New Crossover Scheme for Parallel Distributed Genetic Algorithms

Mitsunori MIKI[†], Tomoyuki HIROYASU[†], Jun-ichi YOSHIDA[††], and Ikki OHMUKAI[††]

[†] Department of Knowledge Engineering,
[††] Graduate School of Engineering,
Doshisha University
Kyo-tanabe, Kyoto, 610-0321, JAPAN
Email: mmiki@mail.doshisha.ac.jp

Abstract

This paper proposes a new crossover method for parallel distributed genetic algorithms (PDGAs). PDGAs with multiple subpopulations provide better solutions than conventional GAs with a single population. The proposed method, including the hybridization crossover and the best combinatorial crossover, is designed to increase the performance of PDGAs. The proposed method, which provides high local search ability in each subpopulation and high global search ability by the migration, is evaluated with four standard test functions. The experimental results show that the proposed method is very effective.

keywords: Parallel Distributed Algorithms, Optimization, Genetic Algorithms, Crossover Scheme

1 Introduction

Genetic Algorithms (GAs) are stochastic search algorithms based on the mechanics of natural selection and natural genetics[1]. When GAs are applied to real domain applications, they require many generations and a large number of individuals in the population. Since massive computational resources are usually required in order to obtain good solutions, a number of research efforts to implement GAs on parallel computers are underway [2].

The island model is a typical implementation of GAs into parallel computers. It divides a large population into smaller subpopulations and executes traditional GAs on each subpopulation separately. It periodically selects individuals from each subpopulation and moves them to different subpopulations in an exchange called migration. In this paper, we call GAs with the island model distributed GAs (DGAs).

DGAs can show better performance than single population GAs (SPGAs). Tanese demonstrated the ability to find fitter individuals than the traditional method[3]. Belding extended Tenese's work on DGAs to the different fitness functions (the royal road problem) in order to determine whether the original results were specific to the Tanese functions[4]. He showed that DGAs outperformed SPGAs.

The DGA is suitable to implement on parallel computers because the communication between processors occurs only in the migration. Therefore, if each subpopulation is assigned to each processor of a parallel computer, a near-linear reduction in speed is expected. In this case, the GAs are called parallel DGAs(PDGAs).

The crossover operation plays an important role in the search of each GA, but the search mechanism of PDGAs is different from that of SPGAs. Therefore, it is important to adjust the crossover scheme to the search mechanism of PDGAs.

Although quite a number of detailed studies have appeared on the effect of the crossover operator and the optimal adjustment of the crossover rate for SPGAs[5, 6, 7, 8], very little has been written for PDGAs. In this paper we investigate the effect of the crossover operator and the crossover rate and propose new crossover schemes suitable for PDGAs.

2 Parallel Distributed GAs

2.1 Genetic Algorithms

Genetic algorithms are stochastic search algorithms based on principles of natural selection and recombination[9]. GAs attempt to find the optimal solution to the problem at hand by manipulating a population of candidate solutions. The population is evaluated and the best solutions are selected to reproduce and mate to form the next generation. Over a number of generations, good traits dominate the population, resulting in an increase in the quality of the solutions. There are three genetic operators in canonical GAs: selection, crossover, and mutation.

2.2 Parallel Distributed GAs

In the PDGA, a large population is divided into smaller subpopulations, and a traditional GA is executed on each subpopulation separately. Some individuals are selected from each subpopulation and migrated to different subpopulations periodically, as shown in Figure 1. Two parameters are introduced in the migration process: migration interval which is the number of generations between each migration, and migration rate which is the percentage of individuals selected for migration from each subpopulation at the time of migration.

Each subpopulation can be assigned to each processor of a parallel computer, and inter-processor communication occurs only at the migration. The migration topology adopted here is a ring with random destinations where each subpopulation has one destination and the destinations are determined randomly at every migration period as shown in Figure 1. The emigrants are selected randomly in their subpopulation.

Migration selects individuals from one subpopulation and sends them to another subpopulation, which may contain individuals with very different kinds of building blocks. After migration, the genetic algorithm will mix the immigrants with the rest of the individuals in the destination subpopulation.

In PDGAs, the variety of the individuals in the whole population is increased by migration, mentioned above. This large variety improves the effectiveness of the search in PDGAs compared with SPGAs.

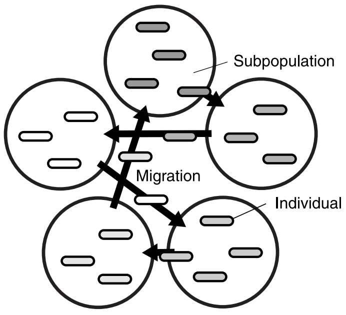

Figure 1: PDGAs

2.3 The growth of solutions in PDGAs

In SPGAs, some individuals who carry relatively good schemata prevent the growth of other individuals. This will cause premature convergence. Therefore, it is important not only to preserve good schemata, but also to maintain the variation of the individuals.

In PDGAs, dividing populations into multiple subpopulations increases the variety of individuals. Certainly, small subpopulations tend to convergence rapidly. However, migration exchange individual whose bit composition is very different from the individuals in the destination subpopulation. Hence, the variety of whole population of PDGAs is higher than that of SPGAs.

Furthermore, different good schemata are growing in each subpopulation, and these schemata are mixed by crossover after migration in PDGA. Hence, PDGAs find good solutions quickly. The search mechanism of DGAs is different from that of SPGAs. Therefore, it is important to adjust the crossover rates and to develop a new crossover operator for PDGAs.

Two terms about migration are introduced here: a native, which is an individual in the destination subpopulation of the migration, and a hybrid, which is an individual generated from a native and a migrant.

3 Effect of Crossover in PDGAs

Crossover is a genetic operator that produces new individuals by recombining a parent's chromosomes. Crossover is employed to perform direct information exchange between individuals in a population. Therefore, the performance of GAs depends on the crossover operator implemented and a good choice of crossover rate.

From the viewpoint mentioned before, the role of crossover in PDGAs can be considered as follows: (1) to raise good schemata in each subpopulation, (2) to recombine the schemata between the immigrants and the natives. Thus, the role of crossover is different before and after migration. In this section, we investigate the crossover operators and rates that enhance these two roles.

3.1 Test functions

In this paper, the performance of the crossover operator is examined with four standard test functions, as shown in Table 1. The optimization problems used here are the minimization of the Rastrigin function (F1), the Schwefel function (F2), the Griewank function (F3), and the Ridge (F4) function with 10 design variables.

The Rastrigin function has many local minima, but it has no epistasis among its variables. The Schwefel function also has many local minima, and it has a global minimum at one of the four corners in a 2-dimensional case. The Griewank function has very small but numerous minima around the global minimum, although it has a unimodal shape in a large scale. This function has an intermediate epistasis among its variables. The Ridge function is a unimodal function, but it has a strong epistasis among its variables.

For these functions, one design variable is represented by 10 bits, 10 design variables make the length of the chromosome 100 bits. The variables in the chromosome are encoded by the Gray code. The roulette selection and elite preservation strategy are used.

3.2 Effect of crossover operator

3.2.1 Crossover operator

The behavior and effect of crossover depends on the crossover operator implemented. The condition of the crossover suitable for PDGAs is that it can preserve and recombine the schemata of parent individuals. In this section, three crossover operators – one-point crossover (1X) [1], two-point crossover (2X), and uniform crossover (UX)[10] – are compared. 1X and 2X tend to prevent the schemata of parents, and UX tends to disrupt that. The detailed features of these crossovers are shown in [5].

Table 1: Test functions

$$F1 = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$$
$$x_i \in [-5.12, 5.12], n = 10$$
$$F2 = \sum_{i=1}^{n} -x_i^2 \sin\left(\sqrt{|x_i|}\right)$$
$$x_i \in [-512, 512], n = 10$$
$$F3 = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n}\left(\cos(\frac{x_i}{\sqrt{i}})\right)$$
$$x_i \in [-512, 512], n = 10$$
$$F4 = \sum_{i=1}^{5}\left(\sum_{j=1}^{i} x_j\right)^2$$
$$x_i \in [-64, 64], n = 10$$

3.2.2 The effect of crossover operators

The first experiment compares the effect of these crossover operators on F1 through F3 shown in Table 1. The population sizes are 400 and 800, respectively. The crossover rate is 0.6, and the mutation rate is $1/L$, where L is the length of the chromosome. The number of subpopulations is set to 8, migration rate is 0.3, and migration interval is 5 generations. Each GA runs 20 times for 2000 generations.

Table 2 shows the number of generations where the optimum solutions are discovered. When the optimum is not discovered, the fitness values at 2000 generations are shown. These results are the averages of 20 trials.

From this table, there are the same tendencies on the performances of the different crossover operators for all functions. PDGAs perform better than SPGAs. In PDGAs, 1X and 2X perform better than UX. 1X and 2X tend to preserve the schemata of the parent individuals. However, UX tends to disrupt schemata and cannot exploit the information of the previous generation. PDGA, UX disrupts the schemata of the migrants, and the effect of migration is decreased.

3.3 The effect of crossover rate

3.3.1 Crossover rate

Empirical studies have shown that the best setting for the crossover rate for SPGAs depends on the choices made regarding other aspects of the overall algorithm. Examples of other parameters are population size, mutation rate, and the selection operator used. Some commonly used crossover rates vary between 0.45 and 0.95[7]. Tuson & Ross[6] carried out an exhaustive search of the operator probabilities. Their results show that the most suitable crossover rates also depend on the problem to be solved. However, the effect of crossover rate in DGAs has not yet been clarified, and the optimal value for crossover rate has not been determined.

3.3.2 The effect of crossover rate

The purpose of the experiment in this section is to examine the optimal crossover rate in PDGAs and SPGAs. The crossover rates used in this experiment are 0.3, 0.6 and 1.0, and population sizes are 160, 400 and 800, respectively. The one-point crossover is used. In DGA, the number of subpopulations is set to 8, migration rate is 0.3, and migration interval is 5 generations.

Table 3 shows the fitness values at the 1000 generations and the number of generations where the optimum solution is discovered. These results are the averages of 20 trials.

Optimal crossover rate for the SPGA depends on the objective function and the population size. However, the crossover rate of 1.0 is the best for each objective function and each population size.

Figure 2 shows the histories of the fitness values for the population size of 400. In the SPGA, the higher crossover rate performs well in the beginning of the search, but not in the later stage. Optimal crossover rate in the SPGA is depends on the stage of the search. In the PDGA, the high crossover rate shows good performance at all stages. These results indicate that the search ability in the PDGA is increased by the crossover after migration. Thus, it is reasonable to set the crossover rate to be 1.0.

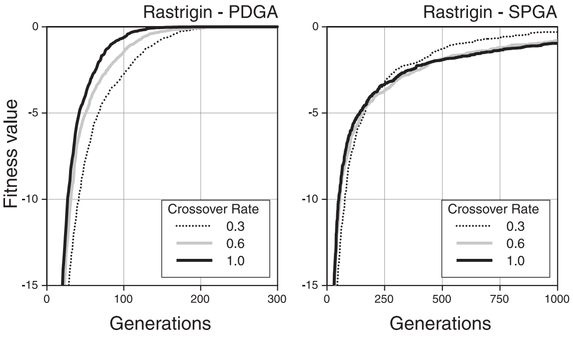

Figure 2: History of the fitness value(PDGA / SPGA)

4 Designing Crossover Operators Suitable for PDGAs

In the section above, it is considered that the search mechanism of PDGAs is different from that of SPGAs. In this section, we propose two new crossover operators that enhance the search mechanism of PDGAs: best combinatorial crossover (BCX), and hybridization crossover (HX).

4.1 Best Combinatorial Crossover

4.1.1 The design concept of BCX

In order to increase the performance of PDGA, it is important to grow good schemata in each subpopulation. In this case, premature convergence does not

Table 2: The effect of the crossover operators : The number of generations(#) where the optimum is discovered or the fitness value at the 1000 generation

	PDGA			SPGA		
	1X	2X	UX	1X	2X	UX
Rastrign	#386	#420	#804	-0.206904	-0.150967	-2.6451
Schwefel	#370	#373	#708	#1228	#1235	-0.00591232
Griewank	-0.100508	-0.137165	-0.179465	-0.286196	-0.327726	-0.329169
Ridge	-0.000935	-0.000575	-0.00435	-0.102055	-0.098925	-0.229515

Table 3: The effect of the crossover rates: The number of generations(#) where the optimum is discovered or the fitness value at the 1000 generation

	PDGA			SPGA		
	$P_c = 0.3$	0.6	1.0	$P_c = 0.3$	0.6	1.0
Rastrign	#220	#197	#166	-0.302925	-0.79939	-0.956702
Schwefel	#214	#174	#168	-0.0059123	-0.0118246	-0.648429
Griewank	-0.126492	-0.103997	-0.0576392	-0.372117	-0.306758	-0.843136
Ridge	-0.0085938	-0.0046875	-0.0039063	-0.202539	-0.420031	-0.911453

have to be considered, since the divergence of individuals is maintained by using multiple subpopulations.

In section 3.2.2, the PDGAs with 1X and 2X perform better than with UX, since 1X and 2X tend to preserve and recombine the schemata of parent individuals. To increase such good performance, we propose a new crossover operator, which is called the best combinatorial crossover (BCX). The purpose of BCX is to maximize the exploitation of the schemata of parent individuals.

In BCX, all possible children are generated from parents using 1X. After evaluating these children, two of the best children are selected and survive as the members of the next generation.

Figure 3 illustrates the behavior of BCX. In this figure, the chromosomes of the parents are assumed to be (11111) and (00000) for simplicity.

The generation of all possible children is performed using 1X and shifting the crossover point one bit. From those candidate children, two children with the highest fitness values are selected, as shown in Figure 3.

We consider two types of BCX: B-BCX and V-BCX. In B-BCX, the crossover point is a point between any adjacent bits, while it is a point between any adjacent variables in V-BCX.

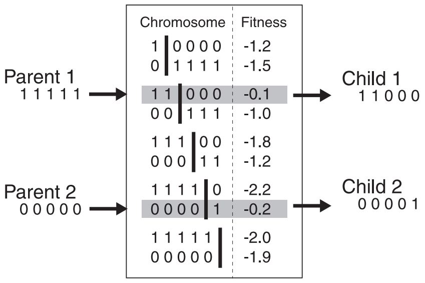

Figure 3: Best Combinatorial Crossover

4.1.2 Reduction of evaluate calculations in BCX

In BCX, a huge number of evaluations are required to find the best children. Therefore, the reduction of the number of the evaluations is very important in the application of BCX. If parents have similar genes in their chromosomes, many children with the same chromosomes are produced by BCX. Such a redundancy should be eliminated to decrease the computational load. The effective way to achieve this is to make the crossover operations only between different genes with respect to the parent chromosomes.

This can be done using a flag string, where the unique chromosomes from the two parents are represented and no redundant candidate children are generated.

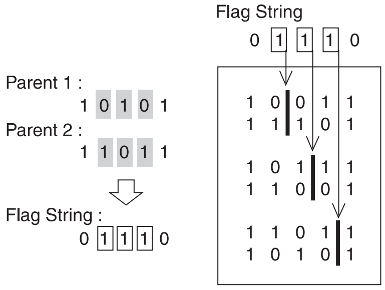

Figure 4: Generating the flag string and choosing of the crossover point

4.1.3 Experiment

To demonstrate the effectiveness of BCX, we compare BCX and 1X with four functions, shown in Table 1. The parameters used in this experiment are as follows: the population size is 400, the crossover rate is 1.0, and

Table 4: The performance of the BCX: The number of the evaluations where the optimum is discovered or The fitness values at the 2000 generations

	1X		**B-BCX**		**V-BCX**	
Rastrigin	#366	(147,040)	#37	(247,286)	#65	(178,618)
Schwefel	-0.126761	(800,000)	#28	(236,272)	#39	(178,640)
Griewank	-0.127025	(800,000)	#137	(794,261)	-0.165822	(800,000)
Ridge	-0.467969	(800,000)	#138	(764,499)	-0.03125	(793,160)

the mutation rate is $1/L$. The number of subpopulations is 8, the migration interval is 5 generations, and the migration rate is 0.5.

Figure 5 shows the histories of the fitness values on the Rastrigin function and the Ridge function, respectively. From these results, we observe that PDGA with BCX provides optimal solutions rapidly. However, this method requires a huge number of evaluations in the crossover operation, which should be taken into account when evaluating BCX. Table 4 shows the number of generations where the optimum solutions are discovered. When the optimum is not discovered, the fitness values at 2000 generations are shown. The figures in parentheses represent the number of evaluations. These results are the averages of 20 trials.

B-BCX shows the best performance on the whole, and V-BCX shows the second best performance. 1X performs well for only the Rastrigin function, but does not provide the optimum solutions for other functions.

V-BCX provides the optima faster than B-BCX for the Rastrigin and the Schwefel functions, but cannot provide any optimum for the Griewank and the Ridge functions; B-BCX provides the optima for those functions. The performance of V-BCX is very good for the functions with no epistasis, but not good for the functions with epistasis. However, the performance of B-BCX is excellent for all the functions.

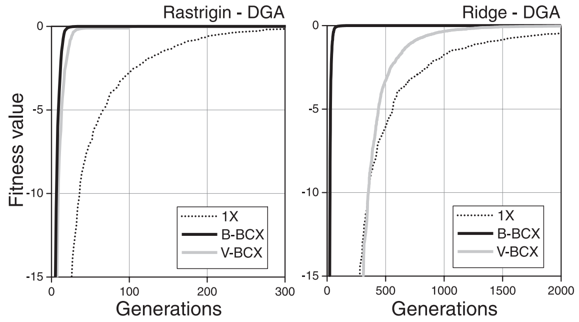

Figure 5: The history of the fitness value (Rastrigin / Ridge)

4.2 Hybridization Crossover

4.2.1 The design concept of HX

In section 3.3.2, the crossover rate of 1.0 shows a good performance in PDGAs. That is, it is important not only to grow schemata in each subpopulation, but also to recombine them by crossover after migration in PDGA. The effective recombination is provided by the hybridization between natives and immigrants. Therefore, it is considered that the performance of PDGAs can be increased by increasing the number of hybrids.

We define a new parameter called a hybridization rate, which shows the parentage of the hybrid individuals in each subpopulation.

Let μ be the migration rate, and P_c be the crossover rate. In the conventional crossover method, we can obtain the hybridization rate (H) as follows:

$$H = 2P_c\mu(1-\mu)$$

The maximum value of the hybridization rate is 0.5 when P_c=1.0 and $\mu = 0.5$, where the maximum value of μ is 0.5. That is, the hybrid individuals that play an important role in the search are generated for only 50% of the subpopulation size. Other individuals are children of immigrants or natives, which are not effective for the search.

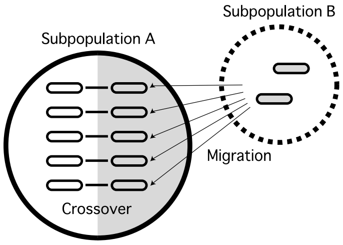

Figure 6: Hybridazation Crossover

To increase the hybridization rate beyond 0.5, we propose a new crossover scheme, which is called the hybridization crossover (HX). The purpose of HX is to mate the immigrants only with the natives in each subpopulation, as shown in Figure 6. In this scheme, the hybridization rate can be obtained as follows:

$$H = 2P_c\mu$$

The maximum value of the hybridization rate becomes 1.0 with HX, and the hybridization rate can be varied from 0% to 100%.

4.2.2 Experiment

To investigate the effect of the hybridization rate in PDGAs, the experiment for the four functions as shown in Table 1 is performed with the parameters shown in Table 5. The hybridization crossover scheme and 1-point crossover are used in the experiment. The crossover rate is 1.0, and the mutation rate is $1/L$.

Table 6 shows the result of the experiment for 8 subpopulations, and the population sizes of 160, 400, and 800. This table shows that higher hybridization rates yield better performance, especially for small population sizes. When the population size increases, the number of hybrid individuals is large regardless of the value of the hybridization rate.

Table 5: Parameters

Hybridization rate	0.2	0.5	1.0
(Migration rate)	0.1	0.25	0.5
Population size	160	400	800
The number of subpopulations	4	8	16

Table 6: The performance of HX:

Pop.	$H = 0.2$	0.5	1.0
Rastrigin			
160	#337	#296	#281
400	#197	#188	#175
800	#154	#147	#138
Schwefel			
160	#263	#336	#228
400	#186	#176	#171
800	#161	#157	#146
Griewank			
160	-0.0768976	-0.0734775	-0.0817898
400	-0.0552862	-0.0451388	-0.0479925
800	-0.0192981	-0.0129534	-0.0277259
Ridge			
160	-0.090625	-0.0789062	-0.0453125
400	-0.0101563	-0.0132812	-0.00625
800	#870	#808	#681

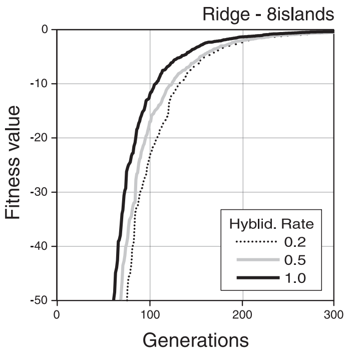

Figure 7: The historiy of the fitness value (8islands)

Figure 7 shows the histories of the fitness values of the Ridge function. It shows that the effect of the hybridization rate is remarkable in the former stage of the search, but it is not remarkable in the latter stage. In the former stage, the difference in the solutions in each subpopulation is large, and the hybrid individuals play an important role for the global search. In the latter stage, however, the difference is very small, and the hybrid individuals are very similar to the natives or immigrants. Thus the effect of HX is decreased in this stage.

5 Conclusions

In this paper, we investigated the effects of crossover operators and the crossover rate in order to improve the performance of parallel distributed genetic algorithms (PDGAs). From the results of these experiments, it is concluded that the appropriate crossover operator in PDGAs should be the one that preserves and recombines the schemata of the parent individuals, and the best crossover rate is 1.0.

Considering that the search mechanism of PDGAs is different from that of SPGAs, new crossover schemes are proposed to increase the performance of PDGAs: (1) the best combinatorial crossover that grows the good schemata in a deterministic manner, and (2) the hybridization crossover that enhances the search mechanism by generating many hybrid individuals after migration. The PDGA with proposed crossover schemes showed good performance for four test functions.

References

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

[2] Erick Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles*, Vol. 10, No. 2, 1998.

[3] Reiko Tanese. Distributed genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, pp. 434–439, 1989.

[4] Theodore C. Belding. The distributed genetic algorithm revisited. *Proc. 6th International Conference on Genetic Algorithms*, pp. 114–121, 1995.

[5] De Jong, Kenneth A., and William M. Spears. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 1992.

[6] Andrew Tuson and Peter Ross. Cost based operator rate adaptation: An investigation. *Proc. 4th Conference of Parallel Problem Solving from Nature*, 1996.

[7] Lashon Booker. *Handbook of Evolutionary Computation*, chapter 3.3:1. Oxford University Press, 1997.

[8] Ágoston Endre Eiben, Robert Hinterding, and Zbigniew Michalewics. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, JULY 1999.

[9] D.E.Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989.

[10] Gilbert Syswerda. Uniform crossover in genetic algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 2 – 9, 1989.

Proceeding of the IASTED International Conference,
Parallel and Distributed Computing And Systems
(Vol.I)
November 6-9, 2000, Las Vegas, Nevada, USA
pp. 145 - 150

Doshisha University/ Graduate School of Doshisha University, Kyoto, Japan
Intelligent Systems Design Laboratory
(http://mikilab.doshisha.ac.jp)