Paper

Increasing Robustness of Binary-coded Genetic Algorithm

Jiangming Mao* Non-member Kotaro Hirasawa** Member
Jinglu Hu** Member Junichi Murata* Member

Genetic algorithms are often well suited for optimization problems because of their parallel searching and evolutionary ability. Crossover and mutation are believed to be the main exploration operators. In this paper, we focus on how crossover and mutation work in binary-coded genetic algorithm and investigate their effects on bit's frequency of population. According to the analysis of equilibrium of crossover, we can see the bit-based simulated crossover (BSC) is strong crossover method. Furthermore, to increase robustness of binary-coded genetic algorithm, multi-generation inheritance evolutionary strategy(MGIS) was proposed. Simulation results demonstrate the effectiveness of the proposed method.

Keywords: Binary Genetic Algorithm, Robustness of GA, Bit Simulated Crossover, Equilibrium of crossover, Multi-generation Inheritance Evolutionary Strategy.

1. Introduction

Genetic algorithm (GA) is a random searching method with some special features. One feature is that GAs are versatile evolutionary computation techniques largely based on the principle of survival of the fittest [1]. Another is the genetic operators such as crossover and mutation. Because of the randomness of genetic operators, GA can not always get good solutions.

In GA, the crossover and mutation are believed to be the main exploration operators in the working of genetic algorithms (GAs) as an optimization tool [2]. Crossover methods can be separated into many categories. One kind of them is point crossover such as single-point crossover, multi-point crossover and uniform crossover [3]. Another is probabilistic crossover methods such as bit-based simulated crossover (BSC) [4], population-based incremental learning (PBIL) [5], compact genetic algorithm (cGA) [6] and so on. In point crossover, the children are produced by combining parent and inherit some "building block" from the parent, while, in probabilistic crossover, the children are produced by a bit's frequency string determined by parent population and don't inherit "building blocks" from the parent population. In this paper, we investigate how the mutation and crossover work on the bit's frequency string and formulate the relationship between BSC and point crossover.

Furthermore, because all message from the population are stored in its bit's frequency string, in order to increase the robustness against uncertainty and search ability of GA, multi-generation inheritance evolutionary strategy (MGIS) [7] is proposed where the child generation is determined not only by parent generation but also by some previous generations.

This paper is organized as follows. Next section is about some mathematical description used in this paper. Section 3. show how mutation works on the bit's frequency string in GA and section 4. show how crossover works in GA and formulate the relationship between point crossover and BSC. Section 5. introduces MGIS and section 6. give some experiment results. The last section offers concluding remarks and future perspectives.

* Graduate School of Information Science and Electrical Eng., Kyushu University,
6-10-1, Hakozaki, Higashi-ku, Fukuoka 812-88581
** Graduate School of Information, Production and Systems, Waseda University,
Hibikino 2-7, Wakamatsu, Kitakyushu 808-0135

2. Mathematical Description

The GA studied in this paper is the one similar to Simple Genetic Algorithm defined in [2].

A k th binary individual X_k in a population can be given by

$$X_k = (x_k^1, \ldots, x_k^j, \ldots, x_k^L), \cdots\cdots (1)$$

where L is the length of the binary individual, x_k^j stands for the j th bit of the k th individual. A population $\vec{X}$ can be defined as

$$\vec{X} = (X_1, \ldots, X_k, \ldots, X_N), \cdots\cdots (2)$$

where N is the population size. The feasible space of bit x_k^j is $\{0,1\}$. The feasible space S_k of the individual X_k is $\{0,1\}^L$. The feasible population space is S_k^N.

Let $f_{\vec{X}}^j$ be the j th bit's frequency of the population $\vec{X}$, where

$$f_{\vec{X}}^j = \frac{1}{N}\sum_{k=0}^{N} x_k^j \cdots\cdots (3)$$

The feasible space S_f of $f_{\vec{X}}^j$ is $[0,1]$. The bit's frequency string $F_{\vec{X}}$ can be given by

$$F_{\vec{X}} = (f_{\vec{X}}^1, \ldots, f_{\vec{X}}^j, \ldots, f_{\vec{X}}^L), \cdots\cdots (4)$$

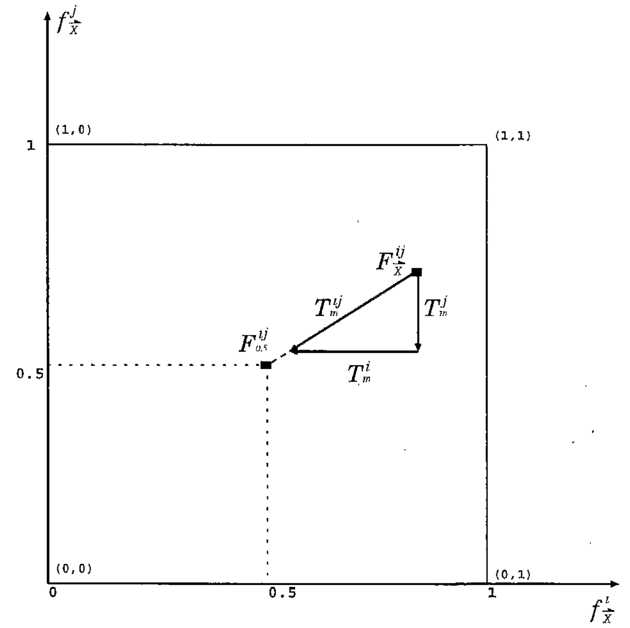


Fig. 1. Mechanism of mutation in $F_{\vec{X}}$.

where the feasible space S^L_f of the bit's frequency string is $[0,1]^L$. Generally, the bit's frequency string $F_{\vec{X}}$ can be seen as the gravity center of the population $\vec{X}$.

3. Mutation Operator

The mutation operator is a force T_m (a vector quantity) to maintain the diversity in the population and is used with a small probability, p_m. To give the direction and strength of the mutation force, we describe the population $\vec{X}$ onto the plane, for example, $(f^i_{\vec{X}}, f^j_{\vec{X}})_{i \neq j}$ plane shown in Fig.1, where $f^i_{\vec{X}}$ and $f^j_{\vec{X}}$ are the lateral and vertical coordinates. In Fig.1, $F^{ij}_{0.5}$ is the two dimensional point of $F_{0.5}$ (where $F_{0.5} = (0.5, \ldots, 0.5)$), $F^{ij}_{\vec{X}}$ is the two dimensional point of $F_{\vec{X}}$, T^{ij}_m is the two dimensional vector quantity of T_m on the plane $(f^i_{\vec{X}}, f^j_{\vec{X}})$, T^i_m and T^j_m are the component quantities of T_m on the $f^i_{\vec{X}}$ and $f^j_{\vec{X}}$ coordinates, respectively. We can easily give T^i_m and T^j_m as follows,

$$T^i_m = \frac{N_{(x^i=1)} - N_{(x^i=0)}}{N} \, p_m,$$

$$T^j_m = \frac{N_{(x^j=1)} - N_{(x^j=0)}}{N} \, p_m,$$

where p_m is the mutation rate, and $N_{(*)}$ is the number of individuals of the population $\vec{X}$ where x^i or x^j are equal to 0 or 1, therefore $N_{(x^i=1)} + N_{(x^i=0)} = N_{(x^j=1)} + N_{(x^j=0)} = N$. So we can easily give the strength of T^{ij}_m as follows,

$$\begin{aligned}|T^{ij}_m| &= \Big(\big(\frac{N_{(x^i=1)} - N_{(x^i=0)}}{N}\big)^2 \\ &\quad + \big(\frac{N_{(x^j=1)} - N_{(x^j=0)}}{N}\big)^2 \Big)^{\frac{1}{2}} p_m \\ &= \Big(\big(f^i - (1-f^i)\big)^2 + \big(f^j - (1-f^j)\big)^2 \Big)^{\frac{1}{2}} p_m \\ &= \big(4(f^i - 0.5)^2 + 4(f^j - 0.5)^2 \big)^{\frac{1}{2}} p_m \\ &= 2 \, |\overrightarrow{F^{ij}_{\vec{X}} F^{ij}_{0.5}}| \, p_m,\end{aligned}$$

where $\overrightarrow{F^{ij}_{\vec{X}} F^{ij}_{0.5}}$ is a vector from the point $F^{ij}_{\vec{X}}$ to $F^{ij}_{0.5}$, $|\overrightarrow{F^{ij}_{\vec{X}} F^{ij}_{0.5}}|$ is the length of the vector $\overrightarrow{F^{ij}_{\vec{X}} F^{ij}_{0.5}}$. The direction of T^{ij}_m can be easily demonstrated to be the same as the direction of the vector $\overrightarrow{F^{ij}_{\vec{X}} F^{ij}_{0.5}}$. Generally, we can easily give the strength of T_m as follows,

$$|T_m| = 2 \, |\overrightarrow{F_{\vec{X}} F_{0.5}}| \, p_m, \cdots\cdots\cdots\cdots\cdots (5)$$

where $|\overrightarrow{F_{\vec{X}} F_{0.5}}|$ is the distance between point $F_{\vec{X}}$ and $F_{0.5}$. The direction of T_m is from point $F_{\vec{X}}$ to $F_{0.5}$.

According to Eq.(5), we can see mutation operator can change the bit's frequency string, where the strength of the mutation force is changed proportionally along with the convergence status (represented by $|\overrightarrow{F_{\vec{X}} F_{0.5}}|$) of the population and the direction of the mutation force is always from the point $F_{\vec{X}}$ to $F_{0.5}$.

4. Crossover Operator

The crossover operator T_c is a very complex operator to recombine the gene of each individual in the population. There exist a number of crossover operators in the GA literature. One kind of them is point crossover such as single-point crossover, multi-point crossover and uniform crossover. Another kind of them is probabilistic crossover such as BSC, PBIL, cGA and so on. To understand how crossover works in GA, let us review multi-point crossover and BSC at first.

(1) Multi-point crossover is a kind of classic crossover method. In multi-point crossover, two parent strings are cut at several random sites. For example, a two-point crossover can be looked like as follows,

parent	11|11|11	child	110011
parent	00|00|00	child	001100

Although point crossover can cause shuffling of the gene in population, it don't change bit's frequency string of the population.

(2) BSC uses the bit's frequency string of the parent population to generate the offspring using following two steps,

- According to the Eq.(3), compute the bit's frequency $(f^1, \ldots, f^j, \ldots, f^L)$ from parent population.
- Sample child individuals where j th bit of individual is generated by using f^j.

4.1 Equilibrium of Crossover

Before investigate the relationship between BSC and point crossover, let us see some definition as follows.

Definition (Equilibrium of crossover) When point crossover doesn't change the distribution probability of the population, we say the population is at the state of equilibrium of crossover. In other words, the population before point crossover and after point crossover have the same distribution probability.

Theorem 1 A population which is produced by using BSC is at the state of equilibrium of crossover.

Proof: First, when using BSC, the population $\vec{X}$ should be distributed with following probability,

$$P_{(x^1,\ldots,x^j,\ldots,x^L)} = \prod_{j=1}^{L} \hat{f}^j_{\vec{X}} \cdots\cdots\cdots\cdots (6)$$

where

$$\hat{f}^j_{\vec{X}} = \begin{cases} f^j_{\vec{X}} & \text{if } x^j = 1 \\ 1 - f^j_{\vec{X}} & \text{if } x^j = 0 \end{cases}$$

and $P_{(x^1,\ldots,x^j,\ldots,x^L)}$ is the probability that the individual $(x^1, \ldots, x^j, \ldots, x^L)$ is produced.

If we describe the population $\vec{X}$ onto the $(f^i_{\vec{X}}, f^j_{\vec{X}})_{i \neq j}$ plane shown in Fig.1, Eq.(6) should be represented as follows,

$$P_{(x^i x^j)} = \hat{f}^i_{\vec{X}} \hat{f}^j_{\vec{X}} \cdots\cdots\cdots\cdots (7)$$

Second, when using point crossover, if we describe the population $\vec{X}$ onto the $(f^i_{\vec{X}}, f^j_{\vec{X}})_{i \neq j}$ plane shown in Fig.1, we can see

- 00 and 01 can create 01 and 00;
- 00 and 10 can create 10 and 00;
- 00 and 11 can create 10 and 01;
- 01 and 10 can create 11 and 00;
- 01 and 11 can create 11 and 01;
- 10 and 11 can create 10 and 11.

Notably, only "00 and 11", "10 and 01" can produce new individuals while others don't produce new individuals. If a population $\vec{X}$ is not changed by the point crossover, it means that the number of individuals at 00, 01, 10 and 11 should be maintained stably. Because only "00 and 11", "10 and 01" can produce new individuals, we can get the following relationship,

$$\frac{N_{(00)}}{N} \frac{N_{(11)}}{N} = \frac{N_{(01)}}{N} \frac{N_{(10)}}{N}, \cdots\cdots\cdots\cdots (8)$$

where $N_{(**)}$ is the number of individuals at $(**)$, $* \in \{0, 1\}$.

Furthermore, the distribution of the population should satisfy the following functions,

$$N_{(11)} + N_{(10)} = N f^i_{\vec{X}},$$
$$N_{(11)} + N_{(01)} = N f^j_{\vec{X}},$$
$$N_{(00)} + N_{(01)} + N_{(10)} + N_{(11)} = N. \cdots\cdots\cdots (9)$$

We can obtain the solution of Eq.(8) and Eq.(9) as follows,

$$N_{(11)} = N f^i_{\vec{X}} f^j_{\vec{X}},$$
$$N_{(01)} = N(1 - f^i_{\vec{X}}) f^j_{\vec{X}},$$
$$N_{(10)} = N f^i_{\vec{X}} (1 - f^j_{\vec{X}}),$$
$$N_{(00)} = N(1 - f^i_{\vec{X}})(1 - f^j_{\vec{X}}).$$

Thus, the solution can be represented by

$$\frac{N_{(x^i x^j)}}{N} = \hat{f}^i_{\vec{X}} \hat{f}^j_{\vec{X}} \cdots\cdots\cdots\cdots (10)$$

Supposing N sufficiently large, Eq.(7) and Eq.(10) are the same and the proof is completed.

According to theorem 1, we can see BSC is equivalent to applying point crossover d times where $d \to \infty$. In other words, the evolutionary process by using BSC should be stabler than by using point crossover because it can decrease the randomness of point crossover operator.

Another advantage by using BSC is that it has the chance to create more candidate solutions than point crossover. Furthermore, because all messages from the environment are stored in the bit's frequency, sometimes, it is very useful to use the BSC in order to decrease the memory required for calculation like PBIL.

In BSC, each bit of individuals is generated independently, so the correlation of the gene which is produced by using the Building Block Hypothesis [1] is ignored. Sometimes, it caused some difficulties in evolutionary process.

5. Multi-Generation Inheritance Evolutionary Strategy (MGIS)

5.1 Description of MGIS The basic structure of genetic algorithm by using BSC is shown in Fig.2. In BSC, the bit's frequency of child population is only determined by parent population. In order to increase the robustness against uncertainty and search ability of GA, MGIS is proposed where the bit's frequency string of child population is determined by many previous generation. Thus, the bit's frequency string $F_{\vec{X}(t+m)}$ of the $t+m$ th generation is given by,

$$F_{\vec{X}(t+m)} = (f^1_{\vec{X}(t+m)} \cdots f^j_{\vec{X}(t+m)} \cdots f^L_{\vec{X}(t+m)}),$$
$$m = 1, 2, 3, \ldots,$$

where

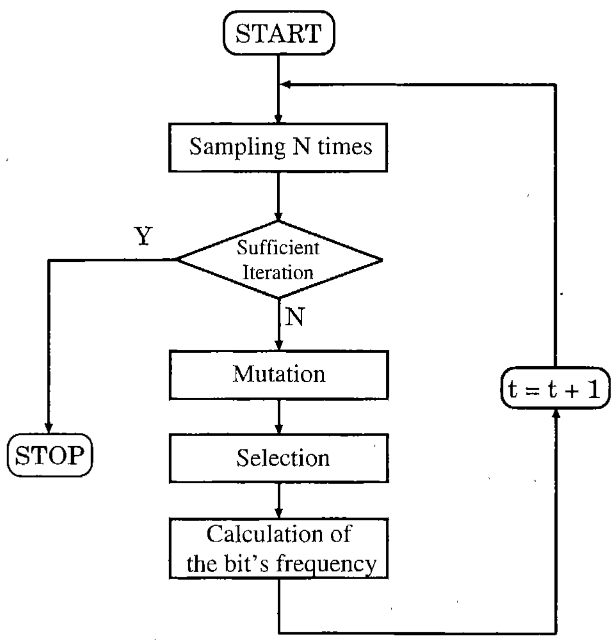


Fig. 2. Structure of a simple GA by using BSC.

$$f^j_{\vec{X}(t+m)} = \frac{1}{m+1}\,(f^j_{\vec{X}'(t+m-1)} + \sum_{i=0}^{m-1} f^j_{\vec{X}(t+i)}).(11)$$

$\vec{X}$ is the population before selection and $\vec{X}'$ is the population after selection. $f^j_{\vec{X}'(t+m-1)}$ means the j th bit frequency of the population $\vec{X}'$ at $t+m-1$ th generation. Eq.11 means the bit's frequency string at $t+m$ th generation is determined not only by the bit's frequency string $F_{\vec{X}'(t+m-1)}$ but also by the bit's frequency string from t th to $t+m-1$ th generation.

5.2 Reason of Robustness To investigate the effects of MGIS, let us see a special case where $m = 1$. If $m = 1$, Eq.11 can be described as follows,

$$f^j_{t+1} = \frac{1}{2}\,(f^j_{\vec{X}'(t)} + f^j_t). \cdots\cdots\cdots\cdots\cdots\cdots (12)$$

Eq.12 is a recursion formula and can be easily converted into as follows,

$$f^j_{t+1} = \frac{1}{2}f^j_{\vec{X}'(t)} + \frac{1}{2^2}f^j_{\vec{X}'(t-1)} + \cdots + \frac{1}{2^t}f^j_{\vec{X}'(1)} + \frac{1}{2^t}f^j_1 \cdots\cdots\cdots\cdots\cdots\cdots (13)$$

According to Eq.13, we can see the effects from the first generation to t th generation on $t+1$ th generation. This is the reason why MGIS can make GA search for solutions more robust against uncertainty than "conventional" GA. When $m \geq 2$, the relationship between f^j_{t+m} and $(f^j_{\vec{X}'(t+m-1)}, \cdots, f^j_{\vec{X}'(1)}, f^j_1)$ is a little difficult to represent.

6. Experiments

6.1 Test Function Generalized Schwefel's Problem which was examined in [9] [10] is used in our experimental studies.

$$\min f(x) = -\sum_{i=1}^{K}(x_i \sin(\sqrt{|x_i|})),$$
$$\text{where} \quad K = 1, 2, \ldots, 30$$
$$-500 < x_i < 500$$

This function is a multimodal function with many local minima, where the number of local minima increases exponentially as the dimension of the function increases like 7^K. The global minimal function's value is $K \times 418.98289$. Fig.3 shows the two-dimensional version of f.

6.2 Parameter Values

- Population size: Since the problem dimensions are high, we choose a moderate population size N=200;
- Representation: Each variable has 30 bits, so the length L of the individuals is $30 \times K$.
- Crossover rate: We set crossover rate 1.0 and 0.5 for the uniform crossover method respectively.
- Mutation probability: We choose $p_m = \frac{1}{L}$.
- Selection pressure: We use the nonlinear ranking method [8] where the selection probability of k th individual can be calculated by $p_k = c \times (1-c)^{i-1}$,

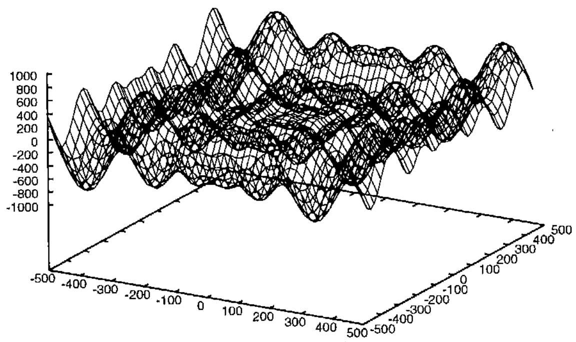


Fig. 3. The two-dimensional version of f

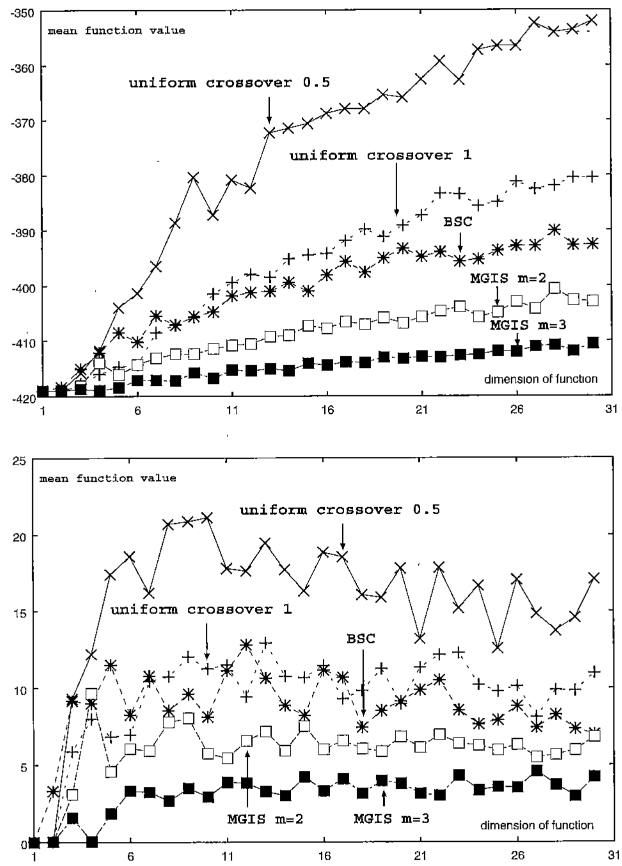


Fig. 4. The mean and the standard deviation of function value by using uniform crossover, BSC and MGIS.

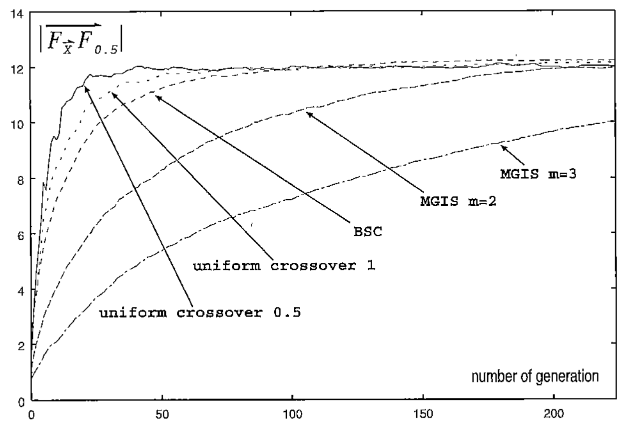


Fig. 5. The convergence status by using uniform crossover, BSC and MGIS.

i is the rank of k th individual. We set the parameter $c = 0.05$.

- Iteration: The stopping generation is $\lfloor 50 \times \sqrt{K} + 0.5 \rfloor$ because $|\overrightarrow{F_{\bar{X}} F_{0.5}}|$ is changed proportionally along with $\sqrt{K}$, where $\lfloor\ \rfloor$ means taking round-off number.

6.3 Discussions We performed 50 independent runs by using

- uniform crossover where the crossover rate is 0.5 and 1.0,
- BSC,
- MGIS where $m = 1$ and $m = 2$,

from $K = 1$ to $K = 30$ and recorded 1)mean function value (the mean value of the best individual of the last generation over 50 runs) and 2)the standard deviation of function value (the standard deviation of the best individual of the last generation over 50 runs). Fig.4 shows the simulation results. The upper part shows the mean function value $\bar{f}$ where the lateral coordinate is the dimension of the test function, while the lower part shows the standard deviation of the function value σ_f. They can be calculated as follows:

$$\bar{f} = \frac{1}{50}\sum_{i=1}^{50}\frac{f_i}{K},$$

$$\sigma_f = \sqrt{\frac{1}{50}\sum_{i=1}^{50}(\frac{f_i}{K} - \bar{f})^2}.$$

Comparing the experiment result of BSC, uniform crossover with crossover rate 0.5 and 1.0, we can see when the dimension of the function is large, the mean function value of BSC is smaller than the uniform crossover with crossover rate 1.0, followed by the uniform crossover with crossover rate 0.5. It means that BSC has higher search ability than the uniform crossover. The reason is because BSC has the chance to create more candidate solutions than uniform crossover. The standard deviation of BSC is a little smaller than that of the uniform crossover and the standard deviation of the uniform crossover with 1.0 crossover rate is smaller than that of the uniform crossover with 0.5 crossover rate. It means that BSC can increase the robustness of GA against uncertainty. The reason is that BSC can decrease the randomness of point crossover operator.

Next, let us see the experiment results of MGIS with $m = 1$ and $m = 2$. We can see the mean function value and the standard deviation of MGIS are smaller than those of BSC and uniform crossover. It means MGIS can increase the search ability and robustness against uncertainty of GA. Comparing $m = 1$ and $m = 2$, we can see the mean function value and the standard deviation of $m = 2$ are smaller than those of $m = 1$. It means the increase of m can make GA search for solutions more robust.

Fig.5 shows the simulation results of the convergence speed where the number of function's dimension K is 20. In Fig.5, the lateral coordinate is the number of generation and the vertical coordinate is $|\overrightarrow{F_{\bar{X}} F_{0.5}}|$. According to the simulation results, we see that the order of the convergence speed from low to high is MGIS with $m = 2$, MGIS with $m = 1$, BSC, uniform crossover with 1.0 crossover rate and uniform crossover with 0.5 crossover rate.

7. Conclusion

In this paper, we focus on how the crossover and mutation work in GA based on the variance of the bit's frequency. Based on the analysis of the equilibrium of point crossover, we can find BSC is equivalent to applying point crossover d times where $d \to \infty$. So it can decrease the randomness of point crossover operator and can increase the robustness against uncertainty of GA. Furthermore MGIS is proposed where $t + m$ th generation is determined not only by $t + m - 1$ th generation but also by generations from t th to $t + m - 2$ th. Some experiments have clarified that MGIS can increase the searching ability and robustness against uncertainty of GA.

Acknowledgment

This research was partly supported by the 21st Century COE Program "Reconstruction of Social Infrastructure Related to Information Science and Electrical Engineering."

(Manuscript received March 25, 2002,
revised Jan. 17, 2003)

References

(1) J.H. Holland: "Adaptation in Natural and Artificial Systems". Ann Arbor, MI: University of Michigan Press (1975)

(2) D.E. Goldberg: "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading, MA (1989)

(3) G. Syswerda: "Uniform crossover in genetic algorithms", Proceedings of the Third International Conference on Genetic Algorithms, pp.1–9 (1989)

(4) G. Syswerda: "Simulated crossover in genetic algorithms", Proceedings of Foundations of Genetic Algorithms, Morgan-Kaufmann, San Mateo pp.239–255 (1993)

(5) S. Baluja: "Population-Based incremental learning: Amethod for integrating genetic search based function optimization and competitive learning", (Tech Rep. No.CMU-CS-94-163) Pittsburg, PA: Carnegie Mellon University, (1995)

(6) G.R. Harik, F.G. Lobo, and D.E. Goldberg, "The compact genetic algorithm", *IEEE Trans. on Evol. Comp.*, Vol.3, No.4, pp.287–297 (1999)

(7) J. Mao, K. Hirasawa, J. Hu, and J. Murata: "Increasing Robustness of Genetic Algorithm", *Proc. of the Genetic and Evolutionary computation Conference*, pp.456–462 (2002)

(8) Z. Michalewicz: "Genetic Algorithm + Data Structures = Evolution Programs", second, extended edition, Springer (1994)

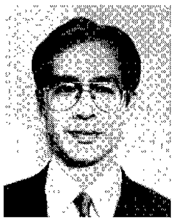
(9) X. Yao and Y. Liu: "Fast evolution strategies", *IEEE Trans. Evol. Comput.* Vol.3, No.2, pp.82–102 (1999)

(10) Y. Leung and Y. Wang: "An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization", *IEEE Trans. Evol. Comput.* Vol.5, No.1, pp.41–53 (2001)

Jiangming Mao (Non-member) received the B.S degree in Electrical Engineering from Nanjing University of Science and Technology, China, in 1995. and M.S. degree in the Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan in 2001. Now He is a Dr. candidate at the Graduate School of Information Science and Electrical Engineering, Kyushu University. His current research interest lies in Evolutionary computation.

Kotaro Hirasawa (Member) received the B.S. and M.S. degrees in Electrical Engineering from Kyushu University, Japan in 1964 and 1966, respectively. From 1966 to 1992, he was with Hitachi Ltd., where he served as a Vice President at the Hitachi Research Laboratory. From December 1992, he was a professor at the Graduate School Information Science and Electrical Engineering, Kyushu university. Since September 2002, he has been a professor at the Graduate School of Information, Production and Systems, Waseda University. Dr. Hirasawa is a member of of SICE, IEEJ and IEEE.

Jinglu Hu (Member) received the B.S degree and the M.S degree in Electronic Engineering from Zhongshan University, China, in 1983 and 1986 respectively, and the Ph.D degree in Computer Science and Engineering from Kyushu Institute of Technology, Japan, in 1997. From 1986 to 1993, he worked at Zhongshan University, where he was a Research Associate and then Lecturer. From 1997 to 2003, he was a Research Associate at Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan. Since April 2003, he has been an Associate Professor at Graduate School of Information, Production and Systems, Waseda University. His current research interests are system identification, learning network and their applications. Dr Hu is a member of SICE, IEEJ.



Junichi Murata (Member) received the Master and Doctor of Engineering degrees from Kyushu University, Japan, in 1983 and 1986, respectively. He then became a Research Associate and an Associate Professor at the Faculty of Engineering, Kyushu University. He is currently an Associate Professor at Graduate School of Information Science and Electrical Engineering. Kyushu University. His current research interests are neural networks, self-organizing systems and their applications to control and identification. Dr. Murata is a member of SICE, ISCIE, IEEJ and IEEE.