A New Model of Parallel Distributed Genetic Algorithms for Cluster Systems: Dual Individual DGAs

Abstract- A new model of parallel distributed genetic algorithm, Dual Individual Distributed Genetic Algorithm (DuDGA), is proposed. This algorithm frees the user from having to set some parameters because each island of Distributed Genetic Algorithm (DGA) has only two individuals. DuDGA can automatically determine crossover rate, migration rate, and island number. Moreover, compared to simple GA and DGA methods, DuDGA can find better solutions with fewer analyses. Capability and effectiveness of the DuDGA method are discussed using four typical numerical test functions.

KeyWords: *Parallel Processings, Distributed Genetic Alogrithms, Optimization, PC cluster system*

1 INTRODUCTION

The genetic algorithm (GA) (Goldberg, 1987) is an optimization method based on some of the mechanisms of natural evolution. The Distributed Genetic Algorithm (DGA) is one model of parallel Genetic Algorithm (Tanese, 1989; Belding, 1995; Miki, et. al., 1999). In the DGA, the total population is divided into sub-populations and genetic operations are performed on several iterations for each sub-population. After these iterations, some individuals are chosen for migration to another island. This model is useful in parallel processing as well as in sequential processing systems. The reduced number of migrations reduces data transfer, so this model lends itself to use on cluster parallel computer systems. Moreover, DGA can derive good solutions with lower calculation cost as compared to the single population model (Gordon and Whitely, 1993; Whitley, et. al., 1997).

Genetic Algorithms (GA) require user-specified parameters such as crossover and mutation rates. DGA users must determine additional parameters including island number, migration rate, and migration interval. Although Cantu-Paz (1999) investigated DGA topologies, migration rates, and populations, the problems relating to parameter setting remain.

The optimal number of islands was investigated for several problems and it was found that a model with a larger number derives better solutions when the total population size is fixed. Using this result, a new algorithm called Dual Individual Distributed Genetic Algorithm (DuDGA) is proposed. In DuDGA, there are only two individuals on each island. Since DuDGA has only two individuals per island, crossover rate, migration rate, and island number are determined automatically, and the optimum solution can be found rapidly. The capability and effectiveness of DuDGA and its automatic parameter setting and lower calculation cost are discussed using four types of typical numerical test functions. The results are derived using a sequential processing system.

2 DISTRIBUTED GENETIC ALGORITHM

Canonical GAs, (CGAs) make use of a single population. In distributed genetic algorithms (DGAs), the total population is divided into sub-populations called "islands." In DGAs, CGAs operations are performed on each island. After a specified number of iterations (migration interval), some individuals are selected to move from one island to another. This operation is called "migration." The period of migration, or "migration interval," can be either fixed or selected randomly. The number of individuals selected to migrated is determined along with the migration rate. Migration period and migration rate are DGA variables. The conceptual DGA flow is shown in Figures 1 and 2.

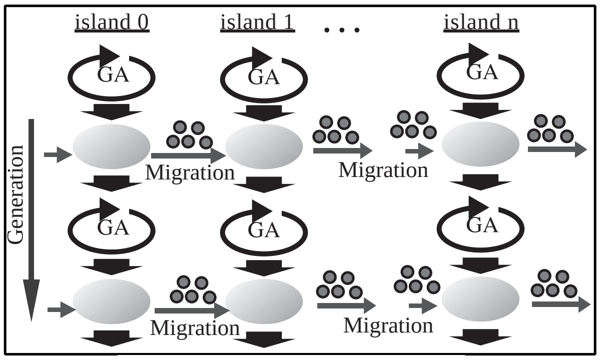


Figure 1: Flow of DGA

Several DGA models exist. The "stepping stones population" and "random migration" models are typical (Nang, et. al., 1994). In the stepping stones population model, the target island to which the individuals migrate is determined in advance (as shown in Figure. 1). This model can be implemented on massive parallel computers. On the other hand, in random migration models, the target island is determined at random at each migration opportunity (as shown in Figure 3). This model is suited for PC cluster parallel machines. The present study uses the latter model. There are also several migration methods (Whitley, et. al., 1997; Munemoto, et. al., 1993). In one, only the island's elite individuals migrate. In another, individuals are chosen at random and selected by a tournament method (Gordon and Whitley, 1993). The present study uses the following simple migration method. The number of individuals is determined in advance by the migration rate; the population size of each island is multiplied by the migration rate. The individuals to migrate are chosen at random at each migration opportunity. In this case, whether or not an individual is elite is not considered.

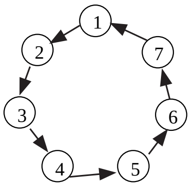


Figure 2: Stepping Stone(Ring)

In addition to their capability of being performed by parallel machines, DGAs can derive better solutions with fewer iterations compared to the CGAs. Dividing the population into sub-populations reduces each island to a size smaller than that of CGAs. Therefore, results are easily converged to local optima. Usually, fewer numbers of function evaluations are required for convergence. However, from the global point of view, because each island converged to different local optima, there are a variety of solutions. The global optimum is derived after these different local optima solutions are exchanged through the migration process.

DGAs are powerful algorithms that can derive better solutions with lower computation costs than CGA. However, DGAs have the disadvantage that they require careful selection of several parameters, such as the migration rate and migration intervals, that affect the quality of the solutions.

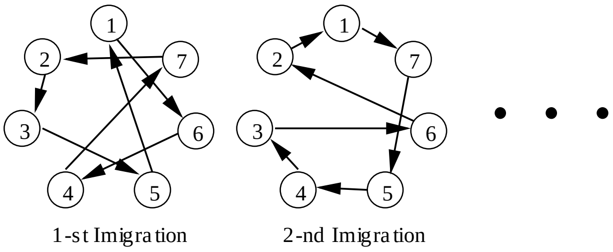


Figure 3: Stepping Stone(Random)

3 DUAL INDIVIDUAL DISTRIBUTED GENETIC ALGORITHMS

This proposed new model of Distributed Genetic Algorithms is called ”Dual Individual Distributed Genetic Algorithms” (DuDGAs). DuDGAs have only two individuals on each island. The concept is shown in Figure 4.

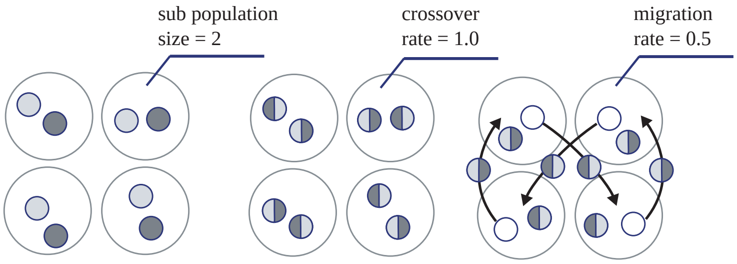


Figure 4: Dual Individual Distributed Genetic Algorithms

In the proposed DuDGA model , the following operations are performed.

- The population of each island is determined (two individuals).
- Selection: only individuals with the best fit in the present and in one previous generation are kept.
- Migration method: the individual who will migrate is chosen at random.
- Migration topology: the stepping stone method where the migration destination is determined randomly at every migration event.

One of the advantages of the DuDGA is that users are free from setting some of the parameters. By limiting the population to two individuals on each island, the DuDGA model enables the following parameters to be determined automatically:

- crossover rate: 1.0
- number of islands: total population size/2
- migration rate: 0.5

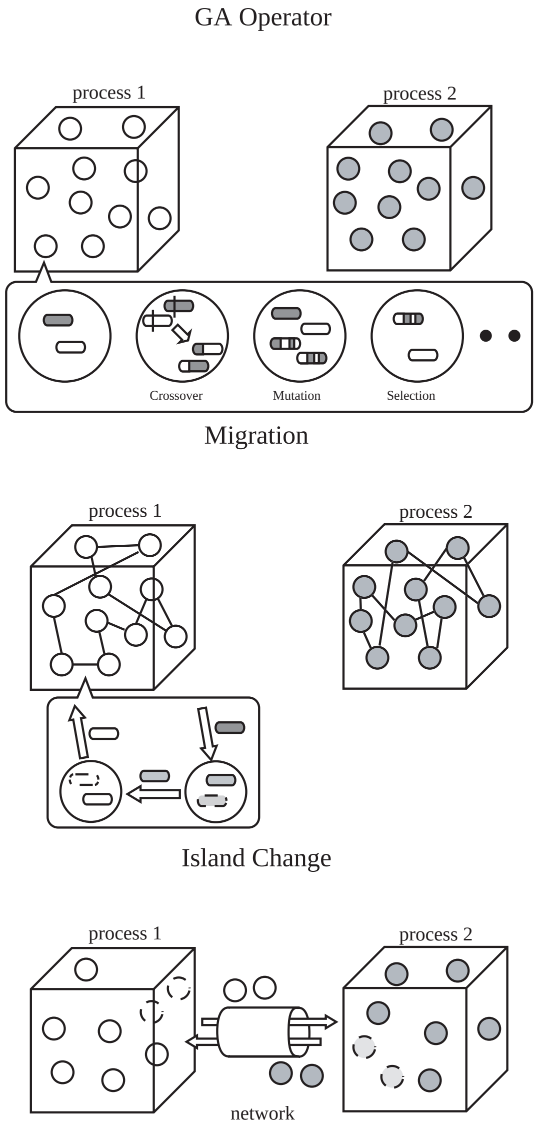
However, because each island has only two individuals, several questions arise. Does the DuDGA model experience a premature convergence problem? Even when the DuDGA can find a solution, does the solution depend on the operation of mutation? The numerical examples clarify answers to these questions. The examples also demonstrate that the DuDGA model can provide higher reliability and achieve improved parallel efficiencies at a lower computation expense than the DGA model.

4 PARALLEL IMPLEMENTATION OF DuDGA

The schematic of the parallel implementation of the DuDGA model is presented in Figure 5. This process is performed as follows:

1. The islands are divided into sub groups. Each group is assigned to one processor.
2. DuDGA is performed for each group. During this step, migration occurs within the group.
3. After some iterations, one of the islands in each group is chosen and is moved to the other group.
4. Step 2 is repeated for the newly formed groups.

Limiting the migration of islands between groups keeps network traffic (data transfer) at a minimum. The schematic in Figure 5 corresponds to a DuDGA implemented on two parallel processors.



GA Operator

Migration

Island Change

Figure 5: Parallel Implementation of DuDGA

5 NUMEIRCAL EXAMPLES

This section discusses numerical examples used to demonstrate the DuDGA model. The effects of the number of islands and population size on the performance of DuDGA are presented. The reliability, convergence, and parallel efficiency of the algorithm are discussed.

5.1 TEST FUNCTIONS AND USED PARAMETERS

Four types of numerical test functions (Equations 1-4) are considered.

$$
\begin{aligned}
F_1 &= 10N + \sum_{i=1}^{N} \{x_i^2 - 10cos(2\pi x_i)\} \\
&(-5.12 \leq x_i < 5.12) && (1) \\
F_2 &= \sum_{i=1}^{N} \{100(x_{2i-1} - x_{2i}^2)^2 + (x_{2i} - 1)^2\} \\
&(-2.048 \leq x_i < 2.048) && (2) \\
F_3 &= 1 + \sum_{n=i}^{10} \frac{x_i^2}{4000} - \prod_{n=1}^{N} cos(\frac{x_i}{\sqrt{i}}) \\
&(-512 \leq x_i < 512) && (3) \\
F_4 &= \sum_{i=1}^{N} (\sum_{j=1}^{i} x_j)^2 \\
&(-64 \leq x_i < 64) && (4)
\end{aligned}
$$

The number of design variables (ND), the number of bits (NB) and the characteristics of the test functions are summarized in Table 1.

Table 1: Test Functions

	Function Name	ND	NB
F_1	Rastrigin	20	200
F_2	Rosenbrock	5	50
F_3	Griewank	10	100
F_4	Ridge	10	100

It is easy for GAs to derive solutions using the Rastrigin function (F1) because it is a linear function of the design variables. Conversely, it is difficult for GAs to find solutions using non-linear functions such as the Rosenbrock (F2) and Ridge (F4) functions. The degree of difficulty in finding solutions using the Griewank function (F3) is in the range between that for F1 and F2. Table 2 summarizes the parameters specified for the DGA and DuDGA operators.

The algorithm is terminated when the number of generations is more than 5,000. Results shown are the average of 20 trials. The DGA needs several parameters which users must set. However, since the DuDGA has only two individuals in its islands, with the exception of population size and the migration interval, the parameters are automatically determined.

5.2 CLUSTER SYSTEM

In this paper, the simulations are performed on a parallel cluster that is constructed with 16 Pentium II (400 Mhz) personal computers (PCs). The specifications of this parallel PC cluster system are shown in Table 3. This cluster

Table 2: Used Parameters

	DGA	DuDGA
Crossover rate	1.0	1.0
Population size	240	240
Mutation rate	$1/L$	$1/L$
Number of islands	4, 8, 12, 24	120
Migration rate	0.3	0.5
Migration interval	5	5

L : Chromosome length

is similar to a Beowulf cluster and has normal networks. Therefore, increase in network traffic decreases the parallel efficiency.

Table 3: Speck of 16 PC Cluster System

CPU	Pentium II (400 MHz) × 16
Memory	128 Mbytes
OS	Linux 2.2.10
Network	FastEthernet Switching HUB TCP/IP
Communication library	MPICH 1.1.2

5.3 EFFECTS OF THE NUMBER OF ISLANDS

The effect of the number of islands on reliability and convergence of the DGA are discussed in this section. Reliability is the fraction of times during 20 trails that an optimum was found. The reliability of DGA for the four test functions and varying number of islands is shown in Figure 6.

Figure 6 shows that the reliability of the DGA increases with the number of islands for test functions F1, F3, and F4. F2 is a problem that GAs are not good at finding solutions. Therefore, DGAs can find good results in F2.

Figure 7 shows the number of evaluations needed to located the optimum solution. A substantial portion of the computation effort is spent in evaluating fitness functions an hence a smaller number of calls for function evaluations is desirable.

The results presented in Figure 7 indicate that the DGA requires the least number of function evaluations with highest number of islands. Hence, it can be concluded that the DGA should have as many islands as possible. DuDGA exploits this characteristic by maximizing the number of islands and minimizing the number of individuals.

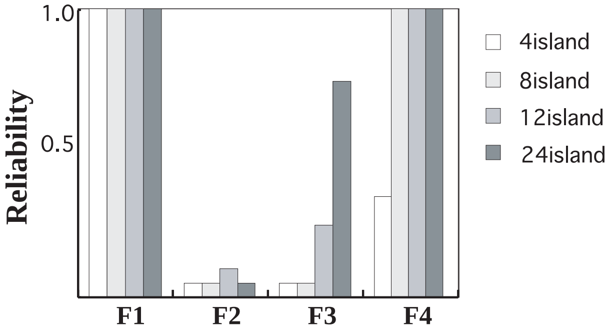


Figure 6: Reliability of DGA

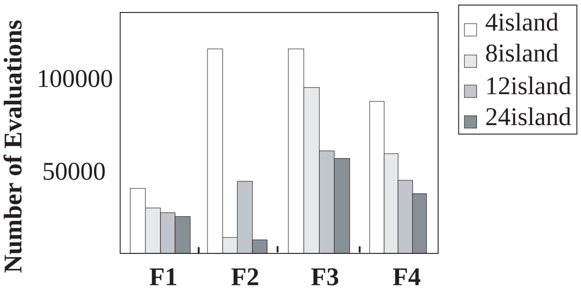


Figure 7: Number of Evaluations

5.4 EVALUATION OF DuDGA PERFORMANCE

5.4.1 RELIABILITY AND CONVERGENCE

Figures 8 and 9 show the reliability and the number of function evaluations for convergence of DuDGA.

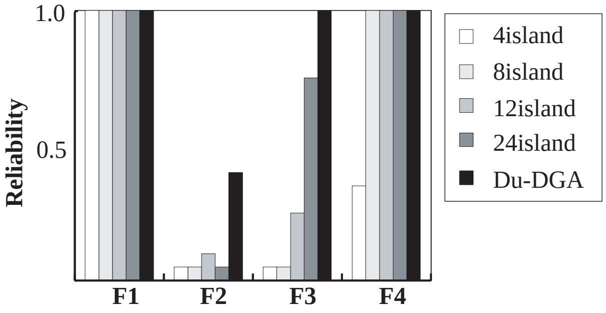


Figure 8: Comparison of Reliability of DGA and DuDGA

Figures 8 and 9 show that DuDGA exhibited higher reliability and faster convergence for all four test functions when compared to DGA. Figures 10 and 11 show the iteration histories of the objective function and hamming distance values. The Hamming distance is a measure of the difference between two strings, in this case the binary coded chromosome.

In figure 10, it is found that the evaluation values of DuDGA are not good at the first generations of the process. Then in the latter process, thoes are better than thoes of other DGAs. It can be said in the same thing in figure 11. The diversity of the solutions can be found from the hamming distances. When the hamming distance is big, the

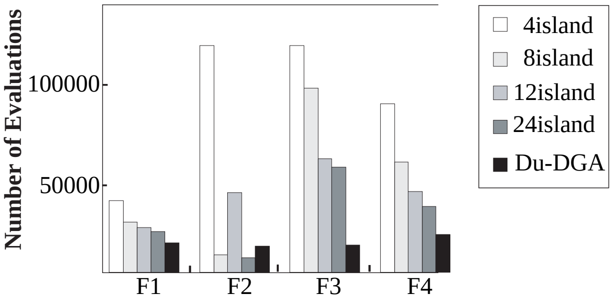


Figure 9: Comparison of Number of Function Evaluations Required by DGA and DuDGA

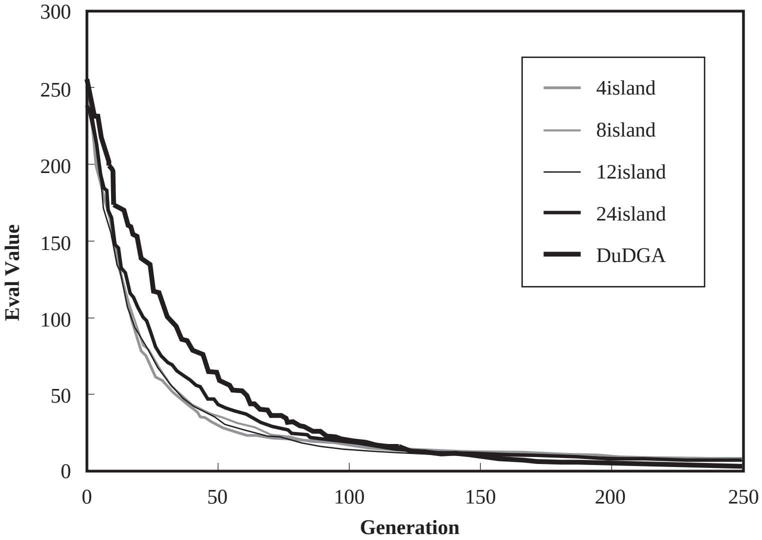


Figure 10: Iteration History of the Objective Function Value

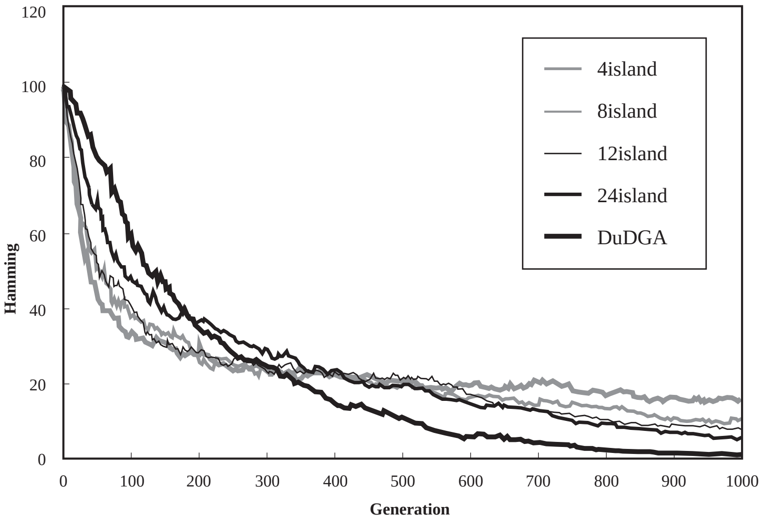


Figure 11: Iteration History of the Hamming Distance to the Optimum Solution

GA still has diversity of the solutions. In the first generations of the process, there is a high diversity in DuDGA. On the other hand, in the latter generations, solutions are converged to the point and DuDGA lost the diversity quickly. Compared to other DGAs, the convergence of DuDGA is slower during the first generations of the process. This is because the DuDGA is searching for global rather than local solutions. Later, when DuDGA is searching for local solutions, values converge quickly, and the model finds better solutions than did the other DGAs.

5.4.2 EFFECTS OF MUTATION AND CROSSOVER IN DuDGA

This section discusses the effect of the mutation and crossover operators on DuDGA performance. DuDGAs may search solutions by both crossover and mutation. However, having the DuDGA only two individuals, one may think the DuDGA is searching solutions only by mutation. Table 4 summarizes the optimum values of the four test functions obtained using the DuDGA with all operators, without mutation (Type A), and without crossover (Type B). These are the results when the simulation terminates: when the operation reaches more than 1,000 generations and are averaged over 20 trials.

Table 4: Role of Mutation and Crossover in DuDGA

Function	DuDGA	Type A	Type B
F1	0.0020	18.842	193.49
F2	0.0196	1.166	7.01
F3	0.0024	1.038	27.10
F4	0.0000	138.032	429.83

Results of from Table 4 indicate that the crossover operator is more important than the mutation operator. However, the solutions obtained with the combined operators in DuDGA are better. Therefore, it is concluded that DuDGA is searching solutions mainly by crossover.

5.4.3 EFFECT OF POPULATION SIZE

When the total population size is too small, GAs can not find optimum solutions. On the other hand, when the total population size is too big, optimum solutions are derived but computation time is wasted. Figure 12 shows the effects of population size on the number of evaluations when the Rastrigin function (F1) was used.

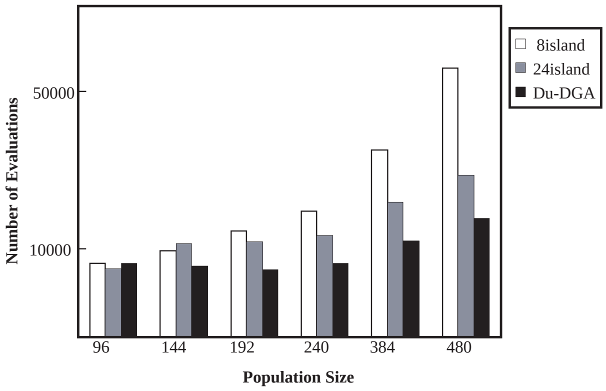


Figure 12: Effects of Population Size

From Figure 12 it appears that for a small number of islands (n=8), the population size required is small. For the case with eight islands, optimization with a population size of 96 required the least number of function evaluations. However, for the 24-island and DuDGA, the population size has an optimum between 192 and 240. Indeed, this was the value (240) used for the reliability and convergence results presented earlier.

5.4.4 PARALLEL EFFICIENCY

Calculation speed up of DuDGA for the Rastrigin function (F1) using multiple parallel processors are shown in Figure 13. Results are for fixed population size and number of islands but with varying number of groups.

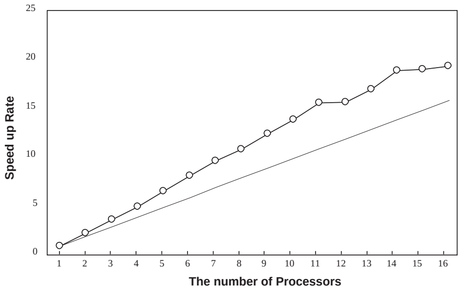


Figure 13: Speed Up (one process per one processor)

The speed up of the DuDGA implemented on multiple processors is more than linear. The speed up is much higher initially (1-10 processors) and levels off later (14-16 processors). There are two significant reasons for this high parallel efficiency. First, DuDGA limits information flow between processes (groups) thereby reducing data transfer and traffic jams. The second reason is that the total number of calculations that must be performed is reduced by distributing the computation processes.

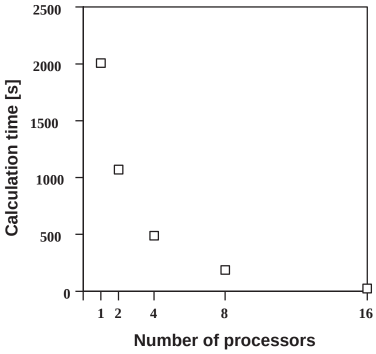


Figure 14: Computation time (16 processes)

In Figure 14, computation times of the PC cluster system are shown. These results are obtained for the 16-group DuDGA problem. When there are two processors, each processor has 8 groups. When there are 8 processors, each processor has 2 groups. As Figure 14 shows, when the number of processors increases, calculation time decreases. When the PC cluster with fewer processors than groups is used, each processor must perform several processes. This is not an efficient option when a Linux based PC cluster system is used. In order to maximize efficiency, the process threads should be parallelized or the user should use the same number of processors as groups.

6 CONCLUSIONS

This paper presents a new model of parallel distributed genetic algorithm called "Dual Individual Distributed Genetic Algorithms." The DuDGA model was applied using four typical test functions–Rastrigin, Rosenbrock, Griewank, and Ridge–to find optimum solutions on PC cluster systems. DuDGA's use of only two individuals in each island enables it to determine some GA parameters automatically. This reduces the time required to implement analyses, reduces problems associated with inadequate parameter selection, and decreases processing time. The evaluation of the method with the test cases examples leads to the following conclusions:

- When the total population size is fixed, the more islands there are, the faster the convergence. The DuDGA exploits this characteristic.
- Compared to the DGA where the number of islands is relatively small, DuDGA can derive better solutions with a smaller number of function evaluations.
- The DuDGA searches using a crossover operation; it cannot search effectively while using only the mutation operation.
- DuDGA performs global searches during the first generations. In the latter part of the analysis, convergence proceeds rapidly, and a local search is performed.
- When the population size is small, a standard GA cannot find an optimum solution due to premature convergence. When the population size is large, GA can derive an optimum solution. However, computation time is wasted. DuDGA does not waste much computational effort, even when the population size is large.
- Because of its high efficiency resulting from the reduced data transfer between groups, DuDGA is an effective method for performing Genetic Algorithms on distributed parallel cluster systems.

References

T. C. Belding (1995). The Distributed Genetic Algorithms Revised, Proc. of 6th Int. Conf. Genetic Algorithms,114-121.

E. Cantu-Paz (1999). Topologies, Migration Rates, and Multi-Population Parallel Genetic Algorithms, *Proceedings of GECCO 1999*,91-98.

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

V. S. Gordon and D. Whitley (1993). Serial and Parallel Genetic Algorithms as Function Optimizers, *Proc. of 5th Int. Conf. Genetic Algorithms* .

M. MIKI, T. HIROYASU and K. HATANAKA (1999). Parallel Genetic Algorithms with Distributed-Environment Multiple Population Scheme, *Proc. 3rd World Congress of Structural and Multidisciplinary Optimization (WCSMO)*, Vol. 1,186-191.

M. Munemoto, Y. Takai and Y. Sato (1993). An efficient migration scheme for subpopulation based asynchronously parallel genetic algorithms, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 649-159.

J. Nang and K. Matsuo (1994). A Survey on the Parallel Genetic Algorithms, *J. SICE*, Vol. 33, No.6, 500-509.

R. Tanese (1989). Distributed Genetic Algorithms, *Proc. of 3rd Int. Conf. Genetic Algorithms*, 432-439.

D. Whitley, et. al. (1997). Island Model Genetic Algorithms and Linearly Separable Problems, *Proc. of AISB Workshop on Evolutionary Computation*

High Performance Computing, Lecture Notes in Computer Science 1940,(2000), pp.374-383