Study on Effect of MOGA with Interactive Island Model using Visualization

Yamamoto Masafumi, Yoshikawa Tomohiro, *Member, IEEE* and Furuhashi Takeshi, *Member, IEEE*

***Abstract*— Genetic Algorithm is one of the most effective optimization algorithms, on which a lot of studies have been reported. Some studies on the application of island model, which is one of the representative methods to keep a diversity of solutions, to Multi-Objective Genetic Algorithm (MOGA) have been conducted. In MOGA, it is difficult to find the solutions which satisfy all objective functions because of their trade-off. Especially when there are many objective functions, it is obvious that it needs a lot of time to search for effective Pareto solutions and find them. This paper proposes the interactive way of addition and deletion of islands to the original ones based on user's requirements with the visualization of acquired solutions in island model for MOGA. This paper applies the proposed method to Nurse Scheduling Problem (NSP) using the visualization by Principal Component Analysis (PCA). Through the experiment, it is confirmed that an interactive tuning of the weights for the objective functions leaded to the acquisition of better Pareto solutions which a user wants while they are difficult to be acquired by the prepared weights.**

I. Introduction

Genetic Algorithm (GA), which comes after the model of living evolution, is one of the most effective optimization algorithms and a lot of studies have been reported. Recently, the application of GA to Multi-objective Optimization Problems (MOPs) has been focused[1]. One of the advantages is that the search in GA can be done with multi-point search. In MOPs, it is usually impossible to acquire the solutions which satisfy all objective functions because of their trade-off. Due to the difficulty, it is required to search for Pareto solutions which are superior to others at least an evaluation value. Especially in the case that the objective functions are large in number, high calculation or evaluation cost is needed to find Pareto solutions. In addition, GA also has a characteristic suitable for parallelization potentially and there are many studies which have tried to solve the problem of calculation cost using parallel computation. Three representative methods of parallel computation are as follows[2] : global single-population master-slave model, single-population fire-grained model and multiple-population coarse-grained model.

Master-slave model uses parallelization to make the calculation cost smaller. This model has a population or all solutions in a processor (master) and evaluating the solutions is carried out on other processors (slaves). This parallel GA is applied to the problems which need high calculation cost to evaluate solutions, although the model is not effective for the problems which need a lot of communication between the master and the slaves. In fire-grained model, each processor has one or a few solutions. This model is suitable for parallelization using a large number of computers, but it has the difficulty in creating the design which makes the use of the model effective. Multiple-population coarse-grained model is well-known as "island model", in which the solutions are divided into some subpopulations which are called islands. The evolution of the solutions in each island is basically independent from the other islands, then some solutions in each island sometimes migrate to another island for the exchange of information, which is called "migration".

As for single-objective optimization problems, lots of studies on island model have been reported. J. Tang et al. investigated the effects of migration topology, for example, random migration and ring-type migration, based on the calculation time[3]. M. Miki et al. proposed the island model in which each island had a different parameter set of genetic operations one another[4]. In GA, it is not easy to adjust the parameters of genetic operations such as population size appropriately because they strongly depend on applied problems. The above method has the advantage in terms of not needing to do preliminary experiments to find appropriate parameters. O. Hatanaka researched the parameters on island model thoroughly. For instance, the number of islands / individuals, the number / frequency of migration[5]. In the field of single-objective optimization problems, the advantage of the island models is mainly to keep the diversity of solutions.

On the other hand, some studies have began to be reported on the application of island model to Multi-Objective Genetic Algorithm (MOGA)[6][7][8][9]. In [6], MOGA is applied to the whole solutions for some generations and then the acquired solutions are divided into some islands based on the evaluation values of the objective functions. In [7], the evolution in each island is occurred as single objective optimization following each objective function, and MOGA is done in another island to satisfy all objective functions in parallel. In addition, [8] gives different weights of the objective functions to each island, then each island gives a role to search for solutions as a single-objective optimization problem. [9] investigates the effectiveness of island model changing the frequency of migrations, codes and so on. The purposes of these studies are to give each island a different role for a wider and effective search in total.

This paper proposes the way of addition of islands to

Yamamoto Masafumi is with the Department of Computer Science and Engineering, Nagoya University, Japan(email: yamasa@cmplx.cse.nagoya-u.ac.jp).

Yoshikawa Tomohiro is with the Department of Computer Science and Engineering, Nagoya University, Japan(email: yoshikawa@cse.nagoya-u.ac.jp).

Furuhashi Takeshi is with the Department of Computer Science and Engineering, Nagoya University, Japan(email: furuhashi@cse.nagoya-u.ac.jp).

the original ones and deletion of islands based on user's requirements interactively with the visualization result of acquired solutions in MOGA. The proposed method enables users to search for the solutions which they want effectively. This paper uses Principal Component Analysis (PCA)[10] to visualize the distribution of the solutions and the change of them, and it investigates the effectiveness of using island model in MOGA (MOGA with island model). Thanks to the visualization, users can know the progress of the evolution of solutions in each island and tendencies of the evolution, then MOGA can search around the area where they want. This paper applies the proposed method to Nurse Scheduling Problem (NSP)[11][12] and investigates the effectiveness of the proposed method. It is confirmed that an interactive tuning of the weights for the objective functions leaded to the acquisition of better Pareto solutions which a user wants while they are difficult to be acquired by the prepared weights.

II. Proposed Method

The steps of the proposed method are as follows:

- Step 1: Generate Pareto solutions by MOGA with island model for some generations. If a user can find satisfied solutions, the search is finished.
- Step 2: Reduce the dimension of the evaluation values to 2 dimensions by PCA, then project the whole solutions onto the 2 dimensional space. Only the solutions whose Pareto ranks are 1 in all are visualized.
- Step 3: Show colour contrast in proportion to the sum of the weighted evaluation values based on the weights that he/she gives to the objective functions. Based on the visualization, the user can add a new island with arbitrary weights decided by the user to the objective functions or delete one of the existing islands. The user can also see the average of the evaluation values of each island for this decision. Then go back to Step 1.

Firstly, Pareto solutions are generated by MOGA with island model for some generations to find various solutions, which helps user's decision-making. This paper employs the island model [7] which consists of some islands with single objectives, each objective function, and an island with all objective functions in parallel. Although it shows the effectiveness of the model in [7], it might be because the number of the objective functions were 2. Then it is not certain if the model performs well to find practical solutions when the problem has many objective functions like NSP, because the solutions with high fitness value just in a single objective function are not useful in practical problems. In the experiment of this paper, we have made a multi-objective island and some single-objective islands which we gave the weights preliminary to find more practical solutions. The weights employed in section IV. are shown in Table I.

Secondly, all acquired solutions are projected onto the 2 dimensional space which is formed by PCA and different colours are given to each island to distinguish among the islands. Only the solutions whose Pareto ranks are 1 in all are visualized. Through this operation, users can grasp the distribution of the solutions and tendencies of the evolution in the islands.

Thirdly, colour contrast is shown to the user in proportion to the weighted evaluation values based on the weights he/she gives to the objective functions. Thanks to this, the user can understand which objective function has an effect on which solutions and how much the solutions are affected by each objective function by giving some sets of weights. Exploiting this information, the user can add a new island where he/she wants to search with arbitrary weights to the objective functions decided by the user or delete one of the existing islands which seems unnecessary. To keep the number of solutions, the solutions of the new single-objective island are copied from the solutions in the existing islands in decreasing order of the weighted evaluation values and the same number of solutions in the multi-objective island is reduced. When users delete an island, all the solutions in the island are moved to the multi-objective island.

TABLE I
WEIGHTS OF EACH ORIGINAL ISLAND

	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$
Island 2	1	1	1	1	1	1	1	1	1
Island 3	0	1	1	1	1	1	1	1	1
Island 4	1	0	1	1	1	1	1	1	1
Island 5	1	1	0	1	1	1	1	1	1
Island 6	1	1	1	0	1	1	1	1	1
Island 7	1	1	1	1	0	1	1	1	1
Island 8	1	1	1	1	1	0	1	1	1
Island 9	1	1	1	1	1	1	0	1	1
Island 10	1	1	1	1	1	1	1	0	1
Island 11	1	1	1	1	1	1	1	1	0

III. Nurse Scheduling Problem : NSP

		Date																
Skill	Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	D	N	M
A	Staff A	N	N		N	N	M	N		D	D	M	N		D	3	6	2
A	Staff B		D	D	M	D	D	M	N	M			D	M	N	5	2	4
A	Staff C		D	D	M		N		D	D	M		N	D	D	6	2	2
B	Staff D	D	M		D	M	N	D		N	N	D	M		D	5	2	3
B	Staff E	D	D	M		N	D	D	M		D	D		N	N	6	3	2
B	Staff F	D	D	M	N	N		D	D	M		N	D	D	M	6	3	3
B	Staff G	N		D	D	M		N	D	D	M	N	D	D	M	6	3	3
C	Staff H	D	M	D	N		D	D	M		N	D	D	M	N	6	3	3
C	Staff I	M		N	D	D	M		N	D	D	M		D		5	2	3
C	Staff J	M	N	D		D	D	M		N	D	D	M	N	D	6	3	3
	D	4	4	4	3	3	4	4	3	4	4	4	4	4	4			
	M	2	2	2	2	2	2	2	2	2	2	2	2	2	2			
	N	2	2	1	3	3	2	2	2	2	2	2	2	2	3			

Fig. 1. Example of Nurse Scheduling Table

This section explains NSP, which is one of the multi-objective optimization problems in the real world. The nurse schedule is updated by a nurse chief of each department in every month. Making a nurse schedule takes a long time. A sample of nurse schedule is shown in Figure 1. In this schedule, one of three working patterns is allocated to each nurse (Staff-A to Staff-J). This schedule is a portion of one-month schedule. (In the experiment of this paper, one-month schedule with 26 nurses is employed.) The symbol D denotes day shift (AM8:00 - PM4:00), N is night shift (PM4:00 - AM0:00), and M is midnight shift (AM0:00 - AM8:00). A box without any symbol means a day off. In the three rows at the bottom of the schedule, the allocated numbers of the staff in each shift is shown. The leftmost column describes a nurse's skill level. The skill level A means that he/she is an expert in nursing. The skill level C means that he/she is a fresh person, and B is in the middle of A and C. The three rightmost columns describe the allocated days to each shift to each nurse. There are many constraints on this scheduling. One of them is the series of shifts for every nurse. An example of a prohibited pattern is to allocate a midnight shift right after the day off. Another constraint is that "One or more experts must be allocated at every midnight shift". NSP has many objective functions such as prohibition work patterns, balances among nurses' teams, fairness of holiday and so on. There are 12 objective functions. That means NSP is one of the multi-objective optimization problems whose objective functions are their constraints. The objective functions of NSP employed in this paper are as follows:

Obj1 The number of requisite nurses in each shift per day
Obj2 Level of requisite nurses in each shift per day
Obj3 Established prohibited working patterns
Obj4 Established compromised working patterns
Obj5 Established preferred working patterns
Obj6 Fairness of the number of working times on night or midnight shifts among nurses
Obj7 Fairness of the number of holidays among nurses
Obj8 Fairness of the number of successive holidays among nurses
Obj9 The prescript number of working times per month on night or midnight shifts in each nurse (within 8 times) (Note that the number of working times on night shifts is more than 3 and less than 5 times, that of midnight is more than 3 and less than 4 times)
Obj10 The prescript number of holidays per month in each nurse (2 days per week)
Obj11 Successive holidays on Saturday and Sunday in each nurse (one or more times per month)
Obj12 Successive holidays in each nurse (one or more times per month)

The number of violations in each objective function as described above is calculated on each candidate of solutions. The number of violations is employed as the evaluation value of each objective function.

IV. Experiment

A. Combine of objective functions

Generally it becomes difficult to search for effective solutions when the objective functions become large in number, because the required number of solutions exponentially increases for the search by the reason that it becomes easy to obtain various Pareto solutions rather than to obtain advanced solutions[13]. To reduce the number of the objective functions, some of the objective functions have been combined based on the correlation coefficients calculated in preliminary experiments using 6000 solutions at 1000th generation with Non-dominated Sorting Genetic Algorithm II (NSGA-II)[14].

The combined objective functions are as follows:

$f1$: *Obj1, Obj2*
$f2$: *Obj6, Obj9*
$f3$: *Obj7, Obj8*
$f4$: *Obj3*
$f5$: *Obj4*
$f6$: *Obj5*
$f7$: *Obj10*
$f8$: *Obj11*
$f9$: *Obj12*

TABLE II
Genetic Parameters

	Multi-objective Island	Single-objective Island
Number of Islands	1	10
Number of Solutions	900	100
Selection Rate	-	0.2
Crossover Rate	1.0	0.6
Mutation Rate	1 / gene length	1 / gene length

The set of parameters used in the experiment is shown in Table II. It used one point crossover and the way of mutation that changed a shift to another in the same day not to be against the constraint. The migration followed the same way in [7]. Island 1 was designed as a multi-objective optimization island and the other islands were single-objective optimization islands using the weights in Table I.

B. Visualization result

In this paper, the indexes of new islands added by a user are after island 12 while the original ones are island 1 to island 11.

Figure 2 is a result of visualization at 500th generation using the colour contrast and the slide bars at the left in Figure 2 are used to decide the weights for the objective functions. This figure shows the colour contrast result when every weight was the same. That means if a user thought every objective function was equally important, the solutions in the circle with dark colour could be regarded as what the user wanted and they would be searched for around Island 2, 7 and 10. Although we could grasp that, we could not find out how much good or bad the evaluation value in each objective function was through only this figure. Then the average of standardized evaluation values in each island will help us to know the detail. Note that "standardized" means the values which are normalized by using all generated solutions at the generation.

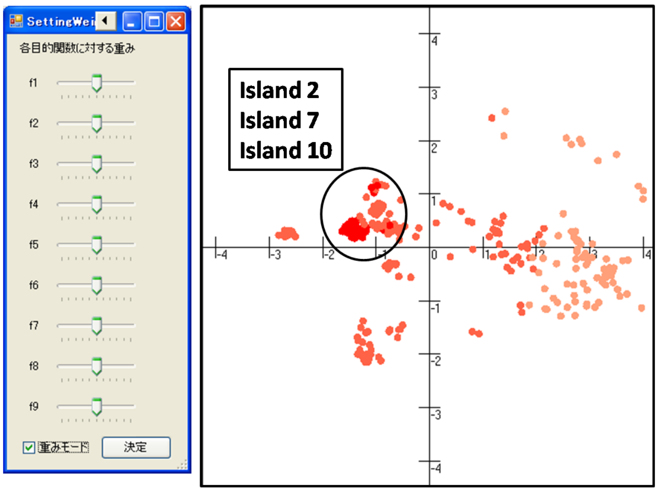

Fig. 2. Visualization result at 500th generation

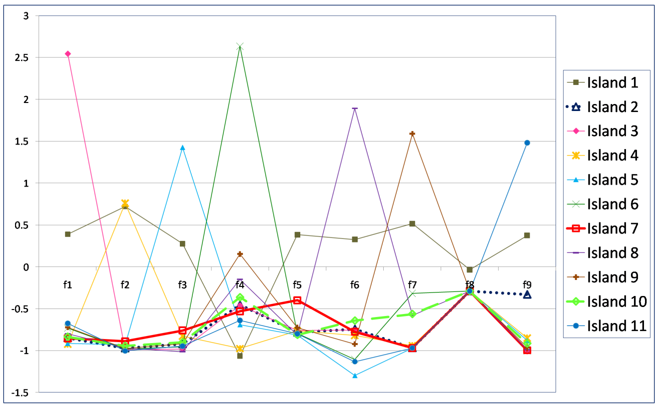

Fig. 3. Average of standardized evaluation value at 500th generation

Figure 3 shows the average of standardized evaluation values of Pareto solutions in each island, in which lower values are better as described in secrion III. In Figure 3, the lines of Island 2, 7 and 10 are bolder than the others to emphasize them. The vertical axis means standardized evaluation value and the horizontal axis means the combined objective functions. From this figure, it is clear that the evaluation values of $f4$ and $f6$ in Island 2, 7 and 10 are worse than the others relatively. Because of this result, a new island, Island 12, was created and the weights of $f4$ and $f6$ in the new island were made heavier than those of other objective functions. The new island and the weights for objective functions in Figure 3 are shown in Figure 4. Island 3, 4, 5, 6, 8, 9 and 11 were considered not being necessary here because the colour contrast in Figure 2 was lighter and the evaluation values were not acceptable.

C. Creating new islands

After the operations above, the solutions were evolved for another 100 generations and the result is shown in Figure 5.

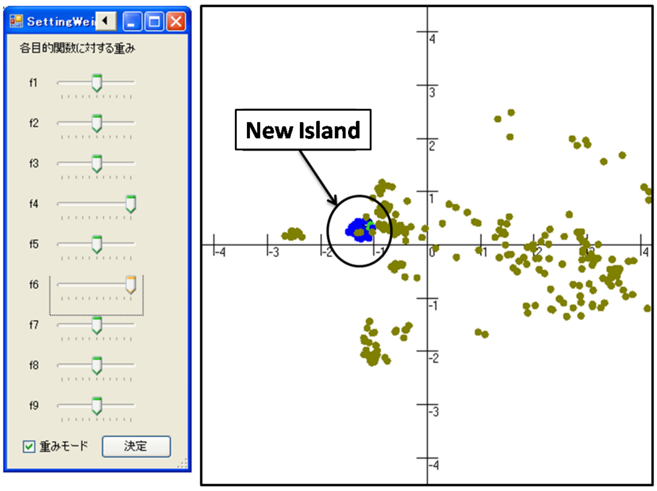

Fig. 4. New island

The definitions of the axes are the same as those in Figure 2. In Figure 5, the colour contrast is not clear, no solution

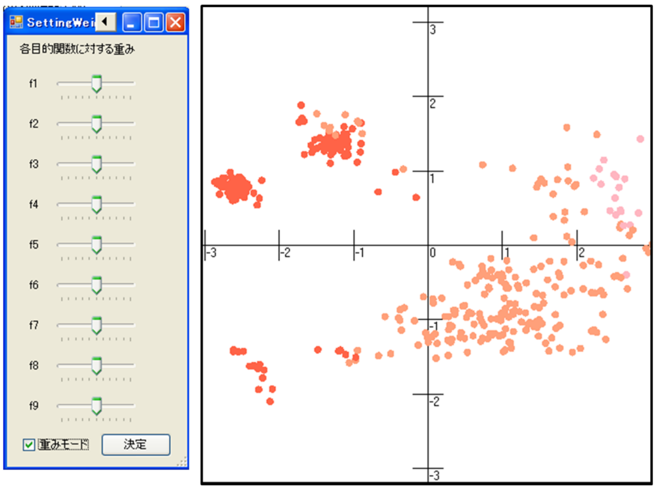

Fig. 5. Visualization result at 600th generation

clearly has dark or light colour compared with the result of visualization at 500th generation (Figure 2). That means the differences of the evaluation values among those solutions in the same weights for every objective function are small.

Figure 6 shows the average of standardized evaluation values at 600th generation. As for the lower values are better. According to this figure, the evaluation values of $f4$ and $f6$ in Island 12 are better than the others. In exchange for that, those of $f2$ and $f7$ in Island 12 became worse. Then next, a new island, Island 13, with heavy weights of $f2$ and $f7$ was designed. The weights of the new island were as follows: w=(w_1,w_2,$\cdots$,w_9)=(0.50, 1.00, 0.50, 1.00, 0.50, 1.00, 1.00, 0.50, 0.50). From this generation to 900th generation, new islands were made at every 100 generations. The weights of new islands, Island 14, 15 and 16, are shown in Table III.

The result of the standardized evaluation values at 1000th generation is shown in Figure 7. According to Figure 7,

TABLE IV

AVERAGE OF EVALUATION VALUE AT 1000TH GENERATION

	$Obj1$	$Obj2$	$Obj3$	$Obj4$	$Obj5$	$Obj6$	$Obj7$	$Obj8$	$Obj9$	$Obj10$	$Obj11$	$Obj12$
Island 2	**35.17**	0.17	30.00	**0.00**	-58.00	0.74	**0.15**	**0.32**	28.67	**0.00**	**0.00**	**0.00**
Island 7	35.75	**0.00**	29.13	1.00	-67.00	0.65	0.23	0.39	28.50	**0.00**	**0.00**	0.13
Island 10	35.62	0.08	25.62	**0.00**	-56.08	**0.60**	0.34	**0.32**	**28.15**	1.00	**0.00**	0.15
Island 12	40.61	0.09	**17.15**	0.06	**-74.58**	0.78	1.02	0.39	28.30	0.06	**0.00**	1.42

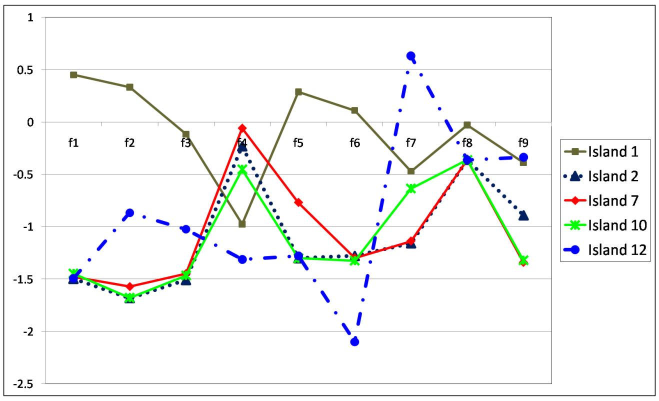

Fig. 6. Average of standardized evaluation value at 600th generation

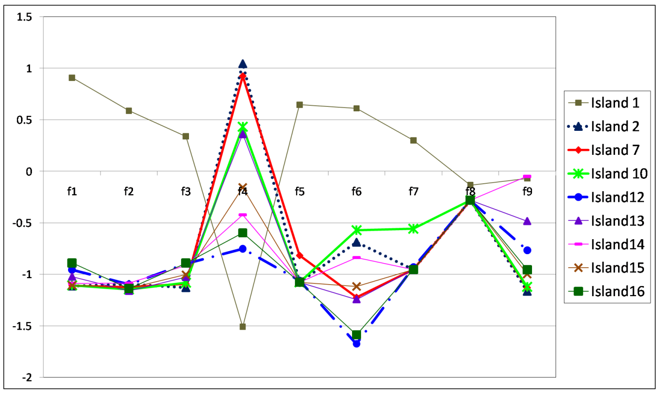

Fig. 7. Average of standardized evaluation value at 1000th generation

the evaluation values of the objective functions in designed islands, Island 12, 13, 14, 15 and 16, were better than those of the original islands, Island 2, 7 and 10, whose weights had not been changed. In addition, the evaluation values of $f2$ and $f7$ in Island 12 could get better evaluation values at 1000th generation compared to those at 600th generation.

As for Pareto solutions at 1000th generation, Table IV shows the averages of evaluation values in the original 12 objective functions in Island 2, 7, 10 and Island 12. In Table IV, the best evaluation value at each objective function among the islands is indicated by boldface. It is obvious that the islands which were made during the evolution could acquire good solutions in terms of the original 12 objective functions. This experiment assumed that the user thought every objective function was equally important. Generally

TABLE III

WEIGHTS OF NEW ISLANDS

	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$
Island 12	0.50	0.50	0.50	1.00	0.50	1.00	0.50	0.50	0.50
Island 13	0.50	1.00	0.50	1.00	0.50	1.00	1.00	0.50	0.50
Island 14	0.50	0.81	0.50	1.00	0.50	0.79	1.00	0.50	0.50
Island 15	0.50	0.62	0.50	1.00	0.50	0.79	1.00	0.50	0.50
Island 16	0.41	0.41	0.41	1.00	0.41	0.81	0.41	0.41	0.41

it is thought that the weights which correspond to this requirement are those in Island 2 with the same weights for every objective function. However, the difficulty in finding better solutions is different in each objective function, which was shown in this experiment, and the actual evaluation values in acquired solutions do not directly correspond to the prepared weights. One of the advantages of the proposed method is that users can interactively change or tune the weights for objective functions based on the evaluation values of acquired solutions during the search.

V. CONCLUSIONS

This paper proposed the way of addition of islands to the original ones and deletion of islands based on user's requirements interactively with the visualization result of acquired solutions in MOGA with island model. This paper applied the proposed method to NSP and showed it could be possible to grasp the distribution and the dominance for the user's requests on the Pareto solutions in islands search. Moreover, it was also confirmed that the proposed method could effectively search for the solutions which users wanted by adding with designed weights or deleting them based on the acquired solutions with prior weights. We will consider the way of the feedback of users' desire and that of visualization of solutions in future work.

REFERENCES

[1] H. Ishibuchi and T. Murata, *A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling*, IEEE Transactions on systems, man, and, cybernetics-part c: applications and reviews, Vol. 28, No. 3, pp. 392-403, 1998.

[2] Erick Cantú-Paz, *A Survey of Parallel Genetic Algorithms*, Calculateurs Paralleles, Vol. 10, No. 2, 1998.

[3] J. Tang, M. H. Lim, Y. S. Ong and M. J. Er, *Study of Migration Topology in Island Model Parallel Hybrid-GA for Large Scale Quadratic Assignment Problems*, The Eighth International Conference on Control Automation and System, Vol. 3, pp. 2286-2291, 2004.

[4] M. miki, T. Hiroyasu, M. Kaneko and K. Hatanaka, *A Parallel Genetic Algorithm with Distributed Environment Schema* (in Japanese), The institute of Electronics, Information and Communication Engineers, Vol. 99, No. 96 (19990528), pp. 87-94, 1999.

[5] K. Hatanaka, M. Miki and T. Hiroyasu, *Optimum Migration Interval in Multiple-Population Genetic Algorithms* (in Japanese), The Japan Society of Mechanical Engineers, 1999.

[6] T. Hiroyasu, M. Miki and S. Watanabe, *Divided Range Multi-Objective Genetic Algorithms* (in Japanese), Information Processing Society Japan (TOM), Vol. 41, pp. 79-89, 2000.

[7] T. Hiroyasu, M. Miki, T. Okuda and S.Watanabe, *Distributed Co-operation model of MOGA and SGA for Multiobjective Optimization Problems* (in Japanese), The Japan Society of Mechanical Engineers, pp. 25-26, 2001.

[8] J. Kamiura, T. Hiroyasu, M. Miki and S. Watanabe, *MOGADES: Multi-objective Genetic Algorithm with Distributed Environment Scheme*, Computational Intelligence and Applications, pp. 143-148, 2002.

[9] Z. Skolicki and K. De Jong, *Improving Evolutionary Algorithms with Multi-representation Island Models*, Lecture Notes in Computer Science, Parallel Problem Solving from Nature - PPSN VIII, Vol.3243, pp. 420-429, 2004.

[10] Y. Tanaka and K. Wakimoto, *Methods of Multivariate Statistical Analysis* (in Japanese), Gendaisuugakusha, 2004.

[11] B. Cheang, H. Li, A. Lim and B. Rodrogues, *Nurse rostering problems - a bibliographic survey*, European Journal of Operational Research 151, pp. 447-460, 2003.

[12] D. Yamashiro, T. Yoshikawa and T. Furuhashi, *Support to Select Satisfying Solutions Using Visualization Method in Multi-Objective Optimization Problem* (in Japanese), Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, Vol. 20, No. 6, pp. 850-859, 2008.

[13] H. Ishibuchi, N. Tsukamoto and Y. Nojima, *Evolutionary Many-Objective Optimization* (in Japanese), Shinkakeisan Synposium, pp. 47-50, 2007.

[14] Deb K., Pratap A., Agarwal S. and Meyarivan T., *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*, IEEE Transactions on evolutionary computation, Vol. 6, No. 2, pp.182-197, 2002.