

On the Effect of Multi-parents Recombination in Binary Coded Genetic Algorithms

Shigeyoshi Tsutsui

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

Lakhmi C. Jain

Knowledge-Based Intelligent Engineering Systems Centre
University of South Australia
Adelaide, The Levels SA 5095, Australia
Lakhmi.Jain@unisa.edu.au

Abstract--Recombination operator plays a very important role in genetic algorithms. In this paper, we present binary coded genetic algorithms in which more than two parents are involved in recombination operation. We propose two types of multi-parent recombination operators, the *multi-cut* (MX) and *seed crossover* (SX). Each of these operators is a natural generalization of two parents recombination operator. These operators are evaluated on the De Jong standard test functions. The results showed clearly that the multi-parent recombinations lead to better performance, although the performance improvement for different techniques were found to be dependent on problems.

1. Introduction

In Evolutionary Algorithms (EAs), recombination operation with two parents is commonly used to produce offspring (thereby emulating natural sexual reproduction system) [16, 17]. However, there is no necessity to restrict to two parents recombinations only as EAs allow us to emulate natural evolution in a more flexible manner. A few attempts to study the effect of using more than two parents for recombination in EAs are reported in the literature [1, 5, 6, 7, 11, 14]. A brief review on multi-parent recombination in EAs has been made in [7]. The first attempt in this line of research was *global recombination* in Evolution Strategies (ESs) [11] that produces one new individual which may inherit genes from more than two parents. Nevertheless, the number of parents is not fixed and thus, global recombination does not make sexuality a graded feature. The same holds for the recently introduced multi-parent Gene Pool Recombination [14] and the Gene Linkage method [12] in Genetic Algorithms (GAs).

An extension of ES, the $(\mu/\rho, \lambda)$ multi-recombination strategy [1], uses an adjustable operator with arity ρ .

Generalized multi-parent recombination operators in GAs are *scanning crossover* and *diagonal crossover* (DX), introduced in [5, 6, 7]. These operators were evaluated on the standard test functions with bit string representation and other types of problems and it was shown that two parent recombination is inferior on the test functions. In [7], these operators were evaluated on Kauffman's NK-landscapes which allow for systematic characterization and user control of ruggedness of the fitness landscape, and the superiority of sexual recombination on mildly epistatic problems was found.

In this paper, we propose two types of new multi-parent recombination operations for binary coded GAs, namely, the *multi-cut crossover operator* (MX) and the *seed crossover operator* (SX) and evaluate them on the De Jong standard functions. In the MX, the number of crossover points are taken to be identical to the number of parents, and m offsprings are generated from m parents. On the other hand, in the SX two parents recombination operation is sequentially used as in a *knock-out tournament* (in inverse order of ranks of individuals), and only one offspring is generated from m parents.

Experiments to see the effectiveness of the proposed techniques are performed on the standard test suite of De Jong [4]. The MX showed consistently good performance when the number of parents were 4-6. The SX showed better performance as the number of parents were increased on simple test functions such as $F1$ and $F5$, and was even better than MX. But for the Rosenbrock function ($F2$), where strong epistasis exists between parameters, the performance

of SX deteriorated as the number of parents increased. The performance of SX, however, was improved when we used high mutation rate. We also evaluated the DX introduced in [5, 6] in comparison with MX and it was shown the performance of MX outperforms DX on all the test functions.

In the next section we give detailed description of the proposed multi-parent recombination operators. In Section 3, we present experimental methodology and analysis of results. Finally, concluding remarks are made in Section 4.

2. Proposed Multi-parent Recombination Operators

In this section, we present two types of multi-parent recombination operators, namely, MX and SX. Each of these operators is a natural generalization of two parents recombination operator.

2.1 Multi-cut crossover (MX)

In MX, the number of crossover points is taken to be identical to the number of parents m , and m offsprings are generated from m parents as shown in Figure 1. Although MX is an extension of the DX, where the number of crossover points is taken to be $m - 1$ as shown in Figure 2, there is a big difference between MX and DX. The difference becomes clear for $m = 2$, where MX is identical to a two-point crossover and DX is identical to a one-point crossover. Thus we can expect MX to inherit the advantages of the two-points crossover which has better performance than one-point crossover.

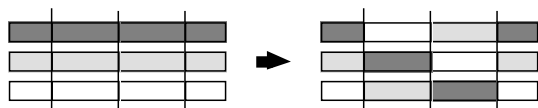


Figure 1 Multi-cut crossover

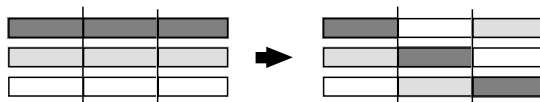


Figure 2 Diagonal crossover

2.2 Seed crossover (SX)

In this technique, m parents are chosen and arranged (seed) according to their fitness values. Then, two parents recombination operation is sequentially (in inverse order of ranks of individuals) used as in a *knockout tournament* (see Figure 3), and only one offspring is generated from m parents. Thus, the offspring generated by SX tends to inherit characteristics of parents which have higher fitness values. For two-parent recombination operation, we can choose any type of operator. In this research, we used the *uniform crossover* operator.

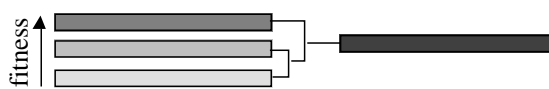


Figure 3 Seed crossover

3. The Experiments

3.1 Experimental Methodology

The basic evolutionary model we used in these experiments is the *simple GA* (SGA) [xx] with population size of 40 and crossover rate of 1.0. The elite strategy was used.

The test functions used here are the De Jong test suite [4] (except $F4$). $F1$ is a simple unimodal function and has the global minimum at $(0, 0, 0)$. $F2$ has strong inter-parameter linkage (epistasis) and has the global minimum at $(1, 1)$. $F3$ is a discontinuous function with the global minimum in the range $x_i \in [-5.12, -5.0)$ for $i = 1, \dots, 5$, i.e., in one corner of the search space. $F5$ is basically a continuous function, but it has 25 deep holes and has the global minimum at $(-31.978, -31.978)$.

We evaluated the algorithms by #OPT (number of runs in which the algorithm succeeded in finding the global optimum) for restricted number of trials. Two hundred runs were performed. In each run, the initial population $P(0)$ is randomly initialized. Each run continues until the global optimum is found or a maximum of 100,000 trials is reached. The performance is tested varying the number of parents from 2 to 8 in steps of 2.

3.2 Results with normal mutation rate

Although optimal mutation rate in GAs depend on the function to be optimized, a mutation rate of 0.006 is often used in the literature. We first evaluated MX and SX with this mutation rate (normal mutation rate). Figure 4 shows the results with this mutation rate.

3.2.1 Results with MX

The results on functions $F2$, $F3$ and $F5$ showed that performance increased as the number of parents (m) was increased from 2. On function $F1$, the best performance is noticed when m was 4. Thus we can confirm the effectiveness of the multi-parents recombination with MX. Comparing MX with DX, DX also showed similar tendency for increasing number of parents, but MX consistently showed better performance than DX on all functions.

3.2.2 Results with SX

The results on functions $F1$ and $F3$ showed increase in performance as the number of parents (m) was increased from 2 as with MX. On function $F5$, the best performance is noticed when m was 4. In the SX, offsprings inherit characteristics of parents having high fitness to greater extent. As the number of parents increases, this tendency becomes prominent. Thus, for simple unimodal functions like $F1$, it works well and it worked on $F3$ and $F5$ too.

On the other hand, the results on $F2$ showed significant decrease in performance as the number of parents increased. Since the function $F2$ has strong epistasis among parameters and the SX fails to use this linkage information when m is more than 2, thereby deteriorating the performance.

These results can be observed by comparing MX and SX directly in Figure 4. On simple unimodal function $F1$, SX showed much better performance than MX. On the contrary, SX on functions $F3$ and $F5$ was a little poorer, and on function $F2$, showed much worse performance than MX.

3.3 Results with high mutation rate

Next we evaluated MX and SX using mutation rate of 0.02, almost three times the normal mutation rate, while other conditions were maintained at the same levels as in 3.2 Fig-

ure 5 shows the results with this high mutation rate.

3.3.1 Results with MX

The results of MX on all functions with high mutation rate showed poorer performance than with normal mutation rate. These results seem reasonable because using the high mutation rate destroys good genetic materials of individuals and thus offsprings could not inherit the good genetic materials from their parents.

3.3.2 Results with SX

The results of SX on functions $F1$, $F3$ and $F5$ with high mutation rate increased the performance as the number of parents (m) was increased from 2 and the increase in performance rate was greater than SX with normal mutation rate (see Figure 4). On function $F2$, although the best performance was obtained for $m = 4$, we observed again significant decrease in performance as the number of parents increased. On function $F2$, the SX with high mutation rate did not work well again, although a little increase in performance was observed compared with the results with normal mutation rate.

In SX, since offsprings inherit characteristics of parents having higher fitness to greater extent, the population has a tendency to lose diversity rapidly. By using high mutation rate, we may compensate the rapid loss of diversity with SX and get higher performance on functions $F1$, $F3$ and $F5$ in comparison to results with the normal mutation rate.

4. Concluding Remarks

In this paper, we proposed three types of multi-parent recombination operators, namely, the *multi-cut crossover operator* (MX) and *seed crossover operator* (SX) and showed the effect of multi-parent recombination on the De Jong standard test functions.

The MX worked well on all functions tested and the results showed clearly that multi-parent recombinations with more than two parents lead to better performance. Further the MX worked better than the DX. The SX performed well on simple functions and the performance increased by ap-

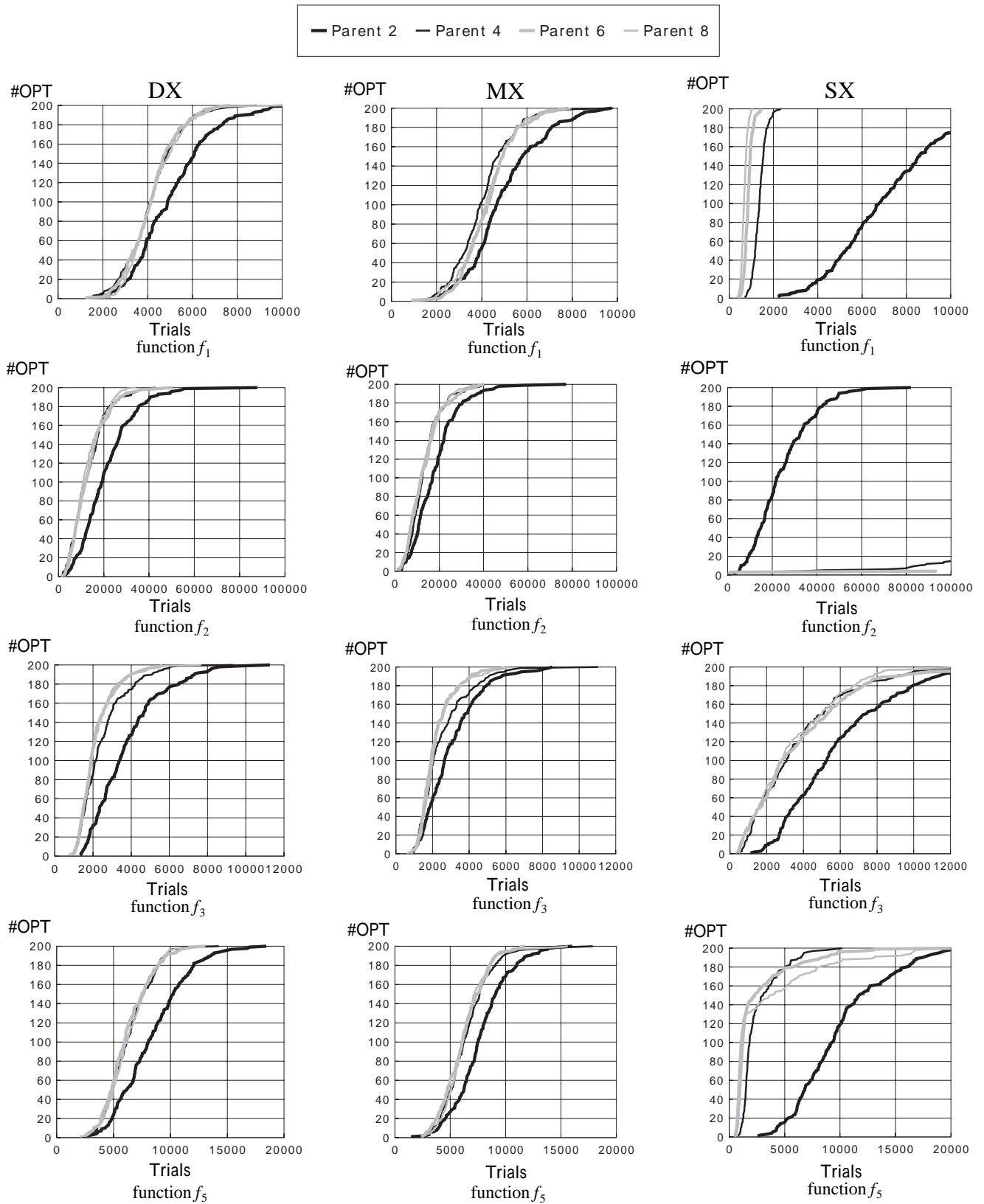


Figure 4 #OPT for restricted number of trials with normal mutation rate (0.006)

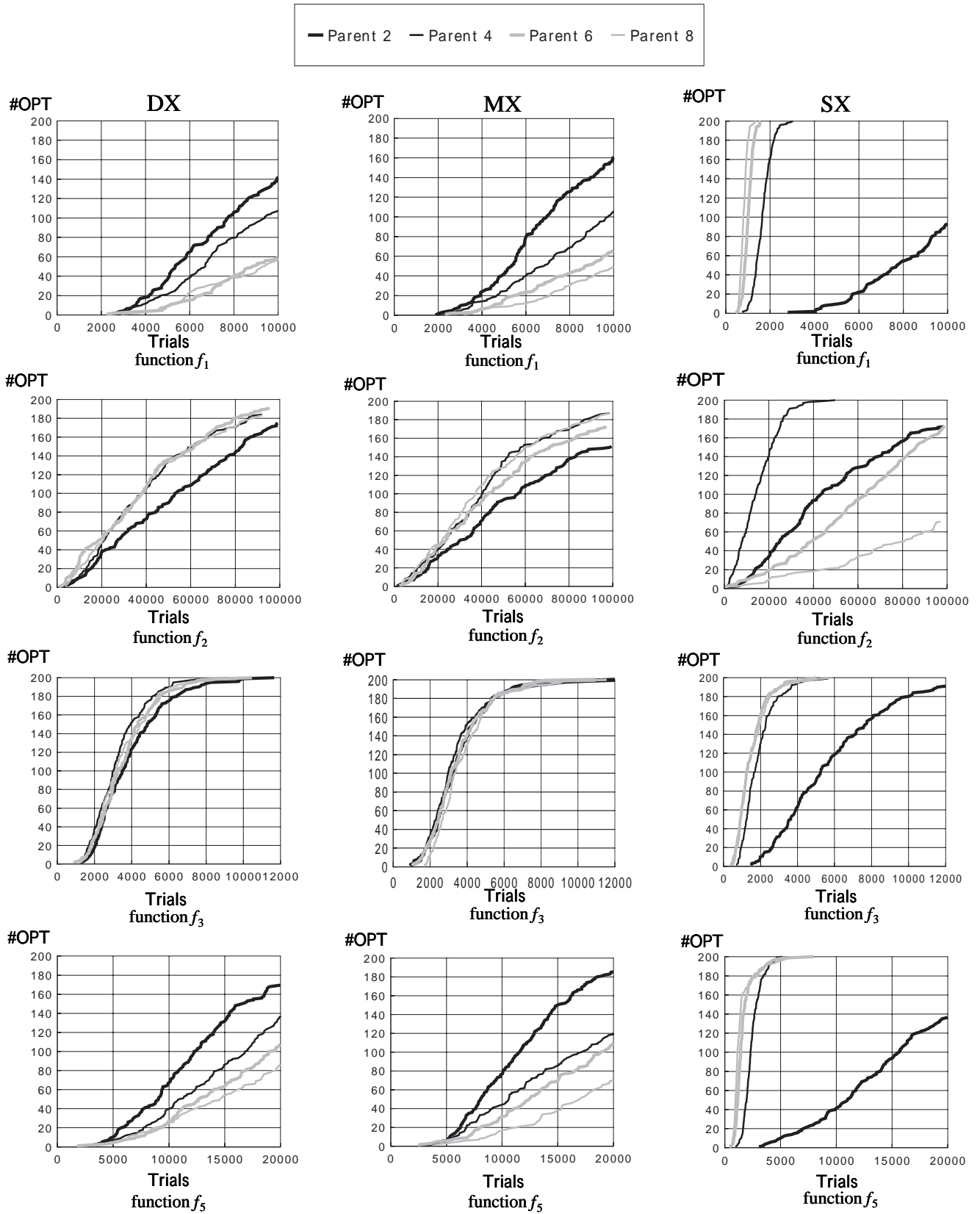


Figure 5 #OPT for restricted number of trials with high mutation rate (0.02)

plying high mutation rate. But it did not work well with large number of parents on functions having epistasis.

In this work, we only tested these operators on a limited number of test functions. The evaluation of multi-parents recombination operator on systematic test functions is a topic for future study. The effect of multi-parent recombination in real-coded GAs is another topic that is currently being studied [15].

Acknowledgments

We like thank Mr. Jyunya Suzuri and Hidenori Nakajima for their help in carrying out the experiments. This research is partially supported by the Ministry of Education, Science, Sports and Culture under Grant-in-Aid for Scientific Research on Priority Areas number 264-08233105, and the Financial Aid Research Program at the Institute of Industrial and Economics, Hannan University, Japan.

References

- [1] Beyer, H.-G.: Toward a Theory of Evolution Strategies: On the Benefits of Sex- the $(\mu/\mu, \lambda)$ Theory, *Evolutionary Computation*, 3(1), pp. 81-111 (1995).
- [2] Davis, L.: *The Handbook of Genetic Algorithms*, Von Nostrand Reinhold, New York (1991).
- [3] De Jong, K. A.: *Analysis of the behavior of a class of genetic adaptive systems*. Ph. D. dissertation, Dept. Computer and Communication Sciences, University of Michigan, Ann Arbor (1975).
- [4] De Jong, K. A., Potter M. A. and Spears, W. M.: Using Problem Generators to Explore the Effects of Epistasis, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp. 338-345 (1997).
- [5] Eiben, A. E., Raue, P-E. and Ruttkay, Zs.: Genetic algorithms with multi-parent recombination, *Proceedings of the Parallel Problem Solving from Nature III*, pp. 78-87(1994).
- [6] Eiben, A. E., van Kemenade, C. H. M. and Kok, J. N.: Orgy in the Computer: Multi-Parent Reproduction in Genetic Algorithms, *Proceedings of the 3rd European Conference on Artificial Life*, LNAI 929, Springer-Verlag, pp. 934-945(1995).
- [7] Eiben, A. E. and Schippers, C. A.: Multi-Parent's Niche: n-ary Crossover on NK-Landscapes, *Proceedings of the Parallel Problem Solving from Nature IV*, pp. 319-328 (1996).
- [8] 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).
- [9] Fogel, D. B., and Atmar, J. W.: Comparing Genetic Operators with Gaussian Mutations in Simulated Evolutionary Processes Using Linear Systems, *Biological Cybernetics*, 66, pp. 111-114 (1990).
- [10] Michalewicz, Z. : *Genetic Algorithms + Data Structures = Evolution Program*, Springer-Verlag (1994).
- [11] Schewefel, H.-P. : *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series, Wiley, New York (1995).
- [12] Smith, J and Fogarty, T. C.: Recombination Strategy Adaptation via Evolution of Gene Linkage, *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, pp. 826-831 (1996).
- [13] Wright, A. H. : Genetic Algorithms for Real Parameter Optimization, *Foundations of Genetic Algorithms*, Morgan Kaufman Publishers, San Mateo, pp. 205-218 (1991).
- [14] Voigt, H.-M. and Mühlenbein, H.: Gene Pool Recombination and Utilization of Covariances for the Breeder Genetic Algorithms, *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pp. 172-177 (1995).
- [15] Tsutsui, S and Ghosh, A.: A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms, *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (ICEC '98)*, (1998, to appear).
- [16] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, pp. 28-33 (1989).
- [17] Vonk, E., Jain, L. C. and Johnson, R. P.: Automatic Generation of Neural network Architecture using Evolutionary Computation, World Scientific Publishing Company Pte Ltd. (1997).