

# A Parallel Genetic Algorithm with Diversity Controlled Migration and its Applicability to Multimodal Function Optimization

Fujio YAMAMOTO and Tomoyuki ARAKI

Dept. of Information and Computer Sciences,  
Kanagawa Institute of Technology  
1030 Shimoogino, Atsugi-shi, Kanagawa 243-0292, Japan  
E-mail : {yamamoto, araki}@ic.kanagawa-it.ac.jp  
Tel : +81-462-91-3181, Fax: +81-462-42-8490

## Abstract

Proposed here is a parallel genetic algorithm accompanied with intermittent migration among subpopulations. It is intended to maintain diversity in the population for a long period. This method was applied to finding out the global maximum of some multimodal functions for which no other methods seem to be useful. Preferable results and their detailed analysis are also presented.

**Keywords :** Parallel Processing, GA, Genetic Algorithm, Function Optimization, migration

## 1. Introduction

Genetic Algorithms are considered extremely powerful in the field of various optimization problems. However, the most serious defect is their non-deterministic features. That is, most parameters concerning genetic operations must be determined through trial and error. Such parameters include the number of individuals, crossing-over rate, mutation rate, selection methods for next generation individuals, and so on. More over, in most cases, a lot of trials using different sequences of random numbers don't necessarily show stable results. Some sequences of random numbers introduce to growth of uniformity in the population. This might result in premature convergence.

One of the promising method to overcome this problem might be parallelization of a genetic algorithm. It is fairly expected to maintain diversity in the population naturally for a long period. Actually, in such a method, each processor of a parallel computer system will have different subpopulation and perform genetic operations concurrently on it. The point here is to migrate some individuals from one subpopulation to another at an appropriate rate. This is a kind of imitation of the actual evolutionary process in the real world. That is, we take advantage of the process reaching the higher culture by appropriate cultural exchange. This kind of parallel genetic algorithms have been studied by several researchers including Tanese, Muhlenbein, etc. [1]-[6].

In this paper, we proposed to adopt suitable mechanism for deciding migration rate and its interval in the genetic algorithm. We also intend to apply such an algorithm to

ever difficult function optimizations that were not yet solved by previous researchers. Two benchmark functions with 2-variables and 6-variables were selected. Both functions have extremely sharp summit and steep slope, so that no numerical methods nor canonical genetic algorithms might be useful.

The parallel computer used for realizing such algorithms was the HITACHI SR2201. The proposed parallel algorithm using eight processors on this machine is investigated concerning diversification in subpopulations and the preciseness of the numerical results obtained.

## 2. Implementation of a parallel genetic algorithm

### 2.1 General considerations

Genetic algorithms can be seen as a kind of parallel random search in their nature. By some mechanism, individuals mate each other and produce offspring. This mating or crossing-over occurs concurrently in the population. But the selection operations for next generation might be performed with a centralized control, because global computations of the average and the maximum of fitness will be required. So that, it is difficult to perform genetic algorithms directly on a parallel computer efficiently [2][7].

In the parallel genetic algorithms, genetic operations accompanied with these selection operations must be parallelized. In order to do this, the whole population should be divided into subpopulations and then they are

allocated to each processor. Then evolution within the subpopulation are performed independently with each other. Such a distributed evolution keeps the diversity of the whole population high. But if this process continues for a long time, probability of falling into premature convergence becomes also high. It is not impossible to take advantage of such a parallelism. If the desired level solution is found in one of the subpopulations, whole computation would be stopped. In this way, one would get an appropriate solution. But this is merely equivalent to running a serial genetic algorithm repeatedly.

## 2.2 PGA with intermittent migration(PGA-IM)

In order to construct more natural and genuine parallel algorithm, we need some migration mechanism. That is, information exchange should be taken among subpopulations in an appropriate interval. In the concrete, some individuals move into another subpopulation and they make mating there. The selected individuals for migration should be of high ranking in their native subpopulation. They evolved in a different way from the new subpopulation. So that, offspring produced by them and the natives in that subpopulation would get different promising feature. This will result in new promising search area. Such landscape would never be obtained by mutation operations.

In order to take advantage of migration, proper control of the rate and the interval of it is required. Basically in our proposed algorithm, predetermined migration interval ( $M_{intv}$ ) is used. In addition to this, migration occurs when the standard deviation of fitness becomes smaller than a threshold value ( $F_{th}$ ). The number of individuals migrating to their neighbor subpopulation is determined by the rate  $M_{rate}$ . After migration, canonical serial

genetic algorithm is taken place until next time migration.

The serial genetic algorithm used here is a popular one. It is composed of (1)initialization of individuals in the population, (2)crossing-over, (3)mutation, (4)fitness evaluation, and (5)selection for next generation. These steps are repeated until reaching specified generation count. The crossing-over is done at a single locus. The function value is directly used as fitness. The selection strategy is based both on elite preserving and ranking selection. Suppose that the current number of individuals is  $n$ , and the number of new offspring produced by crossing-over is  $m$ . The  $m+n$  individuals are sorted according to their fitness value. In the next generation,  $n$  individuals will survive at the maximum. The survival probability of the  $s$ -th ranking individual ( $s=0$  for the top) is given as  $(m+n-s)/(m+n)$ . Therefore, the lowest ranking individual will still survive at the probability of  $1/(m+n)$ .

## 2.3 Outline of the parallel computer

The machine used here was the HITACHI distributed memory parallel computer SR2201. As shown in Fig.1, it is composed of 16 RISC processors, whereas 2 processors out of them are reserved for I/O. In the following experiment, remaining 8 processors are used. The peak performance of each processor is 300 Mflops and the peak data transmission rate among processors is 300 Mbytes/sec. Each subpopulation is allocated to each processor. Migration is performed using this data transmission feature. We see that these processors are connected via one-way torus. Individuals may move from  $i$ -th processor to  $(i-1)$ -th processor. Exceptionally, individuals in the 0-th processor may move to 8-th processor.

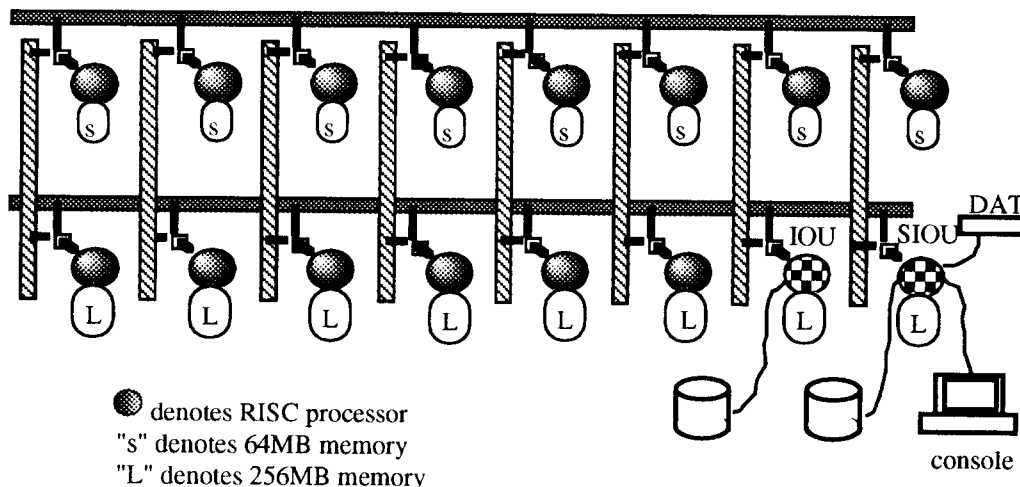


Fig. 1 The Parallel Computer (HITACHI SR2201)

### 3. Multimodal function optimization problems

Two kinds of multimodal function optimization problems are considered here.

#### 3.1 The DeJong's function F5

The first is the 2-variables function called DeJong's F5 and is used widely as a benchmark problem. It is formulated as follows:

$$f(x, y) =$$

$$\left[ 0.002 + \sum_{j=1}^{25} \frac{1}{j + (x - a_j)^6 (y - b_j)^6} \right]^{-1} \dots\dots\dots (1)$$

In this formula,  $a_j$  and  $b_j$  determine the locations of the summits. The domain of x-axis and y-axis are both defined as [-65.535, 65.536] and there exist 25 summits. All the summits have different height. The highest summit is located at [-50.0, -50.0]. It is generally difficult to find out this highest peak because canonical genetic algorithms tend to converge to other lower peaks and there are no way to escape from them through the valley.

The representation of genotype for solution (x, y) is determined by using 34 bits. Because of the domain of x and the required numerical precision of 0.001 for x-axis stepping, a sequence of zeros of 17 bits corresponds to the value -65.535. Also one bit corresponds to a delta x. For the y-axis, the same representation is used. So, the combined 34 bits represents a solution (x, y).

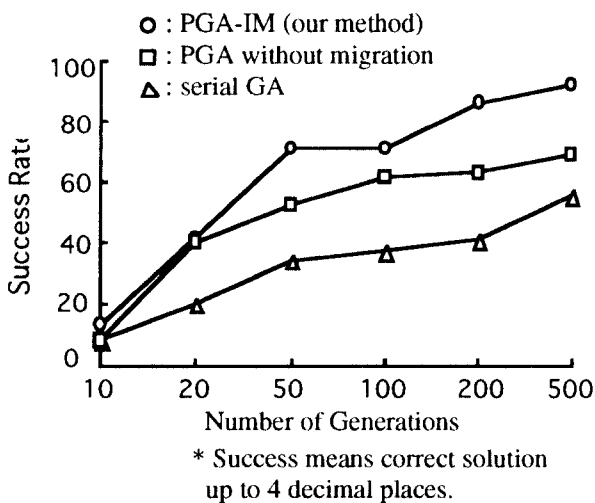


Fig.2 A Hundred Trials for the DeJong's Function F5

Some results for this problem from our parallel algorithm PGA-IM (the Parallel Genetic Algorithm with Intermittent Migration) are shown in Fig.2 comparing with others. This figure illustrates the number of successes in a hundred runs up to the 500 generation. The success means reaching exact solution up to 4 decimal places. Our PGA-IM shows preferable success rate of 93% at 500-th generation. Here we set 400 individuals and equally divided them onto 8 processors. In all of the runs, the crossing-over rate is set to 0.5 and the mutation rate is set to 0.05.

#### 3.2 The Tsuda's exponential function

The second one is called the Tsuda's 6-variables function and is defined as follows:

$$f(x_1, x_2, \dots, x_6) = \prod_{i=1}^3 \{ \exp[-3\{x_{2i-1}^2 + (x_{2i} - 1)^2\}] + \exp[-\{(x_{2i-1} - 1)^2 + x_{2i}^2\}] \} \dots\dots\dots (2)$$

The domain of all the variable is set to [-0.5, 1.5]. This function is a combination of 6 exponential functions and is considered quit difficult to find out the global maximum due to its steep slopes. Tsuda [9] proposed interactive method using graphical interface, and obtained some good solution.

The length of the chromosome used with the PGA-IM is 150 bits (25 bits x 6 variables). This is due to the domain size described above and the required numerical precision of 7 digits. For each subpopulations 500 individuals are initially assigned. In the Table1, complete numerical solutions for this problem from our PGA-IM are shown comparing with other methods. The PGA-IM gives a correct global maximum coinciding with the exact value up to 5 decimal places. The values of each variables corresponding to that function value are also correct up to 4 to 5 decimal places. In this case the crossing-over rate was set to 0.8. On the other hand, the mutation rate is varied among subpopulations, but their values were suppressed in very small values between 0.01 to 0.05.

The preferable results shown above are not accidental one. In Fig.3, results from another 30 runs with varied random number sequences are also presented. In the figure, (a)serial parallel algorithm, (b)parallel genetic algorithm without migration, and our (c)PGA-IM are compared. Solid small circles denote the generation number where the correct function value up to 3 decimal places are obtained. Similarly, the white circle and the triangle

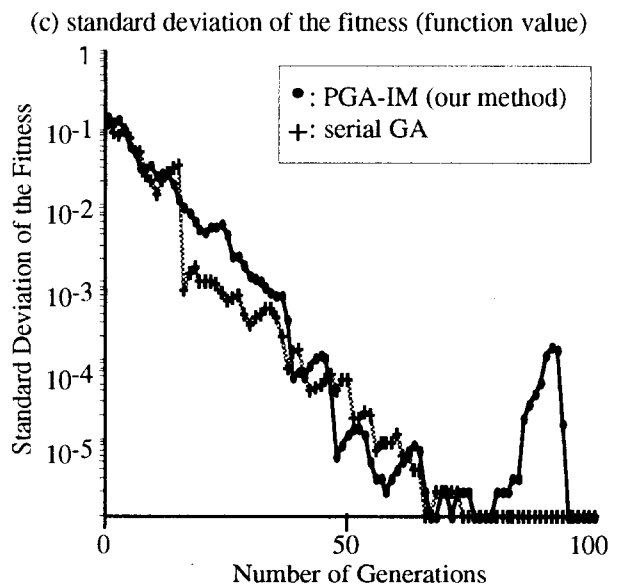
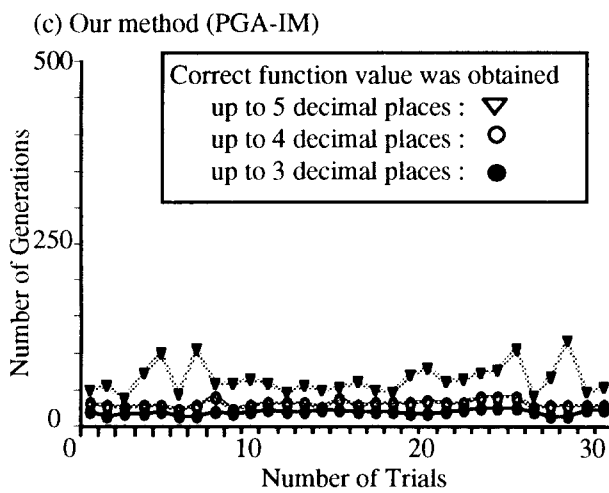
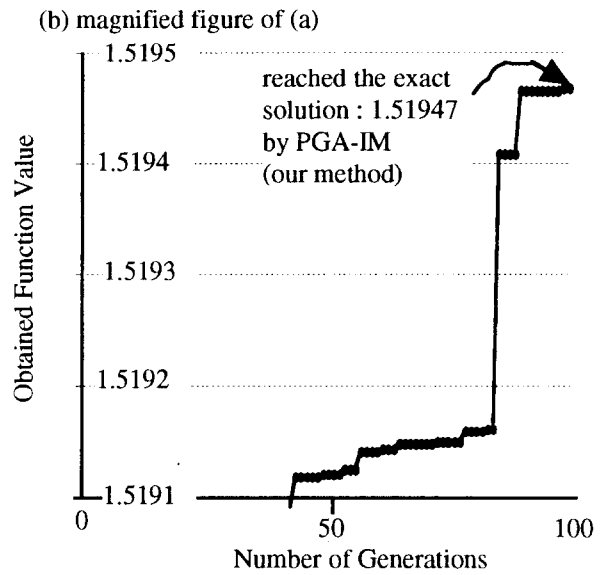
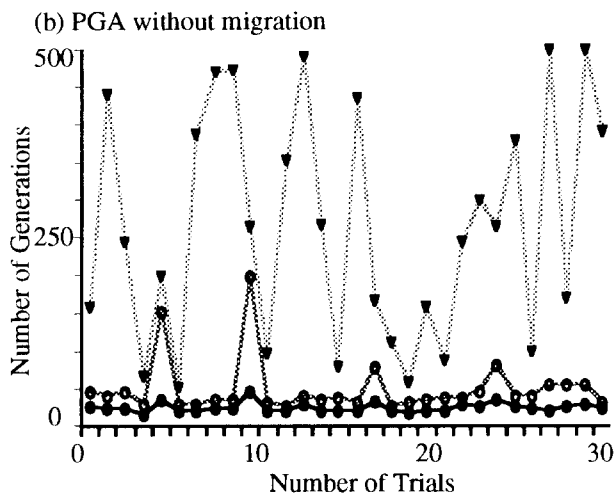
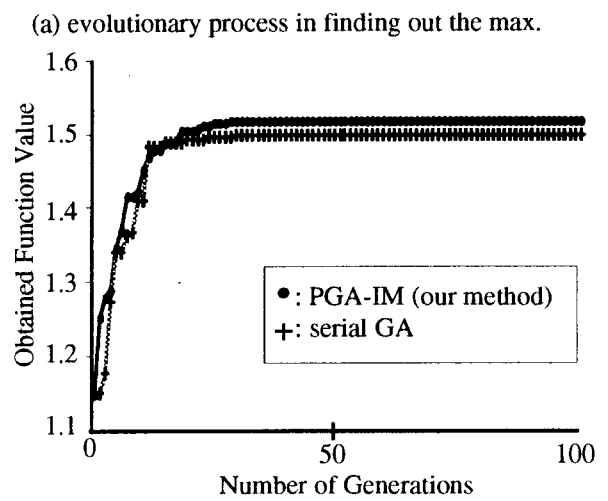
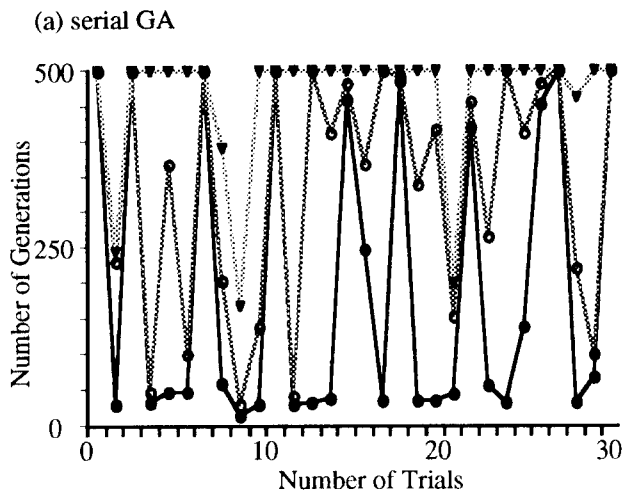


Fig. 3 Results from the Tsuda's Six-variables Function

Fig. 4 Evolutionary process in finding out the max. of Tsuda's function

Table 1 Solutions for the six-variables Tsuda's function optimization obtained by various methods

	$f(x)$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
exact value [9]	1.51947	0.053495	0.946505	0.053495	0.946505	0.053495	0.946505
PGA-IM (our method)	1.51947	0.053501	0.946502	0.053495	0.946501	0.053497	0.946503
GA by Fukushima [8]	1.51838	0.05467	0.93749	0.06248	0.95312	0.05467	0.93749
interactive method [9]	1.51946	0.05471	0.94699	0.05227	0.94645	0.05341	0.94672

denote the correctness up to 4 decimal places and 5 decimal places respectively. One can observe that our PGA-IM always shows high precision numerical solutions without interference from varied random number sequences.

#### 4. The effects of migration on diversification

The good results from our PGA-IM shown above come from the intermittent migration. In the above 30 runs, top 5% individuals migrated to the neighboring subpopulation every 10 generation in the average. The actual effect of this is shown in Fig.4. The small solid circle and the small plus symbol denote the PGA-IM and the serial genetic algorithm, respectively. Figure (a) shows the fitness, i.e. the function value up to 100 generations. Figure (b) is a magnification of the figure (a). With the PGA-IM, the exact solution (1.51947) was obtained at 94-th generation. On the other hand, figure (c) shows standard deviation of the fitness in the subpopulation. Along with the progress of generation, the standard deviation becomes small and all the individuals tend to converge to some optimal. Whereas, it is observed that the standard deviation shows fluctuation caused by migration. This continuous fluctuation finally reveal its implicit effects at near 80-th generation. That is, new promising search area was explored. Actually, in the succeeding several generations, the numerical precision of the solution was improved by the factor of 2 decimal digits.

#### 5. Conclusion

A kind of parallel genetic algorithm with intermittent migration was proposed and applied to function optimization problems. It was successful in getting very precise solution quite effectively. The migration interval was controlled based on the standard deviation of fitness. This method proved clearly effective. Another point is that this method showed favorable robustness against changing

sequences of random numbers used in the algorithm. We plan to apply this method to more attractive and practical problems in the near future.

#### References

- [1] E.G. Talbi and P. Bessiere : A Parallel Genetic Algorithm Applied to the Mapping Problem, SIAM NEWS, July, 1991
- [2] Heinz Mulenbein: Evolution in Time and Space - The Parallel Genetic Algorithm, Foundations of Genetic Algorithm, Morgan Kaufmann Publishers, pp. 316-337, 1991
- [3] R. Tanese: Distributed Genetic Algorithm, Proc. of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, pp. 434-440, 1989
- [4] B. Manderick and P. Spiessens: Fine-grained Parallel Genetic Algorithm, Proc. of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, pp. 428-433, 1989
- [5] J. Koza : Scalable Learning in Genetic Programming using Automatic Function Definition in Advances in Genetic Programming, K.E. Kinnear, Ed. Cambridge, MA, MIT Press, 1994
- [6] H. Juille and J. B. Pollack : Massively Parallel Genetic Programming, in Advances in Genetic Programming II, P. Angeline and K. Jr. Kinnear, MIT Press, 1996
- [7] Yasuo Yonezawa : Genetic Algorithms, Morikita Publinsing Co., 1993 (in Japanese)
- [8] Makoto Fukushima : Search for Global Maximum of a Function of Many Variables using Gentic Algorithm, Trans. IEICE, Vol. J79-D-II, No. 6, pp. 1169-1171, 1996 (in Japanese)
- [9] Takao Tsuda : The Monte Carlo method and Simulations, Bai-fu-kan, 1977 (in Japanese)