Topological Interpretation of Crossover

Alberto Moraglio and Riccardo Poli

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

Abstract. In this paper we give a representation-independent topological definition of crossover that links it tightly to the notion of fitness landscape. Building around this definition, a geometric/topological framework for evolutionary algorithms is introduced that clarifies the connection between representation, genetic operators, neighbourhood structure and distance in the landscape. Traditional genetic operators for binary strings are shown to fit the framework. The advantages of this interpretation are discussed

1 Introduction

Fitness landscapes and genetic operators have been studied for considerable time in connection with evolutionary algorithms. However, a unifying theory of the two is missing and many questions about their relationship remain unanswered. Below we will briefly analyze the current situation in this respect.

Fitness landscapes and genetic operators are undoubtedly connected. Mutation is intuitively associated with the neighbourhood structure of the search space. However, the connection between landscape and crossover is less clear. Complicated topological structures, hyper-neighbourhoods, have been proposed [Culberson, 1995; Jones, 1995; Gitchoff & Wagner, 1996; Reidys & Stadler, 2002] to formally link crossover to fitness landscapes. However, even using these ideas, effectively one is left with a different landscape for each operator [Culberson, 1995], which is deeply unsatisfactory. Important questions then are: is there an easier way of interpreting crossover in connection to fitness landscapes? Are crossover and mutation really unrelated?

An established way of defining a fitness landscape for search spaces where a natural notion of distance exists is to imagine that the neighbourhood of each point includes the points that are at minimum distance from that point [Back et al, 1997]. Once a landscape is defined, typically the notion of distance is not used further. Couldn't distance play a much more important role in explaining the relationship between landscapes and crossover?

Local search and many other meta-heuristics are naturally defined over the neighbourhood structure of the search space [Glover, 2002]. However, a peculiarity of evolutionary algorithms (seen as meta-heuristics) is that the neighbourhood structure over the search space is specified by the way genetic operators act on the representation for solutions. One may wonder whether it is possible to naturally reconcile these two ways of defining structure over the search space.

Yet in another sense, solution representation and neighbourhood structure are just two different perspectives on the solution space. An example is the classical binary string representation and its geometric dual, a hypercube, which has been extremely useful in explaining genetic algorithms [Whitley, 1994]. Can solution representation and neighbourhood structure be two sides of the same coin for other representations, like permutation lists or syntax trees?

The traditional mutation and crossover operators defined for binary strings have been *extended* to other representations [Langdon & Poli, 2002]. Also, there are general guidelines for the *design* of such operators for representations other than binary [Radcliffe, 1994; Surry, 1998]. Is there a way to rigorously *define*, rather than *design* or *extend*, mutation and crossover in general, independently of the representation adopted?

Except for solution representations, many evolutionary algorithms are very similar which suggests that unification might be possible [Stephens & Poli, 2004]. Are all evolutionary algorithms really the same algorithm in some sense?

In this paper we clarify the connection between representation, genetic operators, neighbourhood structure and distance and we propose a new answer to the previous questions. The results of our work are surprising: all the previous questions are connected, and that the central question to address is really only one: what is crossover?

The paper is organized as follows. In section 2, we introduce some necessary definitions. Geometric/topological definitions of crossover and mutation are given in section 3, where we also prove some properties of these operators. As an example, in section 4, we show how traditional mutation and crossover defined over binary strings, fit our general topological definitions for mutation and crossover. In section 5, we discuss some implications of our topological interpretation of crossover. Finally, in section 6, we draw some conclusions and we indicate our future research directions.

2 Preliminary definitions

2.1 Search problem

Let S denote the *solution set*[1] comprising all the candidate solutions to a given *search problem* P. The members of this set must be seen as *formal solutions* not relaying on any specific underlying representation.

The goal of a search problem P is to find specific solution/s in the search space that maximize (minimize) an *objective function*:

$$g : S \to R$$

[1] We distinguish between *solution set* and *solution space*. The first refers to a collection of elements, while the second implies a structure over the elements.

Let us assume, without loss of generality, that the goal of the search problem P is to maximize g. The *global optima* x^* are points in S for which g is a maximum:

$$x^* \in S^* \Leftrightarrow g(x^*) = \max_{x \in S} g(x)$$

Notice that global optima are well defined when the objective function is well defined and are independent of any structure defined on S**.** On the contrary, *local optima* are definable only when a structure over S is defined. A search problem in itself does not come with any predefined structure over the solution set.

2.2 Fitness landscape

A *configuration space* C is a pair (G, Nhd) where G is a set of syntactic configurations (syntactic objects or genotypes) and $Nhd : G \rightarrow 2^G$ is a syntactic neighbourhood function which maps every configuration in C to the set of all its neighbour configurations in C which can be obtained by applying a unitary syntactic modification operator. The unitary syntactic modification operator must be reversible (i.e. $y \in Nhd(x) \Leftrightarrow x \in Nhd(y)$) and connected (any configuration can be transformed into any other by applying the operator a finite number of times). Notice that a configuration set may lead to more than one configuration space if multiple syntactic neighbourhood functions are available.

A configuration space C=(G, Nhd) is said to be a space endowed with a *neighbourhood structure*. This is induced by the syntax of the configurations and the particular notion of syntactic neighbourhood function adopted. Such a neighbourhood structure can be associated with an undirected *neighbourhood graph W= (V, E)*, where V is the set of vertices representing configurations and E is the set of edges representing the relationship of neighbourhood between configurations.

Since the neighbourhood is symmetric ($y \in Nhd(x) \Leftrightarrow x \in Nhd(y)$) and the neighbourhood structure is connected, this space is also a *metric space* provided with a *distance function d* induced by the neighbourhood function (see formal definition below) [Back et al, 1997]. So, we can equivalently write C=(G, Nhd) or C=(G, d). However, we must keep in mind that the notion of distance in the metric space of syntactic configurations has a syntactic nature (and, therefore, may have special features other than respecting distance axioms). Distances arising from graphs are known as *graphic distances* [Van der Vel, 1993].

Formally, 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 $s_1, s_2 \in M$:

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 *representation mapping* is a function $r : G \rightarrow S$ associating any syntactic configuration in G with a formal solution in S. Ideally this mapping would be bijective. However, there are cases where the sizes of G and S differ.

A *fitness landscape F* is a pair *(C, f)* where C=(G, d) is a configuration space and $f: G \rightarrow R$ is a *fitness function* mapping a syntactic configuration to its *fitness value*. The fitness value is a positive real number. It may or may not coincide with the objective function value of the solution represented by the input genotype. For the sake of simplicity, we assume that it is. Therefore, the fitness function is the composition of the representation mapping r with the objective function g: $f = g \circ r$.

2.3. Topological and geometric preliminaries: balls 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 as $[x; y] = H$. 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 ex-

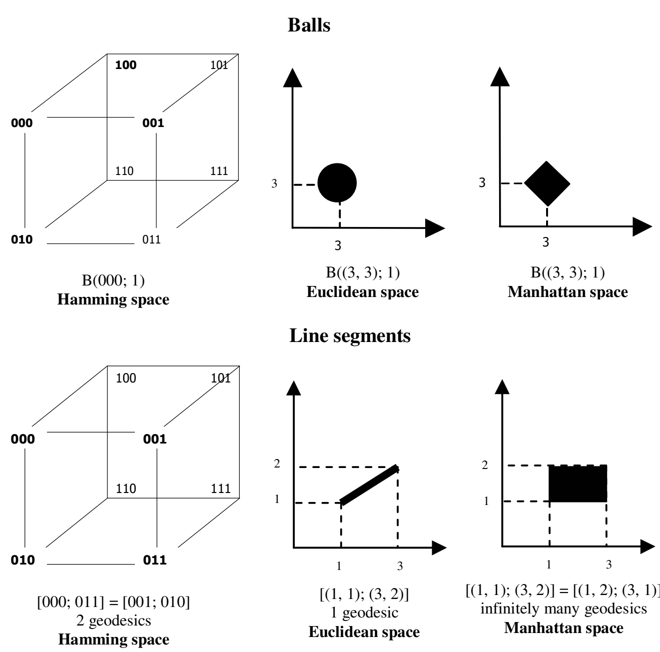

Fig. 1. Balls and segments for different spaces

tremes 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 the 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.

3. Topological genetic operators

We *define*, postponing the justifications of these definitions to the discussion, two *classes* of operators in the landscape (i.e. using the notion of *distance* coming with the landscape): topological mutation and topological crossover. Within these classes, we identify two *specific* operators: topological uniform mutation and topological uniform crossover. *These definitions are representation-independent and therefore the operators are well-defined for any representation.*

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:

$$\Pr\{OP(p_1, p_2, \ldots p_g) = c\} = \Pr\{OP = c \mid P_1 = p_1, P_2 = p_2, \ldots, P_g = p_g\} = f_{OP}(c \mid p_1, p_2, \ldots p_g)$$

Mutation is a unary operator while *crossover* is typically a binary operator.

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

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

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 topological 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 topological ε-mutation* all offspring are at most ε *away* from their parent.

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

This simply means that in a topological 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 *Topological uniform ε-mutation UM is a topological ε-mutation where all z at most ε away from parent x have the same probability of being the offspring:*

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

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

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 *Topological uniform crossover UX is a topological crossover where all z laying between parents x and y have the same probability of being the offspring:*

$$f_{UX}(z \mid x, y) = \Pr\{UX = z \mid P1 = x, P2 = y\} = \frac{\delta(z \in [x, y])}{|[x, y]|}$$

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

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.***

Proof.

Equivalences 1, 2 and 3 are trivial consequences of the fitness landscape definition.

Equivalence 4: given *UM* one has its conditional density function $f_{UM}(z \mid x)$ and, consequently, the image set mapping $\mathrm{Im}[UM(x)]$, i.e. the mapping $x \mapsto B(x,1)$. The structure of the space is therefore given by $Nhd : x \mapsto (B(x,1) \setminus \{x\})$.

Equivalence 5: analogously, given *UX* one has the mapping $(x, y) \mapsto [x; y]$. By restricting this mapping through its co-domain considering only segments of size 2, the corresponding restricted domain coincides with the set of edges *E* of the neighborhood graph, hence the structure of the space is determined.

Equivalence 6: the relation of inclusion between sets $\subseteq$ induces a partial order in **B**. The set of all balls of radius 1 $\mathbf{B}_1$ can be determined by considering all those balls in **B** that have, as only predecessors, balls of size 1 (i.e. balls of radius zero). Given a ball $b \in \mathbf{B}_1$ a point $x \in b$ is the center of the ball if and only if $\forall x' \in (b \setminus \{x\}) \exists b' \in \mathbf{B}_1 : b \neq b' \wedge x, x' \in b'$.[2] Knowing the center of each ball of radius 1, it is possible to form the map $x \mapsto B(x,1)$ and proceed as in equivalence 4.

Equivalence 7: by considering only segments in **H** of size 2, one can form the set *E* of the edges of the neighborhood graph; hence, the structure of the space is determined.■

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

Proof.

Since both *UM* and *UX* identify the structure of the configuration space univocally and also the configuration space structure identify both operators univocally then they 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.*

Proof.

[2] Given two different points in the same ball of radius 1 $x, x' \in b$, they are either at distance 1 or distance 2. If they are at a distance 2, *b* is the only ball in $\mathbf{B}_1$ satisfying this condition since the two points are extremes of a diameter of the ball *b* and identify the ball univocally. If they are at a distance 1, there must exist at least two balls in $\mathbf{B}_1$ containing x, x' one in which one is the center and the other is not, and another one in which the roles are reversed; this symmetry holds because the neighborhood is symmetric.

This follows trivially from the definition of *UM* and *UX* over the space structure. ■

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

Proof.

Given a representation, one has a configuration set for which the structure is not specified. A specific notion of graphic distance transforms the set into a space with a structure. Given such a structure, UM and UX are unique (corollary 2). ■

5. Generalization of binary string crossover

Given two binary strings $s_1 = (x_1,..., x_n)$ and $s_2 = (y_1,..., y_n)$ of length *n*, the *Hamming distance* $d_H(s_1, s_2)$ is the number of places in which the two strings differ, i.e.

$$d_H(s_1,s_2) = \sum_{i=1}^{n} \delta(x_i \neq y_i)$$

A *property* of the Hamming distance is that a binary string $s_3 = (z_1,..., z_n)$ lays between s_1 and s_2 if and only if every bit of s_3 equals al least one of the corresponding bits of s_1 and s_2, i.e. $\forall i : z_i \in \{x_i, y_i\} \Leftrightarrow s_3 \in [s_1, s_2]$.

Traditional (one-point, two-point, uniform, etc.) crossovers for binary strings belong to the class of *mask-based crossover operators* [Syswerda, 1989]. A crossover operator is a probabilistic mapping $cx_m : S \times S \xrightarrow{m} S$ where the mask *m* is a random variable with different probability distributions for different crossover operators. The mask *m* takes the form of a binary string of length *n* that specifies for each position from which parent to copy the corresponding bit to the offspring, i.e. $cx_m(s_1,s_2) = s_3$ and $m = (m_1,..., m_n)$ then $z_i = x_i \cdot \delta(m_i = 0) + y_i \cdot \delta(m_i = 1)$.

Theorem 2 *All mask-based crossover operators for binary strings are topological crossovers. All mutations for binary strings are topological ε-mutations.*

Proof.

We need to show that for any probability distribution over *m* it holds $\mathrm{Im}[cx_m(s_1,s_2)] \subseteq [s_1,s_2]$. Out of all possible mask-based crossovers, those with a non-zero probability of using all the 2^n masks produce the biggest image set for any given pair of parents. Formally, this is given by $\mathrm{Im}[cx(s_1,s_2)] = \{cx_m(s_1,s_2) \mid m \in B^n\}$. So, it is sufficient to prove that $\mathrm{Im}[cx(s_1,s_2)] \subseteq [s_1,s_2]$ for this image set. This is equivalent to prove that $\forall m \in B^n : s_3 = cx_m(s_1,s_2) \rightarrow s_3 \in [s_1,s_2]$.

Given $s_1 = (x_1,...,x_n)$, $s_2 = (y_1,..., y_n)$ and any mask *m* there exists a unique $s_3 = (z_1,...,z_n)$. From the definition of mask-based crossover it follows that $\forall i : z_i \in \{x_i, y_i\}$. Then, from the Hamming distance property mentioned above, it follows that $\forall m : s_3 \in [s_1, s_2]$, which completes the proof of the first part of the theorem.

Let $s_2 = \mu(s_1)$ be the result of mutating s_1, that is $s_2 \in \mathrm{Im}[\ \mu(s_1)]$, then $\exists \varepsilon : \forall s_2 : d_H(s_1, s_2) \le \varepsilon$ whereby $s_2 \in B(s_1, \varepsilon)$ with ε being the smallest possible. ■

Theorem 3. *The topological uniform crossover for the configuration space of binary strings endowed with Hamming distance is the traditional uniform crossover. The topological uniform 1-mutation for the configuration space of binary strings endowed with Hamming distance is equivalent to a zero-or-one-bit mutation.*

Proof.

Let us start by proving that the image sets of traditional uniform crossover and topological uniform crossover coincide. We need to show that $\mathrm{Im}[cx(s_1, s_2)] = [s_1, s_2]$, where $\mathrm{Im}[cx(s_1, s_2)]$ was defined in the proof of theorem 2, from which we know that $\mathrm{Im}[cx(s_1, s_2)] \subseteq [s_1, s_2]$. Consequently, all we need to prove is that $\forall s_3 \in [s_1, s_2] \rightarrow \exists m \in B^n : cx_m(s_1, s_2) = s_3$. For the Hamming distance property this is equivalent to say $\forall s_3 \forall i : z_i \in \{x_i, y_i\} \rightarrow \exists m \in B^n : cx_m(s_1, s_2) = s_3$, where z_i are the bits of s_3. From the definition of crossover this is equivalent to proving that $\forall s_3 \forall i : z_i \in \{x_i, y_i\} \rightarrow \exists m \in B^n : z_i = x_i \cdot \delta(m_i = 0) + y_i \cdot \delta(m_i = 1)$. This is true because it always exists at least a mask for which when the bits in the parents differ, it specifies the parent for which the bit equals the offspring bit. If the bits do not differ, the mask indifferently specifies one parent or the other for that bit. This shows that the image sets of traditional uniform crossover and topological uniform crossover coincide.

Every element of the image set of the traditional uniform crossover has identical probability of being the offspring [Whitley, 1994] and the same is true for the elements of the image set of the topological uniform crossover (by definition). This completes the proof of the first part of this theorem.

Let us now consider the zero-or-one-bit mutation. This is an operator where a string is either mutated by flipping one bit or is not mutated with equal probability. The image sets of this mutation and topological 1-mutation coincide as it is trivial to see by noting that the Hamming ball of radius 1, which is the image set of topological 1-mutation, coincides with the image set of the zero-or-one-bit mutation. Every element of the image set of zero-or-one-bit mutation has identical probability of being the offspring and the same is true for the elements of the image set of the topological uniform 1-mutation (by definition). ■

6. Discussion

In the introduction, we raised various questions, claiming that this way of interpreting crossover lays a foundation to connect all these questions. In the following, we show how our framework answers those questions by highlighting the properties of the class of topological crossovers.

1. *Generalization*: topological crossover is a generalization of the family of crossovers based on masks for binary representation in that it captures and generalizes the distinction between crossover and recombination for binary representation.

2. *Unification*: from preliminary research, we believe that a variety of operators developed for other important representations, such as real-valued vectors, permutations and syntax trees, fit our topological definitions given suitable notions of distance (naturally not *all* pre-existing operators do this, but many do). Hence, topological crossover has the potential to lead to a unification of the different evolutionary algorithms.
3. *Representation independence*: evolutionary computation theory is fragmented. One of the reasons is that there is not a unified way to deal with different solution representation (although steps forward in this direction have recently been made [Langdon & Poli 2002; Stephens & Poli 2004]), which has led to the development of significantly different theories for different representations. In this context, one important theoretical implication of our topological definitions is that the genetic operators are fully defined without any reference to the representation. This may pave the route to a unified treatment of evolutionary theory.
4. *Clarification*: the connections between operators, representation, distance and neighborhood are completely clear when using topological operators.
5. *Analysis*: given a certain representation with pre-existing genetic operators, it is easy to check whether they fit our topological definitions. If they do, their properties are unveiled.
6. *Geometric interpretation*: an evolutionary algorithm using topological operators does a form of geometric search based on segments (crossover) and balls (mutation). This suggests looking at the solution space not as a graph or hyper-graph, as normally done, but rather as a geometric discrete space. The notion of distance arising from the syntax of configurations reveals therefore a natural *dual* interpretation:[3] (i) it is a measure of similarity (or dissimilarity) between two syntactic objects; (ii) and it is a measure of spatial remoteness between points in a geometric space.
7. *Principled design*: one important practical implication of the topological definition of crossover is the possibility of doing crossover principled design. When applying evolutionary algorithms to optimization problems, a domain specific solution representation is often the most effective [Davis, 1991; Radcliffe, 1992]. However, for any non-standard representation, it is not always clear how to define a good crossover operator. Given a good neighborhood structure for a problem, all meta-heuristics defined over such a structure tend to be good. Indeed, the most important step in using a meta-heuristic is the definition of good neighborhood structure for the problem at hand [Glover, 2002]. With topological crossover, given a good neighborhood structure or a good mutation operator, a crossover operator that respects such a structure is *automatically defined*. This has good chances of performing well, being effectively a composition of unitary moves on the landscape. An example is shown in Figure 2, where we assume that we want to evolve graphs with four nodes and we are given a mutation operator for such graphs that either adds or removes exactly one edge. We want to define a good

[3] Any mathematical object/property that admits a definition only based on the concept of distance possesses a dual nature: a syntactic one and a geometric one.

crossover operator that would, for example, produce meaningful offspring when applied to the parent graphs in Figure 2(a). The configuration space for this problem is shown in Figure 2(b). The parent graphs are boxed while the graphs belonging to the segment defined by the parents are encircled. With our definition of topological crossover these are all possible successors, as shown in Figure 2(c).

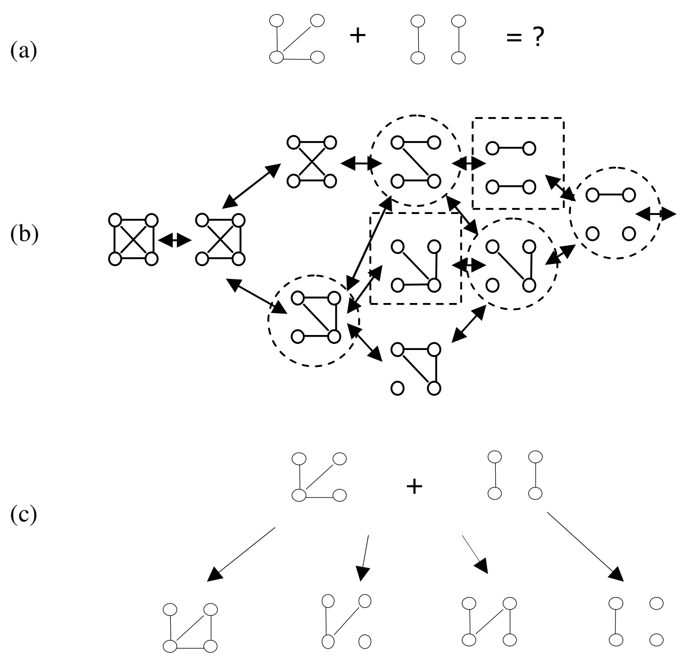

Fig. 2. Inducing crossover from mutation (see text).

8. *Landscape and knowledge*: the landscape structure is relevant to a search method only when the move operators used in that search method are strongly related to those which induce the neighborhood structure used to define the landscape [Back et al, 1997]. This is certainly the case for the topological operators. The problem knowledge used by an evolutionary algorithm that uses topological operators is embedded in the connectivity structure of the landscape. The landscape is therefore a *knowledge interface* between a *formal problem* and a *formal search algorithm* that has no knowledge of the problem whatsoever. In order for the knowledge to be transmissible from the problem to the search algorithm *through the landscape*, there are two requirements: (i) the search operators have to be defined over the connectivity structure of the landscape (i.e. using a distance function); (ii) the landscape has to be *designed* around the specific definitions of the operators employed in such a way to bias the search towards good areas of the search space so as to perform better than random search.

9. *Landscape conditions*: for the no free lunch theorem [Wolpert & Macready, 1996], over all the problems, on average any search algorithm performs the same as random search. So in itself a given search algorithm (any meta-heuristics) is not inherently superior to any other. A search algorithm therefore, to be of use, has to specify the class of problems for which it works better than random search. The geometric definition of mutation (connected with the concept of ball) and the geometric definition of crossover (connected with the concept of segment) suggest, respectively, conditions over the landscape in terms of *continuity* and *convexity*. These conditions, in various guises, are important to guarantee good performance in optimisation [Pardalos & Resende, 2002] and ensuring them should guide the landscape design for the topological operators.

7. Conclusions

In this paper, we have introduced a geometric/topological framework for evolutionary algorithms that clarifies the connections between representation, genetic operators, neighbourhood structure and distance in the landscape. Thanks to this framework a novel and general way of looking at crossover (and mutation) that is based on landscape topology and geometry has been put forward. Traditional crossover and mutation for binary strings have been shown to fit our topological framework, which, from preliminary investigations, appears to also encompass a variety of other representations and associated operators.

This framework presents a number of additional advantages. The theory is representation independent, and therefore it offers a unique opportunity for generality and unification. The theory provides a natural, direct and automatic way of deriving (designing) both mutation *and* crossover from the neighbourhood structure of a landscape. Conversely, if one adopts our topological operators, one and only one fitness landscape is induced: that is we *do not* have a different landscape for each operator, but a common one for both.

In future work we expect to further extend the applications of our framework to other representations and operators, to study the connections between this theory and other evolutionary computation theories (including those based on the notions of schema) and to investigate the links with generalized notions of convexity and continuity for the landscape.

References

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

[Culberson, 1995] J.C.Culberson. *Mutation-crossover isomorphism and the construction of discriminating functions.* Evol. Comp., 2:279-311, 1995.

[Davis, 1991] L. Davis, *Handbook of Genetic Algorithms*, van Nostrand Reinhold, New York, 1991.

[Gitchoff & Wagner, 1996] P. Gitchoff, G. P. Wagner. *Recombination Induced HyperGraphs: A New Approach to Mutation- Recombination Isomorphism.* Journal of Complexity, 37-43, 1996.

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

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

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

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

[Radcliffe, 1992] N. J. Radcliffe. *Nonlinear genetic representations*. In R. Manner and B. Manderick, editors, Proceedings of the 2 nd Conference on Parallel Problems Solving from Nature, pages 259-268. Morgan Kaufmann, 1992.

[Radcliffe, 1994] N. J. Radcliffe, 1994. *The Algebra of Genetic Algorithms*, Annals of Maths and Artificial Intelligence 10, 339 --384.

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

[Stephens & Poli, 2004] R. Stephens and R. Poli. *EC Theory "in Theory": Towards a Unification of Evolutionary Computation Theory*. In A. Menon (ed), *Frontiers of Evolutionary Computation*, pp. 129-156, Kluwer, Boston, 2004.

[Surry, 1998] P. D. Surry. *A Prescriptive Formalism for Constructing Domain-specific Evolutionary Algorithms*. PhD dissertation, University of Edinburgh, 1998.

[Syswerda, 1989] G. Syswerda. *Uniform crossover in genetic algorithms*. In J. D. Scha er, editor, Proceedings of the International Conference on Genetic Algorithms, San Mateo (CA), 1989. Morgan Kaufmann Publishers.

[Van der Vel, 1993] M. van de Vel. *Theory of Convex Structures*, Elsevier, Amsterdam, 1993.

[Whitley, 1994] D. Whitley. *A Genetic Algorithm Tutorial*. Statistics and Computing (4):65-85, 1994.

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