Information Sciences, Vol. 133/3-4, pp. 229-247 (2001.5)

Search Space Boundary Extension Method in Real-Coded Genetic Algorithms

Shigeyoshi Tsutsui* and David E. Goldberg**

* Department of Management and Information Science, Hannan University, 5-4-33 Amamihigashi, Matsubara, Osaka 580-8502, Japan, tsutsui@hannan-u.ac.jp

**Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 117 Transportation Building, 104 S. Mathews Avenue, Urbana, IL 61801, deg@illigal.ge.uiuc.edu

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 a 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 the view point of sampling bias and perform a comparative study on the effect of applying two boundary extension methods, namely the boundary extension by mirroring (BEM) and the boundary extension with extended selection (BES). 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.

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]. The theoretical studies of real-coded GAs have also been perfomed [Goldberg 91], [Crossman 92], [Eshelman 93], [Qi 94a, b].

Previous studies [Tsutsui 98, 99] have proposed several types of multi-parent recombination operators for real-coded GAs. 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, a method was proposed 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 that of the point they map to by mirror reflection across the boundary. This method was called *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. BEM improved in the performance of two-parent recombinations in functions which have their optimum near the boundary of the search space.

Anothor study [Tsutsui 00] proposed 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 using distance measure to the elite individual to select virtual individuals. 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. Future work is discussed in Section 5. Finally, concluding remarks are made in Section 6.

2. Sampling Bias

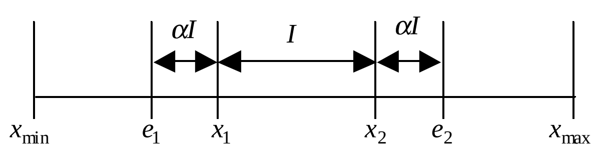
For functions which have their optimum at or near the boundary of the search space, the possibility that a recombination operator generates offspring around the optimum point decreases because a portion of the feasible offspring space located beyond the boundary of the search space is cut away.

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:

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

and one-dimensional BLX-α operator as shown in Fig. 1. BLX-α uniformly picks new individuals with values that lie in $[I\text{-}\alpha I, I+\alpha I]$, 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 three types of sampling methods, type 1, type 2, and type 3 samplings.



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

Fig. 1 BLX-α

(1) Type 1 sampling

Type 1 sampling repeats sampling until offspring becomes feasible as follows:

$$y = \begin{cases} e_1 + u \times (e_2 - e_1) : \text{if } x_{\min} \le y \le x_{\max} \\ \text{repeat sampling : otherwise,} \end{cases} \tag{2}$$

where

$$\begin{aligned} &e_1 = x_1 - \alpha \times (x_2 - x_1) \\ &e_2 = x_2 + \alpha \times (x_2 - x_1) \\ &x_1, x_2 : \text{two parents} \\ &u : \text{uniform random number} \in [0.0, 1.0]. \end{aligned} \tag{3}$$

If sampled point is out of the search space, sampling is repeated until it becomes feasible.
Here, we assume that 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

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

Fig. 2 ($p_1(y)$) shows the the probability density function (p.d.f.) of offspring y with a 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 offsprings produced around center of the search space.

Following type 2 and type 3 sampling are intended to reduce the sampling bias observed in the sampling 1.

(2) Type 2 sampling

An offspring y is sampled from the adjusted space so that it is always feasible as follows:

$$y = e'_1 + u \times (e'_2 - e'_1), \tag{5}$$

where

$$e'_i\ (i = 1,2) = \begin{cases} x_{\min} : e_i < x_{\min} \\ x_{\max} : e_i > x_{\max} \\ e_i \ : \text{otherwise.} \end{cases} \tag{6}$$

(3) Type 3 sampling

Let c be the center of two parents x_1 and x_2 as

$$c = (x_1+x_2)/2. \tag{7}$$

Then, an offspring is sampled as follows:

$$\begin{aligned} &y = \begin{cases} e'_1 + u_1 \times (c - e'_1) : \text{if } u_2 \ge 0.5 \\ c + u_1 \times (e'_2 - c) : \text{if } u_2 < 0.5, \end{cases} \\ &u_1, u_2 : \text{uniform random number} \in [0.0, 1.0], \end{aligned} \tag{8}$$

where e'_1 and e'_2 are obtained from Eq. (3).

$p_2(y)$ and $p_3(y)$ in Fig. 2 show the p.d.fs of offspring y for Type 2 and 3 samplings, respectively. The sampling biases of type 2 and 3 samplings are 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 is fewer than the number of offspring produced around center of the search space.

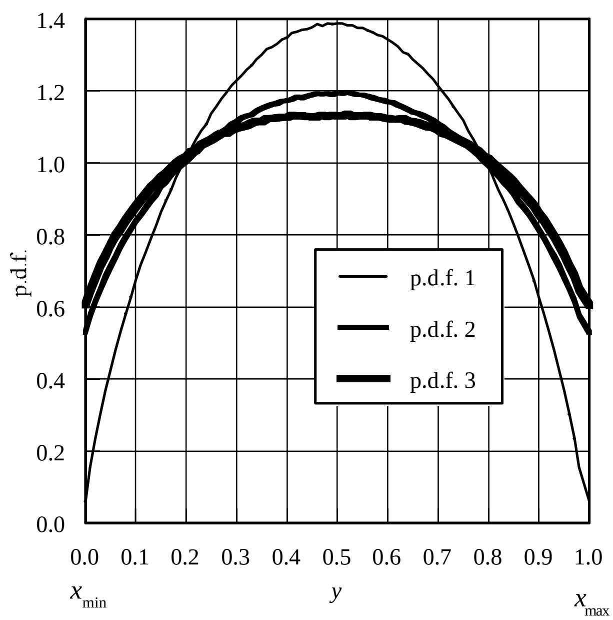


Fig. 2 Sampling biases in BLX-0.5 for type 1, 2, and 3 sampling, respectively

3. Boundary Extension Methods

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

3.1 Boundary extension by mirroring (BEM)

In the BEM (*boundary extension by mirroring*) 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

$$f(y) = f(y'), \quad (9)$$

where

$$y' = \begin{cases} 2\times x_{\min} - y : \text{if } y < x_{\min} \\ 2\times x_{\max} - y : \text{if } y > x_{\max} \\ y \quad : \text{otherwise} \end{cases} \tag{10}$$

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 2 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 whose functional 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.

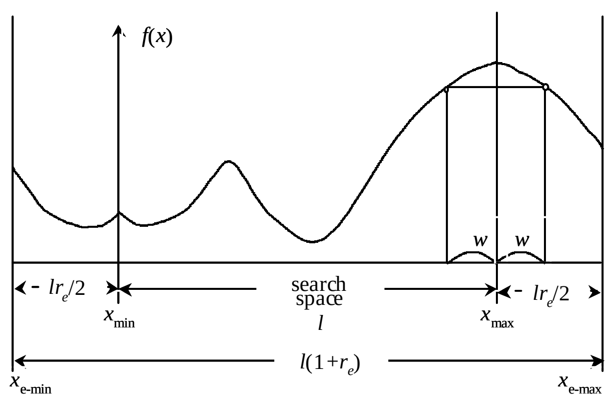


Fig. 3 Boundary extension by mirroring (BEM)

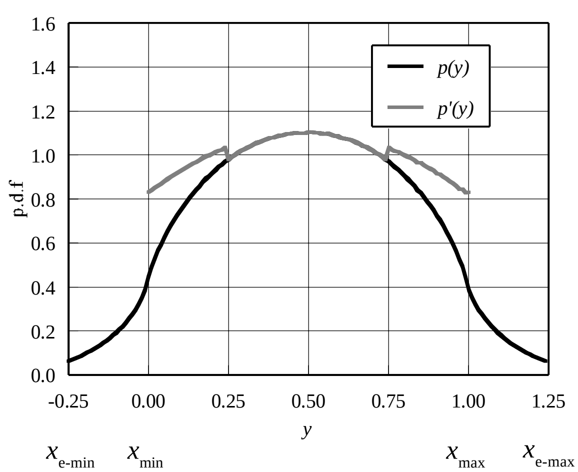


Fig. 4 Sampling bias in BEM

3.2 Boundary extension with extended selection (BES)

In the BEM in Section 3.1, a functional value of each virtual individual is calculated according to Eqs. (9) and (10) and each functional value is used in the selection operator. If we use an *extended selection* that allows 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 on the BEM. We call this method *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 r_h ($0<r_h<1$), the *helper individual rate*, that defines the maximum number of virtual individuals which are included in the population (Figs. 5). Let N, V, and N_h be the population size, the total number of virtual individuals generated, and the number of helper individuals, respectively. Then, N_h is determined as

$$N_h = \begin{cases} V & : \text{if } V \le N \times r_h \\ N \times r_h & : \text{otherwise.} \end{cases} \qquad (11)$$

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 (Fig. 6).

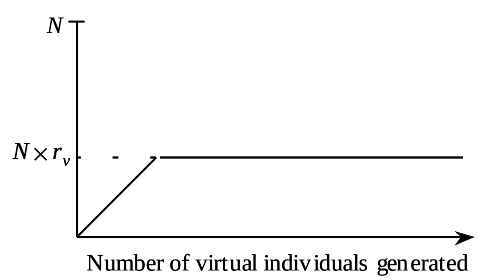


Fig. 5 Number of virtual individuals to be selected

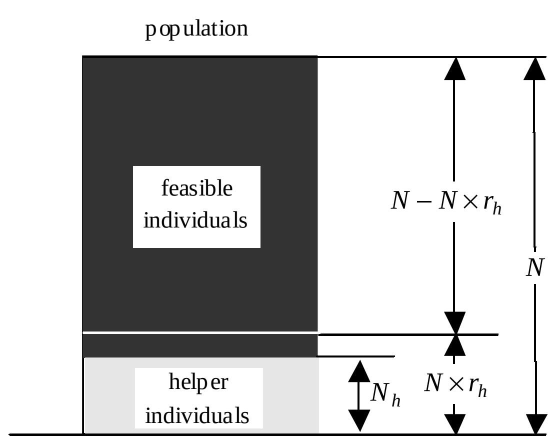


Fig. 6 Boundary extension with extended selection (BES). Feasible and helper individuals are shuffled for recombination.

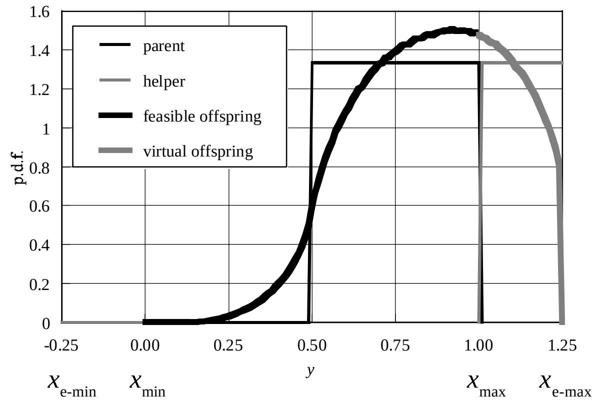


Fig. 7 An example of sampling bias in the BES

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 helper 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 offspring to be generated around the x_{max} zone. Fig. 7 shows the sampling bias in this situation using type 3 sampling, and we can see the sampling bias, in that many feasible offsprings 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 selected as follows: We first find the *elite*

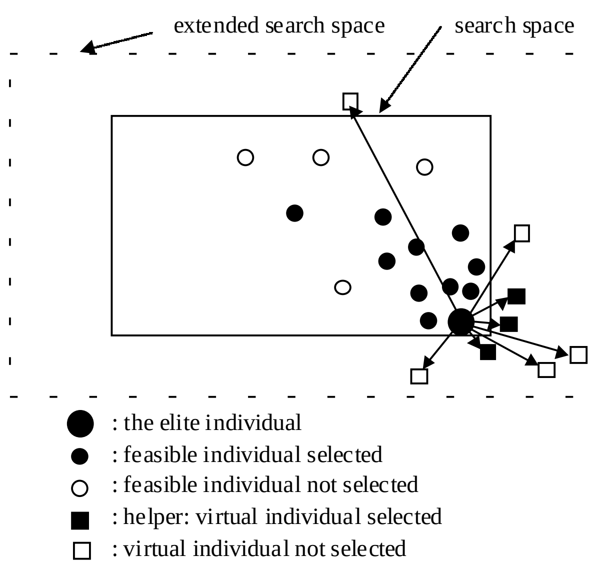


Fig. 8 Boundary extension with extended selection (BES/SD)

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. 8).

(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 space. 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 $r_e =$ 0.3, 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 of 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

$$x_i' = x_i + N(0, \sigma_i) \tag{11}$$

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 $(\mu+\lambda)$-ES [Schewefel 95]. Let the population size be N, and let it, at time t, be represented by $P(t)$. The

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**

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 range $x_i \in$ [-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.

(5) Performance measure

We evaluated the algorithms by measuring their #OPT or number of runs in which the algorithm succeeded in finding the global optimum and MNE or 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 function evaluations was reached. A population size of 400 was used for all functions.

4.2 Analysis of results

The results are summarized in Table 2 for (a) type 1 sampling, (b) type 2 sampling, and (c) type 3 sampling, respectively. Here, "NORMAL" refers to the GA without boundary extension. All runs found optimal solutions, i.e., #OPT = 30.

First, we look at the difference among three different samplings, i.e., type 1, type 2, and type 3 samplings with NORMAL GA. Results on functions which have their optimum at or near the boundaries of the search space (F_3, $F_{Schwefel}$, $F_{M\text{-}Rastrigin}$, $F_{M\text{-}Sphere}$) with type 3 sampling were much better than with type 1 and type 2 samplings. On functions which have their optimum at the center of the search space ($F_{Rastrigin}$ and F_{Sphere}), these three samplings showed almost same the performances. Type 3 sampling has a small sampling bias compared with type 2 and type 3 sampling. So, we may say to use crossover operators which have a smaller sampling bias is important to get high performance since in general we do not know the positions of the optima when we solve a given problem.

Next, we look at the performance of three boundary extension methods, i.e., the BEM, the BES/SD, and BES/SD/A. Results with the BEM showed clear performance improvement on functions F_3, $F_{Schwefel}$, $F_{M\text{-}Rastrigin}$, and $F_{M\text{-}Sphere}$ (except on $F_{Schwefel}$ with type 3 sampling). No meaningful side effect by applying the BEM on functions $F_{Rastrigin}$ and F_{Sphere}, was observed. Results with the BES/SD showed bigger performance improvement on functions F_3, $F_{Schwefel}$, $F_{M\text{-}Rastrigin}$, and $F_{M\text{-}Sphere}$ than the BEM (except on $F_{Schwefel}$ with type 2 and type 3 samplings). However, as we predicted, performance of the BES/SD on functions $F_{Rastrigin}$ and F_{Sphere} (which have their optimum at the center of the search space) showed poorer performance than that of NORMAL. 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 harmful offsprings around the boundary of the search space. The BES/SD/A prevents this because each helper has an age; an individual which has an age greater than k_a is deleted from the population. Thus, the existence of harmful helper individuals can be reduced. As a result, no side effect was observed by applying BES/SD/A on the functions which have their optimum at the center of the search space ($F_{Rastrigin}$ and F_{Sphere}).

To confirm this, let us observe the changes of number of helper individuals N_h in a single typical run with BES/SD/A on each function. Fig. 9 shows typical changes in the number of helper individuals (N_h) with the BES/SD/A for type 3 sampling and k_a value of 2 (two). On the functions which have their optimum at or near the boundaries of the search space (F_3, $F_{Schwefel}$, $F_{M\text{-}Rastrigin}$, $F_{M\text{-}Sphere}$), number of helper individuals N_h remained at 200 ($N \times r_h$: upper limit). But on the functions which have their optimum at the center of the search space ($F_{Rastrigin}$ and F_{Sphere}), the value of N_h decreased as the generations increased. Thus, in the BES/SD/A, the number of helper individuals is adjusted in adaptive. This is evidence of the effectiveness of BES/SD/A. Furthermore, the BES/SD/A showed good performance on the functions F_3, $F_{Schwefel}$, $F_{M\text{-}Rastrigin}$, and $F_{M\text{-}Sphere}$ (which have their optimum at or near the boundaries of the search space) also. This may due to diversity of helper individuals would be maintained since each helper individuals has its age. The value of $k_a = 2$ 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.

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

Table 2 Summary of results

(a) with type 1 sampling

function		NORMAL	boundary extension method					
			BEM	BES/SD	BES/SD/A (k_a)			
					1	2	3	4
F_3	MNE	14,559.5	7,701.4	**6,830.6**	10,778.2	7,289.2	7,011.6	7,032.5
	STD	1,049.0	826.9	924.3	1,022.0	548.9	1,219.3	756.8
F_{Schwefel}	MNE	51,204.9	43,864.0	46,307.0	44,618.7	**37,284.8**	43,675.3	47,798.1
	STD	5,454.3	3,049.7	9,110.8	7,359.1	8,579.4	12,492.4	12,029.2
$F_{\text{M-Rastrigin}}$	MNE	414,271.4	317,736.6	**129,488.9**	362,223.8	236,220.4	163,810.3	143,400.1
	STD	14,248.8	14,410.5	12,630.3	21,957.1	15,804.8	14,891.7	12,239.7
$F_{\text{M-Sphere}}$	MNE	64,772.4	39,804.3	38,084.8	45,542.4	**32,809.9**	34,370.7	35,971.9
	STD	584.6	737.5	949.7	752.9	436.2	790.3	911.7
$F_{\text{Rastrigin}}$	MNE	**101,852.2**	103,884.5	206,242.0	108,087.2	110,583.9	125,328.5	192,033.9
	STD	16,584.0	15,618.7	36,332.0	21,464.1	20,627.9	27,552.6	67,193.4
F_{Sphere}	MNE	38,383.1	38,655.4	47,297.5	37,314.8	**37,113.6**	37,581.7	43,740.2
	STD	764.3	613.9	1,465.0	822.8	1,037.5	808.5	1,516.5

(b) with type 2 sampling

function		NORMAL	boundary extension method					
			BEM	BES/SD	BES/SD/A (k_a)			
					1	2	3	4
F_3	MNE	14,684.7	7,848.8	**6,967.0**	11,019.7	7,189.7	6,990.0	6,936.2
	STD	1,394.4	776.5	910.5	783.6	485.9	986.1	844.9
F_{Schwefel}	MNE	49,841.8	43,742.2	51,377.6	44,620.4	**40,284.5**	44,638.8	47,038.2
	STD	6,144.4	3,305.9	10,010.1	9,101.9	10,458.7	13,831.1	14,755.1
$F_{\text{M-Rastrigin}}$	MNE	417,044.9	315,846.0	127,226.0	361,598.0	239,429.9	167,313.1	**138,867.2**
	STD	15,927.1	11,136.5	11,410.7	22,308.1	13,333.8	15,910.2	9,646.3
$F_{\text{M-Sphere}}$	MNE	64,923.7	39,822.4	38,603.0	45,765.6	**32,810.2**	34,256.8	35,961.5
	STD	587.4	829.9	751.2	652.1	572.7	728.8	955.0
$F_{\text{Rastrigin}}$	MNE	108,552.8	107,869.7	214,435.6	101,140.3	**104,010.5**	133,907.4	180,357.1
	STD	22,008.6	19,510.8	49,722.9	17,933.2	18,448.1	35,939.4	45,439.3
F_{Sphere}	MNE	38,442.1	38,726.2	47,280.4	37,561.8	**37,230.6**	37,748.5	43,818.5
	STD	738.4	842.4	1,245.5	689.3	802.0	643.7	1,138.6

(c) with type 3 sampling

function		NORMAL	boundary extension method					
			BEM	BES/SD	BES/SD/A (k_a)			
					1	2	3	4
F_3	MNE	11,190.8	7,156.0	6,674.4	10,634.5	6,569.4	6,676.4	**6,520.6**
	STD	697.2	678.9	985.6	1,138.1	574.1	1,073.1	824.2
F_{Schwefel}	MNE	39,649.4	40,740.0	43,567.6	39,126.4	**32,243.0**	39,500.5	42,192.2
	STD	2,523.4	2,493.4	8,466.3	7,268.1	7,378.2	7,263.0	9,351.1
$F_{\text{M-Rastrigin}}$	MNE	316,338.0	298,957.6	118,007.6	348,947.4	**212,035.8**	146,432.3	131,514.1
	STD	8,129.4	20,054.6	12,243.9	21,182.5	14,863.8	9,992.7	12,458.7
$F_{\text{M-Sphere}}$	MNE	51,872.1	36,497.1	37,373.9	45,074.9	**32,094.9**	33,234.0	34,765.7
	STD	877.3	751.5	836.8	668.2	619.5	821.3	1,090.5
$F_{\text{Rastrigin}}$	MNE	109,228.5	105,379.4	217,493.0	**101,871.2**	108,243.6	187,949.5	170,045.6
	STD	21,273.7	18,636.5	58,507.2	18,428.4	17,803.7	54,930.8	35,346.2
F_{Sphere}	MNE	38,402.6	38,909.5	45,942.9	37,400.8	**37,275.1**	39,137.7	42,674.2
	STD	857.7	807.2	1,190.8	587.4	747.7	636.3	782.8

degradation by applying BES/SD/A was observed on the test functions which have their optimum at the center of the search space. We can also see also that the BES/SD/A with type 3 sampling showed better performance than the BES/SD/A with type 1 and type 2 samplings on the functions in general.

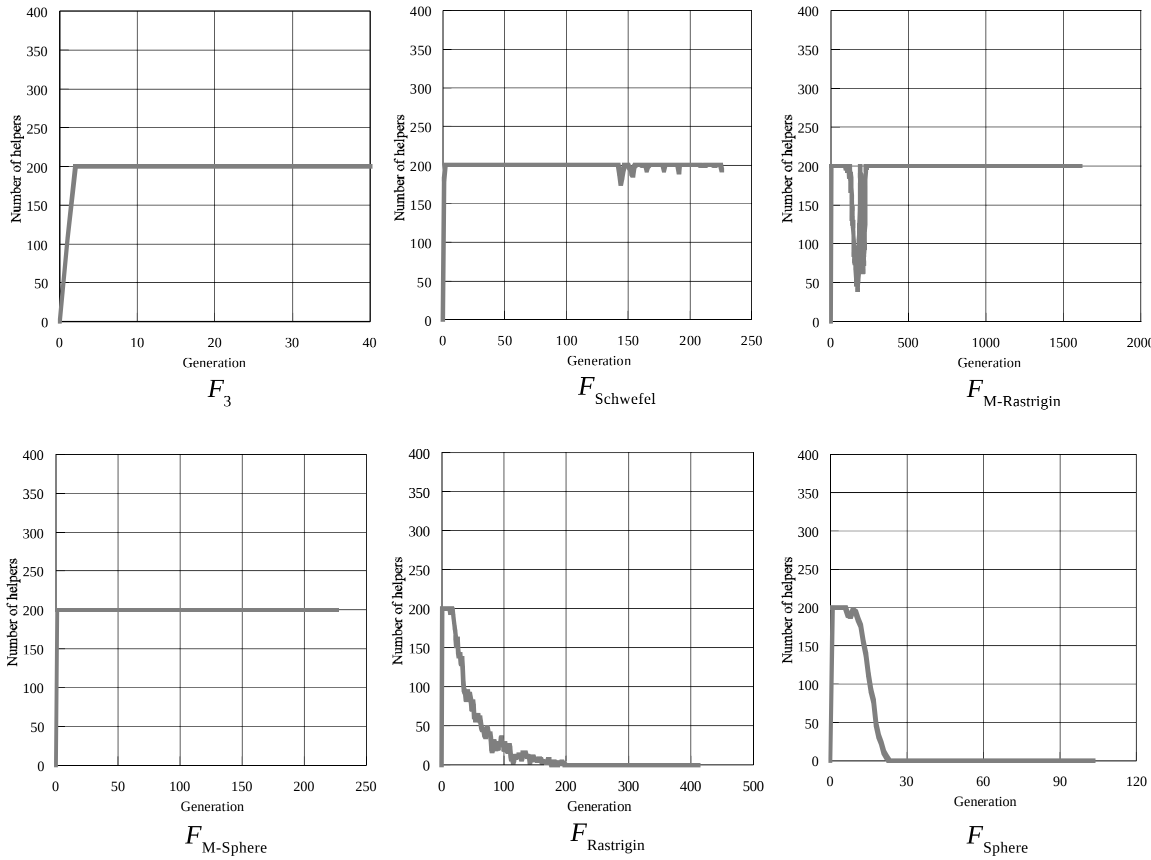


Fig. 9 Typical changes in number of helper individuals with the BES/SD/A for type 3 sampling and k_a = 2

5. Future Work

Although results in Section 4 showed usefulness of proposed boundary extension methods in real code GAs, there are many opportunities for further research related to the proposed technique.

(1) Determine adequate virtual region and helper individual sizing

In this study, we used r_e value of 0.3 to detemine the virtual region and r_h value of 0.5 to determine maximum number of helper individuals. A theoretical analysis to determine the values those parameters remaines to be investigated. To make clear the relationship between these values and dimension of search space also must be investigated. Also to devise a self adaptation scheme of these values in accordance with evolution is also desirable to realize.

(2) Determine what problems are good test functions for this sort of methods

We used several test functions which are commonly used in the literature. The amount of effectiveness of applying this sort of methods was dependent on these test functions. Thus, we need to study what problems are boundary extension hard or what problems are easy. The effectiveness might be tightly related with the fitness landscape of the problems at or near the boundary.

(3) Develop other types of heuristics for special selection

Although we applied the shortest distance method to select helper individuals, using other heuristics also remains for future work, including to combine sharing technique [Goldberg 87] for multi-optimum problems.

(4) Investigate other generational models and crossover operators

In this study, we used a simple (μ+λ)-ES like generational model and BLX-α crossover operators. Results with other generational models and other crossover operators are expected to be studied to see the generality of the proposed approaches.

(5) Apply BEMs to more general constrained problems

Boundary extension methods can be seen as constrained problems. The advantage of the BES 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].

6. 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 three types of BLX-α operators, i.e., type 1 sampling , type 2 sampling, and type 3 sampling, where type 3 sampling has smallest sampling bias in the three samplings.

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 k_a value of 2 had good performance on functions which have their optimum at or near the boundaries of the search space. Since each helper individuals 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.

Finally, as the remaining challenges discussed in Section 5 are solved, boundary extension methods are ready for application in real-world problems and further this approach may be applicable to more general function optimization, such as constrained parameter optimization.

Acknowledgements

Professor Goldberg's contribution to the work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, the U. S. Army, or the U.S. Government.

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 87] D. E. Goldberg and J. Richardson: Genetic Algorithms with Sharing for Multimodal Function Optimization, *Proceedings of the 2nd International Conference on Genetic Algorithms*, Lawrence Erlbaum, Hillsdale, NJ, pp.41-49, 1987.

[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).

[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).

[Qi 94a] Qi, A and Palmieri, F: Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space Part I: Basic properties of selection and mutation, IEEE Trans. on Neural Networks, Vol. 5, No. 1, pp. 102-119 (1994).

[Qi 94b] Qi, A and Palmieri, F: Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space Part II: Analysis of the diversification role of crossover, IEEE Trans. on Neural Networks, Vol. 5, No. 1, pp. 120-129 (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).

[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 00] 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).