PARTICLE SWARM OPTIMIZATION FOR INTERACTIVE FUZZY MULTIOBJECTIVE NONLINEAR PROGRAMMING

T. Matsui, M. Sakawa, K. Kato, T. Uno and K. Tamada

Received February 27, 2007; revised October 22, 2007

Abstract. In recent years, particle swarm optimization (PSO) proposed by Kennedy et al. has been widely used as a general approximate solution method for optimization problems. The authors proposed a revised PSO (rPSO) method incorporating the homomorphous mapping and the multiple stretching technique in order to cope with shortcomings of PSO and showed its efficiency for nonlinear programming problems. In this paper, we construct an interactive fuzzy satisficing method for multiobjective nonlinear programming problems based on the rPSO. In order to obtain better solutions in consideration of the property of multiobjective programming problems, we incorporate the direction to nondominated solutions into the rPSO. Furthermore, the efficiency of the proposed method (MOrPSO) is shown through applications to numerical examples.

1 Introduction A nonlinear programming problem is called a convex programming problem when its objective function and its constraint region are convex. For such convex programming problems, there have been proposed many efficient solution methods such as the successive quadratic programming method, the generalized reduced gradient method and so forth. Unfortunately, there have not been proposed any decisive solution method for nonconvex programming problems. As practical solution methods, meta-heuristic optimization methods such as the simulated annealing method, the genetic algorithm and so on, have been proposed. In recent years, however, more speedy and more accurate optimization methods have been desired because actual problems become more large-size and more complex.

As a new optimization method, the particle swarm optimization (PSO) method was proposed by Kennedy et al. [2]. PSO is a search method simulating the social behavior that individuals (particles) like bird or fish constitute a population (swarm) and each particle in the swarm search better points using the knowledge owned by it as well as that owned by the swarm. Since the original PSO has shortcomings such as the concentration of particles to local solutions and the inapplicability to constrained problems, the authors proposed a revised PSO (rPSO) by incorporating the homomorphous mapping and the multiple stretching technique in order to overcome these shortcomings [5].

In recent years, with the diversification of social requirements, the demand for the programs with multiple objective functions, has been increasing rather than a single-objective function (e.g. maximizing the total profit and minimizing the amount of pollution in a production planning). Since there does not always exist a complete optimal solution which optimizes all objectives simultaneously for multiobjective programming problems, a solution that any improvement of one objective function can be achieved only at the expense of at least one of other objective functions is considered as a reasonable solution, which is

2003 *Mathematics Subject Classification.* Primary 90C29, 90C70; Secondary 90C59, 90C26.

Key words and phrases. particle swarm optimization, multiobjective nonlinear programming problem, interactive fuzzy satisficing method, nondominated solution.

called a Pareto optimal solution or a nondominated solution. For such multiobjective optimization problems, fuzzy programming approaches (e.g. Zimmermann [12], Rommelfanger [7]), considering the imprecise nature of the decision maker's judgments in multiobjective optimization problems, seem to be very applicable and promising. In the application of the fuzzy set theory into multiobjective linear programming problems started by Zimmermann [12], it has been implicitly assumed that the fuzzy decision or the minimum-operator of Bellman and Zadeh [1] is suitable to represent the decision maker's preference for fuzzy goals of objective functions. In general, however, the fuzzy decision is not always the best representation of the decision maker's preference for fuzzy goals. Thereby, Sakawa et al. have proposed interactive fuzzy satisficing methods for various multiobjective programming problems to derive a satisficing solution for the decision maker along with checking the local preference of the decision maker through interactions [8, 9]. In particular, they [10] considered an interactive fuzzy satisficing method for multiobjective nonlinear programming problems. In [10], they adopted a genetic algorithm called RGENOCOPV and showed the feasibility of the proposed method, but there are drawbacks about calculation time and precision of solutions.

In this research, focusing on multiobjective nonlinear programming problems, we attempt to derive a satisficing solution through the interactive fuzzy satisficing method with short calculation time and high precision. As the solution method used in the interactive fuzzy satisficing method, we adopt rPSO [5] which is promising for nonlinear programming problems and we propose an extension of rPSO to multiobjective programming problems, MOrPSO.

2 Multiobjective nonlinear programming problems In this research, we consider multiobjective nonlinear programming problems formulated as:

$$\left.\begin{array}{rl} \text{minimize} & f_l(\boldsymbol{x}),\ l = 1, 2, \ldots, k \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0,\ i = 1, 2, \ldots, m \\ & l_j \leq x_j \leq u_j,\ j = 1, 2, \ldots, n \\ & \boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in \mathcal{R}^n \end{array}\right\} \tag{1}$$

where $f_l(\cdot)$, $g_i(\cdot)$ are linear or nonlinear functions, l_j and u_j are the lower limit and the upper limit of each decision variable x_j. In addition, we denote the feasible region of (1) by X.

3 An interactive fuzzy satisficing method In order to consider the imprecise nature of the decision maker's judgments for each objective function in (1), if we introduce the fuzzy goals such as "$f_l(\boldsymbol{x})$ should be substantially less than or equal to a certain value", (1) can be rewritten as:

$$\underset{\boldsymbol{x} \in X}{\text{maximize}}\ (\mu_1(f_1(\boldsymbol{x})), \ldots, \mu_k(f_k(\boldsymbol{x}))) \tag{2}$$

where $\mu_l(\cdot)$ is the membership function to quantify the fuzzy goal for the l th objective function in (1). To be more specific, if the decision maker feels that $f_l(\boldsymbol{x})$ should be less than or equal to at least f_l^0 and $f_l(\boldsymbol{x})) \leq f_l^1 (\leq f_l^0)$ is satisfactory, the shape of a typical membership function is shown in Fig. 1.

Since (2) is regarded as a fuzzy multiobjective decision making problem, there rarely exist a complete optimal solution that simultaneously optimizes all membership functions. As a reasonable solution concept for the fuzzy multiobjective decision making problem, Sakawa et al. defined M-Pareto optimality on the basis of membership function values by directly extending the Pareto optimality in the ordinary multiobjective programming problem [8].

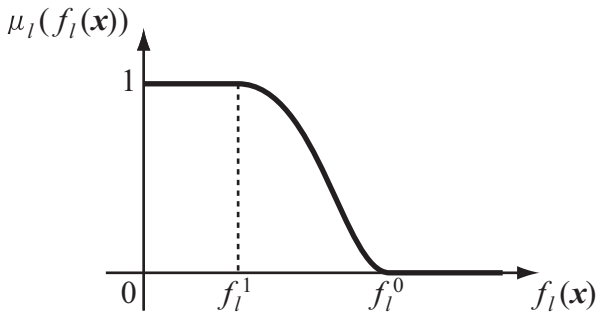

Figure 1: An example of membership functions.

Introducing an aggregation function $\mu_D(\boldsymbol{x})$ for k membership functions in (2), the problem can be rewritten as:

$$\underset{\boldsymbol{x}\in X}{\text{maximize}}\ \mu_D(\boldsymbol{x}) \tag{3}$$

where the aggregation function $\mu_D(\cdot)$ represents the degree of satisfaction or preference of the decision maker for the whole of k fuzzy goals. Following conventional fuzzy approaches, as aggregation functions, the minimum operator and the product operator are often adopted. However, it should be emphasized here that such approaches are preferable only when the decision maker feels that the minimum operator or the product operator is appropriate. In other words, in general decision situations, the decision maker does not always use the minimum operator or the product operator when combining the fuzzy goals. Probably the most crucial problem in (3) is the identification of an appropriate aggregation function which well represents the decision maker's fuzzy preferences. If $\mu_D(\cdot)$ can be explicitly identified, then (3) reduces to a standard mathematical programming problem. Unfortunately, however, this rarely happens. Thereby, as an alternative, an interaction with the decision maker is necessary for finding the satisficing solution of (2).

In the interactive fuzzy satisficing method, in order to generate a candidate for the satisficing solution which is M-Pareto optimal, the decision maker is asked to specify the aspiration levels of achievement for all membership functions, called the reference membership levels [8]. For the decision maker's reference membership levels $\bar{\mu}_l$, $l = 1, 2, \ldots, k$, the corresponding M-Pareto optimal solution, which is nearest to the requirements in the minimax sense or better than that if the reference membership levels are attainable, is obtained by solving the following augmented minimax problem (4).

$$\underset{\boldsymbol{x}\in X}{\text{minimize}}\quad \max_{l=1,\ldots,k}\{(\bar{\mu}_l - \mu_l(f_l(\boldsymbol{x}))) + \rho\sum_{i=1}^{k}(\bar{\mu}_i - \mu_i(f_i(\boldsymbol{x})))\} \tag{4}$$

where ρ is a sufficiently small positive real number.

We can now construct the interactive algorithm in order to derive the satisficing solution for the decision maker from the M-Pareto optimal solution set. The procedure of an interactive fuzzy satisficing method is summarized as follows.

Step 1: Under given constraints, the minimal value and the maximal one of each objective function are calculated by solving following problems.

$$\underset{\boldsymbol{x}\in X}{\text{minimize}}\ f_l(\boldsymbol{x}),\ \ l = 1, 2, \ldots, k \tag{5}$$

$$\underset{\boldsymbol{x}\in X}{\text{maximize}}\ f_l(\boldsymbol{x}),\ \ l = 1, 2, \ldots, k \tag{6}$$

Step 2: In consideration of the minimal value and the maximal one of each objective function, the decision maker subjectively specifies membership functions $\mu_l(f_l(\boldsymbol{x}))$, $l = 1, 2, \ldots, k$ to quantify fuzzy goals for objective functions. Next, the decision maker sets initial reference membership levels $\bar{\mu}_l$, $l = 1, 2, \ldots, k$.

Step 3: We solve the augmented minimax problem corresponding to current reference membership levels (4).

Step 4: If the decision maker is satisfied with the solution obtained in step 3, the interactive procedure is finished. Otherwise, the decision maker updates reference membership levels $\bar{\mu}_l$, $l = 1, 2, \ldots, k$ based on current membership function values and objective function values, and return to step 3.

4 Particle swarm optimization Particle swarm optimization (PSO) [2] is based on the social behavior that a population of individuals adapts to its environment by returning to promising regions that were previously discovered [3]. This adaptation to the environment is a stochastic process that depends on both the memory of each individual, called particle, and the knowledge gained by the population, called swarm.

In the numerical implementation of this simplified social model, each particle has four attributes: the search point in the search space, the search direction vector and the best search point in its track and the best search point of the swarm. The process can be outlined as follows.

Step 1: Generate the initial swarm involving N particles at random.

Step 2: Calculate the next search direction vector of each particle, based on its attributes.

Step 3: Calculate the next search point of each particle from the current search point and its next search direction vector.

Step 4: If the termination condition is satisfied, stop. Otherwise, go to Step 2.

To be more specific, the next search direction vector of the i-th particle at time t, $\boldsymbol{v}_i^{t+1}$, is calculated by the following scheme introduced by Shi and Eberhart [11].

$$\boldsymbol{v}_i^{t+1} := \omega^t \boldsymbol{v}_i^t + c_1 R_1^t (\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + c_2 R_2^t (\boldsymbol{p}_g^t - \boldsymbol{x}_i^t) \tag{7}$$

In (7), R_1^t and R_2^t are random numbers between 0 and 1, $\boldsymbol{p}_i^t$ is the best search point of the i-th particle in its track and $\boldsymbol{p}_g^t$ is the best search point of the swarm. There are three problem-dependent parameters, the inertia of the particle ω^t, and two trust parameters c_1, c_2.

Then, the next search point of the i-th particle at time t, $\boldsymbol{x}_i^{t+1}$, is calculated from (8).

$$\boldsymbol{x}_i^{t+1} := \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{8}$$

where $\boldsymbol{x}_i^t$ is the current search point of the i-th particle at time t. The i-th particle calculates the next search direction vector $\boldsymbol{v}_i^{t+1}$ by (7) in consideration of the current search direction vector $\boldsymbol{v}_i^t$, the direction vector from the current search point $\boldsymbol{x}_i^t$ to the best search point in its track $\boldsymbol{p}_i^t$ and the direction vector from the current search point $\boldsymbol{x}_i^t$ to the best search point of the swarm $\boldsymbol{p}_g^t$, moves from the current search point $\boldsymbol{x}_i^t$ to the next search point $\boldsymbol{x}_i^{t+1}$ calculated by (8). The parameter ω^t controls the amount of move to search globally in early stage and to search locally by decreasing ω^t gradually.

The movement of a particle in PSO is shown in Fig. 2. Comparing the evaluation value of a particle after move, $f(\boldsymbol{x}_i^{t+1})$, with that of the best search point in its track, $f(\boldsymbol{p}_i^t)$, if

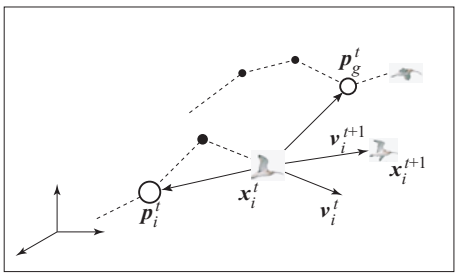

Figure 2: Movement of a particle in PSO.

$f(\boldsymbol{x}_i^{t+1})$ is better than $f(\boldsymbol{p}_i^t)$, then the best search point in its track is updated as $\boldsymbol{p}_i^t := \boldsymbol{x}_i^{t+1}$. Furthermore, if $f(\boldsymbol{p}_i^{t+1})$ is better than $f(\boldsymbol{p}_g^t)$, then the best search point of the swarm is updated as $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_i^{t+1}$.

Such a simple PSO [2, 11] includes two problems. One is that particles concentrate on the best search point of the swarm and they cannot easily escape from the point (local solution) since the search direction vector $\boldsymbol{v}_i^{t+1}$ calculated by (7) always includes the direction vector to the best search point of the swarm. Another is that a particle after move is not always feasible for problems with constraints.

5 Improvement of particle swarm optimization In this study, to prevent the concentration and stop of particles at local optimal solutions in the simple PSO, we introduce the modification of move schemes of a particle, the secession and the multiple stretching technique. In addition, in order to treat constraints, we divide the swarm into two subswarms. In one subswarm, since the move of a particle to the infeasible region are not accepted, if a particle becomes infeasible after move, it is repaired to be feasible. In the other subswarm, the move of a particle to the infeasible region are accepted.

5.1 Move of a particle Let us consider the move from the current search point $\boldsymbol{x}_i^t$ of the i-th particle.

First, if the previous search point $\boldsymbol{x}_i^{t-1}$ is the best search point of the particle in its track $\boldsymbol{p}_i^t$, the next search point $\boldsymbol{x}_i^{t+1}$ moves near the best search point of the swarm $\boldsymbol{p}_g^t$ with high possibility. Thereby, as shown in Fig. 3, we change (7) to determine the next search direction vector $\boldsymbol{v}_i^{t+1}$ as

$$\boldsymbol{v}_i^{t+1} := c_1 R_1^t(\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + c_2 R_2^t(\boldsymbol{p}_k^t - \boldsymbol{x}_i^t) \tag{9}$$

where $\boldsymbol{p}_k^t$ is the best search point of the k-th particle. By the change of the search direction vector determination scheme, we can relax concentration of particles to $\boldsymbol{p}_g^t$.

Next, in case that the current search point $\boldsymbol{x}_i^t$ is the best search point of the particle in its track $\boldsymbol{p}_i^t$, the direction from the previous search point to the current search point is desirable. Thus, as in Fig.4, we change (7) to determine the next search direction vector $\boldsymbol{v}_i^{t+1}$ as

$$\boldsymbol{v}_i^{t+1} := \omega^t \boldsymbol{v}_i^t \tag{10}$$

Otherwise, we use (7) to determine the next search direction vector $\boldsymbol{v}_i^{t+1}$.

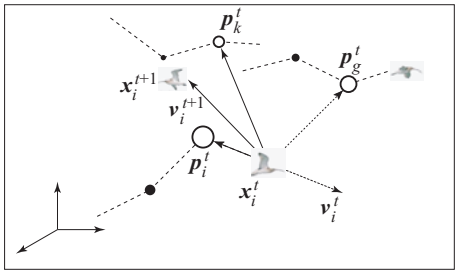

Figure 3: The new search direction vector when the best search point of a particle in its track is renewed at the previous search point.

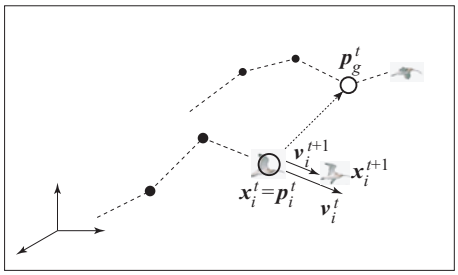

Figure 4: The new search direction vector when the best search point of a particle in its track is renewed at the current search point.

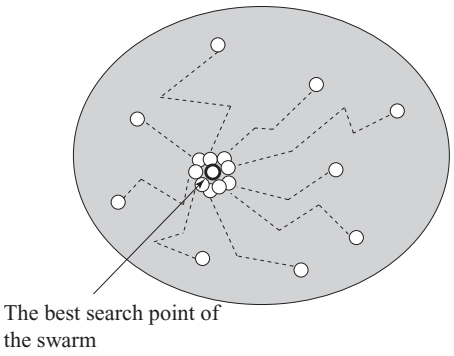

Figure 5: The concentration of particles around the best search point of the swarm.

5.2 Division of the swarm into two subswarms In application of PSO to optimization problems with constraints, a particle after move is not always feasible if we use the move schemes mentioned above. To deal with such a situation, we divide the swarm into two subswarms. In one subswarm, since the move of a particle to the infeasible region are not accepted, if a particle becomes infeasible after move, it is repaired to be feasible. To be more specific, with respect to infeasible particles which violate constraints after move, we repair its search point to be feasible by the bisection method on the direction from the search point before move, $\boldsymbol{x}_i^t$, to that after move, $\boldsymbol{x}_i^{t+1}$. In the other subswarm, the move of a particle to the infeasible region are accepted.

5.3 Secession Since particles tend to concentrate on the best search point of the swarm as the search goes forward in PSO, as shown in Fig. 5, the global search becomes difficult. Thus, we introduce the following secession of a particle (Fig. 6).

(1) [Secession I] A particle moves at random to a point in the feasible region.

(2) [Secession II] A particle moves at random to a point on the boundary of the feasible region.

(3) [Secession III] A particle moves at random to a point in a direction of some coordinate axis.

5.4 Multiple stretching technique Stretching technique is suggested to prevent the stop at a local optimal solution of particles in PSO by Parsopoulos et al. [6]. It enables particles to escape from the current local optimal solution and not to approach the same local optimal solution again by changing original evaluation function $f(x)$ to other evaluation function $H(x)$ defined as:

$$G(\boldsymbol{x}) = f(\boldsymbol{x}) + \gamma_1 \|\boldsymbol{x} - \bar{\boldsymbol{x}}\| \Big(\mathrm{sign}\big(f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}})\big) + 1 \Big) \tag{11}$$

$$H(\boldsymbol{x}) = G(\boldsymbol{x}) + \gamma_2 \frac{\mathrm{sign}\big(f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}})\big) + 1}{\tanh\Big(\mu\big(G(\boldsymbol{x}) - G(\bar{\boldsymbol{x}})\big)\Big)} \tag{12}$$

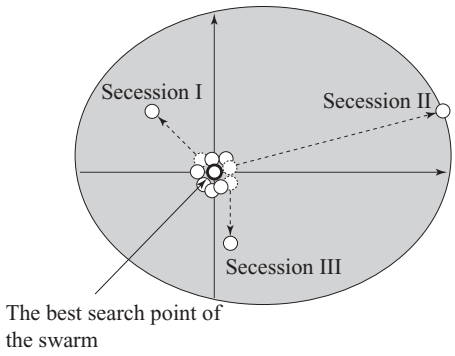

Figure 6: The secession.

where $\bar{\boldsymbol{x}}$ is the current local optimal solution, and γ_1, γ_2, μ are parameters. In addition, sign$(\cdot)$ is defined as follows.

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{13}$$

In $G(\boldsymbol{x})$, the second term is the penalty depending on the distance between a search point $\boldsymbol{x}$ and the current local optimal solution $\bar{\boldsymbol{x}}$. The term is equal to 0 for $\boldsymbol{x}$ whose objective function value $f(\boldsymbol{x})$ is better than $f(\bar{\boldsymbol{x}})$, while it takes a value depending on the distance between $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ for $\boldsymbol{x}$ whose $f(\boldsymbol{x})$ is worse than $f(\bar{\boldsymbol{x}})$. Next, $H(\boldsymbol{x})$ is defined using $G(\boldsymbol{x})$ expressed in (11). The value of $H(\boldsymbol{x})$ for a search point $\boldsymbol{x}$ is equal to the objective function value $f(\boldsymbol{x})$ of $\boldsymbol{x}$ if $f(\boldsymbol{x})$ of $\boldsymbol{x}$ is better than that of $\bar{\boldsymbol{x}}$, while it takes a very large value if $f(\boldsymbol{x})$ of $\boldsymbol{x}$ is worse than that of $\bar{\boldsymbol{x}}$. Using $H(\boldsymbol{x})$ as the new evaluation function, particles can escape from the current local optimal solution and search a new region which may include better solutions than the current local optimal solution.

Although the stretching technique [6] enables particles to escape from the current local optimal solution, they may stop at the same local optimal solution again when we apply the stretching technique to the next local optimal solution. Thus, we use the multiple stretching technique corresponding to plural local optimal solutions. To be concrete, we consider the following functions for Q local optimal solutions $\bar{\boldsymbol{x}}_q$, $q = 1, 2, \ldots, Q$.

$$G_q(\boldsymbol{x}) = f(\boldsymbol{x}) + \gamma_1 \|\boldsymbol{x} - \bar{\boldsymbol{x}}_q\| \Big(\text{sign}\big(f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}_{\min})\big) + 1 \Big) \tag{14}$$

$$H_q(\boldsymbol{x}) = G_q(\boldsymbol{x}) + \gamma_2 \frac{\text{sign}\big(f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}_{\min})\big) + 1}{\tanh\Big(\mu\big(G_q(\boldsymbol{x}) - G_q(\bar{\boldsymbol{x}}_q)\big)\Big)} \tag{15}$$

$$S(\boldsymbol{x}) = \sum_{q=1}^{Q} H_q(\boldsymbol{x})/Q \tag{16}$$

Here, $\bar{\boldsymbol{x}}_{\min}$ is the best among Q local optimal solutions. The value of $S(\boldsymbol{x})$ for a search point $\boldsymbol{x}$ is equal to the objective function value $f(\boldsymbol{x})$ of $\boldsymbol{x}$ if $f(\boldsymbol{x})$ is better than that of $\bar{\boldsymbol{x}}_{\min}$, while it takes a very large value if the distance between $\boldsymbol{x}$ and the nearest local optimal solution is less than a certain value. Otherwise, it takes a value depending on the distance.

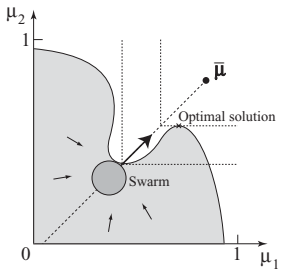

Figure 7: Application of rPSO to the augemented minimax problem.

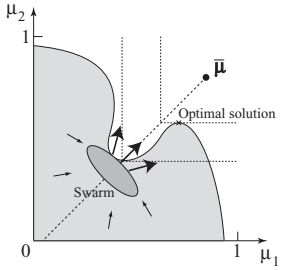

Figure 8: Application of the proposed PSO to the augemented minimax problem.

5.5 Incorporation of the direction to nondominated particles in the swarm
The shape of the objective function of the augmented minimax problem (4), solved in the interactive fuzzy satisficing method for multiobjective nonlinear programming problems, often becomes complex. Therefore, there does not always exist the optimal solution in the search direction of the swarm, as shown in Fig. 7.

Thus, considering the fact that the optimal solution to (4) is one of M-Pareto optimal solutions to (2), we introduce the direction to a nondominated particle (an approximate M-Pareto optimal solution) to carry out search with respect to both M-Pareto optimality and the optimality of the objective function of (4), as shown in Fig. 8.

In order to incorporate the new search direction, we revise (7) to determine the next search direction vector as:

$$\boldsymbol{v}_i^{t+1} = \omega^t \boldsymbol{v}_i^t + c_3 R_3^t (\boldsymbol{x}_K^t - \boldsymbol{x}_i^t) \tag{17}$$

where $\boldsymbol{x}_K^t$ is the search point of a certain nondominated particle K in the current swarm, c_3 is a parameter and R_3^t is the uniform random number in the interval $[0, 1]$.

Moreover, when the current search point is the best search point of a particle, it is considered that the current search point and the current search direction vector are promising since the best point of a particle is just updated. Therefore, the next search direction vector $\boldsymbol{v}_i^{t+1}$ is calculated by the translation equation (10) like Morihara [5]. By the swarm involving particles which move in the manner mentioned above, the search with respect to M-Pareto optimality is carried out.

In rPSO [5], the next search point is decided by a vector to the k-th particle instead of the best search point of the swarm in order to restrain the concentration of particles on the best search point of the swarm when the best search point of a particle is updated

just before that. Unfortunately, it does not seem suitable in order to solve the augmented minimax problem since the direction to the k-th particle does not seem good if the k-th particle is dominated.

Thus, we determine the next search direction vector $\boldsymbol{v}_i^{t+1}$ by the following equation (18) using the direction to the current search point of a nondominated particle K in its track instead of the best search point of the k-th particle.

$$\boldsymbol{v}_i^{t+1} = c_1 R_1^t(\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + c_2 R_2^t(\boldsymbol{x}_K^t - \boldsymbol{x}_i^t) \tag{18}$$

This modification leads to move to the direction to the nondominated solution as well as the restraint of the concentration on local optimal solutions.

5.6 Algorithm of the proposed PSO method The algorithm of the proposed PSO method based on rPSO for multiobjective nonlinear programming problems (MOrPSO) is summarized as follows.

Step 1: Find one feasible solution by rPSO in consideration of the degree of violation of constraints, and set it a basic point solution of the homomorphous mapping $\boldsymbol{r}$. Set $t := 0$, and go to step 2.

Step 2: Generate the initial search point $\boldsymbol{x}_i^0$ of each particle which is feasible or satisfies all constraints, using the homomorphous mapping [4]. To be more concrete, generate N points in a hypercube $[-1, 1]^n$ randomly, and map them into the feasible region using the homomorphous mapping. Let these points obtained by the mapping be initial search points $\boldsymbol{x}_i^0$, $i = 1, 2, \ldots, N$ and let $\boldsymbol{p}_i^0 := \boldsymbol{x}_i^0$, $i = 1, 2, \ldots, N$. Then, find the best search point among $\boldsymbol{p}_i^0$, $i = 1, 2, \ldots, N$, and use it as the best search point of the initial swarm $\boldsymbol{p}_g^0$. Go to step 3.

Step 3: Determine the parameter value ω^t. If the conditions of using the information of nondominated particles are satisfied, calculate the next search direction vector $\boldsymbol{v}_i^{t+1}$ and the next search point $\boldsymbol{x}_i^{t+1}$ of each particle using the move scheme depending on its situation based on (17) and (18). Otherwise, calculate $\boldsymbol{v}_i^{t+1}$ and $\boldsymbol{x}_i^{t+1}$ of each particle using the usual move scheme depending on its situation based on (7), (9) and (10). Go to step 4.

Step 4: For particles of the subswarm limited to the feasible region, if there exist infeasible ones, they are repaired to be feasible by the bisection method. Go to step 5.

Step 5: Check whether all conditions for the execution of the multiple stretching technique are satisfied or not. If satisfied, go to step 6. Otherwise, go to step 7.

Step 6: Calculate $S(\boldsymbol{x}_i^{t+1})$, $i = 1, 2, \ldots, N$ where $S(\cdot)$ is the evaluation function in the multiple stretching technique are satisfied and use them as objective function values instead of $f(\boldsymbol{x}_i^{t+1})$. Go to step 7.

Step 7: Calculate $f(\boldsymbol{x}_i^{t+1})$, $i = 1, 2, \ldots, N$ and use them as objective function values as usual. Go to step 8.

Step 8: For each particle, if $\boldsymbol{x}_i^{t+1}$ is not dominated by any current nondominated particles in the external archive, add $\boldsymbol{x}_i^{t+1}$ to the external archive. Then, remove solutions dominated by $\boldsymbol{x}_i^{t+1}$ from the external archive. Moreover, if $\boldsymbol{x}_i^{t+1}$ is better than $\boldsymbol{p}_i^t$ in the sense of the objective function $f(\cdot)$ or $S(\cdot)$, update the best search point of the particle in its track as $\boldsymbol{p}_i^{t+1} := \boldsymbol{x}_i^{t+1}$. Otherwise, let $\boldsymbol{p}_i^{t+1} := \boldsymbol{p}_i^t$ and go to step 9.

Step 9: Find the best $\boldsymbol{p}_{\mathrm{best}}^{t+1}$ among $\boldsymbol{p}_i^{t+1}$, $i = 1, 2, \ldots, N$. If $\boldsymbol{p}_{\mathrm{best}}^{t+1}$ is better than $\boldsymbol{p}_g^t$ in the sense of the objective function $f(\cdot)$ or $S(\cdot)$, update the best search point of the particle in its track as $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_{\mathrm{best}}^{t+1}$. Otherwise, let $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_g^t$ and go to step 10.

Step 10: For each particle, apply the secession operation. Let $t := t + 1$ and go to step 11.

Step 11: If $t > T_{\max}$ (the maximal search time), the search procedure is terminated. Otherwise, go to step 3.

Here, the termination condition means that the search time counter t reached to given maximal search time $T_{\max}$. An application condition of multiple stretching technique means that the best solution of the swarm has not been updated during a given period.

6 Numerical example In order to show the efficiency of the proposed PSO (MOrPSO), we consider the following multiobjective nonlinear programming problem.

$$
\begin{aligned}
&\text{minimize}\\
&\quad f_1(\boldsymbol{x}) = 7x_1^2 - x_2^2 + x_1x_2 - 14x_1 - 16x_2 + 8(x_3 - 10)^2\\
&\qquad +4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2\\
&\qquad +7(x_8 - 11)^2 + 2(x_9 - 10)^2 + x_{10}^2 + 45\\
&\text{minimize}\\
&\quad f_2(\boldsymbol{x}) = (x_1 - 5)^2 + 5(x_2 - 12)^2 + 0.5x_3^4 + 3(x_4 - 11)^2\\
&\qquad +0.2x_5^5 + 7x_6^2 + 0.1x_7^4 - 4x_6x_7 - 10x_6 - 8x_7\\
&\qquad +x_8^2 + 3(x_9 - 5)^2 + (x_{10} - 5)^2\\
&\text{minimize}\\
&\quad f_3(\boldsymbol{x}) = x_1^3 + (x_2 - 5)^2 + 3(x_3 - 9)^2 - 12x_3 + 2x_4^3\\
&\qquad +4x_5^2 + (x_6 - 5)^2 + 6x_7^2 + 7(x_7 - 2)x_8^2\\
&\qquad -x_9x_{10} + 4x_9^3 + 5x_1 - 8x_1x_7\\
&\text{subject to}\\
&-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4\\
&\qquad\qquad -2x_5x_6x_8 + 120 \geq 0\\
&-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0\\
&-x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 - 6x_5x_6 \geq 0\\
&-0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_5x_8 + 30 \geq 0\\
&3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0\\
&4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 105\\
&10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0\\
&-8x_1 + 2x_2 + 5x_9 - 17x_{10} \leq 12\\
&-5.0 \leq x_j \leq 10.0,\ \ j = 1, \ldots, 10
\end{aligned}
$$

We apply the original rPSO [5], RGENOCOP V [10] and the proposed PSO (MOrPSO) to minimax problems solved in the interactive fuzzy satisficing method for the above problem. The results obtained by these three methods are shown in Table 1. In these experiments, we set the swarm (population) size $N = 70$, the maximal search generation number $T_{\max} = 5000$, the number of trials is 10. In addition, we use the following membership functions: $\mu_{f_1}(\boldsymbol{x}) = (1500 - f_1(\boldsymbol{x}))/1420$, $\mu_{f_2}(\boldsymbol{x}) = (3500 - f_2(\boldsymbol{x}))/3300$, $\mu_{f_3}(\boldsymbol{x}) = (3100 - f_3(\boldsymbol{x}))/3050$.

From Table 1, in the application of rPSO [5], we can get better solutions in the sense of best, average and worst than those obtained by RGENOCOP V [10]. This indicates that rPSO is better than RGENOCOP V about the accuracy of solutions. However, as for the difference between best and worst, that for rPSO is larger than that for RGENOCOP V, i.e., rPSO is worse than RGENOCOP V with respect to the precision of solutions.

Table 1: Results of the application of three methods to augmented minimax problems.

	Interaction	1st	2nd	Time
	$(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3)$	(1.0, 1.0, 1.0)	(0.8, 1.0, 1.0)	(sec)
MOrPSO (proposed)	Best	0.1430	0.0823	9.03
	Average	0.1452	0.0838	
	Worst	0.1488	0.0859	
rPSO [5]	Best	0.1434	0.0826	8.55
	Average	0.1466	0.0854	
	Worst	0.1674	0.1102	
RGENOCOP V [10]	Best	0.1825	0.1406	42.90
	Average	0.1880	0.1568	
	Worst	0.1952	0.1811	

Table 2: Results of application of two methods to augmented minimax problems.

	Interaction	1st	2nd	Time
	$(\bar{\mu}_1, \bar{\mu}_2,)$	(1.0, 1.0)	(0.8, 1.0)	(sec)
MOrPSO (proposed)	Best	0.3861	0.2222	125.83
	Average	0.4843	0.2795	
	Worst	0.6288	0.3869	
rPSO [5]	Best	0.4460	0.2569	124.93
	Average	0.5160	0.3006	
	Worst	0.6335	0.4170	

On the other hand, the results obtained by the proposed MOrPSO are better than those by rPSO in the sense of best, average, worst, the difference between best and worst.

In addition, we applied rPSO and MOrPSO to another problem which has 2 objective functions, 100 decision variables and 55 constraints. Table 2 shows results for the problem. As the results for the previous problem, MOrPSO can obtain better solutions than those by rPSO.

Therefore, MOrPSO proposed in this paper is the best solution method among these methods with respect to both the accuracy of solutions and the precision of solutions.

7 Conclusion In this paper, we focused on multiobjective nonlinear programming problems and proposed a new PSO technique which is efficient for in applying the interactive fuzzy satisficing method. In particular, considering the features of augmented minimax problems solved in the interactive fuzzy satisficing method, we incorporated the new search direction vector determination scheme based on the information of nondominated particles into rPSO. Finally, we showed the efficiency of the proposed MOrPSO by applying it to numerical examples.

Acknowledgment This research was partially supported by The 21st Century COE Program on "Hyper Human Technology toward the 21st Century Industrial Revolution" and the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 18510127, 2006.

References

[1] R.E. Bellman, L.A. Zadeh, Decision making in a fuzzy environment, Management Science, vol. 17, pp. 141–164, 1970.

[2] J. Kennedy and R.C. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.

[3] J. Kennedy, W.M. Spears, Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, Proceedings of IEEE International Conference on Evolutionary Computation, pp. 78–83, 1998.

[4] S. Koziel, Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, Evolutionary Computation, vol. 7, no. 1, pp. 19–44, 1999.

[5] T. Matsui, K. Kato, M. Sakawa, T. Uno, K. Morihara Particle swarm optimization based heuristics for nonlinear programming problems, Proceedings of International MultiConference of Engineers and Computer Scientists 2007, pp. 2312–2317, 2007.

[6] K.E. Parsopoulos, M.N. Varahatis, Recent approaches to global optimization problems through particle swarm optimization, Natural Computing, no. 1, pp. 235–306, 2002.

[7] H. Rommelfanger, Fuzzy linear programming and applications, European Journal of Operational Research, vol. 92, pp. 512–527, 1996.

[8] M. Sakawa, Fuzzy Sets and Interactive Multiobjective Optimization, Plenum Press, New York, 1993.

[9] M. Sakawa, Genetic Algorithms and Fuzzy Multiobjective Optimization, Kluwer Academic Publishers, 2001.

[10] M. Sakawa, K. Kato, T. Suzuki, An interactvie fuzzy satisficing method for multiobjective nonconvex programming problems through genetic algorithms, Proceedings of 8th Japan Society for Fuzzy Theory and Systems Chugoku / Shikoku Branch Office Meeting, pp. 33–36, 2002.

[11] Y.H. Shi, R.C. Eberhart, A modified particle swarm optimizer, Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73, 1998.

[12] H.-J. Zimmermann, Fuzzy programming and linear programming with several objective functions, Fuzzy Sets and Systems, vol. 1 pp. 45–55, 1978.

Graduate School of Engineering, Hiroshima University
1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan
Kosuke Kato
TEL: 082-424-7693 FAX: 082-422-7195
E-mail: kosuke-kato@hiroshima-u.ac.jp