Multi-Objective Optimization by Genetic Algorithms: A Review

Hisashi Tamaki
Department of Electrical and Electronics Engineering,
Kobe University,
Rokkodai, Nada-ku,
Kobe 657, Japan.
tamaki@eedept.kobe-u.ac.jp

Hajime Kita
Department of Electrical Engineering,
Kyoto University,
Yoshida-honmachi, Sakyo-ku,
Kyoto 606-01, Japan.
kita@kuee.kyoto-u.ac.jp

Shigenobu Kobayashi
Department of Intelligent Science,
Graduate School of Interdisciplinary Science and Technology,
Tokyo Institute of Technology,
4259, Nagatsuta, Midori-ku,
Yokohama 226, Japan.
kobayasi@int.titech.ac.jp

Abstract— **This paper reviews several genetic algorithm (GA) approaches to multi-objective optimization problems (MOPs). The keynote point of GAs to MOPs is designing efficient selection/reproduction operators so that a variety of Pareto-optimal solutions are generated. From this viewpoint, the present paper reviews several devices proposed for multi-objective optimization by GAs such as the parallel selection method, the Pareto-based ranking, and the fitness sharing. Characteristics of these approaches have been confirmed through computational experiments with a simple example. Moreover, two practical applications of the GA approaches to MOPs are introduced briefly.**

I. Introduction

Most practical problems require the simultaneous optimization of multiple, often competing, objectives (or criteria). In applications of optimization techniques, the solution to such problems is usually computed by combining the objectives into a single one according to some utility function. In many cases, however, the utility function is not well known prior to the optimization process, and therefore, the whole problem should then be treated as a *multi-objective optimization problem* (MOP) with noncommensurable objectives. In the MOP, it is required to find a number of solutions in order to provide the decision maker with insight into the characteristics of the problem before a final solution is chosen.

The MOP seeks to the point $x = (x_1, \cdots, x_n)$ which minimizes (or, in some cases, maximizes) the values of a set of objective functions $f = (f_1, \cdots, f_p)$ within the feasible region $\mathcal{F}$ of x. Unlike single-objective optimization problems, the solution to this problem may not exist because of tradeoff characteristics among the objectives. Hence a concept of the *Pareto-optimal set*, a family of points which is optimal in the sense that no improvement can be achieved in any objective without degradation in others, is introduced as solutions to the MOP.

Definition: Let $x^0, x^1, x^2 \in \mathcal{F}$.

1. x^1 is said to be *dominated* by (or *inferior* to) x^2, if $f(x^1)$ is partially less than $f(x^2)$, i.e., $f_i(x^1) \geq f_i(x^2)$, $^\forall i = 1, \cdots, p$, and $f_i(x^1) > f_i(x^2)$, $^\exists i = 1, \cdots, p$.
2. x^0 is the *Pareto-optimal* (or *non-dominated*), if there doesn't exist any $x \in \mathcal{F}$ such that x dominates x^0.

In Fig. 1, the Pareto-optimal solutions to a two-objective problem are illustrated.

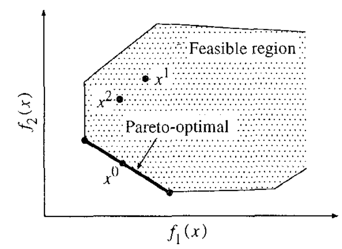


Fig. 1. Feasible region of function space and the Pareto-optimal solutions.

Since a Pareto-optimal solution is a rational solution to the MOP, the first goal of solving the MOP is to obtain a Pareto-optimal set (or, to sample solutions from the set as uniformly as possible). The Pareto-optimal solutions can be obtained by solving appropriately formulated single objective optimization problems, on a one-at-a-time basis. So far, several methods such as the weighted sum method and the ε-constraint method have been proposed [1].

Genetic algorithms (GAs) are methods for approximate optimization simulating the process of natural evolution [2; 3], and they have been successfully applied to several optimization problems which are difficult to solve exactly by conventional methods of the mathematical programming. The search processes of GAs using a population of solutions suggest their application to MOPs, i.e., to search a number of Pareto-optimal solution by GA in parallel.

II. Multi-Objective Optimization by Genetic Algorithms

Most simply, of course, GAs can be applied to MOPs according to the conventional approaches. That is, a MOP is transformed into a single objective optimization problems based on some knowledge of the problem, and then GAs are applied (repetitively, if necessary). This approach, however, can not reveal the attractive property of GAs.

As for population-based methods by GAs to generate a Pareto-optimal set simultaneously, several approaches have been proposed so far [3; 4; 5; 6; 7; 8; 9; 16; 10], and Fonseca and Fleming [11] have published an excellent survey, which also includes some fruitful discussions. In the following, laying weight on selection/reproduction operations,

0-7803-2902-3/96/$4.00

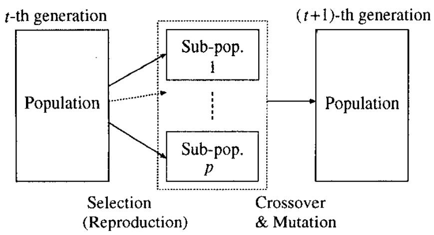
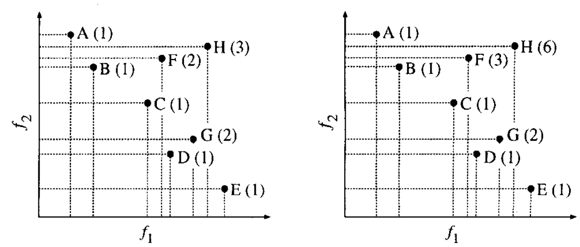


Fig. 2. Outline of generation replacement in VEGA.



(a) Goldberg's ranking (b) Fonseca and Fleming's ranking

Fig. 3. Pareto-based ranking. The rank of each individual is shown in the parensis.

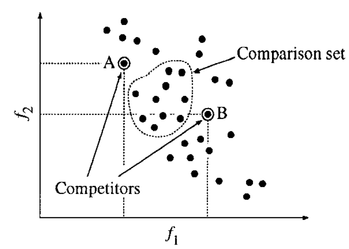


Fig. 4. Pareto-based tournament selection. In this case, the individual A is not dominated by the comparison set, while B is dominated by some of the individuals in it. Hence, the individual A is selected for survival.

population-based methods are reviewed.

A. Population-Based but Non-Pareto Approaches

First, we make a review of the approaches in which selection/reproduction is performed by treating the non-commensurable objective functions separately.

1. Schaffer [4] has developed the VEGA (Vector Evaluated Genetic Algorithm), in which sub-populations of the next generation are reproduced from the current population according to each of the objectives, separately (we say a *parallel selection* method). In Fig. 2, the outline of this approach is schematized.
2. Fourman [5] has proposed a method where selection is performed by comparing pairs of individuals with respect to one of the objectives. An objective is selected either according to the pre-assigned priorities of the objectives or randomly in each comparison.
3. Kursawe [6] has designed a multi-objective evolution strategies, in which a population of the next generation are formed by repeating a selection operation using one objective selected randomly according to predetermined probabilities.

Schaffer's study is important as a pioneering work of multi-objective optimization by GAs. However, these approaches have difficulties that they tend to generate such solutions that one of the objective is extremely good but the other objectives are not so.

B. Pareto-Based Approaches

Next, there have been proposed several approaches where selection/reproduction is performed by referring not only to the objective values themselves but also to the dominance property of them. We call such approaches the Pareto-based approaches.

1. Goldberg [3] has proposed a method of ranking and fitness assignment based on the Pareto-optimality of an individual. This method assigns the rank 1 to the non-dominated individuals in a population and remove them, and then assigns the rank 2 to the non-dominated individuals in the modified population, and so on.
2. Fonseca and Fleming [7] have also proposed another Pareto-based ranking method. In their method, the rank of each individual is defined by one plus the number of individuals in the current population that dominate it.

   In Fig. 3, examples of the above 2 sorts of ranking methods are shown.
3. Horn et al. [8] have proposed tournament selection based on dominance properties. In each tournament, two individuals compete using a set of other individuals in the population (called a *comparison set*). If one competitor (say "A") dominates all individuals in a comparison set and the other is dominated by at least one individual in the set, then the competitor A is selected for survival (Fig. 4). If both competitors are either dominated or non-dominated, a fitness sharing technique is adopted.
4. Tamaki et al. [10; 12] have proposed *the Pareto reservation strategy* on the analogy to the *elitism* for the case of treating scalar objective function. In their method, non-dominated individuals in a population at each generation are all reserved in the next generation. If the number of non-dominated individuals are less than the population size, the rest of the population in the next generation are filled by adopting the parallel selection method (of the VEGA). Oppositely, if the number of such individuals exceeds the population size, individuals in the next generation are selected among the non-dominated individuals by applying the parallel selection. In later version, they have adopted a fitness sharing technique in selecting individuals in the next generation among the non-dominated individuals.
5. Tanaka, Yamamura and Kobayashi [13] have proposed the Pareto-optimal selection method, in which all the non-dominated individuals in a population are kept and dominated individuals are discarded. To make the search process efficient, duplication of individuals are prohibited. They have used the following generation

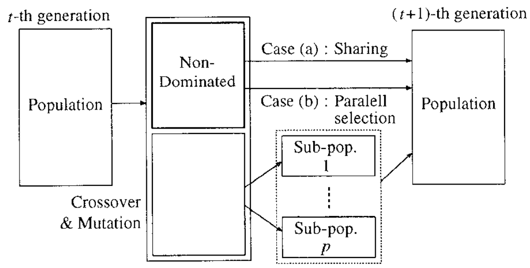


Fig. 5. Outline of the way of applying Pareto reservation strategy: (a) The case that the number of non-dominated individuals are more than population size, and (b) otherwise.

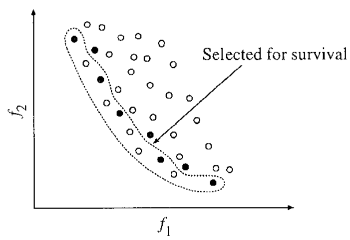


Fig. 6. Outline of Pareto-optimal selection. Black and white circles indicate the parent individuals and the individuals produced by recombination and mutation operations, respectively.

replacement. That is, before adopting the selection operator, quite many individuals are produced by applying recombination and mutation operators. Then, all the non-dominated individuals are selected (Fig. 6). Thus, the population size varies over generations. In some cases, they have applied a slightly modified selection method, where not only the non-dominated individuals but also individuals with a favorable Pareto-based rank (say, less than or equal to five) are selected for survival.

C. Maintenance of Diversity

In solving MOPs, it is required that the solutions are Pareto-optimal, and at the same time they are uniformly sampled from the Pareto-optimal set. The Pareto-based approaches mentioned above achieve the first requirement. However, only by themselves, the second requirement is not met. In most GAs, it is known that the genetic diversity of the population is lost due to their stochastic selection processes even if there is no fitness difference among the individuals. This phenomenon is called *the random genetic drift* [3]. In multi-objective optimization by GAs, it is needed to avoid losing the genetic diversity due to the random genetic drift.

For maintenance of the diversity, fitness sharing methods have been additionally used [3; 7; 8; 9]. In the fitness sharing methods, the fitness value of each individuals is reduced if there exist other individuals in its neighborhood, and therefore an individual located in more crowded area leaves less offsprings [3]. Thus, we can obtain the population distributed more uniformly over the Pareto-optimal set.

As for similarity measures (i.e., distance) between individuals, Fonseca et al. [7] and Horn et al. [8] have used a distance in the objective domain (i.e., the function space). Srinivas et al. [9] perform fitness sharing in the decision variable domain (i.e., the solution space).

In practical applications, it often occurs that genetic representation (genotype) of an individual doesn't correspond directly to the decision variables of a problem. Moreover, it may occur that the genotype contains some redundancy. In these cases, results by using a fitness sharing in the genotypic domain may differ form those by using one in the phenotypic domain.

Besides the fitness sharing techniques, some ways of keeping the genetic diversity in multi-objective optimization have been proposed.

1. Kita et al.[15] have combined the thermodynamical genetic algorithm (TDGA)[14] with the Pareto-based ranking method to maintain the diversity. The TDGA is a genetic algorithm inspired by the principle of the minimal free energy known as a behavior of the stochastic process used in the simulated annealing. In TDGA, the population in the new generation is selected so as to minimize the free energy $F = \langle E \rangle - HT$ of the population, where $\langle E \rangle$ and H are the mean energy and the entropy of the population, respectively, and T is the temperature. Diversity of the population is controlled by adjusting the parameter T. To compare with the fitness sharing method, this approach has an advantage that the suitable parameter T is less sensitive to the population size and to the size of the feasible region.
2. Osyczka and Kundu [16] have also proposed another way of maintaining diversity. That is, a fitness of an newly generated individual is assigned based on a relative distance in the objective domain from the non-dominated individual in the current population. Using this way of fitness calculation, they have designed their own method of generation replacement.

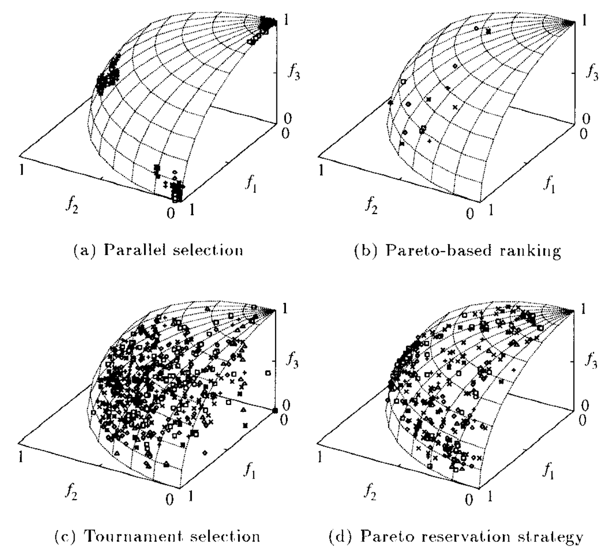
D. Examples

Now, some results of computational experiments are shown, where a simple function optimization problem with 3 objectives:

$$\text{maximize} \quad f_1 = x_1 \tag{1}$$
$$f_2 = x_2 \tag{2}$$
$$f_3 = x_3 \tag{3}$$
$$\text{sub. to} \quad {x_1}^2 + {x_2}^2 + {x_3}^2 \leq 1 \tag{4}$$
$$x_1, x_2, x_3, \geq 0 \tag{5}$$

has been solved by applying the following 4 methods:

(a) Parallel selection: applying the roulette wheel selection in generating each sub-population.
(b) Pareto-based ranking: assigning fitness values linearly according to the Fonseca and Fleming's ranking and applying the roulette wheel selection.
(c) Tournament selection with fitness sharing: combining a fitness sharing in the decision variable domain, and



(a) Parallel selection (b) Pareto-based ranking

(c) Tournament selection (d) Pareto reservation strategy

Fig. 7. Results obtained to the simple function optimization problem with 3 objectives.

setting the size of comparison set and a radius of the sharing to 30 and 1, respectively.

(d) Pareto reservation strategy: combining the parallel selection (as in (a)) and the fitness sharing (as in (c)).

Each variable is represented as an 8-bit string by adopting the *binary* coding technique, and the Hamming distance in the genotypic domain is used in a fitness sharing. Individuals that violate the constraints are treated as lethal ones. 10 trials have been done changing seeds for pseudo-random numbers with the setting of parameters of GAs as follows:

generations:	100
population size:	100
crossover:	1 point
crossover rate:	0.8
mutation rate:	0.01 / individual

In Fig. 7, the individuals in the final generation are shown in the space of the objective values with the Pareto-optimal surface, where the results of the 10 trials are indicated by using the different symbols. The results lead to the following remarks:

1. The parallel selection method (Fig. 7 (a)) can find only such solutions that one of the objective is nearly optimal but the other two objectives remain rather poor, as mentioned in Section II.A.
2. The Pareto-based ranking method (Fig. 7 (b)) can find several nearly Pareto-optimal solutions. The population, however, has converged to small number of solutions, as described in Section II.B.
3. Results obtained by using the methods which combines the Pareto-based selection/reproduction technique with the fitness sharing technique (Fig. 7 (c) and (d)) can find a variety of the Pareto-optimal (or nearly Pareto-optimal) solutions. That is, maintenance of diversity by the fitness sharing works effectively.
4. Furthermore, as compared with the tournament selection method with fitness sharing (Fig. 7 (c)), the parallel selection method with the Pareto reservation strategy (Fig. 7 (d)) can find such solutions that are closer to the Pareto-optimal surface. The progress of a search by using the tournament selection tends to be slow.

To summarize the above, we can say that the most successful approach to MOPs by GAs seems to be a combination of the Pareto-based selection/reproduction and techniques for maintaining diversity such as the fitness sharing, which is also mentioned in [11].

III. Case Studies

In this section, two sorts of applications of GAs to practical MOPs are introduced briefly.

A. Scheduling Problem in Hot Rolling Process

The GA approach is applied to a *hot rolling process* of a steel making factory. The process contains three furnaces and one rolling mill. In the process, several slabs are supplied from the upper (or preceding) process. Each slab is heated up to the prescribed temperature in one of the furnaces, and then is pressed by the rolling mill. The scheduling problem is defined as to find

1. the assignment of slabs to the furnaces, and
2. the pressing order of slabs in the rolling mill,

such that

3. a complexity index determined by the difference of width, thickness and hardness between a pair of slabs pressed subsequently,
4. a total quantity of fuel required for heating all slabs at furnaces,
5. time required for heating all slabs,

are to be minimized under the following constraints: (a) the absolute position of the order in pressing for each slab, (b) the relative order in pressing for each pair of slabs, and (c) the limitation of quantity of fuel supply per unit time.

In [10], the scheduling problem is divided into two problems: the problem of determining the pressing order (called "ordering problem") and the problem of assigning slabs to the furnaces (called "assignment problem"). Then, each problem is transformed to a *mathematical programming problem*.

Based on the mathematical programming formulation, an individual is represented as a combination of two sub-strings (α and β) with the length of the number of slabs, and each sub-string is formed as follows: α_i, the i-th locus of α, represents the index of the furnace to which the i-th slab is assigned, and β_i, the i-th locus of β represents the index of the slab that is pressed in the i-th order.

Then the following three kinds of genetic operators are implemented.

1. Crossover: two individuals are paired randomly in a population, and the crossing sites are selected randomly within the sub-string α. Then, with a prescribed probability (p_c), genes in the crossing sites are

Table 1. Results of Computational Experiments

Example	Objective values (mean (min.– max.))		
	Complexity	Fuel	Time
1	44.4 (38.4 – 52.6)	90.2 (86.7 – 91.7)	95.5 (91.1 – 102.0)
2	85.5 (66.8 – 112.7)	70.2 (66.8 – 85.6)	60.4 (56.9 – 64.4)
3	93.8 (79.3 – 118.0)	90.3 (86.4 – 94.6)	66.6 (61.1 – 72.7)

exchanged.

2. Mutation: within the sub-string α, the gene of each locus is changed to a randomly chosen allele, i.e., an index of the furnace, with a prescribed probability ($p_{\rm m}^{\alpha}$). Within the sub-string β, the order of two slabs selected randomly are exchanged with a prescribed probability ($p_{\rm m}^{\beta}$), if the resultant order satisfies both the absolute and the relative constraints.
3. Selection/reproduction: the parallel selection with the Pareto reservation strategy (as described in Section II.B) is used.

Moreover, to improve the search for the optimal schedule by GAs, a *local search method* (*a hill-climbing method*) is combined. In [10], the local search method is applied to the pressing order and the furnace assignment independently. The neighborhood of a schedule for the local search is defined as follows.

1. Neighborhood of the pressing order is defined as the set of schedules whose pressing order is different in two positions, i.e., one pair of slabs.
2. Neighborhood of the furnace assignment is defined as the set of schedules whose assignment is different in one slab.

When a better schedule is obtained by the local search method, the original genotype is replaced by the result of the local search.

Three kinds of scheduling problems of practical size, where the numbers of slabs are 147 (example 1), 103 (example 2) and 158 (example 3), respectively, are solved with the setting of the parameters as follows:

generations:	400
population size:	120
crossover sites:	10
crossover rate $p_{\rm c}$:	0.8
mutation rate $p_{\rm m}^{\alpha}$:	0.001 / gene
mutation rate $p_{\rm m}^{\beta}$:	0.3 / individual

To each problem, 10 experiments have been executed by changing the initial population, and the local search method has been applied at every 100 generations.

The results are summarized in Table 1, where the performance indexes of the obtained schedules are shown by their ratios to those of the schedule operated actually in percentage. From Table 1, we can observe that all objectives have been improved almost uniformly.

B. Decision Tree Induction Problem

A decision tree induction problem can be formulated as multi-objective optimization of trees trading the accuracy for the simplicity. In [17], a GA based approach has been proposed, which can provide us with a set of Pareto-optimal decision trees directly, without any pre- or post-processings.

An individual is coded by a decision tree itself like S-expression of LISP, and new individuals are produced by crossover and mutation in the following ways.

1. Crossover: the subtree exchange crossover [18] has been used, where $N_{\rm c}$ pairs of parent individuals are sampled randomly from a population with replacement, and then generate $2N_{\rm c}$ individuals.
2. Mutation: to aim at introducing some diversity into the population and to help the crossover for exploring a new search space, mutation alters one node into a randomly generated subtree or leaf-node with a prescribed probability ($p_{\rm m}$).

Then, the following model of generation replacement are adopted.

3. selection: the Pareto-optimal selection (as described in Section II.B) is applied to the set of individuals ($\approx 2N_{\rm c}(1+p_{\rm m})$).

As for the fitness function, the accuracy and the simplicity of a decision tree are represented by the error rate of classification and the number of leaves of the tree, respectively.

To show the effectiveness of the above method, several experiments have been done. For the comparison with the proposed approach, in [17], ID3 and OPT [19] have been chosen as an inducing algorithm and a pruning method, respectively. Moreover, the results of a great variety of pruning methods in [20] are shown for references. As for the feature subset selection (FSS) [21], they have tried to apply all the power set of given attributes because the technique seems not to be well grown.

For DIGIT [20] that is well known as a benchmark problem in the decision tree induction domain, ID3 induces a complete tree, which has 32 leaf-nodes, and whose error rate is 0.193. Note that some leaf nodes are labeled by the most frequent class there because noisy data often does not lead any leaves into a single class. Exhaustive search can not produce only trees with less than 16 leaves.

Fig. 8 shows a comparison between the results by using the GA, in which the setting of parameters is as follows:

generations:	3000
crossover number:	10
mutation rate $p_{\rm m}$:	0.1 / individual

and those by the conventional approach, which is the combination of ID3 and several pruning methods in [20]. The non-dominated trees among the population in the 3,000th of the GA are denoted by '$\times$'. The trees generated by the conventional approaches are denoted by '□' and '■', all of which are entirely dominated by the trees obtained by the GA.

Fig. 9 compares the results by the GA ('$\times$') with those by the combination of FSS, ID3 and OPT ('□' and '+'). All the solutions by those conventional approaches merely succeed in finding Pareto-optimal trees with less than 7 leaves. The above experiments show that the GA is a reasonable method to generate Pareto-optimal decision trees.

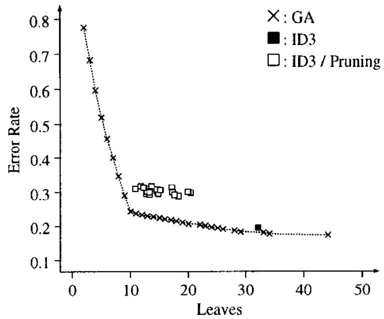


Fig. 8. Comparison between GA and ID3/Pruning [20] in the decision tree induction problem.

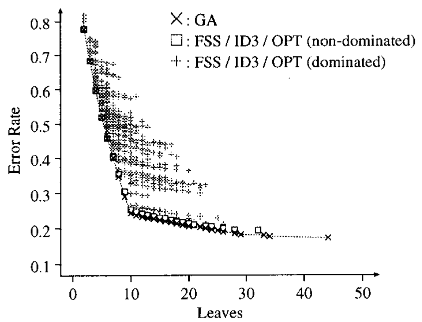


Fig. 9. Comparison between GA and FSS/ID3/OPT.

For DIGIT, the exhaustive search spent more than 1 month with Sun SparcStation20 to find out all the Pareto-optimal trees from 2 to 16 leaves. It found no Pareto-optimal trees more than 17 leaves. On the other hand, the GA-based method spent only about 30 hours to find out all the Pareto-optimal trees from 2 to 21 leaves with the same machine.

IV. Conclusion

In this paper, several genetic algorithm (GA) approaches to multi-objective optimization problems (MOPs) are reviewed, with respect mainly to selection/reproduction methods which are essential for generating a variety of Pareto-optimal solutions. Then, through computational experiments with an example of MOPs, typical characteristics of these approaches have been confirmed. Moreover, two sorts of applications of GAs to practical MOPs are introduced briefly, which show the effectiveness of the GA approaches.

In multi-objective optimization by GAs, however, it is also an important issue to make the recombination operation efficient. If the population is distributed widely near the Pareto-optimal set, recombination operation with random paring may not yield good search points. Hence, some devices are needed to make the search by recombination more efficient [11; 13].

Further, for the multi-objective decision making, it is needed to support the decision maker by providing him or her with insight into the characteristics of the problem, and by estimating his or her preference. To obtain the Pareto-optimal set is the first step of this goal. Processes like interactive evolution [22; 23] may be useful tools to achieve the multi-objective decision making.

Finally, this research was supported by the Grant-in-Aid for Scientific Research on the Priority Area "System Theory of Function Emergence" from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd, *Optimization,* Chap. 10, North-Holland, 1989
[2] J. H. Holland, *Adaptation in Natural and Artificial Systems,* Univ. of Michigan, 1975
[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, 1989
[4] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proc. of the 1st ICGA,* 93–100, 1985
[5] M. P. Fourman, "Compaction of symbolic layout using genetic algorithms," *Proc. of the 1st ICGA,* 141–153, 1985
[6] F. Kursawe, "A variant of evolution strategies for vector optimization," *Parallel Problem Solving from Nature,* 193–197, Springer, 1991
[7] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization : formulation, discussion and generalization," *Proc. of the 5th ICGA,* 416–423, 1993
[8] J. Horn, N. Nafpliotis and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *Proc. of the 1st ICEC,* 82–87, 1994
[9] N. Srinivas and K. Deb, "Multiobjective optimization using non-dominated sorting in genetic algorithms," *Evolutionary Computation,* **2** (3), 221–248, 1994
[10] H. Tamaki, M. Mori, M. Araki, Y. Mishima and H. Ogai, "Multi-criteria optimization by genetic algorithms : a case of scheduling in hot rolling process, *Proc. of the 3rd APORS,* 374–381, 1995
[11] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation,* **3** (1), 1–16, 1995
[12] H. Tamaki, M. Mori and M. Araki, "Generation of a set of Pareto-optimal solutions by genetic algorithms," *Trans. of the Society of Instrument and Control Engineers,* **31** (8), 1185–1192, 1995 (in Japanese)
[13] Y. Takada, M. Yamamura and S. Kobayashi, "An approach to portfolio selection problems using multi-objective genetic algorithms," *Proc. of the 23rd Symposium on Intelligent Systems,* 103–108, 1996 (in Japanese)
[14] N. Mori, J. Yoshida, H. Tamaki, H. Kita and Y. Nishikawa, "A thermodynamical selection rule for the genetic algorithm," *Proc. of the IEEE 2nd Int. Conf. on Evolutionary Computation,* 188–192, 1995
[15] H. Kita, Y. Yabumoto, N. Mori and Y. Nishikawa, "Multi-objective optimization by means of the thermodynamical genetic algorithm," submitted to *PPSN 4,* 1996
[16] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural Optimization,* **10** (2), 94–99, 1995
[17] K. Yoshida, M. Yamamura and S. Kobayashi, "Generating a set of Pareto optimal decision trees by GA," to be appeared in *Proc. of the 4th Int. Conf. on Soft Computing,* 1996
[18] J. R. Koza, "Concept formation and decision tree induction using genetic programming paradigm," *Parallel Problem Solving from Nature,* 124–128, Springer, 1991
[19] M. Bohanec and I. Bratko, "Trading accuracy for simplicity in decision trees," *Machine Learning,* **15**, 223–250, 1994
[20] J. Minger, "An empirical comparison of pruning methods for decision tree induction," *Machine Learing,* **4**, 227–243, 1989
[21] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," *Proc. of the 9th Nat. Conf. on Artificial Intelligence,* 547–552, 1991
[22] R. Dawkins, *The Blind Watchmaker,* Norton, 1986
[23] K. Sims, "Artificial evolution for computer graphics," *Computer Graphics,* **25** (4), 319–328, 1991