ASYNCHRONOUS MIGRATION OF ISLAND PARALLEL GA FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEM

Hirosuke Horii, Mitsunori Miki, Takayuki Koizumi, and Nobuyoshi Tsujiuchi

Doshisha University
Kyotanabe, Kyoto, 610-0321, JAPAN

ABSTRACT

In this paper, we propose a new scheme of asynchronous migration of the island parallel GAs for multi-objective optimization problems. We investigate the feature of our algorithms by applying to four test problems which have variety shapes of pareto-optimal front.

1. INTRODUCTION

Multi-objective optimization problem (MOP) has plural objective functions and requires to search a set of solutions called *pareto-optimal solutions*. Since Genetic Algorithm (GA) searches multi-point simultaneously, GA is suitable for MOP. Multi-objective optimization using GA is called Multi-objective GA (MOGA). The difference between MOGA and single objective optimization GA (SOGA) is that SOGA allows a set of solutions converging to an attractor around the optimal solution, but MOGA requires a set of solutions distributing widely on pareto-optimal front. So MOGA has to maintain the diversity of its population to the end of the search.

The island parallel GA is one of parallelization method of GA which divides a population into plural subpopulations and assigns them to processing elements on a parallel computer. Each subpopulation searches for optimal solution independently, and maintains diversity of genes by exchanging individuals periodically with certain conditions called *migration* . The setting of migration operation is the most important for the island parallel GA. To implement this migration operation, there are two possibilities, synchronously exchange individuals, synchronous migration model [6], and asynchronously exchange individuals, asynchronous migration model [5, 4]. Syncronous migration operation is started among subpopulations simultaneously according to fixed interval called *migration interval*. If individuals called *migrants* are introduced before the search converged, it is difficult to generate superior schemata because good schemata will be destructed. Thus it is effective to introduce individuals after the search converged. However the progress of the search situation differs both the objective problems and every subpopulation, and it makes difficult to set optimal migration interval. So we expect that asynchronous migration is more effective than synchronous migration. In this paper, we propose asynchronous migration operation suitable for Multi-objective optimization problems.

2. ASYNCHRONOUS MIGRATION OF ISLAND PARALLELIZATION OF MOGA

2.1. Implementation

In this research, we unify the conditions of genetic operations exept migration operation. Individuals' genes are coded by real value coding. Crossover operation is *BLX*-α [2]. Selection operation is tournament selection. Individual's fitness value is assigned by Fonseca's Pareto Ranking [3].

2.2. Asynchronous Migration

Asynchronous migration operation is started at each subpopulation respectively according to some conditions. We propose that each subpopulation starts migration operation when the searching in itself converged. Moreover, in order to recover the diversity of the subpopulation and to promote generating good schemata, migrants are introduced from the subpopulation which has the most different individuals. We expect that when the search converged the subpopulation would have some good schemata, and the other different subpopulation would have the other good schemata. Introducing the other good schemata promote to generate higher order schemata by connecting low order good schemata. Then we face the problem. We need scales to grasp the search situation and to measure the difference among subpopulations. Our proposal is as follows.

2.2.1. How to grasp the convergence of the search

We can grasp the search situation using the fitness values' average and standard deviation at the single objective optimization [5]. However in case of the multiobjective optimization, we have to consider the correlation among objective functions. As a solution for this problem we propose to use *Generalized Variance: GV* of covariance matrix. Multivariate data of covariance matrix is constructed by individuals' objective functions. GV indicate the scale of concentration of individuals around the center of balance of individuals. The reduction of GV value indicates the concentration of individuals.

Here, when GV in a subpopulation is smaller than a threshold value K_{GV}, the subpopulation judges convergence of the search. Then the subpopulation starts migration operation and requires introducing of migrants from the other subpopulation. In our algorithm, Generalized Valiance GV is calculated as follows.

$$\begin{array}{l} E(F_j) = 1/n\sum_{i=0}^{n} f_{ij} \\ D^2(F_j) = 1/n\sum_{i=0}^{n} f_{ij}^2 - E^2(F_j) \\ D(F_jF_k) = 1/n\sum_{i=0}^{n} f_{ij}f_{ik} - E(F_j)E(F_k) \end{array} \quad (1)$$

where m is the number of objective functions and n is the size of population. $(f_{i1}, f_{i2}, \ldots, f_{im})$, $i = 1, 2, \ldots, n$ is individual i's objective functions. $E(F_j)$ is the average of objective function value $F_j, j = 1, 2, \ldots, m$. $D^2(F_j)$ is the variance. $D(F_jF_k)$ is the covariance. Generalized Variance GV is a determinant $|\Sigma|$ of a covariance matrix Σ whose elements are the above covariance.

$$\Sigma = \{\sigma_{jk}\} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \ldots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \ldots & \sigma_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ \sigma_{m1} & \sigma_{m2} & \ldots & \sigma_{mm} \end{bmatrix} \quad (2)$$

$$GV = |\Sigma| \quad (3)$$

where $\sigma_{jk} = D(F_jF_k), \sigma_{jj} = D^2(F_j)$.

2.2.2. How to grasp the difference between subpopulations

In our algorithm, subpopulation introduces migrants in order to recover the diversity. The difference between subpopulations is measured by the Euclidean distance between the centers of balance of individuals in each subpopulation. When the subpopulation starts migration operation, the subpopulation calculates the distances among itself and the other subpopulations. Then the subpopulation introduces migrants from the farthest subpopulation.

The distance d between the subpopulation A and B according to the individual's genotype $\mathbf{x}^A$ and $\mathbf{x}^B$ is calculated by the following equation.

$$d = \sqrt[m]{\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} x_{ij}^A - \frac{1}{n}\sum_{i=1}^{n} x_{ij}^B\right)^2} \quad (4)$$

where n is the population size, m is the number of variable.

3. PREPARATION FOR EXPERIMENTS

3.1. Test Functions

In order to evaluate our algorithms, we apply to four test problems which have variety features proposed by Deb and Zitzler et al.

Deb proposed a construction method of test problem [1]. The test problem is two objective optimization problem with m variables. The construction method sets the shape of the pareto-optimal front and the search space at will. The definition of the test problem is as follows.

$$\begin{array}{llll} Minimize & \mathcal{T}(x) & = & (f_1(x_1), f_2(\mathbf{x})) \\ subjectto & f_2(\mathbf{x}) & = & g(\mathbf{x})h(f_1, g) \\ where & \mathbf{x} & = & (x_1, \ldots, x_m) \end{array} \quad (5)$$

Zitzler et al. defined test functions for multi-objective optimization using the above definition [7]. We use four test functions constructed by Zitzler et al. Four test problems are defined as follows.

- The test problem $\mathcal{T}_1$ has a convex pareto-optimal front.

$$\begin{array}{lll} f_1(x_1) & = & x_1 \\ g(x) & = & 1 + 9 \cdot \sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) & = & 1 - \sqrt{f_1/g} \end{array} \quad (6)$$

  where $m = 30$, and $x_i \in [0, 1]$. The pareto-optimal front is formed with $g(\mathbf{x}) = 1$.

- The test problem $\mathcal{T}_2$ has a nonconvex pareto-optimal front.

$$\begin{array}{lll} f_1(x_1) & = & x_1 \\ g(x) & = & 1 + 9 \cdot \sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) & = & 1 - (f_1/g)^2 \end{array} \quad (7)$$

  where $m = 30$, and $x_i \in [0, 1]$. The pareto-optimal front is formed with $g(\mathbf{x}) = 1$.

- The test problem $\mathcal{T}_3$'s pareto-optimal front consists of several noncontiguous convex parts.

$$\begin{array}{lll} f_1(x_1) & = & x_1 \\ g(x) & = & 1 + 9 \cdot \sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) & = & 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{array} \quad (8)$$

  where $m = 30$, and $x_i \in [0, 1]$. The pareto-optimal front is formed with $g(\mathbf{x}) = 1$.

- The test function $\mathcal{T}_4$ has 21^9 local pareto-optimal fronts.

$$\begin{array}{lcl} f_1(x_1) & = & x_1 \\ g(x) & = & 1+10(m-1) \\ & & +\sum_{i=2}^{m}(x_i^2-10\cos(4\pi x_i)) \\ h(f_1,g) & = & 1-\sqrt{f_1/g} \end{array} \quad (9)$$

where $m=10, x_1 \in [0,1]$, and $x_2,\ldots,x_m \in [-5,5]$. The true pareto-optimal front is formed with $g(\mathbf{x})=1$, the best local pareto-optimal front is formed with $g(\mathbf{x})=1.25$.

3.2. Evaluation Method

Multiobjective optimization problem requires to find pareto-optimal solutions on the pareto-optimal front extensively. Therefore, we evaluate the algorithms using Average Error and Cover Rate.

Average Error is the average remainder between the function value of the obtained pareto-optimal solutions f and the function value of corresponding optimal solutions f^{opt} on the true pareto-optimal front.

$$Error = \frac{1}{m}\sum_{i=1}^{m} f_m - f_m^{opt} \quad (10)$$

where m is the number of the obtained pareto-optimal solutions.

Cover Rate (CR) indicates the extent of solutions at pareto-optimal front. Cover Rate measures the extent of the pareto-optimal solutions on the pareto-optimal front by dividing the pareto-optimal front into optional number of areas at each objective function and counting the percentage of areas that solution exists. In this research, we divide the pareto-optimal front into 50 areas. CR is calculated as follows.

$$CR = \frac{1}{n}\sum_{i=1}^{n} cr_i \quad (11)$$

where n is the number of objective functions.

Fig.1 shows the example of CR calculation at two objective functions. Pareto-optimal front is devided into five areas at each objective function. Cover Rate is calculated as 0.5.

$$CR = \frac{1}{2}(0.4+0.6) = 0.5$$

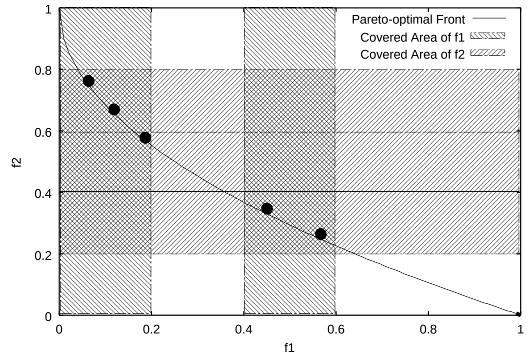

Figure 1: An Example of Cover Rate: CR=0.5

Table 1: EXPERIMENTAL CONDITION

	Sync	Async
Total Population	512	
# of Subpopulation	8	
Mutation Rate	0.01	
Migrats Selection	neighbor	farthest population
Migration Interval	10 generations	asynchronous
Migration Rate	0.25	
Gene Cording	Real Cording	
Crossover	BLX-α,$\alpha=0.5$	

4. EXPERIMENTS

4.1. Comparison among Migration Methods

4.1.1. Experimental Condition

We compare Asynchronous Island Parallel GA (*Async*) and Synchronous Island Parallel GA (*Sync*). The algorithms are implemented on a PC-Cluster system. The experimental condition of these algorithms is showed in TABLE 1 and TABLE 2. The results are evaluated by the average of 10 times trials at each condition.

4.1.2. Experimental Results

At the results of $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$, there were not significant differences between Async and Sync. Both of Async and Sync obtained the pareto-optimal fronts easily. But at the result of $\mathcal{T}_4$, there was significant difference. As space is limited, we show the results only at $\mathcal{T}_4$. Fig.2 shows the average error of obtained pareto-optimal solutions. In case of Sync, the reduction of the average error stops after the 350th generation. But in case of Async, the reduction of the

Table 2: SETTING OF K_{GV} AND FINAL GENERATION

	K_{GV}	Final Generation
$\mathcal{T}_1$	0.002	200
$\mathcal{T}_2$	0.001	200
$\mathcal{T}_3$	0.005	500
$\mathcal{T}_4$	0.05	500

average error continues while searching. This result indicates the effectiveness of asynchronou migration. Introducing migrants from different subpopulation promotes combining lower schemata into higher schemata. Fig.3 shows the cover rate in total population and Fig.4 shows the average of cover rate in each subpopulations. This result indicates the feature of island parallelization. At the 500th generation, the cover rate in total population is 0.7 to 0.8 and the cover rate in each subpopulation is 0.4 to 0.5. Each subpopulation searches for solutions locally, but covers the wide area in total. Fig.5 and 6 plot the obtained pareto-optimal solutions at 10 times trials by Async and Sync. Async obtains better pareto-optimal set than Sync.

5. DISCUSSION

We discuss the effectiveness of asynchronous migration. In the point of the necessity of preliminary setting of migration parameter, synchronous migration and asynchrnous migration are the same. However, in case of synchronous migration, since the progress of the search differs both the problem and every subpopulation, any number of trials are necessary to set the optimal migration parameter. On the other hand in case of asynchronous migration, when the problem and the size of subpopulations are the same, GV indicates the same value as the convergence of the search. In this research, we set the threshold K_{GV} by observing the GV in the subpopulation at one trial with no migration. Fig.7 shows an example of GV in two subpopulations in case of no migration. Converged generation is different between subpopulation A and B, but the indication of GV at converged generation is almost the same. The stability of the search of Async is better than that of Sync.

6. CONCLUSION

In this paper, we propose an asynchronous migration method of the island parallel GA for multi-objective optimization problem. Our asynchrnonous migration is started when the diversity of subpopulation is lost. The loss of diversity is observed by generalized valiance of covariance matrix of which multi-variate data is constructed by individuals' objective functions. Migrants are introduced from the other subpopulation which has different genetic construction. The differences of genetic construction among subpopulations are detected by the Euclidean distance between the centers of balance of individuals in each subpopulation. Asynchronous migration model was compared with synchronous migration model using Zitzler's test functions. At the problem which has 21^9 local pareto-optimal fronts, asynchronous migration model obtains better pareto-optimal set than synchronous migration model. This result suggests the effec-

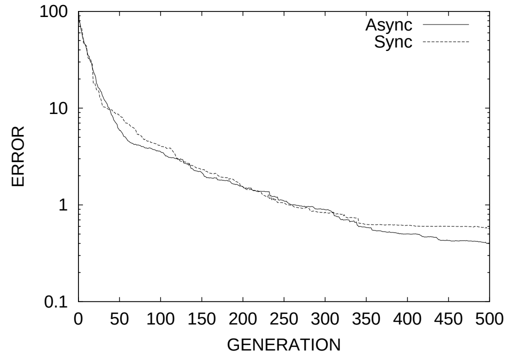

Figure 2: $\mathcal{T}_4$'s Error

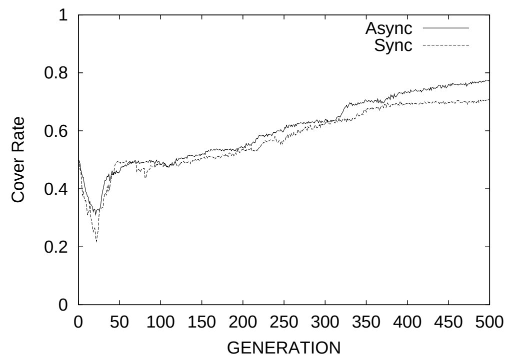

Figure 3: $\mathcal{T}_4$'s Cover Rate in Total Population

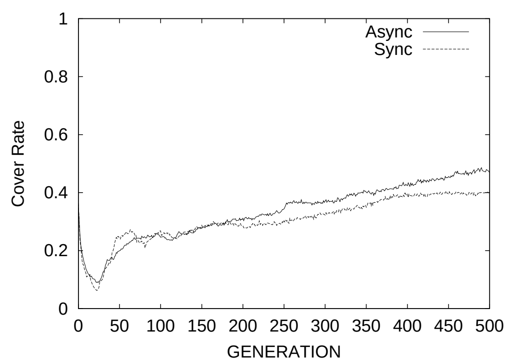

Figure 4: $\mathcal{T}_4$'s Average Cover Rate in Each Subpopulation

tiveness of our algorithms for multi-modal problems at multi-objective optimization.

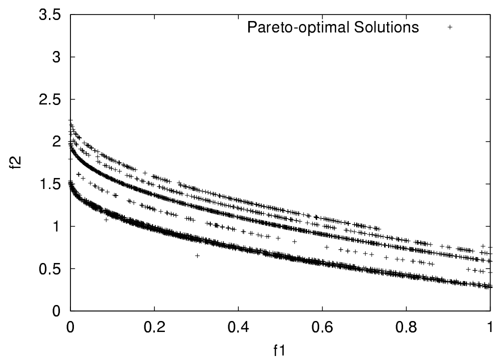

Figure 5: $\mathcal{T}_4$'s Pareto-optimal Solutions by Async

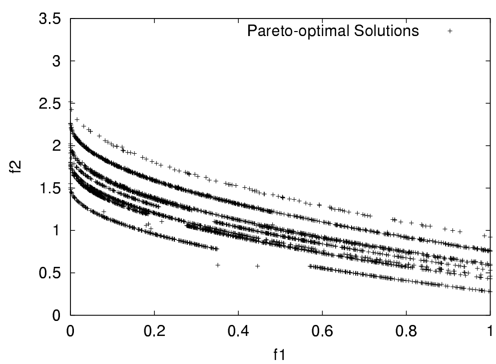

Figure 6: $\mathcal{T}_4$'s Pareto-optimal Solutions by Sync

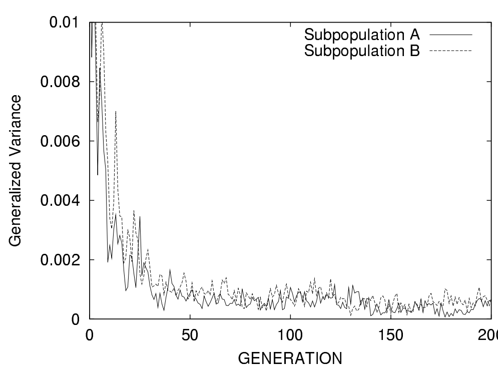

Figure 7: Examples of Generalized Variance in Two Subpopulations

7. REFERENCES

[1] Deb, K., ”Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems,” *Evolutionary Computation*, Vol.7, No.3, pp.205–230 (1999).

[2] Eshelman, L.J. and Schaffer, J.D., ”Real-Coded Genetic Algorithms and Interval-Schemata,” *Foundations of Genetic Algorithms 2*, pp.187–202 (1993).

[3] Fonseca, C.M. and Fleming, P.J., ”Genetic Algorithms for Multi-objective optimization: Formulation, Discussion and Generalization,” *Proc. of the 5th ICGA*, pp.416–423 (1993).

[4] Horii, H., Kunifuji, S., and Matsuzawa, T., ”Asynchronous Island Parallel GA Using Multiform Subpopulations,” Second Asia-Pacific Conference on Simulated Evolution and Learning, LNAI 1585, pp.122–129 (1999).

[5] Munetomo, M., Takai, Y. and Sato, Y., ”An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms,” in *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp.649 (1993).

[6] Tanese, R., ”Distributed Genetic Algorithm,” in *Proceedings of the Third International Conference on Genetic Algorithms*, pp.434–439 (1989).

[7] Zitzler, E., Thiele, L. and Deb, K., ”Comparison of Multiobjective Evolutionary Algorithms: Empirical Results,” in *Evolutionary Computation*, Vol.8, No.2, pp.173–195 (2000).