Geometric Landscape of Homologous Crossover for Syntactic Trees

Alberto Moraglio
Department of Computer Science, University of Essex,
Wivenhoe Park, Colchester, CO4 3SQ, UK
amoragn@essex.ac.uk

Riccardo Poli
Department of Computer Science, University of Essex,
Wivenhoe Park, Colchester, CO4 3SQ, UK
rpoli@essex.ac.uk

Technical Report CSM-430 ISSN 1744-8050

Abstract- Geometric crossover and geometric mutation are *representation-independent* operators that are well-defined once a notion of distance over the solution space is defined. They were obtained as generalizations of genetic operators for binary strings and real vectors. Our geometric framework has been successfully applied to the permutation representation leading to a clarification and a natural unification of this domain. The relationship between search space, distances and genetic operators for syntactic trees is little understood. In this paper we apply the geometric framework to the syntactic tree representation and show how the well-known structural distance is naturally associated with homologous crossover and subtree mutation.

1 Introduction

Fitness landscape and solution representations

The notion of fitness landscape (Back et al, 1997) is very intuitive when the solution representation is a real vector of length two: it can be visualised as a 3D plot resembling the familiar notion of a geographic landscape, hence its name. When considering real vectors of higher dimensions, the landscape cannot be plotted, but its geographic meaning can be extended. A further extension is required when we consider the fitness landscape associated with binary strings; in this case the geography becomes discrete, the solution space is an n-dimensional cube and the landscape consists of a height function over such a structure. When dealing with binary strings and other more complicated combinatorial objects, such as permutations for example, the fitness landscape is better represented as a height function over the nodes of a simple graph (Reidys & Stadler, 2002), where nodes represent locations (solutions), and edges represent the relation of direct neighbourhood between solutions (syntactic similarity between solutions).

An abstraction of the notion of landscape encompassing all the previous cases (but not only) arises naturally leading to an axiomatic definition of solution space and consequently to an axiomatic definition of landscape. The solution space is seen as a metric space and the landscape as a height function over the metric space (Back et al, 1997). A metric space is a set endowed with a notion of distance among any pair of its elements fulfilling few axioms that are meant to be essential properties a distance function must have to be understood as such (Blumental & Menger, 1970). This abstraction is powerful in that encompasses both continuous spaces and discrete (combinatorial) spaces binding them through the abstract notion of distance. Specific spaces have specific distances that fulfil the metric axioms. The ordinary notion of distance associated with real vectors is the Euclidean distance, though there are other less-known possible options, like the family of Minkowski distances for example. The distance associated to combinatorial objects is normally the length of the shortest path between two nodes in the associated neighbourhood graph, the so called *shortest path distance* (Deza & Laurent, 1991). In the case of binary strings, the shortest path distance associated to the hypercube is the well-known Hamming distance.

In general, there may be more than one neighbourhood graph associated to the same representation, simply because there can be more than one meaningful notion of syntactic similarity applicable to the same representation (Moraglio & Poli, 2005). For example, in the case of permutations the adjacent element swap distance and the block reversal distance, to pick just two, are distinct but equally natural notions of distance arising from different types of syntactic similarity between permutations. The existence of different types of similarity between permutations is related to the fact that the same permutation (genotype) can be used to represent different types of solutions (phenotypes) in which different aspects of a permutation are meaningful. Permutations can represent solutions of a problem in which relative order of the elements in the permutation is important, in which case the adjacent element swap distance is a good measure of dissimilarity; permutations can also be used to represent TSP tours, in which case what is meaningful is the adjacency relationship among elements of the permutations and not their relative order, hence a better measure of dissimilarity is the block reversal distance.

Fitness landscape and genetic operators

The notion of fitness landscape was introduced in the field of evolutionary computation borrowing from population genetics. The reason behind the adoption of the fitness landscape metaphor is that evolutionary search, for both biological evolution and evolutionary algorithms, is better understood through the lens of the fitness landscape construct. Nevertheless, the notion of landscape results useful provided that the search operators employed are connected or matched with the landscape: the greater the connection the more landscape properties mirror search properties. This observation led Jones (Jones, 1995) to define the landscape as a function of the search operator employed hence leading to the one operator-one landscape paradigm. Whereas mutation is intuitively associated with

the neighbourhood structure of the search space, hence leading to a reasonably simple notion of landscape, crossover stretches it further leading to search spaces defined over complicated topological structures based on hyper-neighbourhoods (Jones, 1995).

In previous work (Moraglio & Poli, 2004) we introduced a representation-independent geometric generalization of genetic operators for binary string representation and real vector representation. The geometric definitions of mutation and crossover introduced are based on the distance associated with the search space, seen as a metric space, and on the simple geometric notions of ball and line segment. This way of connecting genetic operators and fitness landscape is the opposite of the standard approach introduced by Jones (above). Seeing genetic operators as functions of the search space produces a great deal of simplification and clarification: mutation and crossover share the same simple search space, that naturally corresponds to the classical notion of neighbourhood structure used by many meta-heuristics, and their relationship becomes clear.

Fitness landscape and problem knowledge

Since our definitions of genetic operators are generic and are connected neither with the solution representation nor with the problem at hand, it is important to understand how they relate with the NFL theorem (Wolpert & Macready, 1996). How can problem knowledge be specified and used by the formal evolutionary algorithm to perform better than random search? The key is the difference between problem and landscape, the former being given and the latter being designed. The landscape can be seen as a knowledge interface between formal algorithm and formal problem (Moraglio & Poli, 2005); through a domain-specific solution representation and a distance that makes sense for the problem at hand, one can easily and naturally design such a landscape by embedding problem knowledge codified in a geometric fashion.

In (Moraglio & Poli, 2005) we discussed three heuristics to embed problem knowledge in the landscape in a form usable by an evolutionary algorithm with geometric crossover. Actually, such heuristics suggest landscape conditions only indirectly, taking the form of guidelines to pick a sensible geometric crossover for a given problem. These heuristics are: pick a crossover associated to a good mutation, build a crossover using a neighbourhood based on the small-move small-fitness change principle, or build a crossover using a distance that is connected with a distance that is relevant for the solution interpretation. We tested our heuristics experimentally by designing new crossovers for the N-queens problem with permutation representation and found them corroborated.

The three heuristics above, in different forms, prescribe the same advice, that is, a crossover is likely to be good when is defined over a landscape that is smooth (in some statistical sense). This principle is not new to researchers working on meta-heuristics based on neighbourhood search. In fact, it is a good rule of thumb that has emerged experimentally to build the search space for most meta-heuristics (Pardalos & Resende, 2002), hence likely to apply to geometric crossover as well.

Evolutionary algorithms unification programme

Disregarding non-algorithmic differences, the various evolutionary algorithms differ only in the solution representation and the genetic operators (mutation and crossover) customized for the specific representation. A method to treat different representations uniformly is therefore a prerequisite for unification. What is mutation? What is crossover? What is common in all mutation operators and all crossover operators beyond the specific representation? In (Moraglio & Poli, 2004) we conjectured that a variety of operators developed for important representations, comprising binary strings, real-valued vectors, permutations and syntactic trees, fit our geometric definitions given suitable notions of distance (naturally not *all* pre-existing operators do this, but many do). Hence, our geometric framework has the potential to lead to a unification of the different evolutionary algorithms.

The importance of unification manifest itself in a number of surprising implications: (i) unveiling the geometric nature of evolutionary search (ii) simplifying and clarifying the connections between mutation, crossover, neighbourhood structure and fitness landscape (iii) giving rigorous general representation-independent definitions of genetic operators forming a solid ground for the generalization of pre-existing representation-specific theories and (iv) suggesting a easy and automatic way to do crossover principled design for any solution representation.

In this paper we add a new piece to the jigsaw puzzle of unification. After binary strings, real vectors and permutations, this time we consider syntactic trees.

The fitness landscape associated with genetic operators for syntactic trees is little understood. In this paper we intend to apply the geometric framework and clarify the notion of fitness landscape associated with homologous crossover for syntactic trees. More in detail our contributions are:

1. Application of the geometric framework (Moraglio & Poli, 2004) to the syntactic tree representation discussing the difference with other representations
2. Proof that the family of homologous crossovers (Langdon & Poli, 2002) for syntactic trees are geometric crossover under a family of structural edit distances
3. Clarification of the structure of the search space associated with structural distance
4. Proof that the natural mutation operator associated with homologous crossover and structural distance is the sub-tree mutation operator
5. We show that when syntactic trees are interpreted as GP programs, the structural distance between syntactic trees is also a meaningful distance between GP programs. Hence homologous crossover based on such a distance is a meaningful genetic operator for GP programs.

2 Geometric framework

2.1 Geometric preliminaries

In the following we give necessary preliminary geometric definitions and extend those introduced in (Moraglio & Poli, 2004) and (Moraglio & Poli, 2005) emphasizing the difference between graphic metric space and non-graphic metric space that turns out to be central to understand the peculiarities of the syntactic tree space we introduce later. The following definitions are taken from (Deza & Laurent, 1997).

Metric space and graphic metric space

A *metric space* (*M*, *d*) is a set *M* provided with a metric or distance *d* that is a real-valued map on $M \times M$ which fulfils the following axioms for all $s_1, s_2, s_3 \in M$:

1. $d(s_1, s_2) \geq 0$ and $d(s_1, s_2) = 0$ if and only if $s_1 = s_2$;
2. $d(s_1, s_2) = d(s_2, s_1)$, i.e. *d* is symmetric; and
3. $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$, i.e. *d* satisfies the triangle inequality.

A *graphic metric space* $M=(V, d_G)$ arises from a connected graph as follows: let $G=(V,E)$ be a connected graph and d_G denote the *path metric* of *G* where, for two nodes $i, j \in V$, $d_G(i, j)$ denotes the shortest length of a path from *i* to *j* in *G*. We say that *G* represents *M*. Graphic metric spaces have unique graph representation. We call *non-graphic metric space* any metric space that cannot be represented by a graph.

Similarly, a *metric space* can arise from a weighted graph as follows: if $G=(V,E)$ is a graph and $w = (w_e)_{e \in E}$ are strictly positive weights assigned to its edges, one can define the path metric $d_{G,w}(i, j)$ of the weighted graph (G,w). Namely, for two nodes $i, j \in V$, $d_{G,w}(i, j)$ denotes the smallest value of $\sum_{e \in P} w_e$ where *P* is a path from *i* to *j* in *G*. In general, a metric space induced by a weighted graph is *non-graphic* and has *more than one weighted-graph representation*. Two of them are the nearest-neighbors graph and the all-pairs graph, but there are many intermediate weighted graph representations.

Metric geometry

In classical Euclidean geometry, the measure of the distance between two points in the plane, say *A* and *B*, is calculated using the well known formula: $d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$. This is certainly a very intuitive notion of distance. By redefining the distance function between two points one obtains a new geometry for each distance redefinition. One simple example is the so-called 1st order Minkowski distance: $d(A, B) = |x_A - x_B| + |y_A - y_B|$. This definition of distance is fairly natural: it is the minimum distance that a taxicab would need to travel to reach point *B* from point *A*, if all streets are only oriented vertically and horizontally. For this reason, this metric is often referred to as the Manhattan metric. Many geometric figures, like circles, ellipses, parabolas, are defined in terms of distance. For instance, a circle is just the set of points with a fixed distance to the centre. These, of course, look quite different if we use a non-Euclidean measure of distance.

If we go further and say that a shape corresponds to a particular definition independently from the specific notion of metric used, we are then dealing with *abstract shapes* that are defined axiomatically and present abstract geometric properties that are shape-specific but not distance-specific. These abstract shapes are studied in *metric geometry*. Two of them, balls and segments, turn out to be very useful to define abstractly mutation and crossover and in the following we consider them in more detail.

Ball and segments

In a metric space (S, d) a *closed ball* is the set of the form $B(x; y) = \{ y \in S \mid d(x, y) \leq r \}$ where $x \in S$ and r is a positive real number called the radius of the ball. A *line segment* (or closed interval) is the set of the form $[x; y] = \{ z \in S \mid d(x, z) + d(z, y) = d(x, y) \}$ where $x, y \in S$ are called extremes of the segment. Note that $[x; y] = [y; x]$. The length *l* of the segment $[x; y]$ is the distance between a pair of extremes $l([x; y]) = d(x, y)$. Let *H* be a segment and $x \in H$ is an extreme of *H*, there exists only one point $y \in H$, its conjugate extreme, such that $[x; y] = H$.

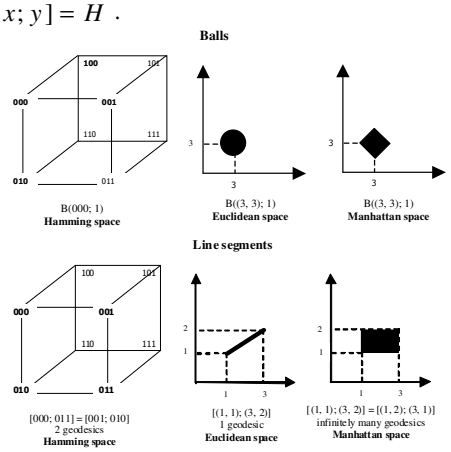

Fig. 1. Balls and segments for different spaces

Examples of balls and segments for different spaces are shown in Figure 1. Note how the same set can have different geometries (see Euclidean and Manhattan spaces) and how segments can have more than a pair of extremes. E.g. in the Hamming space, a segment coincides with a hypercube and the number of extremes varies with the length of the segment, while in the Manhattan space, a segment is a rectangle and it has two pairs of extremes. Also, a segment is not necessarily "slim": it may include points that are not on its boundaries. Finally, a segment does not coincide with a shortest path connecting its extremes (*geodesic*). In general, there may be more than one geodesic connecting two extremes.

Fitness landscape

We assign a structure to the solution *set* by endowing it with a notion of distance *d* respecting the axioms for a metric. *M=(S, d)* is therefore a solution *space* and *L=(M, g)* is the corresponding fitness landscape. Notice that *d* is an arbitrary distance and need not have any particular connection or affinity with the search problem at hand. However, in order to introduce problem knowledge in the search, one has to pick a distance that makes sense for the problem at hand.

2.2 Geometric operators definitions

We *define* two *classes* of operators in the landscape (i.e. using the notion of *distance* coming with the landscape): abstract mutation and abstract crossover. Within these classes, we identify two *specific* operators: abstract uniform mutation and abstract uniform crossover.

A g-ary genetic operator OP takes g parents $p_1, p_2, \ldots p_g$ and produces one offspring c according to a given conditional probability distribution: $f_{OP}(c \mid p_1, p_2, \ldots p_g)$. Mutation is a unary operator while crossover is typically a binary operator.

Definition 1 *The* image set *or accessibility of a genetic operator OP is the set of all possible offspring produced by OP with non-zero probability when the parents are* $p_1, p_2, \ldots p_g$*:*

$$\mathrm{Im}[OP(p_1, p_2, \ldots p_g)] = \{c \in S \mid f_{OP}(c \mid p_1, p_2, \ldots p_g) > 0\}$$

Notice that the image set is a *mapping* from a vector of parents to a set of offspring.

Definition 2 *A unary operator M is a abstract ε-mutation operator if* $\mathrm{Im}[M(p)] \subseteq B(p;\varepsilon)$ *where ε is the smallest real for which this condition holds true.*

In other words, in *a abstract ε-mutation* all offspring are at most *ε away* from their parent.

Definition 3 *A binary operator CX is a abstract crossover if* $\mathrm{Im}[CX(p_1, p_2)] \subseteq [p_1; p_2]$.

This simply means that in a abstract crossover offspring lay *between* parents. We use the term *recombination* as a synonym of any binary genetic operator.

We now introduce two *specific* operators belonging to the *families* defined above.

Definition 4 *Abstract uniform ε-mutation UM is a abstract ε-mutation where all z at most ε away from parent x have the same probability of being the offspring:*

$$f_{UM\varepsilon}(z \mid x) = \frac{\delta(z \in B(x,\varepsilon))}{|B(x,\varepsilon)|}$$

$$\mathrm{Im}[UM_{\varepsilon}(x)] = \{z \in S \mid f_{UM\varepsilon}(z \mid x) > 0\} = B(x,\varepsilon)$$

where δ *is a function which returns 1 if the argument is true , 0 otherwise. When ε is not specified, we mean ε = 1.*

Definition 5 *Abstract uniform crossover UX is an abstract crossover where all z laying between parents x and y have the same probability of being the offspring:*

$$f_{UX}(z \mid x, y) = \frac{\delta(z \in [x, y])}{|[x, y]|}$$

$$\mathrm{Im}[UX(x, y)] = \{z \in S \mid f_{UX}(z \mid x, y) > 0\} = [x, y].$$

These definitions are representation-independent and therefore the operators are well-defined for any representation.

2.3 Uniqueness results for graphic distance

Theorem 1 *The structure over the configuration space C can equivalently be defined by the set G of the syntactic configurations and one of the following objects: 1. The neighborhood function Nhd, 2. The neighborhood graph W= (V, E), 3. The graphic distance function d, 4. Uniform topological mutation UM, 5. Uniform topological crossover UX, 6. The set of all balls* ***B****, 7. The set of all segments* ***H****.* (See (Moraglio & Poli, 2004) for proofs)

Corollary 1 *Uniform topological mutation UM and uniform topological crossover UX are isomorphic.*

Corollary 2 *Given a structure of the configuration search space in terms of neighborhood function or graphic distance function, UM and UX are unique.*

Corollary 3 *Given a representation, there are as many UM and UX operators as notions of graphic/syntactic distance for the representation.*

3 Homologous crossover, hyperschemata and structural distance for trees

Let us now consider the representation of interest of this paper, trees, and a particular class of crossover operators for trees, homologous crossovers.

3.1 Subtree Swap Crossover & Homologous Crossover

The common region is the largest rooted region where the parent trees have the same topology. Figure 2 shows the common region for two trees with the same structure (left side) and with different structures (right side).

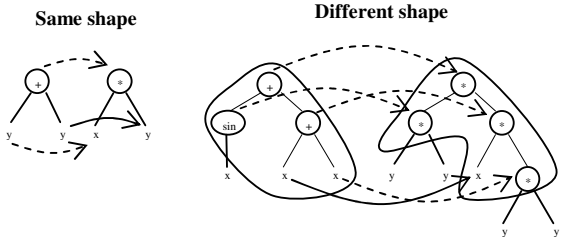

Figure 2 common region

In *homologous crossover* (Langdon & Poli, 2002) parent trees are aligned at the root and recombined using a crossover mask over the common region. If a node belongs to the boundary of the common region and is a function then the entire sub-tree rooted in that node is swapped with it. One special case of homologous crossover is one-point crossover (figure 3) in which two parent trees are aligned at the root, a crossover point picked randomly on a edge belonging to the common region (bold edges) and then the two sub-trees beneath the crossover point exchanged producing two offspring trees. Subtree swap crossover (Koza, 1992) is similar to one-point crossover but less restrictive: any subtree of one parent can be exchanged

with any subtree of the other parent to produce two offspring trees.

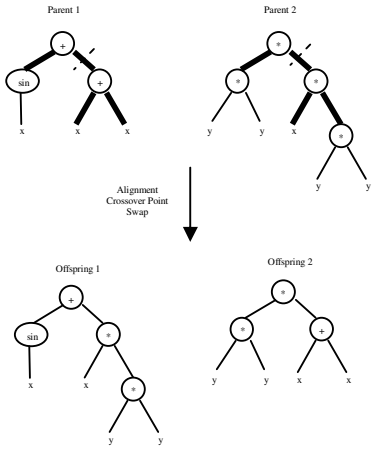

Figure 3 one-point crossover

3.2 Non-existence geometric crossover theorems

Theorem 2. *Subtree swap crossover is not a geometric crossover.*

Proof

For any metric, when two extremes of a segment are the same point the segment contains only that point. Subtree swap crossover applied to two copies of the same parent tree may produce offspring trees different from it. Consequently offspring trees cannot be in the segment between parent trees for any distance. So, subtree swap crossover is not geometric for any distance because it may produce offspring outside the image set of any geometric crossover operator ■

Theorem 3. *Homologous crossover is not geometric crossover under graphic distance.*

Proof

If by absurd homologous crossover were geometric under graphic distance then the edges of the unique graph representing the graphic distance associated with it would coincide with the segments of length 1 including only their two extremes.

Let us consider the image sets under homologous crossover. The image set obtained by crossing over any tree with a tree consisting of only a node is either a set comprising the single node tree or the set comprising the two parent trees only. This means that if the homologous crossover were associated to a graphic distance in the associated graph there would be an edge connecting any tree to a tree with any single node tree. In terms of associated distance we have only four possible cases: (i) distance zero, coinciding extremes, segment containing only the extreme; (ii) distance one, one extreme is single node tree and the other any other tree, segment containing only these two trees; (iii) distance one, the two extremes are not single node trees, segment containing only these two trees; (iv) distance two, the two extremes are not single node trees, segment may contain any trees but must contain all single node trees. This is because when the distance between two trees is two there is a shortest path between the two trees passing on a single node tree. Notice that when the distance between two trees is two, to be graphic, the segment between the two trees must contain a tree that differs from the extreme trees. Distance two is the maximum distance between to trees because is the maximum length of the shortest path connecting any two trees passing through a single node tree.

The image set obtained by crossing over two trees using homologous crossover may contain, beside the two parent trees, one or more offspring trees and do not need to contain any single node tree. In this case it happens that the distance between the two tree parents must be two, but that there is no single node tree on the shortest path between these two trees, hence there is incongruence with condition (iv) above and the homologous crossover cannot be associated with a graphic distance ■

Theorem 2 tells us that there is no distance naturally associable with subtree swap crossover. At least not in the same sense the Hamming distance is associated with mask-based crossover for binary strings. See (Gustafson & Vanneschi, 2005) for a completely different perspective on a distance for this operator.

Theorem 3 about the homologous crossover leaves open two alternatives: either homologous crossover is not a geometric crossover or homologous crossover is a geometric crossover based on a non-graphic metric space. If we find at least one distance that matches homologous crossover, then we know that homologous crossover is a geometric crossover and that the distance we found is a non-graphic distance. In section 3.3 we propose a variation on the well-known structural distance (Ekárt & Németh, 2000) and prove that is a metric. In section 3.4 we show that homologous crossover is geometric crossover under such a distance. This makes of homologous crossover the first non-graphic geometric crossover we meet. Being geometric crossover non-graphic there are various consequences. We analyse them is section 4 comparing the properties of operators arising from graphic metric spaces with those of operators arising from non-graphic metric spaces.

3.3 Structural Distance and Hyperschemata

(Ekárt & Németh, 2000) defined an edit distance specific to genetic programming syntactic trees, adapted from (Nienhuys-Cheng, 1997), which considered the cost of substituting between different node types (functions vs. terminals and within these classes). Two trees are brought to the same tree structure by adding ``null'' nodes to each tree. The differences near the root have more weight. The cost of changing one node into another can be specified for each pair of nodes or for classes of nodes.

In the following we propose a structural distance for GP trees and we prove that is a non-graphic distance for homologous crossover. We call this distance *normalized structural hamming distance (SHD)*.

$$dist(T_1,T_2)=\begin{cases}\delta(p\neq q) & if\ arity(p)=arity(q)=0\\ 1 & if\ arity(p)\neq arity(q)\\ \dfrac{1}{m+1}\left(hd(p,q)+\sum_{i=1,m}dist(s_i,t_i)\right) & if\ arity(p)=arity(q)=m\end{cases}$$

Characteristics of this distance:

- The maximum distance between two trees is 1
- When two subtrees are not comparable (roots of different arities) they are considered to be at a maximal distance
- When two subtrees are comparable their distance is at most 1

The following proof is a variation of the one given in (Nienhuys-Cheng, 1997) theorem 5.

Theorem 4. *SHD is a metric strictly bounded by 1.*

Proof

SHD bounded by 1: we prove it by induction. It is clear that $dist(S,T) \leq 1$ when the arities of root nodes p and q of T and S are either both 0 (p and q are leaves) or different. Now suppose T and S have equal non-zero arities:

$$dist(S,T) = \frac{1}{m+1}\left(hd(p,q) + \sum_{i=1,m} dist(s_i,t_i)\right)$$ and suppose

$dist(s_i,t_i) \leq 1, \forall i$ (induction hypothesis). Then since $hd(p,q) \leq 1$ we have

$$dist(S,T) \leq \frac{1}{m+1}\left(1 + \sum_{i=1,m} 1\right) = \frac{m+1}{m+1} = 1$$

***SHD is a metric*:**

identity: $dist(S,T) = 0 \leftrightarrow S = T$

(i) if $S=T$ then $dist(S,T)=0$. This is true because recursively the distance between all coupled subtrees of S and T is 0.

(ii) $dist(S,T)=0$ implies that the item 2 in the definition of *dist* must not apply to any paired nodes otherwise the distance among two nodes becomes non-zero and consequently the distance of the whole trees becomes non-zero as well. Since for every paired nodes the trees S and T have the same arity then S and T have the same structure. It is easy to see that two trees with the same structure have $dist(S,T)=0$ if and only if $hd(p,q)=0$ for any paired nodes p and q i.e. $p=q$.

symmetry: $dist(S,T)=dist(T,S)$ is trivially true because *dist* is defined using symmetric functions.

triangular inequality:

$dist(R,S) + dist(S,T) \geq dist(R,T)$

We prove it by induction on the depth of the tree.

Base case: suppose $depth(R)=depth(S)=depth(T)=0$, so R, S and T have roots of arity zero. The triangular inequality holds in this case because *dist* degenerates to the hamming distance between roots for which the triangular inequality holds.

Induction hypothesis: suppose the triangular inequality is true if the depth of R, S and T is at most k. Verify induction implication: we now assume the tree among R, S and T that has the greatest depth, has depth $k+1$. Let us consider in the following all possible cases.

- $Arity(root(R)) \neq arity(root(T))$: in this case $dist(R,T)=1$. We have two sub-cases: (i) $arity(root(R)) \neq arity(root(S))=arity(root(T))$ in which case $dist(R,S)=1$ and the triangular inequality holds; (ii) $arity(root(R)) \neq arity(root(S))$ and $arity(root(S)) \neq arity(root(T))$ in which case $dist(R,S)=1$ and $dist(S,T)=1$ so that the triangular inequality holds.
- $Arity(root(R))=arity(root(T))=0$: in this case $dist(R,T)=hd(R,T) \leq 1$. We have two sub-cases: (i) $arity(root(R))=arity(root(S))=arity(root(T))=0$, in which case *dist* degenerates to hamming distance and the triangular inequality holds; (ii) $arity(root(S))>0$, in which case $dist(R,S)=dist(S,T)=1$, hence the triangular inequality holds.
- $arity(root(R))=arity(root(T))=m>0$ *and* $arity(root(S)) \neq m$: in this case $dist(R,T) \leq 1$ and $dist(R,S)=dist(S,T)=1$ because the root node of S in diverse in arity hence not comparable with R and T. Hence the triangular inequality holds.
- $arity(root(R))=arity(root(S))=arity(root(T))=m$:

$$dist(R,S) + dist(S,T) = \frac{1}{m+1}\left(hd(R,S) + \sum_{i=1,m} dist(r_i,s_i)\right) +$$

$$+ \frac{1}{m+1}\left(hd(S,T) + \sum_{i=1,m} dist(s_i,t_i)\right) =$$

$$= \frac{1}{m+1}\left(hd(R,S) + hd(S,T) + \sum_{i=1,m} (dist(r_i,s_i) + dist(s_i,t_i))\right)$$

since $hd(R,S) + hd(S,T) \geq hd(R,T)$ and for induction hypothesis $dist(r_i,s_i) + dist(s_i,t_i) \geq dist(r_i,t_i)$ then

$$dist(R,S) + dist(S,T) \geq \frac{1}{m+1}\left(hd(R,T) + \sum_{i=1,m} dist(r_i,t_i)\right) = dist(R,T)$$

■

3.4 Geometric Crossover Theorems

Theorems 5 and 6 associate bijectively the class of homologous crossovers with the normalized structural hamming distance introduced in the previous section. The proofs of theorems 5 and 6 are based on the property that the SHD distance between two trees is only function of the hyperschema (Langdon & Poli, 2002) associated with the two trees and not directly of the two trees (figure 4).

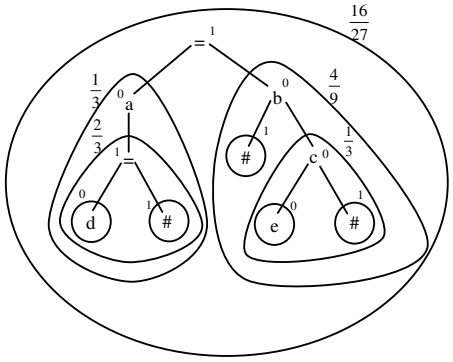

Figure 4 hyperschema and structural distance

The hyperschema associated with two trees is the tree structure that has the topology of the common region of the two trees; its nodes are '=' when two matched nodes differ in the content, or '#' replacing two subtrees whose roots are matched but their arities differ, or any other content when it is the same in both matched nodes. Figure 5 illustrates the relation between parent trees, hyperschema and offspring trees and shows: at the top, two parent trees P1 and P2; at the bottom on the left, their associated

hyperschema H(P1,P2); at the bottom on the right, all the potential offspring applying homologous crossover to parents P1 and P2 (the part in bold means alternative content of the tree; in this case there are 5 independent binary alternatives, resulting in 32 possible offspring).

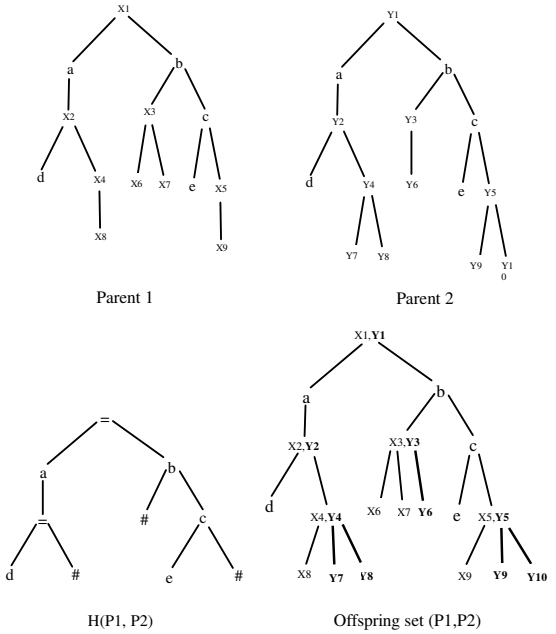

Figure 5 hyperschema and children set

Theorem 5. *Homologous crossover is a geometric crossover under (normalized) structural hamming distance (SHD).*

Proof

Remark 1: as shown in figure 4, the distance between two trees *P1* and *P2* is function *d* of the hyper-schema *H(P1,P2)* identified by the two trees: *SHD(P1,P2)=d(H)*

Remark 2: every offspring of two trees is obtained by substituting each wildcard characters in the hyper-schema with a node (=) or a sub-tree (#) coming either from one parent or from the other at that specific position

Remark 3: be *p1,..,pn* the positions in the structure of *H* of the wildcard characters. Then the distance *d(H)* can be decomposed into a sum of distances that are only functions of the positions of the wildcard characters in the tree: *d(H)=d(p1)+…+d(pn)*

Remark 4: be *O* the offspring of *P1* and *P2*. Then the hyper-schema *H(P1,O)* is obtainable by turning some wildcard characters in *H(P1,P2)* to corresponding nodes/sub-trees from parent P1. The hyper-schema H(O,P2) is obtainable by turning the wildcard characters in H(P1,P2) left untouched to corresponding nodes/sub-trees from parent P2

Remark 5: the positions of wildcard characters in *H(P1,O)*, say *{p(i)}*, and in *H(O,P2)*, say *{p(j)}*, are complementary, which is there is no *i* and *j* such as *p(i)=p(j)*, and taken all together are the same as in *H(P1,P2)*, which is $\{p(i)\} \cup \{p(j)\} = \{p1,\ldots,pn\}$

Remark 6: Hence:

d(H(P1,O))+d(H(O,P2))=d({p(i)})+d({p(j)})=sum{d(p(i))}+sum{d(p(j))}=sum{d(p1),…,d(pn)}=d({p1,…,pn})=d(P1,P2). This means that every offspring O of P1 and P2 is in the segment between P1 and P2 under dist ■

Theorem 6. *The image set of the class of homologous crossover is the segment between the two parent trees under SHD.*

Proof

We need to prove that *O* in *[P1,P2]* implies *O* in the image set of homologous crossover *R(P1,P2)*. Let us assume by absurdum that *O* in *[P1,P2]* but not in *R(P1,P2)*. Then *d(P1,O)+d(O,P2)=d(P1,P2)* and either (i) *O* matches *H(P1,P2)* but does not take the node/sub-tree of either parents at (at least) one position in H or (ii) *O* does not match *H(P1,P2)*. In case (i) the positions of wildcard characters in *H(P1,O)* and *H(O,P2)* are not complementary but their union still equals the positions of the wildcards in *H(P1,P2)*. This means that some of the wildcard positions are present in both *{p(i)}* and *{p(j)}* implying that the sum of the associated distances is greater than the distance associated with their union; hence *d(P1,O)+d(O,P2)>d(P1,P2)*. In case (ii) there are two sub-cases: (a) *O* does not match *H(P1, P2)* but matches its structure; this happens when some nodes of *O* do not match the corresponding non-wildcard characters in *H*. (b) *O* does not match the structure of *H(P1, P2)*; this happens when some nodes of *O* do not have the same arity of the corresponding node in *H*. In sub-case (a) the positions *{p(i)}* and *{p(j)}* of the wildcard characters in *H(P1,O)* and *H(O,P2)* respectively, both contain the positions of the mismatch with *O* plus, each one, a complementary bipartition of the set of positions *{p1,…,pn}*. Since the distance associated with H(P1,P2) is additive function of the set of positions *{p1,…,pn}* and the union of *{p(i)}* and *{p(j)}* is a proper subset of *{p1,…,pn}* then the sum of the distances associated with *H(P1,O)* and *H(O,P2)* is greater than the one associated with *H(P1,P2)*. In the sub-case (b) *O* differs (in arity) at certain position form both parents hence *H(P1,O)* and *H(O,P2)* are obtained by, first, pruning *H(P1,P2)* at the node in which *O* differs in arity and put a wildcard character and, then, substituting some of the wildcards with some nodes/sub-trees at the corresponding position from *P1* and *P2* respectively. The pruning of *H(P1,P2)* always produces an hyper-schema which associated distance is greater or equal to the one associated to the *H(P1,P2)* un-pruned. This is because the weight associated with a wildcard substituting a sub-tree is an upper-bound of the contributions of the sum of the weights of any possible sub-tree put in that position. The positions of the wildcards in *H(P1,O)* and *H(O,P2)* are complementary except for the wildcard attached at the position of the pruned tree, that appears in both trees. This wildcard therefore contributes to both the distances associated with *H(P1,O)* and *H(O,P2)* and being un upper bound of the contributions in the sub-tree of *H(P1,P2)* it replaces, we have *H(P1,O)+H(O,P2)>H(P1,P2)* also in this last case ■

3.5 Analysis of the normalization coefficient

In the previous two sections we have introduced the SHD metric and shown that it matches the homologous crossover. Here we want to analyse the effect of varying the normalization coefficient in terms metricity of the resulting distance and of the matching with crossover.

Reason behind the choice of the coefficient

The normalizing value 1/(m+1) has been chosen to have strict bound at 1 and consistency in the distance between two fully different subtrees in two senses: (i) any two sub-trees that are fully comparable (because they have the same structure) and differ in all nodes must have distance 1 (ii) any two subtrees that are incomparable must have distance 1. For smaller positive values than 1/(m+1), SHD is still a metric but the bound to 1 for the distance is never reached. This would give greater distance to subtrees that are non-comparable than to subtrees that are fully-comparable but differ in all nodes. For little greater values than 1/(m+1), SHD is still a metric, not bounded by one, but still bounded. This would give greater distance to subtress that are fully-comparable but differ in all nodes that to those sub-trees that are non-comparable. For increasing values between 1/(m+1) and 1 there is a critical value for which SHD ceases to be a metric. In the following we consider SHD with coefficient 1, that we call Hamming distance between syntactic trees, and prove that such a distance is not a metric.

Hamming distance for syntactic trees

The Hamming distance (HD) between two syntactic trees can be also seen as the number of mismatching nodes at corresponding positions within their common region.

Theorem 7*. The Hamming distance for syntactic trees is not a metric*

Proof

Let us consider three syntactic trees, T1, T2 and T3. T1 consists only of a single terminal node. T2 and T3 have the same shape and size n but they differ in all matching nodes. If HD is a distance the triangular inequality must hold for any choice of T1, T2 and T3. In the specific case of our example the following must hold $HD(T2,T1) + HD(T1,T3) \geq HD(T2,T3)$. Since we have HD(T2,T1)=1, HD(T1,T3)=1 and HD(T2,T3)=n, it is immediate to see that for n>2 the triangular inequality fails to hold. Hence, HD is not a metric ■

Many distances for homologous crossover

Looking at the proofs of theorem 5 and 6, it is easy to see that they work for any positive value of the coefficient less than 1/(m+1). So there is a whole family of distances that matches homologous crossover.

4 Graphic space vs non-graphic space

In the previous section we have proven that homologous crossover is geometric but non-graphic, that is, it is not associable to a graphic distance. This has some consequences on the geometric framework, since the results of uniqueness regarding space structure, distance, mutation and crossover (theorem 1 and corollaries 1, 2 and 3) are for graphic distances and do not necessarily hold for non-graphic distances. In the following we present the picture for non-graphic spaces and discuss the differences with that for graphic spaces.

Table 1 summarises the cardinality of the relations between solution representation, neighbourhood structure, distance, geometric mutation and geometric crossover in the case of graphic spaces. Table 2 does the same for the case of non-graphic spaces and operators. The items written in bold emphasise the differences with table 1.

Table 1 - graphic space and graphic operators

Graphic	representation	structure	distance	mutation	crossover
Representation	-	many	many	many	many
Structure	many	-	1	1	1
Distance	many	1	-	1	1
Mutation	many	1	1	-	1
Crossover	many	1	1	1	-

Table 2 - non-graphic space and non-graphic operators

Non-graphic	representation	**weighted** structure	distance	mutation	crossover
Representation	-	many	many	many	many
W. structure	many	-	1	1	1
Distance	many	**many**	-	1	1
Mutation	many	**many**	**many**	-	**many?**
Crossover	many	**many**	**many**	**many?**	-

In table 1 and 2, the rows labelled "representation" coincide and tell us that to a solution representation may be associated: (i) more than one neighbourhood structure (imagine neighbourhood structures induced by different edit moves acting on the same representation); (ii) more than one distance because each neighbourhood structure induces a path metric; and consequently (iii) more than one type of geometric mutation operator and (iv) more than one type of geometric crossover operator because, geometric operators are functions of the distance, and so there are as many types available as the distances for the same representation.

The rows labelled (neighbourhood) "structure" are also the same in table 1 and 2. This tells us that there can be different representations that induce the same neighbourhood structure (a trivial example: instead of using a binary string with 0-1, one may use another representation that makes use of the symbols *a* and *b* and as a result one gets the same neighbourhood structure provided that the symbol-flip edit move is used); the three 1's in the row are due to the fact that distance, mutation and crossover are functions of the neighbourhood structure and this does not depend on the type of underlying structure.

The row labelled "distance" says that there may be more than one representation associated to the same distance for both graphic and non-graphic spaces (same as above); the first 1 in that row, in table 1, tells us that a graphic distance has a unique graphic representation (a simple graph). In table 2 in the same cell we find 'many', meaning that for

any non-graphic distance there is no simple graph representation but instead there are many possible weighted graphs representations. *So, the notion of unique and discrete search space structure, like the hyper-cube for the Hamming distance for binary strings for example, in the case of non-graphic distances is lost*[1]. The following two 1s in the row, in both tables, are due to the fact that mutation and crossover are function of the distance, so they are unique to it.

The row for mutation tells us that the same mutation operator, graphic or non-graphic, may arise from different solution representations. Given a graphic mutation operator is always possible to determine the full structure of its underlying graphic space and, hence, its associated graphic distance and its associated graphic crossover. This uniqueness result is possible because of the graphic character of graphic mutation and it is not valid in general (for details see proof of theorem 1 in (Moraglio & Poli, 2004)). The situation for non-graphic mutation is quite different: passing from a weighted graph (structure of the search space) to its induced non-graphic metric space, there is a loss of information; there is a further loss of information when passing from the distance to its induced non-graphic mutation. The same reasoning applies to crossover (last row in the tables). *Hence, for non-graphic operators, the theorem of uniqueness of distance and space structure for crossover and mutation, which holds in the graphic case, ceases to hold, opening up to the possibility of more than one distance and space structure associated with the same non-graphic operator.*

In essence table 1 tells us that no matter what graphic element one knows - space structure, distance, mutation or crossover - one can always determine any other. Table 2 tells us that for non-graphic spaces the weighted structure has more information than the induced distance that, in turn, embeds more information than induced mutation and crossover operators. Since the mutation-crossover isomorphism theorem for graphic operators relies on the uniqueness of their underlying distance, the one-to-one mapping between non-graphic mutation and non-graphic crossover is not provable in this way. This leaves us with an open question on this issue (the question marks in table 2 indicate the open question).

The fact that homologous crossover for syntactic trees turned out to be non-graphic does not preclude the possibility of a graphic crossover for syntactic trees based on a graphic distance between trees. (O'Reilly, 1997) proposed a simple extension of Levinsthein distance for sequences to syntactic trees, that is indeed a graphic distance. The geometric crossover based on such a distance is, therefore, an example of graphic crossover for syntactic trees[2]. *So, the non-graphic label is attached to the distance and to the genetic operators based on it; it is not inherent of the underlying representation or of the geometric operators for a given representation.*

5 SHD mutation

In the following we try to give an answer to the question: what is the mutation operator associated to homologous crossover?

We have seen in section 4 that is not clear weather or not the one-to-one mapping existing between graphic crossover and graphic mutation extends to non-graphic operators. However, since both mutation and crossover are defined as function of a distance, we will consider *one* mutation operator that is connected to the homologous crossover through the metric SHD. We should bear in mind, though, there may be other mutation operators connected to it through other distances.

(Vanneschi et al., 2003) introduced structural mutation operators for syntactic trees and proved that their operators are consistent in some sense with the structural distance. In the following we discuss the geometric mutation operator *defined* over the SHD, that is a variation on the structural distance. *That is, we define the potential mutated offspring of a tree as those trees that are within the ball or radius ε centred on the parent tree.*

Unlike geometric crossover that partitions the set of all binary genetic operators in two clear-cut categories, crossovers and non-crossovers, geometric mutation has a continuous character and any unary operator is a geometric mutation under any distance. The point is to understand how a syntactic change affects the amount of mutation (i.e. the distance between the parent and the offspring) under a given distance. So the questions to ask are: what syntactic change is a micro-mutation under SHD? And what other syntactic change is a macro-mutation? How much a specific syntactic change affects the amount of mutation?

To understand the peculiarity of SHD mutation we compare it with mutation for binary strings. For binary strings the amount of mutation is:

- *non-positional*: mutating any locus results in the same amount of mutation
- *proportional to the syntactic change*: lots of bit changed, lots of mutation
- *based on single-type mutation*: bit-flip only
- *additive*: two bit changed add up in terms of contribution

For syntactic trees the amount of mutation associated with SHD is:

- *positional*: the extent of the mutation depends on the depth at which the mutation occurs: the deeper the level, the smaller the mutation; it depends also on the branching factor of the path from the root node to the node at which mutation takes place: the bigger the branching factor, the smaller the mutation. If we want

[1] This holds for any representation when associated to a non-graphic distance because it is a consequence of the property of the distance function of being non-graphic and not of the underlying solution representation.

[2] However, the problem with such a distance is that the edit operations used do not preserve syntactic feasibility of the trees, extending the search space to infeasible syntactic trees. When such a distance is used as a basis for geometric crossover, it leads to a crossover that is allowed to generate infeasible offspring and this is undesirable.

to restrict the mutation to be within a certain distance from the parent tree, this can be done approximately by picking mutation sites below a certain level in the tree[3]. If we take as a mutation site every node in the parent tree with uniform probability on the node of the tree, we allow for maximal macro-mutation (changing the root of the tree produces a tree a maximal distance (distance 1)) with low probability and micro-mutation with higher probability since the number of nodes increases geometrically with the depth of the node in the tree.

- *Non-proportional to syntactic change*: a big mutation at a big depth may be smaller than a small mutation closer to the root
- *Based on various types of mutation* (Back et al, 2000):
  - Point mutation (Langdon & Poli, 2002): node substitution at a specified position in the tree
  - Subtree-prune mutation: a sub-tree is substituted by a terminal node
  - Subtree-grow mutation: a terminal node of the tree is substituted by a sub-tree
  - Subtree mutation: a sub-tree is substituted by another sub-tree
  - All edit moves considered above are degenerated forms of sub-tree edit move
- *Weighted additive and coherent*: there is only one weighted edit move, the unrestricted sub-tree edit move, which degenerates to specific *coherently and additively weighted edit moves* in special cases.

Footnote: In other paradigms, linking genetic operators and fitness landscape (Jones, 1995), the mutation operator is modelled putting weights on edges corresponding to transition probability. In the geometric framework the weights on edges are measure of distance (remoteness, not probability) and the (conditional) probability distribution of the mutation operator is on the nodes of the graph.

6 Tree Interpretation and Smooth Landscape

In previous sections we have described the search space associated with genetic operators for syntactic trees. In the following we discuss how such a search space and fitness connect together giving a picture of the fitness landscape in its entirety.

In the introduction we mentioned that what is really important for an algorithm to perform better than random search is how problem and algorithm are connected via distance. In (Moraglio & Poli, 2005) we have suggested that if one picks a distance that makes sense for the entity represented (phenotype) by the solution (genotype), then the geometric crossover defined over this distance is likely to perform well. The logic is the following: closer genotypes imply closer phenotypes that in turns imply closer fitness. This allows for a smooth fitness landscape that is good for most meta-heuristics based on neighbourhood search (Glover, 2002). Naturally, this is a rule of thumb and not a proven theorem. A question comes to mind: does the SHD metric associated with homologous crossover make sense when syntactic trees are interpreted as GP programs? Is it a meaningful distance in terms of GP programs?

Because of the way solutions are encoded in genetic programming and since information propagates in the tree from the leaves (that could be never reached during evaluation of a solution) to the root node (that is always considered), the nodes near the root of the tree are much more influential than nodes at lower levels. Such an interpretation of a syntactic tree is very different from that given to other types of tree-like structures. For example, in a tree structure to find the minimum spanning tree of a graph encoding a sub-part of the graph, every node of the tree has presumably the same importance. The syntax of the two representations above is similar, but the part of their syntax having an impact on the phenotype (interpretation) is completely different.

In section 5, we have seen that the distance associated to homologous crossover assigns a greater weight, for the same amount of syntactic change, to the top of the tree and smaller weight to the bottom of the tree. This goes with the previous landscape design principle in that, when the tree is interpreted as a GP program, changes at upper levels of the tree have a much higher impact on the behaviour of a program than changes at lower levels. In turn, the impact on the behaviour is reflected on the fitness. So, programs that are modified at an upper level have much higher probability to behave completely differently and, therefore, to have very different fitnesses than programs that are neighbours for a modification at a lower level in the tree.

Homologous crossover for syntactic trees is therefore a very natural choice when the trees are interpreted as GP programs as it induces a smoother landscape that is likely to facilitate the search.

7 Conclusions

We have shown that the geometric framework naturally connects the notion of homologous crossover, subtree mutation, hyperschema and structural distance for syntactic trees. We have also described the structure of the space of syntactic trees associated with the previous elements and argued that, when using the standard interpretation of syntactic trees as programs, the associated landscape is smooth, hence the homologous crossover is a good choice.

In the future we will be looking at other distances for syntactic trees and corresponding spaces and operators. In particular we will focus on component-wise distances and grammatical distances, that arise by considering, respectively, the syntactic tree as a collection of sub-components, and as a syntactic object based on formal grammar.

To conclude we want to emphasise the significance of the present results in the larger context of our on-going programme of evolutionary algorithms unification: most of

[3] This method is exact only for full balanced trees with nodes of arity 2 or more.

the pre-existing genetic operators for binary strings, permutations, real vectors and now also for syntactic trees, all fit nicely and naturally the geometric framework hence implying a profound geometric unity of all major flavours of existing evolutionary algorithms.

References

Back et al, 2000. T. Back, D. B. Fogel, Z. Michalewicz (eds). Evolutionary Computation 1. IoP, 2000.

Back et al, 1997. Fogel, T. Back, D. B. Fogel, Z. Michalewicz (eds). Handbook of Evolutionary Computation. Oxford press, 1997.

Blumental & Menger, 1970. Studies in geometry, Freeman and Company.

Deza & Laurent, 1991. Geometry of cuts and metrics, Springer

Ekárt & Németh, 2000. A metric for genetic programs and fitness sharing. Genetic Programming, Proceedings of the 3rd European Conference. Springer-Verlag.

Glover, 2002. F. W. Glover (ed). Handbook of Metaheuristics. Kluwer, 2002.

Gustafson & Vanneschi, 2005. Operator-Based Distance for Genetic Programming: Subtree Crossover Distance, EUROGP 2005, Springer, 2005.

Jones, 1995. T. Jones. Evolutionary Algorithms, Fitness Landscapes and Search. PhD dissertation, University of New Mexico, 1995.

Koza, 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992

Langdon & Poli, 2002. W. B. Langdon, R. Poli. Foundations of Genetic Programming. Springer, 2002.

Moraglio & Poli, 2004. Topological interpretation of crossover, Proc. of GECCO 2004.

Moraglio & Poli, 2005. Abstract geometric crossover for the permutation representation, IEEE Transaction in Evolutionary Computation (to appear)

Nienhuys-Cheng, 1997. Distance between Herbrand interpretations: a measure for approximations to a target concept. Proceedings of the 7th International Workshop on Inductive Logic Programming. Springer-Verlag.

O'Reilly, 1997. Using a distance metric on genetic programs to understand genetic operators. IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, volume 5.

Pardalos & Resende, 2002. P. M. Pardalos, M. G. C. Resende (eds) Handbook of Applied Optimization. Oxford University Press, 2002

Reidys & Stadler, 2002. C. M. Reidys, P. F. Stadler. Combinatorial Landscapes. SIAM Review 44, 3-54, 2002.

Vanneschi et al., 2003. Fitness Distance Correlation in Structural Mutation Genetic Programming, EUROGP 2003, Springer, 2003.

Wolpert & Macready, 1996. D. H. Wolpert, W. G. Macready. No Free Lunch Theorems for Optimization. IEEE Transaction on Evolutionary Computation, April 1996.