Geometric Crossover for Multiway Graph Partitioning

Yong-Hyuk Kim
Department of Mathematical Sciences and
Research Institute of Mathematics
Seoul National University
Sillim-dong, Gwanak-gu, Seoul, 151-747 Korea
yhdfly@soar.snu.ac.kr

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

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 generalization of the traditional crossover defined using the distance of the solution space. Using a distance tailored to the problem at hand, the formal definition of geometric crossover allows to design new problem-specific crossovers that embed problem-knowledge in the search. The standard encoding for multiway graph partitioning is highly redundant: each solution has a number of representations, one for each way of labeling the represented partition. Traditional crossover does not perform well on redundant encodings. We propose a new geometric crossover for graph partitioning based on a labeling-independent distance that filters the redundancy of the encoding. A correlation analysis of the fitness landscape based on this distance shows that it is well suited to graph partitioning. Our new genetic algorithm outperforms existing ones.

Categories and Subject Descriptors

G.2.3 [**Mathematics of Computing**]: DISCRETE MATHEMATICS—*Applications*

General Terms

Theory

Keywords

Geometric crossover, labeling-independent distance, multiway graph partitioning

1. INTRODUCTION

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

Geometric crossover [21] is a representation-independent operator defined over the distance of the search space. Traditional crossover can be considered as a geometric crossover based on the Hamming distance. Informally, geometric crossover requires the offspring to lie between parents. The formal definition of geometric crossover can be used to guide the design of new specific crossover operators for non-standard representations using as base for geometric crossover distances rooted on the specific representation [23]. To be effective, specific geometric crossover operators need to be matched to the problem at hand. In order to embed problem knowledge in the crossover operator, this has to be based on a distance that is meaningful for the problem at hand. In previous work [22], we have suggested a rule of thumb: the distance chosen should make the resulting fitness landscape "smooth" in some statistical sense, or in other words, closer solutions tend to have closer fitness.

Grouping problems are interesting and NP-hard [8]. In this paper, we focus on the multiway graph partitioning problem. When applying evolutionary algorithms to grouping problems, the standard solution encoding is highly redundant. This affects badly the performance of traditional crossover. In [4], it was introduced a highly effective procedure to compensate for redundancy that requires a labeling-normalization phase before the actual exchange of genetic material between parents.

Distances are becoming of increasing importance both in the analysis and design of evolutionary algorithm. The Hamming distance is a naïve distance for grouping problems because it does not keep into account the inherent redundancy of the solution encoding. In previous work [17], we have introduced a natural distance for grouping problems, the labeling-independent distance, and we proved that it satisfies the metric axioms and that can be efficiently computed using the Hungarian method [18].

In this paper, we use the labeling-independent distance as basis of geometric crossover to design a new crossover for redundant encodings. Interestingly, this distance gives rise to a crossover that requires a labeling-normalization phase before exchanging genetic material between parents in the traditional way. The new crossover can be implemented exactly and efficiently using the Hungarian method that,

unlike previous normalization heuristics, allows obtaining a perfect normalization in an efficient way. We compare the landscapes under the Hamming distance and labeling-independent distance and found that the second landscape presents characteristics that make it better matched with the corresponding geometric crossover; so we expect that the new geometric crossover will achieve better performance. We compared experimentally the traditional crossover (geometric under the Hamming distance), the new geometric crossover based on labeling-independent distance, and a third one employing a heuristic normalization that is known to be very good [15]. The new geometric crossover showed remarkable performance improvement. This corroborates the goodness of the new distance and the effectiveness of the geometric crossover as design tool. We would like to stress that the important notion of parent normalization before crossover for grouping problems naturally arises from using a distance tailored to redundant encodings together with the geometric definition of crossover.

The remainder of this paper is organized as follows. In Section 2, we introduce the multiway graph partitioning and the labeling-independent metric. In Section 3, we introduce the geometric framework. In Section 4, we present the new geometric crossover based on labeling-independent distance. In Section 5, we present a correlation analysis of the fitness landscapes associated with the Hamming and labeling-independent distances. In Section 6, we show experimental setting and results, and make conclusions in Section 7.

2. GRAPH PARTITIONING AND LABELING-INDEPENDENT DISTANCE

2.1 Multiway Graph Partitioning

Graph partitioning is an important problem that arises in various fields of computer science, such as sparse matrix factorization, VLSI circuit placement, network partitioning, and so on. Good partitioning of a system not only significantly reduces the complexity involved in the design process, but can also improve the timing performance as well as its reliability [9].

Let $G = (V, E)$ be an unweighted undirected graph, where V is the set of vertices and E is the set of edges. *K-way partition* is a partitioning of the vertex set V into K disjoint subsets $\{P_1, P_2, \ldots, P_K\}$. A K-way partition is said to be *balanced* if the difference of cardinalities between the largest and the smallest subsets is at most one. In this paper, we consider only balanced partition. The *cut size* of a partition is defined to be the number of edges whose endpoints are in different subsets of the partition. The *K-way partitioning problem* is the problem of finding K-way balanced partition with minimum cut size.

Since the K-way partitioning problem is NP-hard [8], attempts to solve partitioning problems have focused on finding heuristics which yield approximate solutions in polynomial time. Among such methods, the Kernighan-Lin algorithm [14] and the Fiduccia-Mattheyses algorithm (FM) [7] are representative. They are local search heuristics for 2-way partitioning. There have been a number of algorithms for K-way partitioning [5, 6, 13, 25]. There have been also several methods using genetic algorithms [12, 15, 19].

2.2 Labeling-Independent Distance

The standard representation of a solution for K-way graph partitioning is a vector x of size $|V|$ such as $x_i = j \Rightarrow v_i \in P_j$. Since the specific mapping of indices to partitions does not change how the graph is partitioned, each solution has $K!$ representations. For this encoding, the Hamming distance between two solutions is unnatural because it depends on the specific mapping between indices and partitions that is completely arbitrary. We proposed a distance measure, the labeling-independent distance, that eliminates this dependency completely [17].

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.

DEFINITION 1 (LABELING-INDEPENDENT DISTANCE). *Given two K-ary encodings $\mathfrak{a}, \mathfrak{b} \in U = \{1, 2, \ldots, K\}^{|V|}$ (fixed-length vectors on a K-ary alphabet) and a metric $\mathfrak{d}$ in U, we define the* labeling-independent distance *LI associated to $\mathfrak{d}$ as follows:*

$$LI(\mathfrak{a}, \mathfrak{b}) := \min_{\sigma \in \Sigma_K} \mathfrak{d}(\mathfrak{a}, \mathfrak{b}_\sigma)$$

where Σ_K is the set of all permutations of length K and $\mathfrak{b}_\sigma$ is a permuted encoding of $\mathfrak{b}$ by a permutation σ, i.e., the i^{th} element b_i of $\mathfrak{b}$ is transformed into $\sigma(b_i)$.

THEOREM 1. *Labeling-independent distance LI is a pseudo-metric in U.*

PROOF: The proof is given in [17].

Given an element $\mathfrak{a} \in U$, since $\mathfrak{d}$ 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 following proposition holds.

PROPOSITION 1. *The relation $\sim$ is an equivalence relation.*

PROOF: The proof is given in [17].

THEOREM 2. *Suppose that Q is the quotient set of U by relation $\sim$ (i.e., $Q = U/\sim$). Then, (Q, LI) is a metric space, i.e., the labeling-independent distance LI is a metric in Q.*

PROOF: The proof is given in [17].

So, U is the set of all *labeled partitions* and $\mathfrak{d}$ is a metric on this set. Q is the set of all *unlabeled partitions* associated to U and LI is the corresponding metric on Q associated to $\mathfrak{d}$.

2.3 Efficient Normalization and Labeling-Independent Distance

We say that $\mathfrak{b}_{\sigma^*}$ is normalized to $\mathfrak{a}$ when $LI(\mathfrak{a}, \mathfrak{b}_{\sigma^*}) = \mathfrak{d}(\mathfrak{a}, \mathfrak{b}_{\sigma^*})$ and we call σ^* a normalizing relabeling.

In previous work, we have shown that, in the special case of $\mathfrak{d}$ being the Hamming distance H, the problem of computing LI can be formulated as the optimal assignment problem and it can be solved efficiently by the Hungarian method.

The problem of finding a normalizing relabeling σ^* is equivalent to computing LI. Let $M = (m_{ij})$ be the $K \times K$ assignment weight matrix between two chromosomes X and Y. Each element m_{ij} means $\sum_{k=1}^{|V|} I(X_k = i, Y_k \neq j)$ or $\sum_{k=1}^{|V|} I(X_k \neq i, Y_k = j)$. The problem of computing LI is exactly the problem of finding an assignment (permutation) with minimum summation.

THEOREM 3. *If the metric $\mathfrak{d}$ is the Hamming distance H, then*

$$LI(X,Y) = \min_{\sigma \in \Sigma_K} \sum_{i=1}^{K} \sum_{k=1}^{|V|} I(X_k = i, Y_k \neq \sigma(i)).$$

PROOF: The proof is given in [17].

3. GEOMETRIC FRAMEWORK

In this section, we report the essential concepts behind a theoretical framework of recent introduction that allows to analyze and design new crossover operators for any solution representation tailored to the problem at hand.

3.1 Geometric Preliminaries

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.

In a metric space (S, d), a *closed ball* is the set of the form $B(x;\delta) = \{y \in S \mid d(x,y) \leq \delta\}$ where $x \in S$ and δ is a positive real number called the radius of the ball. A *line segment* (or closed interval) is the set of the form $[x,y]_d = \{z \in S \mid d(x,z) + d(z,y) = d(x,y)\}$ where $x, y \in S$ are called extremes of the segment. Metric 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 over S. Notice that d is arbitrary and need not have any particular connection or affinity with the problem at hand.

3.2 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 2 (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 3 (GEOMETRIC CROSSOVER). *A binary operator GX is a geometric crossover under the metric d if all offspring are in a segment between its parents: $\forall x, y$: $Im[GX(x,y)] \subseteq [x,y]_d$*

A number of general properties for geometric crossover and mutation have been derived in [21].

FACT 1. *The traditional crossover for K-ary vectors with n crossover points is geometric under the Hamming distance [21].*

3.3 Geometric Crossover Landscape

The notion of fitness landscape is useful if the search operators employed are connected or matched with the landscape: the greater the connection the more landscape properties mirror search properties. The conventional way to look at the landscape is to see it as a *function of the search operator employed* (Jones [11]). Whereas mutation is intuitively associated with the neighbourhood structure of the search space, crossover stretches the notion of landscape further leading to search spaces defined over complicated topological structures.

Geometric crossover and mutation are based on the distance associated with the search space. This approach is the dual of Jones' approach: *we see the genetic operators as functions of the search space*. So, mutation and crossover share the same neighbourhood structure. This greatly simplifies the relationship between crossover and fitness landscape and allows to give a simple rule of thumb that tells what makes a fitness landscape well-searchable by crossover.

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. To avoid this one needs to exploit problem knowledge in the search by choosing a distance that makes sense for the problem at hand.

What is a good distance for the problem at hand? That can be turned into: 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 neighbourhood search meta-heuristics [24].

4. GEOMETRIC CROSSOVER BASED ON LABELING-INDEPENDENT METRIC

We design a new geometric crossover based on the labeling-independent metric. We call it n-point LI-GX.

DEFINITION 4 (n-POINT LI-GX). *Normalize the second parent to the first under the Hamming distance. Do the nor-*

mal n-point crossover using the first parent and the normalized second parent.

THEOREM 4. *The n-point LI-GX is geometric under the labeling-independent metric.*

PROOF: Let p_1 and p_2 be parents and c be offspring after n-point LI-GX. It is enough to show that $LI(p_1, p_2) = LI(p_1, c) + LI(c, p_2)$. Since LI is a metric, by triangular inequality, it is trivial that $LI(p_1, p_2) \leq LI(p_1, c) + LI(c, p_2)$. Now, we will show that $LI(p_1, p_2) \geq LI(p_1, c) + LI(c, p_2)$. Let σ' be $argmin_{\sigma \in \Sigma_K} H(p_1, (p_2)_\sigma)$, where $(p_2)_\sigma$ is a permuted encoding of p_2 by σ, i.e., the i^{th} element p_{2i} of p_2 is transformed into $\sigma(p_{2i})$. Then, we let p_2' be $(p_2)_{\sigma'}$.

$$\begin{aligned} LI(p_1, p_2) &= H(p_1, p_2') \\ &= H(p_1, c) + H(c, p_2') \quad (\because \text{Fact 1}) \\ &\geq LI(p_1, c) + LI(c, p_2') \\ &= LI(p_1, c) + LI(c, p_2) \quad \blacksquare \end{aligned}$$

To implement n-point LI-GX, we use the Hungarian method to find a maximum matching of the labels of the second parent to the first parent and then we apply a traditional n-point crossover. The time complexity of n-point LI-GX is $O(|V| + K^3)$.

5. FITNESS LANDSCAPE ANALYSIS

5.1 Smoothness

5.1.1 Measuring Smoothness

To measure the smoothness of a fitness landscape, we plot its autocorrelation function (or *correlogram*): a graph that on the x-axis has distance between solutions and on the y-axis has the correlation between their fitness. The autocorrelation function for fitness landscapes was introduced by Weinberger [26].

To have a meaningful comparison between correlation functions of fitness landscapes based on different search space structures with different distance distributions, we normalize the x-axis dividing it by the average distance $E(d)$ between any two points of the search space. Let $r_d(\delta)$ be the normalized correlogram of a fitness landscape based on a metric d. Then $r_d(1)$ is the average distance correlation and is a measure of global smoothness of the landscape. If $r_d(1)$ is positive, we say that the landscape is globally smooth. If it is negative, the landscape is said to be globally discontinuous. If the normalized correlogram of a landscape is always above the normalized correlogram of a second landscape, then the former is smother than the latter.

5.1.2 Correlogram of Local-optimum Space

Since our evolutionary algorithm uses crossover to search the local-optimum space, rather than the whole space, we have to consider the correlogram of such a restricted space. However, it is not easy to plot it exactly. Given a distance value δ, it is not easy to find pairs of local optima such that their distances are equal to the value δ. If we get it by forming distance classes from a pool of randomly sampled local optima, since the local-optimum space is too big, the range of correlation becomes limited. However, an approximated correlogram can be obtained under the following experimental framework.

Get randomly generated M local optima s_i's;
for each $p \in (0, 100]$ {
  Get corresponding M local optima t_i's after
    "random p% perturbation + local optimization";
  for each pair of M pairs (s_i, t_i)'s,
    Compute the Hamming distance (h_i)
      and labeling-independent distance (l_i);
  Compute the correlation coefficient (c)
    between the fitness of M pairs;
  Plot a point (average of h_i's/$E(H)$, c);
  Plot a point (average of l_i's/$E(LI)$, c);
}

$*$ In our experiments, we set M to be 3,000.

First, we get randomly sampled M local optima. For each perturbation rate p, we perturb each local optimum by randomly perturbing p percent of the bits and apply the local optimizer to the perturbed solution to obtain a local optimum. Then, we compute the Hamming distance and labeling-independent distance between the new local optimum and the previous one. After computing the correlation coefficient between the fitness of M pairs, we plot the relation between the distances and the correlation coefficient.

We have some remarks. The average of h_i's and the average of l_i's will increase as the perturbation rate p increases. c's are the same for two plots.

5.1.3 Distance Distributions

Let us consider the search space for 32-way partitions on a graph with 500 vertices (G500.2.5). We can compute the expected value $E(H)$ in all-partition space approximately. In all-partition space, the probability that the i^{th} positions of two solutions are the same is 1/32. Hence, $E(H)$ is approximately $500 \times (1 - 1/32) = 484.375$. This fact hints that "most pairs of points are nearly of a maximum distance." The moral is that the problem space may be far from our expectation. The following table shows the average and the maximum distance for each space. It shows that the problem space becomes much narrower in the labeling-independent space (LI-space).

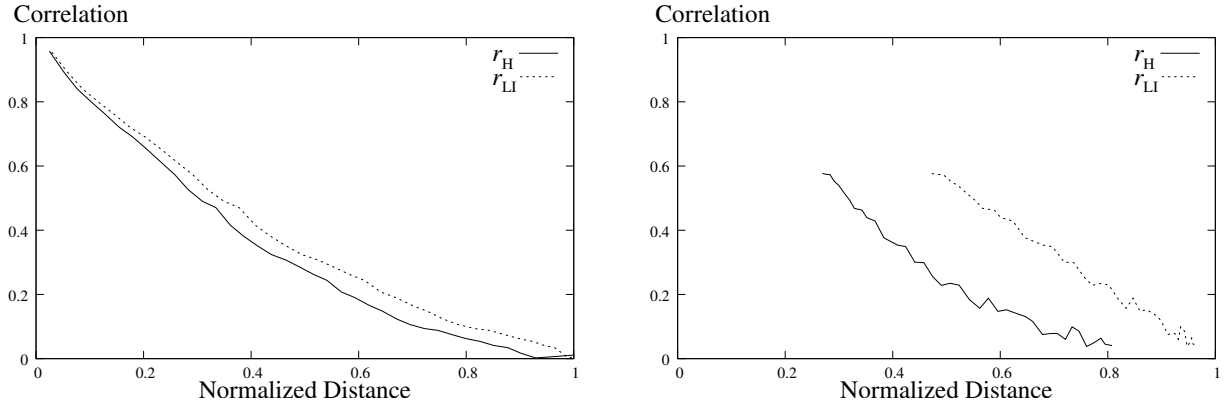
Space	$E(d)^\dagger$	max(d)
(all-partition, H)	484.364	500
(local-optimum, H)	484.369	$\leq$ max(whole, H)
(all-partition, LI)	429.010	$\leq 484^\ddagger$
(local-optimum, LI)	274.301	$\leq$ max(whole, LI)

† The values of $E(d)$ were empirically computed from a pool of 500 randomly sampled solutions.

‡ This value is the simple upper bound. Let a and b be 32-way partitions of 500 vertices. The size of the largest subset for a or b is $\lceil 500 \times 1/32 \rceil = 16$. Let i and j be the indices of the largest subsets of a and b, respectively. Then, $LI(a, b) \leq H(a, b_\sigma)$, where σ is a permutation such that $\sigma(j) = i$. Since $H(a, b_\sigma) \leq 500 - 16 = 484$, $LI(a, b) \leq 484$.

5.1.4 Comparison

Figure 1 shows normalized correlograms for all-partition space and local-optimum space. The x-axis means normalized distance between solutions and the y-axis is the correlation coefficient between their fitness. First, we com-



(a) All-partition space (b) Local-optimum space

Figure 1: Normalized correlogram (G500.2.5)

pare between H-space and LI-space based on all partitions. The normalized correlograms look similar (see Figure 1a). The LI-correlogram is always above the H-correlogram, but not of much, so we can say that LI-landscape is slightly smoother than H-landscape. For both correlograms, points that are at a distance smaller than average have increasingly strong correlation; this makes both landscapes globally smooth and so suitable for geometric crossover.

The reason that we expect the geometric crossover based on LI-landscape to perform better than the one based on H-landscape is only partially due to the stronger correlation, but mostly due to the fact that the sizes of the spaces that are actually searched by geometric crossover are different: H-crossover searches the space of all labeled-partitions and LI-crossover searches the much smaller space of all unlabeled-partitions.

Next, we compare H-space and LI-space on local optima. It is interesting to note that the average distance of local-optimum LI-space is much smaller than the average space of all-partition LI-space, whereas for the H-space there is almost no difference. This happens because by local optimization, vertices of some subgraphs tend to belong to the same partition. In local-optimum space, the effect of normalization (associated to LI-space) is very large. The reason must be topological: all the peaks are somehow remapped and clustered by the normalization. Hence, it is natural that the average distance of local-optimum LI-space is much smaller than that of all-partition LI-space.

Since the average distance between local optima for LI is so much smaller than that for H, the respective normalized correlograms are strongly affected (see Figure 1b): the LI-landscape is really much smoother than the H-landscape (LI-correlogram is much above H-correlogram). So, from the rule-of-thumb that smoother landscape has better geometric crossover performance, we expect that, on the local-optimum space, the geometric crossover based on labeling-independent distance (LI-GX) performs much better than the geometric crossover based on the Hamming distance (H-GX).

5.2 Global Convexity

Given a set of local optima, Boese *et al.* [2] plotted, for each local optimum, the relationship between the cost and the average distance from all the other local optima. They performed experiments for the graph bisection and the traveling salesman problem, and showed that both problems have strong positive correlations. This fact hints that the best local optimum is located near the center of the local-optimum space and, roughly speaking, the local-optimum space is globally convex. In this section, we repeat their experiments for multiway graph partitioning with different distances.

The solution space for the experiment is selected as follows. First, we choose five hundreds of random solutions and obtain the corresponding set of local optima by locally optimizing them[1]. Figure 2 shows the plotting results for 32-way partitioning of an instance (G500.2.5). The result with the labeling-independent metric was consistent with Boese *et al.*'s results with strong cost-distance correlation (correlation coefficient was 0.79). On the other hand, the result with Hamming distance showed little correlation (correlation coefficient was -0.11).

In summary, there are three reasons we expect LI-GX to perform better than H-GX: (i) global convexity (by cost-distance correlation [2]), (ii) the smaller size of problem space (from average distance), and (iii) smoothness (by autocorrelation [26]). Properties (ii) and (iii) are more obvious in local-optimum space.

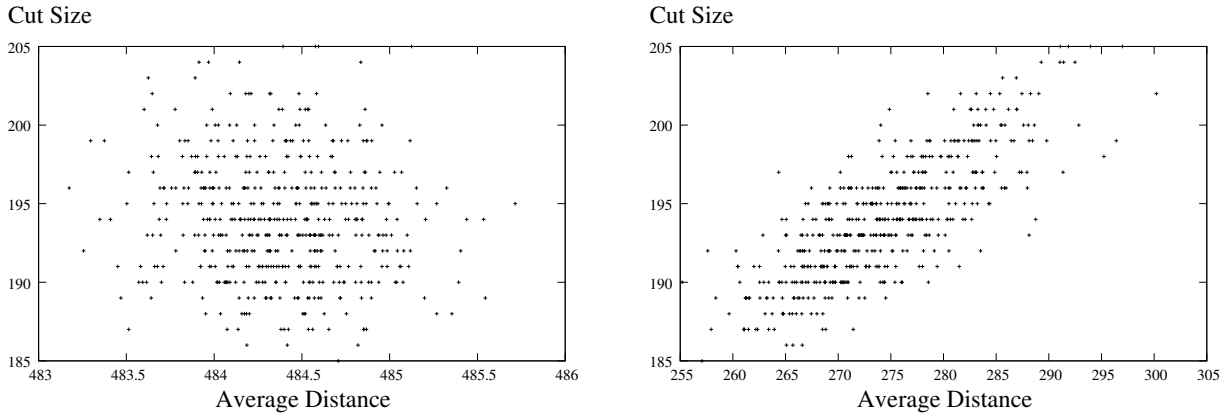
6. EXPERIMENTS

6.1 Genetic Framework

We used the general structure of hybrid steady-state genetic algorithms. In the following, we describe the framework of genetic algorithm used in our experiments. Under this framework, we will change only the crossover operator.

- *Encoding*: We use a K-ary string for each chromosome to represent a K-way partition. For example, if vertex v_i belongs to partition P_j, the value of the i^{th} gene is j.
- *Initialization*: We randomly create p chromosomes. Each chromosome satisfies a balance criterion. We set the population size p to be 50.

[1]We used the local optimization algorithm described in Section 6.1.



(a) Hamming distance

(b) Labeling-independent distance

Figure 2: The relationship between cost and distance (G500.2.5)

- *Selection*: We use the roulette-wheel-based proportional selection scheme. The probability that the best chromosome is chosen was set to four times higher than the probability that the worst chromosome is chosen.
- *Mutation*: After crossover, chromosomes are usually not balanced. We start at a random position on the chromosome and adjust the gene values to the right until the balance is satisfied. This makes some mutation effect, so we do not add any specific mutation.
- *Local optimization*: Sanchis [25] extended the FM algorithm for K-way partitioning. The algorithm considers all possible moves of each vertex from its home set to any of the others. He showed that this direct multiway partitioning approach obtained better solutions compared to the recursive approach for random networks. As local optimization engine in our genetic algorithm, we use its variation proposed in [15]. Its time complexity is $O(K|E|)$.
- *Replacement*: If it is superior to the closest parent, the offspring replaces the closest parent, and if not, the other parent is replaced if the offspring is better. Otherwise the worst in the population is replaced.
- *Stopping criterion*: For stopping, we use the number of consecutive fails to replace one of the parents. We set the number to be 50.

6.2 Test Environment

Before showing the experimental results, we first introduce the benchmarks used in this experiment and test environment. We tested on a total of eight graphs which consist of two groups of graphs. They are composed of eight graphs with 500 vertices from [10] (four random graphs G*.* and four random geometric graphs[2] U*.*). The two classes were used in a number of other graph-partitioning studies [1, 3, 16, 20]. More detailed description of them is given in [16].

We conducted tests on 32-way and 128-way partitioning. A *C* language program was used on a Pentium III 1GHz computer with Linux operating system. It was compiled using *gcc* compiler.

[2]Geometric graphs are one of the classes of random graphs that are believed to be most similar to actual VLSI-circuit and computer-network graphs in the sense that they tend to have local clusters.

6.3 Results

Under the same genetic framework described in Section 6.1, we compare the geometric crossover based on the Hamming distance (5pt H-GX), the geometric crossover based on labeling-independent distance (5pt LI-GX), and a crossover with previous normalization technique that is known to be very good [15] (GEFM).

Let LI be the labeling-independent metric and let p_1, p_2 be parents, c be offspring after crossover, and o be offspring after mutation (balance adjustment). Each value in Table 1 and Table 2 means the average value of $(LI(p_1, c) + LI(c, p_2) - LI(p_1, p_2))/LI(p_1, p_2)$ and $(LI(p_1, o) + LI(o, p_2) - LI(p_1, p_2))/LI(p_1, p_2)$. To measure the degree of distortion from the shortest path, we make this experiments. By Theorem 4, it is trivial that the values of "After xover" in 5pt LI-GX are zero. The values of "After mutation" in 5pt LI-GX were also the smallest.

Table 3 and Table 4 show the partitioning results. The statistics are from 100 independent runs. In 32-way partitioning, 5pt LI-GX was better than 5pt H-GX on the best and the average. LI-GX was better than GEFM on the average and also wins on the best (4 victories, 2 draws, and 2 defeats). Similar numbers of generations were reported. In 128-way partitioning, LI-GX showed more improved performance. LI-GX was better than 5pt H-GX and GEFM both on the average and on the best.

The local optimization used is much more computationally expensive than maximum matching using the Hungarian method. For small number K, normalization by the Hungarian method affects computational time little. In 32-way partitioning, similar computational time was required. In 128-way partitioning, our algorithm took about 1.3 times more than the others.

From a geometrical viewpoint, it is not surprising that 5pt LI-GX and GEFM performed similarly, even if 5pt LI-GX was better, because after all GEFM is almost geometric under the distance 5pt LI-GX is based upon. Since, from the landscape analysis we know that this is a very meaningful distance for the problem at hand, this just corroborates the rule-of-thumb that given a good distance, one get a good geometric crossover, other details being of secondary importance (exact probability distribution of the search operator). The argument that geometricity is what really counts is re-

Table 1: The Degree of Distortion on 32-way Partitioning

Graph	5pt H-GX		GEFM[15]		5pt LI-GX	
	After xover	After mutation	After xover	After mutation	After xover	After mutation
G500.2.5	0.56	0.69	0.01	0.14	0.00	0.12
G500.05	0.42	0.53	0.01	0.14	0.00	0.13
G500.10	0.42	0.53	0.01	0.13	0.00	0.11
G500.20	0.39	0.49	0.01	0.12	0.00	0.12
U500.05	1.07	1.25	0.01	0.15	0.00	0.14
U500.10	1.49	1.72	0.01	0.15	0.00	0.15
U500.20	1.90	2.18	0.01	0.15	0.00	0.14
U500.40	0.96	1.12	0.01	0.15	0.00	0.14

Average over 100 runs.

Table 2: The Degree of Distortion on 128-way Partitioning

Graph	5pt H-GX		GEFM[15]		5pt LI-GX	
	After xover	After mutation	After xover	After mutation	After xover	After mutation
G500.2.5	0.75	0.99	0.03	0.25	0.00	0.21
G500.05	0.69	0.91	0.03	0.24	0.00	0.22
G500.10	0.62	0.83	0.03	0.24	0.00	0.21
G500.20	0.55	0.75	0.03	0.23	0.00	0.21
U500.05	1.14	1.45	0.03	0.25	0.00	0.22
U500.10	0.83	1.07	0.03	0.25	0.00	0.22
U500.20	0.55	0.74	0.03	0.23	0.00	0.21
U500.40	0.42	0.60	0.03	0.22	0.00	0.19

Average over 100 runs.

inforced by noting (see Table 2) that GEFM was a less good approximation of geometric crossover for 128-way partitioning (0.03 non-geometric) and this was mirrored in Table 4 with a bigger difference in absolute values between the averages of GEFM and 5pt LI-GX. So, when GEFM was less geometric, this was immediately reflected in a loss of performance.

7. CONCLUSIONS

In this paper, we have shown that the important notion of normalization before recombination, that is very effective on problems with redundant encodings, can be naturally cast in geometric terms using a distance that filters the redundancy of the encoding together with the formal definition of geometric crossover.

This geometric point of view on normalization has allowed us to study its effect on the fitness landscape and explain, within the geometric framework, why normalization before recombination for redundant encodings is a good idea.

The landscape analysis also provided evidence for the fact that the labeling-independent distance is more suitable for the solution space of multiway graph partitioning problem than the Hamming distance.

We designed a geometric crossover based on the labeling-independent distance and showed its performance by experiments. It outperformed existing genetic algorithms. We expect that our geometric crossover will be working well on other grouping problems.

8. ACKNOWLEDGMENTS

The authors are grateful to Riccardo Poli for his valuable comments. The authors would also like to thank Jong-Pil Kim for providing the source code of GEFM. This work was supported by the Brain Korea 21 Project in 2006. This was also partly supported by grant No. (R01-2003-000-10879-0) from the Basic Research Program of the Korea Science and Engineering Foundation. This work was also supported in part by a grant from the Research Institute of Mathematics, Seoul National University. The ICT at Seoul National University provided research facilities for this study.

9. REFERENCES

[1] Battiti, R., and Bertossi, A. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Transactions on Computers 48*, 4 (1999), 361–385.

[2] Boese, K. D., Kahng, A. B., and Muddu, S. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters 15* (1994), 101–113.

[3] Bui, T. N., and Moon, B. R. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers 45*, 7 (1996), 841–855.

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

[5] Cong, J., and Lim, S. K. Multiway partitioning with pairwise movement. In *Proceedings of the International Conference on Computer-Aided Design* (1998), pp. 512–516.

[6] Cong, J., Lim, S. K., and Wu, C. Performance driven multi-level and multiway partitioning with retiming. In *Proceedings of the ACM/IEEE-CAS/EDAC Design Automation Conference* (2000), pp. 274–279.

[7] Fiduccia, C., and Mattheyses, R. A linear time heuristics for improving network partitions. In *Proceedings of the 19th ACM/IEEE Design Automation Conference* (1982), pp. 175–181.

[8] Garey, M., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of*

Table 3: The Results of 32-way Partitioning

Graph	5pt H-GX			GEFM[15]			5pt LI-GX		
	Best	Ave†	Gen(CPU‡)	Best	Ave†	Gen(CPU‡)	Best	Ave†	Gen(CPU‡)
G500.2.5	182	185.18	1091(173.33)	178	181.83	1503(159.39)	178	**181.77**	1529(180.30)
G500.05	626	637.25	1424(330.64)	624	631.27	1974(359.25)	624	**630.07**	2367(334.80)
G500.10	1576	1587.23	1984(713.07)	1575	1582.42	2250(672.64)	1573	**1581.40**	2422(571.50)
G500.20	4040	4049.44	2247(1502.26)	4038	4045.41	2425(1402.75)	4034	**4044.89**	2522(1245.96)
U500.05	112	120.65	1327(319.04)	113	117.06	1592(314.26)	112	**116.75**	1599(331.95)
U500.10	534	542.75	1163(449.76)	529	537.81	1468(464.70)	531	**537.04**	1494(483.50)
U500.20	1837	1846.30	1123(814.94)	1829	1841.68	1327(715.80)	1832	**1841.02**	1353(747.04)
U500.40	5363	5389.93	1043(1399.31)	5355	**5380.19**	1353(1410.09)	5353	5380.30	1374(1398.19)

† Average over 100 runs.
‡ CPU seconds on Pentium III 1GHz.

Table 4: The Results of 128-way Partitioning

Graph	5pt H-GX			GEFM[15]			5pt LI-GX		
	Best	Ave†	Gen(CPU‡)	Best	Ave†	Gen(CPU‡)	Best	Ave†	Gen(CPU‡)
G500.2.5	316	320.14	844(535.13)	318	320.08	853(513.96)	310	**314.08**	950(759.72)
G500.05	850	853.09	869(688.03)	847	852.38	936(684.05)	839	**843.61**	1020(947.68)
G500.10	1904	1907.78	932(1300.94)	1901	1906.71	996(1224.89)	1894	**1898.03**	1168(1316.21)
G500.20	4568	4571.93	965(2333.56)	4565	4570.48	1028(2187.80)	4560	**4566.40**	1116(2261.33)
U500.05	697	704.06	935(910.83)	698	703.30	971(830.69)	695	**702.90**	978(1227.62)
U500.10	1679	1684.13	913(1302.62)	1676	1683.92	925(1223.74)	1676	**1683.47**	921(1618.17)
U500.20	3836	3841.45	890(1437.07)	3836	3841.57	895(1378.42)	3836	**3841.44**	874(1790.69)
U500.40	8066	**8068.54**	853(1971.41)	8066	8068.71	850(1847.12)	8065	8068.77	831(2250.17)

† Average over 100 runs.
‡ CPU seconds on Pentium III 1GHz.

NP-Completeness. Freeman, San Francisco, 1979.

[9] Hu, T. C., and Kuh, E. S. *VLSI Curcuit Layout Theory and Design*. IEEE Press, New York, 1985.

[10] Johnson, D. S., Aragon, C., McGeoch, L., and Schevon, C. Optimization by simulated annealing: An experimental evaluation, Part 1, graph partitioning. *Operations Research 37* (1989), 865–892.

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

[12] 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.

[13] Karypis, G., and Kumar, V. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing 48*, 1 (1998), 96–129.

[14] Kernighan, B., and Lin, S. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal 49* (Feb. 1970), 291–307.

[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] Kim, Y. H., and Moon, B. R. Lock-gain based graph partitioning. *Journal of Heuristics 10*, 1 (January 2004), 37–57.

[17] Kim, Y. H., and Moon, B. R. New topologies for genetic search space. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2005), pp. 1393–1399.

[18] Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart. 2* (1955), 83–97.

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

[20] Merz, P., and Freisleben, B. Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning. *Evolutionary Computation 8*, 1 (2000), 61–91.

[21] Moraglio, A., and Poli, R. Topological interpretation of crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2004), pp. 1377–1388.

[22] Moraglio, A., and Poli, R. Geometric crossover for the permutation representation. *Technical Report CSM-429, University of Essex* (2005).

[23] Moraglio, A., and Poli, R. Topological crossover for the permutation representation. In *GECCO 2005 Workshop on Theory of Representations* (2005).

[24] Pardalos, P. M., and Resende, M. G. C., Eds. *Handbook of Applied Optimization*. Oxford University Press, 2002.

[25] Sanchis, L. A. Multiple-way network partitioning. *IEEE Transactions on Computers 38*, 1 (1989), 62–81.

[26] Weinberger, E. D. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics 63* (1990), 325–336.