MOGADES: Multi-Objective Genetic Algorithm with Distributed Environment Scheme

Jiro KAMIURA[††] Tomoyuki HIROYASU[†], Mitsunori MIKI[†], Shinya WATANABE[††]

[†] Department of Knowledge Engineering,
[††] Graduate Student,

Doshisha University
Kyo-tanabe, Kyoto, 610-0321, JAPAN
Email: tomo@is.doshisha.ac.jp

1 Introduction

Recently, computers have made rapid progress in hardware and software. Because of this background, when structures such as cars and air plains, electric devices such as circuits and controllers and so on are designed, designers begin to use computer simulations for their decision-makings. In this case, optimization techniques are often utilized. However, especially in real world problems, there is not only one objective but also several objectives. Therefore, multi-objective optimization problems should be solved. One of the goals of multi-objective optimization problems may be to obtain a set of Pareto optimal solution. Since Pareto optimal set is an assemble of the solutions, genetic algorithm, which is one of multi point search methods is suitable to derive the Pareto optimal set [1]. In this few years, several new algorithms that can find good Pareto optimal solutions with small calculation cost are developed [2]. Those are NSGA-II [3], SPEA-II [4], NPGA-II [5] and MOGA [6].

One of the disadvantages of these methods is a high calculation cost [7]. Performing GA on parallel computer is one of the solutions of this problem. Evolutionary algorithms (EAs) are algorithms that have implicit parallelism [8]. Therefore, algorithms of parallel EAs are very important. However, there are few studies that are related to parallel algorithms of GAs for multi objective optimization problems (MOPs).

In this paper, we propose a parallel genetic algorithm for multi objective optimization problems. That is called " Multi-Objective Genetic Algorithm with Distributed Environment Scheme (MOGADES)". This is an expanded algorithm of distributed GA (DGA) and it also use the concept of environment DGA [9].

To clarify the characteristics and the effectiveness of MOGADES, we apply MOGADES to some test functions. Through the comparison of MOGADES to SPEA2 [4] and NSGA-II [3], the advantages and disadvantages of MOGADES are made clarified.

2 Genetic Algorithms for Multi-Objective Optimization Problems

2.1 Multi-Objective Optimization Problems

In optimization problems, when there are several objective functions, the problems are called Multi-objective Optimization Problems (MOPs) [1]. Multi-objective optimization problems are formulated as follows,

$$\begin{cases} minimize \quad \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x}))^T \\ subject \quad to \quad the \quad constraints \quad g_j(\vec{x}) \leq 0, (j = 1, \dots, m) \end{cases} \tag{1}$$

Usually these objectives cannot minimize or maximize at the same time, since there is a trade off relationship between the objectives. Therefore, one of the goals of multi-objective optimization problem is to find a set of Pareto optimal solutions.

Several algorithms were proposed and they were applied to multi-objective optimization problems [1]. Those algorithms are roughly divided into two categories; those are the algorithms that treat Pareto optimal implicitly and explicitly.

VEGA [10] is a traditional GA for multi-objective problems and this is an algorithm that treats Pareto optimal implicitly. MOGLS [11] is also an algorithm that treats Pareto optimal implicitly. On the other hand, most of multi-objective GAs treats Pareto optimal explicitly. Among those algorithms, SPEA2 [4] and NSGA-II [3] have the powerful search mechanisms and derived the good results. They have some similar concepts; those are archive of non-dominated solution, sharing without parameters, and assignment method of fitness function.

2.2 Parallelization of Multi Objective Genetic Algorithms

Genetic algorithm (GA) is an optimization algorithm that mimics the process of evolution [6, 12]. Since GA is one of multi point search methods, there are several types of parallel methods. Parallel genetic algorithms are roughly classified into three categories; those are a master-slave population model, an island model, and a cellular model [13].

In an island model that is also called Distributed GA (DGA), a population is divided into sub populations. In each island, a conventional GA is performed for several iterations. After that, some individuals are chosen and moved to the other islands. This operation is called migration. After migration, GA operations are started again in each island. Since the network traffic is not huge and each island has small number of individuals, an island model can gain high parallel efficiency [13]. In a single-objective problem, it is reported that DGA can find a good solution with the small calculation cost [14]. However, since the number of individuals is small in an island, DGA cannot find good Pareto optimal solutions in multi-objective problems. Therefore, to find good solutions in an island model, some mechanisms to find solutions should be included. In the former study, we developed an island model that is called a Divided Range Multi Objective Genetic Algorithm (DRMOGA) [7]. This is an algorithm that treats Pareto optimal explicitly. However, the searching ability of DRMOGA is not good compared to SPEA2 [4] and NSGA-II [3].

3 Environment Distributed Genetic Algorithms

Usually each island of DGAs is assigned to one processor of parallel computers [8]. Since the network cost is not high in DGAs, the high parallel efficiency can be derived. It is also reported that DGAs can find optimum solution with smaller calculation cost than that of simple GA. Therefore, DGAs have a lot of advantages.

Generally, every island has the same environment that is population size, crossover rate, mutation rate and so on. However, the environment can be different in each island. For example, when the crossover rate and mutation rate are different in each island, the searching ability of DGA is increased. We called this DGA as Environment Distributed Genetic Algorithm (EDGA) [9]. This scheme can be applied to the other problems. In the following section, the proposed algorithm, MOGADES, is explained. MOGADES is an algorithm where the EDGA is extended for multi-objective optimization problems.

4 Multi-Objective Genetic Algorithm with Distributed Environment Scheme

In the former section, EDGA is explained. In this section, EDGA is extended for Multi-Objective optimization problems. The proposed algorithm is called ”Multi-Objective Genetic Algorithm with Distributed Environment Scheme (MOGADES)”.

A multi-objective optimization problem can be changed into a single objective optimization using weight parameters w_i as follows,

$$\min_{x \in X} \quad f(x) = \sum_{i=1}^{k} w_i f_i(x) \tag{2}$$

$$where \quad w_i \geq 0, \quad \sum_{i=1}^{k} w_k = 1 \tag{3}$$

To derive the Pareto optimal solutions, a lot of simulations with different weight parameters should be needed. Because MOGADES is one of EDGAs, there are several islands and each island can have a different weight parameter. This is the basic concept of MOGADES. At the same time, the search mechanisms of SPEA2 and NSGA-II are included into MOGADES. The overall process of MOGADES is summarized as follows. In this case, the problem that has two objects is explained. Each island has own weight value, an elite archive and a Pareto archive. The weight value is used when the fitness value is derived. During the search, the solutions that have the best fitness values are preserved in an elite archive. In the same way, the solutions that are non-dominated to the other solutions are stored in a Pareto archive.

Step 1: Initialization: Generate new individuals. Those individuals are divided into islands P_i^0 $(i = 1, 2, \ldots M)$. Set the weight value w_i of ith island. At first, the weight values are arranged equally from 0.0 to 1.0. For example, when $M = 5$, the weights are 0, 0.25, 0.5, 0.75 and 1.0. In this time, the elite archive EA_i^0 and Pareto archive PA_i^0 are empty. Set generation $t = 0$. Calculate the values of function 1, 2 and the fitness value of each individual.

Step 2: Starting new generation: Set $t = t + 1$.
The Steps from 3 to 9 are performed in an island independently.

Step 3: Crossover and mutation: Perform crossover and mutation operations.

Step 4: Evaluation: Calculate the values of function 1 and 2. Normalize the values of functions by the maximum value of each function. Calculate the fitness value of each individual. The fitness value is derived from equation (3).

Step 5: Selection: Perform selection operation to P_i^t.

Step 6: Terminal Check: When the terminal condition is satisfied, terminate the simulation. Otherwise, the simulation is continued.

Step 7: Pareto Reservation: Choose the individuals of P_i^t and PA_i^{t-1} that are non-dominated and copy them into PA_i^t. When the number of PA_i^t overcomes the maximum number of the Pareto archive, the sharing operation is performed. The sharing method of MOGADES is carried out as follows. The individuals of P_i^t are sorted with along to the fitness value. Calculate the distance of the fitness value between the neighborhood individuals. Truncate the individual who has the smallest distance.

Step 8: Elite Reservation: According to the fitness values, reserve the individuals who have good fitness values into EA_i^t.

Step 9: Renewal the search individuals: $P_i^t = P_i^{t-1} + PA_i^t + EA_i^t$.

Step 10: Migration: Choose some individuals and move to the other island. In MOGADES, migration topology is fixed. When migration is performed, the weight value of the island is changed in the following equation,

$$\begin{aligned} w_i(new) &= w_{i+1}\frac{d_{(i+1,i)}}{d_{(i,i-1)} + d_{(i+1,t)}} \\ &\quad + w_{i-1}\frac{d_{(i,i-1)}}{d_{(i,i-1)} + d_{(i+1,i)}} \end{aligned} \tag{4}$$

In this equation, w_i is the weight value of ith island, $d_{(i+1,i)}$ the distance between the individuals who has the best value in $i+1$th island and ith island.

Step 11: Returne to Step 2.

5 Numerical Examples

In this section, to discuss the effectiveness of MOGADES, MOGADES is applied to find Pareto optimal solutions of test functions. The results are compared with those of SPEA-2 [4] and NSGA-II [3].

5.1 Test Functions

In this paper, we use a continuous function and a knapsack problem. These problems are explained as follows. In these equations, f denotes an objective function and $g (g \geq 0)$ indicates a constraint.

$$KUR: \begin{cases} \min \quad f_1 = \sum_{i=1}^{n}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})) \\ \min \quad f_2 = \sum_{i=1}^{n}(|x_i|^{0.8} + 5\sin(x_i)^3) \\ \qquad x_i[-5,5],\ n = 100 \end{cases} \tag{5}$$

$$KP750-2: \begin{cases} \min \quad f_i(x) = \sum_{j=1}^{n} x_j \cdot p_{i,j} \\ s.t. \\ \qquad g_i(x) = \sum_{j=1}^{n} x_j \cdot w_{i,j} \leq W_i \\ \qquad p_{i,j}(\text{profit value}) \\ \qquad w_{i,j}(\text{weight value}) \\ \qquad 1 \leq i \leq 2,\ n = 750 \end{cases} \tag{6}$$

KUR was used by Kursawa [15]. It has a multi-modal function in one component and pair-wise interactions among the variables in the other component. Since there are 100 design variables, it needs a high calculation cost to derive the solutions. KP750-2 is a 0/1 knapsack problem and it is a combinatorial problem [4, 16]. There are 750 items and two objects. The profit and weight values are the same as those of the Reference [17].

5.2 Parameters of GAs

In this paper, to discuss the effectiveness of the algorithm, the bit coding is applied for all the problems. It is known that the good results are derived when the real value coding is applied. Similarly, one point crossover and bit flip are used for crossover and mutation. The length of the chromosome is 20 bit per one design variable for the continuous problems and 750 bit for the knapsack problems. In the continuous problems, population size is 100 and the simulation is terminated when the generation is over 250. In the knapsack problems, population size is 250 and the simulation is terminated when the generation exceeds 2000.

5.3 Evaluation methods

To compare the results derived by each algorithm, the following evaluation methods are used in this paper.

5.3.1 Ratio of Non-dominated Individuals (RNI)

This performance measure is obtained by comparing two solutions which are derived by two methods. RNI is derived from the following steps. At first, two populations from different methods are mixed. Secondly, the solutions that are non-dominated are chosen. Finally, RNI of each method is determined as the ratio of the number of the solutions who are in chosen solutions and derived by the method and the total number of the solutions. By RNI, the accuracy of the solutions can be compared.

Figure 1 shows an example of RNI. In this figure, method A and B are compared. This case suggests that A and B are almost the same.

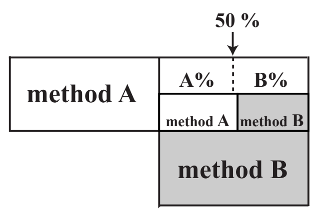


Figure 1: Example of RNI

5.3.2 Maximum, Minimum and Average values of each object of derived solutions (MMA)

To evaluate the derived solutions, not only the accuracy but also the expanse of the solutions is important. To discuss the expanse of the solutions, the maximum, minimum and average values of each object are considered.

5.3.3 KUR

In this problem, there are 100 design variables. Therefore, a lot of generations should be needed to derive the solutions. The results of RNI and MMA are shown in figure 2.

In this case, RNI of MOGADES is superior to the other methods. Since MOGADES has islands who are searching the edges of Pareto optimal solutions, it can derive the widespread solutions especially in difficult problems.

5.3.4 KP-2

KP-2 is the knapsack problem and it is very difficult to search the real Pareto optimal solutions. The results of RNI and MMA are shown in figure 3.

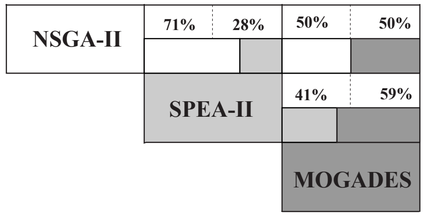
In this case, MOGADES obtained very good results. Because this problem is very difficult, the factor of MOGADES that can derive the widespread solutions acts effectively.

6 Conclusion

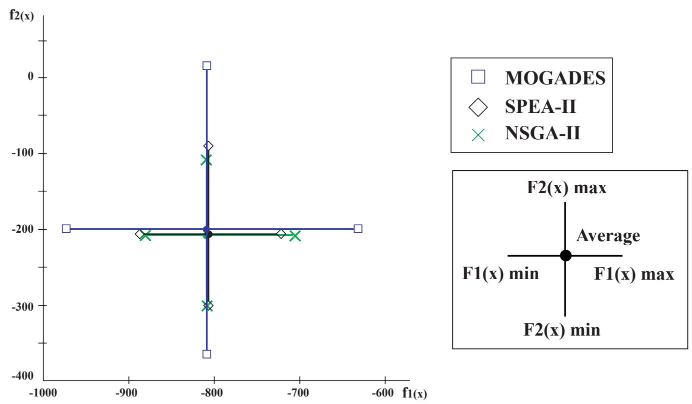
In this paper, a new distributed genetic algorithm for multi-objective optimization problems is proposed. That is called Multi-Objective Genetic Algorithm with Distributed Environment Scheme (MOGADES). Using the weight parameter, a multi-objective problem is turned into a single-objective problem. At the same time, each island has the different weight value and archive that preserves the individuals who are non-dominated to the others. Because of these mechanisms, MOGADES can derive the widespread solutions. To discuss the effectiveness of the proposed method, MOGADES was applied to test functions and the results were compared to the other methods; those are SPEA-2 and NSGA-II. Through the numerical examples, it became clear the following two points.

1) MOGADES is superior to the other methods in the problems that are used in this paper.

2) MOGADES can derive the widespread Pareto optimal solutions.

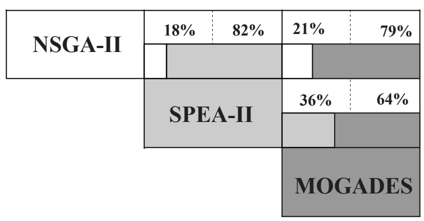


(a) RNI of KUR

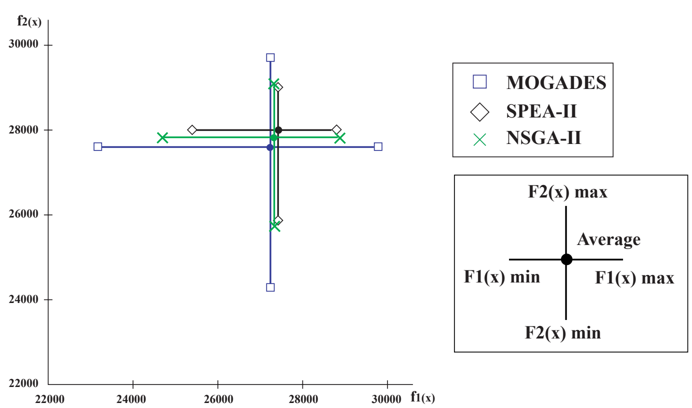


(b) MMA of KUR

Figure 2: Results of KUR



(a) RNI of KP750



(b) MMA of KP750

Figure 3: Results of KP750

Acknowledgement

This research is funded by the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] K.Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK : Wiley, 2001.

[2] C. A. Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *In 2000 Congress on Evolutionary Computation*, Vol. 1, pp. 30–37, 2000.

[3] K.Deb, S.Agrawal, A.Pratab, and T.Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *KanGAL report 200001, Indian Institute of Technology, Kanpur, India*, 2000.

[4] E.Zitzler, M.Laumanns, and L.Thiele. Spea2: Improving the performance of the strength pareto evolutionary algorithm. In *Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich*, 2001.

[5] M.Erickson, A.Mayer, and J.Horn. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In *In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization,Springer-Verlag. Lecture Notes in Computer Science No. 1993*, pp. 681–695, 2000.

[6] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international coference on genetic algorithms*, pp. 416–423, 1993.

[7] T. Hiroyasu, M. Miki, and S. Watanabe. The new model of parallel genetic algorithm in multi-objective optimization problems -divided range multi-objective genetic algorithms. In *IEEE Proceedings of the 2000 Congress on Evolutionary Computation*, pp. 333–340, 2000.

[8] T. Hiroyasu, M. Miki, and Y. Tanimura. Characteristics of models of parallel genetic algorithms on pc cluster systems. In *Proceedings of 1st Internation Workshop on Cluster Computing (IWCC '99)*, 1999.

[9] M.Kaneko, T.Hiroyasu, and M.Miki. A Parallel Genetic Algorithm with Distributed Environment Scheme. In *Proceedings of the International Conference on Parallel and Distributed Processing Techiniques and Applications*, Vol. 2, pp. 619–625, 2000.

[10] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of 1st International Conference on Genetic Algorithms and Their Applications*, pp. 93–100, 1985.

[11] T. Murata and H. Ishibuchi. Moga: Multi-objective genetic algorithms. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computing*, pp. 289–294, 1995.

[12] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesly, 1989.

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

[14] R.Tanese. Distributed Genetic Algorithms. In *Proc.3rd International Conf.Genetic Algorithms*.

[15] F.Lirsawe. A variant of evolution strategies for vector optimization. In *PPSN I, volume 496 of Lecture Notes in Computer Science*, pp. 193–197, 1991.

[16] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.

[17] Test problems for multiobjective optimizers (eckart zitzler's page). http://www.tik.ee.ethz.ch/ zitzler/testdata.html.