A Parallel Genetic Algorithm with Distributed Environment Scheme

Mika KANEKO

Abstract: This paper introduces an alternative approach to make the task of choosing optimal mutation and crossover rates easier by using a parallel and distributed GA with distributed environments. It is shown that the best mutation and crossover rates depend on the population sizes and the problems, and those are different between a single and multiple populations. The proposed distributed environment GA uses various combinations of the parameters as the fixed values in the subpopulations. The excellent performance of the new scheme is experimentally demonstrated for four standard test functions. It is concluded that the distributed environment GA is a useful method to find the best solution under a given population size and uncertainty for the appropriate crossover and mutation rates.

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, and a huge amount of computational resource is usually required in order to obtain good solutions. Therefore, there are a lot of research efforts to implement GAs on parallel computers$^{2)}$. Researches in the area of Parallel GAs can be separated into three categories: global population models, island models, and massively parallel GAs$^{3)}$. The last approach is less useful for current parallel computers since it is suitable for SIMD-type parallel computers with many small processors and the current computers are usually MIMD machines.

The global population models have a single large population where each processor is assigned to treat some individuals and each individual has global interaction with each other. The island models have multiple and smaller populations and exchange information among the subpopulations; this exchange is performed by moving some individuals from one population to another and is known as migration. In this paper, genetic algorithms with divided subpopulations and the migration operation are called distributed GAs (DGAs), and distributed GAs are called parallel DGAs (PDGAs) when they are performed on parallel computers. Genetic algorithms with a single population are called SPGAs.

DGAs have another merit in addition to the speedup. Tanese demonstrated the ability to find fitter individuals than the traditional one$^{4)}$. Belding extended Tanese's work on DGAs to the different fitness functions (the royal road problem) in order to determine whether her results were specific to the Tanese functions$^{5)}$. He showed that DGAs outperformed SPGAs.

On the other hand, the performance of each GA depends on a good choice for the crossover and the mutation rates. However, it is difficult to choose a proper combination of these parameters. Tuson and Ross$^{6)}$ showed that the most suitable crossover rates also depend upon the problem to be solved. Those results were obtained for SPGAs, and there is little research that deals with finding good choices for the crossover and mutation rates in DGA.

This paper presents the effect of those genetic parameters in PDGAs, and proposes a new approach where some of the parameters in the algorithms are not necessary to be appropriately adjusted.

2 Parallel Distributed GA

DGAs (Distributed GAs) are GAs with multiple subpopulations and a migration procedure, as shown in Figure 1. The migration is performed at a certain migration interval and some individuals are migrated from one subpopulation to another. The migration ratio is the ratio of the number of emigrants to the number of individuals in one subpopulation. Each subpopulation can be assigned to each processor of a parallel computer, and DGAs are suitable for parallel processing since inter-processor communication occurs only at the migration. In this case, the GAs are called PDGAs (Parallel DGAs). The migration topology adopted here is a ring with random destinations where each subpopulation has one destination, but the destinations are determined randomly at every migration period like in Figure 1. The emigrants are selected randomly in their subpopulation.

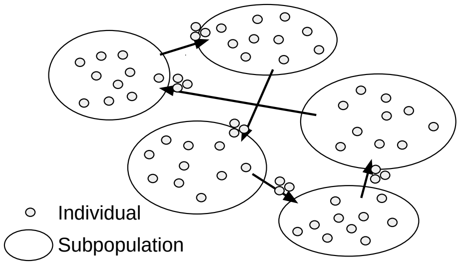


Fig. 1 Multiple subpopulation and migration

3 Determination of GA Parameters

3.1 Best GA Parameters

The roles of crossover and mutation are significantly important in GAs[1)]. Crossover is employed to perform direct information exchange between individuals in a population, while mutation is employed to avoid stagnation in evolution.

The performance of each GA depends on making a good choice for the crossover and the mutation rates. Empirical studies have shown that the best setting for the crossover rate depends on the choices made regarding other aspects of the overall algorithm, such as the settings for other parameters such as population size and 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 upon the problem to be solved. On the other hand, the best setting for the mutation rate also depends on the particular optimization problems[8)] , and the lower bound for the optimal mutation rate is found to be 1/L[9)] where L is the length of the binary strings.

However, the effect of crossover and mutation rates in DGAs has not been clarified yet, and the optimal values for such parameters have not been found yet.

3.2 Optimization Problems and GA Procedures

The effect of crossover and mutation rates in DGAs is examined with standard test functions[10)] .

The optimization problems used here are the minimization of the Rastrigin function (F_{Ra}), the Schwefel function (F_{Sc}), the Griewank function (F_{Gr}), and the Rosenbrock function (F_{Ro}) with 10 design variables.

$$\begin{aligned}
F_{\mathrm{Ra}} &= 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right) && (1)\\
&-5.12 < x_i \leq 5.12, n = 10 \\
F_{\mathrm{Sc}} &= \sum_{i=1}^{n} -x_i^2 \sin\left(\sqrt{|x_i|}\right) && (2)\\
&-512 < x_i \leq 512, n = 10 \\
F_{\mathrm{Gr}} &= 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) && (3)\\
&-512 < x_i \leq 512, n = 10 \\
F_{\mathrm{Ro}} &= \sum_{i=2}^{n}[100(x_1^2 - x_i)^2 + (x_i - 1)^2] && (4)\\
&-2.048 \leq x_i \leq 2.048, n = 10
\end{aligned}$$

The Rastrigin function has many local minima, but it has no dependency between 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 uni-modal shape in a large scale. This function has an intermediate dependency between its variables. The Rosenbrock function is a uni-modal function, but it has a strong dependency between its variables.

For the Rastrigin, Schwefel, and Griewank functions, one design variable is represented by 10 bits, and 10 design variables make the length of the chromosome 100 bits. For the Rosenbrock function, one design variable is 12 bits, and 10 design variables make the length of the chromosome 120 bits. The Gray coding is used. A standard GA with a single-point crossover is used with the maximum generations of 1000. The fixed crossover and mutation rates are used in a run, and the combinations of the crossover rates of 0.3, 0.6 and 1.0, and the mutation rates of 0.1/L, 1/L and 10/L, where L is the length of the chromosome, are used.

The results are represented by the mean values of the fitness functions of the fittest individuals over 10 trials which are extracted from 12 trials omitting highest and lowest ones. The parallel computer used is nCUBE2E with 64 processors, and one processor is assigned to one subpopulation in the PDGA.

Two experiments on the PDGA were conducted with

9 subpopulations having different subpopulation sizes, 20 and 180, respectively. The overall population sizes are 180 and 1620, respectively. Here, the notation 180 x 9 means that the subpopulation size is 180 and the number of subpopulations is 9. The migration rate is 0.3 and the migration interval is 20 generations, but these values do not have a large sensitivity to the results.

3.3 Experimental Results

Figure 2 shows the histories of the fitness value for the 20 x 9 case. When the mutation rate is small (0.1/L), the better results are obtained with the higher crossover rates, that is, the appropriate value of the crossover rate is 1.0.

When the mutation rate is very high (10/L), the best crossover rate is not clear. In PDGAs, the crossover plays not only a role of global search, but also a role of mating between migrated and native individuals. Therefore, a simple tendency is not observed here. When the crossover rate is medium (1/L), it can be seen that the high crossover rate yields good performance.

To investigate the effect of the population size on the best crossover rate on PDGA, similar numerical experiments with 1620 individuals in total were carried out with 9 subpopulations. In these experiments, the best crossover rate is 1.0 for the small and medium mutation rates and it is 0.3 for a large mutation rate.

In PDGAs, the number of individuals or the subpopulation size is small, and therefore, the diversity of individuals almost disappears, while the diversity of individuals among subpopulations are maintained in PDGAs. In this case, the crossover in each subpopulation does not play any role except for mating between migrated and native individuals. The crossover rate of 1.0 promotes the evolution of good solutions by exchanging good partial solutions developed in each subpopulation, and it provides the best performance for the small and medium mutation rates. On the other hand, the high mutation rate causes diversity in individuals even for a small subpopulation size. In this case, the high crossover rate ruins the good solutions, and the appropriate crossover rate is less than 1.0.

The best values for the crossover and mutation rates are found to be (Pc, Pm) = (0.1, 1/L) for the Rastrigin function, (1.0, 0.1/L), (1.0, 1/L) and (0.6, 1/L) for the Schwefel function, (1.0, 0.1/L) for the Griewank function, (1.0, 0.1/L) for the Rosenbrock function. Thus, the best values vary with the problems, and therefore, it takes a lot of pre-experiments to find the best crossover and mutation rates for tuning a PDGA.

4 GA with Distributed Environments

4.1 Distributed Environment Scheme

From the above results, it was concluded that to obtain the best results the crossover rate should be adjusted properly according to the mutation rate, the population size, and the number of populations as well as the problems to be solved. But, the determination of the best mutation rate and the best crossover rate is a time consuming task.

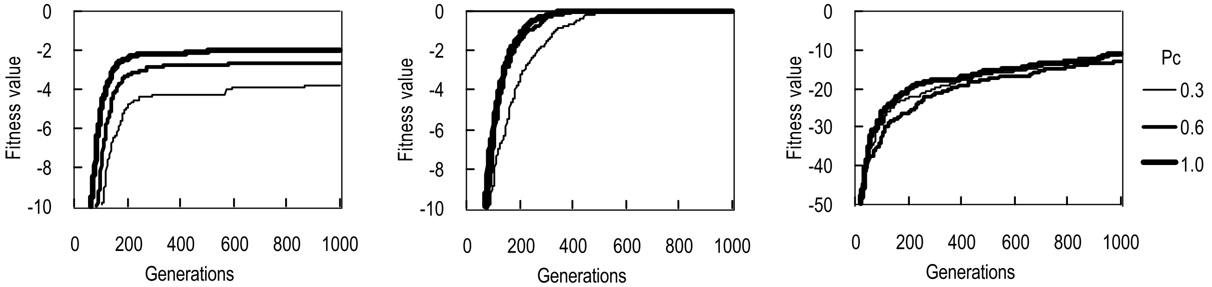
To overcome this problem, we propose a new PDGA with a distributed environment scheme. In this scheme, a whole population is divided into several subpopulations, and the GA parameters such as the mutation rate and the crossover rate in each subpopulation are different from each other. The migration operation was performed similarly to the conventional DGAs.

The distributed environment scheme is schematically shown in Figure 3, where the thermometers represent the mutation rates and the heart symbols represent the crossover rates. High temperature means a high mutation rate and a big heart symbol means a high crossover rate. This scheme is called the PDGA/DE (PDGA/Distributed Environment). With the PDGA/DE, it can be expected that the various building blocks of optimum solutions for a particular problem are evolved in the various subpopulations, and migration provides the global optimum.

The tuning of the GA parameters is not necessary with the PDGA/DE since many combinations of such GA parameters occur in many subpopulations. Consequently, it can be expected that a global optimum can be easily obtained without any pre-experiments with the PDGA/DE.

4.2 Effectiveness of PDGA/DE

To demonstrate the effectiveness of the proposed scheme, a PDGA/DE with 9 subpopulations was performed. The combination of the mutation and crossover rates in the 9 subpopulations are shown in Table 1. The comparison is made for the PDGA/DE and a PDGA/CE, where CE means constant environment and the PDGA/CE is the same as the conven-



(a) Pm = 0.1/L

(b) Pm = 1/L

(c) Pm = 10/L

Fig. 2 Effect of crossover rate in PDGA (Population size = 20 x 9)

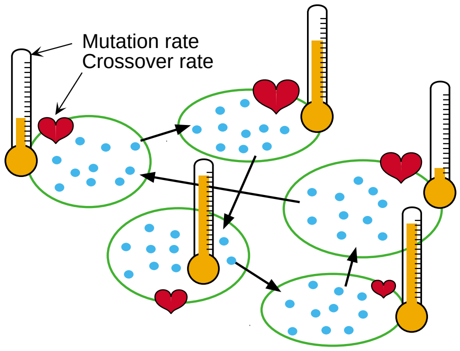


Fig. 3 PDGA/DE (Distributed Environment)

tional PDGA with constant values of crossover and mutation rates in the entire subpopulations. The computer used is a parallel computer, nCUBE2, with 64 processors, and one processor is assigned to one subpopulation.

The performance of the PDGA/DE can be seen in Figure 4 for the overall population size of 180, where the function values at 1000 generations are compared. It should be noticed that smaller values represent good solutions. This result shows the performance of the PDGA/DE compared with the results obtained by the PDGA/CE . The number of combinations of mutation rates and crossover rates is 9, as shown in Table 1, and the performance of the PDGA/DE is compared with these 9 results. It should be noticed that these fitness values are the average of the fitness of the fittest individuals over 10 trials out of 12 trials omitting the highest and lowest.

Table 1 Combinations of mutation and crossover rates

		Mutation rate		
		0.1/L	1/L	10/L
Crossover rate	0.3	0.3 - 0.1/L	0.3 - 1/L	0.3 - 10/L
	0.6	0.6 - 0.1/L	0.6 - 1/L	0.6 - 10/L
	1.0	1.0 - 0.1/L	1.0 - 1/L	1.0 - 10/L

First of all, the effect of crossover rate and the mutation rate is remarkable, and the determination of the optimal rates is necessary.

The performances of the PDGA/CE and the PDGA/DE are compared as follows. It is clear that the performance of the PDGA/DE is relatively high although it is not the best. It should be noticed that the y-axis is in logarithmic scale. The superiority of the PDGA/DE can be recognized from Figure 4. The excellent performance is surprising since it has many subpopulations with improper GA parameters. The mechanism for providing such excellent performance is not clear. One possible reason is that there is at least one best combination of GA parameters, and another possible reason is that the various environments yields better solutions than the best but the same environments. It can be concluded from these results that the PDGA/DE is a very effective method unless the optimal set of crossover and mutation rates is known.

4.3 Speedup

In order to find the efficiency of parallel processing for the PDGA/DE, the calculation time in the PDGA/DE is compared to the one in a SPGA where the population size is the same as the total population size (50 x 9) of the whole subpopulations, and the best crossover and mutation rates are used among the values listed before. Figure 5 shows the speedup for the four optimization problems, where (1) is obtained

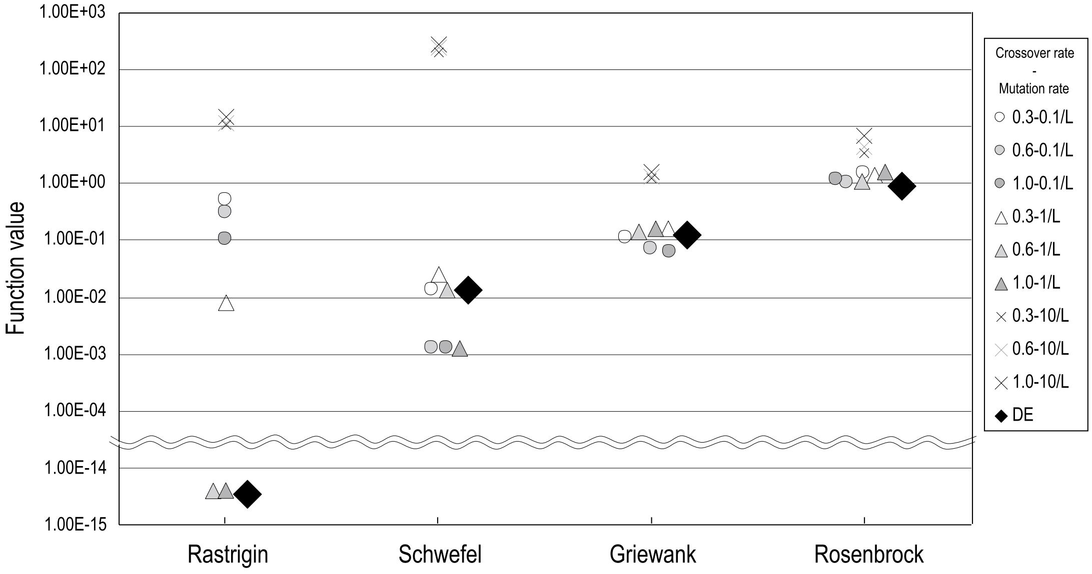


Fig. 4 Comparison of the performance of PDGA/DE with PDGA/CE

at 1000 generations, and (2) is obtained at the same quality of solutions.

From this figure, the values of the speedup for the same generations are approximately 8.6 and they are similar to the ideal speedup which is 9 since 9 processors are used. It can be recognized that the PDGA/DE can provide good parallel efficiency since the inter-processor communication occurs only when the migration is performed.

On the other hand, the values of the speedup for the same quality of solutions are between 22 to 25 except for the Rosenbrock function. This surprising speedup is due to the increase in the performance of the PDGA/DE, that is, the PDGA/DE provides good solutions 2.6 to 2.9 times faster than the SPGA. The histories of the fitness values for the PDGA and the SPGA are shown in Figure 6, and it is clear that the PDGA/DE outperforms the SPGA with best crossover and mutation rates.

For the Rosenbrock function, the PDGA/DE does not outperform the SPGA with the best crossover and mutation rates. This is because the Rosenbrock function has a strong interaction between its variables, and PDGAs do not show good performance for such problems. However, the performance of the PDGA/DE ranks third among the nine results of the SPGA on various combinations of crossover and mutation rates.

As a result, the PDGA/DE shows approximately a linear speedup with the number of processors and it provides a remarkable increase in GA performance without any consideration on the appropriate values of the crossover and mutation rates.

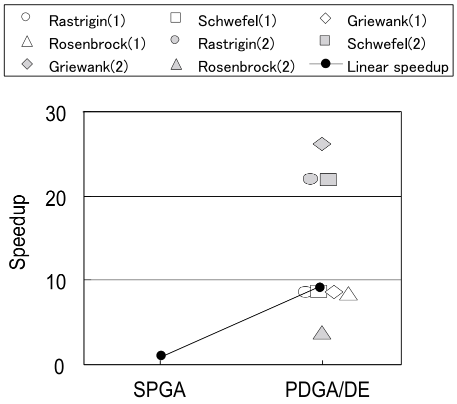


Fig. 5 Computational speedup

5 Conclusions

The effect of crossover and mutation rates on the performance of GAs with a single population and multiple populations is clarified, and the optimum rates vary according to the population size and the problem to be solved. It is found that the optimal rates for a multiple population GA is different from the ones for a single population GA. Thus, the difficulty in determining the optimal set of crossover and mutation rates exists also in parallel distributed GAs.

A parallel distributed GA with distributed environments (PDGA/DE) is proposed, and the superiority of

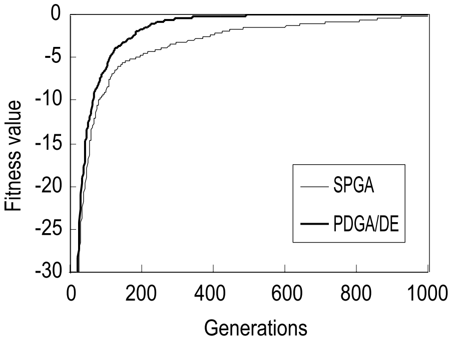


Fig. 6 Histries of the fitness for SPGA and PDGA/DE (Rastrigin)

this scheme is experimentally proved. For four different types of problems, the PDGA/DE shows almost best performance in comparison with a single population GA having the optimal crossover and mutation rates, and it shows relatively high performance in comparison with a multiple population GA having the optimal crossover and mutation rates. Consequently, the PDGA/DE is the fastest way to gain the best solution under the given population size and uncertainty of the appropriate crossover and mutation rates.

参考文献

1) J.H.Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975, MIT Press, 1992.

2) Marco Dorigo & Vittorio Maniezzo. Parallel genetic algorithms: Introduction and overview of current research. Parallel Genetic Algorithms: Theory and Applications, pp. 5-42, IOS Press, 1993.

3) Gordon, V. S., Whitley, D. and Boehm, A. P. W. Dataflow Parallelism in Genetic Algorithms. *Parallel Problem Solving from Nature, 2, Elsevier Science Pub*, pp.533–542, 1992.

Calculateurs Paralleles. Vol. 10, No. 2. Paris: Hermes, 1998

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

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

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

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

8) Hinterding, R., Gielewski, H. & Peachey, T.C.: The Nature of Mutation in Genetic Algorithms. *Proc . 6th Int. Conf. on Genetic Algorithms*, pp. 65-72, 1995.

9) Bäck, T.: Optimal Mutation Rates in Genetic Search. *Proc. 5th Int. Conf. on Genetic Algorithms, Morgan Kaufmann*, pp. 2-8, 1993.

10) Whitley, D., Mathias, K., Rana, S. and Dzubera, J.: Building Better Test Functions. *Proc. 6th Int. Conf. on Genetic Algorithms, Morgan Kaufmann*, pp. 239-246, 1995.