On the sampling property of real-parameter crossover

Shin Ando[1], Shigenobu Kobayashi[1]

[1] Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, 6259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa, Japan
{ando, kobayashi}@fe.dis.titech.ac.jp

Abstract. In solving real-world optimization problems using evolutionary algorithms (EAs), researchers have recently developed a number of real-parameter genetic algorithms (GAs). In these studies, the main focus of the research is the recombination operator, which use probability distributions calculated from the parent solutions. Such operator is intrinsically a non-parametric modeler and estimator, which the success of the real-parameter GA depends on. In this paper, we propose an implementation of a graph structure among the population to select such non-parametric kernels more efficiently during the search process. By recording the successful crossover as an edge between the individuals, clusters within the evolved population were observed. Additionally, a simple experiment exploiting such clusters showed efficient result in multimodal optimization.

1 Introduction

Recently, there have been increasing interests for real-parameter genetic algorithms, which uses real-number vector to encode the GA individuals. Many past studies devoted to developing powerful and efficient crossover operators. Different types of real-parameter crossover are described by [1]. A family of crossovers, Parent Centric Crossover (PCX), Unimodal Normal Distribution Crossover (UNDX), and Simplex Crossover (SPX) are extensively studied and well-applied to many practical problems.

Intrinsically, a real-parameter crossover operator work as a non-parametric probabilistic model estimator and sampler. The parent-individuals in the crossover form a local probability distribution. A child-individual is a sample from such local distribution. The entire distribution of the child-individual, if the parent-individuals are selected probabilistically, is the linear combination of such kernels, i.e., probability distribution formed by all of the possible crossovers. The conceptual illustration of non-parametric probability distribution composed of real-parameter crossover is shown in Fig. 1. The crossover using two parents (represented by circles) creates a kernel (represented by ovals) which includes themselves within its region.

As the figure implicitly suggests, the kernels can be represented as graphs. In this paper, we propose to represent the kernels with a graph-structure to test and utilize the kernels it represents. Section 2 describes the concept of non-parametric model estimation. Section 3 describes the usage of graph structure represent non-parametric

models. Section 4 describes the experiments to analyze how the graph is formed and to exploit the graph structure for efficient parameter optimization.

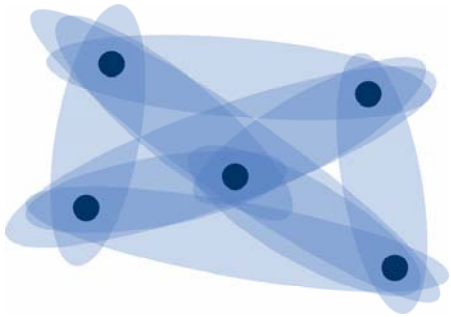


Fig. 1. A conceptual view of non-parametric probabilistic model latent in GA population and the real-parameter crossover: The circles represent the individuals, where as the ellipsoids represent the probabilistic distribution of the offspring in a possible crossover of two selected parents. The linear combination of all possible ellipsoid makes up a non-parametric probabilistic distribution.

2 Non-parametric Modeling in Real-parameter Crossover

The properties of the kernels are defined by specification of the crossovers, e.g., BLX-α [2] uses a hyper-cube, UNDX [3] uses a Gaussian kernel, and SPX [4] uses a hyper-pyramid or an expanded simplex of parent-individuals. Above crossovers center the kernel at the geometric centroid of the parents (mean-centric), [1] proposed a parent-centric crossover (PCX) which places a Gaussian kernel centered at one of the parents.

Under specific conditions, the combined probability distributions of these crossovers satisfy one of the guidance for design of real-parameter GA [5], i.e., "Genetic operator should conserve the statistical property of the population, mainly the mean and variance-covariance matrix".

The UNDX crossover is designed as follows: μ parents $\mathbf{x}_i$ (i=1 to μ, $\mathbf{x}\in R^n$) are randomly chosen from population P. Taking μ-1 parents, geometric center $\mathbf{g}$ is computed, as well as $\mathbf{d}_i=\mathbf{x}_i. - \mathbf{g}$ (i=1 to μ-1). Using μ-th parent, $\mathbf{d}_\mu=\mathbf{x}_\mu. - \mathbf{g}$ is computed. The orthogonal component of $\mathbf{d}_\mu$ to all $\mathbf{d}_i$ (i=1 to μ-1) has length D. The ortho-normal basis of the complementary subspace to space spanned by $\mathbf{d}_i$ (i=1 to μ-1) is $\mathbf{e}_i$ (i=μ to n). The offspring $\mathbf{y}$ is given by (1).

$$\mathbf{y} = \mathbf{g} + \sum_{i=1}^{\mu-1} w_i \mathbf{d}_i + \sum_{i=\mu}^{n} v_i D \mathbf{e}_i \qquad (\mathbf{1})$$

The w_i and v_i are normal distribution $N(0, \sigma^2_\zeta)$, $N(0,\sigma^2_\eta)$ respectively. The second term in (1) is known as the primary search component, and the third term is called the secondary search component. For parameter σ^2_ζ and σ^2_η, suggested value are given in [3]. This crossover creates offspring within a Gaussian kernel formed by the parent individuals.

As the GA population evolves, parents are replaced by the offspring and the kernels are updated by the new individuals. In Minimal Generation Gap [6] replacement scheme, used with UNDX, SPX, PCX [1, 3, 4], large number of offspring are generated from the same parents and only two individuals (two of the parents) are under selective pressure at a time.

One iteration of the MGG model is described as follows.

1. Select μ parents from the Population *P*.
2. Generate λ offspring from parents using genetic operators.
3. Choose two elimination candidate from the parents.
4. Create a subpopulation F of offspring and elimination candidate.
5. Replace elimination candidate with best and roulette wheel selection from F.

In UNDX+MGG, the suggested value of μ is 3 to 6, while λ is 100 [3]. In the following sections, we consider UNDX and MGG as our default genetic operator and replacement model.

With each step of the MGG, the kernel is tested and updated/maintained. Using MGG model with real-parameter crossover, the behavior of the GA can be interpreted as testing and updating the kernels represented by the population. Our intuition is that by explicitly handling the kernels, it is possible to perform more efficient search with crossover.

From another view point, sampling from the kernels is equivalent to rough analysis of the local fitness landscape. We define the distance on fitness landscape Δ_{ij} between individuals *i*,*j*, as the integral of objective function over the region covered by the kernel κ_{ij}.

$$\Delta_{ij} = \int_{\kappa_{ij}} f(\mathbf{x})\, d\mathbf{x} \tag{2}$$

Parents with small Δ should form better kernel or cluster as it is more probable they belong in the same basin rather than with that of larger Δ. The Δ should present better index for closeness between individual than Euclidean distance *d*. The conceptual illustration of the landscape distance is shown in Fig. 2.

If the kernel includes many promising individuals, it indicates that the parents' landscape distance is small and should be clustered together. Meanwhile, if the kernel fails to produce better individuals, it is probable that the parents are distant on fitness landscape and has small correlation.

In multimodal objective function optimization, the result of finding a better kernel is similar to that of clustering. For successful multimodal optimization, the population should form clusters around local optima. As the evolution proceeds, the discrepancy between the clusters widens and effective crossover/kernel between clusters becomes sparse.

Simplistically, the kernel which generate improved individual should be emphasized, while the kernel which failed should be thrown out. But since the GA is a

population based search, the parents are maintained as well as the kernel, e.g. MGG. The next section describes a supplementary data structure to represent and exploit the kernels.

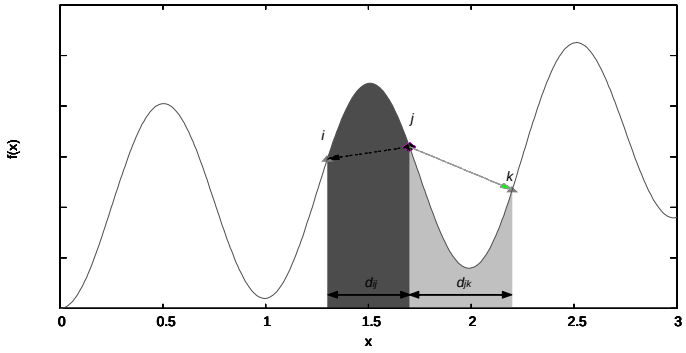


Fig. 2. The distance on 1-dimentional fitness landscape. The curve represents the one dimensional schwefel function. The triangular and rectangular points (*i*, *j*, *k*) indicate the individuals. The landscape distance Δ_{ij} indicated by the dashed arrow is defined by the area of dark-gray filled region. The landscape distance Δ_{jk} indicated by the solid arrow is defined by the area of gray filled region. In this case, $\Delta_{ij} > \Delta_{jk}$ where as in Euclidean distance, $d_{ij} < d_{jk}$.

2 Representation of the non-parametric kernel with graph

One straight-forward representation of the kernel would be a graph structure, since the kernels correspond to a set of individuals (parents). We used an undirected graph G(V, E) where vertices V_i (i=1 to P) represent the individual.

The kernel can be represented as a set of vertex connected by the edges. The Fig.3 shows a basic procedure for building graph from successful crossovers.

2. 1 Kernel Evaluation

In this section, we will define a specific kernel evaluation index, which uses concept similar to landscape distance.

The landscape distance Δ can be used to compare the relations between individuals. But since it is nearly an integral along the path between individual, it is affected by the ill-scaled problems, i.e., if one or more variables has different scaling, that variable will be over/under-treated. Solution to this is using the average. The mean fitness function value M over a kernel κ_{ij}, is defined by $M=\Delta_{ij}/\int\kappa_{ij}d\mathbf{x}$. M is not affected by ill-scaling, since it does not differentiate far and near individuals. By comparing M to a standardized fitness value α, we measure the quality of a kernel, whether it covers a promising region.

GA(MGG) Procedure	Graph Procedure
1. Initialize population Initialize $\mathbf{x}_i$ (i = 1 to P) All individuals have specific index *id*	Initialize graph: *G* 1. prepare an empty graph 2. add vertices V_i (i = 1 to P) to the graph
2. Select μ parent individuals at random	
3. Select one elimination candidate ε from the parents randomly	
4. Generate λ offspring from using cross-over	Evaluate the kernel: Compare the offspring fitness with standard fitness value α.
5. Create subpopulation F=λ+ε	
6. Replace elimination candidate with best in F	Add/Remove vertices/edges 1. Erase replaced vertex from the graph 2. Connect the ends of outgoing edge from erased vertex with new edges. 3. If the kernel evaluation is a satisfactory, insert edges between new individual and the remaining parent 4. Otherwise, Remove the edge (if exists) between parent-individuals
7. Go back to 2.	

Fig. 3. GA and Graph building procedure

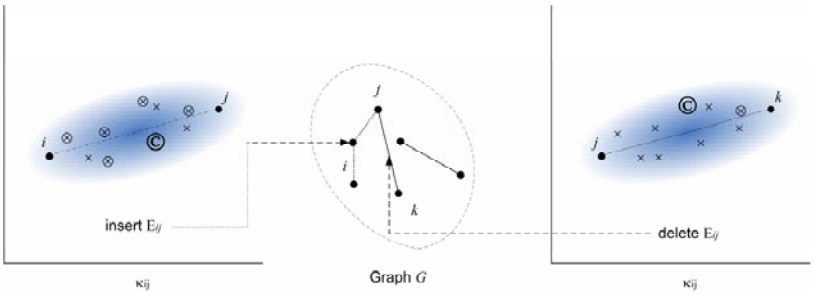


Fig. 4. The edge insertion/deletion by kernel evaluation. On the left, the kernel sampled many above standard fitness individual, which inserts an edge between the parents and the offspring. On the right, the kernel samples only few fit individuals which triggers the deletion of edge between parents.

In our work, we implemented the approximated, discrete version of the procedure described above. This is important for efficiency and avoiding fitness scaling issues.

We estimate the M by sampling from the kernel. This is efficient since this process is part of the MGG. The β percentile of sampled offspring's fitness values f_i, (i=1 to λ) is compared to the standardized fitness α. For α, we used the median of the individuals' fitness in entire population P. The graph building and kernel evaluation is illustrated in Fig. 4.

The described kernel evaluation and graph-building proceed alongside the genetic operation of the GA. With the iterated genetic operation, the updated kernel is evaluated in higher resolution since the offspring is more likely to be placed between the parents.

2.2 Experiment

The graph building process can be operated completely independent from the GA. We conduct an experiment to analyze how the conventional GA, namely UNDX+MGG, build a graph with the proposed method.

The objective function we used is a two-peak sphere function F_1 shown in (3).

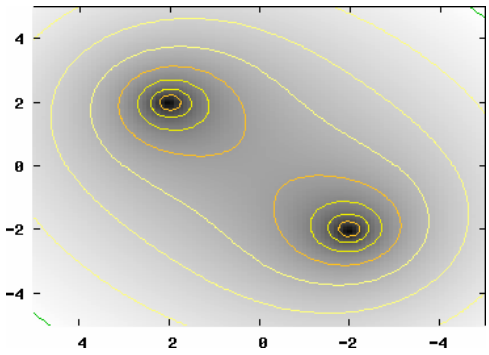
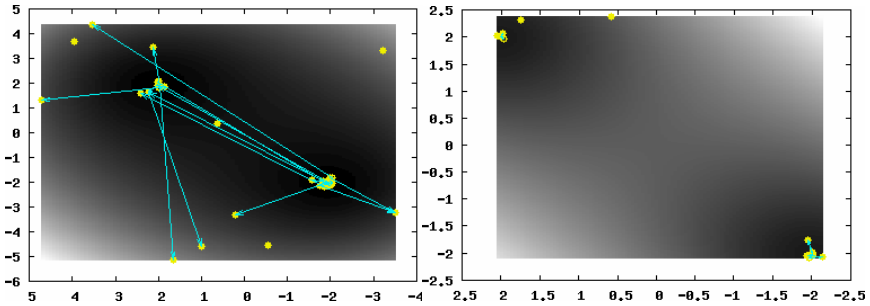


Fig. 5. The fitness landscape of F_1 mapped onto R^2. The contour is shown by lines and the gradient. The darker region indicates better fitness.

$$F_1(\mathbf{x}) = \left(\sum_{i=1}^{n} (x_i + 2)^2 \right) \left(\sum_{i=1}^{n} (x_i - 2)^2 \right) \quad (3)$$

The problem dimension n=10. The parameters for MGG are μ=3, λ=50. The parameters for UNDX are σ_ζ=1 and σ_η=0.11, using the suggested value in [3]. The population size P=40, and the GA terminated at 10^4 function evaluations. The percentile parameter β=10 was used.



(a)after 70 evaluation (b)after 200 evaluation

Fig. 6. The graph structure after 70 and 200 generation.

Results

Fig. 6 shows the edges between the individuals on fitness landscape after 7,000 and 20,000 function evaluations. There are 20 and 11 edges in the respective graphs. After 20,000 evaluation, edges that connect different optima are removed. The

3 Exploiting the graph structure with clustering

This section describes the method to utilize the graph structure in the GA search. A simple approach is to uses the graph as an efficient alternative for clustering. The clustering method is known to improve the performance of GA in multimodal optimization. The graph connects individuals whose mutual landscape distance is small. If the Since the graph records successful kernels, UNDX crossover should benefit from selecting parents from individuals connected by the edges.

In [1], Deb stated, while comparing the mean-centric crossover to the parent-centric crossover, that it may be demanding to assume that region near the geometric center of the parent-individuals contain good solutions, compared to the region near the parent-individuals itself, especially when the parents are sparse in the search domain. To supplement, the case where mean-centric kernel can be effective is when the landscape is smooth enough for the population to form its contour. Additionally, it can generally cover larger region of the search domain than parent-centric approach. In our implementation, the graph structure can identify the kernels that are likely to be successful while preserving the merits of the mean-centric crossover.

The procedure for selecting parents from the graph G(V, E) are given as follows

1. Select parent V_i (i=1 to P) randomly from G
2. Add all vertices connected to V_i by less than two edges from to set Q
3. If Q has m(<μ) elements, add μ-m vertices randomly selected from G.
4. Otherwise select μ individuals randomly from Q

If the first selected parent does not have enough neighboring individuals, randomly selected parents are used in the crossover. To differentiate the *m* adjacent and distant μ-*m* parents, offspring **y** are generated from following formulae:

$$\mathbf{y} = \mathbf{g} + \sum_{i=1}^{m-1} w_i \mathbf{d}_i + \sum_{i=m}^{\mu-1} u_i \mathbf{d}_i' + \sum_{i=\mu}^{n} v_i D \mathbf{e}_i \quad \left(\mathbf{g} = \sum_{i=1}^{m} \mathbf{x}_i, \mathbf{d}_i = \mathbf{x}_i - \mathbf{g} \right) \qquad (4)$$

$\mathbf{d}_i'$ is the vector $\mathbf{d}_i$ in subspace complementary to space spanned by $\mathbf{d}_i$ (i=1 to m). u_i is an Gaussian distribution N(0, 0.1). By initializing the graph as complete graph, the selection and the crossover operator's behaviors are equivalent to that of UNDX+MGG.

3.1 Experiment

We evaluate the performance of the proposed method on two test functions. One is the Himmelblau function F_2 (5), and the other is the checker-stripe function F_3 (6). Both function have multiple global optima, thus the objective is to find all of them.

$$F_2(x, y) = -\left(x^2 + y - 11\right)^2 - \left(x + y^2 - 7\right)^2 \qquad (5)$$

$$F_3 = 10n - \sum_{i=1}^{n} \mathrm{pos}\left(\sin\left(x^2 \pi\right)\right) \cdot \delta\left(x - \mathrm{int}(x) - 0.1\,\mathrm{int}(x)\right) \qquad (6)$$

$$\mathrm{pos}(t) = \begin{cases} t \mid t > 0 \\ 0 \mid t < 0 \end{cases}, \delta(t) = \begin{cases} 1 \mid t > 0 \\ 0 \mid t < 0 \end{cases}$$

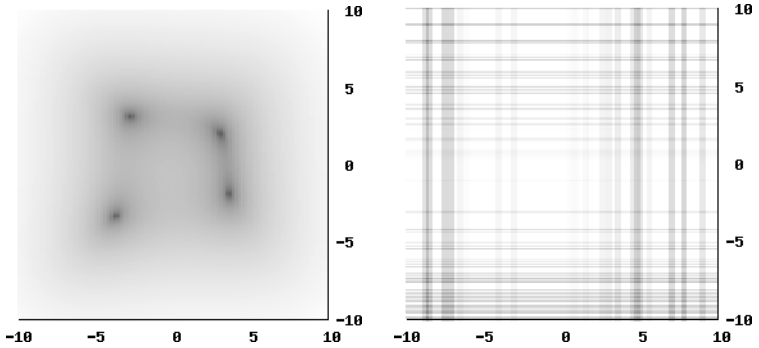
F_3 has many local optima, and is a deceptive function, i.e., the basins of the worse local optima cover larger region of the domain. The landscape of F_3 and F_2 is shown in Fig 7.

The experiment was performed with following settings: μ=3, λ=40, σ_ζ=1, σ_η=0.11, P=100, β=10, and termination at 10^6 function evaluations. 20 runs were performed for each experiment.

3.2 Results

In function F_2, The proposed approach reached all of the optima in all runs, while UNDX+MGG converged to 3 or less optima in all runs, depending on the initial population.

In function F_3, the proposed approach reached the global optima in all runs. UNDX+MGG did not reach the global optima when the population converged on one of the dimensions before it reached the optimal. The average percentage of the optima reached by UNDX+MGG was 9.8. As shown in Table 3, the average number of function evaluation to reach the global optima was generally smaller with the proposed method.



(a) Himmelblau (b) Checker-stripe

Fig. 7. The fitness landscape of two dimensional F_2 and F_3 is shown as gradient. The region with higher fitness is colored with darker color.

Table 1. The number of runs where the optima of F_2 were reached with UNDX+MGG and the proposed method.

Optima	Coordinates	F_2 value	UNDX+MGG	Proposed method
1	(3.584,-1.848)	1.0 e-20	18	20
2	(3.000, 2.000)	1.0 e-20	20	20
3	(-2.805, 3.131)	1.0 e-20	15	20
4	(-3.779,-3.283)	1.0 e-20	19	20

Table 2. The average percentage of the optima of F_3 reached with UNDX+MGG and the proposed method.

Coordinates	UNDX+MGG	Proposed method
(±9.92474,..., ±9.92474)	9.8	100

Table 3. The average number of evaluation required to reach optimal fitness function for F_3.

UNDX+MGG	Proposed method
168,802± 2,375	108,072+4,323

4 Conclusions

We proposed a novel approach to utilize crossover and estimate the correlation of the individuals. Using the graph-structure, were able to form the clusters adaptively and with very few additional computational cost. The result shows that the proposed

method can search multiple optima in one run. This shows that at the beginning of the search the clusters of the graph are overlapping, and as the evolution progress, the population is divided into clusters which respectively search a nearby basin.

The proposed method uses autonomous and adaptive approach to clustering. Compared to other clustering methods such as k-means, this method is less vulnerable in ill-scaled problems because it does not use distance in determining the class.

5 Future Works

We plan to perform more specific analyses of the proposed method. One is the quantitative analysis of the clusters in the graph.

We also plan to compare the proposed method to other multimodal search genetic algorithm on common benchmarks. Additionally, initializing a complete graph is very costly since many graph modification procedure require O(E), where E is the number of edges. We plan to address this problem by applying clustering algorithm to the initial population.

References

1. Kalyanmoy Deb, Ashish Anand and Dhiraj Joshi. Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. *Evolutionary Computation*, Volume 10, Issue 4 (2002).
2. Eshelman, L. J. and Schaffer, J. D.: Real-Coded Genetic Algorithms and Interval-Schemata, in Foundations of Genetic Algorithms 2, pp. 187–202 (1993).
3. Ono, I. and Kobayashi, S.: A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, in Proceedings of 7th International Conference on Genetic Algorithms, pp. 246–253 (1997).
4. Higuchi, T., Tsutsui, S., and Yamamura, M. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In Schoenauer, M. et al., editors, Parallel Problem Solving from Nature (PPSN-VI), pages 365–374, Springer, Berlin, Germany. (2000).
5. Kita, H. and Yamamura, M. A functional specialization hypothesis for designing genetic algorithms. Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, pages 579–584, IEEE Press, Piscataway, New Jersey. (1999).
6. Satoh, H., Yamamura,M., and Kobayashi, S. Minimal generation gap model for GAs considering both exploration and exploitation. In Yamakawa, T. and Matsumoto, G., editors, Proceedings of the IIZUKA: Methodologies for the Conception, Design, and Application of Intelligent Systems, pages 494–497, World Scientific, Singapore. (1996).