

A Crossover Operator Using Independent Component Analysis for Real-Coded Genetic Algorithms

Masato Takahashi

Interdisciplinary Graduate School of
Science and Engineering

Tokyo Institute of Technology

4259 Nagatsuta, Midori, Yokohama 226-8502, JAPAN

jin@ktl.dis.titech.ac.jp

Hajime Kita

Faculty of University Evaluation and Research
National Institution for Academic Degrees

3-29-1 Otsuka, Bunkyo, Tokyo 112-0012, JAPAN

kita@niad.ac.jp

Abstract- For Real-coded Genetic Algorithms, there have been proposed many crossover operators. The blend crossover (BLX- α) proposed by Eshelman and Schaffer shows good search ability for separable fitness functions. However, because of its component-wise operation, the BLX- α faces difficulties in optimization of non-separable fitness functions. The present paper proposes a novel crossover operator that combines the BLX- α with the Independent Component Analysis (ICA). That is, by applying the ICA to the population, the coordinate system of the search space is transformed so as to increase separability of the fitness function, and then the BLX- α is applied. Computer simulation shows good search ability of the proposed method for non-separable fitness functions.

1 Introduction

Real-Coded Genetic Algorithms (RCGAs) that utilize floating point representation attract attention as methods for global optimization in continuous search spaces as well as the Evolution Strategies[14], and there has been proposed many crossover operators for the RCGA[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. The RCGAs with sophisticated crossover operators have shown excellent optimization abilities[5, 6, 7, 8, 9, 10, 11]. Further, as for the design guidelines of the crossover operators for RCGAs, Kita et al. has proposed those of ‘preservation of statistics’ and ‘diversity of offsprings’ based on theoretical analysis[13, 9].

Crossover operators for the RCGA and recombination operators for the ES[14] can be categorized into two: Operators in the first category combine parental information in a component-wise manner in generating offspring. Among these operators, the blend crossover[6, 7] and the simulate binary crossover[5] shows good search ability. However, because of their component-wise operation, these crossover operators face difficulty in optimization of non-separable functions[12, 7, 8].

In the second category, for mixture of the parental information, linear combination is used. The unimodal

normal distribution crossover[8], its multi-parental extension[9], and the simplex crossover[10] shows good search ability. Because of their linear combination operation, these operators cope with non-separability well. On the other hand, they loses fundamental concept of crossover that crossover operation combines the building blocks composing parents in generating offspring.

This paper proposes a novel approach to cope with non-separability using the component-wise crossover. That is, through statistical analysis of the population by means of the principal component analysis (PCA) and/or the independent component analysis (ICA)[16, 17, 18], the coordinate system of the search space is transformed so as to eliminate the non-separability of the fitness function. Then the component-wise crossover operator is adopted.

2 Crossover Operators for RCGA

In this section, as representatives of the aforesaid two categories of the crossover, two crossover operators showing good search ability are introduced briefly.

2.1 Blend Crossover

In the BLX- α [6], offspring are generated as follows:

- (1) Choose two parents $\mathbf{x}^1, \mathbf{x}^2$ randomly from the population.
- (2) A value of each element x_i^c of the offspring vector \mathbf{x}^c is chosen randomly from the interval $[X_i^1, X_i^2]$ following the uniform distribution, where

$$\begin{aligned} X_i^1 &= \min(x_i^1, x_i^2) - \alpha d_i \\ X_i^2 &= \max(x_i^1, x_i^2) + \alpha d_i \\ d_i &= |x_i^1 - x_i^2| \end{aligned} \quad (1)$$

and x_i^1 and x_i^2 are the i -th elements of \mathbf{x}^1 and \mathbf{x}^2 , respectively, and α a positive parameter.

For the value of α , Eshelman and Schaffer have used 0.5[6]. With our calculation, the value of α that preserves the variance of the parental population is 0.366.

Preliminary experiments shows that the latter value achieves better performance.

Because of the component-wise nature of the BLX- α , mutual dependency among the variables are not well considered. Hence, the BLX- α faces difficulty in optimization of non-separable fitness functions[8]. Considering this problem, Eshelman et al. have proposed a method that restrict generation of offspring around the diagonal region connecting the parents[7]. However, it introduces more parameters to be adjusted, and therefore, tuning of the parameters by trial-and-error gets difficult.

2.2 Unimodal Normal Distribution Crossover

Ono et al. have proposed the unimodal normal distribution crossover (UNDX)[8], and Kita et al. has extended the original UNDX to utilize the information of multiple parents[9]. It is called the UNDX- m . The UNDX-1 corresponds to the original UNDX. The procedure of generating offspring in the UNDX- m is as follows:

- (1) Choose $m + 1$ parents $\mathbf{x}^1, \dots, \mathbf{x}^{m+1}$ randomly from the population.
- (2) Let the center of mass of these parents be $\mathbf{p} = \frac{1}{m+1} \sum_i \mathbf{x}^i$, and let the difference vectors between \mathbf{x}^i and \mathbf{p} be $\mathbf{d}^i = \mathbf{x}^i - \mathbf{p}$.
- (3) Choose another parent \mathbf{x}^{m+2} from the population randomly.
- (4) Let D be the length of elements of $\mathbf{d}^{m+2} = \mathbf{x}^{m+2} - \mathbf{p}$ orthogonal to $\mathbf{d}^1, \dots, \mathbf{d}^m$.
- (5) Let $\mathbf{e}^1, \dots, \mathbf{e}^{n-m}$ be the orthonormal basis of the subspace orthogonal to $\mathbf{d}^1, \dots, \mathbf{d}^m$.
- (6) Generate offspring \mathbf{x}^c as follows:

$$\mathbf{x}^c = \mathbf{p} + \sum_{i=1}^m w_i \mathbf{d}^i + \sum_{i=1}^{n-m} v_i D \mathbf{e}^i \quad (2)$$

where w_i and v_i are normal random numbers following $N(0, \sigma_\xi^2)$ and $N(0, \sigma_\eta^2)$, respectively, and σ_ξ^2 and σ_η^2 are parameters.

For the parameters of the UNDX- m , Kita et al. has proposed

$$\sigma_\xi = 1/\sqrt{m}, \quad \sigma_\eta = 0.35/\sqrt{n-m}$$

which are obtained from the condition of preserving the variance-covariance matrix of the parental population and empirical values suggested for the UNDX[8]. As for the value of the parameter m they shows that $4 \sim 5$ gives good results empirically.

Linear combination of the parents (the first and the second terms of the RHS of Eq. 2.2 plays dominant role

in the UNDX- m , and therefore, the UNDX- m well preserves the mutual relationship among variables. However, since it uses relatively small m , diversity of offspring generated by the UNDX- m may not sufficient. Further, the basic concept of crossover that crossover operation combines the building blocks of parents is lost in the crossover operators of the linear combination type.

3 Proposed Method

3.1 Concept of the Proposed Method

In the previous section, we have discussed that the BLX- α has an advantage in generating offspring having diversity on one hand, and has a disadvantage in consideration of mutual relationship among variables. This paper pursues a method of finding the mutual relationship through statistical analysis of the population, transforming the coordinate system of the search space so as to reduce the identified relationship, and then applying the BLX- α .

Due to the selection operation in GAs, the population distribute in a region of small fitness values (minimization problems are considered). If there exists mutual relationship among the variables, the distribution is not parallel to the axes. Applying multivariate analysis to the population, we extract such structure, and transform the coordinate system so as to reduce the mutual relationship among variables. As for statistical methods, we employ the principal component analysis (PCA), a well established method for extracting the structure of multivariate data, and the independent component analysis (ICA). The ICA is a method for such purpose studied intensively these years[16, 17, 18]. The reason of adopting the ICA is discussed in Appendix.

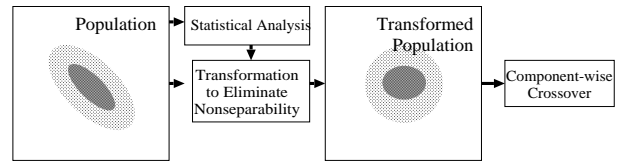


Figure 1: Concept of the Proposed Method

3.2 Principal Component Analysis

Suppose there exists m sets of data of n variables, which is represented by an $n \times m$ matrix $X = \{x_{ij}\}$ ($i = 1, \dots, n$, $j = 1, \dots, m$). The variance-covariance matrix $S = \{s_{ij}\}$ of X is given by

$$s_{ij} = \frac{1}{m-1} \sum_{k=1}^m x_{ik} x_{jk} \quad (3)$$

where data are normalized in advance so that the mean of each variables becomes zero. Transforming the data

X with a $n \times n$ matrix A to $Y = AX$, the variance-covariance matrix $S' = \{s'_{ij}\}$, $s'_{ij} = \frac{1}{m-1} \sum_{k=1}^m y_{ik}y_{jk}$ of $Y = \{y_{ij}\}$ is given by

$$S' = ASA^T \quad (4)$$

The principal component analysis (PCA) is to find a matrix A that makes S' the identical matrix. That is, to find a matrix that eliminates the correlation among variables of Y .

Since S is a real symmetric matrix, there exists an orthogonal matrix P and a diagonal matrix Λ such that $P^T S P = \Lambda$. A diagonal matrix Λ whose diagonal elements are eigen values $\lambda_1, \dots, \lambda_n$ of S , and a matrix P whose columns are corresponding eigen vectors are a pair of such matrices. Hence, we obtain

$$S' = (AP)\Lambda(AP)^T. \quad (5)$$

To make S' the identical matrix, a transformation matrix A can be obtained as follows:

$$A = \text{diag}(1/\sqrt{\lambda_1}, 1/\sqrt{\lambda_2}, \dots, 1/\sqrt{\lambda_n})P^T \quad (6)$$

Assuming all the eigen values are positive, we can obtain the inverse transformation A^{-1} also as follows:

$$A^{-1} = P \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_n}) \quad (7)$$

3.3 Independent Component Analysis

As shown in the previous section, the PCA is a method of eliminating the *correlation* among the data. The independent component analysis (ICA) is a method that makes the p.d.f. of variables mutually *independent*. Let the transformation matrix and the transformed data be C and $Z = CX$, respectively.

In the fixed point algorithm proposed by Hyvärinen et al.[17], first, the data are made uncorrelated by the PCA:

$$Y = AX \quad (8)$$

Then, by obtaining a unit weight vector \mathbf{b} that minimizes or maximizes the kurtosis of $\mathbf{b}Y$, and with n such vectors $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)^T$, we obtain

$$Z = BY = BAX \quad (9)$$

That is, $C = BA$. Since B is obtained as an orthogonal matrix, we obtain the inverse transformation as

$$\begin{aligned} C^{-1} &= A^{-1}B^{-1} = A^{-1}B^T \\ &= P \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})B^T \end{aligned} \quad (10)$$

In our study, the method of obtaining $\mathbf{b}_1, \dots, \mathbf{b}_n$ one after another takes long computation time, and therefore we employed the method of calculating the basis vectors in parallel[18].

An example of the PCA and the ICA is shown in Fig. 2. The dimension of the space and the number of the data are $n = 2$ and $m = 2000$, respectively. The data X is generated in a parallelogram uniformly. With the PCA, an uncorrelated distribution Y is obtained. However, it is still not parallel to the axes. With the ICA, a distribution parallel to the axes is obtained. Thus, transformation to an independent distribution is achieved.

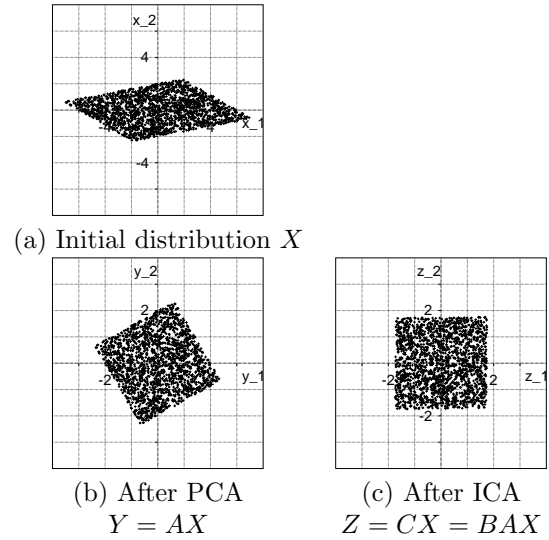


Figure 2: Transform of the population with the PCA and the ICA.

3.4 Crossover Combined with the PCA/ICA

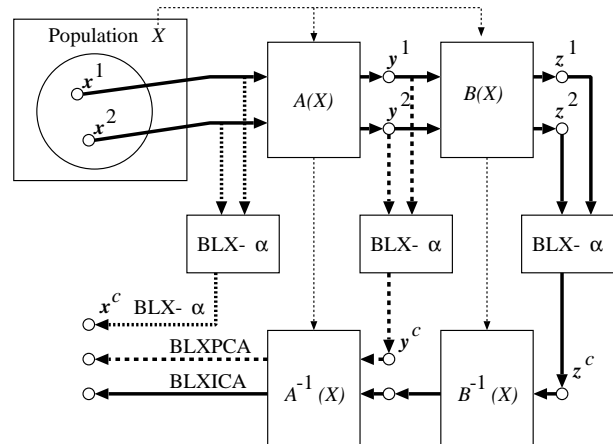


Figure 3: Procedures of BLX- α , BLXPCA and BLXICA

Let X be an $n \times m$ matrix representing the population of m individuals in the n dimensional search space. Using X , calculate transformation matrices $A(X)$ and $C(X) = B(X)A(X)$ of the PCA and the ICA, respectively.

Then, choose parents \mathbf{x}^1 and \mathbf{x}^2 randomly from the population, and apply the ICA transformation:

$$\mathbf{z}^1 = C(X)\mathbf{x}^1 \quad (11)$$

$$\mathbf{z}^2 = C(X)\mathbf{x}^2 \quad (12)$$

Next, with the BLX- α applied to the transformed parents, we obtain a child in the transformed space

$$\mathbf{z}^c = \text{BLX-}\alpha(\mathbf{z}^1, \mathbf{z}^2) \quad (13)$$

Finally, applying the inverse transformation to it, we obtain a child in the original search space:

$$\mathbf{x}^c = C^{-1}(X)\mathbf{z}^c = A^{-1}(X)B^{-1}(X)\mathbf{z}^c \quad (14)$$

We call this procedure ‘the BLXICA.’ A similar operation can be obtained using the PCA instead of the ICA. We call it ‘the BLXPCA.’ Figure 3 illustrates the process of the original BLX- α , the BLXPCA and the BLXICA.

The following is a realization of a genetic algorithm using the BLXICA (or BLXPCA). For generation alternation, the minimal generation gap (MGG) model, a variation of the steady state GA, proposed by the Satoh et al.[15] is used. No mutation is used because the BLX- α shows basically good search ability without mutation, and simple-minded mutation without self-adaptation having difficulty in the RCGA:

1. Generate an initial population X .
2. Obtain matrices $A(X)$ and $B(X)$ applying the PCA and ICA to the population.
3. Choose parents \mathbf{x}^1 and \mathbf{x}^2 randomly and apply the ICA (or PCA) transformation to them.
4. Generate c children of the parents with the BLX- α in the transformed space, where c is a parameter of the MGG.
5. Apply the inverse transformation to the children.
6. Evaluate the fitness values of the children, and replace the parents with the individual having the best fitness and an individual chosen by roulette selection among the union of the parents and children.
7. If the generation reaches the prescribed value, terminate the algorithm. Otherwise, go to step 2.

It should be also noted that the proposed approach is an intermediate one between the conventional RCGAs and the optimization technique called ‘the estimation of distribution algorithms (ESA)’ in continuous domain[19, 20, 21]. In the latter approach, first, the probability density function (p.d.f.) of good solutions is estimated through statistical analysis of the population, and second, novel solutions are sampled using the

estimated p.d.f. The approach proposed in this paper shares a common idea of applying statistical analysis to the population with EDA on one hand, and on the other hand, it shares crossover operation for sampling novel solutions with the conventional RCGAs.

4 Numerical Experiments

So as to examine the search ability of the BLXPCA and BLXICA, we have carried out numerical experiments. The purpose of the experiments is to capture the characteristics of the search of the proposed method. Comprehensive test to confirm the search ability of the proposed method for broad range of the test functions is a subject of future study.

4.1 Test Functions

For experiments, three non-separable functions are chosen to examine the effects of the proposed method. The dimension of the space n is 20 for all the functions. All the functions are to be minimized.

Rotated Rastrigin Function : it is a function obtained by the rotation transformation by $\pi/6$ for all the pairs of the axes to the Rastrigin function:

$$f(x) = 10n + \sum_{i=1}^n \{x_i^2 - 10\cos(2\pi x_i)\}$$

It is highly multi-modal. Non-separability is introduced by the rotation transformation. Optimal solution is the origin of the coordinate system. Initial population is generated following the uniform distribution on $[-5.12, 5.12]^{20}$.

Rosenbrock Function : it is given by

$$f(x) = \sum_{i=2}^n \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\} \\ -2.048 < x_i < 2.048$$

While this function is unimodal, it is strongly non-separable. The optimal solution is $(1, \dots, 1)$, which is located on a curved and steep-walled valley.

Ill-scaled Rosenbrock Function : it is given by

$$f(x) = \sum_{i=2}^n \{100(x_1 - (ix_i)^2)^2 + (ix_i - 1)^2\} \\ -2.048/i < x_i < 2.048/i$$

This function is a poorly scaled version of the previous Rosenbrock function.

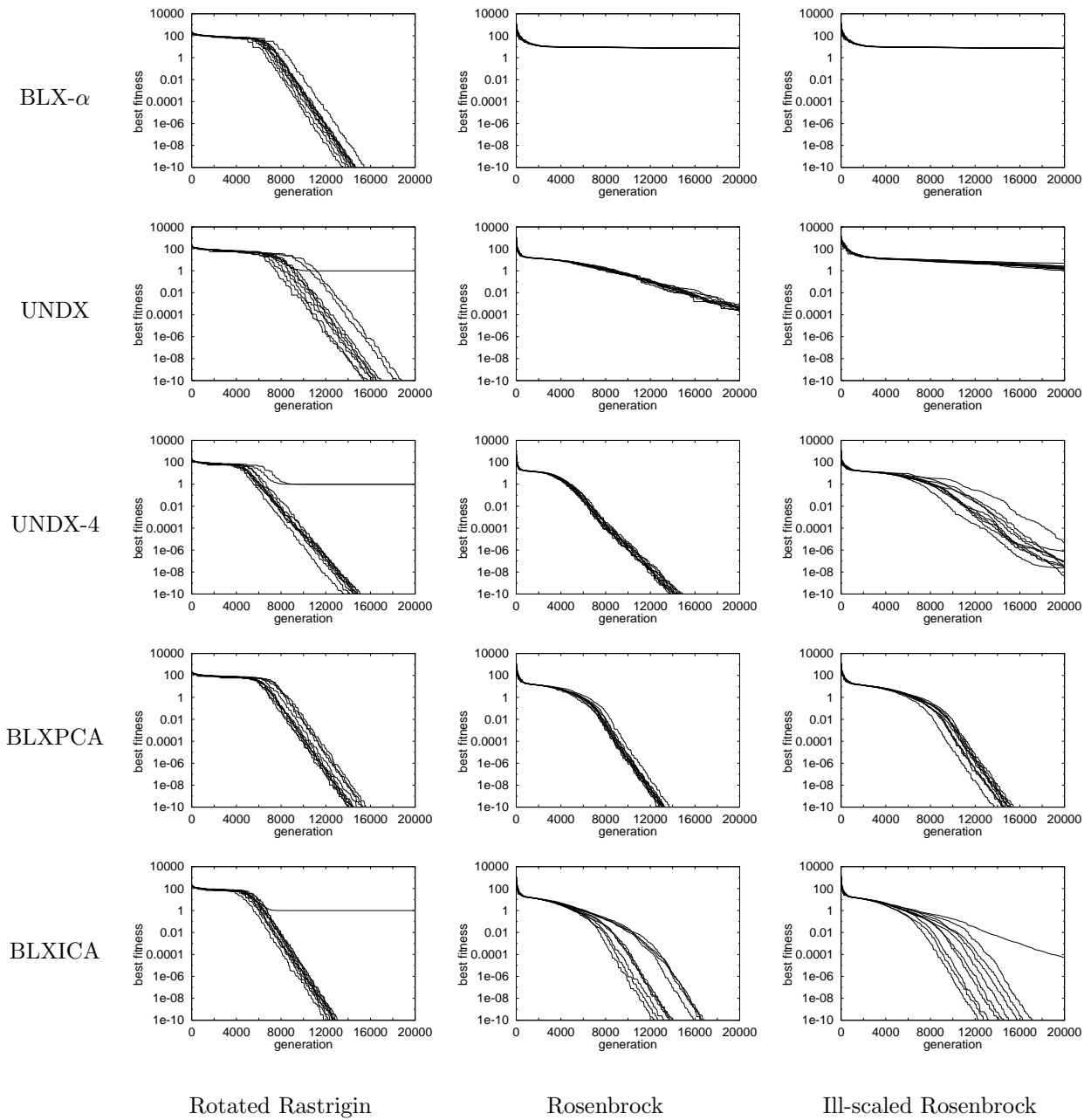


Figure 4: Comparison of the crossover operators. The abscissa shows the generation, and the ordinate shows the best fitness value among the population.

4.2 Setup of Experiments

Using the aforesaid test functions, performances of the proposed methods (BLXPCA and BLXICA) are compared with those of the conventional crossover operators.

- Compared Crossover Operators: Original BLX- α , UNDX, UNDX- m , BLXPCA and BLXICA.
- Population Size: 300 .
- Number of Children c : 200/generation.
- Trials: 10 runs with different random seeds for each test function and crossover operator.
- Initial Population: Generated randomly in the prescribed region following the uniform distribution.
- Parameter value of the BLX- α : $\alpha = 0.366$.
- Parameter values of the UNDX: $\sigma_\xi = 1.0$ and $\sigma_\eta = 0.35/\sqrt{20} - 1$
- Parameter value of the UNDX- m : $m = 4$, $\sigma_\xi = 1.0$ and $\sigma_\eta = 0.35/\sqrt{20} - 4$

4.3 Results

The results of the experiments are shown in Fig. 4. For the rotated Rastrigin function, besides a few runs that fails in finding the optimum, the UNDX-4 finds solution faster than the BLX- α and the UNDX. Performance of the BLXPCA is similar to that of BLX- α . Contrary to this, quicker convergence is achieved by the BLXICA. The performance of similar to that of the BLX- α for the original Rastrigin function which is a multi-modal but separable function. It suggests that the effectiveness of combining the ICA with the BLX- α for non-separable multi-modal fitness function.

For the Rosenbrock function that has strongly non-separable, the original BLX- α fails in finding the optimum. The UNDX and UNDX-4 achieve successful search, however, convergence of the UNDX is rather slower than the UNDX-4. With the BLXICA and BLXPCA, optimization performance similar to that of the UNDX-4 is achieved. It shows that with the transformation of coordinate system using the PCA or ICA, the BLX- α can find the optimal solution effectively. The convergence obtained by the BLXICA fluctuates by runs more than the BLXPCA. Reasons of this instability have not been clarified yet.

For the ill-scaled Rosenbrock function, not only the BLX- α but also the UNDX fails in finding optimum. Convergence with the UNDX-4 also slows down compared with the case of the Rosenbrock function. Contrary to this, with the BLXPCA and BLXICA, similar performance to the case of the Rosenbrock function is obtained. It shows that the transformation with the PCA and the ICA absorbs the effect of scale change of the coordinate systems.

Thus, the BLXICA and the BLXPCA achieves good search ability for non-separable, and poorly scaled fitness functions.

5 Conclusions

In this paper, for the real-coded genetic algorithms, a novel crossover operators that combine transformation of the coordinate system through the statistical analysis of the population and the blend crossover in the transformed spaces. While the numerical experiments are preliminary, the results shows that the performance of the proposed methods is promising. Subjects of future study are 1) treatment of the degeneration in the PCA/ICA, 2) reduction of computational load in the PCA/ICA, 3) refinement of selection operation more suitable to the PCA/ICA, 4) analysis of the transformation obtained by the PCA/ICA more in detail, and 5) more comprehensive numerical experiments to examine the effectiveness and limitations of the proposed methods. As for the reduction of computational load, a preliminary study shows that adopting the ICA for every five generations

reduces computational load remarkably without performance degradation.

Acknowledgments

The authors are grateful to Professor Shigenobu Kobayashi of Tokyo Institute of Technology for his valuable comments on this research. This research was supported by The ‘Research for the Future’ Program, Biologically Inspired Adaptive Systems (JSPS-RFTF96I00105)’ of The Japan Society for the Promotion of Science.

Bibliography

- [1] L. Davis: *The Handbook of Genetic Algorithms*, Van Nostrand Reinhold (1990).
- [2] C.Z. Janikow and Z. Michalewicz: An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms, R. K. Belew and L.B. Booker Eds.: *Proceedings of the 4th International Conference on Genetic Algorithms*, 31-36 (1991).
- [3] A. Wright: Genetic Algorithms for Real Parameter Optimization, *Foundations of Genetic Algorithms*, pp. 205-218 (1991).
- [4] Z. Michalewicz: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag (1992).
- [5] K. Deb and R. B. Agrawal: Simulated Binary Crossover for Continuous Search Space, *Complex Systems*, 9, 115-148 (1995).
- [6] L.J. Eshelman and J.D. Schaffer: Real-Coded Genetic Algorithms and Interval-Schemata, *Foundations of Genetic Algorithms 2*, pp. 187-202 (1993).
- [7] L.J. Eshelman, K.E. Mathias and J.D. Schaffer: Crossover Operator Biases: Exploiting the Population Distribution, *Proc. ICGA97*, 354/361 (1997)
- [8] I. Ono and S. Kobayashi: “A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover,” *Proc. 7th ICGA*, pp. 246-253 (1997).
- [9] H. Kita, I. Ono and S. Kobayashi: Multi-Parental Extension of the Unimodal Normal Distribution Crossover for Real-Coded Genetic Algorithms, *Proc. of CEC’99*, III-646-651 (1999).
- [10] S. Tsutsui, M. Yamamura and T. Higuchi: Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms, *Proc. GECCO’99* (1999)

- [11] I. Ono, H. Kita and S. Kobayashi: A Robust Real-Coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover: Effects of Self-Adaptation of Crossover Probabilities, *Proc. of the Genetic and Evolutionary Computation Conference'99*, 496-503 (1999).
- [12] R. Salomon: Performance Degradation of Genetic Algorithms under Coordinate Rotation, *Proc. of 5th Annual Conference on Evolutionary Programming*, pp. 155-161 (1996).
- [13] H. Kita, I. Ono and S. Kobayashi: Theoretical Analysis of the Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithms, *Proc. of the ICEC'98*, 529-534 (1998).
- [14] H.-P. Schwefel: *Evolution and Optimum Seeking*, Wiley (1995).
- [15] H. Satoh, M. Yamamura and S. Kobayashi: Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation, *Proc. IIZUKA '96*, pp. 494-497 (1997).
- [16] C. Jutten and J. Herault: Blind separation of sources, Part I: Adaptive algorithm based on neuromimetic architecture, *Signal Processing*, 24, 1-20 (1991).
- [17] A. Hyvärinen and E. Oja: A Fast Fixed-Point Algorithm for Independent Component Analysis, *Neural Computation*, 9(7):pp.1483-1492, (1997)
- [18] <http://www.cis.hut.fi/projects/ica/fastica/>
- [19] M. Pelikan, D.E. Goldberg, and F. Lobo.: A survey of optimization by building and using probabilistic models, IlliGALs Technical Report 99018 (1999).
- [20] M. Sebag and A. Ducoulombier: Extending population-based incremental learning to continuous search spaces, *Proc. PPSN V*, pp. 418-427 (1998).
- [21] P.A.N. Bosman and D. Thierens: Expanding from Discrete to Continuous Estimation of Distribution Algorithms: The IDEA, *Proc. PPSN VI*, pp. 767-776 (2000).

Appendix: Population Distribution and Independent Components

Assume that the fitness function $f(\mathbf{x})$ can be expressed by an additive form of functions $f_i(\mathbf{w}_i^T \mathbf{x})$ of weighted sums $\mathbf{w}_i^T \mathbf{x}$ of decision variables \mathbf{x} :

$$f(\mathbf{x}) = \sum_i f_i(\mathbf{w}_i^T \mathbf{x})$$

As a result of selection by the GA, the population distributes more densely where the fitness function takes smaller values (for minimization problems). Assume that the distribution of the population $P(\mathbf{x})$ follows the Boltzmann one:

$$P(\mathbf{x}) = \frac{\exp(-f(\mathbf{x})/T)}{Z} = \frac{1}{Z} \prod_i \exp\left(-\frac{f_i(\mathbf{w}_i^T \mathbf{x})}{T}\right)$$

where $T > 0$ is a parameter.

Transforming the coordinate system by the equations $\mathbf{y}_i = \mathbf{w}_i^T \mathbf{x}$, we obtain

$$P(\mathbf{y}) = C \prod_i \exp(-f(\mathbf{y}_i)/T) = \prod_i P_i(\mathbf{y}_i) \quad (15)$$

where C is a constant. Equation (15) shows that the distribution can be expressed by a product of the marginal probability density function of each component. Hence, if we can find such a transformation by the ICA of the population, the BLX- α that blends the solution in a component-wise manner will work effectively.