Geometric Landscape of Homologous Crossover for Syntactic Trees

Alberto Moraglio
Department of Computer Science, University of Essex, UK
amoragn@essex.ac.uk

Riccardo Poli
Department of Computer Science, University of Essex, UK
rpoli@essex.ac.uk

Abstract-The relationship between search space, distances and genetic operators for syntactic trees is little understood. Geometric crossover and geometric mutation are *representation-independent* operators that are well-defined once a notion of distance over the solution space is defined. In this paper we apply this geometric framework to the syntactic tree representation and show how the well-known structural distance is naturally associated with homologous crossover and sub-tree mutation.

1 Introduction

A fitness landscape (Wright, 1932) can be visualised as a 3D plot resembling a geographic landscape when the problem representation is a real vector of dimension 2. This interpretation can be extended for real vectors of higher dimensions. 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.

An abstraction of the notion of landscape encompassing all the previous cases is possible. 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 (Blumental & Menger, 1970). 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 options, e.g. Minkowski distances. The distance associated to combinatorial objects is normally the length of the shortest path between two nodes in the associated neighbourhood graph (Deza & Laurent, 1991). In the binary case, the shortest path distance associated to the hypercube is the 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 are equally natural notions of distance arising from different types of syntactic similarity between permutations. Different notions of similarity are possible because the same permutation (genotype) can be used to represent different types of solutions (phenotypes). For example, permutations can represent solutions of a problem where relative order is important. However, they can also be used to represent tours, where the adjacency relationship among elements is what matters.

The notion of fitness landscape is useful if the search operators employed are connected or matched with the landscape: the greater the connection the more landscape properties mirror search properties. Therefore, the landscape can be seen as a function of the search operator employed (Jones, 1995). Whereas mutation is intuitively associated with the neighbourhood structure of the search space, crossover stretches the notion of landscape further leading to search spaces defined over complicated topological structures (Jones, 1995).

In (Moraglio & Poli, 2004) we introduced a representation-independent geometric generalization of crossover and mutation for binary strings and real vectors. These operators are based on the distance associated with the search space, seen as a metric space, and on the geometric notions of ball and line segment. This approach is the dual of Jones' approach: *we see the genetic operators as functions of the search space*. So, mutation and crossover share the same neighbourhood structure.

Since our geometric operators are representation-independent, it is important to understand how they relate with the NFL theorem (Wolpert & Macready, 1996). The key is the difference between problem and landscape, the former being given and the latter being chosen by the designer. The landscape can be seen as a knowledge interface between algorithm and problem (Moraglio & Poli, 2005). Through a domain-specific solution representation and a distance that makes sense for the problem at hand, one can embed problem knowledge in the landscape. 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: pick a crossover associated to a good mutation, build a crossover using a neighbourhood structure based on the small-move/small-fitness-change principle, or build a crossover using a distance that is relevant for the solution interpretation.

Disregarding minor differences, many evolutionary algorithms differ only in the solution representation and the genetic operators. In (Moraglio & Poli, 2004) we conjectured that many operators developed for important representations, comprising binary strings, real-valued vectors, permutations and syntactic trees, fit our geometric definitions given suitable notions of distance and that, therefore, our geometric framework could lead to a unification of different evolutionary algorithms. 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. Here we provide the following contributions: a) Application of the geometric framework (Moraglio & Poli, 2004) to the syntactic tree representation discussing the difference with other representations; b) Proof that the family of homologous crossovers (Langdon & Poli, 2002) for syntactic trees are geometric crossover under a family of structural distances; c) Clarification of the structure of the search space associated with structural distances; d) Proof that the natural mutation operator associated with homologous crossover and structural distances is the sub-tree mutation operator; f) Corroboration that homologous crossover based on structural distances between syntactic trees is a meaningful genetic operator for genetic programming.

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). The following definitions are taken from (Deza & Laurent, 1997).

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. $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$ (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 length of a shortest path from *i* to *j* in *G*. We say that *G* represents *M*. Graphic metric spaces have unique graph representations. Any metric space that cannot be represented by a graph is a *non-graphic metric space*.

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.

In Euclidean geometry, the distance between two points in $\mathbf{R}^2$, say *A* and *B*, is calculated using the formula: $d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$. By redefining the distance function one obtains new geometries. One example is the 1st order Minkowski distance, $d(A, B) = | x_A - x_B | + | y_A - y_B |$ which 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 look quite different if we use a non-Euclidean distance. Indeed, we can go further and say that shapes are defined independently from the specific notion of metric used. These *abstract shapes* are studied in *metric geometry*. Two of them, balls and segments, are very useful to define abstractly mutation and crossover.

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. The length l of the segment $[x; y]$ is the distance between a pair of extremes $l([x; y]) = d(x, y)$. Note that $[x; y] = [y; x]$ and that segments can have more than a pair of extremes. Also, a segment does not always coincide with a shortest path connecting its extremes (*geodesic*). Indeed, there may be more than one geodesic.

We assign a structure to the solution *set S* by endowing it with a distance *d*. *M=(S, d)* is a solution *space* and *L=(M, g)* is the corresponding fitness landscape, where *g* is the fitness function. Note that *d* is an arbitrary distance and need not have any connection with the search problem at hand. However, to exploit problem knowledge in the search, one has to pick a distance that makes sense for the problem at hand.

2.2 Geometric operator definitions

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

Definition 1 *The* image set *of a genetic operator OP for parents* $p_1, p_2, ... p_g$ *is*

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

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

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

This simply means that in an 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 an abstract ε-mutation where*

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

δ(x) is a function which returns 1 if x is true , 0 otherwise. When ε is not specified, we mean ε = 1.

Definition 5 *Abstract uniform crossover UX is an abstract crossover where*

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

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

2.3 Uniqueness results for graphic distances

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 Crossovers and distances for trees

Let us now consider the tree representation and the class of homologous crossovers for trees.

3.1 Subtree Swap Crossover & Homologous Crossover

The *common region* is the largest rooted region where two parent trees have the same topology. 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 in which a common crossover point is picked randomly from the nodes belonging to the common region and then the two sub-trees rooted at the crossover point are swapped. In *subtree swap crossover* (Koza, 1992) any subtree of one parent can be exchanged with any subtree of the other.

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 a 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 two 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 does not need to contain any single node tree. In this case the distance between the two tree parents must be two, but 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 the search space of subtree swap crossover. (See (Gustafson & Vanneschi, 2005) for a different notion of distance for this operator.)

Because of Theorem 3 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.

3.3 Structural Distance and Hyperschemata

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

We propose the following *normalized structural hamming distance (SHD)* for trees

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

With SHD 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.

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

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

■

The hyperschema (Langdon & Poli, 2002) 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 1 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).

The SHD distance between two trees is only function of the hyperschema associated with the two trees and not directly of the two trees (figure 2).

3.4 Geometric Crossover Theorems

Theorem 5. *Homologous crossover is a geometric crossover under SHD.*

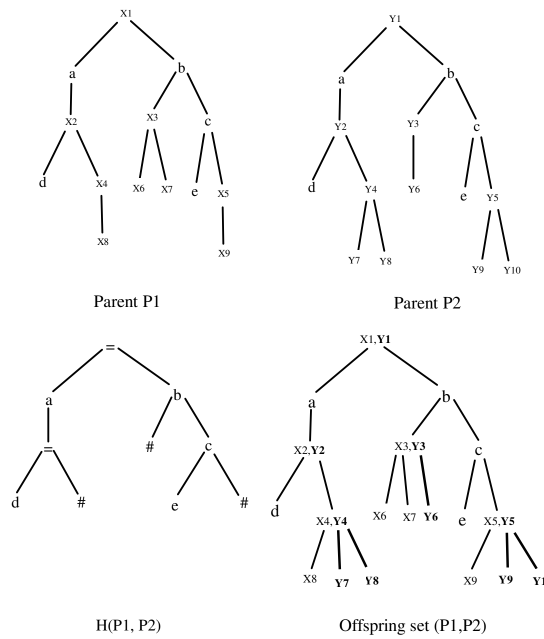

Figure 1 Hyperschema and offspring set

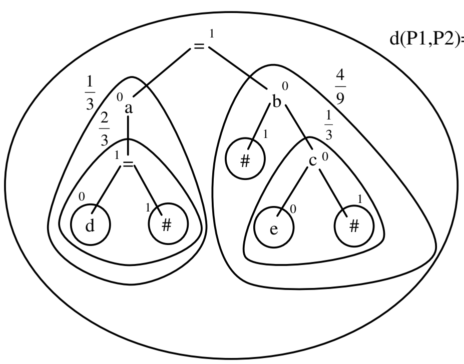

Figure 2 Hyperschema and structural distance

Proof

Remark 1: as shown in figure 2, 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 $\{p(i)\} \cup \{p(j)\} = \{p1,\dots,pn\}$

Remark 6: Hence: $d(H(P1,O))+d(H(O,P2)) =$
$=d(\{p(i)\})+d(\{p(j)\})=sum\{d(p(i))\}+sum\{d(p(j))\}=$
$=sum\{d(p1),\dots,d(pn)\}=d(\{p1,\dots,pn\})=d(P1,P2)$

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 crossovers 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

The normalizing value n=1/(m+1) in SHD has been chosen to have a strict bound at 1 and consistency in the distance between two fully different subtrees in two senses: (i) any two sub-trees that 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 values of n, SHD is still a metric but the bound 1 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 n slightly bigger than 1/(m+1), SHD is still a metric, not bounded by 1, but still bounded. Between 1/(m+1) and 1 there is a critical value of n for which SHD ceases to be a metric. In the following we consider SHD with n=1, that we call Hamming distance (HD) between syntactic trees. (HD can be also seen as the number of mismatching nodes at corresponding positions within the common region.)

Theorem 7. *HD 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 k 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)=k, it is immediate to see that for k>2 the triangular inequality fails to hold. Hence, HD is not a metric ■

Looking at the proofs of theorem 5 and 6, it is easy to see that they work for 0<n<1/(m+1). So there is a whole family of distances that match homologous crossover.

4 Graphic space vs non-graphic space

In the previous section we have shown that homologous crossover is geometric but non-graphic. Since Theorem 1 and Corollaries 1, 2 and 3 are for graphic distances, they do not necessarily hold for tree homologous crossover.

Tables 1 and 2 summarise the cardinality of the relations between solution representation, neighbourhood structure, distance, geometric mutation and geometric crossover in the case of graphic and non-graphic spaces, respectively. The rows labelled “representation” are identical and tell us that to a solution representation may be associated: (i) more than one neighbourhood structure; (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.

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?**	-

The rows labelled (neighbourhood) “structure” are also the same. This tells us that there can be different representations that induce the same neighbourhood structure; 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; the first 1 in that row, in table 1, tells us that a graphic distance has a unique graphic representation. 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.* The following two 1s in the row, in both tables, are there because 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 (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 the 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.

The fact that homologous crossover for syntactic trees is 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 (however, the geometric crossover based on such a distance is allowed to generate infeasible offspring and this may be undesirable). *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

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 functions of a distance, we will consider *one* mutation operator that is connected to the homologous crossover through the SHD metric. We should bear in mind, though, that 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 consider 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 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 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. 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.

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 (some of which might never be reached during evaluation of a solution) to the root node (that is always considered), the nodes near the root of the tree are typically 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 well 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 these elements and argued that, when using the standard interpretation of syntactic trees as programs, the associated landscape is smoother, 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 a 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 the pre-existing genetic operators for binary strings, permutations, real vectors and now also some operators 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. T. Back, D. B. Fogel, Z. Michalewicz (eds). Handbook of Evolutionary Computation. Oxford press, 1997.

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

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

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

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. Proceedings of GECCO 2004, Springer, 2004.

Moraglio & Poli, 2005. Abstract geometric crossover for the permutation representation, Tech. Rep. CSM-429, Essex University, 2005.

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, 1997.

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, 1997.

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

Wright, 1932. S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution, Proc. 6th International Conference on Genetics, 1932.