e.g. for tours of Traveling Salesman Problems) or the exchange operator (e.g. for Graph Bipartitioning), see [34] and the references therein. These moves, which depend on a single “parent”, not on a population, naturally define *edges* on the set X of configurations. Therefore, the search space has the structure of a graph. Obviously, this graph is connected if and only if the move set is ergodic.

2.2 Recombination Spaces

The situation becomes more complicated, however, when recombination (crossover) is considered [17]. The analogue of the adjacency relation of the graph is the recombination set $\mathcal{R}(x, y)$, which is defined as the set of all (possible) recombinants of two parents x and y . Recombination sets satisfy at least two axioms:

- (X1) $\{x, y\} \in \mathcal{R}(x, y)$,
- (X2) $\mathcal{R}(x, y) = \mathcal{R}(y, x)$.

Condition (X1) states that replication may occur without recombination, and (X2) means that the role of the two parents is interchangeable. Often a third

condition

$$(X3) \quad \mathcal{R}(x, x) = \{x\}$$

is assumed, which is, however, not satisfied by models of unequal crossover [35,38]. Functions $\mathcal{R} : X \times X \rightarrow \mathcal{P}(X)$ satisfying (X1-X3) were considered recently as *transit functions* [9] and as *P-structures* [43,41].

In the case of strings of fixed lengths n one requires additional properties. We write x_i for the i -th letter in string x . We may assume a different alphabet \mathcal{A}_i for each position i . While for GAs one usually has $\mathcal{A}_i = \{0, 1\}$ for all positions i , one may have a different number alleles for different genes e.g. in a population genetic setting.

$$(X4) \quad \mathcal{R}(x, y) \subseteq \text{span}\{x, y\}, \text{ where}$$

$$\text{span}A = \{z \in X \mid \forall i : \exists x \in A : z_i = x_i\} \quad (1)$$

is the *linear span* of a set A .

$$(X5) \quad \text{For } x, y \in X \text{ and } i \neq j \text{ there is a recombinant } z \in \mathcal{R}(x, y) \text{ with } z_i = x_i \text{ and } z_j = y_j.$$

The linear span $\text{span}A$ correspond to Antonisse’s definition of a schema [1]. It also can be interpreted as a “hyperplane” in the Hamming graph with vertex set X and Hamming neighborhood on $X = \prod_{i=1}^n \mathcal{A}_i$, see e.g. [28]. We will not pursue the vector spaces aspects of this construction here, however.

It follows directly from equ.(1) that span is idempotent:

$$\text{span}(\text{span}A) = \text{span}A. \quad (2)$$

For string recombination operators, (X4) implies (X3) since $\text{span}\{x\} = \{x\}$. Furthermore, we note that for *uniform crossover*, $\mathcal{R}(x, y) = \text{span}\{x, y\}$.

Condition (X5) characterizes *proper recombination operators* in which any two sequence positions can be separated by cross-over. Note that strict 2-point cross-over (i.e., exactly two break-points *within* the strings) is not proper in this sense, since the first and the last sequence position always stay together in the off-springs. The more common definition, which calls for at most break-points, is of course proper.

We note for later usage we collect here a few simple properties of recombination spaces.

Lemma 1 $z \in \mathcal{R}(x, y)$ implies $\text{span}\{x, z\} \subseteq \text{span}\{x, y\}$.

Proof. $u \in \text{span}\{x, z\}$ iff $u_i = x_i$ or $u_i = z_i$ for $i = 1, \dots, n$ and $z \in \text{span}\{x, y\}$

iff $z_i = x_i$ or $z_i = y_i$. Thus $u_i = x_i$ or $u_i = y_i$, and hence $u \in \text{span}\{x, y\}$. \square

Lemma 2 *Let \mathcal{R} be a proper recombination operator. Then $\mathcal{R}(x, y) \setminus \{x, y\} \neq \emptyset$ if and only if $d_H(x, y) \geq 2$.*

Proof. If $d_H(x, y) \geq 2$ then there are two sequence positions $i \neq j$ such that $x_i \neq y_i$ and $x_j \neq y_j$. Since \mathcal{R} separates i and j , there is a recombinant z with $z_i = x_i \neq y_i$ and $z_j = y_j \neq x_j$, i.e., $z \neq x$ and $z \neq y$. \square

In the case of Genetic algorithms it seems natural to define reachability via the union of those of mutation and recombination.

2.3 Closure Functions

In the most general case we are given a collection \mathfrak{X} of pairs $(A, B[A])$ where $A, B[A] \subseteq X$ and $B[A]$ are interpreted as the off-springs that are generated from A . In the case of a genetic algorithm, for example, \mathfrak{X} encodes both mutational off-springs from individual sequences and pair-wise cross-over products. The collection \mathfrak{X} can be extended to a set-valued set-function $c : \mathcal{P}(X) \rightarrow \mathcal{P}(X)$ that describes for **each** subset A of X the collection of all possible offsprings, i.e., the set of points in X that are *accessible* from A by a single application of the search operator:

$$c(A) = \bigcup \{B[A'] \mid A' \subseteq A \wedge A' \neq \emptyset \wedge (A', B[A']) \in \mathfrak{X}\} \quad (3)$$

The condition $A' \neq \emptyset$ prohibits “spontaneous creation”, i.e., enforces $c(\emptyset) = \emptyset$. In the case of recombination operators on strings, for example, we have

$$c_{\mathcal{R}}(A) = \bigcup_{x, y \in A} \mathcal{R}(x, y). \quad (4)$$

Equ.(4) together with the definition of the linear span implies

$$c_{\mathcal{R}}(A) \subseteq \text{span}A \quad (5)$$

The function c defined in equ. (3) is a *closure function* in the sense of generalized topology. Indeed, following Kuratowski [31], topological spaces are often defined in terms of such a closure function instead of open or closed sets:

- (K0) $\text{cl}(\emptyset) = \emptyset$.
- (K1) $A \subseteq B$ implies $\text{cl}(A) \subseteq \text{cl}(B)$ (isotonic).
- (K2) $A \subseteq \text{cl}(A)$ (expanding).
- (K3) $\text{cl}(A \cup B) \subseteq \text{cl}(A) \cup \text{cl}(B)$ (sub-additive).
- (K4) $\text{cl}(\text{cl}(A)) = \text{cl}(A)$ (idempotent).

In our case, however, we can only verify that the first three axioms will hold. (K0) and (K1) follow directly from the construction of c from \mathfrak{X} . Such spaces are *isotone*. In addition, axiom (K2) is satisfied if and only if the parental genotypes *can* be transmitted to the next generation. This type of space is known as *neighborhood space*. Axiom (K3) holds for mutation only models, in which the set of off-springs are generated from a parent without regard to the rest of the population. In the finite case, spaces satisfying (K0) through (K3) are exactly the finite graphs. No good argument can be made for idempotency (K4) in our setting. The structure of search spaces thus is strictly weaker than that of topological spaces. The recombination closure space $(X, c_{\mathcal{R}})$, equ.(4) is an example of a finite neighborhood space. Such structure we recently studied as “ \mathfrak{N} -structures [29].

If X is finite, we can obtain a an idempotent function \bar{c} by repeated application of c :

$$\bar{c}(A) = c^N(A) = \underbrace{c(c(c(\dots(c(A)\dots)))}_{N \text{ times}} \quad (6)$$

for large enough N . (This construction works also in the infinite case, where N is in general an ordinal number.) In the most prominent cases of recombination operators, 1-point crossover and uniform crossover, it is not hard to verify that

$$\bar{c}_{\mathcal{R}}(A) = \text{span}A \quad (7)$$

In [33] it is shown that $\bar{c}_{\mathcal{R}}(A)$ is always a schema in the sense of Antonisse for the “usual” string recombination operators.

The idempotent closure function \bar{c} , in contrast to c , gives a rather coarse grained description of the search space. Furthermore, it is known that a meaningful not-trivial topological theory can be constructed without the (K4) axiom. Indeed, Eduard Čech [8] wrote a classical treatise of point set topology based on non-idempotent closure functions. Even more general spaces, lacking also the additivity assumption (K3) were also considered in the literature, see e.g. [23,24,10,5,18].

As in classical topology one can speak about the interior of a set ($I(A) = X \setminus c(X \setminus A)$) and of neighborhoods N of a point x ($N \in \mathcal{N}(x)$ iff $x \in I(N)$) or of a set A ($N \in \mathcal{N}(A)$ iff $A \subseteq I(N)$). Both the interior function I and the neighborhood systems \mathcal{N} of individual points can be used as alternative, equivalent, definitions of the same mathematical structures. We refer to [36,37] for details on this topic our previous papers for details on this topic. In the following discussion we will mostly avoid the use of interior and neighborhood functions.

2.4 Continuity

The notion of continuity lies at the heart of topological theory. Its importance is emphasized by a large number of equivalent definitions, see e.g. [26,19]. Let (X, cl) and (Y, cl) be two isotone spaces, i.e., spaces satisfying (K0) and (K1). Then $f : X \rightarrow Y$ is continuous if one (and hence all) of the following equivalent conditions holds:

- (i) $\text{cl}(f^{-1}(B)) \subseteq f^{-1}(\text{cl}(B))$ for all $B \in \mathcal{P}(Y)$.
- (ii) $f^{-1}(IB) \subseteq If^{-1}(B)$ for all $B \in \mathcal{P}(Y)$.
- (iii) $B \in \mathcal{N}(f(x))$ implies $f^{-1}(B) \in \mathcal{N}(x)$ for all $x \in X$.
- (iv) $f(\text{cl}(A)) \subseteq \text{cl}(f(A))$ for all $A \in \mathcal{P}(X)$.

3 Connectedness

3.1 Topological Connectedness

Topological connectedness is closely related to separation. Two sets $A, B \in \mathcal{P}(X)$ are *semi-separated* if there are neighborhoods $N' \in \mathcal{N}(A)$ and $N'' \in \mathcal{N}(B)$ such that $A \cap N'' = N' \cap B = \emptyset$. A set $Z \in \mathcal{P}(X)$ is *connected* in a space (X, cl) if it is not a disjoint union of nontrivial semi-separated pairs of non-empty sets $A, Z \setminus A$. There have been several attempts to use connectedness as the primitive notion in topological theory [44,25,27].

If (X, cl) is isotone (as we shall assume throughout this manuscript) then A and B are semi-separated if and only if $\text{cl}(A) \cap B = A \cap \text{cl}(B) = \emptyset$. Connectedness in isotonic spaces can thus be characterized by the Hausdorff-Lennes condition: A set $Z \in \mathcal{P}(X)$ is *connected* in an isotonic space (X, cl) if and only if for each proper subset $A \subseteq Z$ holds

$$[\text{cl}(A) \cap (Z \setminus A)] \cup [\text{cl}(Z \setminus A) \cap A] \neq \emptyset \quad (8)$$

The collection of connected sets satisfies the following three properties in isotonic spaces [36,20]:

- (c1) If Z consists of a single point, then Z is connected.
- (c2) If Y and Z are connected and $Y \cap Z \neq \emptyset$ then $Y \cup Z$ is connected
- (c3) If Z is connected and $Z \subseteq \text{cl}(Z)$, then $\text{cl}(Z)$ is also connected.
- (c4) Let I be an arbitrary index set and $x \in X$. Suppose Z_i is connected and $x \in Z_i$ for $i \in I$. Then $W := \bigcup_{i \in I} Z_i$ is connected.

As a short example of the formalism, we give here an elementary proof of property

(c4). We first observe the following simple fact: Suppose A and B are semi-separated and $A' \subseteq A$, $B' \subseteq B$; then A' and B' are also semi-separated. Now suppose W as defined above is not connected, i.e., there is a semi-separation $W = W' \dot{\cup} W''$. Assume w.l.o.g. $x \in W'$. Since the Z_i collectively cover W , there is a set Z_v such that $Z_v \cap W'' \neq \emptyset$. Since $x \in Z_v$ we also have $Z_v \cap W' \neq \emptyset$. Since W' and W'' are semi-separated, $Z_v \cap W'$ and $Z_v \cap W''$ are also semiseparated, i.e., $(Z_v \cap W') \dot{\cup} (Z_v \cap W'')$ is a semiseparation of Z_v , and hence Z_v is not connected, a contradiction.

In particular, a neighborhood space is connected if and only if it is not the disjoint union of two closed (open) sets [20, Thm.5.2]. This result generalizes the analogous well-known statement for topological spaces.

Now consider a set $A \subseteq X$ and a point $x \in A$ and define:

$$A[x] = \bigcup \left\{ Z \subseteq A \mid x \in Z \text{ and } Z \text{ is connected} \right\}, \quad (9)$$

i.e., $A[x]$ is the union of all connected subsets of A that contain x . By (c4), the set $A[x]$ is itself connected, i.e., $A[x]$ is the unique maximal connected subset of A that contains x . By construction, the collection of subsets $\{A[x] \mid x \in X\}$ defines a partition of the set A into maximal connected subsets. These sets are called the *connected components* of A . In particular, $X[x]$ is the connected component of x in the entire space.

The relationship of connected components and semi-separations is more complicated than one might guess. The following result for additive spaces matches our intuition:

Lemma 3 *Suppose $A \subseteq X$ has a finite number $k > 1$ of connected components and let $Q \subseteq A$ be a such a connected component. In a Pr's-topology, Q and $A \setminus Q$ are semi-separated.*

Proof. Observe that there is a semiseparation A' , $A \setminus A'$ since A is not connected. Suppose $Q \subseteq A'$. Now either $Q = A'$ or A' is not connected. In the latter case there is a set $A'' \subseteq A'$ such that $A'' \dot{\cup} (A' \setminus A'')$ is a semiseparation and $Q \subseteq A''$. Thus $c(A \setminus A') \cap A'' = \emptyset$, $c(A' \setminus A'') \cap A'' = \emptyset$, and by (K3) $c(A \setminus A'') \cap A'' = c((A \setminus A') \cup (A' \setminus A'')) \cap A'' = (c(A \setminus A') \cup c(A' \setminus A'')) \cap A'' = \emptyset$. Furthermore, $c(A'') \cap (A' \setminus A'') = \emptyset$ and $c(A'') \subseteq c(A')$ implies $c(A'') \cap (A \setminus A') = \emptyset$. $c(A'') \cap (A \setminus A'') = [c(A'') \cap (A' \setminus A'')] \cup [c(A'') \cap (A \setminus A')] = \emptyset$. We conclude that A'' and $A \setminus A''$ are semi-separated. Repeating this argument a finite number of times shows that we can “cut away” parts of A by means of semi-separations until we are left with Q . \square

This is *not* true in more general neighborhood spaces. Consider the closure space defined by 1-point crossover on strings of length 4 and consider the set $A = \{x = 0000, y = 0011, z = 1100\}$. We have the semi-separations $\{x\}|\{y\}$, $\{x\}|\{z\}$, $\{y\}|\{z\}$, $\{x, y\}|\{z\}$, $\{x, z\}|\{y\}$. Thus the connected components are

the isolated points. Nevertheless, $\{y, z\}, \{x\}$ is not a semi-separation since $x \in \mathcal{R}(y, z)$.

In fact, recombination alone does not lead to connected spaces at all.

Theorem 1 *The closure space $(X, c_{\mathcal{R}})$ is disconnected, i.e., $X[x] = \{x\}$ for all strings x and any string recombination operator \mathcal{R} .*

Proof. The search space is $X = \prod_{i=1}^n \mathcal{A}_i$, where \mathcal{A}_i is the alphabet (or set of alleles) for sequence position i , and consider an arbitrary point $x^* \in X$. Denote by $X_1 = \{(x_1^*, y_2, \dots, y_n) | y_i \in \mathcal{A}_i, i \geq 2\}$ the “hyperplane” defined by the first coordinate of x^* . Its complement is $X'_1 = X \setminus X_1 = \{(y_1, y_2, \dots, y_n) | y_1 \in \mathcal{A}_1 \setminus \{x_1^*\}, y_i \in \mathcal{A}_i, i \geq 2\}$. By construction, $X_1 \cap X'_1 = \emptyset$, $X_1 \cup X'_1 = X$, and $c_{\mathcal{R}}(X_1) \subseteq \text{span}X_1 = X_1$, $c_{\mathcal{R}}(X'_1) \subseteq \text{span}X'_1 = X'_1$. Thus $X_1 \dot{\cup} X'_1$ is a semiseparation of X . It follows, that the connected component of x^* is confined to X_1 , in symbols $X[x^*] \subseteq X_1$.

Now we define $X_2 = \{(x_1^*, x_2^*, y_3, \dots, y_n) | y_i \in \mathcal{A}_i, i \geq 3\}$ and $X'_2 = \{(x_1^*, y_2, y_3, \dots, y_n) | y_3 \in \mathcal{A}_3 \setminus \{x_3^*\}, y_i \in \mathcal{A}_i, i \geq 3\}$. As above, we have $X_2 \cap X'_2 = \emptyset$, $X_2 \cup X'_2 = X_1$, and $c_{\mathcal{R}}(X_2) \subseteq \text{span}X_2 = X_2$ and $c_{\mathcal{R}}(X'_2) \subseteq \text{span}X'_2 = X'_2$, i.e., $X_2 \dot{\cup} X'_2$ is a semiseparation of X_1 and hence $X[x^*] \subseteq X_2$. Repeating for X_k , the “hyperplane” defined by the first k coordinates of x^* , and its complement X'_k w.r.t. X_{k-1} , we obtain $X[x^*] \subseteq X_k$ and finally $X[x^*] \subseteq X_n = \{x^*\}$. \square

In order to meaningfully study connectedness in the context of Genetic Algorithms, we thus have consider connectedness for the closure operators that are derived from the superposition of mutation and crossover. The following statements may serve as examples.

Lemma 4 *Let (X, c_{GA}) be a closure space derived from point mutations and a string recombination operator \mathcal{R} . Then $\text{span}A$ is connected for all $A \subseteq X$. Furthermore, $\mathcal{R}(x, y)$ is connected if \mathcal{R} is uniform crossover or 1-point crossover.*

Proof. Let $x, y \in \text{span}A$. We can convert x into y by exchanging one character that differs between x and y after the other (say from left to right). The string obtained in each step differs in a single position from the previous one and is again contained in $\text{span}A$. It follows that for every $x, y \in \text{span}A$ there is path in $\text{span}A$ that leads with Hamming distance $d_H = 1$ steps from x to y . Thus $\text{span}A$ is connected in c_{GA} because it is connected w.r.t. to mutation contribution to c_{GA} alone. For uniform crossover, $\mathcal{R}(x, y) = \text{span}\{x, y\}$. In the case of 1-point crossover we consider the recombinants in the order in which they arise by crossover after position k . For $k = 1$, the two offsprings are either identical to the two parents, or differ by letter in the first sequence position from one of the parents, i.e., $d_H(x, x^1) \leq 1$ and $d_H(y, y^1) \leq 1$. The offsprings obtained from crossover at position k and $k + 1$ can be divided into two

pairs (x^k, x^{k+1}) and (y^k, y^{k+1}) with $d_H(x^k, x^{k+1}) \leq 1$ and $d_H(y^k, y^{k+1}) \leq 1$. For crossover before the end of the string we obtain $d_H(x^{n-1}, y) \leq 1$ and $d_H(y^{n-1}, x)$. Thus the recombination products of x and y are located on two paths connecting x and y in Hamming distance $d_H = 1$ steps, see also [17]. \square

3.2 Productive Connectedness

In [4] a less stringent definition of connectivity is introduced that is in particular suitable for chemical reaction networks.

We say that $A, B \in \mathcal{P}(X)$ are *productively separated* if for all $Z \subseteq A \cup B$ holds

- (1) $\text{cl}(Z \cap A) \cap B = \emptyset$ and $\text{cl}(Z \cap B) \cap A = \emptyset$
- (2) $\text{cl}(Z) = \text{cl}(Z \cap A) \cup \text{cl}(Z \cap B)$.

If (X, cl) is an isotonic space, then A and B are semi-separated if condition (1) holds for all $Z \subseteq A \cup B$.

It is now natural to call a set Z *productively connected* if it cannot be decomposed into two non-empty subsets Z' and $Z'' = Z \setminus Z'$ with $Z' \cap Z'' = \emptyset$ that are productively separated. In general, if Z is connected, then it is also productively connected. In pretopological spaces (and in particular in digraphs), semi-separation and productive separation coincide, hence Z is productively connected if and only if it is connected in this case.

Lemma 5 *Let $(X, c_{\mathcal{R}})$ be the closure space deriving from a proper recombination operator \mathcal{R} . Then a 2-point set $\{x, y\} \subseteq X$ is productively connected if and only if $d_H(x, y) \geq 2$.*

Proof. By (X3), $c_{\mathcal{R}}(x) = \{x\}$ for all $x \in X$. Thus $\{x, y\}$ is productively connected if and only if $c(\{x, y\}) \neq \{x, y\}$. By Lemma 2 this is the case if and only if $d_H(x, y) \geq 2$. \square

It follows immediately, that any subset $A \subseteq X$ is productively connected for any GA-closure c_{GA} that is derived from point mutation and a proper recombination operator. This implies that the notion of productive connectedness is too weak to be of much use for our purposes. We will therefore not consider productive connectedness in the following.

3.3 Path-Connectedness

Path-connectedness is a widely used notion of connectedness that in general is stronger than topological connectedness. From the topological point of view, a *path* is a *continuous* function $p : [0, 1] \rightarrow X$ whose endpoints are $p(0)$ and

$p(1)$. (Here the interval $[0, 1]$ is assumed to have the usual topology of real number.) A set A is *path-connected* if for any two points $x, y \in A$, there is a path p with $p(0) = x$ and $p(1) = y$. $A \subseteq X$ is path-connected if and only if for every pair of points $x, y \in A$ there is a path in A with endpoints x and y . One easily checks that the concatenation of two paths with $p_1(1) = p_2(0)$,

$$p_1 \bullet p_2(t) = \begin{cases} p_1(2t) & t \in [0, 1/2] \\ p_2(2t - 1) & t \in [1/2, 1] \end{cases} \quad (10)$$

is again a path. Conversely, if the restriction of a path p to an interval $[t', t''] \subseteq [0, 1]$ is again a path $p'(t) = p(t(t'' - t') + t')$. In the finite case, paths reduce to simple combinatorial objects, as we shall see in the following two results.

Lemma 6 *Let (X, c) be a neighborhood space and $\{x, y\} \subseteq X$ a 2-point subset. Then the following statements are equivalent:*

- (1) $y \in c(\{x\})$ or $x \in c(\{y\})$.
- (2) $\{x, y\}$ is path-connected.
- (3) $\{x, y\}$ is connected.

Proof. (i \Rightarrow ii) Suppose $y \in c(\{x\})$. Then $p : [0, 1] \rightarrow \{x, y\}$, $p(t) = x$ for $t \in [0, 1/2)$ and $p(t) = y$ for $t \in [1/2, 1]$ is continuous since $p(c([0, 1/2))) = p([0, 1/2]) = \{x, y\} \subseteq c(p([0, 1/2])) = c(\{x\}) = \{x, y\}$ and $p(c([1/2, 1])) = p([1/2, 1]) = \{y\} \subseteq c(p([1/2, 1])) = c(\{y\})$. Analogously, $p : [0, 1] \rightarrow \{x, y\}$, $p(t) = x$ for $t \in [0, 1/2]$ and $p(t) = y$ for $t \in (1/2, 1]$ is continuous if $x \in c(\{y\})$. (ii \Rightarrow iii) The continuous image of a connected set is connected [20, Thm.5.4], thus path-connectedness in general implies connectedness.

(iii \Rightarrow i) Since $\{x, y\}$ is connected, $\{x\}, \{y\}$ is not a semiseparation, i.e., $x \in c(\{y\})$ or $y \in c(\{x\})$. \square

Theorem 2 *Let (X, c) be a finite neighborhood space. Then there is a path from x to y in $A \subseteq X$, $x, y \in A$, if and only if there is a sequence of points $(x = x_0, x_1, \dots, x_{\ell-1}, x_{\ell} = y)$, $x_i \in A$, such that the two-point sets $\{x_{i-1}, x_i\}$ are connected.*

Proof. Since X is finite, we only need to consider paths p along which the function value $p(t)$ changes a finite number of times. Thus we can decompose $p = p_1 \bullet p_2 \bullet \dots \bullet p_{\ell}$ into sub-paths $p_i : [0, 1] \rightarrow \{x_{i-1}, x_i\}$ that connect subsequent function values, i.e., p is path if and only if each p_i is continuous. By Lemma 6 such a path with subsequent function values $x_0 = x, x_1, \dots, x_{\ell}$ exists if and only if each subset $\{x_{i-1}, x_i\}$ is connected. \square

With a finite neighborhood space (X, c) we can therefore associate a graph $\vec{\Gamma}$ with vertex set X and (directed) edges (x, y) if $y \in c(\{x\})$. If c is additive, then $\vec{\Gamma}$ is an equivalent representation of (X, c) . This correspondence between finite pretopologies and finite digraphs is discussed e.g. in [39]. In general,

we can use the graph $\vec{\Gamma}$ to represent path-connectedness in (X, c) . It follows directly from theorem 2 that a set $A \subseteq X$ is path-connected (w.r.t. c) if and only if the subgraph of $\vec{\Gamma}$ induced by A is connected.

We remark that, for an GA-closure deriving from point mutations and a string recombination operator, $\text{span}A$ is path-connected for all A . Furthermore, $\mathcal{R}(x, y)$ is path-connected for 1-point crossover.

Theorem 1 implies that in string-recombination-only closures, there are no connected two-point sets (since the connected components are the individual points). It follows that path-connectedness is determined by the mutation component of the GA.

4 Basins and Barriers

Consider a landscape (X, c, f) , where $f : X \rightarrow \mathbb{R}$ is an arbitrary function. We define the level-sets

$$X_\eta = \{x \in X \mid f(x) \leq \eta\} \quad (11)$$

Let \mathbb{P}_{xy} be the set of all paths from x to y . We say that x and y are *mutually accessible at level η* , in symbols

$$x \leftarrow \underline{\mathbb{P}}^\eta \rightarrow y, \quad (12)$$

if there is path $\mathbf{p} \in \mathbb{P}_{xy}$ such that $f(z) \leq \eta$ for all $z \in \mathbf{p}$, respectively. The path-connected components of X_η are therefore

$$P_\eta[x] = \{y \in V \mid y \leftarrow \underline{\mathbb{P}}^\eta \rightarrow x\} \quad (13)$$

So-called cycles play a central role in the theory of simulated annealing, see e.g. [6,7]. In the landscape setting “cycles” correspond to the connected components of the level sets. In the literature on “disconnectivity graphs”, the cycles are usually called “super-basins” [3]. More precisely, the *cycle of $x \in V$ at height η* , $C_\eta[x]$, is the *connected component* of the level set $\{y \in V \mid f(y) \leq \eta\}$ that contains x . In finite pretopological spaces, i.e., digraphs, we have $P[x] = C[x]$. In finite neighborhood spaces (and in infinite pretopologies), however, we only have $P[x] \subseteq C[x]$.

For simplicity of the following discussion, let us assume that X is finite and the landscape is non-degenerate (i.e., $f(x) = f(y)$ implies $x = y$). In the case, we may order the points in X by increasing cost, i.e., $f(x^{(i)}) < f(x^{(j)})$ iff $i < j$. Thus $x^{(1)}$ is the unique global minimum. We say that a point x is a *local minimum* if $C_{f(x)}[x] = \{x\}$. In the case of non-degenerate landscapes we would use a more complicated definition. For example, we might say that

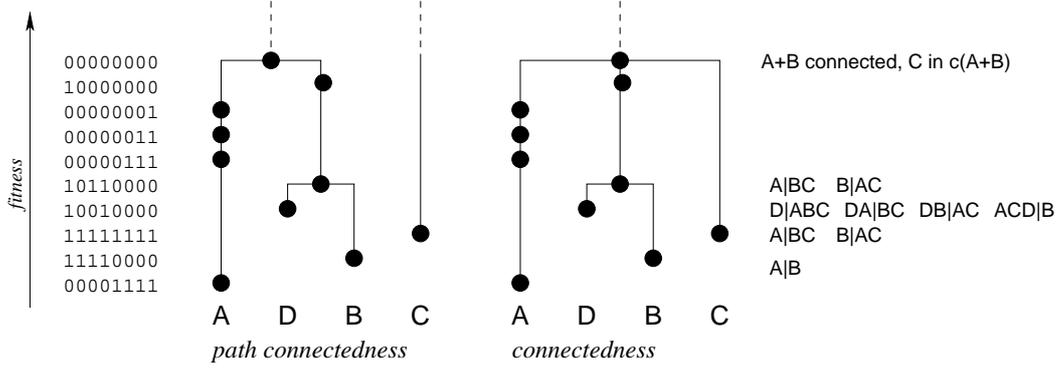


Fig. 1. Example of the lowest part of path-connectedness and connectedness barrier trees for a GA with point mutation and 1-point crossover. The first difference is the simultaneous connection between the connected components A, B, and C via the string 00000000. Just below the fitness level of 00000000, we have $A = \{00001111, 00000111, 00000011, 00000001\}$, $B = \{10000000, 10110000, 10010000, 11110000\}$, $C = \{11111111\}$. Note that A, B, and C are connected by means of point mutations alone.

00000000 connects A and B to a single connected component also via point mutations. Since $11111111 \in C$ is a recombination product of $00001111 \in A$ and $11110000 \in B$. Since at the level of 00000000, $A \cup B$ is connected and $C = \{11111111\}$ is contained in the closure $\text{cl}(A \cup B)$, there is no semi-separation in $A \cup B \cup C$, i.e., this set is connected.

In the right-most column, we show the complete list of all maximal semi-separations.

x is a local optimum if $f(y) = f(x)$ for all $y \in C_{f(x)}[x] : f(y) = f(x)$. The complications of degenerate landscapes are discussed in detail in [15] for the case of landscapes on finite graphs.

In the following we consider the relationship of the (path)connected components of the level sets X_η and $X_{\eta'}$ in some more detail:

Lemma 7 *Let A be a component of X_η and B a component of $X_{\eta'}$ with $\eta < \eta'$. Then either $A \subseteq B$ or $A \cap B = \emptyset$.*

Proof. Consider a point $x \in A$ and let A' be the component of $X_{\eta'}$ that contains x . Since $X_\eta \subseteq X_{\eta'}$, it follows that $A \subseteq A'$. Either $A' = B$, in which case $A \subseteq B$, or $A' \cap B = \emptyset$, in which case $A \cap B = \emptyset$. \square

Note that this result holds for all notions of connectedness, including topological connectedness, path-connectedness and productive connectedness.

Since f is non-degenerate, we may choose the difference between η and η' so that $X_{\eta'} = X_\eta \cup \{x^*\}$. The “new” point x^* will in general interfere with the connectedness structure of the “old” set X_η . It is important to notice that much of the connectedness structure of $X_{\eta'}$ is inherited from the old level

set X_η .

Theorem 3 *Suppose $X_{\eta'}$ and X_η , $\eta' > \eta$ are level sets of a fitness landscape and $X_{\eta'} \setminus X_\eta = \{x^*\}$. Let A' be a connected component of $X_{\eta'}$. Then either $x^* \notin A'$, in which case A' is a connected component of X_η , or $x^* \in A'$, in which case exactly one of the following three statements is true:*

- (1) $A' = \{x^*\}$ is a connected component of $X_{\eta'}$. In this case x^* is a local minimum.
- (2) There is a unique connected component A of X_η such that $A' = A \cup \{x^*\}$.
- (3) There are two or more connected components A_i , $i = 1, 2, \dots$, of X_η such that $A' = \cup_i A_i \cup \{x^*\}$. In this case we call x^* a saddle point.

Proof. If $X_{\eta'}$ is connected, there is no component that does not contain x^* . Otherwise there is a semi-separation, say $Q'_1 \dot{\cup} Q''_1$, of $X_{\eta'}$. Suppose Q''_1 contains x^* . Then $Q'_1 \dot{\cup} (Q''_1 \setminus \{x^*\})$ is also a semi-separation of $X_{\eta'}$. If $A' \subseteq Q'_1$ it follows immediately that A' is a connected component of Q'_1 and hence also of X_η . We repeat the argument with Q''_1 : Unless Q''_1 is connected, there is a semi-separation $Q'_2 \dot{\cup} Q''_2$ of Q''_1 . By the same argument as above, A' lies either in Q'_2 or in $Q''_2 \setminus \{x^*\}$, or it contains x^* . In the first case, A' is a connected component of Q'_2 and hence also of X_η . After a finite number of steps we have either identified A' as a connected component of X_η , or $A' = Q''_k$ is a connected set that contains x^* . In this case $A' = X_{\eta'}[x^*]$

A connected component A_i of X_η is also connected in $X_{\eta'}$. Thus either $A_i \subseteq X_\eta[x^*]$ or $A_i \cap X_\eta[x^*] = \emptyset$. It follows that $X_\eta[x^*] = \cup_{i \in I} A_i \cup \{x^*\}$ for a suitable finite index set I . The three cases in the statement of theorem correspond to $|I| = 0$, $|I| = 1$, and $|I| \geq 2$, respectively. \square

Note that this result is rather trivial in finite pretopologies (i.e., graphs). In this case the connected component of x^* in $X_{\eta'}$ is the union of all connected sets $A_i \cup \{x^*\}$, while lemma 3 guarantees that the remaining connected components of X_η are also connected components of $X_{\eta'}$. Unfortunately, this simple construction does not work in non-additive spaces.

Algorithmically, it seems to be useful to keep track not only of the connected components but also of the semi-separations between them. Since semi-separations are inherited by subsets, we can, conversely argue, that a semi-separation of $U \cup \{x\}$ is either of the form $\{x\}|U$ or there is semi-separation $U = U'|U''$ such that $U' \cup \{x\}|U''$ or $U'|U'' \cup \{x\}$ is a semiseparation. Here U , U' and U'' are unions of connected components.

In practise, one first checks whether the new point $\{x\}$ can be connected to one or more connected components via mutation. If so, there is no semi-separation between $\{x\}$ and these components. For the remaining candidates, it is sufficient to consider recombination. A candidate semi-separation of the form $U' \cup \{x\}|U''$ can be ruled out if there is either $\mathcal{R}(x, z) \cap U'' \neq \emptyset$ for some $z \in$

Algorithm 1 Barrier Trees with arbitrary search operators

```
1: procedure BARRIERS
2:    $\mathcal{L}^{act} \leftarrow \text{MERGE}(\mathcal{L}, x)$ 
3:   if  $\mathcal{L}^{act} = (\{x\})$  then
4:      $x$  is a local minimum
5:   else if  $\mathcal{L}^{act} = (\{x\} \cup A)$  then
6:      $x$  belongs to basin A
7:   else
8:      $x$  is a saddle point merge all  $A \in \mathcal{L}^{act} \setminus \{x\}$ 
```

U' or if there is $y, z \in U''$ such that $x \in \mathcal{R}(y, z)$. Note that, if $U' \cup \{x\} | U''$ can be ruled out as a semi-separation, then $V' \cup \{x\} | V''$ with $U' \subseteq V'$ and $U'' \subseteq V''$ is also not a semi-separation. Thus it is in particular sufficient to compute for every connected component A , whether x is a recombinant of A , which other connected components contain recombinants of x and members of A , and for every pair of components A' and A'' , whether x is recombinant of parents in A' and A'' , respectively. More precisely, $U' \cup \{x\} | U''$ is a semiseparation, if and only if (1) for all connected components $A \subseteq U'$ and $B \subseteq U''$, $A \cup \{x\} | B$ is a semiseparation and (2) $\{x\} | B' \cup B''$ is semiseparation for all $B', B'' \subseteq U''$. Two connected components of W belong to the same connected component of $W \cup \{x\}$ if and only if there is not semi-separation left that separates them.

Algorithm 1 summarizes the basic logic of computing barrier trees. In algorithm 2 we outline the steps that are necessary to update the collection \mathcal{L} of connected components when a single point x is added for the case of mutation/recombination operators. Our approach relies on updating the list \mathcal{S} of maximal semiseparations. These steps are independent of the details of closure function. In the case of recombination we use the $R(A, B \rightarrow C)$ to store the information whether there are parents $a \in A$ and $b \in B$ that give rise to an off-spring in connected component C . If one were consider search operators that construct off-springs from more than two parents, these data-structures would have to be modified accordingly. In Algorithm 3 we collect simplified presentations of higher-level procedure utilized in Algorithm 2.

In practice, we use a trie data structure to store the connected components s . This allows a more efficient check of $R(A, C \rightarrow \{x\})$. If the sum of the longest prefix of x in A and the longest suffix of x in C has at least the length of x , then x can be produced via recombination from members of the connected components A and C .

Fig. 2 shows barrier trees for landscapes of quadratic spin glasses with randomly generated interaction coefficients. At least for small instances, here 16 bits, examples of landscapes for which recombination changes the barrier structure are rare. Further computational studies will be necessary see if this finding is related to the fact that highly correlated landscapes are usually

Algorithm 2 Update Connected Components

▷ \mathcal{L} List of connected components
▷ \mathcal{S} List maximal semiseparations
▷ R 3-dimensional array $R(A, B \rightarrow C) \forall A, B, C \in \mathcal{L}$
▷ $\mathcal{S}(x)$ point mutation neighbors of x
▷ $\mathcal{R}(x, y)$ crossover neighbors of x and y

1: **procedure** MERGE
2: initialize \mathcal{L}, \mathcal{S} and R
 ▷ Update \mathcal{R}
3: **for all** $A, C \in \mathcal{L}$ **do**
4: $R(A, \{x\} \rightarrow C) = 1$ if $\exists a \in A, c \in C : c \in \mathcal{R}(a, x) \vee c \in \mathcal{M}(x)$
5: $R(\{x\}, A \rightarrow C) = R(A, \{x\} \rightarrow C)$
6: $R(A, C \rightarrow \{x\}) = 1$ if $\exists a \in A, c \in C : x \in \mathcal{R}(a, c) \vee x \in \mathcal{M}(a) \cup \mathcal{M}(b)$
 ▷ New list of semiseparations
7: $\mathcal{S}' \leftarrow \text{MAXSEMISEPARATIONS}(\{x\}|\mathcal{L})$
8: **for all** $(U'|U'') \in \mathcal{S}$ **do**
9: $\mathcal{S}' \leftarrow \mathcal{S}' \cup \text{MAXSEMISEPARATIONS}(\{x\} \cup U'|U'')$
10: $\mathcal{S}' \leftarrow \mathcal{S}' \cup \text{MAXSEMISEPARATIONS}(U'|\{x\} \cup U'')$
11: Remove duplicates and non maximal elements from \mathcal{S}'
 ▷ Update \mathcal{L}
12: $\mathcal{L}^{act} \leftarrow \mathcal{L}$
13: **while** $\exists U'|U'' \in \mathcal{L}^{act}$ with $\mathcal{L}^{act} \cap (U' \cup U'') \wedge U' \cap \mathcal{L}^{act} \neq \emptyset \wedge U'' \cap \mathcal{L}^{act} \neq \emptyset$
 do
14: **if** $\{x\} \in U'$ **then**
15: $\mathcal{L}^{act} \leftarrow U'$
16: **else**
17: $\mathcal{L}^{act} \leftarrow U''$
18: $X \bigcup_{A \in \mathcal{L}^{act}} A$
 ▷ Update R
19: $R(X, B \rightarrow C) \leftarrow \bigvee_{A \in \mathcal{L}^{act}} R(A, B \rightarrow C)$
20: $R(A, B \rightarrow X) \leftarrow \bigvee_{C \in \mathcal{L}^{act}} R(A, B \rightarrow C)$
21: **for all** $A \in \mathcal{L}^{act}$ **do**
22: **for all** $(B, C \in \mathcal{L})$
23: **remove** $R(A, B \rightarrow C), R(B, A \rightarrow C), R(B, C \rightarrow A)$
 ▷ Update \mathcal{S}
24: **for all** $U'|U'' \in \mathcal{S}'$ **do**
25: $\mathcal{S} \leftarrow \mathcal{S} \cup U'|U''$ **unless** $\exists A \in U' \cap \mathcal{L}^{act} \wedge \exists B \in U'' \cap \mathcal{L}^{act}$
26: $\mathcal{L} \leftarrow (\mathcal{L} \setminus \mathcal{L}^{act}) \cup X$
 return \mathcal{L}^{act}

efficiently searchable by means of hill-climbing along.

Algorithm 3 Update Connected Components

```
1: procedure REMOVESETS
2:   for all  $U'|U''$  do
3:     if  $\exists V', V'' \in \mathcal{S}'$  with  $V' \subset U' \wedge V'' \subset U''$  then
4:       remove  $V'|V''$  from  $\mathcal{S}'$ 
5:     if  $U' \subset V' \wedge U'' \subset V''$  then
6:       ignore  $U'|U''$ 

7: procedure ISSEMISEPARATION
    $U'|U''$  is a semiseparation if
    $\forall A, B \in U' \wedge C \in U'' : R(A, B \rightarrow C) = 0$ 
   and  $\forall A, B \in U'' \wedge C \in U' : R(A, B \rightarrow C) = 0.$ 

8: function MAXSEMISEPARATIONS( $U'|U''$ )
9:   if  $U'|U''$  is a semiseparation then
10:    return( $U'|U''$ )
11:  else
12:    subsequently delete each element from  $U'$  and  $U''$  and call
13:    MAXSEMISEPARATIONS if this candidate is not yet checked.
```

5 Discussion

In this contribution we have extended the notion of barrier trees to search spaces of arbitrary structure. Two complications arise beyond the realm of finite graphs: (1) There are several natural notions of connectedness, each of which appears as **the** most natural in different applications. For example, productive connectedness was introduced to properly describe chemical reaction networks, while topological connectedness appears as the natural framework to study Genetic Algorithms. (2) The non-additivity of the close function in non-graphical search spaces poses substantial algorithmic challenges in actually computing barrier trees.

In this contribution we have briefly described a prototypical approach that is, however, not practical for large problems since its runtime and memory requirements are quadratic in the size of the (part of the) search space under investigation. More efficient algorithms, or at least efficient heuristics to check connectedness will be required before this approach can be applied to interestingly large optimization problems.

Interestingly, path-connectedness, which is more stringent than topological connectedness, turns out to be equivalent to “connectedness via mutations only” in the context of genetic algorithms. This results provides an *a posteriori* justification of the approach by Halam and Prügel-Bennet to study the dynamics of GAs by mapping the populations onto a barrier tree based on Hamming-neighborhoods. It also suggests to use a comparison of path-connectedness and

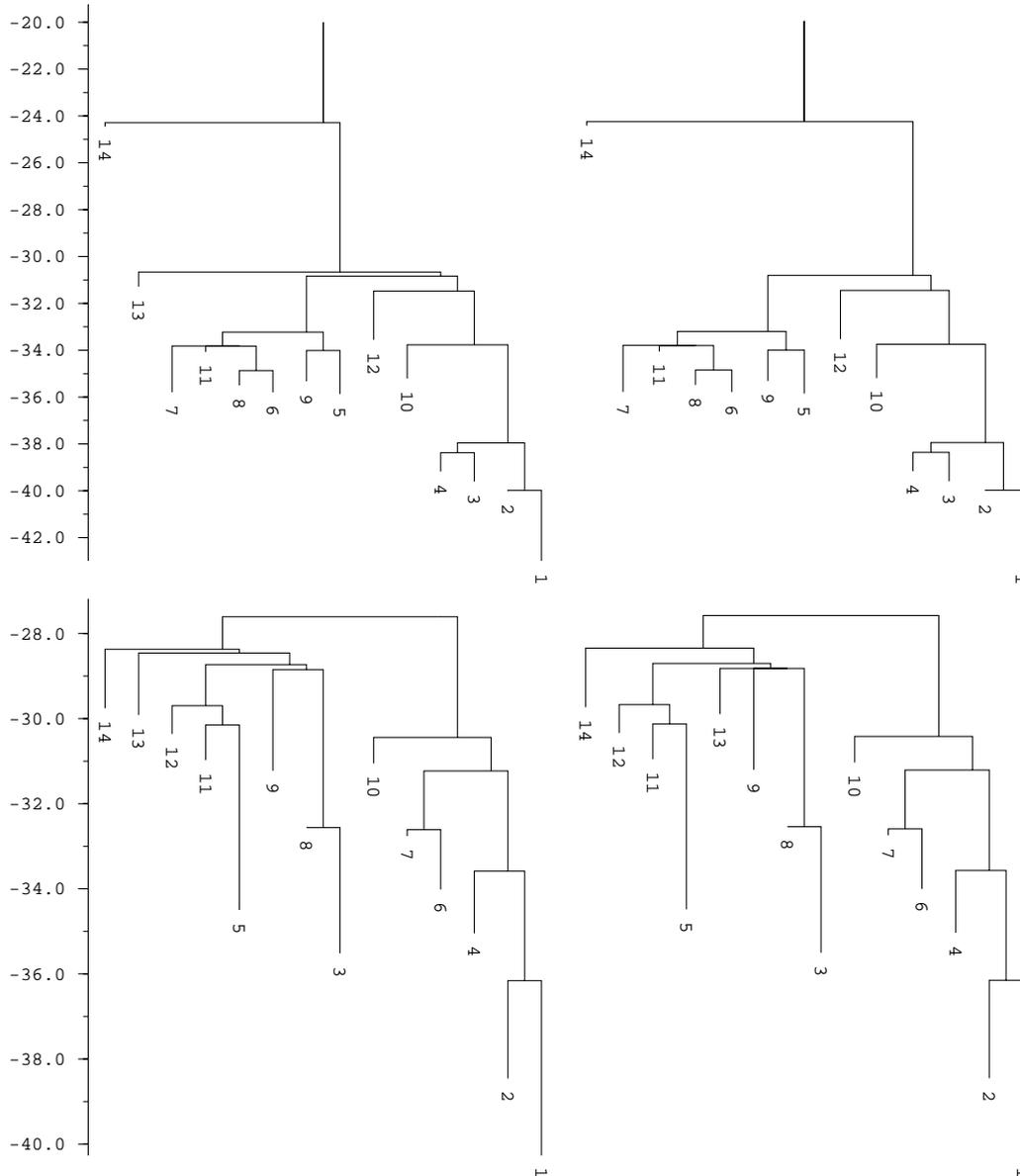


Fig. 2. Two Examples (rows) of barrier trees for quadratic spin glasses of size 16 without and with 1-point recombination (columns). In the first example (upper row) local minimum 13 is lost in the case of recombination because it can be produced from members of local minimum 1. In the second example, local minimum 13 can be connected to local minimum 3 at a lower energy level in the case of recombination. In both cases the remaining part of the landscape is unaffected.

topological connectedness trees to study the effects of crossover on a given landscape. By construction, the topological connectedness barrier tree is a homeomorphic image of the path-connectedness tree (since connected components are unions path-connected components). It follows that the effect of recombination appears as the collapsing of nodes and consequently as the re-

duction of of the number of local minima and their separating barrier heights. Intuitively, a recombination has big impact if many nodes are collapsed when going from path-connectedness to topological connectedness; conversely, recombination is not helpful when the two tree are (almost) the same.

Acknowledgments

This work was supported in part by the European Union FP-6 as part of the EMBIO project (<http://www-embio.ch.cam.ac.uk/>) and by the DFG Bioinformatics Initiative (BIZ-6/1-2).

References

- [1] J. Antonisse. A new interpretation of schema notation the overturns the binary encoding constraint. In *Proceedings of the Third International Conference of Genetic Algorithms*, pages 86–97. Morgan Kaufmann, 1989.
- [2] R. Azencott. *Simulated Annealing*. John Wiley & Sons, New York, 1992.
- [3] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *J. Chem. Phys.*, 106:1495–1517, 1997.
- [4] G. Benkő, F. Centler, P. Dittrich, C. Flamm, B. M. R. Stadler, and P. F. Stadler. A topological approach to chemical organizations. *Alife*, 2006. submitted.
- [5] M. M. Brissaud. Les espaces prétopologiques. *C. R. Acad. Sc. Paris Ser. A*, 280:705–708, 1975.
- [6] O. Catoni. Rough large deviation estimates for simulated annealing: Application to exponential schedules. *Ann. Probab.*, 20:1109–1146, 1992.
- [7] O. Catoni. Simulated annealing algorithms and Markov chains with rate transitions. In J. Azema, M. Emery, M. Ledoux, and M. Yor, editors, *Seminaire de Probabilites XXXIII*, volume 709 of *Lecture Notes in Mathematics*, pages 69–119. Springer, Berlin/Heidelberg, 1999.
- [8] E. Čech. *Topological Spaces*. Wiley, London, 1966.
- [9] M. Changat, S. Klavžar, and H. M. Mulder. The all-path transit function of a graph. *Czech. Math. J.*, 51:439–448, 2001.
- [10] M. M. Day. Convergence, closure, and neighborhoods. *Duke Math. J.*, 11:181–199, 1944.
- [11] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, NY, 2001.

- [12] J. P. Doye, M. A. Miller, and D. J. Welsh. Evolution of the potential energy surface with size for Lennard-Jones clusters. *J. Chem. Phys.*, 111:8417–8429, 1999.
- [13] M. Eigen and P. Schuster. *The Hypercycle*. Springer-Verlag, New York, Berlin, 1979.
- [14] C. Flamm, W. Fontana, I. Hofacker, and P. Schuster. RNA folding kinetics at elementary step resolution. *RNA*, 6:325–338, 2000.
- [15] C. Flamm, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216:155–173, 2002.
- [16] P. Garstecki, T. X. Hoang, and M. Cieplak. Energy landscapes, supergraphs, and “folding funnels” in spin systems. *Phys. Rev. E*, 60:3219–3226, 1999.
- [17] P. Gitchoff and G. P. Wagner. Recombination induced hypergraphs: a new approach to mutation-recombination isomorphism. *Complexity*, 2:37–43, 1996.
- [18] S. Gnilka. On extended topologies. I: Closure operators. *Ann. Soc. Math. Pol., Ser. I, Commentat. Math.*, 34:81–94, 1994.
- [19] S. Gnilka. On continuity in extended topologies. *Ann. Soc. Math. Pol., Ser. I, Commentat. Math.*, 37:99–108, 1997.
- [20] E. D. Habil and K. A. Elzenati. Connectedness in isotonic spaces. *Turk. J. Math*, 30:247–262, 2006.
- [21] J. Hallam and A. Prügel-Bennett. Barrier trees for search analysis. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Genetic and Evolutionary Computation — GECCO 2003, Part II*, volume 2724, pages 1586–1587, 2003.
- [22] J. Hallam and A. Prügel-Bennett. Large barrier trees for studying search. *IEEE Trans. Evol. Comput.*, 9:385–397, 2005.
- [23] P. C. Hammer. General topology, symmetry, and convexity. *Trans. Wisconsin Acad. Sci., Arts, Letters*, 44:221–255, 1955.
- [24] P. C. Hammer. Extended topology: Set-valued set functions. *Nieuw Arch. Wisk. III*, 10:55–77, 1962.
- [25] P. C. Hammer. Extended topology: Connected sets and Wallace separations. *Portug. Math.*, 22:77–93, 1963.
- [26] P. C. Hammer. Extended topology: Continuity I. *Portug. Math.*, 25:77–93, 1964.
- [27] J. M. Harris. Continuity and separation for point-wise symmetric isotonic closure functions. Technical Report 0507230, arXiv:math.GN, 2005.
- [28] W. Imrich and S. Klavžar. *Product Graphs: Structure and Recognition*. Wiley, New York, 2000.
- [29] W. Imrich and P. F. Stadler. A prime factor theorem for a generalized direct product. *Discussiones Math. Graph Th.*, 26:135–140, 2006.

- [30] T. Klotz and S. Kobe. “Valley Structures” in the phase space of a finite 3D Ising spin glass with $\pm i$ interactions. *J. Phys. A: Math. Gen.*, 27:L95–L100, 1994.
- [31] C. Kuratowski. Sur la notion de limite topologique d’ensembles. *Ann. Soc. Polon. Math.*, 21:219–225, 1949.
- [32] J. Maynard-Smith. Natural selection and the concept of a protein space. *Nature*, 225:563–564, 1970.
- [33] B. Mitavskiy. Crossover invariant subsets of the search space for evolutionary algorithms. *Evol. Comp.*, 12:19–46, 2004.
- [34] C. M. Reidys and P. F. Stadler. Combinatorial landscapes. *SIAM Review*, 44:3–54, 2002.
- [35] M. Shpak and G. P. Wagner. Asymmetry of configuration space induced by unequal crossover: implications for a mathematical theory of evolutionary innovation. *Artificial Life*, 6:25–43, 2000.
- [36] B. M. R. Stadler and P. F. Stadler. Generalized topological spaces in evolutionary theory and combinatorial chemistry. *J. Chem. Inf. Comput. Sci.*, 42:577–585, 2002. Proceedings MCC 2001, Dubrovnik.
- [37] B. M. R. Stadler and P. F. Stadler. The topology of evolutionary biology. In Ciobanu, editor, *Modeling in Molecular Biology*, Natural Computing Series, 267–286, 2004. Springer Verlag.
- [38] B. M. R. Stadler, P. F. Stadler, M. Shpak, and G. P. Wagner. Recombination spaces, metrics, and pretopologies. *Z. Phys. Chem.*, 216:217–234, 2002.
- [39] B. M. R. Stadler, P. F. Stadler, G. Wagner, and W. Fontana. The topology of the possible: Formal spaces underlying patterns of evolutionary change. *J. Theor. Biol.*, 213:241–274, 2001.
- [40] P. F. Stadler and C. Flamm. Barrier trees on poset-valued landscapes. *Genetic Prog. Evol. Mach.*, 7-20:4, 2003.
- [41] P. F. Stadler, R. Seitz, and G. P. Wagner. Evolvability of complex characters: Population dependent Fourier decomposition of fitness landscapes over recombination spaces. *Bull. Math. Biol.*, 62:399–428, 2000.
- [42] P. F. Stadler and B. M. R. Stadler. The genotype-phenotype map. *Biological Theory*, 2, 2006. in press.
- [43] P. F. Stadler and G. P. Wagner. The algebraic theory of recombination spaces. *Evol. Comp.*, 5:241–275, 1998.
- [44] A. D. Wallace. Separation spaces. *Ann. Math.*, pages 687–697, 1941.