Sampling Bias and Search Space Boundary Extension in Real Coded Genetic Algorithms

Shigeyoshi Tsutsui
Department of Management and Information Science, Hannan University
5-4-33 Amamihigashi, Matsubara, Osaka 580-5802 Japan
e-mail: tsutsui@hannan-u.ac.jp

Abstract
In Real coded genetic algorithms, some crossover operators do not work well on functions which have their optimum at the corner of the search space. To cope with this problem, we have proposed boundary extension methods which allows individuals to be located within a limited space beyond the boundary of the search space. In this paper, we give an analysis of the boundary extension methods from a view point of the sampling bias and perform a comparative study on the effect of applying boundary extension methods, namely the BEM (boundary extension by mirroring, the BES (boundary extension with extended selection). We were able to confirm that to use sampling methods which have smaller sampling bias had good performance on both functions which have their optimum at or near the boundaries of the search space, and functions which have their optimum at the center of the search space. The BES/SD/A (BES by shortest distance selection with aging) had good performance on functions which have their optimum at or near the boundaries of the search space. We also confirmed that applying the BES/SD/A did not cause any performance degradation on functions which have their optimum at the center of the search space. This feature of the BES/SD/A is very useful because when we solve some function optimization problems, we do not know the position of their optimal point.

1. INTRODUCTION

In recent years, many researchers have concentrated on using real-valued genes in genetic algorithms (GAs). It is reported that, for some problems, real-valued encoding and associated techniques outperform conventional bit string approaches [Davis, 91], [Eshelman 93], [Wright 91], [Janikow 91], [Surry 96], [Ono 97, 99].

In previous studies [Tsutsui 98, 99], we have proposed several types of multi-parent recombination operators for real-coded GAs. We found these operators did not work well on functions which have their optimum at or near the boundaries of the search space. To cope with this problem, we proposed a method which allows individuals to be located within a limited space beyond the boundary of the search space [Tsutsui 98]. The functional value of individuals located beyond the boundary of the search space was set to be the same as those of the point they map to by mirror reflection across the boundary. We called this method *boundary extension by mirroring* (BEM). With this method, the performance of multi-parent recombination operators improved in the test functions which have their optimum at or near the boundary of the search space. Further, by applying BEM, we observed clear improvement in the performance of two-parent recombinations in functions which have their optimum near the boundary of the search space.

In [Tsutsui 2000], we proposed another boundary extension method *boundary extension with extended selection (BES)* and presented a preliminary study on it. In the BES, virtual individuals are also produced inside the extended space. They are included in the population up to a defined maximum number by distance measure to the elite individual. No functional values of virtual individuals are used in this method. In this paper, we give an analysis of these boundary extension methods from a view point of sampling bias and do a comparative study on the effect of applying these methods to test functions which have their optimum at or near the boundary of the search space using a traditional two-parent recombination operator for real-coded GAs.

In the remainder of this paper, first we do an analysis of the sampling bias of crossover operators for real coded GAs in Section 2. In Section 3, we describe previously proposed boundary extension methods and propose an extension of the BES and analyze these methods. Then, in Section 4, empirical results and their analysis are given. Finally, concluding remarks are made in Section 5.

2. SAMPLING BIAS

For functions which have their optimum at or near the boundary of the search space, the possibility that a recombination operator generates offspring around the optimum point decreases because a portion of the feasible offspring space located beyond the boundary of the search space is cut away.

To see this bias, we provide an analysis using BLX-α operator [Eshelman 91]. Other recombination operators for real coded GAs such as UNDX [Ono 97], multi-parent recombination operators [Tsutsui 98], SPX [Tsutsui 99] also have this kind of bias. An analysis of sampling bias was done from a different angle in [Eshelman 97], [Kita 99].

Here, for simplicity, without loss of generality, we consider one dimensional search space X:

$$X = \{x; x_{min} \le x \le x_{max}\}, \quad (1)$$

and one-dimensional BLX-α operator as shown in Fig. 1. BLX-α uniformly picks new individuals with values that lie in $[I-\alpha I, I+\alpha I]$, where x_1 and x_2 are two parents. We must note that e_1 or e_2 in Fig. 1 must be between x_{min} and x_{max}. Here, we consider two types of sampling methods, type 1 sampling and type 2 sampling.

(1) Type 1 sampling

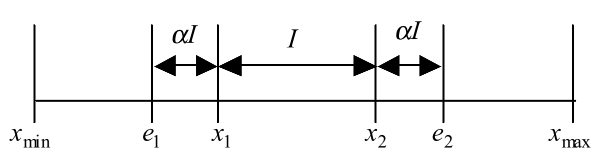
Type 1 sampling is very simple. An offspring y is sampled as follows:

$$y = e'_1 + u_1 \times (e'_2 - e'_1), \quad (2)$$

where

$$\begin{aligned} e_1 &= x_1 - \alpha \times (x_2 - x_1), \\ e_2 &= x_2 + \alpha \times (x_2 - x_1), \\ e'_i\ (i=1,2) &= \begin{cases} x_{min} & : \text{if } e_i < x_{min} \\ x_{max} & : \text{if } e_i > x_{max} \\ e_i & : \text{otherwise,} \end{cases} \\ u_1 &: \text{uniform random number} \in [0.0, 1.0]. \end{aligned} \quad (3)$$

Here, we assume parents are distributed uniformly in the range of $[x_{min}, x_{max}]$ and two parents x_1 and x_2 are randomly picked up independently as



BLX-a uniformly picks new individuals with values that lie in $[I\text{-}\alpha\ I, I\text{+}\alpha\ I]$, where x_1 and x_2 are two parents.

Fig. 1 BLX-α

$$\begin{aligned} x_1 &= v_1 \times (x_{max} - x_{min}), \\ x_2 &= v_2 \times (x_{max} - x_{min}), \\ v_1, v_2 &: \text{uniform random number} \in [0.0,1.0]. \end{aligned} \quad (4)$$

Fig. 2 ($p_1(y)$) shows the probability density function (p.d.f) of offspring y with α value of 0.5. From this figure, we can see that the sampling is biased toward the center of the search space as the number offspring produced around boundary of the search space are fewer than the number of offspring produced around center of the search space.

(2) Type 2 sampling

Type 2 sampling is intended to reduce the sampling bias observed in the sampling 1. Let c be the center of two parents x_1 and x_2 as

$$c = (x_1+x_2)/2. \quad (5)$$

Then, an offspring is sampled as follows:

$$y = \begin{cases} e'_1 + u_1 \times (c - e'_1) : \text{if } u_2 \ge 0.5 \\ c + u_1 \times (e'_2 - c) : \text{if } u_2 < 0.5, \end{cases} \quad (6)$$

where e'_1, e'_2, and u_1 are obtained from Eq. (3) and (4), and u_2 is an independent uniform random number from [0.0, 1.1].

Fig. 2 ($p_2(y)$) shows the p.d.f of offspring y. The sampling bias of type 2 sampling is reduced compared with the sampling bias of type 1. But an amount of bias still remains. Thus the number of offspring produced around the boundary of the search space are fewer than the number of offspring produced around center of the search space.

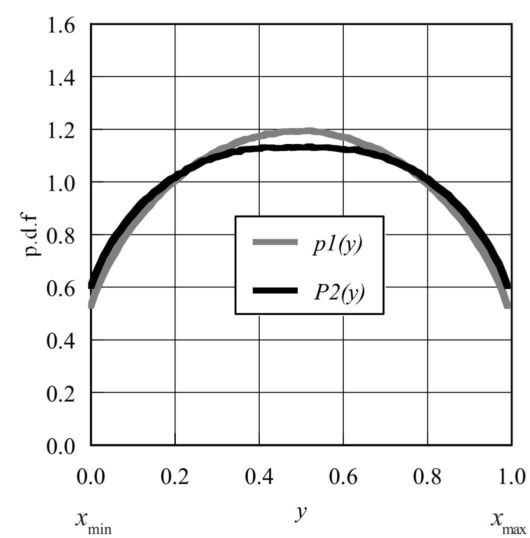


Fig. 2 Sampling bias in BLX-0.5. $p_1(y)$ and $p_2(y)$ are p.d.fs of offspring in type 1 and 2 samplings, respectively.

3 BOUNDARY EXTENSION METHODS

Boundary extension methods discussed in this section are introduced to cope with this sampling bias of recombination in real coded GAs.

3.1 Boundary extension by mirroring (BEM)

In the *boundary extension by mirroring* (BEM) method proposed in [Tsutsui 98], we allow individuals to be located within a limited space beyond the boundary of the search space, as shown in Fig. 3. The functional values of individuals located beyond the boundary of the search space (virtual individuals) are calculated as if they are located inside of the search space by setting the boundary as the mid-point of a mirror-image reflection and calculating the reflected point within the boundary. The functional value of offspring with real value y is obtained as within the boundary. We introduced an extension rate r_e ($0<r_e<1$) to control how much of the search space should be extended beyond the boundary. The search space is centered in a space extended by a factor of $1+r_e$ along each dimension. The functional value of offspring with real value y is obtained as

$$f(y) = f(y'), \tag{7}$$

where

$$y' = \begin{cases} 2\times x_{\min} - y : \text{if } y < x_{\min} \\ 2\times x_{\max} - y : \text{if } y > x_{\max} \\ y \quad : \text{otherwise,} \end{cases} \tag{8}$$

and x_{min} and x_{max} are the lower and upper limits of the search space.

Fig. 4 shows an example of sampling bias in the BEM using type 1 sampling. The r_e value of 0.5 is used and parents are assumed to be distributed uniformly in the range of [x_{min}, x_{max}]. $p(y)$ shows the p.d.f of sampled offspring in the range [$x_{e\text{-}min}$, $x_{e\text{-}max}$]. $p'(y)$ shows the p.d.f of sampling points which functional value values are evaluated in the range of [x_{min}, x_{max}], i.e., the p.d.f of offspring generated in the range of [x_{min}, x_{max}] + through mirror-image reflection of the virtual offspring. Although this is a rough estimate, we can see a certain degree of reduction of the sampling bias.

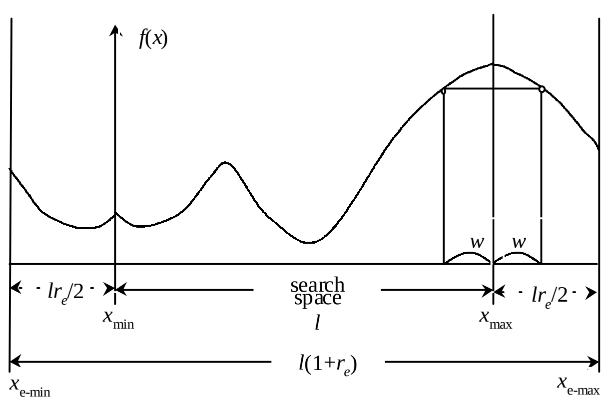


Fig. 3 Boundary extension by mirroring (BEM)

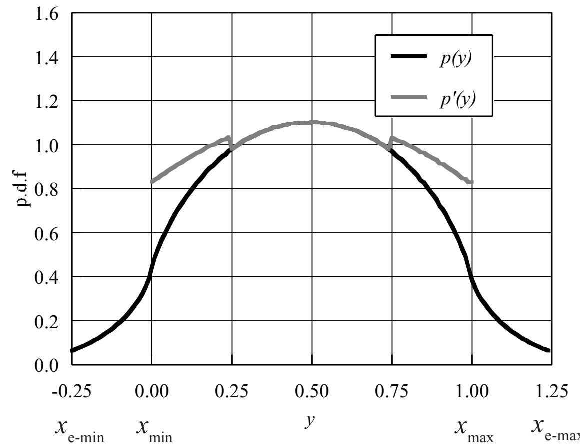


Fig. 4 Sampling bias in BEM

3.2 Boundary extension with extended selection (BES)

In the BEM in Section 3.1, a functional value of each virtual individual is calculate according to Eqs. (7) and (8) and each functional value is used in the selection operator. If we use an *extended selection* that allow us to select a number of virtual individuals as members of the new population without calculating their functional values, we may expect to get a similar effect to the BEM. We call this method *boundary extension with extended selection* (*BES*).

In the BES, virtual individuals are also produced inside the extended space defined r_e in the BEM. We call virtual individuals that are included in the population by an extended selection *helper individuals*. We introduce a control parameter *rh* (0<*rh*<1)*, helper individual rate*, that defines the maximum number of virtual individuals which are included in the population (see Fig. 5). Let N, V, and N_h be population size, the total number of virtual individuals generated, and the number of helper individuals, respectively. Then, N_h is determined as

$$N_h = \begin{cases} V \quad : \text{if } V \le N\times r_h \\ N\times r_h : \text{otherwise.} \end{cases} \tag{9}$$

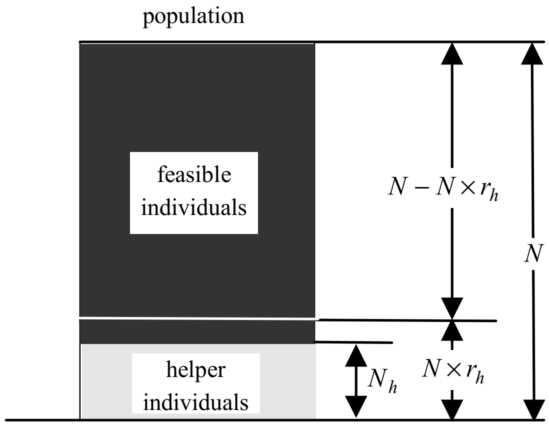


Fig. 5 Boundary extension with extended selection (BES). Feasible and helper individuals are shuffled for recombination.

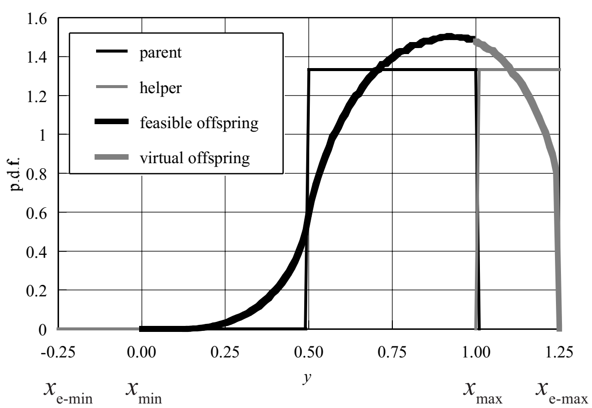


Fig. 6 An example of sampling bias in the BES

If the total number of virtual individuals generated is smaller than $N \times r_h$, then we select all the existing virtual individuals as helpers. But if it is greater than $N \times r_h$, we select helper individuals up to $N_h = N \times r_h$. For the later case, we must define the method to select $N \times r_h$ helper individuals from V virtual individuals. In the population, feasible and helper individuals are shuffled for recombination, and thus the helper individuals would help to produce more offspring around the boundary of the search space.

Now let's see the sampling bias of BES under a special situation as follows: The feasible individuals are distributed uniformly in the range of $[x_{min}+L/2, x_{max}]$ and helpers individuals are distributed uniformly in the range of $[x_{max}, x_{max}+0.25 \times L]$, where, $L=x_{max}-x_{min}$, with a helper rate of $r_h=0.5$, and an extension rate of $r_e = 0.5$. This situation implicitly expects more offsprings to be generated around the x_{max} zone. Fig. 6 shows the sampling bias in this situation, and we can see the sampling bias, in that many feasible offsprings

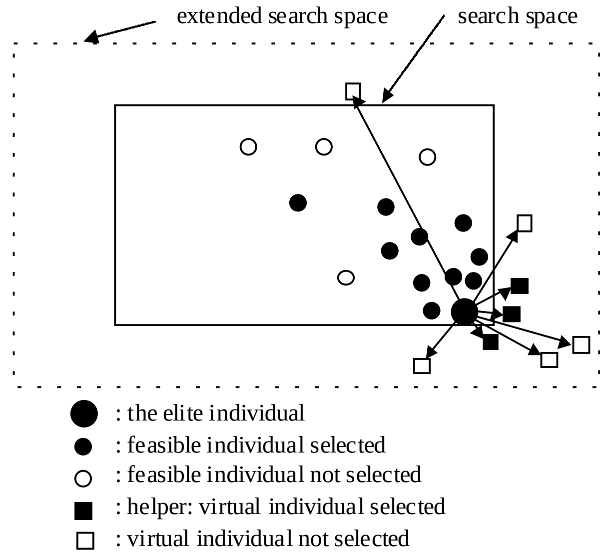


Fig. 7 Boundary extension with extended selection (BES/SD)

are sampled around x_{max}.

We can consider several methods to select N_h helpers from V virtual individuals. In this study, we propose the following two methods. Here, note that we can use any type of traditional selection operators to select N-N_h feasible (non-virtual) individuals.

(1) BES by shortest distance selection (BES/SD)

In BES by shortest distance selection (BES/SD), $N \times r_h$ helpers are select as follows: We first find the *elite* individual (i.e. individual which has the highest functional value) from the feasible individuals. Next, we calculate the Euclidean distance between each virtual individual and the elite individual. Then, we select $N \times r_v$ helper individuals that are nearest to the elite individual (see Fig. 7).

(2) BES by shortest distance selection with aging (BES/SD/A)

In the preliminary study in (Tsutsui 2000), the BES/SD showed relatively fair performance on the test functions which have their optimum at or near the boundary of the search space. But a slight performance degradation was observed on the test functions which have their optimum in the center of the search. This would be because we allow helper individuals to survive through continuous generations if the distance condition is satisfied. In fact, when we solve a problem, we do not know whether the solution is around the boundary or center of the search space.

In the BES/SD/A, we introduce the scheme of aging of individuals similar to that proposed in [Ghosh 97]. When a virtual individuals is selected as a helper i, its age a_i is set to zero (0). When it is mated with another and it produces offspring, a_i increases by 1. If a_i reaches a maximum defined age k_a, the helper i is removed from the population. For a problem which has its optimum around the center of the search space, the number of helper individuals may decrease. Thus, the number of offspring produced around the boundary of the search space becomes small and we can expect reduced performance degradation on the problems which have their optimum around center of the search space.

4. THE EXPERIMENTS

4.1 Experimental methodology

To evaluate proposed boundary extension methods, we ran a real coded GA. The experimental conditions were as follows.

(1) Boundary extension methods

We test the BEM, the BES/SD, and the BES/SD/A. The effect of these methods was evaluated for extension rate $r_e = 0.2$, helper individual ratio $r_h = 0.5$. For the BES/SD/A, we evaluated for maximum age k_a=1, 2, 3, 4.

(2) Crossover and mutation operators

For crossover, we use BLX-α with type 1 and type 2 samplings in Section 2. Here note type 1 sampling has greater sampling bias than type 2 sampling. The α value for BLX-α is 0.5 for all functions.

We use a simple static Gaussian mutation. The i-th parameter x_i of an individual in $I(t)$ is mutated by

$$x_i^{'} = x_i + N(0,\sigma_i) \qquad (10)$$

with a mutation rate of p_m, where $N(0,\sigma_i)$ is an independent random Gaussian number with a mean of zero and standard deviation of σ_i. In this study, σ_i is fixed to $(\max_i - \min_i)/2$ 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) Basic evolutionary model

The basic evolutionary model we used in these experiments is similar to that of the CHC [Eshelman 91] and (μ+λ)-ES [Schewefel 95]. Let the population size be N, and let it, at time t, be represented by $P(t)$. The population $P(t+1)$ is produced as follows: A collection of $N/2$ pairs is randomly composed, and crossover is then applied to each pair, generating N offspring which are placed in a temporal pool $I(t)$.

For the BEM, the individuals are ranked and the best N from the $2\times N$ in $P(t)$ and $I(t)$ are selected to form $P(t+1)$. For the BES, the individuals are ranked and the best $N - N_h$ from the $2\times N - V$ of feasible individuals in $P(t)$ and $I(t)$ are selected, and N_h of helper individuals from V of virtual individuals in $P(t)$ and $I(t)$ are selected.

In either case, the best solution obtained so far is always included in $P(t+1)$.

(4) Test Functions

We selected test functions which are commonly used in the literature and have their optimum at or near the boundaries of the search space, which includes the De Jong F_3, 10-parameter Schwefel (F_{Schwefel}), modified 20-parameter Rastrigin ($F_{\text{M-Rastrigin}}$), original 20-parameter Rastrigin ($F_{\text{Rastrigin}}$), 20-parameter sphere function F_{Sphere}, and modified 20-parameter sphere function $F_{\text{M-Sphere}}$ (Table 1). $F_{\text{M-Rastrigin}}$ and $F_{\text{M-Sphere}}$ are modified so that its optimum is located just at the corner of the search space. Original $F_{\text{Rastrigin}}$ and 20-parameter sphere function F_{Sphere} were used to see the side effect of applying the boundary extension methods.

F_3 is a discontinuous function with a global minimum in the rage $x_i \in [-5.12, -5.0)$ for $i = 1,...,5$, i.e., in one corner of the search space. F_{Schwefel} is a multimodal function and the global minimum is at (420.968746,...,420.968746), very close to one corner of the search space. $F_{\text{M-Rastrigin}}$ is also a multimodal function and the global minimum is at (0,...,0), in just one corner of the search space. $F_{\text{M-Sphere}}$ is a unimodal function and the global minimum is at (0,...,0), in just one corner of the search space.

(4) Performance measure

We evaluated the algorithms by measuring their #OPT (number of runs in which the algorithm succeeded in finding the global optimum) and MNE (mean number of function evaluations to find the global optimum in those runs where it did find the optimum). We used Δx_j value as resolution (borrowed from bit string based GAs, Table1) to determine whether the optimal solution was found. We defined the successful detection of the solution as being within Δx_j range of the actual optimum point. We represented the optimal solution of a function by $(o_1,...,o_n)$. If all parameters $(x_1,...,x_n)$ of the best individual are within the range $[(o_j - \Delta x_j/2), (o_j + \Delta x_j/2)]$ for all j, we assumed the real-coded GA to have found the optimal solution.

Thirty (30) runs were performed. In each run, the initial population $P(0)$ was randomly initialized in the original search space. Each run continued until the global optimum was found or a maximum of 500,000 trials was reached. A population size of 500 was used for all functions.

4.2 Analysis of results

The results are summarized in Table 2 for type 1 sampling and Table 3 for type 2 sampling, respectively. Here, "NORMAL" refers to the GA without boundary extension.

First, we will look at the results with type 1 sampling in Table 2. Results with both the BEM and the BES show clear

Table 1. Test Functions

function	range of x_j	Δx_j	position of the optimum
$F_3 = \sum_{i=1}^{5} \lfloor x_i \rfloor$	[-5.12, 5.11]	0.01	**corner**
$F_{\text{Schwefel}} = \sum_{i=1}^{10} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-512, 511]	1.0	**corner**
$F_{\text{M-Rastrigin}} = 20\times 10 + \sum_{i=1}^{20} (x_i^2 - 10\cos(2\pi x_i))$	[0.0, 5.11]	0.01	**corner**
$F_{\text{M-Sphere}} = \sum_{i=1}^{20} x_i^2$	[0.0, 5.11]	0.01	**corner**
$F_{\text{Rastrigin}} = 20\times 10 + \sum_{i=1}^{20} (x_i^2 - 10\cos(2\pi x_i))$	[-5.12, 5.11]	0.01	**center**
$F_{\text{Sphere}} = \sum_{i=1}^{20} x_i^2$	[-5.12, 5.11]	0.01	**center**

performance improvement on the test functions (F_3, F_{Schwefel}, $F_{\text{M-Rastrigin}}$, $F_{\text{M-Sphere}}$) which have their optimum at or near the boundaries of the search space. The performance with the BES slightly outperforms the performance with the BES. However, on the test functions ($F_{\text{Rastrigin}}$ and F_{Sphere}), which have their optimum at the center of the search space, the BES/SD showed poorer performance (on $F_{\text{Rastrigin}}$: MNE for NORMAL = 126,683.8, MNE for the BES/SD = 237,998.4, and on F_{Sphere}: MNE for NORMAL = 47,546.1, MNE for the BES/SD = 59,028.6). This performance degradation of the BES/SD arose because, in the BES/SD, we allowed helper individuals to survive through continuous generations if the distance condition is satisfied. Thus, helper individuals tended to produce more offspring around the boundary of the search space. The BES/SD/A prevents this because helpers each have an *age,* and individuals which have an age greater than k_a are deleted from the population. Thus, the number of useful helper individuals decreased.

Fig. 8 shows the change of N_h (the number of helper individuals) on function $F_{\text{Rastrigin}}$ in a single typical run for the BES/SD and the BES/SD/A. The value of N_h of the BES/SD remained at 250 ($N \times r_h$: upper limit). But the value of N_h of the BES/SD/A (a_g = 2) decreased as the generations increased. Thus, in the BES/SD/A the number of helper individuals was adaptively adjusted. This is evidence of the effectiveness of BES/SD/A. The value of a_g = 2 shows fairly good performance consistently on both functions which have their optimum at or near the boundaries of the search space, and those which have their optimum at the center of the search space.

Next we will look at the results with type 2 sampling in

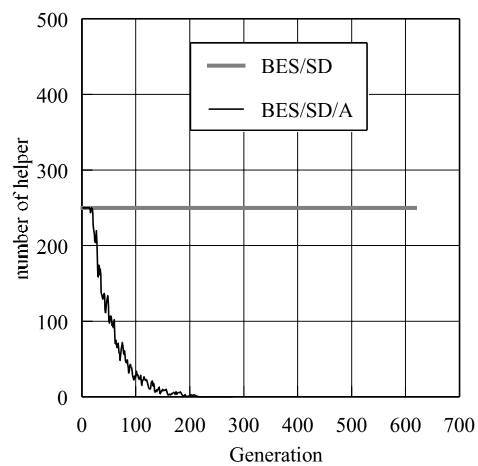


Fig. 8 Typical change of number of helper individuals on function FRastrigin for k_a = 2

Table 3. We can see that the NORMAL GAs with type 2 sampling showed much better performance than the NORMAL GAs with type 1 sampling 1 on functions which have their optimum at or near the boundaries of the search space. On the test functions which have their optimum at the center of the search space, the performance is almost the same in both the type 1 and type 2 samplings. Thus, we can confirm that sampling methods which have a smaller sampling bias are preferred. The performance of the BEM is almost the same with that of NORMAL GAs on all test functions except F_3.

Again, the BES/SD/A with a_g value of 2 shows fairly good performance. No meaningful performance degradation

Table 2 Summary of results (type 1 sampling)

function		NORMAL	boundary extension method					
			BEM	BES/SD	BES/SD/A (k_a)			
					1	2	3	4
F_3	#OPT	20	20	20	20	20	20	20
	MNE	17,733.6	9,263.6	8,680.3	13,682.9	8,762.3	8,239.6	8,811.5
	STD	1,505.9	809.2	1,435.0	1,022.6	573.6	906.1	1,024.5
F_{Schwefel}	#OPT	20	20	20	20	20	20	20
	MNE	64,633.6	56,137.2	61,040.4	51,768.8	45,676.3	59,936.6	53,197.7
	STD	8,284.6	2,995.4	14,400.5	8,228.8	8,259.1	14,382.2	12,768.0
$F_{\text{M-Rastrigin}}$	#OPT	2	20	20	20	20	20	20
	MNE	492,673.5	396,085.7	160,366.7	447,971.4	292,577.1	206,445.0	170,403.9
	STD	5,411.5	17,757.8	16,794.5	22,391.4	22,299.3	20,678.8	17,541.7
$F_{\text{M-Sphere}}$	#OPT	20	20	20	20	20	20	20
	MNE	79,914.8	47,757.7	46,987.4	56,532.2	40,817.5	42,069.6	43,959.4
	STD	1,044.0	986.8	1,198.4	1,089.5	556.1	952.7	739.3
$F_{\text{Rastrigin}}$	#OPT	20	20	20	20	20	20	20
	MNE	126,683.8	118,731.9	237,998.4	119,788.8	118,767.9	144,867.6	200,761.4
	STD	18,824.4	17,094.5	67,558.3	22,684.2	21,749.3	24,687.4	41,644.5
F_{Sphere}	#OPT	20	20	20	20	20	20	20
	MNE	47,546.1	48,639.0	59,028.6	46,530.6	46,328.4	46,751.5	55,509.6
	STD	759.0	703.4	1,856.7	953.3	839.4	805.4	1,341.5

Table 3 Summary of results (type 2 sampling)

function		NORMAL	boundary extension method					
			BEM	BES/SD	BES/SD/A (k_a)			
					1	2	3	4
F_3	#OPT	20	20	20	20	20	20	20
	MNE	13,685.3	9,160.9	8,085.7	13,428.0	8,682.9	7,945.6	8,125.3
	STD	823.3	923.8	935.6	934.0	700.1	1,006.0	919.6
F_{Schwefel}	#OPT	20	20	20	20	20	20	20
	MNE	49,014.9	58,261.9	54,654.7	45,493.5	38,108.3	48,551.2	51,687.5
	STD	2,965.0	4,663.7	9,904.9	6,265.1	7,623.3	14,287.1	14,935.3
$F_{\text{M-Rastrigin}}$	#OPT	20	20	20	20	20	20	20
	MNE	392,606.4	387,795.9	144,717.0	431,693.6	263,958.5	174,000.7	159,759.1
	STD	8,944.9	26,214.9	14,507.9	16,332.5	12,716.2	14,679.2	16,552.5
$F_{\text{M-Sphere}}$	#OPT	20	20	20	20	20	20	20
	MNE	64,700.7	45,321.7	45,487.4	56,315.9	39,585.4	41,367.0	42,683.4
	STD	906.5	836.5	1,272.1	949.7	602.2	823.9	725.2
$F_{\text{Rastrigin}}$	#OPT	20	20	20	20	20	20	20
	MNE	123,208.2	123,276.4	241,807.5	126,365.2	128,446.4	196,824.7	191,266.5
	STD	13,453.5	21,559.5	69,983.8	17,873.1	19,966.0	48,794.8	57,986.9
F_{Sphere}	#OPT	20	20	20	20	20	20	20
	MNE	48,492.0	49,011.5	57,107.2	46,718.6	46,672.4	48,875.3	53,250.3
	STD	792.9	1,127.9	2,036.9	838.2	743.5	1,406.7	1,530.5

by applying BES/SD/A was observed on the test functions which have their optimum at the center of the search space. We can see also that the BES/SD/A with type 2 sampling showed better performance than the BES/SD/A with type 1 sampling on the functions in general.

5. CONCLUSIONS

In this paper, we gave an analysis of the boundary extension methods from a view point of the sampling bias and did a comparative study on the effect of applying three boundary extension methods, namely the BEM, the BES/SD, and BES/SD/A, to test functions which have their optimum at or near the boundary of the search space and functions which have their optimum at the center of the search space. We used two types of BLX-α operators, i.e., type 1 sampling and type 2 sampling, where type 2 sampling has smaller sampling bias than type 1 sampling.

First, we were able to confirm that to use sampling methods which have smaller sampling bias had good performance on both functions which have their optimum at or near the boundaries of the search space, and functions which have their optimum at the center of the search space.

Next, the BES/SD/A with maximum age a_g value of 2 had good performance on functions which have their optimum at or near the boundaries of the search space. Since each helper has a maximum age, applying the BES/SD/A did not cause any performance degradation on functions which have their optimum at the center of the search space. This feature of the BES/SD/A is very useful because when we solve some function optimization problems, we do not know the position of their optimal point.

Another advantage of the BES/SD/A is that it does not use the functional value of each virtual (helper) individual. So, this approach may be applicable to more general function optimization, such as constrained parameter optimization, where the functional value in non-feasible regions are difficult to calculate [Michalewicz 94]. Further work remains to be done in applying BES/SD/A to such constrained parameter optimization problems. Although we applied the shortest distance method to select helper individuals, using other heuristics also remains for future work.

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).

[Eshelman 91] Eshelman, L. J.: The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, *Foundations of Genetic Algorithms*, Morgan Kaufmann, pp.265-283 (1991).

[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 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).

[Ghosh 97] A. Ghosh, S. Tsutsui and H. Tanaka: Genetic search with aging of individuals, *International Journal of Knowldge-Based Intelligent Engineering Systems*, Vol.1, no. 2 pp. 86-103 (1997).

[Goldberg 91] Goldberg, D. E.: Real-coded genetic algorithms, virtual alphabets, and blocking, *Complex Systems*, No. 5, pp. 139-167 (1991).

[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).

[Kita 99] Kita, H. and Yamamura, M.: A functional specialization hypothesis for designing genetic algorithms, Proc. IEEE International Conference on Systems, Man, and Cybernetics (in CD-ROM) (1999).

[Michalewicz 94] Michalewicz, Z. : *Genetic algorithms + data structures = evolution program*, Springer-Verlag (1994).

[Renders 94] Renders, J-M. and Bersini, H.: Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways, Proc. of the First IEEE Conference on Evolutionary Computation, pp. 312-317 (1994).

[Ono 97] Ono, I. and Kobayashi, S.: A Real-coded genetic algorithm for function optimization using unimodal normal distribution crossover, *Proc. of the Seventh International Conference on Genetic Algorithms*, pp. 246-253 (1997).

[Ono 99] Ono, I., Kita, H., and Kobayashi, S.: 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 (GECCO-99)*, Morgan Kaufmann, pp.496-503 (1999).

[Schewefel 95] Schewefel, H.-P.: *Evolution and optimum seeking*, Sixth-Generation Computer Technology Series, Wiley (1995).

[Surry 96] Surry, P. D. and Radcliffe, N. : Real representations, *Foundations of Genetic Algorithms 4*, Morgan Kaufman Publishers, San Francisco, pp. 343-363 (1996).

[Tsutsui 2000] Tsutsui, S.: A Comparative study on search space boundary extension methods in real coded genetic algorithms, *Proc. of The Third International Workshop on Frontiers in Evolutionary Algorithms (FEA 2000)* (to appear).

[Tsutsui 98] Tsutsui, S.: Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring, *Proc. of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V)*, Springer-Verlag, pp. 428-437 (1998).

[Tsutsui 99] Tsutsui, S., Yamamura, M., and Higuchi, T: Multi-parent recombination with simplex crossover in real coded genetic algorithms, *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-99)*, Morgan Kaufmann, pp.657-664 (1999).

[Wright 91] Wright, A. H. : Genetic algorithms for real parameter optimization, *Foundations of Genetic Algorithms*, Morgan Kaufman Publishers, San Mateo, pp. 205-218 (1991).