Robust Evolutionary Algorithms with Toroidal Search Space Conversion for Function Optimization

Hiroshi Someya
Information Science Center,
Nagasaki University,
1-14 Bunkyo Nagasaki 852-8521 Japan.
someya@net.nagasaki-u.ac.jp

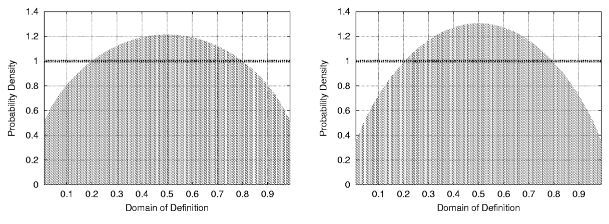
Masayuki Yamamura
Tokyo Institute of Technology,
4259 Nagatsuta Midori-ku
Yokohama 226-8502 Japan.
my@dis.titech.ac.jp

Abstract

This paper presents a new method that improves robustness of Real-Coded Evolutionary Algorithms (RCEAs), such as Real-Coded Genetic Algorithms and Evolution Strategies, for function optimization. It is reported that most crossover (or recombination) operators for RCEAs has sampling bias that prevents to find the optimum near the boundary of search space. They like to search the center of search space much more than the other. Therefore, they will not work on functions that have their optima near the boundary of the search space. Although several methods have been proposed to reduce this sampling bias, they could not cancel the whole bias. In this paper, we propose a new method, *Toroidal Search Space Conversion* (TSC), to remove this sampling bias. TSC converts bounded search space into toroidal one with no parameters. Experimental results show that a RCEA with TSC has higher performance to find the optimum near the boundary of search space and it has improved robustness concerning the relative position of the optimum.

1 INTRODUCTION

Function optimization is one of the most important optimization problems. Several Real-Coded Evolutionary Algorithms (RCEAs) such as Real-Coded Genetic Algorithms and Evolution Strategies, which use the real number vector representation, have been proposed [3–6, 9, 10, 12, 15, 19] and they have shown higher performance than EAs using binary or gray representation [3, 5, 7]. In RCEAs, generally, initial individuals are placed in the search space uniformly. In this case, most crossover operators, such as BLX-α [5], Unimodal Normal Distribution Crossover (UNDX) [10], Center of Mass Crossover (CMX) [17], Simplex Crossover (SPX) [19], like to search the center of search space much more than the other [1, 4, 9, 18]. This bias is called "Sampling Bias" [4, 18]. Fig.1 explains the sampling biases of BLX-α and UNDX. The horizontal axis is domain of definition. The vertical axis is theoretical probability density of generating children when a crossover produces them from a pair of parents, chosen out of the population that is distributed in $0 \sim 1$ uniformly. When a crossover operator has such bias, it will not work on functions whose optima are near the boundary of the search space.



(a) BLX-α (b) UNDX

Figure 1: The sampling biases of BLX-α and UNDX

The sampling bias grows exponentially stronger as the dimension of search space. Therefore, in case objective function is high dimensional and its optima are in the corner of the search space, RCEAs like to be trapped at a local minimum located around the center of the search space. Recently, RCEAs have been applied to real-applications [11, 13, 16]. In real-applications, since we cannot know where are the optima, robust RCEAs considering the sampling bias are needed.

The purpose of this paper is to present a method that cancel the sampling bias to improve robustness of RCEAs. In the next section, we briefly review several major methods that reduce the sampling bias and discuss their features. We propose a new method, *Toroidal Search Space Conversion*, in section three and reflect on the computational complexity in section four. Empirical verification is performed in section five. In the last section, we conclude this paper.

2 RELATED WORKS

2.1 Existing Methods and Their Features

Several methods, such as Boundary mutation [7], (UX, UNDX)+EMGG [9], boundary extension by mirroring (BEM) and boundary extension with extended selection (BES) [18], have been proposed. Boundary mutation produces individuals on boundary of search space. (UX, UNDX)+EMGG improves the sampling bias of UNDX using Uniform Crossover (UX) [3]. This method selects either UNDX or UX, they complement their searching region each other, as the crossover operator dynamically. BEM and BES extend the search space in order to move the relative position of the optimum toward the center of the search space. They allow individuals to be located outside the search space. The individuals are called “virtual individuals”. The details of BEM are introduced in section 2.2. In BES, the number of the virtual individuals is limited by helper individual rate and no functional value of the virtual individuals is used. We can mention that these methods have the following three disadvantages.

(1) Dependence on Search Operator: In (UX, UNDX)+EMGG, UX and UNDX complement their searching region each other. However, when we use another crossover operator as the one of the search operators, we must invent or find the other one that has complementary characteristics to the first one.

(2) Parameter Tuning: All methods introduced in this section have at least one parameter, such as the mutation rate of boundary mutation, the initial probability of applying UNDX of (UX, UNDX)+EMGG, the extension rate of BEM and the helper individual rate of BES, to control how much the sampling bias is reduced. Although we cannot know the positions of the optima and the landscape of the search space, we must tune the parameters before search.

(3) Remaining the Sampling Bias: Although all methods shown in this section succeed in reducing the sampling bias, they cannot remove it. From the viewpoint of robustness, no sampling bias is desirable.

Next, we show BEM in detail because we believe it is the best method in the existing methods. It is independent on the search operator. The number of its parameters is only one. The effectiveness is relatively high.

2.2 BEM [18]

BEM aims to shift the optimum located in the corner toward the center. In BEM, individuals are allowed to be located beyond the boundary of search space. The functional value of individual i with real vector $\vec{X}^{(i)} = (x_1^{(i)}, \dots, x_n^{(i)})$ is calculated as follows:

$$\begin{aligned} f(\vec{X}^{(i)}) &= f(\vec{Y}^{(i)}), \\ \vec{Y}^{(i)} &= (y_1^{(i)}, \dots, y_n^{(i)}), \\ y_j^{(i)} &= \begin{cases} 2\,\min_j - x_j^{(i)} & : \text{ if } x_j < \min_j \\ 2\,\max_j - x_j^{(i)} & : \text{ if } x_j > \max_j \\ x_j^{(i)} & : \text{ otherwise,} \end{cases} \end{aligned} \tag{1}$$

where, $\min_j$ and $\max_j$ are the lower and upper limits of parameter range on the j-th dimension of the original search space respectively. BEM has one parameter, r_e $(0 < r_e < 1)$, that controls how much search space is extended. The parameter range of the extended search space is $l_j(1 + r_e)$ when that of the original one is l_j. The initial individuals are placed in the original search space uniformly.

3 TOROIDAL SEARCH SPACE CONVERSION (TSC)

TSC converts search space with boundary into toroidal one. This conversion is performed as follows:

step1 Extend the search space to the extended search space like BEM with r_e=1.0, (see section 2.2)

step2 Connect each e-max$_j$ of the extended search space to corresponding e-min$_j$.

where, e-min$_j$ and e-max$_j$ are the lower and upper limits of parameter range on the j-th dimension of the extended search space respectively. An example of the converted search space is shown in Fig.2. The converted search space becomes torus. In this converted search space, the crossover operation is performed as the following pseudo-codes (like *C++*):

```
choose k required parents;
for (int i=1; i<k; i++){
  make pow(2, n)-1 clones of parent_i;
  // ---- n is the dimension
  select the clone whose distance from
    parent_0 is the shortest out of the
    clones and parent_i;
}
do crossover using parent_0 and the
  k-1 selected clones;
```

Fig.3 shows an example of a crossover in a converted search space. First, three clones (clone_1, `clone_1*, clone_1**`) at the corresponding points on the virtual search space are copied from `parent_1`. Next, `clone_1` is allowed to join the crossover operation because its distance from `parent_0` is shortest. Thus, the crossover operation, using UNDX as the crossover operator, searches in the gray region.

For implementation on a computer program, the procedures in the above `for` are described as follows:

```
clone_i = parent_i;
// ---- copy the parent_i vector to clone
for (int j=0; j<n; j++){
  const double distance
    = clone_i[j] - parent_0[j];
  if (fabs(distance) > l){
  // ---- l is the half width of the
  //        extended search space
    if (distance >= 0){ clone_i[j] -= 2l; }
    else              { clone_i[j] += 2l; }
  }
}
```

Although the volume of the search space grows exponentially, the increase of the computational cost for this crossover is only linear, $O(k \times n)$. Since the converted search space is torus, a generated individual i, $\vec{X}^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)})$, is modified as follows:

$$\begin{aligned} \vec{X}^{(i)} &= \vec{Z}^{(i)}, \qquad (2) \\ \vec{Z}^{(i)} &= (z_1^{(i)}, \ldots, z_n^{(i)}), \\ z_j^{(i)} &= \begin{cases} x_j^{(i)} + 2l & : \text{ if } x_j < \text{e-min}_j \\ x_j^{(i)} - 2l & : \text{ if } x_j > \text{e-max}_j \\ x_j^{(i)} & : \text{ otherwise.} \end{cases} \end{aligned}$$

For example, in Fig.2, when A and B are generated by a crossover operation, they are modified as A' and B' respectively. Using this modification, when the distance between parents is far, crossover does not generate children in the center of the search space, but does them near the boundary close to the parents (in the gray region in Fig.3). In TSC, initial individuals are placed in the extended search space uniformly. Accordingly, by this proposed method, any position on this search space become equivalent to any others.

TSC clears the three disadvantages of the existing methods. Since TSC is a conversion method, it is independent on any search operator. TSC has no parameter. The converted search space has no sampling bias when the initial individuals are placed in the extended search space uniformly because it is torus.

TSC has one more significant feature. The converted search space maintains global continuity of landscape.

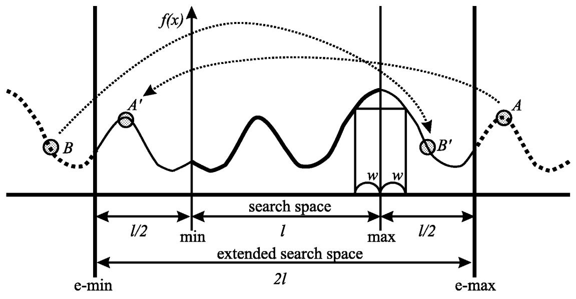


Figure 2: An example of 1-dimensional converted search space by TSC

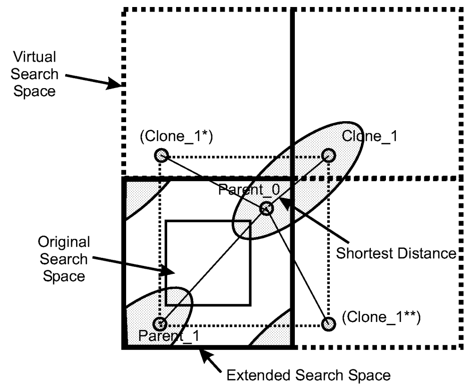


Figure 3: An example of crossover procedure, using UNDX as the crossover operator, on a 2-dimensional converted search space by TSC.

The "global continuity of landscape" means that individuals around an individual have approximate equivalent functional value. In [16], the authors use a method that connects $\min_j$ and $\max_j$ when the coded vector represents an angle. In this method, children are produced only in the supplementary angle region because -180 degrees correspond to 180 degrees. When we apply this method to a search space that does not have such characteristic, the global continuity of landscape should be lost because $f(x_1, \ldots, \min_j, \ldots, x_n)$ will be different from $f(x_1, \ldots, \max_j, \ldots, x_n)$. Since EAs assume that search space has the global continuity of landscape [8], the global continuity of landscape should be maintained. In TSC, it is satisfied because e-min$_j$ corresponds to e-max$_j$, even if the original search space does not have the above characteristic.

4 COMPUTATIONAL COMPLEXITY

Let discuss the number of samplings required to find the optimum in a n-dimensional search space whose volume is D, as shown in Fig.4. First, we discuss an EA without any selection mechanism. Then, we consider an EA equipped with a selection mechanism.

Table 1: The test functions

function	equation (n specifies the dimension)	mul.^{*1}	disc.^{*2}	domain	d_i
Sphere	$\sum_{i=1}^{n} x_i^2$	no	no	[-5.12+d_i, 5.12+d_i]	0.0, 1.5, 3.0, 4.5
Step	$\sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2$	no	strong	[-5.12+d_i, 5.12+d_i]	0.0, 1.5, 3.0, 4.5
Schwefel	$418.9828873n + \sum_{i=1}^{n} x_i \sin\sqrt{\lvert x_i \rvert}$	low	no	[-512, 512]	-
Rastrigin	$10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	high	no	[-5.12+d_i, 5.12+d_i]	0.0, 1.5, 3.0, 4.5
Griewangk	$\frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	high	no	[-512+d_i, 512+d_i]	0, 150, 300, 450

$*1$: multi-modality, $*2$: discontinuity

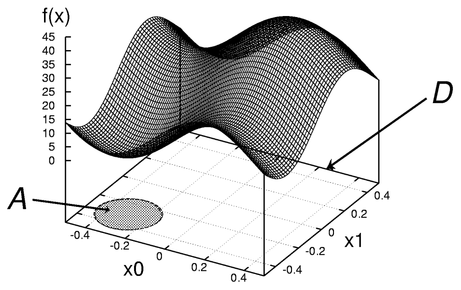


Figure 4: n-dimensional objective function, in which D is its volume and A is the volume of the region as the optimum.

4.1 Without Selection

In case BLX-α is used as the search operator, the probability density curve of generating children, shown in Fig.1 (a), is expressed as follows (the details of BLX-α and $g(x)$ are shown in Appendix A and B respectively):

$$g(x) = \frac{2\{\ln\frac{3}{2} + (x-1)\ln(1-x) - x\ln x\}}{2\ln\frac{3}{2} + 1} . \quad (3)$$

Therefore, when n is 1, the number of samplings required to find the optimum that is located at the corner of the search space is $\frac{1}{g(0)} = \frac{1}{g(1)} \simeq 2.233$ times as many as in case Uniform Random Search (URS) [20], which searches in the domain uniformly, is used. When n is 10 or 20, $\frac{1}{g(0)^n}$ is about 3,000 or 9,500,000, respectively. In case another crossover operator whose sampling bias is stronger than BLX-α, such as UNDX, is used, more samplings are required. The probability to find the optimum, P, when URS is used as the search operator is expressed as follows [20]:

$$P = 1 - \left(1 - \frac{A}{D}\right)^m , \quad (4)$$

where m is the number of samplings and A is the volume of the region as the optimum. When the search space is converted by TSC, since there is no sampling bias even if the search operator is BLX-α, the search works like URS. In this case the probability to find the optimum is equivalent to URS as $\frac{2^n A}{2^n D} = \frac{A}{D}$.

4.2 With Selection

When we consider selection mechanism, the complexity of landscape is important. Unimodal function is often converted into multimodal one by TSC. Generally, optimization of multimodal function is more difficult than that of unimodal one. Moreover TSC converts multimodal function into more complex multimodal one in which the number of local minima is exponentially larger. It has not been cleared that the relation between complexity of landscape and the difficulty of optimization for EAs. However, it has been known that big hill including local minima influences the effectiveness of EAs.

5 EXPERIMENTS

In order to confirm the robustness of EAs in converted search space by TSC, we perform experiments.

5.1 Test Functions

How test functions should be selected has been mentioned in [2]. The five functions in Table 1 are selected under the recommendations. The optimum of Schwefel function, $f(-420.968746, \ldots, -420.968746) = 0$, and those of the others, $f(0, \ldots, 0) = 0$, are located in the corner and at the center of the search space, respectively. To achieve the purpose of these experiments, the relative positions of the optima in their search space are moved by d_i except that of Schwefel function[1], as shown in Fig.5.

5.2 Experimental Conditions

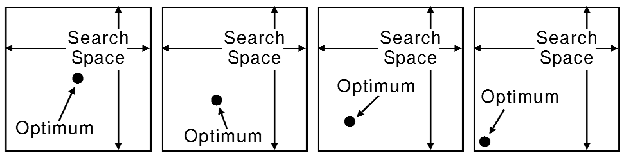
We select UNDX+MGG [10,14] as the performed EA. It has been reported that UNDX has strong sampling

[1]In Schwefel function, when the domain is changed by d_i, the optimum will be changed.

Table 2: The experimental results (#OPT)

		NoExt				BEM				BEMe				TSC			
function	n	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45
Sphere	50	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	100	100	100	100	100	100	100	**99**	100	100	100	**98**	100	100	100	100	100
	150	100	100	100	100	100	100	**98**	100	100	100	**96**	100	100	100	100	100
Step	30	100	96	78	**62**	100	96	**80**	93	100	95	**86**	97	**92**	96	99	100
	40	99	56	27	**11**	95	57	**35**	56	96	64	**28**	54	62	**52**	80	99
	50	81	17	**7**	10	75	19	**8**	12	88	17	**7**	12	**6**	16	35	83
Schwefel	5	87				100				99				100			
	10	8				57				56				100			
	15	0				1				1				98			
Rastrigin	4	100	99	94	**26**	100	100	99	**86**	100	97	98	**87**	100	100	100	100
	6	100	67	20	**1**	99	77	57	**13**	98	84	80	**8**	96	79	**44**	100
	8	84	27	3	**0**	91	26	10	**0**	79	34	22	**1**	79	35	**5**	97
Griewangk	30	80	68	69	**65**	70	71	74	**65**	68	76	72	**64**	65	66	**63**	73
	40	71	74	**63**	70	**65**	67	75	73	**66**	67	69	76	66	**60**	66	64
	50	**69**	73	77	70	76	74	**70**	73	**61**	69	65	68	63	**58**	60	62

$*$ 00, 15, 30, 45 under the method names specify d_i. For example, 15 means $d_i = 1.5$ or $d_i = 150$.



$d_i = 0$ $d_i = 1.5$ or 150 $d_i = 3.0$ or 300 $d_i = 4.5$ or 450

Figure 5: The relative positions, caused by d_i, of the optimum in a search space

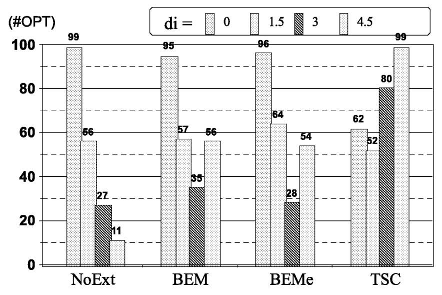
bias as shown in Fig.1 (b) and UNDX+MGG does not work well in a search space whose optimum is in the corner, such as Schwefel function [9]. The details of this EA are shown in Appendix A and C. No mutation is used for focusing on the sampling bias caused by crossover. The population size is set to be 30 for unimodal functions but 100 for multimodal ones. Fifty children are produced in each generation.

TSC is compared to "No Extension method (NoExt)", BEM and BEMe. NoExt means that the EA is performed in the original search space. BEMe is introduced to be fair in our comparison. BEM and BEMe are the same except that BEMe places initial individuals like TSC. TSC places initial individuals in the extended search space, but BEM does them in the original one. The r_e of BEM and BEMe are set to be 0.25 because the value has been used in [19]. In all experiments except TSC, when an individual is generated outside their search space, the crossover retries to generate another inside. Each experiment is performed 100 trials. Each run continues until the optimum is found or the number of evaluation reaches a constant that was set to be enough large number determined in pilot study. The performance measure is the numbers of runs in which the method succeeded in finding the global optimum (#OPT). The robustness of each method is evaluated through the lowest performance in all cases of d_i.

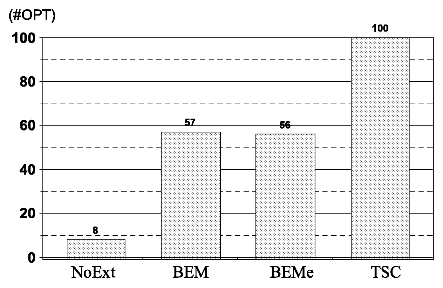
5.3 Results and Discussion

The experimental results are shown in Table 2. Several results that explain the features of the methods obviously are shown in Fig. 6.

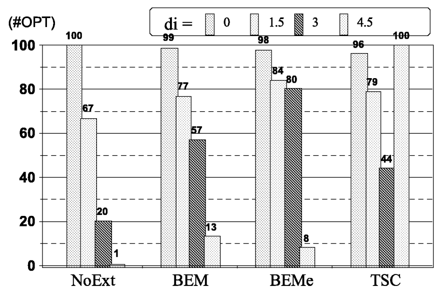
In Sphere function, the all #OPTs are approximately 100. We believe that the optimization in the converted search space by TSC has not become more difficult, because it has had no local minimum although it has become multimodal. Fig.6 (a) and (c) show the robustness of the EA performed in the converted search space by TSC. In the original search space (NoExt), the performance when $d_i = 4.5$ is terrible. We believe that this is caused by the sampling bias. You might consider why the #OPTs are different among the d_i despite no sampling bias when TSC is used. Note, the landscapes are different among the d_i although the equations are the same. In Schwefel function, which has the optimum in the corner of the search space, we can confirm that the performance of the EA is extremely improved by TSC. In Griewangk function, all methods show the robustness as shown in Fig.6 (d). From Table 1, the characteristic of this function seems to be the same as that of Rastrigin function. However, the landscape of this function is similar to that of Sphere function on the broad level, as shown in Fig.7. We believe that this robustness is caused by this similarity. In 50-dimensional Step function and 8-dimensional Rastrigin function, the difference of the effectiveness among the methods is little. Hence, we



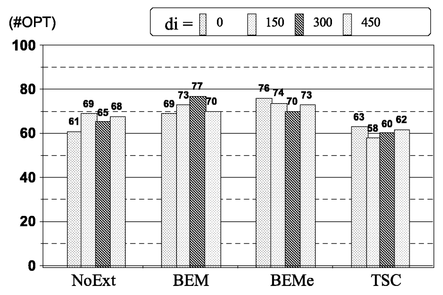
(a) 40-dimensional Step function



(b) 10-dimensional Schwefel function



(c) 6-dimensional Rastrigin function



(d) 50-dimensional Griewangk function

Figure 6: The experimental results (#OPT) that explain the features of the methods obviously

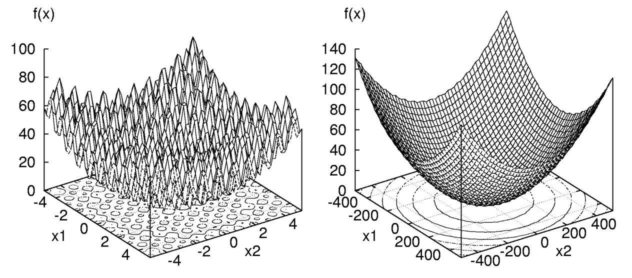


Figure 7: Rastrigin function (left) and Griewangk function (right)

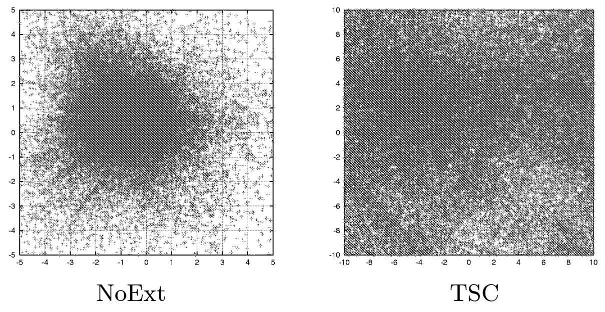


Figure 8: The distribution of the overall generated individuals in the function, $f(X) = 1$

studied the average, the worst and the variance of the runs. The statistics have shown that TSC works better than the others.

The stability of convergence speed when TSC is used is the lowest than that when the others are used. The EA performed in the converted search space by TSC can find the optimum located in the corner of the search space rapidly. However, when the optimum is located at the other positions, the convergence velocity is slower. It is a disadvantageous feature of TSC.

5.4 Confirmation of No Sampling Bias

We perform experimental confirmation of no sampling bias in the converted search space by TSC. $f(\vec{X}) = 1$ whose dimension is two is used as the objective function. The domain of definition is $[-5.0, 5.0]$. The number of evaluation is 5.0×10^4. The population size is set to be 100. The other conditions are the same as the previous experiments. Since MGG performs random sampling when all individuals have the same fitness value, the all region should be searched equally if there is no sampling bias. We plot the distribution of the overall individuals generated in a run. Fig.8 shows the results of NoExt and TSC. Although near boundary in the left figure is hardly searched, all region in the right figure are searched equally. We can confirm that there is no sampling bias.

6 CONCLUSIONS

This paper proposed a new method, *Toroidal Search Space Conversion* (TSC), which converts search space with boundary into toroidal one, to improve the robustness of RCEAs. Experimental results showed that the effectiveness of TSC is greater than those of the other methods. TSC has following three advantages: 1. TSC can be applied widely because it is independent on search operator., 2. It is easy to apply TSC because it has no parameter., 3. There is no sampling bias in the converted search space by TSC. On the other hand, TSC has one disadvantage. The landscape of the converted search space by TSC is often more complex than that of the original search space. The variance of the convergence velocity is also caused by this complexity. To cope with this disadvantageous feature is future work.

References

[1] P. J. Angeline. Using Selection to Improve Particle Swarm Optimization. In *Proc. of the ICEC'98*, pages 84–89, 1998.

[2] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, 1996.

[3] L. Davis. *The Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1990.

[4] L. J. Eshelman, K. E. Mathias, and J. D. Schaffer. Crossover Operator Biases: Exploiting the Population Distribution. In *Proc. of the 7th ICGA*, pages 354–361, 1997.

[5] L. J. Eshelman and J. D. Schaffer. *Foundations of Genetic Algorithms 2*, chapter Real-Coded Genetic Algorithms and Interval-Schemata, pages 187–202. Morgan Kaufman, 1993.

[6] H. Kita, I. Ono, and S. Kobayashi. Multi parental Extension of the Unimodal Normal Distribution Crossover for Real-Coded Genetic Algorithms. In *Proc. of the CEC'99*, pages 1581–1587, July 1999.

[7] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Program.* Springer, third edition, 1996.

[8] M. Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, 1996.

[9] I. Ono, H. Kita, and S. Kobayashi. A Robust Real-coded Genetic Algorithm using Unimodal Normal Distribution Crossover Augmented by Uniform Crossover : Effects of self-Adaptation of Crossover Probabilities. In *Proc. of the GECCO-99*, pages 496–503, July 1999.

[10] I. Ono and S. Kobayashi. A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover. In *Proc. of the 7th ICGA*, pages 246–253, 1997.

[11] I. Ono, Y. Tatsuzawa, S. Kobayashi, and K. Yoshida. Designing Lens Systems Taking Account of Glass Selection by Real-coded Genetic Algorithms. In *Proceedings of 1999 IEEE International Conference on Systems, Man and Cybernetics*, pages III–592–597, 1999.

[12] I. Ono, M. Yamamura, and S. Kobayashi. A Genetic Algorithm with Characteristic Preservation for Function Optimization. In *Proceedings of IIZUKA'96*, pages 511–514, 1996.

[13] S-J. Park and M. Yamamura. An Approach to Structural Alignment with Genetic Algorithm. In *Proceedings of the Second International Conference on Bioinformatics of Genome Regulation and Structure*, pages 201–203, 2000.

[14] H. Satoh, M. Yamamura, and S. Kobayashi. Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation. In *Proceedings of IIZUKA'96*, pages 494–497, 1996.

[15] H. Someya and M. Yamamura. Where should Children be Generated by Crossover Operator on Function Optimization ? In *Proc. of the GECCO-2000*, page 382, July 2000.

[16] O. Tomobe, I. Ono, and S. Kobayashi. Experimental Study on Determination of Protein three dimensional Structure using Genetic Algorithm (in Japanese). In *Proceedings of 25th SICE Symposium on Intelligent Systems*, pages 35–40. The Society of Instrument and Control Engineers, 1998.

[17] S. Tsutsui. Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring. In *Proc. of the PPSN V*, pages 428–437, 1998.

[18] S. Tsutsui and D. E. Goldberg. Search Space Boundary Extension Method in Real-Coded Genetic Algorithms. *Information Sciences*, 133(3-4):229–247, 2001.

[19] S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms. In *Proc. of the GECCO-99*, pages 657–664, July 1999.

[20] A. A. Zhigljavsky. *Theory of Global Random Search*, volume 65 of *Mathematics and Its Applications.* Kluwer Academic Publishers, 1991.

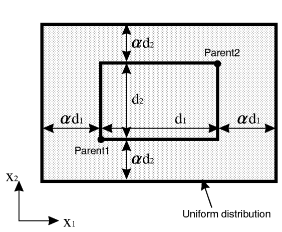
APPENDIX

A BLX-α [5] and UNDX [10]

BLX-α produces a child in the gray region in Fig.9 (left) randomly. The child vector, $\vec{C}$, which is encoded by real number vector, is determined as follows:

$$\begin{aligned}\vec{C} &= \{c_1, \ldots, c_n\} \,, \\ c_i &= u(\min(p_{1i}, p_{2i}) - \alpha d_i,\ \max(p_{1i}, p_{2i}) + \alpha d_i) \,,\end{aligned}$$

where $\vec{P}_1 = \{p_{11}, \ldots, p_{1n}\}$ and $\vec{P}_2 = \{p_{21}, \ldots, p_{2n}\}$ are parent vectors of Parent1 and Parent2 respectively. $d_i = |p_{1i} - p_{2i}|$. n is the dimension of the objective function. $u(x, y)$ is the uniform random number selected from $[x, y]$.



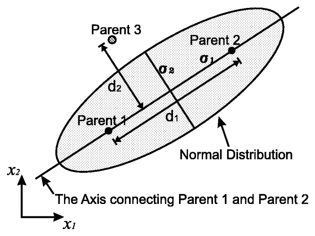


Figure 9: BLX-α (left) and UNDX (right)

UNDX generates two children around their parents using the normal distribution whose standard deviation is determined by the third parent, Parent3, as shown in Fig.9 (right). The children vectors, $\vec{C}_1$ and $\vec{C}_2$, are determined as follows:

$$\begin{aligned}\vec{C}_1 &= \vec{m} + z_1\vec{e}_1 + \sum_{k=2}^{n} z_k\vec{e}_k \ ,\\ \vec{C}_2 &= \vec{m} - z_1\vec{e}_1 - \sum_{k=2}^{n} z_k\vec{e}_k \ ,\end{aligned}$$

where $\vec{m} = (\vec{P}_1 + \vec{P}_2)/2$. $\vec{e}_1 = (\vec{P}_2 - \vec{P}_1)/|\vec{P}_2 - \vec{P}_1|$, $\vec{e}_k (k = 2, \ldots, n)$ are the orthogonal unit vectors. $z_1 \sim N(0, \sigma_1^2)$ and $z_k \sim N(0, \sigma_2^2)(k = 2, \ldots, n)$ are normally distributed random numbers, where $\sigma_1 = \alpha d_1$ and $\sigma_2 = \beta d_2/\sqrt{n}$. d_1 is the distance between Parent1 and Parent2. d_2 is the distance of the Parent3 from the line connecting Parent1 and Parent2. α and β are constants.

B EQUATION (3)

B.1 Variables

In this section, we use the following variables:

y, z	the positions of parents ($0 < y < z < 1$)
w	the width in which BLX-α produces children
c	the center of parents
$n(y, z)$	the probability of generating children in one crossover operation

α is set to be 0.5, which is recommended value.

B.2 Equation $g(x)$

After the definition of BLX-α, a child is produced in the range of x that satisfies the following inequality.

$$c - \frac{w}{2} < x < c + \frac{w}{2} \ .$$

Substitute $w = (z - y)/\alpha = 2(z - y)$ and $c = (z + y)/2$ into the above inequality,

$$-1 < \frac{2x - (z + y)}{2(z - y)} < 1 \ .$$

Therefore,

$$\begin{aligned}3y &< 2x + z \ ,\\ y &< 3z - 2x \ .\end{aligned}$$

Since the domain of x is $[0.0, 1.0]$, $g(x) = k\{g_A(x) + g_B(x) + g_C(x)\}$, as follows:

$$\begin{aligned}g_A(x) &= \int_x^{\frac{1}{3}+\frac{2}{3}x} \int_{3y-2x}^{1} n(y, z)\ dzdy \ ,\\ g_B(x) &= \int_0^x \int_x^1 n(y, z)\ dzdy \ ,\\ g_C(x) &= \int_0^x \int_{\frac{1}{3}y+\frac{2}{3}x}^{x} n(y, z)\ dzdy \ ,\end{aligned}$$

where $n(y, z) = \frac{1}{w} = \frac{1}{2(z-y)}$. Integrate the above,

$$\begin{aligned}g(x) &= k\left\{\frac{(1 - x)(\ln 3 - \ln 2)}{2}\right.\\ &\quad + \frac{(x - 1)\ln(1 - x) - x\ln x}{2}\\ &\quad + \left.\frac{x(\ln 3 - \ln 2)}{2}\right\}\\ &= \frac{k}{2}\{\ln 3 - \ln 2 + (x - 1)\ln(1 - x) - x\ln x\} \ .\end{aligned}$$

In order to satisfy $\int_0^1 g(x)\ dx = 1$, $k = \frac{4}{2(\ln 3 - \ln 2)+1}$. Hence,

$$g(x) = \frac{2\{\ln\frac{3}{2} + (x - 1)\ln(1 - x) - x\ln x\}}{2\ln\frac{3}{2} + 1} \ .$$

C MGG [14]

MGG is a generation-alternation model. It is described as follows:

step1 Generate an initial population randomly.
step2 Choose a pair of individuals as parents from the population randomly.
step3 Generate a certain number of children by a crossover.
step4 Select the best individual out of the family, the parents and the children.
step5 Choose an individual except the best, selected at *step4*, out of the family randomly according to fitness-based (or ranked-based) wheel selection.
step6 replace the two individuals, selected at *step4* and *step5*, to the parents.
step7 Iterate *step2* $\sim$ *step6* until certain condition is satisfied.