

The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems

– Divided Range Multi-Objective Genetic Algorithm –

Tomoyuki HIROYASU

Mitsunori MIKI

Sinya WATANABE

Doshisha University, Dept. of Knowledge Engineering and Computer Sciences
1-3 Tatara Miyakodani, Kyotanabe
Kyoto 610-0321, Japan

tomo@is.doshisha.ac.jp

mmiki@mail.doshisha.ac.jp

sin@mikilab.doshisha.ac.jp

Abstract- In this paper, Divided Range Multi-Objective Genetic Algorithm (DRMOGA) is proposed. The DRMOGA is a model of genetic algorithm in multi-objective problems for parallel processing. In the DRMOGA, the population of GAs is sorted with respect to the values of the objective function and divided into sub populations. In each sub population, simple GA for multi-objective problems is performed. After some generations, all individuals are gathered and they are sorted again. In this model, the Pareto optimum solutions which are close to each other are collected by one sub population. Therefore, this algorithm increases the calculation efficiency, and the neighborhood search can be performed. Through the numerical examples, the followings are become cleared. The DRMOGA is very suitable GA model for parallel processing. In some cases, the DRMOGA can derive the better solutions compared to both the single population model and the distributed model.

1 Introduction

When the real world problems are solved, the multi-objective optimization techniques are often used. In multi optimization problems, there are several types of the objectives, which usually have the trade off relation. One of the strongest algorithms that solve the multi-objective problem is Genetic Algorithms (GAs).

There are several studies that concerned with GA applied to the multi-objective function (Fonseca and Fleming, 1994; Tamaki, et al., 1995; Coello, 1999). Since GA is one of the multi searching methods, it is suitable for finding the Pareto optimum solutions. There are several models are proposed for the multi-objective GA. Schaffer developed the VEGA (Schaffer 1985). Goldberg et al. introduced the ranking method (Goldberg et al. 1989), and Fonseca et al. also developed the MOGA (Fonseca and Fleming, 1993). In their methods, the Pareto optimum solutions are treated explicitly. Tamaki et al.

(Tamaki, et al. 1995) introduced their model where the VEGA used, and the Pareto optimum individuals are remained.¹ Additionally, there is a method of Murata et al. (Murata, et al. 1995). In their method, by weighting the values to each objective function, they convert the objective optimization problems to single objective optimization problems.

Like this way, there are several models of multi-objective GAs and they can derive the good Pareto optimum solutions. However, they need a lot of iterations to calculate the values of objective functions and the constraints. This leads to the high calculation costs. One of the solutions of this problem is to perform the multi-objective GA in parallel processing.

There are several studies that concerned with the parallelization methods of GA for a single object (Nang and Matsuo 1994; Cantu-Paz 1999, Sawai and Adachi 1999). For multi-objective optimization problems, there are few studies of GA, and the models of these studies are almost the same as for a single object. For example, there is a model where the parts of evaluation of fitness are performed in parallel (Jones and Crossley 1998). There is another model where the total population is divided into sub populations, and the multi-objective optimization is performed in each sub population (Vicini 1998). However, the mechanism of searching the optimum is different between the single objective GA and the multi-objective GA. In the single objective GA, only one optimum should be derived. Therefore, the diversity of the searching point is important in the first stage and the local search is important in the latter stage. On the other hand, in the multi-objective GA, both the diversity and the local search are important for all stages, because it should derive not only one point but also the meeting of the points. This fact suggests the following; in the multi-objective optimization in parallel, the different model should be used from the one in the single objective GA.

¹Tamaki et al. called the individuals that are in the Pareto front as the Pareto optimum individuals.

Because of the necessity of a new model of GA in multi-objective problems for parallel processing, we introduced a model of Divided Range Genetic Algorithms in Multi-Objective Optimization Problems: DRMOGA (Hiroyasu, et al. 1999). In the DRMOGA, the individuals are divided into sub populations by the values of their objective function. Therefore, the efficient search can be performed and the adequate local search also carried out. In this study, DRMOGA is explained briefly and applied to the numerical test problems that are difficult to derive the solutions. Through the numerical examples, the validity of the model and the characteristics of the solutions are discussed. Especially, the test functions that Deb developed (Deb 1999) used and the high searching performance of DRMOGA is got cleared.

2 Divided Range Multi-Objective Genetic Algorithm

2.1 Overview of DRMOGA

In this study, a new model of a multi-objective genetic algorithm is explained. That is Divided Range Multi-Objective Genetic Algorithm: DRMOGA. This model is suitable for parallel processing.

The flow of Distributed Range Multi-Objective Genetic Algorithm is explained as follows.

- **Step 1** Initial population (population size is N) is produced randomly. All the design variables that are shown with the individuals satisfy the constraints.
- **Step 2** The individuals are sorted by the values of focused objective function f_i . This focused objective function f_i is chosen in turn, and turned with the loop. Then, the individuals of number N/m are chosen in accordance with the value of this focused objective function f_i . As the result, there exist m sub populations.
- **Step 3** In each sub population, the multi-objective GA has been performed for some iterations. The multi-objective GA that is used in this paper is explained in the next section. The end of each generation, the terminal condition is examined and the process is terminated when the condition is satisfied. When the terminal condition is not satisfied the process progress into the next step.
- **Step 4** After the multi-objective optimization has been performed for k generations, all of the individuals are gathered (virtually). Then the process is going back to Step 2. This generation k is called the sort interval.

In this study, the number of distribution m and the sort interval k is determined in advance. In Figure 1, the

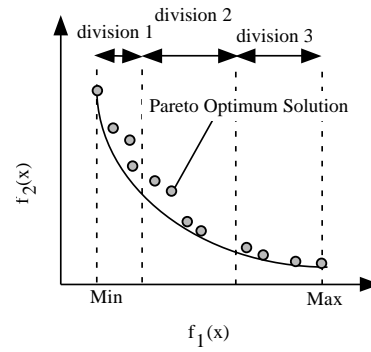


Figure 1: DRMOGA

concept of the DRMOGA is shown. In Figure 1, there are two objective functions. Individuals are divided into three by the value of the focused objective function f_1 .

The area with respect to the focused objective function determines the sub population of the DRMOGA. This mechanism is supposed to functions as the sharing. Therefore, the derived Pareto optimum solutions of the DRMOGA might have the high diversity.

2.2 Configuration of Genetic Algorithm

2.2.1 Expression of individuals

In genetic algorithms, usually, the individuals are shown in bit sequences. In this paper, the example problems are the real value problems. Therefore, we show the individuals as the vectors of the real values. For example, the individuals are shown like

$$a_1 = \{0.02, 10.03, \dots, 7.52\}. \quad (1)$$

Each element expresses the value of the design variable.

2.2.2 Crossover

Because the design variables are directly shown in real values, Center Neighborhood Crossover (CNX) is used in this paper.

In the CNX, $N + 1$ parent individuals are selected randomly. N expresses the number of design variables. The vector of the gravity of the selected individuals \vec{r}_g is derived with the following equation.

$$\vec{r}_g = \frac{1}{n+1} \sum_{i=1}^{n+1} \vec{r}_i \quad (2)$$

In this equation, \vec{r}_i is the vector of the parent individual. The concept of CNX is shown in Figure2 where there are two design variables.

New individual called child individual \vec{r}_{child} is generated with the following equation.

$$\vec{r}_{child} = \vec{r}_g + \sum_{i=1}^{n+1} t_i \vec{e}_i \quad (3)$$

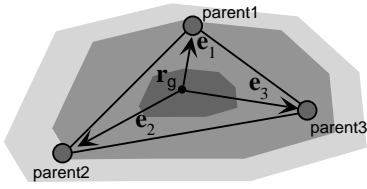


Figure 2: CNX Crossover

In this equation, \vec{e}_i is a vector from the gravity to the parent individual and t_i is a normal distribution random number whose normal distribution is σ_i and the average is 0. It is also supposed that σ_i is derived with the following equation.

$$\sigma_i = \alpha |\vec{r}_i - \vec{r}_g|, \quad (i = 1, \dots, n) \quad (4)$$

In this equation, α is the control parameter of the normal distribution random number. The probability density of the normal distributions when $\alpha = 3$ is shown in Figure 3. This is the case where there are two design variables. In these figures, the edges of the triangles are the parent individuals.

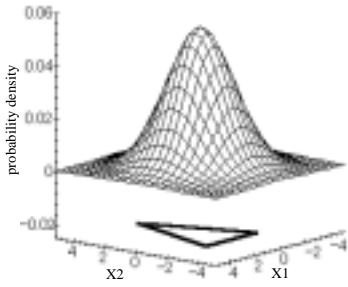


Figure 3: Normal Distribution ($\alpha = 3$)

It is obvious that the parameter α is small, the child individual who has the characteristics of the parents is generated. When the parameter α is big, the child individual is different from the parents. Therefore, when α is small, the child individual is close to the parents and when α is big, child individual is generated randomly. The optimum parameter of α is different from the type of the problem. In the following numerical examples, the two types of α are prepared and these parameters are examined their affection to the solutions.

2.2.3 Selection

In the following numerical examples, we used the three types of the selection methods. Those are

- Pareto elite reservation + sharing
- Roulette selection

- Roulette selection + sharing .

In the first strategy, the selection is an elite selection and all of the individuals whose ranking is 1 are selected. When the population size is over the certain number, the individuals are chosen by the roulette selection with the fitness values. The fitness values are determined by the sharing.

In the second strategy, the values of the fitness function for the roulette is only determined by the values of the rankings.

In the third strategy, the values of the fitness function are constructed with the values of the rankings and the sharing.

2.2.4 Terminal condition

Many researchers used the number of generation as the terminal conditions. However, this condition is not practical, because the optimum generation can be determined after the solutions are derived.

We use the movement of the Pareto frontier as the terminal condition. When the movement of the pareto frontier is small, the simulation is terminated.

3 Numerical Examples

In the numerical examples, the proposed DRMOGA was adapted to four test functions on PC cluster systems. By adapting the DRMOGA to these test functions, the validity of the DRMOGA and the characteristics of the solutions are discussed.

3.1 Cluster System and Used Parameters

3.1.1 PC Cluster system

In the numerical examples, the DRMOGA is applied to the test functions on the PC cluster system. The spec of the used PC cluster is summarized in Table 1.

Table 1: Cluster system	
CPU	Pentium II (500MHz)*5
Memory	128 Mb
OS	Linux2.2.10
Network	FastEthernet TCP/IP
Communication library	MPICH1.1.2

3.1.2 Used parameters

GAs in multi-objective optimization problems need many parameters. In this paper, the parameters shown in Table 2 are used for the numerical examples.

Because of the characteristics of the test functions, the optimal parameter is different. Among the parameters, the selection method and the parameter α that is

Table 2: Used parameters

	<i>SGA</i>	<i>DGA</i>	<i>DRMOGA</i>
<i>Population size</i>	500		
<i>Crossover rate</i>	1.0		
<i>Mutation rate</i>	0.0		
<i>Number of islands</i>	—	5	
<i>Migration interval</i> (Sort interval)	—	5	
<i>Migration rate</i>	—	0.1	—

used in the crossover affects the solutions. Therefore, in each example, the following 6 types of the cases are applied.

Table 3: Cases

Case	selection method	α
Case 1	Pareto optimal	3
Case 2	Pareto optimal	6
Case 3	Only roulette selection	3
Case 4	Only roulette selection	6
Case 5	Roulette selection with sharing	3
Case 6	Roulette selection with sharing	6

All of the results are the average of 30 trials. The results of DRMOGA is compared those of the simple island model and one population model. In this paper, the results of DRMOGA, simple island and simple GA are shortened to "DR", "Island" and "Simple" respectively.

3.2 Matrix

One of the most difficult problems in multi-objective optimization problems is to evaluate the Pareto optimum individuals. Because the Pareto optimum individuals are the assembly of the points, there is no good quantitative way of evaluating. Many researchers only show the derived Pareto optimum individuals in figures. This evaluation is not quantitative way and can only apply for two and tree objective functions.

Hiyane (Hiyane 1997) introduced his matrix for the accuracy and the quality of the Pareto optimum individuals. In this study, the matrix that is simplified the Hiyane's methods are utilized as follows.

3.2.1 Error

When the real Pareto optimum solutions are given, the average of Euclid distances between the real Pareto solutions and each Pareto optimum individuals. When the error is small, the Pareto optimum individuals are very close to the real Pareto solutions. This matrix only can apply to the problem where the Pareto solutions are given. We used the shorthand expression of errors. In the test functions, the Pareto solutions exist on the constraints. Therefore, when $g(x) = 0$ is the real Pareto

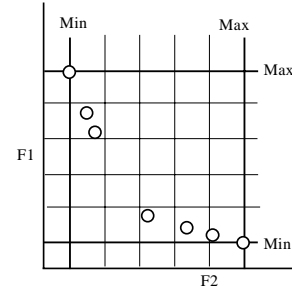


Figure 4: Cover rate

solutions, the following shorthand errors are used.

$$Error = \sqrt{\sum_{i=1}^N g(x_i)^2 / N} \quad (5)$$

In this expression, N expresses the number of the Pareto optimum individuals.

3.2.2 Cover rate

Cover rate is the index for the diversity of the Pareto optimum individuals. The cover rate is derived in the following steps. At first, one of the object functions is focused. Secondly, the distance between the individuals that have the maximum and the minimum values is divided into the certain number of the division. Thirdly, the division area that have the Pareto optimum individuals is counted. Fourthly, the counted number is divided by the number of division. When every divided area has at least one Pareto optimum individual, this number becomes 1. When there are no area that has the Pareto optimum individuals, this number becomes 0. Fifthly, these steps are treated for every objective function. Finally, the cover rate is determined to average the number of each objective function. When the cover rate is close to 1, it means that the Pareto optimum individuals are not concentrated on one point but they spreads. In Figure 4, the concept of the cover rate are shown, when there are two objective functions.

3.3 Example 1

Example 1 is a convex problem that Tamaki et al. used (Tamaki et al. 1995). Among the four examples, this is the easiest problem to find the Pareto solutions. The equations of this problem is shown as follows,

$$f_1(x) = 2\sqrt{x_1} \quad (6a)$$

$$f_2(x) = x_1(1 - x_2) + 5 \quad (6b)$$

$$g_1(x) = x_1 - 1 \geq 0 \quad (6c)$$

$$g_2(x) = 4 - x_1 \geq 0 \quad (6d)$$

$$g_3(x) = x_2 - 1 \geq 0 \quad (6e)$$

$$g_4(x) = 2 - x_2 \geq 0. \quad (6f)$$

The results of this example are summarized in Table 4.

Table 4: Results of Problem 1

Case	number of solutions	error	cover rate	generations
Simple Case1	436	0.00	1.00	799
Case2	382	0.03	1.00	1000
Case3	471	0.00	1.00	35
Case4	444	0.00	1.00	367
Case5	461	0.00	1.00	39
Case6	330	0.00	1.00	1000
Island Case1	436	0.01	1.00	43
Case2	438	0.01	1.00	59
Case3	423	0.01	1.00	273
Case4	435	0.01	1.00	44
Case5	431	0.01	1.00	66
Case6	404	0.01	1.00	927
DR Case1	500	0.00	1.00	40
Case2	500	0.00	1.00	48
Case3	494	0.00	1.00	105
Case4	494	0.00	1.00	548
Case5	495	0.00	1.00	199
Case6	494	0.00	1.00	814

The DRMOGA can derive the Pareto solutions that have high accuracy in all the cases. In the DRMOGA, the Pareto optimal selection is useful (case 1 and 2). This is come from the fact that the problem is easy and the Pareto individuals also can be firmed easily.

When the selection type is Pareto optimum (case 1 and 2), the number of generation of DRMOGA is smaller than that of island GA or simple GA. This shows the fact that DRMOGA can search the Pareto solutions efficiently. In the other examples, when DRMOGA is useful to find the solutions, DRMOGA also can find the Pareto solutions efficiently.

3.4 Example 2

In example 2, there are three objects and it is difficult to get the Pareto solution all over the objective domain (Veldhuizen and Lamont 1999). The equations of this problem are shown as follows,

$$f_1(x) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \quad (7a)$$

$$f_2(x) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \quad (7b)$$

$$f_3(x) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2) \quad (7c)$$

$$g_1(x) = x_1 \geq -3 \quad (7d)$$

$$g_2(x) = x_2 \leq 3. \quad (7e)$$

The results are summarized in Table 5.

Table 5: Results of Problem 2

Case	number of solutions	cover rate	generations
Simple Case1	500	0.75	15
Case2	500	0.74	18
Case3	491	0.51	19
Case4	485	0.50	30
Case5	316	0.48	19
Case6	207	0.47	198
Island Case1	428	0.79	19
Case2	426	0.79	36
Case3	434	0.76	22
Case4	403	0.77	55
Case5	6	0.04	1000
Case6	125	0.43	943
DR Case1	386	0.95	44
Case2	330	0.96	256
Case3	429	0.92	82
Case4	255	0.85	277
Case5	337	0.88	66
Case6	90	0.53	117

The derived solutions are shown in Figure 5 to 7. These are the typical results of case 1.

It is obvious that DRMOGA can derive the good Pareto solutions compared to the island model and the simple GA model. Usually, the results of one population model (simple GA) are better than those of the island model. The DRMOGA is one of the island model. However, the results of DRMOGA are better than those of the simple GA. In this problem, the solutions are scattered $f_2(x) = [15, 17.5]$. On the other hand, they are easily concentrated on the plane $f_2(x) = 15$. This fact makes the problem is difficult. DRMOGA is sorted by the values of the objective functions. This operation finds the solutions all over the $f_2(x) = [15, 17.5]$.

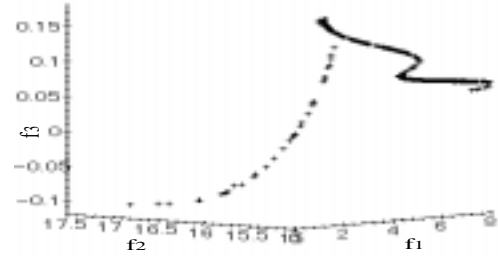


Figure 5: Pareto solutions of Example 2 (Simple)

3.5 Example 3

Example 3 is a problem of two design variables that Deb developed (Deb 1999). The equations are shown as follows,

$$f_1 = 1 - \exp(-4x_1) \sin^6(5\pi x_1) \quad (8a)$$

$$f_2 = gh \quad (8b)$$

$$g(x_2, \dots, x_N) = 1 + 10 \left(\frac{\sum_{i=2}^N x_i}{N-1} \right)^{0.25} \quad (8c)$$

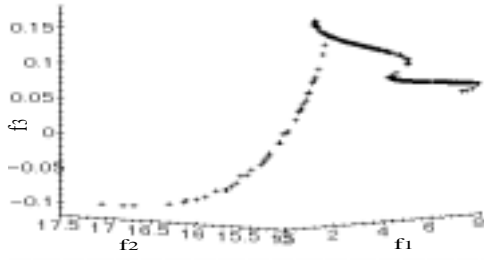


Figure 6: Pareto solutions of Example 2 (Island)

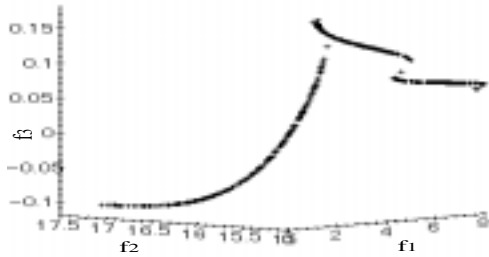


Figure 7: Pareto solutions of Example 2 (DRMOGA)

$$h(f_1, g) = \begin{cases} 1 - \left(\frac{f_1}{g}\right)^2, & \text{if } f_1 \leq g \\ 0, & \text{otherwise.} \end{cases} \quad (8d)$$

This is a very difficult problems to find the solutions, because the solutions are $x_i = 0 (i \neq 1)$ and $x_1 = [0.0, 0.2]$. Therefore, it often happens that the solutions are concentrated on the local minimum.

The results of this problem is summarized in Table 6.

Table 6: Results of Problem 3

Case	number of solutions	error	cover rate	generations
Simple Case1	500	7.21	0.41	394
Case2	456	5.92	0.32	612
Case3	469	3.75	0.14	1000
Case4	374	2.37	0.47	1000
Case5	482	3.84	0.48	1000
Case6	423	2.60	0.43	1000
Island Case1	345	6.41	0.46	570
Case2	322	5.88	0.48	919
Case3	301	3.70	0.22	1000
Case4	220	2.60	0.35	1000
Case5	283	3.39	0.43	1000
Case6	240	2.41	0.31	1000
DR Case1	412	6.87	0.38	533
Case2	363	5.38	0.28	774
Case3	425	4.53	0.40	780
Case4	293	0.01	0.99	1000
Case5	393	3.92	0.41	692
Case6	254	0.14	0.94	971

The derived Pareto solutions are shown in Figure 8 to 10. These are the results of case 4.

Only the case 4 and 6 of DRMOGA can derive the solutions. Especially, Simple GA nor island GA can not find the any real Pareto solutions. Therefore, it can be said that the DRMOGA has high searching ability in

difficult problems.

However, even in DRMOGA, the Pareto solutions cannot be derived with other parameters. The parameter α of crossover is big in the cases 4 and 6. This leads the fact that this problem needs the factor of high randomness. When the randomness of the algorithm is low, the Pareto solutions can not be derived.

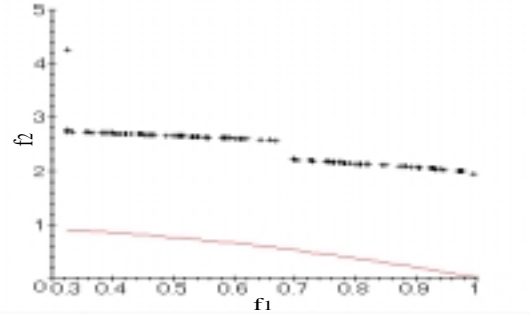


Figure 8: Pareto solutions of Example 3 (Simple)

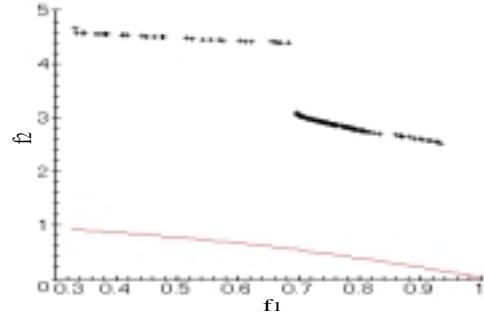


Figure 9: Pareto solutions of Example 3 (Island)

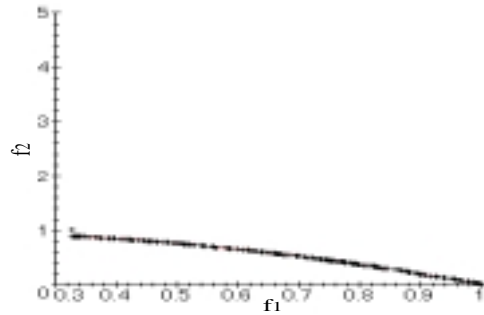


Figure 10: Pareto solutions of Example 3 (DRMOGA)

3.6 Example 4

Example 4 is also a problem of the two design variables that Deb developed (Deb 1999).

$$f_1 = x_1 \quad (9a)$$

$$f_2 = gh \quad (9b)$$

$$g(x_2, \dots, x_N) = 1 + 10 \frac{\sum_{i=2}^N x_i}{N-1} \quad (9c)$$

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{0.25} - \frac{f_1}{g} \sin(10\pi f_1) \quad (9d)$$

This is also a difficult problem and the Pareto solutions are not continuous but discrete. The results are summarized in Table 7.

Table 7: Results of Problem 4

Case	number of solutions	error	cover rate	generations
Simple Case1	500	1.70	0.31	209
Case2	500	1.71	0.38	358
Case3	470	0.32	0.22	1000
Case4	477	0.03	0.40	1000
Case5	492	0.34	0.58	855
Case6	493	0.08	0.60	899
Island Case1	385	1.89	0.40	333
Case2	409	1.75	0.53	403
Case3	304	0.31	0.33	1000
Case4	361	0.24	0.46	1000
Case5	376	0.27	0.60	1000
Case6	365	0.25	0.60	1000
DR Case1	494	1.93	0.37	212
Case2	457	3.10	0.34	54
Case3	460	0.39	0.30	262
Case4	451	0.03	0.52	387
Case5	442	0.39	0.47	291
Case6	402	0.07	0.61	654

Because the Pareto solutions are discrete, the highest value of the cover rate is around 0.6. Most of the cases, the values of errors are not so bad. On the other hand, in the most cases, the values of cover rate are not good. This means that the Pareto solutions are concentrated on some points. In any model, the parameter set of case 6 derives the good solutions. Therefore, it can be concluded that this problem needs very high randomness. In Figure 11, the derived Pareto solution in case 6 of DRMOGA is shown.

In case 5, island GA and simple GA derive the solutions whose values of the cover rates are good. On the other hand, the crossover of DRMOGA is not good in case 5. The Derived Pareto solution in case 5 of DRMOGA is shown in Figure 12.

From Figure 12, it is found that the Pareto solutions are not revealed all over the f_1 region. It may come from the crossover method. In this paper, the CNX is used for the crossover. In this method, the child individuals are close to the parent individuals. Therefore, it can be said that the randomness of this crossover method is low.

4 Conclusion

In this paper, we proposed the new parallel model of genetic algorithm for multi-objective optimization problems: that is called a Divided Range Multi-Objective

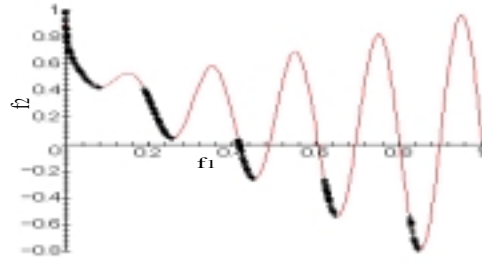


Figure 11: Pareto solutions of Example 4 (DRMOGA, Case 6)

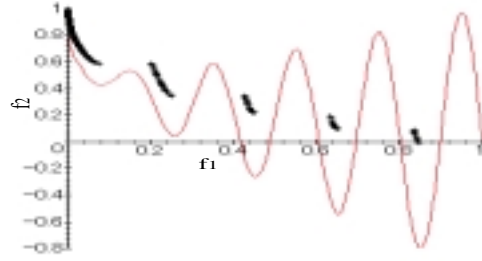


Figure 12: Pareto solutions of Example 4 (DRMOGA, Case 5)

Genetic Algorithm (DRMOGA). In the DRMOGA, the population is divided into some sub populations and normal multi-objective genetic algorithm is performed in each sub population. After some generations, all of the individuals are gathered virtually and they are sorted and divided into sub populations again. The proposed DRMOGA is applied to some numerical test functions. Through the numerical examples, DRMOGA has high searching ability of Pareto solutions. Especially, the operation of the sorting the individuals works well in searching the Pareto solutions in the difficult problems.

Bibliography

- [Coello 99] C. A. Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In *Proceedings of Congress on Evolutionary Computation*, pages 1–11, 1999.
- [Cantu-Paz 99] E. Cantu-Paz. Topologies, migration rates, and multi-population parallel genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 1, pages 91–98, 1999.
- [Deb 99] K. Deb. Construction of test problems for multi-objective optimization. In *Proceedings*

of the Genetic and Evolutionary Computation Conference (GECCO'99), pages 164–171, 1999.

- [Fonseca 93] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international conference on genetic algorithms*, pages 416–423, 1993.
- [Fonseca 94] C. M. Fonseca and p. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(6):1–16, 1994.
- [Goldberg 89] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [Hiyane 97] K. Hiyane. Generation of a set of pareto-optimal solutions for multiobjective optimization by parallel genetic algorithms and its quantitative evaluation. In *Proceedings of the No. 9 Distributed System Symposium*, pages 295–300, 1997.
- [Hiroyasu 99] T. Hiroyasu, M. Miki, and S. Watanabe. Divided range genetic algorithms in multiobjective optimization problems. In *Proceedings of International Workshop on Emergent Synthesis (IWES'99)*, pages 57–66, 1999.
- [Jones 98] B.R. Jones, W.A. Crossley, and A.S. Lyrintzi. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. In *Proceedings of the 7th AIAI/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 1–11, 1998.
- [Murata 95] T. Murata and H. Ishibuchi. Moga: Multiobjective genetic algorithms. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computing*, pages 289–294, 1995.
- [Nang 94] L. Nang and K. Matsuo. A survey on the parallel genetic algorithms. *J.SICE*, 33(6):500–509, 1994.
- [Sawai 99] H. Sawai and S. Adachi. Parallel distributed processing of a parameter-free ga by using hierarchical migration methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 1, pages 579–586, 1999.
- [Schaffer 85] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of 1st International Conference on Genetic Algorithms and Their Applications*, pages 93–100, 1985.
- [Tamaki 95] H. Tamaki, M. Mori, and M. araki. Generation of a set of pareto-optimal solutions by genetic algorithms. *Transaction of SICE*, 31(8):1185–1192, 1995.
- [Vicini 98] D.Q. Vicini. Sub-population policies for a parallel multiobjective genetic algorithm with applications to wing design. In *Proceedings of International Conference on Systems, Man, and Cybernetics*, pages 3142–3147, 1998.
- [Veldhuizen 99] D. A. V. Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithm test suites. In *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, 1999.

