Quotient Geometric Crossovers

Yourim Yoon
School of Computer Science & Engineering
Seoul National University
Sillim-dong, Gwanak-gu, Seoul, 151-744, Korea

yryoon@soar.snu.ac.kr

Alberto Moraglio
Department of Computer Science
University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK

amoragn@essex.ac.uk

Yong-Hyuk Kim
Department of Mathematical Sciences
Seoul National University
Sillim-dong, Gwanak-gu, Seoul, 151-747, Korea

yhdfly@soar.snu.ac.kr

Byung-Ro Moon
School of Computer Science & Engineering
Seoul National University
Sillim-dong, Gwanak-gu, Seoul, 151-744, Korea

moon@soar.snu.ac.kr

ABSTRACT

Geometric crossover is a representation-independent definition of crossover based on the distance of the search space interpreted as a metric space. It generalizes the traditional crossover for binary strings and other important recombination operators for the most frequently used representations. Using a distance tailored to the problem at hand, the abstract definition of crossover can be used to design new problem specific crossovers that embed problem knowledge in the search. In previous work we have started studying how metric transformations of the distance associated with geometric crossover affect the original geometric crossover. In particular, we focused on the product of metric spaces. This metric transformation gives rise to the notion of product geometric crossover that allows to build new geometric crossovers combining pre-existing geometric crossovers in a simple way. In this paper, we study another metric transformation, the quotient metric space, that gives rise to the notion of *quotient geometric crossover*. This turns out to be a very versatile notion. We give many examples of application of the quotient geometric crossover.

Keywords

Geometric crossover, metric transformation, quotient metric space, quotient geometric crossover

1. INTRODUCTION

Geometric crossover and geometric mutation are representation-independent search operators that generalize many pre-existing search operators for the major representations used in evolutionary algorithms, such as binary strings [7], real vectors [7], permutations [6], syntactic trees [3] and sequences [1]. They are defined in geometric terms using the notions of line segment and ball. These notions and the corresponding genetic operators are well-defined once a notion of distance in the search space is defined. Defining search operators as functions of the search space is opposite to the standard way [13] in which the search space is seen as a function of the search operators employed. This viewpoint greatly simplifies the relationship between search operators and fitness landscape and has allowed us to give simple *rules-of-thumb* to build crossover operators that are likely to perform well.

Theoretical results of metric spaces can naturally lead to interesting results for geometric crossover. In particular, in previous work [5] we have shown that the notion of *metric transformation* has great potential for geometric crossover. A metric transformation is an operator that constructs new metric spaces from pre-existing metric spaces: it takes one or more metric spaces as input and outputs a new metric space. The notion of metric transformation becomes extremely interesting when considered together with distances firmly rooted in the syntactic structure of the underlying solution representation (e.g., edit distance). In these cases it gives rise to a simple and *natural interpretation in terms of syntactic transformations*.

In previous work [5] we have extended the geometric framework introducing the notion of product crossover associated with the Cartesian product of metric spaces. This is a very important tool that allows one to build new geometric crossovers customized to problems with mixed representations by combining pre-existing geometric crossovers in a straightforward way. Using the product geometric crossover, we have also shown that traditional crossovers for symbolic vectors and blend crossovers for integer and real vectors are geometric crossover.

In this paper we extend the geometric framework introducing the important notion of *quotient geometric crossover*.

The metric transformation associated with it is the quotient metric space. Quotient geometric crossover reduces the search space actually searched by geometric crossover, introduces problem knowledge in the search by using a distance better tailored to the specific solution interpretation and it can be used to remove the inherent search bias of the original crossover operator. The notion of quotient geometric crossover is extremely versatile, so we will illustrate its many applications.

The paper is organized as follows. In Section 2, we present the geometric framework including the notion of geometricity-preserving transformation. In Section 3, we introduce the notion of quotient geometric crossover. In Section 4, we study several useful applications related with quotient geometric crossover. In Section 4.1 and 4.2, we show how previous work on a grouping problem [2] and graphs can be recast and understood more simply in terms of quotient geometric crossover. Here, quotient geometric crossover is used to filter out inherent redundancy in the solution representation. In Section 4.3, we apply the quotient geometric crossover in a completely different way: we show how homologous crossover for variable-length sequences [1] can be understood as a quotient geometric crossover. In Section 4.4, we present a thorough study of the application of quotient geometric crossover to glued real-valued representations that remove the inherent bias of geometric crossover based on simple real-valued representations. In Section 4.5, we consider functional representation and shows how the concept of quotient geometric crossover is connected to the search of the functions. Genetic programming is shown as an example. In Section 4.6, we explain that quotient geometric crossover can be used to understand how crossover and neutral code interact. In Section 5, we give conclusions.

2. GEOMETRIC FRAMEWORK

2.1 Geometric Preliminaries

In the following we give necessary preliminary geometric definitions and extend those introduced in [7, 3]. The following definitions are taken from [10].

The terms *distance* and *metric* denote any real valued function that conforms to the axioms of identity, symmetry and triangular inequality. In a metric space (S, d) a *line segment* (or closed interval) is the set of the form $[x; y]_d = \{z \in S \mid d(x, z) + d(z, y) = d(x, y)\}$ where $x, y \in S$ are called extremes of the segment. Metric segment generalizes the familiar notions of segment in the Euclidean space to any metric space through distance redefinition. 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 S by endowing it with a notion of distance d. $M = (S, d)$ is therefore a solution *space* and (M, f) is the corresponding fitness landscape, where f is the fitness function over S.

2.2 Definition of Geometric Crossover

The following definitions are *representation-independent* therefore applicable to any representation.

DEFINITION 1 (IMAGE SET). *The* image set $Im[OP]$ *of a genetic operator OP is the set of all possible offspring produced by OP.*

DEFINITION 2 (GEOMETRIC CROSSOVER). *A binary operator GX is a* geometric crossover *under the metric d if all offspring are in the segment between its parents x and y, i.e.,*

$$Im[GX(x, y)] \subseteq [x; y]_d.$$

A number of general properties for geometric crossover and geometric mutation have been derived in [7] where we also showed that traditional crossover is geometric under Hamming distance.

2.3 Formal Evolutionary Algorithm and Problem Knowledge

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 is the case of geometric crossover and geometric mutation). Indeed, the fitness landscape is a knowledge interface between the problem at hand and a formal, problem-independent search algorithm.

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 [19]. We operate according to the following rule-of-thumbs:
Rule-of-thumb 1: if we have a good distance for the problem at hand then we have good geometric mutation and good geometric crossover.
Rule-of-thumb 2: a good distance for the problem at hand is a distance that makes the landscape "smooth."

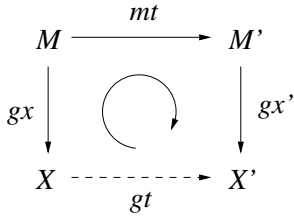### 2.4 Geometricity-Preserving Transformations

Figure 1: Commutative diagram linking metric and crossover transformations.

In previous work we have proven that a number of important pre-existing recombination operators for the most frequently used representations are geometric crossovers. We have also applied the abstract definition of geometric crossover to distances firmly rooted in a specific solution representation and designed brand-new crossovers. An appealing way to build new geometric crossovers is starting from recombination operators that are known to be geometric and deriving new geometric crossovers by *geometricity-preserving transformations/combinations* that when applied to geometric crossovers, return geometric crossovers.

The definition of geometric crossover is based on the notion of metric. Therefore, a natural starting point to seek geometricity-preserving transformations is to consider transformations of the underlying metrics that are known to return metric spaces and study how the geometric crossover associated to the transformed metric space relates with the geometric crossover associated with the original metric space.

There are a number of metric space transformations [10, 20] that are potentially of interest for geometric crossover: sub-metric space, product space, quotient metric space, gluing metric space, combinatorial transformation, non-negative combinations of metric spaces, Hausdorff transformation, concave transformation, and biotope transformation.

Let us consider the geometric crossover X associated to the original metric space M, and the geometric crossover X' associated to the transformed metric space $M' = mt(M)$ where mt is the metric transformation. The functional relationship among metric spaces and geometric crossovers can be nicely expressed through a commutative diagram (Figure 1). gx means application of the formal definition of geometric crossover and gt means *induced geometricity-preserving* crossover transformation associated to the metric transformation mt. This diagram becomes remarkably interesting when the metric transformation mt is associated to an induced geometricity-preserving crossover transformation gt that has a simple interpretation in terms of syntactic manipulation. This indeed allows one to get new geometric crossovers starting from recombination operators that are known to be geometric by simple *geometricity-preserving syntax manipulation*.

We study those metric-preserving transformations which induced geometricity-preserving transformations have a simple and natural interpretation on the solution representation.

3. QUOTIENT GEOMETRIC CROSSOVER

3.1 Quotient Metric Space

Let (S, d) be a metric space and $\sim$ be an equivalence relation on S. Consider the *quotient space* $S/\sim$. Now we will give a metric on $S/\sim$ induced by the original metric d on S.

DEFINITION 3 (INDUCED DISTANCE MEASURE).
For $\bar{x}, \bar{y} \in S/\sim$,

$$d_\sim(\bar{x}, \bar{y}) := \inf_{x \in \bar{x}, y \in \bar{y}} d(x, y).$$

Then, the following theorem holds [8].

THEOREM 1. *If the equivalence relation arises from an isometry subgroup*[1], $d_\sim$ *is a metric on* $S/\sim$.

This metric space $(S/\sim, d_\sim)$ is called *quotient metric space*. Later we will directly prove that $d_\sim$ becomes a metric instead of showing that its related equivalence relation $\sim$ comes from an isometry subgroup.

In a metric space (S, d) a *quotient line segment* is the set of the form $[x; y]_{d_\sim} = \{z \in S \mid d_\sim(\bar{x}, \bar{z}) + d_\sim(\bar{z}, \bar{y}) = d_\sim(\bar{x}, \bar{y}), \bar{z} \in S/\sim\}$ where $\bar{x}, \bar{y} \in S/\sim$. Now we can define quotient geometric crossover.

DEFINITION 4 (QUOTIENT GEOMETRIC CROSSOVER).
A binary operator GX_q is a quotient geometric crossover *under the metric d and the equivalence relation $\sim$ if all offspring are in the quotient line segment between its parents x and y, i.e., $GX_q(x, y) \subseteq [x; y]_{d_\sim}$.*

3.2 Genotype-Phenotype Mapping

The notion of quotient geometric crossover is important because it lies at the heart of the relation between geometric crossover and genotype-phenotype mapping as we illustrate in the following.

Genotype means solution representation: some structure that can be stored in a computer and manipulated. Phenotype means solution itself without any reference to how it is represented. Sometimes it is possible to have a one-to-one mapping between genotypes and phenotypes, so the distinction between genotype and phenotype becomes purely formal. However in many interesting cases phenotypes cannot be represented uniquely by genotypes. So the same phenotype is represented by more than one genotypes. In this case we say that we have a *redundant representation*. For example, to represent a graph we need to label its nodes and then we can represent it using its adjacency matrix. This representation is redundant: the same graph can be represented with more than one adjacency matrix by relabeling its nodes.

There are quite a few problems in that it is hard to represent one phenotype by just one genotype using traditional representations. Roughly speaking, redundant representation leads to severe loss of search power in genetic algorithms, in particular, with respect to traditional crossovers [9]. To alleviate the problems caused by redundant representation, a number of methods such as adaptive crossover have been proposed [11, 16, 17, 21]. Among them, a technique called *normalization*[2] is representative. It transforms the genotype of a parent to another genotype to be consistent with the other parent so that the genotype contexts of the parents are as similar as possible in crossover. There have been a number of successful studies using normalization. An extensive survey about normalization is appeared in [9].

---

[1]For details, see [8].
[2]The term of *normalization* is firstly appeared in [14]. However, it is based on the adaptive crossovers proposed in [16, 17].

Although many of studies about normalization did not use the concept of distance, once a distance d_G on the genotypes G is defined, we can formally redefine the normalization p_2' of the second parent p_2 to the first p_1 as follows:

$$p_2' := \operatorname*{argmin}_{s \in w(p_2)} d_G(p_1, s),$$

where $w(s)$ is the set of all the genotypes with the same phenotype as the genotype s. The use of distance to define normalization is important because it generalizes and makes rigorous the notion of normalization for *any solution representation*.

Now we formally present the general relation between geometric crossover and genotype-phenotype mapping. The concept of normalization defined by distance is closely related with the quotient geometric crossover. Let us consider genotype-phenotype mappings $g : G \to P$ that are not injective (redundant representation). The mapping g induces a natural equivalence relation $\sim$ on the set of genotypes: *genotypes with the same phenotype belong to the same class*. Given a distance d_G on genotypes G, the quotient with the relation $\sim$ produces a distance d_P on the phenotypes P: $P = G/\sim$ and $d_P(\bar{x}, \bar{y}) = \inf_{x \in \bar{x}, y \in \bar{y}} d_G(x, y)$.

By applying the formal definition of geometric crossover to the metric spaces (G, d_G) and (P, d_P), we obtain the geometric crossovers X_G and X_P, respectively. X_G searches the space of genotypes and X_P searches the space of phenotypes. Searching the space of phenotypes has a number of advantages: (i) it is smaller than the space of genotypes, hence quicker to search (ii) the phenotypic distance is better tailored to the underlying problem, hence the corresponding geometric crossover works better (iii) the space of phenotypes has different geometric characteristics from the genotypic space. This can be used to remove unwanted bias from geometric crossover.

However, the crossover X_P cannot be directly used itself because it recombines phenotypes that are objects that cannot be directly represented. The quotient geometric crossover allows us to search the space of phenotypes with the crossover X_P *indirectly* by manipulating the genotypes G. This is possible because for the commutative diagram there exists an induced geometricity-preserving transformation gt of the genotypic crossover X_G that allows us to use the genotypic representation to implement a geometric crossover in the space of phenotypes $gt(X_G)$ without making explicit use of phenotypes. The type of the transformation gt depends on the type of the equivalence relation $\sim$ used in the quotient of the underlying metric space that in turns depends on the underlying syntax of the solution representation. It may happen that the induced geometricity-preserving transformation may be difficult to implement and/or computationally intractable. In these cases, it may not be feasible using an exact equivalent of the phenotypic geometric crossover, but an approximation may be preferable and still retaining most of the advantages of the exact equivalent.

In the following section we consider a number of equivalence classes for the quotient operation and its related induced genotypic crossover transformation.

4. APPLICATIONS

4.1 Groupings

Now we introduce an example class of problems using the

normalization method. The problem that we consider is the *grouping* problem [12]. Grouping problems are commonly concerned with partitioning given item set into mutually disjoint subsets. Examples belonging to this class of problems are multiway graph partitioning, graph coloring, bin packing, and so on. Grouping representation is also used to solve the joint replenishment problem, which is a well-known problem appeared in the field of industrial engineering [18]. In this class of problems, the normalization decreased the problem difficulty and led to notable improvement in performance.

Most normalization studies for grouping problems were focused on the k-way partitioning problem. In the problem, the k-ary representation, in which k subsets are represented by the integers from 0 to $k-1$, has been generally used. In this case, a phenotype (a k-way partition) is represented by $k!$ different genotypes. In the problem, a normalization method was used in [14]. Other studies for the k-way partitioning problem used the same technique [9, 15]. In sense that normalization pursues the minimization of genotype inconsistency among chromosomes, in previous work [4], we proposed an optimal, efficient normalization method for grouping problems and a distance measure, the *labeling-independent distance*, that eliminates this dependency completely.

Given two k-ary encodings $\mathfrak{a}, \mathfrak{b} \in U = \{1, 2, \ldots, k\}^n$ (fixed-length vectors on a k-ary alphabet) and the Hamming distance H in U, we define the *labeling-independent distance* LI associated to H as follows:

$$LI(\mathfrak{a}, \mathfrak{b}) := \min_{\sigma, \sigma' \in \Sigma_k} H(\mathfrak{a}_\sigma, \mathfrak{b}_{\sigma'})$$

where Σ_k is the set of all permutations of length k and $\mathfrak{a}_\sigma$ is a permuted encoding of $\mathfrak{a}$ by a permutation σ, i.e., the ith element a_i of $\mathfrak{a}$ is transformed into $\sigma(a_i)$. Then, labeling-independent distance LI is a *pseudo-metric*[3] on U (see [4]).

Given an element $\mathfrak{a} \in U$, since H is a metric, there are $k!$ elements such that the labeling-independent distance LI to $\mathfrak{a}$ is zero. If the labeling-independent distance LI between two elements is equal to zero, we define them to be *in relation* $\sim$. Then, the relation $\sim$ is an equivalence relation (see [4]). $(U/\sim, LI^4)$ is a metric space, i.e., the labeling-independent distance LI is a metric on $U/\sim$ (see [4]).

We can design a quotient geometric crossover based on the labeling-independent metric.

DEFINITION 5 (LABELING-INDEPENDENT CROSSOVER). *Normalize the second parent to the first under the Hamming distance H. Do the normal crossover using the first parent and the normalized second parent.*

THEOREM 2. *The labeling-independent crossover is geometric under the metric LI [2].*

- *Genotypes G*: labeled partitions represented as vectors of symbols

---

[3]Formally, the term *distance* or *metric* denotes any real valued function that conforms to the axioms of identity, symmetry and triangular inequality. A distance for which the axiom of identity is relaxed so that distance zero does not necessarily implies equality (but equality still implies distance zero) is called *pseudo-metric*.

[4]Strictly speaking, this labeling-independent distance is not exactly the same as LI on U because it is a metric on $U/\sim$. However, the definition of LI on $U/\sim$ can be naturally induced from LI on U, i.e., $LI(\bar{\mathfrak{a}}, \bar{\mathfrak{b}}) := \min_{\sigma, \sigma' \in \Sigma_k} H(\mathfrak{a}_\sigma, \mathfrak{b}_{\sigma'})$.

- *Phenotypes P*: unlabeled partitions
- *Equivalence relation $\sim$*: labeled partitions with the same partition structure
- *Distance on genotypes d_G*: Hamming distance
- *Distance on phenotypes d_P*: labeling-independent distance
- *Crossover on genotypes X_G*: traditional crossover for vectors
- *Crossover of phenotypes X_P*: label normalization before traditional crossover
- *Induced crossover transformation gt*: label normalization

The benefit of understanding normalization for grouping problems in terms of quotient geometric crossover is the possibility of understanding the benefit of normalization in terms of landscape analysis. We have done this in previous work [2].

4.2 Graphs

In this subsection, we consider any problem naturally defined over a graph in which the fitness of the solution does not depend on the labels on the nodes but only on the structural relationship, i.e., edge between nodes.

Formally, let $A \in \mathfrak{M}_n$ be the adjacency matrix of a labeled graph using labels of n nodes and let P be an $n \times n$ permutation matrix[5]. Then the matrix PA means the labeled graph obtained by relabeling A according to the permutation represented by P. The fitness $f : \mathfrak{M}_n \to \mathbb{R}$ satisfies that for every $A \in \mathfrak{M}_n$ and every permutation matrix P, $f(A) = f(PA)$.

Let $(\mathfrak{M}_n, H)$ be a metric space on the labeled graphs under the Hamming distance H. Notice that this metric is labeling-dependent. In particular, $H(A, PA)$ may not be zero although A and PA represent the same structure. If A is equal to PA' for some permutation matrix P, we define A and A' to be *in relation* $\sim$, i.e., $A \sim A'$. Then, the relation $\sim$ is an equivalence relation.[6]

An *unlabeled graph* $\mathfrak{g}$ is the equivalence class of all its labeled graphs, i.e., $\mathfrak{g}(A) = \{PA \mid P \text{ is a permutation matrix}\}$. *unlabeled-graph space* $\mathfrak{M}_n / \sim$ is the set of all equivalence classes partitioning the set $\mathfrak{M}_n$.

We define *induced distance measure LI* on $M_n / \sim$ as follows: for each $\mathfrak{g}, \mathfrak{g}' \in \mathfrak{M}_n / \sim$,

$$LI(\mathfrak{g}, \mathfrak{g}') := \min_{A \in \mathfrak{g}, A' \in \mathfrak{g}'} H(A, A').$$

Then, $(\mathfrak{M}_n / \sim, LI)$ is a metric space, i.e., LI is a metric on $\mathfrak{M}_n / \sim$.[7] It shows that the metric space $(\mathfrak{M}_n, H)$ induces a quotient metric space $(\mathfrak{M}_n / \sim, LI)$.

DEFINITION 6 (LABELING-INDEPENDENT CROSSOVER). *Do the graph matching of the second parent p_2 to the first p_1 under the Hamming distance H, i.e.,*

$$p_2' := \operatorname*{argmin}_{A \in \mathfrak{g}(p_2)} H(p_1, A).$$

Do the normal crossover using the first parent p_1 and the graph-matched second parent p_2'.

The following theorem shows that the labeled-graph geometric crossover for $(\mathfrak{M}_n, H)$ induces the unlabeled-graph geometric crossover for $(\mathfrak{M}_n / \sim, LI)$.

THEOREM 3. *The labeling-independent crossover is geometric under the metric LI.*

PROOF. Omitted by space limit. □

The labeling-independent crossover is defined over unlabeled graphs $\mathfrak{M}_n / \sim$. This space is much smaller than labeled graphs $\mathfrak{M}_n$. More precisely, $|\mathfrak{M}_n / \sim| = |\mathfrak{M}_n| / n!$. This means that the more the labels are, the smaller the unlabeled-graph space is compared with the labeled-graph space. Smaller space means better performance given the same amount of evaluations.

The previous theorem tells how to guide the implementation using graph matching for specific geometric crossovers. To implement the geometric crossover over unlabeled graphs, we need to use labeled graphs. The labeling results are necessary to represent and handle the solution, even if in fact it is only an auxiliary function and can be considered not being part of the problem to solve. Graph matching before crossover allows to implement the geometric crossover on the unlabeled-graph space using the corresponding geometric crossover over the auxiliary space of the labeled graph after graph matching.

- *Genotypes G*: labeled graphs with the same number of nodes represented as adjacency matrices of the same size
- *Phenotypes P*: unlabeled graphs
- *Equivalence relation $\sim$*: adjacency matrices with the same underlying unlabeled graph
- *Distance on genotypes d_G*: Hamming distance between adjacency matrices
- *Distance on phenotypes d_P*: labeling-independent distance between unlabeled graphs. This equals the *edge edit distance*.
- *Crossover on genotypes X_G*: traditional crossover on adjacency matrices seen as vectors
- *Crossover of phenotypes X_P*: graph matching before traditional crossover on adjacency matrices
- *Induced crossover transformation gt*: graph matching

The benefit of applying the quotient geometric crossover on graphs is the design of a crossover better tailored to graphs. The notion of graph matching before crossover arises directly from the definition of quotient geometric crossover. Graphs are very important because they are ubiquitous. In future work we will test this crossover on some applications. Graphs and groupings can be seen as particular cases of labeled structures in which the fitness of a solution depends only on the structure and not on the specific labeling. In future work we will study the class of labeled structures in combination with quotient geometric crossover.

4.3 Sequences

In this subsection we recast alignment before recombination in variable-length sequences as a consequence of quotient geometric crossover. In previous work [1] we have applied geometric crossover to variable-length sequences. The distance for variable-length sequences we used there is the *edit distance LD*[8]: the minimum number of insertion, deletion and replacement of single character to transform one sequence into the other. The geometric crossover associated with this distance is the *homologous geometric crossover*: two sequences are aligned optimally before recombination. Alignment here means allowing parent sequences to be *stretched*

---

[5] Permutation matrix is a $(0, 1)$-matrix with exactly one 1 in every row and column.

[6] The proof is not trivial but omitted by space limit.

[7] The proof is not trivial but omitted by space limit.

[8] The notation LD comes from *Levenshtein distance* that is another name of edit distance.

to match better with each other. Formally stretching sequences means interleaving '-' anywhere and in any number in the sequences to create two stretched sequences of the same length that have minimal Hamming distance. For example, if we want to recombine `agcacaca` and `acacacta`, we need to align them optimally first: `agcacac-a` and `a-cacacta`. Notice that the Hamming distance between the aligned sequences is less than the Hamming distance between the non-aligned sequences.

After the optimal alignment, one does the normal crossover and produce a new *stretched* sequence. The offspring is obtained by removing '-', so by unstretching the sequence. How does quotient geometric crossover fit in here? We can define a relation $\sim$ on stretched sequences: *each stretched sequence belongs to the class of its unstretched version.* Then, we can easily check that the relation $\sim$ is an equivalence relation. Let $\langle s \rangle$ be the set of all stretched sequences of sequence s. We define the induced distance measure $d_\sim$. Let s_1, s_2 be variable-length sequences. If H is the Hamming distance for stretched sequences,

$$d_\sim(s_1, s_2) := \min_{s_1' \in \langle s_1 \rangle, s_2' \in \langle s_2 \rangle} H(s_1', s_2').$$

Then, by the definition of edit distance, $d_\sim$ is equal to LD. Hence $d_\sim$ is a metric on variable-length sequences.

THEOREM 4. *Homologous crossover is geometric under the edit distance [1].*

- *Genotypes G*: variable-length stretched sequences
- *Phenotypes P*: variable-length (unstretched) sequences
- *Equivalence relation $\sim$*: stretched sequences with the same unstretched sequence
- *Distance on genotypes d_G*: If the two stretched sequences have different length, add as many '-' as necessary at the right end of the shorter sequence to make it become equal in length to the longer sequence. Their genotypic distance is then their Hamming distance.
- *Distance on phenotypes d_P*: edit distance between sequences
- *Crossover on genotypes X_G*: traditional crossover on stretched sequences. If the two stretched sequences have different length, add as many '-' as necessary at the right end of the shorter sequence to make it become equal in length to the longer sequence.
- *Crossover of phenotypes X_P*: homologous crossover for sequences
- *Induced crossover transformation gt*: optimal alignment

Phenotypes are variable-length sequences that are directly representable. So in this case the quotient geometric crossover is not used to search a non-directly representable space (phenotypes) through an auxiliary directly representable space (genotypes). The benefit of applying the quotient geometric crossover on variable-length sequences is that the homologous crossover over sequences X_P is naturally understood as a transformation gt of the geometric crossover X_G over stretched sequences G rather than a crossover acting directly on sequences P. This is because the notion of optimal alignment is inherently defined on stretched sequences and not on simple sequences. In previous work [1] we have tested the homologous crossover on the protein motif discovery problem. In future work we want to study how the optimal alignment transformation affects the fitness landscape associated with geometric crossover with and without alignment.
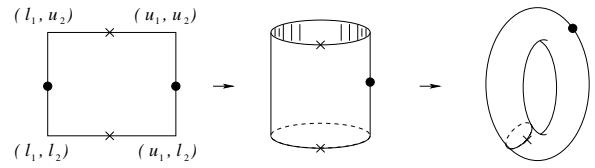
4.4 Glued Space

Figure 2: Glued space on $\mathbb{R}^2$. This can be considered as a quotient space.
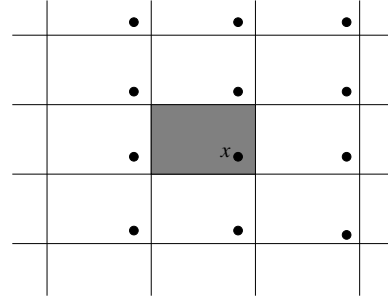

Figure 3: Equivalent class on $\mathbb{R}^2$. The shadowed rectangle represents given bounded real space X. Each rectangle has the same size as X.

In this subsection we show how to use quotient geometric crossover in connection with glued spaces for real domain.

In general, solution space of real problems has the range. Let the solution space X be $\{x \in \mathbb{R}^n \mid l_i \le x_i < u_i \text{ for each } i\}$ where $l = (l_1, l_2, \ldots, l_n)$ is a lower bound and $u = (u_1, u_2, \ldots, u_n)$ is an upper bound. If we apply geometric crossover on this bounded domain with Euclidean distance, offspring have bias toward the center of the space. One method to eliminate this bias is gluing the boundaries by identifying u_i to l_i for each i. Figure 2 shows this glued space for $\mathbb{R}^2$ case.

The interesting fact is that this glued space can be considered as a quotient space. For the definition of quotient space which gives an effect equivalent to gluing, equivalence relation on $\mathbb{R}^n$ is defined as follows:

DEFINITION 7. $x \sim y$ *if and only if for each* $i = 1, 2, \ldots, n$, *there exists* $a_i \in \mathbb{Z}$ *such that* $x_i - y_i = a_i(u_i - l_i)$.

It can be easily checked that $\sim$ is an equivalence relation. Let $\langle x \rangle$ be the equivalence class of $x \in \mathbb{R}^n$. In Figure 3, points indicated by bullets are in the same equivalent class in $\mathbb{R}^2$.

X can be considered as a quotient set $\mathbb{R}^n / \sim$ by considering $x \in X$ as $\bar{x} \in \mathbb{R}^n / \sim$. However, this gives another topology to the same set. We need to define a distance tailored to this new topology. We define a new distance induced by Euclidean distance ED. Let $x, y \in X$.

$$d_\sim(x, y) := \min_{x' \in \langle x \rangle, y' \in \langle y \rangle} ED(x', y').$$

THEOREM 5. $d_\sim$ *is a metric in* X.

PROOF. Omitted by space limit. □

The definition of $d_\sim$ is clear, but it is impossible to calculate distances considering all points in equivalence class since the number of the points is infinite. Fortunately, there is a practical way to calculate it.
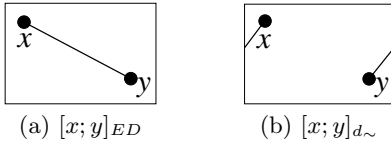
(a) $[x; y]_{ED}$ (b) $[x; y]_{d_\sim}$

Figure 4: Line segments on Euclidean space and quotient space

Let $x, y \in X$. For each i, let $T_i(y) = \{y_i, y_i + (u_i - l_i), y_i - (u_i - l_i)\}$ and $M_i(y) = \underset{m \in T_i(y)}{\text{argmin}}\{|x_i - m|\}$. $M_i(y)$ is the set because the number of maximizers can be more than one. Let $M(y) = \{(y'_1, y'_2, \ldots, y'_n) \in \mathbb{Z}^n \mid y'_i \in M_i(y) \text{ for each } i\}$.

THEOREM 6. *Let* $x, y \in X$. *If ED is Euclidean distance and* $y^* \in M(y)$, $d_\sim(x, y) = ED(x, y^*)$.

PROOF. Omitted by space limit. □

According to Theorem 6, to calculate $d_\sim(x, y)$, we need only to find y^* by choosing minimizer among three elements from $T_i(y)$ for each coordinate and get Euclidean distance between x and y^*. Now, the segment between x and y on quotient space is induced by the segment between x and y^* on $\mathbb{R}^n$.

THEOREM 7. *If ED is Euclidean distance, the line segment* $[x; y]_{d_\sim}$ *is the set*

$$\bigcup_{y' \in M(y)} \{z \in X \mid z \sim z' \text{ for some } z' \in [x; y^*]_{ED}\}.$$

PROOF. Omitted by space limit. □

In Figure 4, (a) shows the segment on Euclidean space and (b) shows the segment on quotient space. In quotient space, segments may cross the boundaries.
- *Genotypes G*: bounded continuous space
- *Phenotypes P*: glued continuous space
- *Equivalence relation* $\sim$: points between which, for each coordinate, the difference is a multiple of the range size
- *Distance on genotypes* d_G: Euclidean distance
- *Distance on phenotypes* d_P: glued distance (Pacman-world distance)
- *Crossover on genotypes* X_G: segment crossover (in the Euclidean space geometric crossover becomes the line recombination.)
- *Crossover of phenotypes* X_P: segment crossover in Pacman-world spaces (the 2D Euclidean space becomes the surface of a torus. Geometric crossover is defined on segments in this space.)
- *Induced crossover transformation gt*: points that were distant and close to opposite boundaries become close.

The original crossover X_G on genotypes is biased toward the center of the space. The benefit of quotient geometric crossover in this case is to transform the topology of the space and search the space of phenotypes that is isotropic (every point is the center, or there is no center) whose associated geometric crossover X_P has no such a bias. An unbiased space gives the same a priori opportunity to each point of the space to be searched without giving arbitrary preference to points close to the center. The search is therefore guided by the fitness values encountered only. In future work we will extend these results to more general spaces and test the unbiased crossovers on standard benchmark problems for continuous optimization.

4.5 Functions

Here we consider functional representations: any representation that encodes a function. Examples of this type of representation are genetic programming trees, finite state automata, neural networks, and so on. Among them, we will see an example for genetic programming (GP). We can define an equivalence relation: *all symbolic expressions that represent the same function.* We can also define a less strong equivalence relation: *consider as equivalent those syntactic trees that differ in the order of the operands in nodes with commutative operations.* For example, the multiplication operation '$*$' is commutative and two different trees represent the same function. The quotient geometric crossover corresponds to homologous crossover for GP trees with re-ordering of commutative subtrees to have minimum structural Hamming distance. This quotient geometric crossover is based on the less strong equivalence relation. So it is not *fully* semantical. However this quotient geometric crossover cannot be implemented efficiently because the complexity to compute the structural Hamming distance between rooted unordered trees grows exponentially with the number of nodes in the trees.
- *Genotypes G*: parse trees that is a compact (shorter than extensive form), redundant (the same function can be represented by more than one parse tree) and biased (some functions have more associated parse trees than other functions) representation of functions.
- *Phenotypes P*: computed functions. A generic function can be directly represented in an extensive form as a vector in which for every combinations of the input values there is a cell that contains the output of the function for those values. We call this vector the output vector representation of the function. Clearly this direct representation in practice is not used because it is too long.
- *Equivalence relation* $\sim$: parse trees that correspond to the same function or equivalently with the same output vector.
- *Distance on genotypes* d_G: structural Hamming distance between parse trees
- *Distance on phenotypes* d_P: (weighted) Hamming distance on output vectors
- *Crossover on genotypes* X_G: homologous crossover for parse trees
- *Crossover of phenotypes* X_P: traditional crossover on the output vectors of the functions
- *Induced crossover transformation gt*: expand/reduce/change syntactic trees before crossover without changing the underlying computed functions such as they have minimal structural Hamming distance

The benefit of applying quotient geometric crossover to parse trees is to search the space of the functions represented by the parse trees rather than the space of parse trees. This is done indirectly by manipulating parse trees. Even if in principle a function can be represented directly using its output vector representation, so making not strictly necessary to recur to an auxiliary genotypic representation and to the quotient geometric crossover to search this space, such direct representation is simply too long for any practical purpose, and a concise genotypic representation is needed.

The implementation of the phenotypic geometric crossover using the transformation gt on the genotypic crossover X_G presents a problem: it is simply not possible to compute efficiently the transformation gt because one needs to compute the smallest structural Hamming distance between all pos-

sible transformations of the syntactic trees that keep their underlying functions invariant. We could relax the problem and consider a weaker equivalence relation in which two parse trees are equivalent if exchanging subtrees of nodes with commutative operations (syntactic transformation that keeps the computed function invariant) they become equal. In this case d_P becomes the distance between rooted (partially) unordered trees. The computational cost of this distance grows exponentially with the number of commutative nodes in the syntactic trees. This could be still hard to compute and so the associated geometric crossover X_P. However there are quick approximated algorithms to compute this distance. In future work we will try this crossover.

We can apply the concept of quotient geometric crossover to other examples with functional representations such as finite state machines and neural networks in similar way.

4.6 Neutrality

The role of neutrality is little understood. Notice that *neutrality* is a synonym of redundancy. As a rule of thumb one would like to filter out redundancy as in normalization for structural problem to improve performance. However neutrality may have some beneficial aspect on variable-length representation. In fact it can be used to have a self-adaptive mutation rate at a phenotypic level: imaging you have a constant number of mutations at a genotype level. If the informative part, the one used to get the phenotype, is small as compared with the non-informative one, mutation at genotype level have a small chance to affect the phenotype. So the same mutation rate at genotype level can correspond to a smaller or equivalent mutation at a phenotype level depending on the amount of neutral code in the genotype. Since the mutation itself inserts or deletes neutral code, this combined with selection develops a self-adaptive mechanism that selects genotypes with the right amount of neutral code to be more evolvable. Neutrality is wide-spread in nature, so studying neutrality is important. Quotient geometric crossover can be used to understand how crossover and neutrality interact. In fact the induced geometricity-preserving transformation tells what *trick* to use to remove redundancy for crossover but still keep it there for mutation to obtain the self-adaptive mutation rate *trick*, so to take advantage of both genotypic and phenotypic spaces.
- *Genotypes G*: sequence with neutral code (part of the sequence that if removed would not affect the phenotype)
- *Phenotypes P*: sequence without neutral code. There is a one-to-one mapping between these sequences and the phenotypes. So this is a direct representation of the phenotypes, rather than the phenotype itself.
- *Equivalence relation* $\sim$: two sequences with neutral code are equivalent if when the neutral code is removed they become the same phenotypic sequence
- *Distance on genotypes* d_G: edit distance on sequences with neutral code
- *Distance on phenotypes* d_P: edit distance on sequences without neutral code
- *Crossover on genotypes* X_G: homologous crossover for sequences
- *Crossover of phenotypes* X_P: homologous crossover for sequences
- *Induced crossover transformation gt*: identity transformation

The benefit of the quotient geometric crossover is to show how crossover and neutral code interact. We have seen that neutrality may be beneficial in terms of adaptive muta-

tion rate. Since the induced crossover transformation is the identity transformation, this means that the same crossover that searches the genotypes space can be understood as a crossover searching the phenotype space indirectly using the genotypes. In other words, the neutral code is completely transparent to the search done by crossover and it does not affect its search or performance. So, neutral code retains the advantage of an adaptive mutation rate together with being transparent to the action of crossover.

5. CONCLUDING REMARKS

In this paper we have extended the geometric framework introducing the notion of quotient geometric crossover. This could be clearly understood using the concept of geometricity-preserving transformation. Quotient geometric crossover is a very general and versatile tool. We have given a number of interesting examples as its applications. As shown in applications, quotient geometric crossover is not only theoretically significant but also has a practical effect of making search more effective by reducing the search space or removing the inherent bias. More theoretical analysis and more detailed applications for each case are left for future study.

6. REFERENCES

[1] Geometric crossover for biological sequences. *Hiding Other Information for Double Blind Review.*

[2] Geometric crossover for multiway graph partitioning. *Hiding Other Information for Double Blind Review.*

[3] Geometric crossover for the permutation representation. *Hiding Other Information for Double Blind Review.*

[4] New topologies for genetic search space. *Hiding Other Information for Double Blind Review.*

[5] Product geometric crossover. *Hiding Other Information for Double Blind Review.*

[6] Topological crossover for the permutation representation. *Hiding Other Information for Double Blind Review.*

[7] Topological interpretation of crossover. *Hiding Other Information for Double Blind Review.*

[8] BURAGO, D., BURAGO, Y., IVANOV, S., AND BURAGO, I. D. *A Course in Metric Geometry.* American Mathematical Society, 2001.

[9] CHOI, S. S., AND MOON, B. R. Normalization in genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2003), pp. 862–873.

[10] DEZA, M. M., AND LAURENT, M. *Geometry of Cuts and Metrics.* Springer, 1997.

[11] DORNE, R., AND HAO, J. K. A new genetic local search algorithm for graph coloring. In *Proceedings of the Fifth Conference on Parallel Problem Solving from Nature* (1998), pp. 745–754.

[12] FALKENAUER, E. *Genetic Algorithms and Grouping Problems.* John Wiley & Sons, 1998.

[13] JONES, T. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, 1995.

[14] KANG, S. J., AND MOON, B. R. A hybrid genetic algorithm for multiway graph partitioning. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2000), pp. 159–166.

[15] KIM, J. P., AND MOON, B. R. A hybrid genetic search for multi-way graph partitioning based on direct partitioning. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2001), pp. 408–415.

[16] LASZEWSKI, G. Intelligent structural operators for the *k*-way graph partitioning problem. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (1991), pp. 45–52.

[17] MÜHLENBEIN, H. Parallel genetic algorithms in combinatorial optimization. In *Computer Science and*

Operations Research: New Developments in Their Interfaces (1992), pp. 441–456.

[18] OLSEN, A. L. An evolutionary algorithm to solve the joint replenishment problem using direct grouping. *Computers and Industrial Engineering 48*, 2 (2005), 223–235.

[19] PARDALOS, P. M., AND RESENDE, M. G. C., Eds. *Handbook of Applied Optimization.* Oxford University Press, 2002.

[20] SUTHERLAND, W. A. *Introduction to Metric and Topological Spaces.* Oxford University Press, 1975.

[21] VAN HOYWEGHEN, C., NAUDTS, B., AND GOLDBERG, D. E. Spin-flip symmetry and synchronization. *Evolutionary Computation 10* (2002), 317–344.