

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/243776468>

Multi-parent recombination with simplex crossover in real-coded genetic algorithms

Article · January 1999

CITATIONS

316

READS

810

3 authors, including:



Shigeyoshi Tsutsui

Hannan University

105 PUBLICATIONS 3,272 CITATIONS

SEE PROFILE



Masayuki Yamamura

Tokyo Institute of Technology

127 PUBLICATIONS 1,597 CITATIONS

SEE PROFILE

Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms

Shigeyoshi Tsutsui

Department of Management and Information Science
Hannan University
5-4-33 Amamihigashi, Matsubara,
Osaka 580, JAPAN
tsutsui@hannan-u.ac.jp

Masayuki Yamamura, Takahide Higuchi

Graduate School of Interdisciplinary Science and Engineering,
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku,
Yokohama 226-8502, JAPAN
{my, higuchi}@dis.titech.ac.jp

Abstract

In this paper, we proposed *simplex crossover* (SPX), a multi-parent recombination operator for real-coded genetic algorithms. SPX generates offspring vector values by uniformly sampling values from simplex formed by m ($2 \leq m \leq \text{number of parameters} + 1$) parent vectors. The SPX features an independence from of coordinate systems. Experimental results using test functions, which are commonly used in studies of evolutionary algorithms, showed SPX works well on functions having multimodality and/or epistasis with a medium number of parents: 3-parent on a low dimensional function or 4 parents on high dimensional functions.

1. Introduction

In many Evolutionary Algorithms (EAs), a recombination operation with two parents is commonly used to produce offspring. However, we need not restrict ourselves to two parents recombinations only as EAs allow us to emulate natural evolution in a more flexible manner. Recently, several attempts studying the use of more than two parents for recombination in EAs have been reported [Eiben 94, 95, 96, 97, Schewefel 95, Smith 96, Voigt 95, Ono 97, Tsutsui 98a].

The generalized multi-parent recombination operators in GAs are *scanning crossover* and *diagonal crossover*, which were introduced by Eiben et al. in [Eiben 94, 95 96]. In [Eiben 94, 95], these operators were evaluated on the standard test functions with bit string representation and other types of problems; they showed that 2-parent recombination was inferior when evaluated on the test functions. In [Eiben 96], these operators were evaluated on Kauffman's NK-landscapes, and their evaluation noted the superiority of sexual recombination on mildly epistatic problems. In [Eiben 97], Eiben and Bäck extended Evolution Strategies (ESs) to multi-parent recombination involving a variable number of parents to create an offspring, then performed an empirical in-

vestigation of multi-parent recombination. The investigated multi-parent operators were a generalized version of intermediary recombination, scanning crossover, and diagonal crossover. Although the performance of the algorithm depended on the particular combination of recombination operators and objective functions, in most cases they observed a significant increase in performance as the number of parents increased.

In recent years, several real-coded Genetic Algorithms (GAs) for function optimization, which use real number vector representation of chromosomes, have been studied [Goldberg 91, Davis 91, Eshelman 93, Janikow 91, Michalewicz 94, Wright 91, Ono 97, Herrera 98]. These GAs have been shown to outperform traditional bit string based representation. The effect of multi-parent recombination has also been studied for real-coded GAs.

In [Ono 97], Ono and Kobayashi proposed *unimodal normal distributed crossover* (UNDX) for real-coded GAs. The UNDX generates offspring using a normal distribution defined by three parents. Offspring are generated around the line segment connecting the two parents, Parent 1 and 2. The third parent, Parent 3, is used to decide the standard deviation of the distance to the axis connecting Parent 1 and 2. This operator has a feature of independence from coordinate systems and is excellent in characteristics preservation for function optimization. Ono and Kobayashi showed the UNDX has high performance compared with the BLX- α of Eshelman and Schaffer [Eshelman 93]. Tsutsui and Ghosh proposed three types of multi-parent recombination operators, namely, the *center of mass crossover operator* (CMX), *multi-parent feature-wise crossover operator* (MFx), and *seed crossover operator* (SX) for real-coded GAs in [Tsutsui 98a]. Each of these operators is a natural generalization of the 2-parent recombination operator. They used the BLX- α as the base operator. The results showed that in these three operators, performance with CMX improved as the number

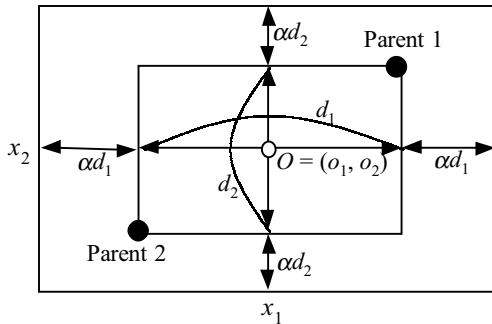
of parents increased on functions having multimodality and/or epistasis.

In this paper, we propose *simplex crossover* (SPX), a new multi-parent recombination operator for real-coded GAs. The SPX is a simple crossover operator which uses property of a *simplex* in the search space. The SPX has a balance between exploration and exploitation, and is independent from coordinate systems in generating offspring. The experimental results showed high performance on test functions with multimodality and/or epistasis.

In the next section, we give a detailed description of the SPX operator. In Section 3, we describe experimental methodology. Then in Section 4, an empirical analysis of the results is given. Finally, concluding remarks are made in Section 5.

2. Simplex Crossover (SPX)

A well known two parents crossover operator for real-coded GAs is the BLX- α of Eshelman and Shaffer [Eshelman 93] shown in Fig. 1. This operator is a simple one and it is reported to work well on a wide range of problems. The BLX- α uniformly and independently picks new individuals with values that lie in $[d_i - \alpha d_i, d_i + \alpha d_i]$ for each diagonal axis. Eshelman, Mathias and Schaffer extended BLX- α and proposed box-BLX- α - β , pool-wise box-BLX- α - β and directional-BLX- α - β - γ [Eshelman 96, 97]. Whether the pool-wise or pair-wise version of box-BLXs is better depends upon the problem. However, they discovered that if there is strong linkage among the parameters, then the pair-wise box-BLX is the better operator. In the pair-wise case, biases in the population distribution are likely to reflect linkage among



BLX- α uniformly picks new individuals with values that lie in $[o_1 - d_1/2 - \alpha d_1, o_1 + d_1/2 + \alpha d_1]$ on axis x_1 and $[o_2 - d_2/2 - \alpha d_2, o_2 + d_2/2 + \alpha d_2]$ on axis x_2 , where (o_1, o_2) is center of the two parents and (d_1, d_2) is distance between the parents

Fig. 1 BLX- α ($n=2$) [Eshelman 93]

parameters, so it is better to exploit these bias; in the pool-wise case, biases in the population distribution are likely to be spurious, so it is better to ignore these biases [Eshelman 97]. If the population distribution lies along a region that forms a diagonal in the problem space, it makes more sense to sample in a region that parallels the diagonal [Surry 97]. A directional-BLX is intended to introduce such a sampling bias [Eshelman 97].

The simplex crossover (SPX) proposed in this paper is devised so that the operator has both box- and directional-BLX features with properties of the *simplex* with the m -parent ($m \geq 2$) parameter vectors.

In R^n , $n+1$ points that are independent of each other form a simplex. For simplicity, let us first consider a 3-parent SPX in a two-dimensional search space as shown in Fig. 2, where $X^{(1)}$, $X^{(2)}$ and $X^{(3)}$ are parameter vectors of the three parents. Then, these vectors form a simplex. As is done in the BLX- α , we expand this simplex for each direction $(X^{(j)} - O)$ by $(1+\epsilon)$ ($\epsilon \geq 0$) times, where O is the center of mass of the three parents calculated as

$$O = \frac{1}{3} \sum_{j=1}^3 X^{(j)}$$

and

$$Y^{(j)} = (1+\epsilon)(X^{(j)} - O) + O$$

We then generate three offspring by **uniformly picking** vector values from this expanded simplex. Thus, we generate three offspring from these three parents.

In general, the SPX is specified as SPX- n - m - ϵ , where n is the number of parameters of the search space, m is the number of parents and ϵ is a control parameter that defines the expanding rate. The SPX shown in Fig. 2 is specified as SPX-2-3- ϵ . Now let us define SPX- n - m - ϵ for a more general case where m is in the range $[2, n+1]$. Let $X = (x_1, \dots, x_n)$ be an n dimensional real number vector representing a possible solution (chromosome). In SPX- n - m - ϵ , m ($\leq n+1$) individuals $X^{(j)} = (x_1^{(j)}, \dots, x_n^{(j)})$, $j = 1, \dots, m$ are randomly selected for crossover from the parental pool (population) $\{X_1, \dots, X_N\}$.

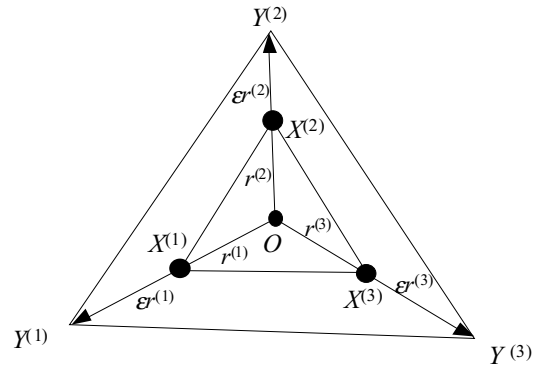


Fig. 2 SPX-2-3- ϵ

Then, we divide R^n into h sets of R^{m-1} and one R^q spaces by randomly combing n dimensions of coordinates with a non-duplicate as

$$R^n = \underbrace{R^{m-1} \times \dots \times R^{m-1}}_h \times R^q,$$

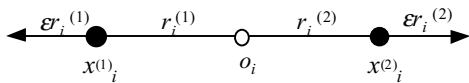
where, $h = \text{integer}(n/(m-1))$ and $q = \text{remainder}(n/(m-1))$. In each R^{m-1} , we develop a *simplex* using m parental vector elements that belong to R^{m-1} . We then generate m offspring vectors by uniformly picking vector element values from the simplex in each R^{m-1} . Finally, we replace the generated vector element values with the vector element values of $X^{(j)}$ in corresponding dimensional positions of R^n , and we obtain the m offspring vector $X^{(j)} = (x_1^{(j)}, \dots, x_n^{(j)})$, $i = 1, \dots, m$. Here, we leave q parameter values of $X^{(j)}$ that belong to R^q unchanged, i.e.,

$$x_k^{(j)} = x_k^{(j)} \mid \forall k \in R^q, j = 1, \dots, m.$$

Now, let us consider another extreme where m is 2, i.e., SPX- $n-2-\epsilon$. In this case R^{m-1} is R^1 . Since a simplex in a one-dimensional space is a *segment*, a offspring parameter value on an axis x_i is obtained by uniformly sampling the value from the segment as shown in Fig. 3. This sampling is identical with the sampling of the BLX- α in Fig. 1, i.e., BLX- $\alpha \equiv$ SPX- $n-2-\epsilon$. Furthermore, we can note that the value of ϵ corresponds to $2 \times \alpha$ of the BLX- α .

Thus, the designed SPX is an extension of BLX- α and expected to have the following features:

- (1) Since a simplex is basically independent of coordinate systems, the SPX can inherit this independency.
- (2) Since offspring vector values are uniformly sampled around m -parent vector values, they inherit characteristic of parents and the sampling likely reflects a certain linkage among the parameters.
- (3) As a result, SPX has a balance between exploration and exploitation in generating offspring, and it works well on functions having multimodality and/or epistasis among the parameters.
- (4) Since the uniform sampling in a simplex can be performed with simple procedures, SPX is a simple and non-time consuming operator.



SPX- $2-\epsilon$ uniformly picks new individuals with values that lie in $[o_i - r_i^{(1)} - \epsilon r_i^{(1)}, o_i + r_i^{(2)} + \epsilon r_i^{(2)}]$ on each axis x_i , where $r_i^{(1)} = r_i^{(2)}$. Since $r_i^{(1)} + r_i^{(2)}$ correspond to d_i of BLX- α in Fig. 1, ϵ is $2 \times \alpha$ of BLX- α .

Fig. 3 SPX- $n-2-\epsilon$

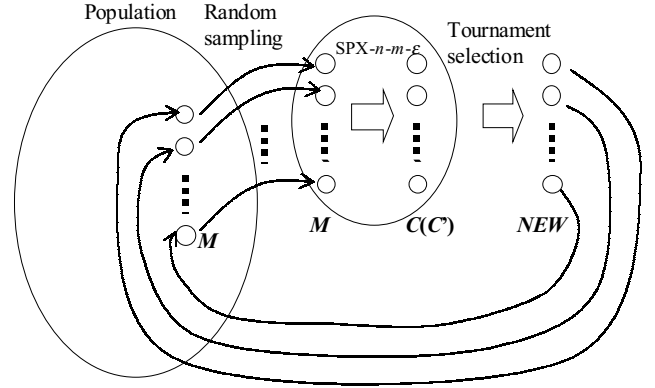


Fig. 4 Basic evolutionary model

3. Experimental Methodology

To evaluate the proposed SPX, we ran a real-coded GA. The methodology is as follows.

3.1 Basic Evolutionary Model

The basic evolutionary model we used in these experiments is based on the *minimal generation gap* (MGG) proposed by Satoh, Yamamura and Kobayashi [Satoh 96]. The MGG model has a desirable convergence property maintaining the diversity of the population, and shows higher performance than the other conventional models in a wide range of applications. We extend MGG to multi-parent version as follows (see also Fig. 4):

1. Set generation counter $t = 0$. Generate N individuals randomly as a initial population $P(t)$.
2. Select a set of m parents M by random sampling from the population $P(t)$.
3. Generate a set of m offspring C by applying the SPX- $m-\epsilon$ to M . Then apply a mutation to C (see Section 3.2) and get C' .
4. Select a set of m individuals NEW from set $M + C'$ using the tournament selection of size 2. Then replace M with NEW in population $P(t)$ and get population $P(t+1)$.
5. Stop if a certain specified condition is satisfied, otherwise set $t = t + 1$ and go to step 2.

3.2 Mutation Operator

In evolution strategies (ESs) [Schwefel 95], mutation is used as the main search operator. In contrast, mutation in GAs is used as a secondary operator, although it plays an important role in escaping from local optima. Several mutation operators for real-coded GAs are proposed in [Davis 91, Janikow 91, Michalewicz 94]. Since study in this paper places its

main focus on testing the effect of multi-parent crossover, so we use a simple static Gaussian mutation. The i -th parameter x_i of an individuals in C (see Section 3.1) is mutated by

$$x'_i = x_i + N(0, \sigma_i)$$

with a mutation rate p_m , where $N(0, \sigma_i)$ is a independent random Gaussian number with mean of zero and standard deviation σ_i . In this study, σ_i is fixed to $(\max_i - \min_i)/4$ and p_m is fixed to $0.2/n$ for all experiments, where \min_i and \max_i are the lower and upper limits of the parameter range on the i -th dimension of the search space.

3.3 Applying boundary extension by mirroring

For functions that have their optimum in the corner of the search space, it is difficult for a simplex to cover the optimum suitably. As a result, the possibility of the SPX generating offspring around the optimum point decreases, and the SPX's performance may be degraded for functions having their optimum in the corner of the search space. This feature in the SPX is very similar to that of the center of mass crossover (CMX) proposed in [Tsutsui 98a]. To cope with this problem in CMX, we have proposed the *boundary extension by mirroring* (BEM) method [Tsutsui 98b]. This method allows individuals to be located beyond the boundary of the search space to some extent, as shown in Fig. 5. The functional values of individuals located beyond the boundary of the search space are calculated as if they are located inside the search space at points symmetrical to the boundary. We have introduced an extension rate r_e ($0 < r_e < 1$) for a control parameter. The search space is centered in an extended space extended by a factor of $1+r_e$ along each dimension. The functional value of individual j with real vector $X^{(j)} = (x_1^{(j)}, \dots, x_n^{(j)})$ is obtained as

$$f(X^{(j)}) = f(Y^{(j)}),$$

where,

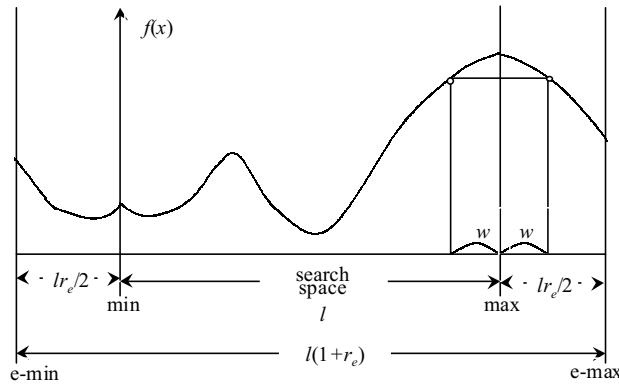


Fig. 5 Boundary extension by mirroring (BEM)

$$Y^{(i)} = (y_1^{(i)}, \dots, y_n^{(i)}),$$

$$y_j^{(i)} = \begin{cases} 2 \min_j - x_j^{(i)} & \text{if } x_j < \min_j \\ 2 \max_j - x_j^{(i)} & \text{if } x_j > \max_j \\ x_j^{(i)} & \text{otherwise.} \end{cases}$$

The \min_i and \max_i are the lower and upper limits of parameter range on the i -th dimension of the search space, respectively. By applying the BEM with the extension rate values of [0.2, 0.3] to CMX, the CMX's performance on the test functions which have their optimum on the corner of the search space was much improved. Moreover, the technique did not cause side effects for the functions whose optima are located around the center of the search space [Tsutsui 98b]. We applied the BEM method with extension rate r_e of 0.25 in this study.

3.4 Test functions

We use test functions commonly used in the literature, which includes the De Jong test suite (except $F4$), and the 20-parameter Rastrigin ($F_{Rastrigin}$), the 20-parameter rotated Rastrigin ($F_{rotated-Rastrigin}$), the 10-parameter Schwefel ($F_{Schwefel}$) and the 10-parameter Griewank ($F_{Griewank}$) functions. These functions are summarized in Table 1. $F1$ is a simple unimodal function and has the global minimum at (0, 0, 0). $F2$ has strong inter-parameter linkage (epistasis) and has the global minimum at (1, 1). $F3$ is a discontinuous function with the global minimum in the range $x_i \in [-5.12, -5.0)$ for $i = 1, \dots, 5$, i.e., in one corner of the search space. $F5$ is basically a continuous function, but it has 25 deep holes, and has the global minimum at (-31.978, -31.978). $F_{Rastrigin}$ is a multimodal function and the global minimum is at (0, ..., 0). There are many local minima around the global minimum. $F_{rotated-Rastrigin}$ is obtained by randomly rotating the original $F_{Rastrigin}$ function on $20 \times (20-1)/2 = 190$ pairs of a 2-dimensional hyperplane around the origin [Ono 97]. This is a multimodal function and has a strong epistasis among parameters. The global minimum is at (0, ..., 0). $F_{Schwefel}$ is also a multimodal function and the global minimum is at (420.968746, ..., 420.968746), very close to one corner of the search space. $F_{Griewank}$ is a multimodal function and the global minimum is at (0, ..., 0). This function has an inter-parameter linkage due to the presence of the product term. However, the effect of decreases as the number of parameters increases. Thus, there is weak epistasis in the 10 parameter version used here.

3.5 Performance measure

We evaluated the algorithms by measuring their #OPT (number of runs in which the algorithm succeeded in finding the global optimum) and MNT (mean number of trials to find the global optimum in those runs where it did find the optimum). We used the Δx_i value for resolution (borrowed from

Table 1 Test Functions

Functions	Range of x_i	Δx_i	Functional Characteristics			
			pos. *1	epis. *2	multi. *3	disc. *4
$F1 = \sum_{i=1}^3 x_i^2$	[-5.12, 5.11]	0.01	center	none	none	none
$F2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	[-2.048, 2.047]	0.001	mid*5	strong	none	none
$F3 = \sum_{i=1}^5 \lfloor x_i \rfloor$	[-5.12, 5.11]	0.01	corner	none	none	medium
$F5 = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	[-65.536, 65.535]	0.001	mid*5	none	low	strong
$F_{Rastrigin} = (20 \times 10) + \left[\sum_{i=1}^{20} (x_i^2 - 10 \cos(2\pi x_i)) \right]$	[-5.12, 5.11]	0.01	center	none	high	none
$F_{rotated-Rastrigin} = \text{rotated } F_{Rastrigin}$	[-5.12, 5.11]	0.01	center	strong	high	none
$F_{Schwefel} = \sum_{i=1}^{10} -x_i \sin(\sqrt{ x_i })$	[-512, 511]	1.0	corner	none	high	none
$F_{Griewank} = 1 + \sum_{i=1}^{10} \frac{x_i^2}{4000} - \prod_{i=1}^{10} (\cos(x_i / \sqrt{i}))$	[-512, 511]	1.0	center	weak	high	none

*1: Position of the optimum point, *2: Epistasis, *3: Multimodality, *4: Discontinuity, *5: middle of center and corner

bit string based GAs, Table1) to determine whether the optimal solution is found. If the solution detected is within the Δx_i range of the actual optimum point, we assume that the solution is detected. Let us represent the optimal solution of a function by (o_1, \dots, o_n) . Then, we assume that the real coded GA is able to find the optimal solution if all parameters (x_1, \dots, x_n) of the best individual are within the range $[(o_i - \Delta x_i/2), (o_i + \Delta x_i/2)]$ for all i . The real number is represented as type of data *double* in C language. The experiments were performed on a workstation with UltraSPARC chips.

4. Empirical Analysis of Results

Twenty (20) runs were performed. Each run continued until the global optimum was found or a maximum of 300,000 trials (function evaluations) was reached. A population size of 50 was used for functions $F1$, $F2$, $F3$ and $F5$. For the multimodal functions $F_{Rastrigin}$, $F_{rotated-Rastrigin}$, $F_{Schwefel}$ and $F_{Griewank}$ the population size of 500 was used.

The control parameters for SPX- n - m - ϵ are set as fol-

lows. The value of ϵ was set to 1.0, which corresponds to the α value of 0.5 in BLX- α . This value was used as a standard value in [Eshelman 93, 96, 97]. We tested the performance by varying m from 2 to $n+1$ for function $F1$, $F2$, $F3$ and $F5$. For 10 parameter functions ($F_{Schwefel}$, $F_{Griewank}$), we tested the performance by varying $m = 2, 3, 4, 5, 6$ and 11 ($n+1$). For 20 parameters function ($F_{Rastrigin}$, $F_{rotated-Rastrigin}$), we tested the performance by varying $m = 2, 3, 4, 5, 6, 11$ and 21 ($n+1$).

The results are summarized in Table 2. We evaluated both the with and without BEM methods. Note again that the SPX for $m = 2$ corresponds to BLX- α with α at 0.5.

The results on functions $F1$ (*unimodal, no epistasis*) showed slight improvement in performance as the number of parents increased. On function $F2$ (*unimodal, strong epistasis*) the performance for $m = 3$ was significantly higher than the performance for $m = 2$. On function $F3$ (*corner*), the SPX without BEM showed clear performance degradation as the number of parents increased from 2, as predicted. The SPX with BEM did not show this degradation and showed almost similar performance for $m = 2, 3, 4$. However, for $m = 5$ and 6, a significant performance degradation was observed and the

Table 2 Summary of results

Functions	r_e	m (number of parents)													
		2		3		4		5		6		11		21	
		#OPT	MNT (STD)	#OPT	MNT (STD)	#OPT	MNT (STD)	#OPT	MNT (STD)	#OPT	MNT (STD)	#OPT	MNT (STD)	#OPT	MNT (STD)
F_1	0.0	20	1,362.2 (183.9)	20	1,155.8 (158.1)	20	1,042.4 (316.2)								
	0.25	20	1,418.5 (128.5)	20	1,180.1 (127.3)	20	1,188.0 (562.6)								
F_2	0.0	20	9,318.0 (737.9)	20	4,840.5 (911.9)										
	0.25	20	21,535.1 (2418.8)	20	5,935.9 (1124.2)										
F_3	0.0	20	3,389.4 (467.5)	20	4,923.6 (802.5)	20	4,965.9 (1848.1)	20	9,091.1 (1695.3)	20	46,942.1 (8409.2)				
	0.25	20	1,366.6 (243.1)	20	1,387.4 (224.5)	20	1,350.0 (246.7)	20	2,296.2 (373.5)	20	4,605.0 (921.9)				
F_5	0.0	20	5,529.5 (894.6)	20	7,145.9 (1345.5)										
	0.25	20	5,770.6 (856.7)	20	7,145.9 (1345.5)										
$F_{\text{Rastrigin}}$	0.0	20	221,872.6 (12,030.5)	20	85,424.6 (5,211.0)	20	47,888.6 (2,368.7)	20	53,514.7 (9,518.8)	20	72,550.4 (9,000.2)	20	115,637.2 (8,398.7)	20	143,740.9 (36,046.9)
	0.25	20	229,467.7 (14,521.4)	20	88,049.7 (3,752.0)	20	51,142.5 (2,726.5)	20	58,607.7 (6,501.1)	20	75,250.3 (10,657.1)	20	125,250.5 (16,098.1)	19	151,616.7 (18,068.7)
$F_{\text{rotated-Rastrigin}}$	0.0	0	-	20	97,240.9 (3,787.6)	20	65,771.4 (16,231.4)	18	99,173.4 (12,485.9)	20	120,615.2 (17,795.9)	20	164,471.0 (20,047.5)	19	195,960.8 (18,997.7)
	0.25	0	-	19	101,765.6 (4,470.6)	19	67,365.5 (9,392.3)	19	105,238.9 (21,555.0)	17	127,146.1 (14,102.5)	18	178,772.3 (13,883.8)	18	196,514.8 (23,125.0)
F_{Schwefel}	0.0	20	120,770.0 (8,686.8)	2	251,312.0 (19,278.0)	2	279,754.5 (301.5)	6	255,418.2 (25,477.1)	0	-	0	-		
	0.25	20	93,002.1 (4,702.6)	20	96,522.6 (6,770.9)	20	112,676.7 (18,701.6)	20	108,850.0 (15,214.8)	2	260,909.5 (17,605.5)	0	-		
F_{Greiwank}	0.0	20	57,070.9 (4,232.9)	20	31,639.9 (1,765.7)	20	23,179.6 (1,462.6)	19	20,396.1 (1,496.6)	9	74,988.1 (68,908.5)	0	-		
	0.25	20	61,505.8 (4,868.8)	20	32,764.7 (1,520.7)	20	24,680.2 (1,468.9)	15	21,649.4 (2,283.3)	5	130,974.4 (104,449.5)	1	48,948.0 (-)		

worst performance was scored for $m = 6$ ($= n+1$). The results on function F_5 (discontinuous) showed slight performance degradation for $m = 3$ compared with the performance for $m = 2$.

The results on function $F_{\text{Rastrigin}}$ (highly multimodal) showed a significant increase in performance as the number of parents increased for $m = 2, 3$ and 4; the best performance was observed for $m = 4$. For $m = 5, 6, 11$ and 21 ($= n+1$), the performance degraded as the number of parents

increased.

The results on function $F_{\text{rotated-Rastrigin}}$ (highly multimodal, strong epistasis) showed clear characteristics of the SPX. No optimal solution was found for $m = 2$ (BLX-0.5), but SPX found an optimal solution fairly well for $m = 3$ and 4, and the MNTs were almost similar to the MNTs on function $F_{\text{Rastrigin}}$. Again the best value of MNT was observed for $m = 4$. For $m = 5, 6, 11$ and 21 ($= n+1$), the performance degraded as the number of parents increased as it did on func-

tion $F_{Rastrigin}$. Thus, we can see that the SPX with more than two parents has a clear feature of relative independence from coordinate systems. This feature of the SPX is a natural result of using *simplexes* in generating offspring, because a simplex is basically independent of coordinate systems and in SPX there is no sampling bias against coordinate systems in generating offspring. This feature of SPX should hold for $m = n+1$ (SPX-20-21- ϵ). However, the results for $m = 21$ with $F_{rotated-Rastrigin}$ were slightly worse than for $m = 21$ with $F_{Rastrigin}$.

The results on function $F_{Schwefel}$ (highly multimodal, corner) without BEM showed very poor performance for more than 2 parents. #OPTs for $m = 3, 4$ and 5 were 2, 2 and 6, respectively. No optimal solution was found for $m = 6$ and 11. Thus, as on function F_3 , the SPX without BEM did not work well on $F_{Schwefel}$ which has its optimum in the corner of the search space. On the other hand, the SPX results with BEM showed significant performance improvement with the best performance obtained for $m = 2$. However, the SPX with BEM did not work well for $m = 6$ and 11 (#OPTs of 2 and 0, respectively).

The results on function $F_{Griewank}$ (highly multimodal, weak epistasis) showed a significant increase in performance as the number of parents increased for $m = 2, 3$ and 4. The best performance was observed for $m = 4$. For $m = 5, 6$ and 11, the performance degraded as the number of parents increased and the worst performance was observed for $m = 11$ ($= n+1$) (without BEM #OPT: 0 and with BEM #OPT: 1).

Thus, SPX worked well on functions F_2 (strong epistasis), $F_{Rastrigin}$ (highly multimodality), $F_{rotated-Rastrigin}$ (highly multimodality, strong epistasis) and $F_{Griewank}$ (highly multimodality). In addition, the number of parents that showed

stable performance was 3 (F_2) or 4 (highly dimensional functions). By applying the BEM to SPX, the SPX performance on the test functions with an optimum on the corner of the search space (F_3 and $F_{Schwefel}$) was greatly improved. Furthermore, the BEM technique did not cause side effects for the functions whose optima are located around the center of the search space by choosing the appropriate extension rate r_e .

In the early stage of this study, we estimated that the SPX with $m = n+1$ (SPX- $n-n+1-\epsilon$) would work well since it uses a complete simplex in R^n , but the results of SPX- $n-n+1-\epsilon$ showed poor performance on high dimensional functions.

Fig. 6 shows a typical convergence process on function $F_{Griewank}$ for $m = 4$ and for $m = 11$. No mutation was applied in these runs. The vertical axes of the figure show mean values (MEAN) and the standard deviation (STD) of each parameter in the population with trials, respectively. An optimal solution was found at trial number 19,553 for $m = 4$ and no optimal solution was found for $m = 11$. For $m = 4$, the STDs of 10 parameters converged with an almost equal rate of decrease as the search proceeded, and MEANs converged to the values of the optimal solution (0,...,0). However, for $m = 11$, STDs converged faster than STDs for $m = 4$ with a variety of decreasing rates among parameters as search proceeded. This means the population rapidly loses diversity, and vector values in some specific dimensions converge faster than vector values in other dimensions. Here, simplexes reproduce dimensions. In this situation, a simplex becomes a *hyperplanes* in R^{10} and SPX has difficulty exploring the search space effectively (as the population finally gets trapped by a local optimal).

We may say an SPX with a large number of parents has sampling biases that reflect biases in the population distribution too much. We can prevent the over-sampling biases with a medium number of parents.

5. Conclusions

In this paper, we proposed *simplex crossover* (SPX), a multi-parent recombination operator for real-coded genetic algorithms. SPX generates offspring vector values by uniformly sampling values from the simplex formed by m ($2 \leq m \leq n$) parents vectors. The SPX features an independence from coordinate systems. The experimental results with test functions commonly used in the studies of evolutionary algorithms showed SPX works well on functions having multimodality and/or epistasis with a medium number of parents: 3-parent on a low dimensional function or 4 parents on high dimensional functions.

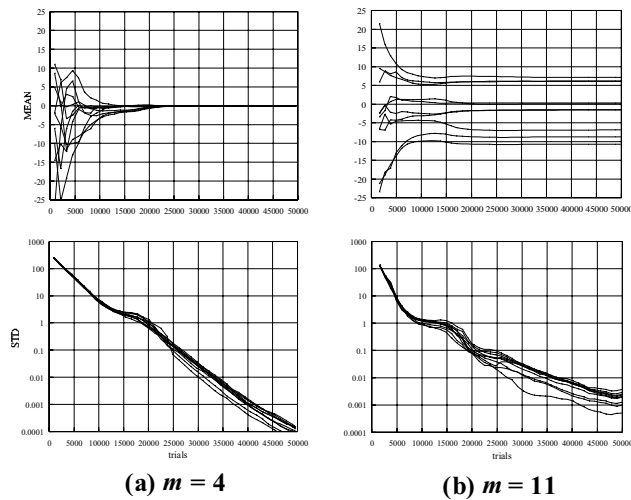


Fig. 6 A typical convergence process on function $F_{Griewank}$ for $m = 4$ (a) and for $m = 11$ (b)

There are many opportunities for further research related to the proposed SPX. We have now some theoretical results to determine the appropriate values for expansion rate ε , which produce good performance. We mainly analyzed SPX using #OPT and MNT. The detailed convergence properties of the SPX are still under investigation. A theoretical analysis of the SPX, comparing the SPX with other varieties of crossover operators in the literature and exploring SPX on real life problems are also to be tried.

This research is partially supported by the Ministry of Education, Science, Sports and Culture of Japan under Grant-in-Aid for Scientific Research number 10680396.

References

- [Davis 91] Davis, L.: *The Handbook of Genetic Algorithms*, Von Nostrand Reinhold, New York (1991).
- [Eiben 94] Eiben, A. E., Raue, P-E. and Ruttkay, Zs.: Genetic Algorithms with Multi-parent Recombination, *Proceedings of the Parallel Problem Solving from Nature III*, pp. 78-87(1994).
- [Eiben 95] Eiben, A. E., van Kemenade, C. H. M. and Kok, J. N.: Orgy in the Computer: Multi-Parent Reproduction in Genetic Algorithms, *Proceedings of the 3rd European Conference on Artificial Life*, LNAI 929, Springer-Verlag, pp. 934-945(1995).
- [Eiben 96] Eiben, A. E. and Schippers, C. A.: Multi-Parent's Niche: n-ary Crossover on NK-Landscapes, *Proceedings of the Parallel Problem Solving from Nature IV*, pp. 319-328 (1996).
- [Eiben 97] Eiben, A. E., Back, T.: Empirical Investigation of Multiparent Recombination Operators in Evolution Strategies, *Evolutionary Computation*, 5(3), pp. 347-365 (1997).
- [Eshelman 93] Eshelman, L. J. and Schaffer, J. D. : Real-Coded Genetic Algorithms and Interval-Schemata, *Foundations of Genetic Algorithms 2*, Morgan Kaufman Publishers, San Mateo, pp. 187-202 (1993).
- [Eshelman 96] Eshelman, L. J., Mathias, K. E. and Schaffer, D. : Convergence Controlled Variance, *Foundations of Genetic Algorithms 4*, Morgan Kaufman Publishers, San Francisco, pp. 203-224 (1996).
- [Eshelman 97] Eshelman, L. J., Mathias, K. E. and Schaffer, J. D.: Crossover Operator Biases: Exploiting the Population Distribution, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 354-361 (1997).
- [Goldberg 91] Goldberg, D. E.: Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking, *Complex Systems*, No. 5, pp. 139-167 (1991).
- [Herrera 98] Herrera, F., Lozano, M. and Verdegay, J. L.: Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, *Artificial Intelligence Review*, Vol. 12, No. 4, pp. 265-319 (1998).
- [Janikow 91] Janikow, C. Z. and Michalewicz, Z.: An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 31-36 (1991).
- [Michalewicz 94] Michalewicz, Z. : *Genetic Algorithms + Data Structures = Evolution Program*, Springer-Verlag (1994).
- [Ono 97] Ono, I and Kobayashi, S: A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 246-253 (1997).
- [Schwefel 95] Schwefel, H.-P. : *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series, Wiley, New York (1995).
- [Smith 96] Smith, J and Fogarty, T. C.: Recombination Strategy Adaptation via Evolution of Gene Linkage, *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, pp. 826-831, 1996.
- [Satoh 96] Satoh, H, Yamamura, M and Kobayashi, S: Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation, *Proceedings of the IIZUKA'96*, pp. 494-497 (1996).
- [Surry 96] Surry, P and Rsdcliffe, N. : Real Representations, *Foundations of Genetic Algorithms 4*, Morgan Kaufman Publishers, San Francisco, pp. 343-363 (1996).
- [Tsutsui 98a] Tsutsui, S and Ghosh, A.: A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms, *Proceedings of the 1998 IEEE ICEC*, pp. 828-833 (1998).
- [Tsutsui 98b] Tsutsui, S.: Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*, Springer-Verlag, pp. 428-437 (1998).
- [Voigt 95] Voigt, H.-M. and Mühlenbein, H. : Gene Pool Recombination and Utilization of Covariances for the Breeder Genetic Algorithms, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pp. 172-177, 1995.
- [Wright 91] Wright, A. H. : Genetic Algorithms for Real Parameter Optimization, *Foundations of Genetic Algorithms*, Morgan Kaufman Publishers, San Mateo, pp. 205-218 (1991).