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- This paper proposes Divided Range Multi-Objective Genetic Algorithm (DRMOGA), a model that uses parallel processing to perform genetic algorithms with multi-objective functions. In DRMOGA, GA individuals are sorted with respect to the values of the objective function and divided into sub populations. Simple GA for multi-objective problems is performed for each sub population. After a pre-determined number of generations, all individuals are gathered and sorted again. In this model, Pareto optimum solutions which are close to each other are collected into one sub population. In this way, calculation efficiency is increased and neighborhood searches can be performed. Numerical test cases show that DRMOGA is suitable GA model for parallel processing. In some cases, DRMOGA can derive better solutions when compared with both single population and distributed GA models.

1 Introduction

Multi-objective optimization techniques are often used to solve real world problems. Multi-optimization problems have several types of objectives which usually exhibit trade-off relationships. The Genetic Algorithm (GA) is one of the most powerful algorithms for solving multi-objective problems.

Several studies have been concerned with application of GA to multi-objective functions (Fonseca and Fleming, 1994; Tamaki, et al., 1995; Coello, 1999). Since GA is a multiple search method, it is suitable for finding Pareto optimum solutions. Several models have been are proposed for multi-objective GAs. Schaffer developed VEGA (1985) ; Goldberg et al. introduced the ranking method (1989); and Fonseca and Fleming developed the MOGA (1993). Their methods treat Pareto optimum solutions explicitly. Tamaki et al. (1995) introduced a model that used VEGA where Pareto optimum individuals are retained. [1] Murata et al. (1995) converted objective optimization problems to single objective optimization problems by weighing the values of each objective function.

Several models of multi-objective GAs are capable of deriving good Pareto optimum solutions. However, they are costly because they require numerous iterations to calculate the objective functions and constraints values. One solution to this problem is to use parallel processing to perform the multi-objective GA.

Several studies have been concerned with parallelization methods of GA for a single object (Nang and Matsuo, 1994; Cantu-Paz, 1999, Sawai and Adachi, 1999). There are a few studies of GA for multi-objective optimization problems and the models for these studies are similar to those for a single object. Jones and Crossley (1998) describe a one model in which evaluation of fitness calculations are performed in parallel. In another model, the total population is divided into sub populations and multi-objective optimization is performed for each sub population (Vicini 1998). However, the mechanism to search for the optimum differs between single and multi-objective GAs. In the single objective GA, only one optimum should be derived. Therefore, diversity of the searching point is important during the first stage and the local search is important during the later stage. On the other hand, because it should derive several points in addition to where point convergence occurs, both diversity and local search are important during all stages in the multi-objective GA. This fact suggests that a specified model should be used for multi-objective optimization using parallel processing.

We introduced a model called Divided Range Genetic Algorithms in Multi-Objective Optimization Problems: DRMOGA (Hiroyasu, et al. 1999) to address the need for a new model for parallel processing of multi-objective GA problems. In DRMOGA, individuals are divided into sub populations based on the values of their objec-

[1] Tamaki et al. called the individuals in the Pareto front "Pareto optimum individuals."

tive function, enabling efficient global and local searches to be performed. The present study explains the DRMOGA model and applies it to numerical test problems for which solutions are difficult to derive. The validity of the model and the characteristics of the solutions are discussed using these examples. The examples use test functions developed by Deb (1999) and demonstrate the effectiveness of DRMOGA's search performance.

2 Divided Range Multi-Objective Genetic Algorithm

2.1 Overview of DRMOGA

The flow of the Distributed Range Multi-Objective Genetic Algorithm can be explained as follows.

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

In this study, the number of distributions m and the sort interval k are determined in advance. Figure 1 shows the DRMOGA concept where there are two objective functions. Individuals are sorted into three divisions based on the value of the focused objective function f_1.

DRMOGA sub populations are determined with respect to the focused objective function. This mechanism is used to determine the pool in which they will be treated. Therefore, DRMOGA can produce a high degree of diversity in its derived Pareto optimum solutions.

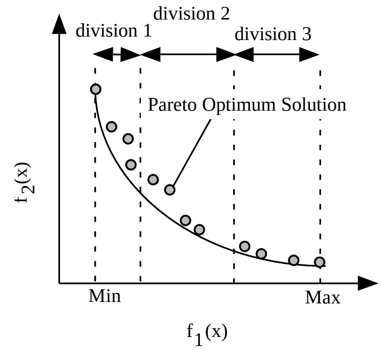

Figure 1: DRMOGA

2.2 Configuration of Genetic Algorithm

2.2.1 Expression of individuals

Individuals in genetic algorithms are usually shown in bit sequences. However, the example problems in this paper use real values. Therefore, individuals are the vectors of the real values and are shown as

$$a_1 = \{0.02, 10.03, \cdots, 7.52\}. \tag{1}$$

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 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 using the following equation.

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

In this equation, $\vec{r}_i$ is the vector of the parent individual. The concept of CNX is shown in Figure 2 where there are two design variables.

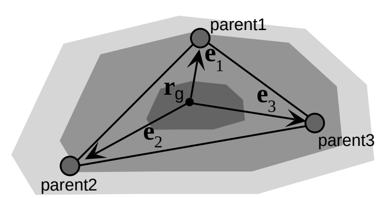

Figure 2: CNX Crossover

New individuals called "child individuals" $\vec{r}_{child}$ are generated using the following equation.

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

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

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

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

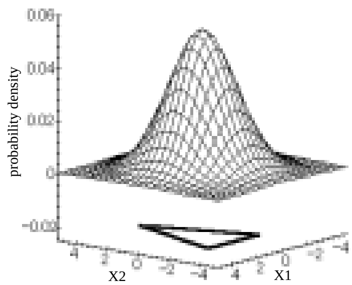

Figure 3: Normal Distribution ($\alpha = 3$)

It is obvious that where the parameter α is small, a child individual with the characteristics of the parents is generated. When the parameter α is big, the child individual is different from the parents. That is, when α is small, the child individual is closely related to the parents and when αa is big, child individual has been generated randomly. The optimum parameter of α differs depending on the problem type. The following examples present two types of α and examine the effect of the parameters on the solutions.

2.2.3 Selection

The following three selection methods were used for the example problems. Those are

- Pareto elite reservation + sharing function
- Roulette selection
- Roulette selection + sharing .

The first strategy selects an elite population composed of all of the individuals whose ranking is 1. When that population size is above a predetermined number, individuals are chosen by roulette selection using the fitness value. Fitness values are determined by the sharing function.

For the second strategy, the values of the fitness function are determined solely based on the ranking values.

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

2.2.4 Terminal condition

Many researchers have used the number of generations as the terminal conditions. However, this is not practical because it may cause premature termination before the optimum generation is derived.

We used 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

For the following test cases, the proposed DRMOGA model was adapted to four test functions using personal computer (PC) cluster systems. The validity of the DRMOGA model and the characteristics of it solutions are discussed using these adaptations.

3.1 Cluster System and Used Parameters

3.1.1 PC Cluster system

Table 1 summarizes specifications of the PC cluster system used in application of the DRMOGA model to the test functions.

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 Parameters used

Many parameters are required for GAs in multi-objective optimization problems . Table 2 shows the parameters used for the examples presented in this paper.

Table 2: Used parameters

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

The optimal parameters vary based on characteristics of the test functions. Solutions are affected by the selection method and the parameter α that is used in crossover. For these reasons, the six types of cases shown in Table 3 are applied for each example.

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 report the average of 30 trials. The DRMOGA results are compared with those for simple island and one population models. Results of DRMOGA, simple island and simple GA analyses are shown as "DR", "Island" and "Simple," respectively.

3.2 Matrix

One of the most difficult tasks in multi-objective optimization problems is to evaluate the Pareto optimum individuals. Because the Pareto optimum individuals represent an assembly of points, there is no good quantitative way of evaluating them. Many researchers only show derived Pareto optimum individuals in figures. This evaluation is not quantitative and is limited to two or three objective functions.

Hiyane (1997) introduced a matrix to assess accuracy and quality of Pareto optimum indivuduals. In the present study, the Hiyane's matrix is simplified and applied as follows.

3.2.1 Error

Pareto optimum solutions represent the average of Euclid distances between real Pareto solutions and each Pareto optimum individual. Where the error is small, the Pareto optimum individuals are very close to the real Pareto solutions. However, this matrix only can apply to problems where Pareto solutions are given. We used a shorthand expression for errors. In test functions, Pareto solutions exist on the constraints. Therefore, the following shorthand error is used.

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

Here, N expresses the number of Pareto optimum individuals. When $g(x) = 0$, the solutions are the real Pareto solutions.

3.2.2 Cover rate

Cover rate is the index for diversity of Pareto optimum individuals. The cover rate is derived as follows. First,

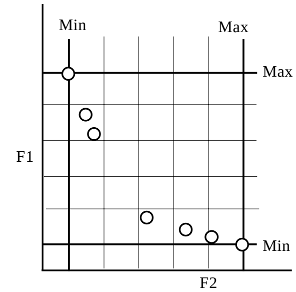

Figure 4: Cover rate

one of the object functions is focused. Then, the space between individuals with maximum and minimum values is broken up into a certain number of divisions. The number of division areas that have Pareto optimum individuals are counted. This number is divided by the total number of divisions. When every divided area has at least one Pareto optimum individual, this number becomes 1. When there are no areas that have Pareto optimum individuals, this number becomes 0. These steps are repeated for every objective function. Finally, an average cover rate is determined for each objective function. When the cover rate is close to 1, Pareto optimum individuals are not concentrated on one point. Figure 4 shows the cover rate concept where there are two objective functions.

3.3 Example 1

Example 1 is a convex problem that Tamaki et al. (1995) used. Among the four examples, this is the easiest for finding Pareto solutions.

$$f_1(x) = 2\sqrt{x_1} \tag{6a}$$
$$f_2(x) = x_1(1 - x_2) + 5 \tag{6b}$$
$$g_1(x) = x_1 - 1 \geq 0 \tag{6c}$$
$$g_2(x) = 4 - x_1 \geq 0 \tag{6d}$$
$$g_3(x) = x_2 - 1 \geq 0 \tag{6e}$$
$$g_4(x) = 2 - x_2 \geq 0. \tag{6f}$$

The results of this example are summarized in Table 4.

DRMOGA can derive Pareto solutions that have high accuracy in all six cases. In cases 1 and 2, Pareto individuals can be found easily because the problem is relatively simple; here, DRMOGA readily handles Pareto optimal selection. When the selection type is Pareto optimum (as in cases 1 and 2), the number of generations of DRMOGA is smaller than that for island or simple GAs. This demonstrates that DRMOGA can find Pareto solutions efficiently.

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

3.4 Example 2

In example 2, there are three objective functions and it is difficult to get Pareto solutions throughout the objective domain (Veldhuizen and Lamont. 1999). The following equations are used.

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

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

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

$$g_1(x) = x_1 \geq -3 \tag{7d}$$

$$g_2(x) = x_2 \leq 3. \tag{7e}$$

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

Derived solutions are shown in Figures 5 to 7. The results for case 1 are typical.

Here, it is obvious that DRMOGA derives better Pareto solutions compared with the island and simple GA models. Normally, results for simple GA are better than those for the island model. However, DRMOGA is an island model, and its results here are better than those for simple GA. The solutions for this problem 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 difficult. Because DRMOGA sorts based on the values of the objective functions, it can find solutions throughout the plane $f_2(x) = [15, 17.5]$.

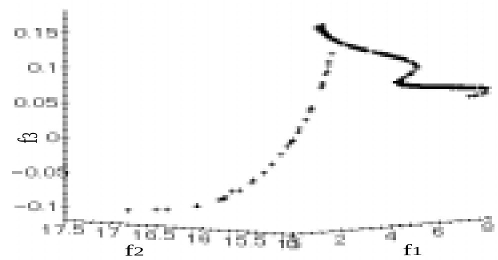

Figure 5: Pareto solutions of Example 2 (Simple)

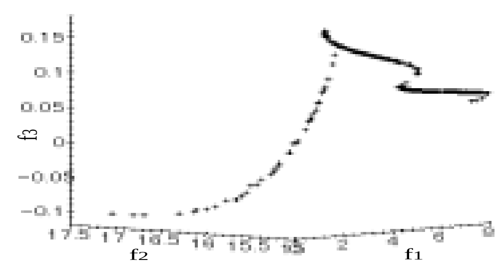

Figure 6: Pareto solutions of Example 2 (Island)

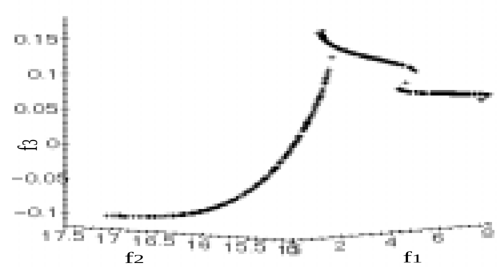

Figure 7: Pareto solutions of Example 2 (DRMOGA)

3.5 Example 3

Example 3 is a two design variable problem developed by Deb (1999). The equations are as follows:

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

$$f_2 = gh \tag{8b}$$

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

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

Because the solutions are$x_i = 0 (i \neq 1)$ and $x_1 = [0.0, 0.2]$, this is a very difficult problem. Solutions are frequently concentrated on the local minimum. Results are 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

Derived Pareto solutions for case 4 are shown in Figures 8 to 10.

DRMOGA can derive solutions for cases 4 and 6, where neither simple GA nor island GA can find any real Pareto solutions. Here, the parameter α of crossover is large. DRMOGA can solve this problem because it requires a great degree of randomness. Therefore, it can be concluded that DRMOGA has greater search ability for difficult problems. However, even in DRMOGA, Pareto solutions cannot be derived using other parameters where there is a low degree of randomness.

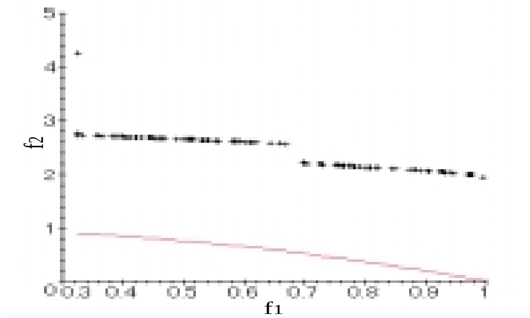

Figure 8: Pareto solutions of Example 3 (Simple)

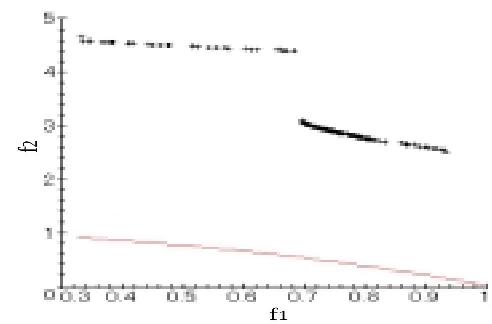

Figure 9: Pareto solutions of Example 3 (Island)

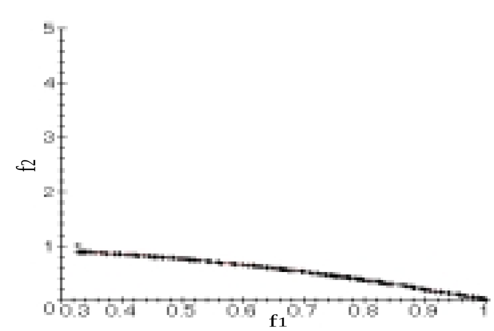

Figure 10: Pareto solutions of Example 3 (DRMOGA)

3.6 Example 4

Example 4 is another two design variable problem developed by Deb (1999).

$$f_1 = x_1 \tag{9a}$$

$$f_2 = gh \tag{9b}$$

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

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

This is also a difficult problem in which Pareto solutions are discrete rather than continuous. 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 Pareto solutions are discrete, the highest value of the cover rate is approximately 0.6. In many of the cases, the error values are not bad; however, the cover rate values are not good. Rather than being spread, Pareto solutions are concentrated on some points. Here, as in example 3, it can be concluded that the problem requires a high degree of randomness because the case 6 parameter set derives good solutions for all the models. Figure 11, shows Pareto solutions derived using the DRMOGA model for case 6.

For case 5, island GA and simple GA derive solutions with good cover rate values. The crossover rates for DRMOGA are not good in case 5. Figure 12 shows Pareto solutions derived by the DRMOGA model for case 5. Here, Pareto solutions are do not occur all over the f_1 region. This may be due to the crossover method used. The CNX method is used in this paper for crossover yields a low level of randomness because in generates child individuals that are close to parent individuals.

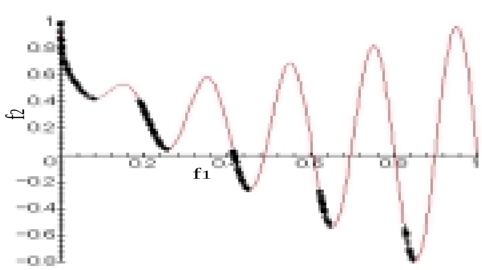

Figure 11: Pareto solutions of Example 4 (DRMOGA, Case 6)

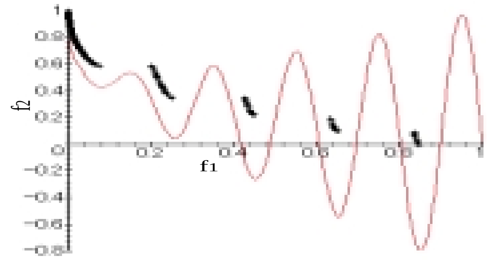

Figure 12: Pareto solutions of Example 4 (DRMOGA, Case 5)

4 Conclusion

This paper proposes a new parallel model of genetic algorithm for multi-objective optimization problems called Divided Range Multi-Objective Genetic Algorithm (DRMOGA). DRMOGA divides the population into sub populations and performs normal multi-objective genetic algorithms for each sub population. After some generations, all individuals are re-gathered and re-sorted. Numerical test examples demonstrate that the DRMOGA model is an effective and efficient method for searching for Pareto solutions. The model is particularly effective in finding Pareto solutions for difficult problems that generate highly random populations.

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 multi-objective optimization. *Evolutionary Computation*, 3(6):1–16, 1994.

[Goldberg 89] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesly, 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: Multi-objective 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.