

# Multi-objective Optimization with Improved Genetic Algorithm

Hiroyuki Ishibashi, Hernán E. Aguirre, Kiyoshi Tanaka and Tatsuo Sugimura  
Faculty of Engineering, Shinshu University, Japan  
E-mail: {isibasi@sugi, {ahernan, ktanaka, tsugimu}@gipwc}.shinshu-u.ac.jp

## Abstract

In this work, we extend an improved GA (GA-SRM) to multi-objective flowshop scheduling problem (FSP) in order to obtain better pareto-optimum solutions (POS). Two kinds of cooperative-competitive genetic operators in GA-SRM, CM and SRM, are extended to the ones suitable for FSP in which solutions (individuals) are represented as permutations. Simulation results verify that GA-SRM shows better performance for multi-objective optimization problem (MOP), and consequently better POS are obtained rather than conventional approaches with canonical GA.

## 1 Introduction

Evolutionary Algorithms (EAs) have drawn great attention as one of the most powerful techniques to solve various kinds of optimization problems. In particular, a number of methods and applications using Genetic Algorithms (GAs) have so far been developed mainly for single objective optimization problems (SOP). However, because many problems in real world applications, such as decision making, include multiple objective functions in trade-off relationship, the problem often becomes a multi-objective optimization problem (MOP). In this case, it is required to obtain a set of non-dominated (compromised) solutions called pareto-optimal solutions (POS)[1]-[3]. There has been several works that apply GA to solve MOP in order to obtain POS[2]-[6] so far.

On the other hand, although many works that improve the search performance of GA have been proposed, recent works have pointed out that the mutation operator plays more important roles to improve the performance of GA[7]-[11]. An improved GA (GA-SRM)[10] introduces an adaptive mutation operator (called Self-Reproduction with Mutation: SRM) in parallel with conventional crossover with small mutation operator (called Crossover and Mutation: CM). An extinctive selection is applied to offspring created by both SRM and CM operators in GA-SRM to guarantee the preservation of the beneficial genetic information to the next generation. Compared with canonical and other enhanced GAs, GA-SRM remarkably accelerates the search speed and shows robust

and reliable search performance for SOP in the 0/1 multiple knapsack problem[10] and in the image halftoning problem[11] as well.

In this work, we especially focus on solving the multi-objective flowshop scheduling problem (FSP)[12], and try to obtain better POS by using GA-SRM. Two kinds of genetic operators in GA-SRM, CM and SRM, are extended suitable for FSP in which solutions (individuals) are represented as permutations of job numbers. Simulation results verify that GA-SRM shows better performance for MOP, and consequently better POS are obtained rather than conventional approaches with canonical GA.

## 2 Solving MOP with GA

### 2.1 MOP

The optimization of POS in MOP is defined as follows[5]. Here we consider the minimization problem. For a vector evaluation function

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_K(\mathbf{x})) \quad (1)$$

consisted of  $K$  kinds of objective functions  $f_i (i = 1, 2, \dots, K)$ , we try to minimize  $f_i (i = 1, 2, \dots, K)$  with  $\mathbf{x}$  that is an element of the solution space  $\Psi$ . It is, in general, difficult to minimize all  $f_i (i = 1, 2, \dots, K)$  for  $\mathbf{x} \in \Psi$  because there is a trade-off relationship between objective functions. In such cases, we must try to minimize each  $f_i(\mathbf{x})$  by compromising them. A solution  $\mathbf{x}_u \in \Psi$  of which evaluation vector is  $\mathbf{u} = \mathbf{f}(\mathbf{x}_u) = (u_1, u_2, \dots, u_K)$  is considered as a POS if and only if there is no dominated solution  $\mathbf{x}_v \in \Psi$ . It is defined that a solution  $\mathbf{x}_v$  of which evaluation vector is  $\mathbf{v} = \mathbf{f}(\mathbf{x}_v) = (v_1, v_2, \dots, v_K)$  dominates a solution  $\mathbf{x}_u$  if

$$\forall i \in \{1, 2, \dots, K\}, v_i \leq u_i \quad (2)$$

and

$$\exists i \in \{1, 2, \dots, K\}, v_i < u_i \quad (3)$$

are satisfied. The objective of MOP is to obtain a set of POS  $\{\mathbf{x}_u\}$ .

## 2.2 GA Approaches

There have so far been proposed several approaches to solve MOP by using GA[2]-[6]. In this paper we focus on [6], which uses random weight vectors

$$w = (w_1, w_2, \dots, w_K) \quad (4)$$

and try to minimize

$$f_w(x) = \sum_{i=1}^K w_i f_i(x) \quad (5)$$

as an evaluation function of Eq.(1). Here each component  $w_i (i = 1, 2, \dots, K)$  in  $w$  is calculated by

$$w_i = \frac{r_i}{\sum_{j=1}^K r_j} \quad (6)$$

with random variables  $r_j (j = 1, 2, \dots, K)$  where  $w_i \geq 0, \sum_{i=1}^K w_i = 1$ . In this approach, an independent selection is made by Eq.(5) with a different weight vector  $w$  every time a new offspring is created. Since  $\lambda$  times of selection are made with  $\lambda$  random vectors to create  $\lambda$  new offspring for the next generation, it is expected to search solutions towards  $\lambda$  kinds of direction in a solution space. If the weight vector in Eq.(5) is fixed to a specific vector, on the other hand, the problem will become SOP. In this case it is difficult to obtain desirable POS because it is expected to search solutions towards a fixed direction. Since random weight vector selection by Eq.(5) makes GA introduce much diversity into the population, we can expect to obtain better POS with higher fitness values. The flow of solving MOP with GA is illustrated in Fig.1, in which a sub-population consisted with only POS is separately preserved as well as actual population that is evolved with GA generation by generation. An elitist strategy that includes  $E$  individual of POS into the next actual population is adopted similar to [6].

## 3 Extension of GA-SRM to Multi-objective FSP

### 3.1 Multi-objective FSP

FSP (Flowshop Scheduling Problem) is well-known as one of combinatorial problems that optimize the permutations of job numbers[12]. There are  $N$  kinds of jobs and  $M$  kinds of machines in a given problem. The processing times  $t(k, l)$  for job  $k (1 \leq k \leq N)$  on machine  $l (1 \leq l \leq M)$  are predetermined. Let us denote a solution (job permutation)  $x = (J_1, J_2, \dots, J_N)$  and  $J_k (1 \leq k \leq N)$  as  $k$ -th job to be processed. Job  $J_k$  must be serially processed from machine 1 to  $M$  but cannot be in process on

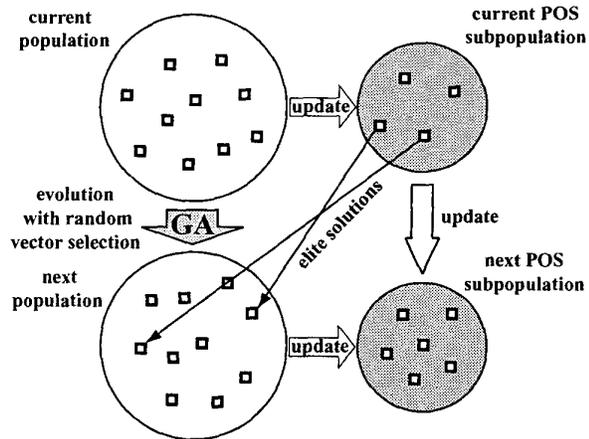


Fig.1 The flow of solving MOP with GA

the next machine until the previous job  $J_{k-1}$  is completed on it. Also, let us define the time to begin  $J_1$  on the machine 1 as 0, and the time when  $J_k$  is completed on the machine  $l$  as  $T(J_k, l)$ . In FSP the following relationships are formulated.

$$T(J_1, 1) = t(J_1, 1) \quad (7)$$

$$T(J_k, 1) = T(J_{k-1}, 1) + t(J_k, 1) \quad (k = 2, \dots, N) \quad (8)$$

$$T(J_1, l) = T(J_1, l-1) + t(J_1, l) \quad (l = 2, \dots, M) \quad (9)$$

$$T(J_k, l) = \max\{T(J_{k-1}, l), T(J_k, l-1)\} + t(J_k, l) \quad (10)$$

$(k = 2, \dots, N; l = 2, \dots, M)$

In this work, we use the following three kinds of criteria in FSP and combine them in order to construct MOP.

- (i) **makespan** : it is defined as the total processing time that all machines completed all jobs.

$$T_m = T(J_N, M) \quad (11)$$

- (ii) **total flowtime** : it is defined as the sum of flowtime, which is the time a job spends in process from machine 1 to machine  $M$ .

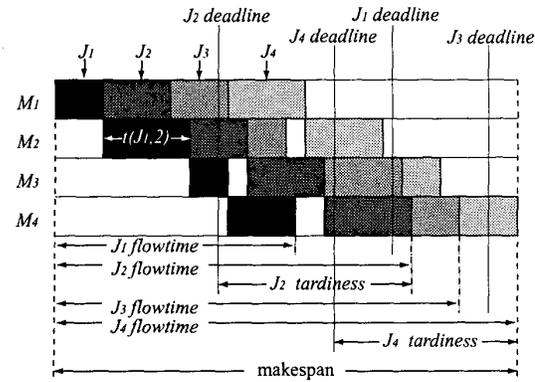
$$T_f = \sum_{k=1}^N T(J_k, M) \quad (12)$$

- (iii) **total tardiness** : it is defined as the sum of tardiness, which is the time a job is delayed for the predetermined deadline.

$$T_t = \sum_{k=1}^N \max\{0, T(J_k, M) - d_k\}, \quad (13)$$

where  $d_k (k = 1, 2, \dots, N)$  denotes the deadline for job  $J_k$ .

An example of FSP with four jobs ( $N = 4$ ) and four machines ( $M = 4$ ) is illustrated in Fig.2.



$$\begin{aligned} \text{Total Flowtime} &= J_1 \text{ flowtime} + J_2 \text{ flowtime} + J_3 \text{ flowtime} + J_4 \text{ flowtime} \\ \text{Total Tardiness} &= J_1 \text{ tardiness} + J_2 \text{ tardiness} + J_3 \text{ tardiness} + J_4 \text{ tardiness} \\ &\quad (J_1 \text{ tardiness} = 0, J_3 \text{ tardiness} = 0) \end{aligned}$$

Fig.2 An example of FSP ( $N = 4, M = 4$ )

### 3.2 GA-SRM

GA-SRM[10] uses two kinds of cooperative and competitive genetic operators in parallel, CM and SRM, to produce offspring and assign them specific roles. Offspring created by both operators compete for survival through an extinctive selection mechanism.

CM (Crossover and Mutation) operator creates offspring by conventional crossover and successive mutation with small mutation probability  $p_m^{(CM)}$ . This operator has a role to propagate beneficial genetic information into the population by combining segments from parent individuals. CM creates  $\lambda_{CM}$  offspring.

On the other hand, SRM (Self-Reproduction with Mutation) operator creates offspring by an adaptive mutation operator with dynamic mutation probability  $p_m^{(SRM)}$  varying from high to low values. This operator has a role to introduce diversity into the population by creating offspring that cannot be created by CM. SRM creates  $\lambda_{SRM}$  offspring.

Offspring created by CM and SRM compete for survival through  $(\mu, \lambda)$  proportional selection which is widely used in Evolution Strategy (ES)[13]. This method selects only the best  $\mu$  offspring by discarding offspring with low fitness value from  $\lambda (= \lambda_{CM} + \lambda_{SRM})$  offspring created, and preserve them as parent individuals for the next generation.

### 3.3 Extension to Multi-Objective FSP

Local permutations of job numbers are important rather than their loci on an individual in FSP. Thus the genetic operators in GA-SRM originally designed for bi-

nary representation[10],[11] are accordingly modified for permutation representation in this work. Two-point order crossover[14] and a shift mutation scheme[15] with small mutation probability are used in CM. The illustration of two-point order crossover is shown in Fig.3. The same job number cannot appear twice in a chromosome in FSP. Here we randomly select two crossing points and decide which segment (parent A or B, and inside or outside of two points) should be inherited to the offspring. In this example, the genetic information of the segment inside  $c_1$  and  $c_2$  in parent A and the remaining job numbers in parent B were inherited to the offspring being created. After the two-point crossover a shift mutation is applied to the offspring with small probability  $p_m^{(CM)}$  as shown in Fig.4. The initial segment point and the direction of shifting are randomly determined. This small mutation works to introduce small change into a beneficial combination obtained by crossover.

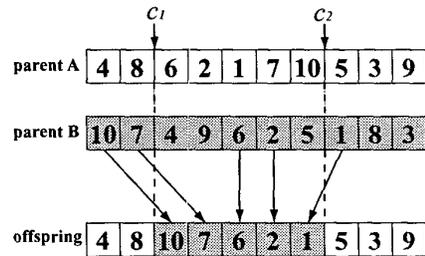


Fig.3 Two-point order crossover

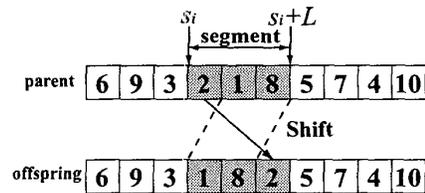


Fig.4 Shift mutation

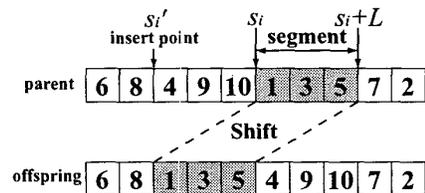


Fig.5 Extended shift mutation

On the other hand, an adaptive mutation scheme that is extended from the shift mutation in this work is used in SRM in order to get offspring that cannot be created by CM. The illustration of the extended shift mutation is shown in Fig.5. The initial point of a segment  $s_i$  and a segment inserting point  $s_i'$  are randomly determined, and then the segment is extracted and inserted at  $s_i'$ . Here the segment

length  $L$  is properly reduced by controlling the mutation probability  $p_m^{(SRM)}$  from high to low values.

The segment length  $L$  in both shift and extended shift mutation schemes is determined as the times a random valuable is below  $p_m^{(CM)}$  or  $p_m^{(SRM)}$  in  $n$  times of trial. Although  $p_m^{(CM)}$  is set to a constant small value,  $p_m^{(SRM)}$  are dynamically controlled based on the normalized mutant survival ratio in SRM specified by

$$\gamma = \frac{\mu_{SRM}}{\lambda_{SRM}} \cdot \frac{\lambda}{\mu} \quad (14)$$

where  $\mu_{SRM}$  denotes SRM's offspring number that survived extinctive selection. That is, we consider the contribution by SRM is no longer effective if  $\gamma < \tau$ , which is a predetermined threshold, and reduce  $p_m^{(SRM)}$  to  $p_m^{(SRM)}/\beta$  ( $\beta > 1$ ). This adaptive mutation control preserves proper balance between CM and SRM operators and offspring created by SRM effectively contributes to improve the search performance of the algorithm.

Finally, we show how to apply the extinctive selection mechanism into MOP. Here in Eq.(5) the  $(\mu, \lambda)$  proportional selection is done with a random vector in order to select two parents every time an offspring is created by CM or SRM.  $\lambda$  times of  $(\mu, \lambda)$  proportional selection are totally done by using  $\lambda$  kinds of vectors in Eq.(5), which expects to improve the search performance by discarding offspring with low fitness value in  $\lambda$  kinds of different direction.

## 4 Simulation Results and Discussion

### 4.1 Test Problems and Evaluation Method

Computer simulations were conducted for 30 jobs-10 machines FSP. Two or three criteria are selected from  $T_m$ ,  $T_f$  and  $T_t$ , and used with minus value as multi-objective functions since these criteria should be minimized in FSP. Average results are observed for 100 random test problems in which processing times  $t(k, l)$  for job  $k$  ( $1 \leq k \leq N$ ) on machine  $l$  ( $1 \leq l \leq M$ ) are randomly determined in the range  $[1, 99]$ . Also, deadlines for each job  $d_k$  ( $k = 1, 2, \dots, N$ ) are randomly determined by adding random variables  $[-200, 0]$  to flowtimes  $T^*(J_k, M)$  for a specific job permutation  $\mathbf{x}^* = (J_1^*, J_2^*, \dots, J_N^*)$ .

In this work, Set Quality Measure (SQM) [16] is used in order to evaluate the obtained POS sub-population. SQM for a population  $\Omega$  is calculated by

$$Q(\Omega) = \frac{1}{R} \sum_{m=1}^R \max \left\{ \sum_{i=1}^K w_i^{(m)} f_i(\mathbf{x}) \mid \mathbf{x} \in \Omega \right\} \quad (15)$$

where  $w_i^{(m)}$  is an  $i$ -th element in a random weight vector  $\mathbf{w}^{(m)}$  determined at  $m$ -th selection, and  $R$  is the number of random vectors to be used.

### 4.2 Obtained Results and Discussion

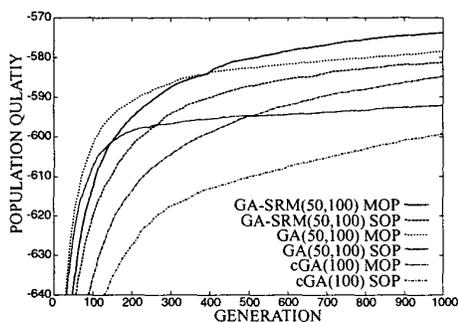
We conducted computer simulation for test problems described in 4.1 with parameters in Table 1. In this work, the following comparison are done: (i) the results for MOP (using random weight vectors in Eq.(5)) and for SOP (using a fixed vector with same weight), (ii) the results by a canonical GA (denoted as cGA) and GA-SRM with  $(\mu, \lambda)$  proportional selection (denoted as GA-SRM( $\mu, \lambda$ )), (iii) the results by a canonical GA with  $(\mu, \lambda)$  proportional selection (denoted as GA( $\mu, \lambda$ )) and GA-SRM( $\mu, \lambda$ ) in order to isolate the contribution by extinctive selection and SRM operator. Since there is no concept of POS in SOP, solutions equivalent to POS found in the search process are preserved for SOP.

The transition of SQM and final POS obtained for  $K = 2$  (two objective functions) and  $K = 3$  (three objective functions) are shown in Fig.6 and Fig.7, respectively. In case of  $K = 3$ , POS are plotted by mapping them onto a 2-dimensional plane. (i) All approaches, cGA(100), GA(50,100) and GA-SRM(50,100), for MOP clearly attained better results than those for SOP, which explains random vector evaluation by Eq.(5) is effective in order to obtain better POS in MOP. (ii) GA-SRM outperforms cGA in both search velocity and search reliability. In addition, GA-SRM(50,100) for SOP (a fixed weight vector was used in Eq.(5)) attained same or better performance compared with cGA(100) for MOP. These results means SRM operator and extinctive selection effectively contributes to search POS in MOP. (iii) Let us see the details on SQM transition. The search velocity of GA-SRM(50,100) tends to be inferior to GA(50,100) at the beginning of the search. This means CM contributes more than SRM and  $(\mu, \lambda)$  proportional selection assists its contribution in initial generations. Therefore, it is concluded that the acceleration of search speed in GA-SRM is mainly achieved by  $(\mu, \lambda)$  proportional selection mechanism. However, while the search performance of GA(50,100) is gradually deteriorated, the one of GA-SRM(50,100) is successively improved without premature convergence.

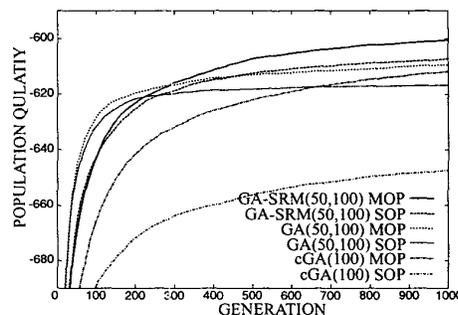
This fact is verified in Fig.8, which shows the ratio of average POS number newly appeared for created offspring number by each operator in each generation. From this figure, although SRM's contribution is inferior to CM's one at the beginning of the search process, SRM begins to increase its contribution around 100-th generation, and after 200-th generation keeps superior contribution to CM in GA(50,100). Also, it should be noted that CM's contribution in GA-SRM(50,100) is a little bit better than that in GA(50,100), which shows the importance of synergetic effects of two cooperative and competitive genetic operators to improve the performance of GA. Therefore, it is

**Table 1 Genetic algorithms parameters**

Parameter	cGA(100)	GA(50, 100)	GA - SRM(50, 100)
Representation	permutation	permutation	permutation
Selection	Proportional	$(\mu, \lambda)$ Proportional	$(\mu, \lambda)$ Proportional
Scaling	Linear	Linear	Linear
Crossover	two-point order	two-point order	two-point order
Mating condition	$(x_i, x_j), i \neq j$	$(x_i, x_j), i \neq j$	$(x_i, x_j), i \neq j$
$p_c$	0.8	0.8	0.8
Mutation(CM)	Shift	Shift	Shift
$p_m^{(CM)}$	0.025	0.025	0.025
Mutation(SRM)	-	-	Extended Shift
$p_m^{(SRM)}$	-	-	$[0.5, 0.025], \tau = 0.48, \beta = 1.5$
$\mu : \lambda$	-	1 : 2	1 : 2
$\lambda_{CM} : \lambda_{SRM}$	-	-	1 : 1
E	SOP : 1, MOP : 3	SOP : 1, MOP : 3	SOP : 1, MOP : 3
R	100	100	100

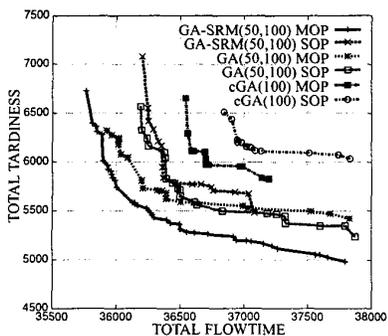


(a)  $K = 2 (T_m \& T_f)$

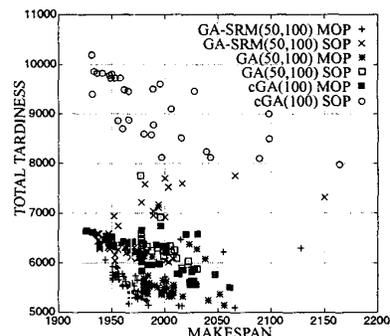


(b)  $K = 3 (T_m \& T_f \& T_t)$

**Fig.6 Transition of SQM**



(a)  $K = 2 (T_f \& T_t)$

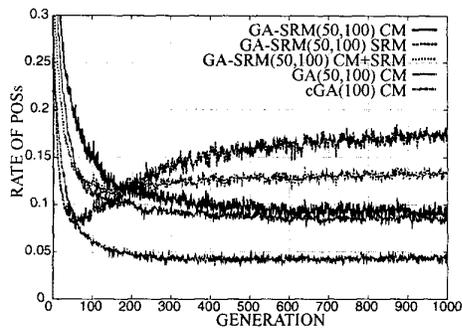


(b)  $K = 3$  (mapped onto  $T_m \& T_t$  plane)

**Fig.7 Final POS obtained**

**Table 2 Domination ratio of GA-SRM (MOP) for other approaches**

approaches	K=2			K=3	average
	$T_m \& T_f$	$T_m \& T_t$	$T_f \& T_t$	$T_m \& T_f \& T_t$	
GA - SRM(50, 100) SOP	75%	84%	76%	79%	79%
GA(50, 100) MOP	69%	83%	73%	80%	76%
GA(50, 100) SOP	85%	89%	79%	89%	86%
cGA(100) MOP	79%	84%	81%	88%	83%
cGA(100) SOP	95%	96%	94%	100%	96%



**Fig.8 Contribution of each operator**

concluded that the robust and reliable search performance in GA-SRM is mainly achieved by SRM operator. Consequently, we could obtain better POS by GA-SRM rather than conventional approaches with canonical GA. The ratio which POS obtained by GA-SRM for MOP dominates those obtained by other approaches are shown in **Table 2**.

## 5 Conclusions

In this work, we have extended an improved GA (GA-SRM) to multiple-objective flowshop scheduling problem (FSP). Two kinds of cooperative-competitive genetic operators in GA-SRM, CM and SRM, could be effectively extended for permutation representation in FSP, and consequently GA-SRM achieved clearly better results in MOP rather than conventional approaches with canonical GA for both solution set quality (SQM) and POS. In addition, we isolated the contribution by extinctive selection and SRM operator in GA-SRM; the former contributes to accelerate the search speed and the latter contributes to robust and reliable search performance.

As future works, we are planning to improve the domination ratio of POS by GA-SRM further, and apply GA-SRM for another kind of MOP.

## References

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.
- [2] J. D. Schaffer, "Multi-objective optimization with vector evaluated genetic algorithm", *Proc. 1st ICGA*, Morgan Kaufmann, pp.93-100, 1985.
- [3] F. Kursawa, "A variant of evaluation strategies for vector optimization", *Proc. Parallel Problem Solving from Nature*, pp.193-197, Springer-Verlag, 1991.
- [4] J. Horn, N. Nafpliotis and D. E. Doldberg, "A niched Pareto genetic algorithm for multi-objective optimization", *Proc. 1st IEEE ICEC*, pp.82-87, 1994.
- [5] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization", *Proc. 5th ICGA*, Morgan Kaufmann, pp.416-423, 1995.
- [6] T. Murata and H. Ishibuchi, "MOGA: Multi-objective genetic algorithms", *Proc. 2nd IEEE ICEC*, pp.289-294, 1995.
- [7] R. Hinterding, H. Gielewski and T. C. Peachey, "The nature of mutation in genetic algorithms", *Proc. 6th ICGA*, Morgan Kaufmann, pp.65-72, 1995.
- [8] T. C. Fogarty, "Varying the Probability of Mutation in the Genetic Algorithm", *Proc. 3rd ICGA*, Morgan Kaufmann, pp.104-109, 1989.
- [9] T. Back and M. Schutz, "Intelligent Mutation Rate Control in Canonical Genetic Algorithm", *Lecture Notes on Artificial Intelligence*, Vol.1079, Springer, pp.158-167, 1996.
- [10] H. E. Aguirre, K. Tanaka and T. Sugimura, "Cooperative Model for Genetic Operators to Improve GAs", *Proc. IEEE ICIS*, pp.98-106, 1999.
- [11] H. E. Aguirre, K. Tanaka and T. Sugimura, "Accelerated Halftoning Technique Using Improved Genetic Algorithm with Tiny Populations", *Proc. IEEE SMC*, pp.905-910, 1999.
- [12] R. L. Daniels and R. J. Chambers, "Multi-objective flowshop scheduling", *Naval Research Logistics*, Vol.37, pp.981-995.
- [13] T. Back, *Evolutionary Algorithms in Theory and Practice*, Oxford Univ. Press, 1996.
- [14] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley and C. Whitley, "A Comparison of Genetic Sequencing Operators", *Proc. 4th ICGA*, Morgan Kaufmann, pp.69-76, 1991.
- [15] T. Murata, H. Ishibuchi and H. Tanaka, "Flowshop Scheduling by Genetic Algorithm and Its Application to Multi-Objective Problems", *Trans. Society of Instrument and Control Engineers*, Vol.31, No.5, pp.583-590, 1995 (in Japanese).
- [16] H. Esbenson, "Defining solution set quality", *Memorandum*, No.UCB/ERL M96/1, Electric Research Lab., College of Eng., Univ. of California, Berkeley, 1996.