Genetic Algorithm based on a Pareto Neighborhood Search for Multiobjective Optimization

Takanori Tagami† and Tohru Kawabe‡

†*Dept. of General Education, Anan College of Technology*
Tokushima 774-0017 Japan, tagami@anan-nct.ac.jp
‡*Institute of Information Sciences and Electronics, University of Tsukuba*
Ibaraki 305-8573 Japan, kawabe@is.tsukuba.ac.jp

Abstract— In this paper, we examine the performance of a genetic algorithm based on a Pareto neighborhood search for multiobjective optimization. The purpose of the proposed method is to generate a set of non-dominated solutions that is properly distributed in the neighborhood of the trade-off surface. Simulation results show that the GA based on the proposed method has good performances better than the traditional GA approaches for several multiobjective flowshop scheduling problems.

I. INTRODUCTION

In practical problems, we often want to optimize more than one measure of performance at once. Multiobjective optimization seeks to optimize the components of a vector-valued cost function. Unlike single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set, also called non-dominated set. Each point in the trade-off surface is optimal in the sense that no improvement can be achieved in one cost vector component that does not lead to degradation in at least one of the remaining components. Conventional optimization techniques, such as gradient-based method, simplex-based method and simulated annealing, are difficult to extend to the true multiobjective optimization case, because they were not designed with multiple solutions in mind.

By using evolutionary computation, parallel optimization techniques such as Genetic algorithms (GAs)[1,2] have been recognized to be well suited to multiobjective optimization since early in their development. Multiple individuals can search for multiple solutions in parallel, eventually taking advantage of any similarities available in the family of possible solutions to the problem. Extensions of GAs to multiobjective optimization were proposed in several manners [3-6]. Schaffer [3] proposed an extension of the simple GA to accommodate vector-valued fitness measures, which he called the Vector Evaluated Genetic Algorithm (VEGA). In VEGA, appropriate fractions of the next generation, or sub-populations, were selected from the whole of the old generation according to each of the objective values, separately. However, in the case of a concave trade-off surface, the population will tend to split into different species, each of them particularly strong in one of the objectives. This phenomenon, known as speciation, is undesirable in that it is opposed to the aim of finding a compromise solution.

While a Pareto-based approach was first proposed by Goldberg [1], as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. Fonseca and Fleming [4] proposed a multiobjective ranking based on the Pareto-based fitness assignment. In this method, an individual's rank corresponds to the number of individuals in the current population by which it is dominated. Multiobjective ranking correctly assigns all non-dominated individuals the same fitness, but that, on its own, does not guarantee that the Pareto-optimal set be uniformly sampled. When presented with multiple equivalent solutions, finite population member tend to converge to only one of them, due to stochastic errors in the selection process. This phenomenon, known as genetic drift, has been observed in natural as well as in artificial evolution, and can also occur in the Pareto- based evolutionary optimization.

In this paper, a genetic algorithm based on a Pareto neighborhood search method to multiobjective optimization problems is proposed. The purpose of the proposed method is to generate a Pareto-optimal set that is properly distributed in the trade-off surface. The genetic search using the proposed method properly adjusts the convergence of solutions in the objective space. Experimental results for several 2-objective flowshop scheduling problems indicate that the standard GA with the multiobjective ranking leads to the partial convergence of the solutions. On the other hand, the GA with the proposed method are effective in directing the genetic search in the objective space.

II. PROPOSED ALGORITHM

A. Pareto-Based Approach

Pareto-based ranking is an alternative to avoid using the objective vectors corresponding to the solutions directly to perform the selection process. Instead, they are projected in the objective space and the Pareto dominance principle is applied to rank the solutions. The fitness of each solution will eventually be derived from this Pareto-based ranking. However, as we said earlier, this process is prone to lead to a convergence on a single point. To avoid this specific problem, we introduce a Pareto neighborhood search method.

B. Pareto Neighborhood Search Method

To prevent a partial convergence of non-dominated solutions in the trade-off surface, we propose a Pareto neighborhood search method. The proposed method assigns all individuals the differ fitness on the basis of the distribution in objective space.

The proposed method consists of following procedure. First, the objective space is divided pre-specified regions. The edge points of the whole region correspond the best solutions and the worst solutions for each objective function. Then, the fitness f_i of the individual p_i is defined as $f_i = 1/n_i$. The value of n_i denotes the number of individuals in the identical region with the individual p_i. An example of the Pareto neighborhood search method in 2-objective minimizing problem is shown in Figure 1.

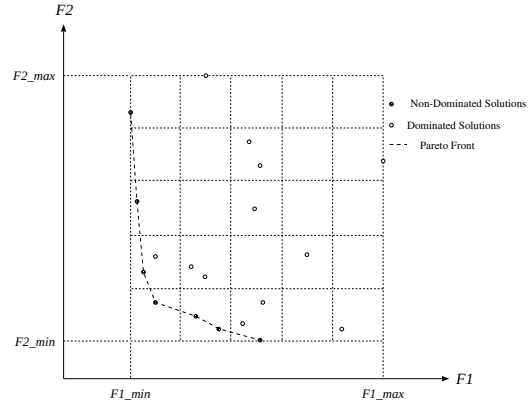


Figure 1: Pareto neighborhood search method

C. Proposed Procedure

In this paper, we employ a continuous generation model with a Pareto-optimal preservation strategy. In this model, a Pareto-optimal solution is always forced to appear in the following generation. The proposed procedure consists of the following steps:

Step 1. Set a generation number $t = 0$. Randomly generate an initial population $P(t)$ of M individuals.

Step 2. Calculate the fitness of each individuals in the current population according to the evaluation function.

Step 3. Select M individuals according to the fitness, then the pairs of crossover are chosen randomly in the neighboring regions.

Step 4. Generate a new population $P'(t)$ from $P(t)$ by using a crossover operator.

Step 5. Apply a mutation operator to the newly generated population $P'(t)$.

Step 6. Calculate the fitness both of $P(t)$ and $P'(t)$ using the Pareto neighborhood search method.

Step 7. Select M individuals from all population member on the basis of the fitness.

Step 8. If a terminal condition is satisfied, stop and return the best individuals. Otherwise set $t = t + 1$ and go to **Step 2**.

In this procedure, update of the current population size is always constant M. There is no question that selection of individuals for the next generation should be proportional to their fitness values. However, when a pure proportional approach is adopted, in practice the population is rapidly dominated by a few super-individuals, resulting in a dramatic decrease in genetic diversity. Here, to avoid the rapid loss of genetic diversity, multiple equivalent individuals are eliminated from the current population.

III. COMPUTER SIMULATIONS

A. Flowshop Scheduling Problem

Flowshop scheduling is one of the most well-known scheduling problems in the field of scheduling. The flowshop scheduling problem is generally described as follows: There are m machines and n jobs, each job consists of m operations, and each operation requires a different machine. n jobs have to be processed in the same sequence on m machines.

In this paper, we use two criteria (makespan and maximum lateness) in order to allow visualization. We specify two objective minimizing problem as follows:

$$\min \quad F(x) \;=\; (F_1(x), F_2(x)) \tag{1}$$

$$\begin{cases} F_1(x) \;=\; \max\limits_j C_j & \text{(makespan)} \\ F_2(x) \;=\; \max\limits_j L_j & \text{(maximum lateness)} \end{cases}$$

where x is a feasible solution of the flowshop scheduling problem. C_j denotes the completion time of job j and L_j denotes the lateness of job j.

In computer simulations, we generate 30 jobs, 40 jobs and 50 jobs scheduling problems. Thus, our scheduling problem is to determine the sequence of given jobs that are processed on the machines. The processing time of each job on each machine is specified as a random integer in the interval [1,10].

The duedate of each job are specified through three steps. First, randomly generate a sequence of the given jobs. Secondly, calculate the completion time of each job when the given jobs are processed in the job sequence. Finally, define the duedate of each job by

$$d_j = C_j + rand[-200, 200]$$

where d_j is the duedate of job j, and $rand[-200, 200]$ means a random integer in the interval $[-200, 200]$.

B. GA Formulation

To design a GA for multiobjective optimization problem, certain problem-dependent algorithm elements need to be defined.

(a) Representation Because the flowshop scheduling problem is essentially a permutation schedule problem, we can use the permutation of jobs as the representation scheme of individuals, which is the natural representation for the sequencing problem. For example, let the ith individual be

$$p_i = [\ 4\ 5\ 2\ 3\ 1\ ]$$

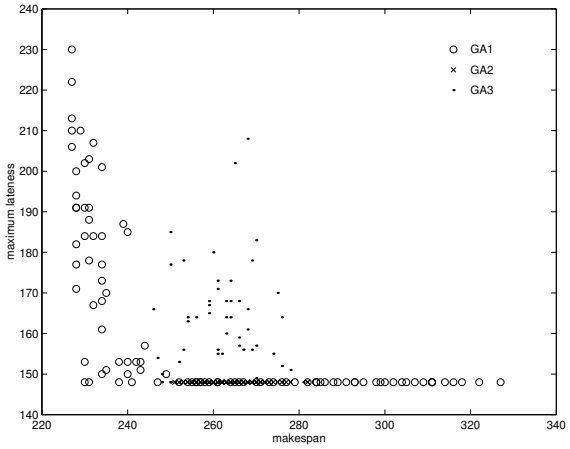
This means that the job sequence is j_4, j_5, j_2, j_3, j_1. In this way, each individual represents a possible scheduling configuration.

(b) Evaluation Function The linear scaling method base on the multiobjective ranking is employed as an evaluation function. In computer simulations, the values of maximum fitness and minimum fitness are held constants $(10.0, 1.0)$, namely, the fitness of best individuals is 10.0, on the contrary, the fitness of worst one is 1.0.

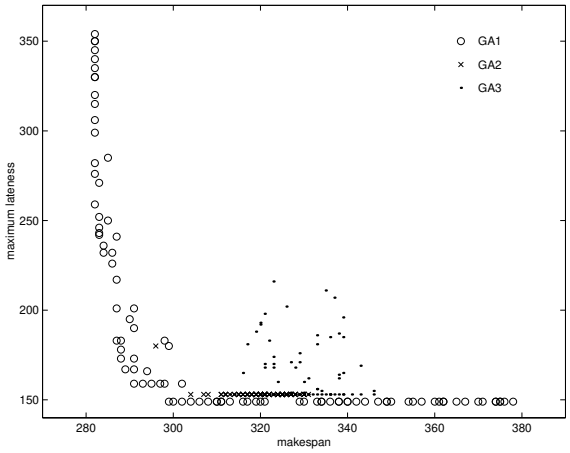
(c) Crossover and Mutation There are many well-known crossover operations available for permutation representation. Here, Partially Mapped Crossover(PMX) is used. Mutation operation is designed to perform random exchange; that is, it selects two genes randomly in an individual and exchanges their positions.

(d) Control Parameters The following parameter specifications are employed for all experiments.

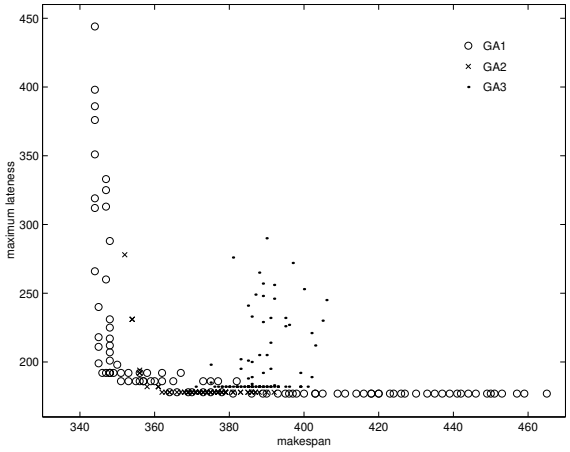
Population size:	100
Mutation rate:	0.1
Stopping condition	2000



(a) Results of 30 jobs scheduling problem

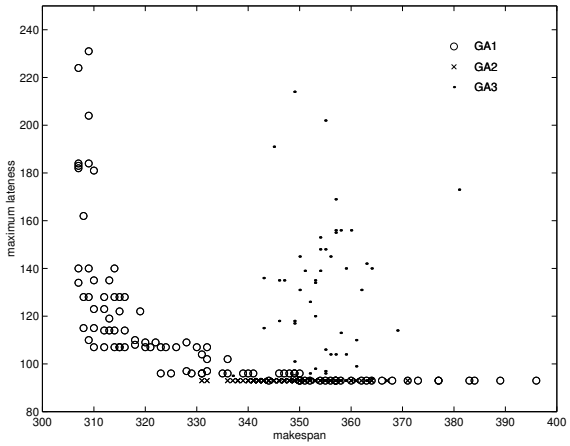


(b) Results of 40 jobs scheduling problem

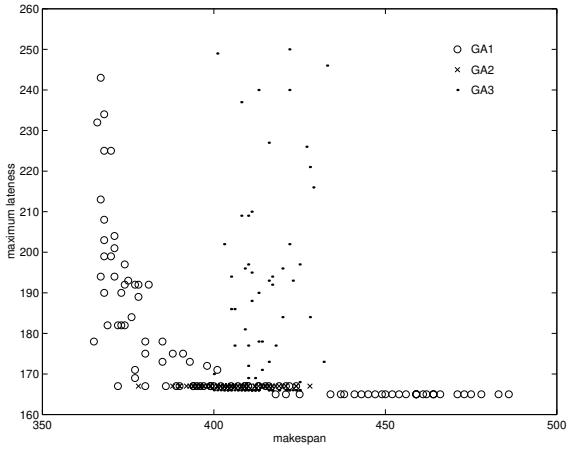


(c) Results of 50 jobs scheduling problem

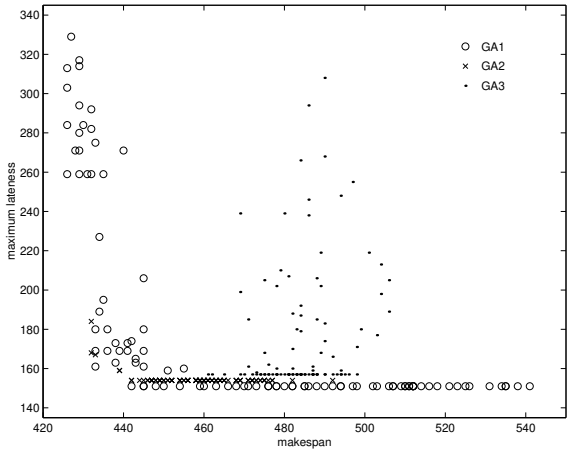
Figure 2: Distribution of solutions (10 machines)



(a) Results of 30 jobs scheduling problem



(b) Results of 40 jobs scheduling problem



(c) Results of 50 jobs scheduling problem

Figure 3: Distribution of solutions (20 machines)

C. Simulation Results

Three GA approaches are used of comparison. The GA with the Pareto neighborhood search method is called GA1. The standard GA with multiobjective ranking is called GA2. The standard GA with weighted-sum approach is called GA3. The genetic operators used in GA2 and GA3 are the same as those used in GA1. The simulation results are shown in Figure 2,3. In all figures, small marks indicate the solutions in the final population.

The following results are obtained:

- From the results of GA1, the solutions are widely distributed in the trade-off surface, and the search performance does not deteriorate significantly even in the all cases.
- The conventional Pareto-based approach, GA2 leads to the partial convergence of the solutions because of stochastic errors in the iterative process.
- It is obvious that the proposed method can seek for the extensively distributed solutions in comparison with the standard GA approaches.

IV. CONCLUSION

In this paper, we have proposed a genetic algorithm based on the Pareto neighborhood method in order to avoid the partial convergence of the solutions. The proposed Pareto neighborhood method properly adjusts a distribution of non-dominated solutions in the objective space. We can confirm from several numerical simulations that the proposed method is effective for the generation of a Pareto-optimal set that is widely distributed in the trade-off surface.

References

[1] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine learning", *Addison Wesley*, 1989.

[2] Z. Michalewics and C. Janikow, "Genetic Algorithms + Data Structures = Evolution Programs", *Springer Verlag*, 1992.

[3] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Proc. of the 1st ICGA*, pp.93-100, 1985.

[4] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", *Proc. of the 5th ICGA*, pp.416-423, 1993.

[5] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization", *Evolutionary Computation*, vol.3, no.1, pp.1-16, 1995.

[6] D. H. Loughlin and S. Ranjithan, "The Neighborhood Constraint Method: A Genetic Algorithm-Based Multiobjective Optimization Technique", *Proc. of the 7th ICGA*, pp.666-673, 1997.