Locus-Shift Operator for Function Optimization in Genetic Algorithms

Hiroshi Inazawa[*]

Center for Education in Information Systems,
Kobe Shoin Women's University,
1-2-1 Shinohara Obanoyama, Nada, Kobe 657-0015, Japan

Kazuhisa Kitakaze

Department of Information Systems,
Tokyo University of Information Sciences,
1200-2 Yato Wakaba, Chiba 265-8501, Japan

Function optimization is the most important context for studying genetic algorithm (GA) operators. In this paper a new GA operator is introduced which greatly improves several well-known benchmark functions used in function optimization. The new operator cuts a circular chromosome at any locus selected randomly. By this operation, various types of linear chromosomes can be formed from a parent circular chromosome. The new operator is called locus-shift (LS) because the locus of the linear chromosome produced by LS almost always shifts. In this paper, we study the dynamics of the evolution of chromosomes by the LS in simulation and show the effects of LS by various benchmark functions.

1. Introduction

Function optimization is the most important context for studying genetic algorithm (GA) operators [1]. In this paper, we introduce a new GA operator that works on a circular chromosome and cuts it at any locus selected randomly. This operator can give various types of linear chromosomes from a parent circular chromosome. We call the operator locus-shift (LS) because the locus of the linear chromosome produced by LS almost always shifts. A similar operator, which works on plasmid, exists in nature [2]. In fact, we came up with the idea of the LS from the plasmid.

LS yields various types of gene configurations from a circular chromosome and is very similar to applying a high rate of mutation to a

---

linear chromosome. If we consider the fact that LS has a mutation feature, we can say that it plays the role of dynamic mutation since the mutation rate generally changes after every LS operation. Here we should note that the mutation rate changes randomly, but not adaptively [3–5]. There is another important feature of the LS: although the linear chromosome generated by the LS is different from the parent circular chromosome as a whole, partial configurations of genes from the parent remain locally in the linear chromosome. We believe that this will produce some interesting results since some good building blocks are conserved locally: we expect that the conserved partial configurations may be effective in building a good chromosome through subsequent application of the LS.

The GA has been established as a very strong method for finding the optimal values of functions which are otherwise difficult to find [6–15]. However, some tasks remain, such as improving search performance and deciding on the values of GA parameters. We believe that LS contributes to the context of function optimization, especially by improvements in search performance.

In this paper, we examine the dynamics of the evolution of chromosomes by LS in simulation and show the effects of LS on various benchmark functions used in function optimization. In section 2 we describe the features of LS in detail. In section 3 we provide an outline of the simulation, the benchmark functions used, and describe the results. In section 4 we summarize our findings and indicate areas for future research.

2. Locus-Shift operator

2.1 Features of the locus-shift operator

We now define the LS and illustrate its fundamental behavior. As shown in Figure 1, the initial chromosome has a circular structure consisting of binary digits, where the black segments indicate "1" and the others indicate "0". We consider the binary digits as "genes."

The LS operation cuts a chromosome at any position (locus) on the circle. After the LS operation, the chromosome becomes linear. For instance, when the chromosome breaks at the second locus, it becomes the chromosome of No. 2. In the same way, when a break occurs at the third locus, it becomes No. 3. By applying LS, a circular chromosome becomes a linear chromosome with a gene configuration that depends on the break position of the circular chromosome. Thus, we can have a linear chromosome with various configurations of genes from the parent circular chromosome. Note that partial configurations of the chromosome are preserved locally.
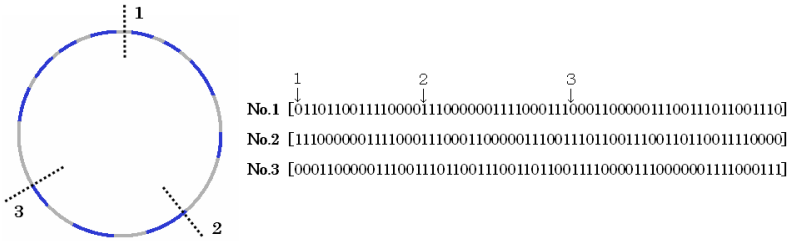
No.1 [0110110011110000111000000111100011100011000001110011011001110]

No.2 [1110000001111000111000110000011100111011001110011011001110000]

No.3 [0001100000111001110110011001101100111100001110000001111000111]

Figure 1. Illustrative image of the function of the LS operator. Chromosomes consist of binary digits and they have a circular structure. Black parts have "1" as their gene's value, the others consist of "0". When a chromosome spontaneously breaks due to LS at the second locus in the circular state, it becomes the chromosome of No. 2 on the right-hand side. In the same way, when the break points are at the first or third locus, the chromosome becomes No. 1 or No. 3.

Here are the two main features of LS.

- *Feature 1.* A circular chromosome becomes a linear chromosome having a different configuration of genes. However, partial configurations of the circular chromosome are preserved locally.

- *Feature 2.* The effect is similar to a high rate of mutation because the loci positions change as a whole in the linear state.

Now, if LS is not applied, a chromosome will always break at the first locus to interact with other linear chromosomes.

2.2 Basic behavior of the locus-shift operator

First, we examine the basic behavior of the LS in simulation. We used chromosomes with an extremely biased structure:

[111111111111111000000000000000],

where each chromosome consists of 30 genes with half of the values being "1", and the remainder "0". Note that we describe the chromosomes as linear for convenience. We prepared 100 chromosomes in the population and applied only the LS in the simulation, using 0.5 as the LS rate: $P_s = 0.5$. We calculated the relationship between the generations and the Hamming distances among chromosomes, the standard deviation of the Hamming distances, and the averaged mutation rates for each generation. Figure 2 shows the results together with the non-biased structure case, where "1" and "0" are randomly distributed in a chromosome. We can see that the Hamming distances of the biased case quickly become almost equal to the nonbiased case. Also, the dynamic
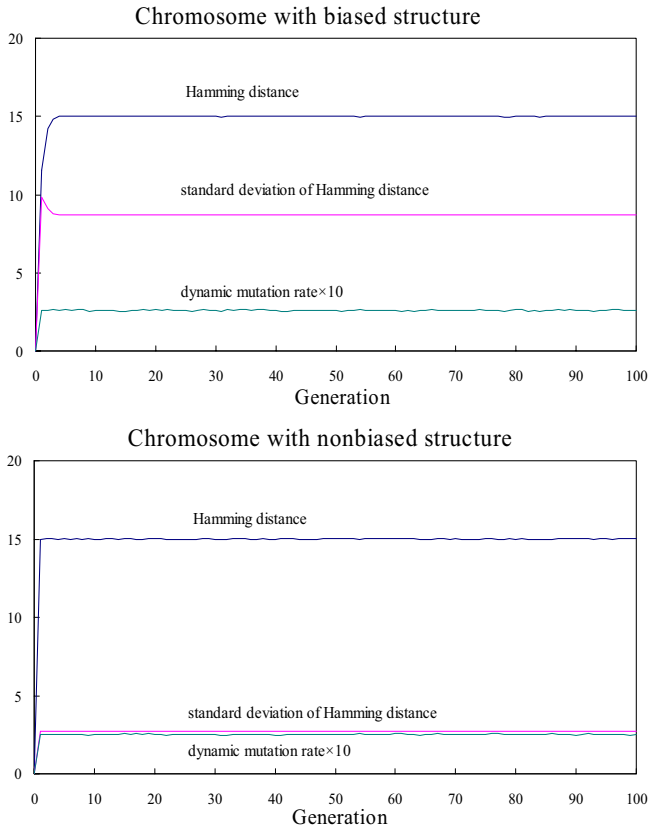
Figure 2. Relations among Hamming distances, standard deviation of the Hamming distances, dynamic mutation rate, and generations. It can be seen that the Hamming distances of the biased case quickly reaches almost the same value as the nonbiased case, where the maximum value is 15. The Hamming distances of both cases saturate to the maximum value at the same rate. The standard deviations of the biased case are a little bit high in comparison with the results of the nonbiased case. This indicates that the biased structure still remains in the population. The dynamic mutation rates are about 0.2 for both cases. Note that we show the real values times 10.

mutation rates quickly reached a definite value, which was about 0.2. We concluded that LS is able to produce randomness in chromosomes with the biased structure, which is almost the same as the nonbiased case. On the other hand, the standard deviations of the biased case are a little bit higher than the results of the nonbiased case. This indicates that the biased structure still remains in the population. However, we believe that this would be insignificant if other GA operators were applied together with LS. Also, note that if almost all of the chromosomes

in a population initially have radical structures, such as

[111111111111111111111111111111]

or

[000000000000000000000000000000],

LS is totally inoperative.

Next, we examined whether or not LS is able to build good building blocks in simulation, where we used the Royal Road function (R_1) [16–18]. R_1 is defined as:

$$R_1 = \sum_{i=1}^{n} c_i \delta_i(x), \tag{1}$$

where $n = 8$, $c_i = 8(i = 1, \ldots, 8)$. The chromosome consists of 64 genes and the fitness function is defined as $64/R_1$. Then the global optimal value of R_1 is 64:

$$\delta_i(x) = \begin{cases} 1 & \text{for } x \in s_i \\ 0 & \text{for } x \notin s_i \end{cases}, \tag{2}$$

where x is a bit string and s_i indicates a schema which consists of a block of "11111111". For example, s_1 and s_2 are described as follows:

$s_1 = $ "11111111 *

* ",

$s_2 = $ " * * * * * * * *11111111 * * * * * * * * * * * * * * * * * *

* ",

and so on, where "*" means "0" or "1". We used 0.6 as the crossover rate Pc; the inverse of L (the number of genes in a chromosome) as the mutation rate Pm; and 0.5 as Ps. The following algorithm was used in the simulation.

0. Generate an initial population: Population consists of N circular chromosomes with random gene configurations, where $N = 100$.

1. Apply the LS: Cut a chromosome at a locus selected randomly with Ps, where we suppose that every circular chromosome becomes linear.

2. Apply roulette selection: Select N chromosomes from a population in proportion to their fitness values.

3. Apply crossover: Make parents crossover with Pc to form new offspring.

4. Apply mutation: Make new offspring mutate at each locus with Pm.

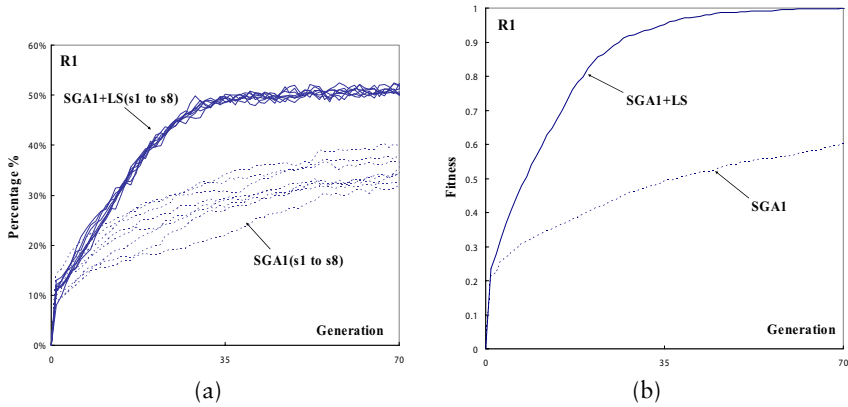5. Apply elite strategy: Conserve chromosomes with the best fitness value in the present population.

Figure 3. Existing percentage of every schema s1 to s8 in the population for the Royal Road function (R_1). The upper group (solid lines) shows the case of SGA1+LS. The lower group (dotted lines) shows the case of SGA1.

We show the results in Figure 3, where SGA1 means a simple GA with one-point crossover and SGA1+LS means SGA1 with the LS. Figure 3(a) shows the existing percentage of $s_i (i = 1, \ldots, 8)$ in the population, where the upper group (solid lines) shows SGA1+LS and the lower group shows SGA1. As we can see in this plot, SGA1+LS works very well at accumulating good schemata. Figure 3(b) shows the searching performance of SGA1+LS and SGA1. It can be seen that SGA1+LS performs better than SGA1. We believe that Feature 1 worked well because the optimal structure of R_1 consisted of a nugget of "11111111". In conclusion, LS is very effective for R_1. We can say that the LS would work very well for chromosomes with a global optimal value, consisting of parts of the same configuration as a bit string.

Furthermore, we calculated the Hamming distance between chromosomes with the best fitness in each generation to study the situation of building blocks for R_1. In addition, we studied the transition of gene values in the best chromosome from each generation. In the simulation, we calculated 100 steps, where each step began with an initial population and the procedure of 1 through 5 was iterated for 70 generations. We used the following GA parameter values: Pc = 0.6, Pm = 1/(the number of genes), and Ps = 0.5. The results are shown in Figure 4, where the Hamming distances and the number of ones in the chromosome are indicated. The values of the Hamming distance and the number of ones are averaged by 100 steps for each generation. We can see that the Hamming distance of SGA1+LS quickly decreases in comparison with SGA1, and the number of ones rapidly increases in the chromosome. This shows that a nugget "11111111" is rapidly built by LS. We conclude that these results were caused by the special
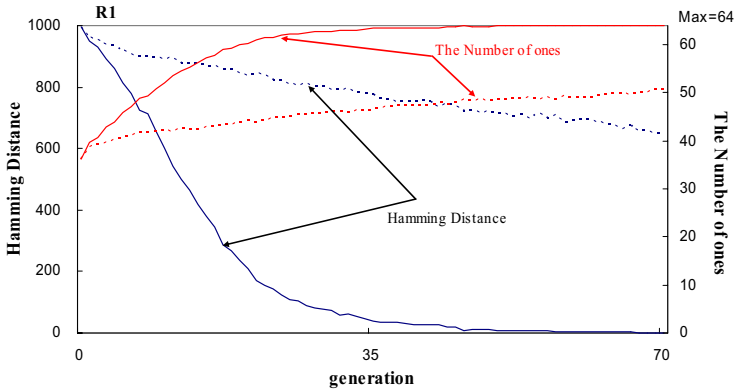
Figure 4. Hamming distances and the number of ones in the best chromosome. The solid lines indicate SGA1+LS and the dotted lines indicated SGA1. The values are averaged by 100 steps, where each step consists of 70 generations. The test function is R_1.

feature of the optimal state for R_1, which consists of all ones. Generally, we believe that LS has the potential to accumulate some good building blocks for more common tasks, especially in cases with an optimal state similar to R_1. Next, we examine the effects of LS for various famous benchmark functions.

3. Simulation

3.1 Definition of benchmark functions

We use eight popular benchmark functions for studying the performance of GAs and GA operators. They are De Jong's function [13] as F1 to F5, the Twin-Peak function [19] as F6, Griewank's function as F7, and Michalewicz's function as F8 [20, 21]. We examined the performance of the LS for these functions.

These functions and the fitness functions are described in the following, where we defined the global optimal value of each fitness function to be "1".

F1 is defined as

$$F1 = -\sum_{i=1}^{n} x_i^2, \tag{3}$$

where $n = 3$, $-5.11 \leq x_i \leq +5.12$, $\Delta x_i = 0.01$, a chromosome consists of 30 genes: L = 30. The fitness function is defined as $1/(1 - F1)$.

F2 is defined as

$$F2 = -(100(x_1^2 - x_2)^2 + (1 - x_1)^2), \tag{4}$$

where $-2.047 \le x_i \le +2.048$, $\Delta x_i = 0.001$, $L = 24$, and the fitness function is defined as $1/(1 - F2)$.

F3 is defined as

$$F3 = -\sum_{i=1}^{n} [x_i], \tag{5}$$

where "[ ]" is Gauss's symbol, and $n = 5$, $-5.11 \le x_i \le +5.12$, $\Delta x_i = 0.01$, $L = 50$, and the fitness function is defined as $1/(1 - (F3 - 30))$.

F4 is defined as

$$F4 = \sum_{i=1}^{n} i \cdot x_i^4 + \text{Gauss}(0, 1), \tag{6}$$

where $n = 30$, $-1.27 \le x_i \le +1.28$, $\Delta x_i = 0.01$, $L = 240$, and the fitness function is defined as $1/(1 + |F4|)$. Gauss$(0, 1)$ means a normal distribution with the average of "0" and the variant of "1".

F5 is defined as

$$F5 = -\left(\frac{1}{500} + \sum_{j=1}^{m} \frac{1}{j + \sum_{i=1}^{n}(x_i - a_{ij})^6} \right)^{-1}, \tag{7}$$

where $n = 2$, $m = 25$, $-65.535 \le x_i \le +65.536$, $\Delta x_i = 0.001$, $L = 34$, and the fitness function is defined as $1/(1 - (F5 + 0.998004))$. a_{ij} is defined as:

$$a_{ij} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \ldots & 32 & 32 & 32 \end{pmatrix}. \tag{8}$$

F5 has a global optimum value of -0.998004 (rounded off to $O(10^{-7})$).

F6 is defined as

$$F6 = (x_1 \wedge x_2 \wedge x_3 \ldots x_{n-1} \wedge x_n) \vee (\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3 \ldots \bar{x}_{n-1} \wedge \bar{x}_n), \tag{9}$$

where $n = 100$ and $x_1, \ldots, x_n = 1$ or 0, each boolean variable x_i is determined to be $F6 = 0$ or 1, $L = 100$, and the fitness function is defined as:

$$\max\left(\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}, \sqrt{\frac{\sum_{i=1}^{n}(1 - x_i)^2}{n}} \right). \tag{10}$$

F7 is defined as

$$F7 = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{1}} \right), \tag{11}$$

where $n = 10$, $-512 \le x_i \le +511$, $\Delta x_i = 1.0$, $L = 100$, and the fitness function is defined as $1/(1 + F7)$.

F8 is defined as

$$\text{F8} = \sum_{i=1}^{n} \sin(x_i) \cdot \sin^{2m}\left(\frac{ix_i^2}{\pi}\right), \tag{12}$$

where $n = 5$, $m = 10$, $0 \le x_i \le \pi$, $\Delta x_i = \pi/(2^{12} - 1)$, L = 60, and the fitness function is defined as $1/(1 + \text{F8} + 4.688)$.

∎ 3.2 Outline of the simulation

In the simulation, the procedure of 1 to 5 from section 2.2 is iterated until the best value of fitness is obtained, or the number of generations reaches the maximum, which was decided *a priori* to be 20000. We decided the precision of the fitness evaluation to be $O(10^{-3})$. We call each procedure a *trial* in the simulation and we had 100 trials, changing the initial population for each. We used the following values for the GA parameters: Pc = 0.6, Pm = 1/L, and Ps = 0.5.

∎ 3.3 Results

We show the results in Table 1, where AVG.G means the averaged values of the number of generations in which a global optimal value was found, or in which the maximal value of 20000, fixed *a priori*, was reached. "Successful rate" means the percentage of a global optimal value found out of 100 trials. σ means the standard deviation of AVG.G. In addition, a two-point crossover and a three-point crossover were used in SGA2 and SGA3. Moreover, we tried to calculate various cases of SGA1+LS:

1. Pall ≡ Pc ≠ Pm ≠ Ps ≠ 0,

2. Pc ≠ 0, Pm = 0, Ps ≠ 0,

3. Pc = 0, Pm ≠ 0, Ps ≠ 0, and

4. Pc = Pm = 0, Ps ≠ 0.

As shown in Table 1, the search performance of SGA1+LS is remarkably better than that of SGA1, 2, and 3 for all functions except F8. In order to find the work performed by the LS, we examined the configuration of the gene's value in the functions' good chromosomes. We found that the good chromosomes with improved results consist mainly of nuggets with the same value: "000...00" or "111...11", and so on. We found these structures for F1 through F7. We conclude that Features 1 and 2 of LS work well for these functions. However, the situation of F4 was a little different from the other functions. The good chromosome consisted mainly of many smaller nuggets with the same value: "11", "111", "1111", and "00". The common configuration of F1 through F7 is that the good chromosomes have some of the same nuggets repeated in their structure. We believe that this structure gave

Model	SGA1+LS (Pall≠0)	SGA1+LS (Pm=0)	SGA1+LS (Pc=0)	SGA1+LS (Pm=Pc=0)	SGA1	SGA2	SGA3	SGA1+LS (Pall≠0)	SGA1+LS (Pm=0)	SGA1+LS (Pc=0)	SGA1+LS (Pm=Pc=0)	SGA1	SGA2	SGA3
Test Fuction	F1							F2						
Avg. of the best fitness value	**0.999**	0.999	0.999	0.569	0.999	0.999	0.999	0.999	0.999	**0.999**	0.852	0.999	0.999	0.999
AVG.G (Successful rate)	**24.850 (100%)**	34.920 (100%)	28.970 (100%)	20000 (0%)	31.350 (100%)	35.860 (100%)	32.480 (100%)	31.73 (100%)	567.630 (99%)	**25.75 (100%)**	20000 (0%)	98.290 (100%)	85.970 (100%)	98.120 (100%)
σ	**13.210**	16.646	15.724	0.000	16.334	16.369	15.952	23.983	778.898	**20.246**	0.000	145.657	75.647	109.834
Dynamic Mutation Rate	**0.245**	0.240	0.247	0.250	/	/	/	0.249	0.247	**0.251**	0.247	/	/	/
Test Fuction	F3							F4						
Avg. of the best fitness value	1.000	**1.000**	1.000	0.103	1.000	1.000	1.000	1.000	**0.999**	0.999	0.016	1.000	1.000	0.999
AVG.G (Successful rate)	28.140 (100%)	**24.010 (100%)**	62.360 (100%)	20000 (0%)	33.950 (100%)	34.720 (100%)	35.460 (100%)	110.410 (100%)	**88.540 (100%)**	214.240 (100%)	20000 (0%)	129.570 (100%)	128.920 (100%)	119.770 (100%)
σ	7.321	**6.294**	32.400	0.000	9.252	11.384	12.050	45.308	**32.349**	66.070	0.000	41.076	41.589	34.808
Dynamic Mutation Rate	0.224	**0.223**	0.215	0.237	/	/	/	0.248	**0.248**	0.250	0.249	/	/	/
Test Fuction	F5							F6						
Avg. of the best fitness value	**1.000**	0.982	1.000	0.215	0.918	0.902	0.914	1.000	**1.000**	1.000	0.796	1.000	1.000	1.000
AVG.G (Successful rate)	**23.530 (100%)**	1215.300 (96%)	784.280 (100%)	19800 (0%)	9069.090 (51%)	10604.660 (54%)	9417.860 (40%)	1126.290 (100%)	**207.450 (100%)**	716.870 (100%)	20000 (0%)	1567.160 (100%)	1586.600 (100%)	1494.370 (100%)
σ	**105.003**	4745.891	1049.636	1989.776	9907.388	9977.038	9966.019	622.042	**41.268**	200.690	0.000	599.343	585.892	420.383
Dynamic Mutation Rate	**0.244**	0.236	0.241	0.247	/	/	/	0.199	**0.175**	0.184	0.233	/	/	/
Test Fuction	F7							F8						
Avg. of the best fitness value	**0.981**	0.880	0.866	0.017	0.889	0.880	0.885	0.999	0.999	0.985	0.348	0.999	**0.998**	0.999
AVG.G (Successful rate)	**3200.210 (83%)**	10069.060 (65%)	20000 (0%)	20000 (0%)	20000 (0%)	20000.000 (0%)	20000 (0%)	2668.890 (89%)	2070.750 (100%)	10643.430 (73%)	20000 (0%)	1795.770 (100%)	**1656.740 (100%)**	1727.110 (99%)
σ	**5858.921**	9589.565	0.000	0.000	0.000	0.000	0.000	2658.854	2398.884	7731.098	0.000	2999.309	**2550.541**	2823.449
Dynamic Mutation Rate	**0.213**	0.194	0.243	0.250	/	/	/	0.245	0.242	0.246	0.249	/	/	/

Table 1. Comparisons of SGA1+LS with various cases of Pc, Pm, Ps, SGA1, SGA2, and SGA3 for F1 to F8. SGA2 and SGA3 means SGA with two-point crossover and three-point crossover. Each optimal fitness value is 1.0. AVG.G indicates the average minimum generation number. Successful rate indicates the percentage of trials in which an optimal value is found. That is, 100% means that the operation found the optimal value in every trial. On the other hand, when the rate is 0%, the number of generations is 20000 which was set *a priori*. The results written in bold are the best AVG.G. The standard deviation σ of AVG.G and the dynamic mutation rate are also indicated.

us the good results produced by the LS, especially with Feature 1. On the other hand, the good chromosome of F8 does not have this common configuration, consisting instead of some nuggets with different types of values and the repeated structure did not hold a dominant position. Because of this, LS did not produce good results for F8.

Another result shown in Table 1 is that LS works well together with the other GA operators; the mutation and crossover operators are essential for LS. However, we can find some differences in this situation. F2 needs "LS+mutation" without crossover and F4 and F6 need "LS+crossover" without mutation. Unfortunately, we could not identify the specific properties of good chromosomes that would account for these differences. It would be necessary to study many other functions to throw light on these properties. We conclude that LS needs ordinary

GA operators to work well, and that LS is effective for good chromosomes with a repeated structure, consisting of some nuggets with the same values.

4. Conclusion and future research

In this paper, we investigated the behavior of the locus-shift (LS) operator and its effect on function optimization. We have established that LS gives excellent performance for almost all of the benchmark functions considered. Search performance of the functions was remarkably improved by the LS as compared to the results using only ordinary genetic algorithm (GA) operators. On the other hand, we found that the LS works well together with the ordinary GA operators. Moreover, we indicated that LS is effective for good chromosomes with a repeated structure, consisting of some nuggets with the same gene's value, which is produced by Features 1 and 2 of the LS. Thus, we conclude that the LS plays an important role as a fundamental operator in the field of function optimization.

It would be necessary to examine the best rates of the LS and ordinary GA operators, and the combination of these rates in order to obtain the optimal values of the benchmark functions. In addition, we would need to study the performance of the LS for other functions. We propose to conduct this research in the near future.

Acknowledgment

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).

[2] J. M. Smith, *Evolutionary Genetics* (Oxford University Press, Oxford, 1989).

[3] T. Bäck, "Self-Adaptation in Genetic Algorithms," *Proceedings of the First European Conference on Artificial Life* (MIT Press, Cambridge, 1992).

[4] T. Bäck, *The Interaction of Mutation Rate, Selection, and Self Adaptation within a Genetic Algorithm, Parallel Problem Solving from Nature 2* (North-Holland, Amsterdam, 1992).

[5] T. Bäck, "Optimal Mutation Rates in Genetic Search," *Proceedings of the Fifth International Conference on Genetic Algorithms* (Morgan Kaufmann, Publishers, San Mateo, CA 1993).

[6] J. J. Grefenstette, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications* (Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1985).

[7] J. J. Grefenstette, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms and Their Applications* (Lawrence Erlbaum Associates, Publishers, 1987).

[8] J. D. Schaffer, *Genetic Algorithms, Proceedings of the Third International Conference on Genetic Algorithms* (Morgan Kaufmann, Publishers, 1989).

[9] R. K. Belew, and L. B. Booker, *Genetic Algorithms, Proceedings of the Fourth International Conference on Genetic Algorithms* (Morgan Kaufmann, Publishers, 1991).

[10] K. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs* (Springer-Verlag, Berlin, 1992, Second extended edition, 1994).

[11] S. Forest, *Genetic Algorithms, Proceedings of the Fifth International Conference on Genetic Algorithms* (Morgan Kaufmann, Publishers, 1993).

[12] X. Yao, Y. Liu, and G. Lin, "Evolutionary Programming Made Faster," *IEEE Transactions on Evolutionary Computation*, **3**(2) (1999) 82–102.

[13] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral dissertation (University of Michigan, Ann Arbor, 1975).

[14] R. A. Caruana and J. D. Schaffer, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms," in *Proceedings of the Fifth International Conference on Machine Learning*, Los Altos, CA 1988 (Morgan Kaufmann, Publishers, 1988).

[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison Wesley, Boston, 1989).

[16] M. Mitchell, S. Forrest, and J. H. Holland, "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance," in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, edited by F. J. Varela and P. Bourgine (MIT Press, 1992).

[17] S. Forrest and M. Mitchell, "Relative Building Block Fitness and the Building Block Hypothesis," in *Foundations of Genetic Algorithms 2*, edited by L. D. Whitley (Morgan Kaufmann, Publishers, 1993).

[18] M. Mitchell, J. H. Holland, and S. Forrest, "When Will a Genetic Algorithm Outperform Hill Climbing?" in *Advances in Neural Information Processing Systems 6*, edited by J. D. Cowan, G. Tesauro, and J. Alspector (Morgan Kaufmann, Publishers, 1994).

[19] S. A. Cook, "The Complexity of Theorem-Proving Procedures," in *Proceedings of the Third Annual ACM Symposium: Theory of Computing*, 1971.

[20] K. E. Mathias and L. D. Whitley, "Changing Representations During Search: A Comparative Study of Delta Coding," *Evolutionary Computation*, **2** (1995) 249–278.

[21] The Organizing Committee: H. Bersini, M. Doringo, S. Langerman, *et al.*, *Results of the First International Contest on Evolutionary Optimization (1st ICEO)* (1996 IEEE International Conference on Evolutionary Computation ICEC'96, 1996).