Geometric Unification of Evolutionary Algorithms

Alberto Moraglio

University of Essex, UK
amoragn@essex.ac.uk

Abstract. Evolutionary algorithms are only superficially different and can be unified within an axiomatic geometric framework by abstraction of the solution representation. This framework describes the evolutionary search in a representation-independent way, purely in geometric terms, paving the road to a general theory of evolutionary algorithms. It also leads to a principled design methodology for the crossover operator for any solution representation.

1 Context of the research

Evolutionary Algorithms (EAs) are successful and widespread general problem solving methods that mimic in a simplified manner biological evolution. Whereas all EAs share the same basic algorithmic structure, they differ in the solution representation - the genotype - and in the search operators employed - mutation and crossover - that are representation-specific. Is this difference only superficial? Is there a deeper unity encompassing all mutation and crossover operators beyond the specific representation, hence all EAs? So far, no one has been able to attack this question successfully and has proposed a general mathematical framework that unifies search operators for all solution representations.

In the research community there is a strong feeling that the EC field needs unification and systematization in a rational framework to survive its own exceptional growth (De Jong [4]). Beside De Jong, there are important researchers who have been promoting EC unification: Radcliffe pioneered a unified theory of representations [11], although he never used the word "unification". Riccardo Poli unified the schema theorem for traditional genetic algorithms and genetic programming [3]. Chris Stephens suggests that all evolutionary algorithms can be unified using the language of dynamical systems and coarse graining [1]. Franz Rothlauf has initiated a theory of representations [12].

2 Research and study

2.1 Research questions and goals

My research questions are:

1. *Possibility: Is the unification of evolutionary algorithms within a general mathematical framework possible?*
2. *Utility: What are the advantages and consequences of the unification?*

The significance of the unification is not obvious a priori and resides in the important consequences and insights brought about by seeing evolutionary algorithms from a new and more general viewpoint. For this reason the focus of my research is both on possibility and utility of unification.

My goals are: developing a *formal framework* for the unification, show that this framework helps to *rethink* various familiar aspects of evolutionary algorithms simplifying and clarifying their roles, show that the *unification is possible* and many preexisting representations and operators fit the framework, show that the formal framework can be used to do *crossover principled design* for any representation, show that the framework forms a solid basis for a *representation-independent theory* that applies to all evolutionary algorithms.

2.2 Current status: overview of the achievements

In this section, a short overview of the research achievements is reported. A sample of the results that will appear in the thesis is reported in section 4. Most of this research has been already published [5] [6] [8] [7] [9] or is about to be published.

1. *Possibility of unification*: mathematical unification is possible. Developed an axiomatic geometric framework for unification. The definition of search operators is axiomatic and intentionally does not involve the notion of representation: unification by abstraction of the representation[1].
2. *Clarification and simplification*: the change in perspective coming with unification completely reverses the orthodoxy [2] clarifying and simplifying many fundamental aspects of evolutionary algorithms. Clarified the consequences of this new perspective on fundamental notions: common search space for mutation and crossover, problem-independent and representation-independent formal evolutionary algorithm, simple fitness landscape of crossover, geometric format of problem knowledge, role of the EA designer, duality of neighborhood search and representation-based evolutionary search.
3. *Interesting scope of unification*: the significance of unification lies on how many interesting cases it encompasses. Shown that many interesting pre-existing operators for the most-used representations fit the requirements of the unification.
4. *Crossover principled design*: by reversing the abstraction and applying the abstract definition of crossover to a distance firmly rooted in a specific representation one obtains a formal recipe to build new crossovers for any representation. When the distance chosen as basis for the new crossover is

[1] This does not mean I regard solution representation as an "implementation detail" and unimportant. This means that the relationship between *representation and search operators*, which is ultimately what is relevant to the search, can be expressed in a geometric language that is representation-independent.

meaningful in terms of the problem addressed, the new operator is likely to perform well. This way of doing crossover principled design is the representation dual of the neighborhood search meta-heuristic path-relinking that suggests picking new solutions on a path (not necessarily shortest) in the search space connecting parent solutions. Unlike path-relinking that neglects altogether the underlying solution representation and does not show how to actually generate offspring (that is left as "implementation details"), geometric crossover tells exactly how to manipulate the syntax of the solution representation to build offspring solutions. Various examples of crossover design are given with good experimental results.

5. *Depth of the abstraction and general theory*: the significance of an abstraction lies on the kind of results that can be inferred using only the axioms the abstraction is based upon; if only trivialities encompassing all evolutionary algorithms can be inferred, the abstraction is not significant. The geometric abstraction is indeed significant. (i) A fundamental result is that all evolutionary algorithms with any solution representation and with search operators that fit the geometric framework do the same search, convex search. This is a simple, deep and important result arising from the geometric interpretation of mutation and crossover only. (ii) Convexity plays a major role in the way evolutionary algorithms search the space: the evolutionary search can be naturally recast from the point of view of the underlying metric convexity associated with the search space. This allows showing the correspondence of metric convex sets with inheritable genetic traits, and generalize in a representation-independent way the notion of schema. Then, doing coarse-graining of the equation of the evolutionary dynamics over the convex sets one obtains a representation-independent schema theorem. The importance of this result relies on the fact that it reconciles two fundamental notions that until now were separated: the notion of inheritance and the notion of fitness landscape. Specifying the representation-independent definition of inheritable trait (schema) for a distance rooted on a specific solution representation, one can reveal the syntactic appearance of schemata for any representation. When applied to DNA strands with edit distance, this can have important application in genetics to discover new genes. (iii) Knowing how all evolutionary algorithms search the space is preliminary to understand under what general condition on the fitness landscape they perform well. Fitness distribution, correlation of the fitness landscape and performance are studied together to show why positive correlation in the landscape makes geometric crossover and geometric mutation perform better than random search.

2.3 Future planning

From February to June 2006 focus on: writing-up the thesis. I have most of the material in the form of draft/submitted/published papers. Background activities: writing a journal paper, writing grant proposal and maybe submitting a paper to PPSN.

Future after PhD: ideally, I would like to stay in academia, becoming a lecturer and continuing this research.

2.4 Study

I am finishing my third year of PhD study. In theory, one could finish within 3 years; in practice, it is more common to do 3 years of research and then doing the writing-up and submitting the thesis the 4th year, the so-called completion year. Every 6 months there is a board meeting in which a panel of academics evaluate the progress of PhD students from the previous board meeting using a system with milestones and give suggestions/feedback on the research and research plan.

3 Methodology and thesis outline

3.1 Methodology: focus on unification

Unification is a delicate matter: besides proving that unification is possible in principle, one has to explain the consequences of the new angle on a network of related concepts. Plus, one has to go wide and show that many interesting cases are actually encompassed, but one also needs to go deep to show that the unification does capture some fundamental aspects common to all evolutionary algorithms and not only trivialities can be inferred for all evolutionary algorithms. Moreover one needs also to show that unification is not only theory for its own sake but that has practical advantages too. This all needed to be done within a time-window of a PhD study. To different extent I think I managed to address all the previous points. Naturally, since a complete and thorough unification is a monumental task, I focussed on particular topics that I felt had the priority and showed that a particular important territory is encompassed by the unification. By no means, I have exploited systematically each topic covered as much as it would have deserved. Indeed, each topic would have taken a PhD to be exhaustively addressed alone. Since the focus is the unification, for each topic I have shown instead how the geometric framework reveals its connection with all the others and this in turn had shed light on the specific topic itself.

3.2 Thesis outline

Title: *Geometric unification of evolutionary algorithms*
Thesis: Evolutionary algorithms are only superficially different and can be unified within an axiomatic geometric framework by abstraction of the solution representation. This framework describes the evolutionary search in a representation-independent way, purely in geometric terms, paving the road to a general theory of evolutionary algorithms. It also leads to a principled design methodology for the crossover operator for any solution representation.

4 Results

In this section I report a sample of the results that will appear in the thesis. Section 4.1 introduces the representation-independent definition of geometric crossover and describes how crossover connects with fitness landscape. The merit of the geometric framework here is that it simplifies enormously this connection: unlike the established paradigm [2] the geometric interpretation of crossover in the landscape is simple and intuitive. Section 4.2 shows that a number of recombination operators for 5 important representations are actually specific instances

of geometric crossover. Section 4.3 gives an example of crossover principled design for the TSP problem. Experimental results (not reported) show that this crossover performs extremely well. Finally, section 4.4 gives an example of a very general representation-independent theoretical result that holds for all evolutionary algorithms using instances of geometric crossover.

4.1 Geometric framework

Geometric preliminaries The terms *distance* and *metric* denote any real valued function that conforms to the axioms of identity, symmetry and triangular inequality. A simple connected graph is naturally associated to a metric space via its *path metric*: the distance between two nodes in the graph is the length of a shortest path between the nodes. Similarly, an edge-weighted graph with strictly positive weights is naturally associated to a metric space via a *weighted path metric*.

In a metric space (S, d) a *closed ball* is the set of the form $B(x; r) = \{y \in S | d(x, y) \leq r\}$ 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 $[x; y] = \{z \in S | d(x, z) + d(z, y) = d(x, y)\}$ where $x, y \in S$ are called extremes of the segment. Metric ball and metric segment generalize the familiar notions of ball and segment in the Euclidean space to any metric space through distance redefinition. These generalized objects look quite different under different metrics. Notice that a metric segment does not coincide to a shortest path connecting its extremes (*geodesic*) as in an Euclidean space. In general, there may be more than one geodesic connecting two extremes; the metric segment is the union of all geodesics.

We assign a structure to the solution set by endowing it with a notion of distance d. $M = (S, d)$ is therefore a solution *space* and $L = (M, g)$ is the corresponding fitness landscape (where g is the fitness function). Notice that d is arbitrary and need not have any particular connection or affinity with the search problem at hand.

Geometric crossover definition The following definitions are *representation-independent* therefore crossover is well-defined for any representation. It is only *function of the metric d* associated with the search space being based on the notion of metric segment.

Definition 1. *(Image set) The* image set $Im[OP]$ *of a genetic operator OP is the set of all possible offspring produced by OP with non-zero probability.*

Definition 2. *(Geometric crossover) A binary operator is a geometric crossover under the metric d if all offspring are in the segment between its parents.*

Definition 3. *(Uniform geometric crossover) Uniform geometric crossover UX is a geometric crossover where all z laying between parents x and y have the same*

probability of being the offspring:

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

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

A number of general properties for geometric crossover and mutation have been derived.

Geometric crossover landscape Geometric operators are defined as functions of the distance associated to the search space. However, the search space does not come with the problem itself. The problem consists only of a fitness function to optimize, that defines what a solution is and how to evaluate it, but it does not give any structure on the solution set. The act of putting a structure over the solution set is part of the search algorithm design and it is a designer's choice. A fitness landscape is the fitness function plus a structure over the solution space. So, for each problem, there is one fitness function but as many fitness landscapes as the number of possible different structures over the solution set. In principle, the designer could choose the structure to assign to the solution set completely independently from the problem at hand. However, because the search operators are defined over such a structure, doing so would make them decoupled from the problem at hand, hence turning the search into something very close to random search.

In order to avoid this one can exploit problem knowledge in the search. This can be achieved by carefully designing the connectivity structure of the fitness landscape. For example, one can study the objective function of the problem and select a neighborhood structure that couples the distance between solutions and their fitness values. Once this is done problem knowledge can be exploited by search operators to perform better than random search, even if the search operators are problem-independent (as in the case of geometric crossover and mutation).

Under which conditions is a landscape well-searchable by geometric operators? As a rule of thumb, geometric mutation and geometric crossover work well on landscapes where the closer pairs of solutions, the more correlated their fitness values. Of course this is no surprise: the importance of landscape smoothness has been advocated in many different context and has been confirmed in uncountable empirical studies with many neighborhood search meta-heuristics [10].

4.2 Representation unification

We consider the application of geometric crossover to five important representations: real vectors, n-ary strings, permutations, variable-size sequences and syntactic trees. The aim here is to give a portfolio of concrete examples of application of the theoretical ideas outlined before. We will see how geometric crossover

unifies pre-existing crossovers for different representations (real vectors, binary strings, permutations, syntactic trees), how it casts new interpretations of pre-existing crossovers (crossover for permutations are sorting algorithms), how it guides to the principled design and implementation of new crossovers for new representations (variable-size sequences), how it may connect artificial and biological evolution (variable-size sequences) and how it helps to understand what is the search space and distance associated to a pre-existing crossover operator (syntactic trees).

Real vectors Pre-existing crossover operators, both blending type crossovers and discrete type recombinations fit the definition of geometric crossover naturally (see fig. 1). The extended version of blending crossovers does not fit the definition of geometric crossover (for *any* distance).

Binary strings The Hamming scheme is the association scheme where the elements are vectors of length d over some alphabet of size q. The Hamming distance of two vectors is the number of coordinates where they differ. The Hamming graph H(d,q) is the graph that describes the distance-1 relation in the Hamming scheme. It is the direct product of d complete graphs of size q. For d = 2 one gets the qq grid, also known as the lattice graph of order q. The Hamming graphs H(d,2) are the familiar hyper-cubes associated with binary strings of size d. The search operators are implemented by syntactic manipulation of strings equivalent to the geometric transformations required by the search operators. A geodesic between two points is a shortest path between two points. A shortest path in the edit distance graph is a minimal sequence of edit moves that transforms the syntax of one parent to the syntax of the other. In the case of binary string the edit move is bit-flip and in the case of multary strings the edit move is a substitution. The uniform crossover for binary strings is equivalent to picking any minimal sequence of bit-flip (that changes at most once one bit) that transforms one parent string into the other and interrupt the transformation somewhere in the middle. The one-point crossover (selecting one crossover point and swapping strings tails) is equivalent to applying a macro edit-move equivalent to the application of a specific minimal sequence of edit moves connecting the two parent strings interrupted somewhere in the middle.

All traditional mask-based crossovers for binary and multary strings are geometric under Hamming distance. See also fig. 2.

Permutations Pre-existing operators for permutations are sorting algorithms in disguise because picking offspring on the shortest path based on edit distances for permutations translates into picking offspring on a minimal sorting trajectories between parent permutations (see fig. 3). For example, PMX is geometric under swap distance.

Syntactic trees The search space and distances associated with pre-existing genetic operators for syntactic trees are little understood. Here, differently form

the previous representations, we want to use the geometric definitions not to guide the design and implementation of new operators for syntactic trees but rather we want to find the distance, hence the search space, associated to pre-existing operators if any of such distance exists. There are various notions of distance defined for GP trees. Distances among GP trees are used to (1) maintain diversity in the population and (2) predict performance (fitness-distance correlation). If the distance employed does not match the operator used its use is meaningless. So far it has been difficult to show that a certain distance matches a certain operator. Here we propose a new distance, the structural hamming distance (SHD) (a variation of the well-known structural distance for trees), that perfectly matches with Poli's Homologous crossover. Fig.4 shows an example of how SHD and hyperschema connect.

Koza's crossover is not geometric: there is provably no distance for which it is a geometric crossover. Poli's homologous crossovers family (1-point crossover, uniform crossover, homologous crossover, point mutation) is geometric under SHD.

Variable-length sequences Let us consider a recombination for variable-length sequences that requires an inexact sequence alignment before applying the traditional crossover on the alignment.

```
Parent1=AGCACACA
Parent2=ACACACTA

best inexact alignment by dynamic programming:
AGCA|CAC-A
A-CA|CACTA

Child1=AGCACACTA
Child2=ACACACA
```

This crossover and its generalization can be proven to be geometric under Levinshtein edit distance (insertion, deletion, substitution).

Does it have biological significance? Edit distance is a very meaningful distance to compare DNA strands. The present model of crossover (based on perfect alignment) cannot explain molecular evolution. Molecular evolution is tried to be explained by mutation only or by unequal crossover (imperfect crossover). Two DNA strands before crossover align on their contents to minimize the free-energy admitting gaps, compressions and mismatch alignment (molecular annealing). Geometric crossover based on edit-distance could be a better model of biological crossover in that model more realistically the effect of molecular annealing and could explain molecular evolution.

4.3 Crossover principled design: TSP

We consider a solution as a tour of cities and, therefore, rather than being defined for permutations geometric crossover is defined over *circular permutations*.

In the case of circular permutations, the block-reversal move is the notion of edit distance that makes sense for TSP. In a single application to a tour, this does the minimal change to the adjacency relation among elements in the permutation. This move is the well-known 2-change move, and it is the basis for successful local search algorithms for TSP . Figure 5 shows the possible offspring (the segment) between two circular (parent) permutations under topological crossover.

Analogously to the linear case, the circular permutations in the segment under reversal distance are those laying in a minimal sorting trajectory from a parent circular permutation to the other. Sorting circular permutations by reversals is NP-hard. So, *the topological crossover under this notion of distance cannot be implemented efficiently.*

Sorting circular permutations by reversals is tightly connected with the problem of sorting linear permutations by reversals. So all the algorithms developed for the latter task can be used with minor modifications also for the former. Sorting linear permutations by reversals is NP-hard too. However a number of approximation algorithms (running in polynomial time) exist to solve this problem within a bounded error from the optimum. This allows implementing efficiently approximate crossovers whose image set is a super-set of that of the exact crossover. We have implemented this crossover and found that it outperforms edge recombination that is known to be very good for TSP.

4.4 Representation-independent theory: convex evolutionary search

Using the axioms of distance and the definition of geometric crossover we can prove a main result: an evolutionary algorithm using geometric crossover with any probability distribution, any kind of representation, any problem, any selection and replacement mechanism, does the same search: convex search. Proof based on abstract convexity (axiomatic geodesic convexity) and axiomatization of search process (abstract search process). See Fig. 6 for an example of convex search in the intuitive case of Euclidean space.

5 Achievements and future directions

Achievements: this work answers fully the research questions, now there is no doubt about possibility and utility of unification, and explores the main research directions the unification opens up discovering new territories that now are ready to be conquered. For its own nature, a unification project can be evaluated only at the end when the whole picture is laid down, since each block reinforces the significance of unification as a unity only when put aside all the others. My hope is that the overall picture that emerges looks like a harmonious balance between all aspects of the unification.

Future directions: there are three main roads to follow from here.

1. *The road to principled design*: the first one is to unify a design methodology merging path-relinking and geometric crossover design.

2. *The road to computational complexity*: the second road is developing a theory which aim is general computational complexity results for evolutionary algorithms.
3. *The road to biological evolution*: the third road is casting a computational perspective on biological evolution to understand why is able to do "intelligent design" so effectively and efficiently without intelligence.

6 Feedback

Unification is a delicate matter: to deliver a meaningful unification, on one hand, many different aspects needed to be explored in parallel without losing sight of the overall objective. On the other hand, every topic needed to be exploited up to a point for which the consequences of unification on that particular direction were crystal clear. This all needed to be done within a time-window of PhD study. To different extent I think I managed to address all the important aspects of unification. However a question remains: is the overall emerging picture a harmonious balance between all aspects?

7 Acknowledgments

The author would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

1. A. Zamora C. R. Stephens. Ec theory: A unified viewpoint. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1394–1405, 2003.
2. T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, 1995.
3. W. B. Langdon and R. Poli. *Foundations of Genetic Programming.* Springer, 2001.
4. A. Menon, editor. *Frontiers of Evolutionary Computation.* Springer, 2004.
5. A. Moraglio and R. Poli. Topological interpretation of crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1377–1388, 2004.
6. A. Moraglio and R. Poli. Geometric crossover for the permutation representation. *Technical Report CSM-429, University of Essex*, 2005.
7. A. Moraglio and R. Poli. Geometric landscape of homologous crossover for syntactic trees. In *Proceedings of CEC 2005*, pages 427–434, 2005.
8. A. Moraglio and R. Poli. Topological crossover for the permutation representation. In *GECCO 2005 Workshop on Theory of Representations*, 2005.
9. A. Moraglio, R. Poli, and R. Seehuus. Geometric crossover for biological sequences. In *Proceedings EuroGP (to appear)*, 2006.
10. P. M. Pardalos and M. G. C. Resende, editors. *Handbook of Applied Optimization.* Oxford University Press, 2002.
11. N. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.
12. F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms.* Springer, 2002.

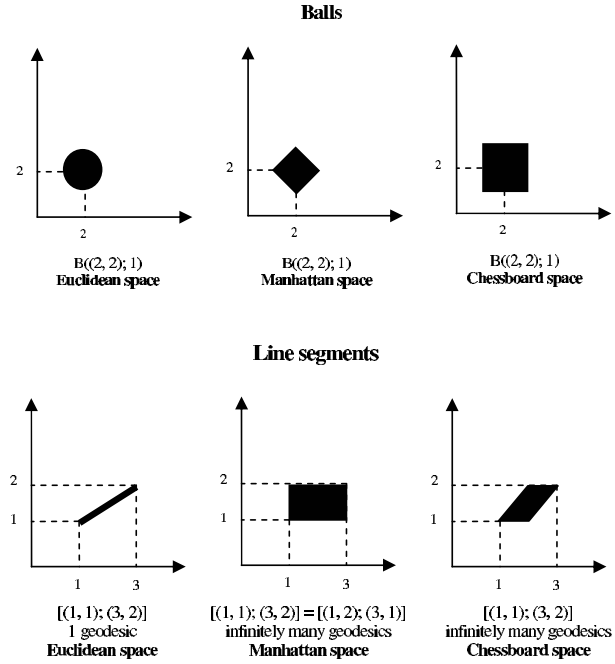

Fig. 1. Examples of balls and segments in Minkowsky spaces

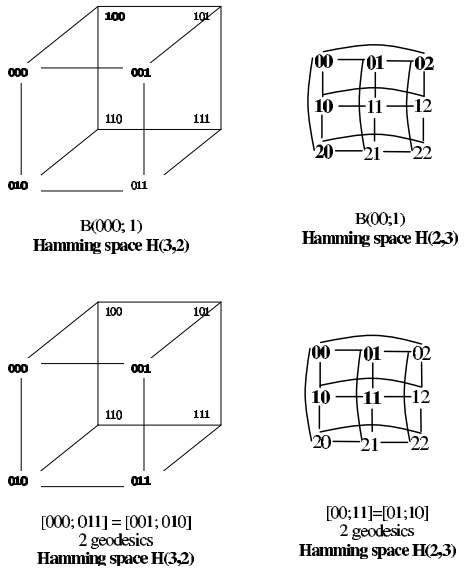

Fig. 2. Examples of balls and segments in Hamming spaces

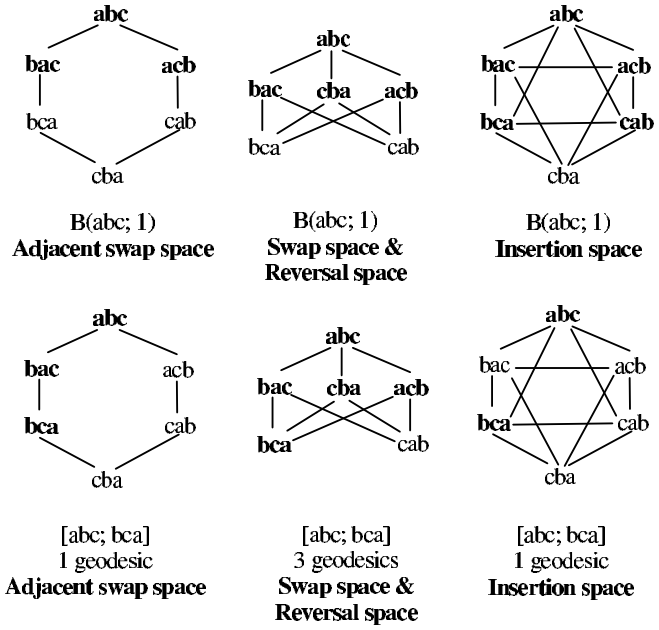

Fig. 3. Examples of balls and segments in Cayley spaces

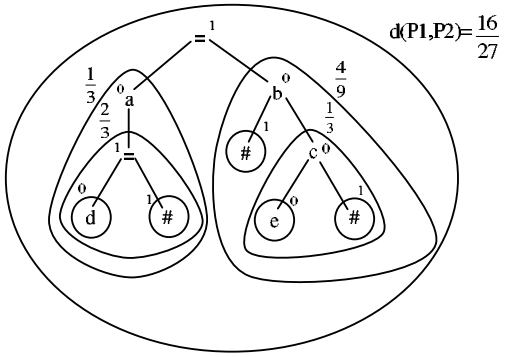

Fig. 4. Structural Hamming distance and hyper-schema

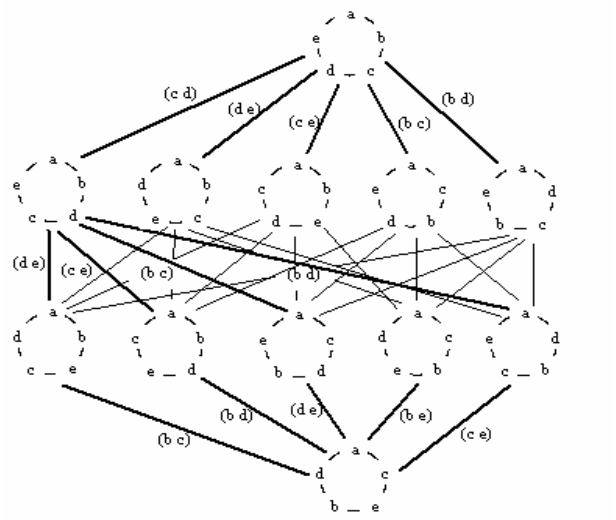

Fig. 5. Example of topological crossover between two circular permutations.

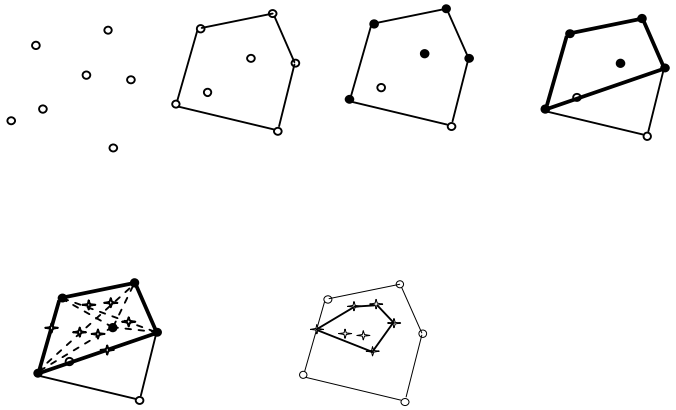

Fig. 6. Geometric crossover + selection = convex search.