Electronics and Communications in Japan, Part 3, Vol. 81, No. 8, 1998
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J80-A, No. 1, January 1997, pp. 170–177

An Interactive Fuzzy Method for Multiobjective 0-1 Programming Problems with Fuzzy Number Criteria Using Genetic Algorithms

Masatoshi Sakawa

Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, Japan 739

Toshihiro Shibano

System Products Division, Sinryou Cooperation, Yokohama, Japan 220-81

Hidenobu Obata

Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, Japan 739

SUMMARY

The multiobjective 0-1 programming problem with fuzzy numbers is a formalization designed to represent expert judgment. Using the non-fuzzy α-multiobjective programming problem, in which the membership degrees of components of the coefficient vector are set in accordance with the decision maker's objectives, the concept of an α-Pareto optimal solution with respect to the fuzzy parameters of the problem and the decision maker's fuzzy objectives is introduced. An interactive fuzzy satisficing method is proposed in which α-Pareto optimal solutions are found by the expanded minimax method, the evaluation membership function and the fuzziness are interactively updated if the decision maker is not satisfied, and a solution acceptable to the decision maker is derived from the set of α-Pareto optimal solutions. A character string-coded genetic algorithm is used in solving the expanded minimax problem. The validity of the method is demonstrated by means of numerical examples. © 1998 Scripta Technica, Electron Comm Jpn Pt 3, 81(8): 64–72, 1998

Key words: Satisficing method; fuzzy satisficing; multiobjective 0-1 programming problem; fuzzy numbers; α-Pareto optimal solution; interactive method; genetic algorithm.

1. Introduction

During the seventies, Professor John Holland, his colleagues, and students from Michigan University proposed genetic algorithms (GA) as a new learning paradigm that modeled the natural evolution mechanism [1], although this concept was not known initially by that name. After the publication of a book by Goldberg [2] GAs have attracted attention in various fields as a methodology for optimization adaptation and learning. Furthermore, since Michalewicz book [3] was published in 1992, research concerning

CCC1042-0967/98/080064-09

applications to optimization problems has increased. The second edition [3] was published in 1994, and further developments are expected in this field. Vector evaluated genetic algorithms (VEGA), proposed by Schaffer [4] for multiobjective optimization problems are based on an extension of the traditional fitness function of genetic algorithms from a scalar to a vector function. Further developments followed as a result of this idea [5]. However, in all of these multiple evaluation genetic algorithm techniques, the focus has been on how to determine efficiently the (sub)set of (locally) Pareto optimal solutions to multiple objective problems. However, the important problem in multiple decision making tasks is that of choosing a solution or a compromise, from among the set of Pareto optimal solutions that satisfies the decision maker (DM). This aspect of multiple objection problem solving has absolutely not been considered.

Recently, Sakawa and coworkers have formalized multiobjective 0-1 programming problems that consider fuzzy goals of the DM. They introduced two string genetic algorithms [6] in the extraction of cooperative solutions according to fuzzy decision rules that reflect the DM fuzzy goals. They also extended the method to interactive techniques [7].

In this context, the focus in this paper is on multiobjective 0-1 programming problems. In order to better represent the human judgment of experts concerned with problem formalization where several parameters are involved, the ambiguity of these parameters is a characteristic that is treated as a fuzzy number. By formalizing multiple objective 0-1 programming problems involving fuzzy numbers it is possible to approximate very well the actual decision making conditions. In a coefficient vector for which the degrees of membership of all the membership functions of the fuzzy numbers in the problem are greater or equal to α, there are some that are particularly good for the DM. We introduce a nonfuzzy α- 0-1 programming problem that establishes this α value. Furthermore, we propose an interactive decision making method in which we introduce the concept of an extended α-Pareto optimal solution that considers the ambiguity involved in the problem. We determine quasi α-Pareto extended mini-max optimal solutions for the base membership values and a degree of ambiguity α, subjectively established by the DM. If the solutions do not satisfy the DM, then a solution that satisfies the DM is extracted from the set of α-Pareto optimal solutions through interactive upgrading of the base membership values and the degree of ambiguity. Since the extended mini-max problem used to determine the α-Pareto optimal solutions becomes a 0-1 programming problem, we show that the solutions can be very easily determined by application of the two-string coded genetic algorithm proposed by Sakawa and others [6]. Finally, we demonstrate the feasibility of the proposed method by means of a numerical example.

2. Multiobjective 0-1 Programming Problem with Fuzzy Numbers

In order to adequately represent the human judgment of experts concerned with formalizing the problem, rather than setting up immediately a traditional heuristic or subjective method, in this paper we adopt a method in which parameters of the type "number approximately equal to *m*" are taken as fuzzy numbers for the purpose of more conveniently reflecting the actual multiobjective decision conditions. We formalize a 0-1 multiobjective programming problem with fuzzy numbers, by using fuzzy numbers to denote the ambiguity of the parameters involved in the objective functions and constraints of the problem,

$$\begin{aligned} &\text{minimize } (\tilde{\boldsymbol{c}}_1\boldsymbol{x}, \tilde{\boldsymbol{c}}_2\boldsymbol{x}, \ldots, \tilde{\boldsymbol{c}}_k\boldsymbol{x})^T \\ &\text{subject to } \tilde{A}\boldsymbol{x} \leqq \tilde{\boldsymbol{b}} \\ &\quad x_j \in \{0,1\}, \quad j = 1, \ldots, n \end{aligned} \tag{1}$$

Here, $\overline{A}$ is an $m \times n$ matrix with fuzzy coefficients, $\overline{c}$ and $\overline{b}$ are, respectively, n and m dimensional vectors of fuzzy numbers. For the sake of simplification in this paper, we assume that all the fuzzy number components of $\overline{A}$ and $\overline{b}$ are positive, and we can then regard the problem as a multiobjective multidimensional **knapsack** problem.

Since the coefficients in the objective functions and the constraints in the problem, Eq. (1), have the characteristics of fuzzy numbers, we cannot apply directly the concept of a Pareto optimal solution to a conventional multiobjective 0-1 programming problem. Therefore, we first introduce an α-level set [8, 9] in which the membership value of each fuzzy number is greater than or equal to α.

Definition 1. The set of all triplets (A, b, c) of values of the fuzzy number membership functions contained in the fuzzy parameters $\overline{A}$, $\overline{b}$, and $\overline{c}$ that are greater than or equal to α is called the α-level set and is denoted as $(\overline{A}, \overline{b}, \overline{c})_\alpha$.

Assuming that the Decision Maker (DM) for the problem in Eq. (1) judges a solution correct if the values of the membership functions of the fuzzy numbers involved in the objective function and constraints $(A, b, c) \in (\overline{A}, \overline{b}, \overline{c})_\alpha$ for some value greater than or equal to α, then we can introduce a nonfuzzy α-multiobjective 0-1 programming problem as follows,

$$\begin{aligned} &\text{minimize } (\boldsymbol{c}_1\boldsymbol{x}, \boldsymbol{c}_2\boldsymbol{x}, \ldots, \boldsymbol{c}_k\boldsymbol{x})^T \\ &\text{subject to } \boldsymbol{x} \in X(A, \boldsymbol{b}) \\ &\quad = \{\boldsymbol{x} \in \{0,1\}^n \mid A\boldsymbol{x} \leqq \boldsymbol{b}\} \\ &\quad (A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}})_\alpha \end{aligned} \tag{2}$$

The DM selects the most desirable value that is greater than or equal to α, which determines the degree of realizability of the problem out of the $(A, b, c) \in (\overline{A}, \overline{b}, \overline{c})_\alpha$.

We must pay attention to the fact that for the problem in Eq. (2), the coefficient vectors (A, b, c) are no longer regarded as coefficients but as variables. Thus, we define the α-Pareto optimal solution concept, that takes into account the vagueness involved in the problem by extending the ordinary Pareto optimal solution concept to Eq. (2).

Definition 2. $x^* \in X(A^*, b^*)$ is called an α-Pareto optimal solution for the problem in Eq. (2) and the corresponding set of coefficients (A^*, b^*, c^*) are called α-level optimal coefficients, if there does not exist an $x \in X(A, b)$, $(A, b, c) \in (\overline{A}, \overline{b}, \overline{c})_\alpha$ such that $c_i x \leq c_i x^*$, $i = 1, \ldots, k$, with at least one j satisfying $c_j x < c_j x^*$.

Let us consider in a little more detail the meaning of α-Pareto optimal solution and α-level Pareto coefficients for Eq. (2). Since the fuzzy number coefficients included in the problem are decision variables that can move freely over an α-level set in which the membership function value is greater than or equal to the α value in Eq. (2), it is possible to obtain a solution that reflects the vagueness created in the problem formalization. In this case the value α, which expresses a degree of realizability of all the fuzzy numbers involved in Eq. (1), can be set subjectively to the most desirable value according to the DM. It is, therefore, obvious that in Eq. (2) the α Pareto optimal solution x^* and α-level Pareto coefficients (A^*, b^*, c^*) correspond to a Pareto optimal solution (x^*, A^*, b^* , c^*) when the decision variables, considered as (x, A, b, c), have the ordinary meaning.

Now, considering the vagueness of judgment by human decision makers, we might consider that the DM has an ambiguous goal with respect to each objective function of the α-multiobjective 0-1 programming problem [8–10] and that the fuzzy goals for the minimization problem are something like "I want the objective function $c_i x$ most of the time less than or equal to p_i." The linear membership function

$$\mu_i(\boldsymbol{c}_i\boldsymbol{x}) = \begin{cases} 0 & ; \boldsymbol{c}_i\boldsymbol{x} > z_i^0 \\ \dfrac{\boldsymbol{c}_i\boldsymbol{x} - z_i^0}{z_i^1 - z_i^0} & ; z_i^1 < \boldsymbol{c}_i\boldsymbol{x} \leq z_i^0 \\ 1 & ; \boldsymbol{c}_i\boldsymbol{x} \leq z_i^1 \end{cases} \tag{3}$$

illustrated in Fig. 1 is frequently used as a membership function for such fuzzy objective characteristics within the range of individual minimum and maximum for each objective function. Here the straight line from 1 to 0 joins at the corresponding values z_i^0 and z_i^1 of the objective function. The DM subjectively evaluates the z_i^0 and z_i^1.

A concrete decision method with linear membership functions of this kind is found in Zimmermann [10]. He used $z_i^{min} = c_i x^{i0}$ and

$$z_i^m = \max(\boldsymbol{c}_i\boldsymbol{x}^{1o}, \ldots, \boldsymbol{c}_i\boldsymbol{x}^{i-1,o}, \boldsymbol{c}_i\boldsymbol{x}^{i+1,o}, \ldots, \boldsymbol{c}_i\boldsymbol{x}^{ko}) \tag{4}$$

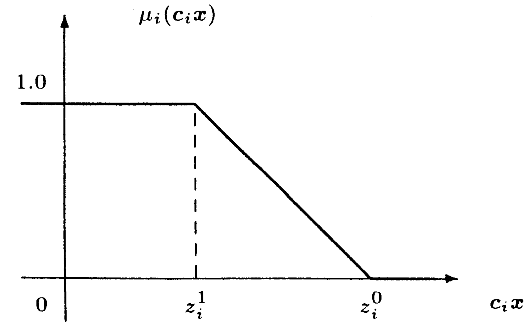

Fig. 1. Linear membership function.

where x^{i0} is the optimal solution to the particular optimization problem for each objective function under the given constraint conditions. Zimmerman proposed $z_i^1 = z_i^{min}$, $z_i^0 = z_i^m$, $i = 1, \ldots, k$ in the linear membership functions of Eq. (3).

Having determined the membership functions for each objective function, the decision problem for the multiobjective 0-1 programming problem with fuzzy parameters consists of trying to maximize all of the membership functions and, simultaneously, to maximize the degree of realizability of the problem. The decision problem that considers simultaneously the $k + 1$ competing objective functions of the DM can be formalized by means of the following definition, assuming the existence of an aggregation function that expresses the DM's selected structure.

$$\begin{aligned} &\text{maximize } \mu_D(\mu_1(\boldsymbol{c}_1\boldsymbol{x}), \ldots, \mu_k(\boldsymbol{c}_k\boldsymbol{x}), \alpha) \\ &\text{subject to } (\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \in P(\alpha),\ \alpha \in [0, 1] \end{aligned} \tag{5}$$

Here, $P(\alpha)$ is the set of α-level optimal coefficients corresponding to the α-Pareto optimal solutions. We assume also that the aggregation function $\mu_D(\cdot)$ is usually strictly monotonically increasing with respect to $\mu_i(\cdot)$ and α. If we could identify directly the function form for $\mu_D(\cdot)$ then the problem would become an ordinary single objective optimization problem. However, since it is difficult to identify $\mu_D(\cdot)$ globally, it is necessary to determine a self satisfying solution from among the ordinarily infinite point set of α-Pareto solutions by means of interaction with the DM.

3. Extended Mini-Max Problem

Let us now assume that the DM has subjectively set up the base membership functions $\overline{\mu}_i$, $i = 1, \ldots, k$ that reflect the DM reference levels for each of the membership functions $\mu_i(c_i x)$. In this case, if the establishment of the base membership value is moderately attainable, we look for a desirable α-Pareto optimal solution that is better than the membership values. If the determination of the base membership values is not attainable, than it is desirable to determine an α-Pareto optimal solution as close as possible to the membership value. Such α-optimal solutions can be obtained as the solution to the following mini-max problem [8, 9].

$$\begin{aligned} &\text{minimize} \max_{i=1,\ldots,k} \{\bar{\mu}_i - \mu_i(\boldsymbol{c}_i\boldsymbol{x})\} \\ &\text{subject to } A\boldsymbol{x} \leqq \boldsymbol{b} \\ &\quad x_j \in \{0,1\}, \qquad j = 1,\ldots,n \\ &\quad (A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}})_\alpha \end{aligned} \tag{6}$$

However, if the uniqueness of the optimal solution obtained by solving the mini-max problem is to be guaranteed, then one is not limited to α-Pareto optimal solutions. Hence, we encounter the problem of having to do tests for common α-Pareto optimality. In order to circumvent this problem it is recommended to search for an α-Pareto optimal solution close to the meaning of a mini-max in the base membership values by using the following extended mini-max problem instead.

$$\begin{aligned} &\text{minimize} \\ &\max_{i=1,\ldots,k} \left\{ (\bar{\mu}_i - \mu(\boldsymbol{c}_i\boldsymbol{x})) + \rho \sum_{i=1}^{k} (\bar{\mu}_i - \mu(\boldsymbol{c}_i\boldsymbol{x})) \right\} \\ &\text{subject to } A\boldsymbol{x} \leqq \boldsymbol{b} \\ &\quad x_j \in \{0,1\} \qquad j = 1,\ldots,n \\ &\quad (A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}})_\alpha \end{aligned} \tag{7}$$

Here, ρ is a sufficiently small positive number.

We should observe that, since in Eqs. (6) and (7) A, b, and c are treated as variables, the constraint conditions in this situation also become nonlinear.

Fortunately, because of the properties of the α-level set for the matrix $\overline{A}$ of fuzzy coefficients and the vectors $\overline{c}$ and $\overline{b}$ of fuzzy numbers, the domain of realizability of $\overline{A}$, $\overline{b}$, and $\overline{c}$ can be expressed by closed intervals $[A^L_\alpha, A^R_\alpha]$, $[b^L_{j\alpha}, b^R_{j\alpha}]$, and $[c^L_{i\alpha}, c^R_{i\alpha}]$ by means of the left and right end points of the α-level sets. Thus, obtaining an optimal solution for the extended mini-max problem is equivalent to solving the following problem:

$$\begin{aligned} &\text{minimize} \\ &\max_{i=1,\ldots,k} \left\{ (\bar{\mu}_i - \mu(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x})) + \rho \sum_{i=1}^{k} (\bar{\mu}_i - \mu(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x})) \right\} \\ &\text{subject to } A^L_\alpha\boldsymbol{x} \leqq \boldsymbol{b}^R_\alpha \\ &\quad x_j \in \{0,1\}, \qquad j = 1,\ldots,n \end{aligned} \tag{8}$$

Since the constraints of this 0-1 programming problem are linear and all the coefficients are positive, we try to apply the two-string coded genetic algorithms proposed by Sakawa and others [6, 7].

4. Double String Genetic Algorithms

4.1. Double string coding

The double string coding as shown in Fig. 2 has been proposed for multiobjective 0-1 programming with linear constraints Ax^2b, with all the components of A and b positive [6, 7]. The elements in the upper part are the variable indices and those in the lower part the values of the variables. Here, $s(i)$ corresponds to the index j of the variable x_j, and $x_{s(i)}$ is the value of the variable x_j, corresponding to $s(i)$.

If we decode the double string according to the following algorithm, we generate only realizable solutions. Here, the length of the string is n, the position of a string is i, the index of the variable $s(i)$, the value of the variable $x_{s(i)}$, the coefficient of the constraint formula $a_{s(i)}$, and the constraint condition $\Sigma^n_{i=1} a_s(i) x_{s(i)} \le b$. Denote by $p_{s(i)}$ the decoded value of $x_{s(i)}$.

Step 1: Make $i = 1$, sum $= 0$

Step 2: If $x_{s(i)} = 0$ take $p_{s(i)} = 0$, $i = i + 1$ and go to step 4; if $x_{s(i)} = 1$, go to step 3.

Step 3: If sum $+ a_{s(i)} \le b$, take $p_{s(i)} = 1$, $\sum = \sum + a_{s(i)}$, $i = i + 1$ and go to step 4; otherwise take $p_{s(i)} = 0$, $i = i + 1$ and go to step 4.

Step 4: If $i > n$, stop. Otherwise, go back to step 2.

In this algorithm, the elements for which the variable $x_{s(i)}$ is 1 satisfy the constraints and are sequentially fixed starting from the left-hand side of the strings within the range, all other variables are fixed to 0.

4.2. Fitness

It is natural to set the fitness function $f(s)$ of the string s as

$$f(\boldsymbol{s}) = 1.0 - \max_{i=1,\ldots,k} \left\{ (\bar{\mu}_i - \mu_i(\boldsymbol{c}^L_i\boldsymbol{x})) + \rho \sum_{i=1}^{k} (\bar{\mu}_i - \mu_i(\boldsymbol{c}^L_i\boldsymbol{x})) \right\}$$

Index	$s(1)$	$s(2)$	$\cdots\cdots$	$s(i)$	$\cdots\cdots$	$s(n)$
Variable value	$x_{s(1)}$	$x_{s(2)}$	$\cdots\cdots$	$x_{s(i)}$	$\cdots\cdots$	$x_{s(n)}$

Fig. 2. Double string.

In addition, we use a linear scaling $f_i' = a \cdot f + b$ to scale the fitness. The coefficients a and b here become the invariant points of the average fitness of the group of individuals and the optimal fitness is set up so as to linearly reflect on the squared value of the average fitness.

4.3. Reproduction

Sakawa and coworkers [6] investigated six types of reproduction operators (ranking selection, elitist ranking selection, expected value selection, elitist expected value selection, roulette selection, and elitist roulette selection). The results showed that the elitist expected value selection is relatively more efficient. Therefore, in this paper we use the elitist expected value selection that includes in the expected value selection the so-called elitist preserving selection. The expected value selection carries out reproduction by the expected value of the individuals remaining in the following generation. The elitist preserving selection unconditionally retains individuals with maximum fitness out of the present generation individuals.

4.4. Crossover

Since in two-string coding there are letters other than 0 and 1 in the upper subset of indices, it is possible when doing the traditional one point and multiple point crossovers that similar index variable numbers of the variables emerge in the upper part of indices. To avoid this inconvenience, in this paper we use the partially matched crossover PMX [5]. The PMX for two strings is given by the following sequence.

Step 1: Set two 2-strings X and Y. Let the elements of index i in the strings X and Y be $s_X(i)$, $s_Y(i)$, and let $x_{sX(i)}$, $x_{sY(i)}$ be the corresponding values of the variables. Select two crossover points at random and determine the substrings to be exchanged.

Step 2: Taking the indices of the substring of Y that changes to be identical to the indices of the substring of X that changes, use the operations (1), (2), and (3) below and let X' be the result of changing the values of the indices and variables of X.

(1) Make h and k ($> h$) the first and last ends of the changing substring and make $i = h$.

(2) Determine j such that $s_Y(i) = s_X(j)$ and exchange the i-th column $(s_X(i), x_{s_X}(i))^T$ of X with the j-th column $(s_X(j), x_{s_X}(j))^T$ of X, and set $i = i + 1$.

(3) If $i > k$, stop. Otherwise, go to (2).

Step 3: Let X^* be the string that results when substituting the transformed substring of Y into the respective X'. Similarly, determine Y^*. Having obtained X^* and Y^* by crossover, stop.

4.5. Mutations and inversions

In individuals coded by single strings, mutations are carried out by exchanging the elements at two arbitrary positions in the string. However, in the representation of individuals by double strings, decoding is achieved by giving preference in a sequence from the left hand elements, so it becomes difficult to yield a better individual if only elements of the indices in the upper subset are exchanged. In order to deal with this problem, it is convenient to append a so-called inversion genetic operator that reverses the order of a substring of some length. The inversion for double strings is expressed by the following steps:

Step 1: Select two positions in the double string, k and h, ($k > h$), and in the upper index section select the substring going from position h to position k.

Step 2: Rearrange the substring between positions h and k in reverse order.

Step 3: Replace the original substring in the upper index section of the double string with the reversed substring and stop.

5. Interactive Fuzzy Method

The basic algorithm that introduces into interactive fuzzy programming [8, 9] a GA which generates only realizable solutions [7] is as follows.

Step 0: For $\alpha = 0.1$, determine the respective maximum and minimum values of each objective function in the given constraint domains.

Step 1: The DM subjectively defines the linear membership functions taking into consideration the different maximum and minimum values of each objective function.

Step 2: The DM sets the initial value of $\alpha (0 \le \alpha \le 1)$ and sets the initial membership values to 1.

Step 3: Randomly create N double string coded individuals and generate the initial group of individuals.

Step 4: Determine the fitness of each individual and carry out reproductions corresponding to each fitness under fixed rules.

Step 5: Carry out crossover with a predefined crossover probability p_c and generate new individuals.

Step 6: Carry out mutations with a predefined mutation probability p_m and generate new individuals.

Step 7: Repeat steps 4 to 6 until the final conditions are satisfied. Take then the individuals with maximum fitness as optimal solutions and go to Step 8.

Step 8: If the attainment levels of the present membership functions and objective functions of the α-Pareto optimal solution satisfy the α value, stop. Otherwise upgrade the base membership values and the α value considering the present membership values, the objective function achievement levels, and the α value and go back to Step 3. The DM should pay attention to (1) given a fixed α value, do not sacrifice the satisfaction degree of membership functions in order to improve the satisfaction degree of some specific membership functions of the α-Pareto optimal solution; and (2) for a fixed base membership value, do not sacrifice the objective function and the membership function achievement level by enhancement of the α-value that shows the realizability of the problem.

With respect to the generation method used here for the initial group of individuals in each interactive iteration, part of the quasi α-Pareto optimal solution obtained in an interaction is retained as part of the initial group of individuals of the subsequent interaction and the rest of the individuals in the group are generated randomly. By introducing a method that generates initial groups keeping a quasi α-Pareto optimal solution into a GA that uses both the expected value selection and the elitist selection as reproduction models, we expect to determine at least one solution that is not governed by the previous interactive solution.

When introducing genetic algorithms into an interactive fuzzy programming method, it is necessary to obtain an approximate solution within an adequate time interval. Therefore, we introduce the following convergence condition algorithm that uses the so-called lowest generation searching index I_{min} and maximum generation searching index value I_{max}.

Step 1: Let $t = 1$ and use a convergence error ε.

Step 2: Carry out a series of GA searches (regeneration, crossover, mutation).

Step 3: Determine the average and maximum fitness, f_{mean} and f_{max}, of the group of individuals.

Step 4: If $t > I_{min}$ and $(f_{max} - f_{mean})/f_{max} \cdot < \varepsilon$, discontinue the search.

Step 5: If $t > I_{max}$, discontinue the search. Otherwise, set $t = t + 1$ and go back to Step 2.

6. Numerical Example

To provide a numerical example we take a three-objective 0-1 programming problem with 30 variables and two constraints involving fuzzy numbers, assuming that all fuzzy numbers membership functions are triangular and generating the problem coefficients randomly as follows:

(1) Using random numbers of the type 0.1000 (real numbers equal or up to one rank less than the small number points), we generate the coefficients a_{ij}. Similarly, we generate the negative coefficients c_{1j}, half of the c_{2j} negative and the other half positive, and the positive coefficients c_{3j}.

(2) The value b_i is created by multiplying the value of $\Sigma_{j=1}^{n} a_{ij}$ by a random number generated in the interval [0.25, 0.75].

(3) Use fuzzy numbers statistically with a 90% ratio from among the a_{ij}, b_i, and c_{ij} determined in (1) and (2). Set the left extreme points c_{ij0}^{L}, $c_{i0}^{L} > 0$ by multiplying the respective a_{ij} and c_{ij} by random numbers in [0.9, 1.0], the right extreme points b_{i0}^{R} and left extreme points $c_{i0}^{L} < 0$ by multiplying the respective b_i, and by random numbers in the interval [1.0, 1.1].

As an example of a problem executed in this way, we solved the extended mini-max problem for each interactive iteration for the values shown in Table 1 by means of 30 simulations using genetical algorithms and determined the α-Pareto optimal solutions. With respect to the parameters for the GA, we set 50 individuals, a crossover probability of $p_c = 0.9$, a mutation probability $p_m = 0.02$, a convergence condition error $\varepsilon = 0.05$, a lowest generation searching index $I_{min} = 300$ and maximum generation searching index value $I_{max} = 500$. We also set the coefficient of the extended mini-max problem to $\rho = 0.0001$.

In this problem, the values of $(\bar{\mu}_1, \bar{\mu}_2, \bar{\mu}_3; \alpha)$ are upgraded according to an imaginary DM as shown in Table 2, from (1.0, 1.0, 1.0; 1.0) to (0.9, 1.0, 1.0; 1.0), (0.9, 0.95, 1.0; 1.0), and (0.9, 0.95, 1.0; 0.9). Here, the objective functions values (z_1^1, z_1^0) (z_2^1, z_2^0) (z_3^1, z_3^0) whose membership function degree of satisfaction becomes 1 or 0, are determined by means of the Zimmermann technique [10], but depending here on the value of α. When $\alpha = 1.0$ they are given as (–10836.6, 0), (–10177.7, 0), (0, 8481.6), and when $\alpha = 0.9$ they become (–0890.38, 0) (–10242.74, 0), (0, 8435.05). Also, the number of α-Pareto optimal solutions obtained for each interactive session in the 30 GA simulations were respectively 23, 24, 28, and 30. Clearly, good results were obtained.

In the interactive stages of Table 2, the α-Pareto optimal solutions were first determined for initial standard membership values of 1 with respect to an ambiguity level $\alpha = 1.0$ established by the imaginary DM. However, the DM of this example is not satisfied with the attainment levels (0.6356, 0.6304, 0.6300) of the membership functions in the α-Pareto optimal solutions obtained and considers that if μ_1 is sacrificed, μ_2 and μ_3 will improve. For this reason, keeping the ambiguity level constant at 1, the DM decreases the base membership value of μ_1 from 1.0 to 0.9 and upgrades the base membership values to (0.9, 1.0, 1.0). With this DM option, the attainment levels of the member-

Table 1. Numerical example of three-objective 0-1 programming problem with 30 variables and 2 constraints involving fuzzy numbers

a_1	262.7	793.6	203.5	601.2	15.0	553.6	521.5	457.0	889.3	868.1
	178.9	400.3	218.4	910.7	741.5	943.5	862.4	203.3	914.4	996.7
	585.9	140.2	276.6	401.3	782.4	998.7	906.9	565.2	989.0	754.7
a_2	432.2	791.4	38.1	76.2	469.4	301.7	148.7	782.8	387.7	600.2
	583.3	206.4	828.3	694.1	373.6	27.8	977.6	648.2	681.3	487.1
	984.0	492.6	678.7	396.3	613.7	23.0	838.0	240.4	756.6	519.8
a_{10}^L	258.5	716.5	200.5	552.1	14.5	537.0	506.5	449.1	876.0	789.4
	175.3	378.6	210.4	835.2	690.6	934.7	832.9	188.0	831.0	906.8
	565.6	127.2	249.3	378.7	782.4	916.5	859.9	513.9	964.5	749.6
a_{20}^L	394.8	780.8	35.2	69.4	467.0	282.4	148.7	726.0	382.0	556.7
	525.6	187.9	813.9	677.2	346.0	27.2	910.4	618.1	681.3	473.6
	923.3	451.3	675.2	361.7	557.3	22.1	818.7	229.9	718.3	491.1
c_1	−263.7	−184.1	−446.8	−227.1	−954.8	−877.1	−608.5	−801.9	−981.1	−600.0
	−54.4	−463.8	−24.9	−568.3	−113.9	−21.1	−224.5	−327.2	−663.5	−658.64
	−749.3	−742.8	−13.0	−137.5	−397.7	−232.7	−66.8	−448.4	−39.3	−626.4
c_2	−50.2	209.5	−707.6	187.7	−655.84	989.1	−925.2	−116.9	−675.4	−606.24
	−981.1	190.9	121.6	597.6	937.3	963.0	−309.7	377.0	−766.7	−550.8
	−213.4	−609.9	−787.6	−11.7	−839.8	589.0	−39.3	−717.4	−604.0	608.8
c_3	656.6	277.4	65.0	665.3	340.7	618.7	124.7	100.9	497.1	792.9
	912.5	332.7	611.2	581.0	788.1	653.6	360.9	548.3	78.9	627.9
	969.5	368.2	828.1	797.7	331.0	465.7	12.7	326.3	957.1	539.7
c_{10}^L	−270.5	−188.8	−486.7	−244.9	−954.8	−892.3	−664.5	−871.2	−1022.9	−627.0
	−54.6	−502.1	−24.9	−614.7	−118.3	−21.8	−231.5	−347.6	−690.4	−668.6
	−786.4	−807.0	−14.0	−146.8	−414.9	−232.7	−67.4	−486.7	−42.0	−626.4
c_{20}^L	−50.2	208.7	−843.7	175.8	−706.9	930.2	−955.5	−127.8	−714.6	−639.3
	−1018.8	173.3	109.5	572.6	920.9	941.5	−340.0	343.4	−773.2	−580.8
	−230.8	−650.7	−846.0	−12.4	−919.6	568.2	−42.8	−772.1	−658.1	588.8
c_{30}^L	653.6	269.1	60.4	659.1	309.0	561.2	122.6	98.6	456.6	731.7
	889.1	301.4	608.6	540.7	728.1	602.8	326.2	495.1	78.9	593.9
	969.5	337.6	768.8	761.2	303.0	426.0	-11.7	310.4	954.7	505.8

$b_1 = 9548.8$, $b_{10}^R = 9591.8$, $b_2 = 10555.0$, $b_{20}^R = 10852.7$

ship levels in the α-Pareto optimal solutions obtained in the second interactive iteration are reflected as (0.6248, 0.6548, 0.6612) and we see that μ_2 and μ_3 are ameliorated by sacrificing μ_1. As seen in Table 2, the hypothetical DM of this example thinks that an additional third interactive iteration μ_3 will be ameliorated by sacrificing μ_2 a little, and upgrades the base membership values to (0.9, 0.95, 1.0). The attainment levels of the membership levels in the α-Pareto optimal solutions obtained in the third interactive iteration are (0.5848, 0.6313, 0.6927), ameliorating the degrees of satisfaction of the sequence μ_1, μ_2, μ_3, and reflecting the DM choice. The hypothetical DM of this example is for the moment satisfied with the balance between the attainment levels of the membership functions as they have been obtained, but thinks that by further sacrificing the α value in a fourth interactive iteration the objective functions and the attainment values of the membership functions can be ameliorated, and thus diminishes the α value from 1.0 to 0.9. Satisfied with the α-Pareto optimal solutions obtained in the fourth interactive iteration for $\alpha = 0.9$ and respective membership attainment levels of (0.5852, 0.6313, 0.6933), the imaginary DM of this example stops the iterations. In this example, an α-Pareto optimal solution satisfying the DM was obtained at the fourth interactive stage, after two upgrades for the base membership and one upgrade of α. It is clear however that if the α-Pareto optimal solution obtained with such interactions do not satisfy the DM, similar interactions are continued until a satisfactory solution is obtained.

7. Conclusions

Focusing on multiobjective 0-1 programming problems involving fuzzy numbers, we introduced in this paper a nonfuzzy multiobjective 0-1 programming problem. Furthermore, we proposed a method called an interactive fuzzy satisfaction method which, once the fuzzy objectives of the Decision Maker are described by membership functions, determines an approximate α-Pareto optimal solution in the

Table 2. Simulation results

		c_1x	c_2x	c_3x	μ_1	μ_2	μ_3	Number of elements
1st interaction	Obtained solution	−6888.2	−6416.0	3138.1	0.6356	0.6304	0.6300	23
$\bar{\mu}_1 = 1.0$		−6955.0	−6455.3	3150.8	0.6418	0.6343	0.6285	1
$\bar{\mu}_2 = 1.0$		−6770.9	−6694.8	2873.4	0.6248	0.6548	0.6612	4
$\bar{\mu}_3 = 1.0$		−6861.6	−6327.4	3215.0	0.6331	0.6217	0.6210	1
$\alpha = 1.0$		−6712.3	−6720.2	3038.4	0.6194	0.6603	0.6418	1
	Distributed solution	−6888.2	−6416.0	3138.1	0.6356	0.6304	0.6300	
2nd interaction	Obtained solution	−6770.9	−6664.8	2873.4	0.6248	0.6548	0.6612	24
$\bar{\mu}_1 = 0.9$		−6704.1	−6625.8	2860.7	0.6187	0.6510	0.6627	1
$\bar{\mu}_2 = 1.0$		−5910.4	−6603.3	2937.5	0.5454	0.6488	0.6537	2
$\bar{\mu}_3 = 1.0$		−6058.5	−6862.3	3060.9	0.5591	0.6743	0.6391	1
$\alpha = 1.0$		−6888.2	−6416.0	3138.1	0.6356	0.6304	0.6300	1
	Distributed solution	−6770.9	−6664.8	2873.4	0.6248	0.6548	0.6612	
3rd interaction	Obtained solution	−6336.8	−6423.7	2606.4	0.5848	0.6312	0.6927	28
$\bar{\mu}_1 = 0.9$		−6270.7	−6384.4	2593.7	0.5786	0.6273	0.6942	2
$\bar{\mu}_2 = 0.95$								
$\bar{\mu}_3 = 1.0$								
$\alpha = 1.0$								
	Distributed solution	−6336.8	−6423.7	2606.4	0.5848	0.6312	0.6927	
4th interaction	Obtained solution	−6372.9	−6466.0	2587.3	0.5852	0.6313	0.6933	30
$\bar{\mu}_1 = 0.9$								
$\bar{\mu}_2 = 0.95$								
$\bar{\mu}_3 = 1.0$								
$\alpha = 0.9$								
	Distributed solution	−6372.9	−6466.0	2587.3	0.5852	0.6313	0.6933	

extended mini-max sense for base membership functions and an ambiguity fitness level α subjectively established by the DM. The method then draws out a solution that satisfies the DM from the set of the α-Pareto optimal solutions, by interactively upgrading the base membership functions and the ambiguity fitness level α that satisfy the DM. Concerning the extended mini-max problem used to determine the α-Pareto optimal solutions, we showed here that it can be solved statistically by application of two-string genetic algorithms. Further, by using the simulation results from a numerical example, it was shown that DM satisfying solutions are relatively easily obtained by upgrading the base membership functions and the α value interactively.

In the future, we expect to extend the proposed method and to obtain a more general interactive fuzzy method for multiobjective integer programming problems involving fuzzy numbers.

REFERENCES

1. J.H. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press (1975). MIT Press, Cambridge (1992).
2. D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Massachusetts (1989).
3. Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin (1992). Second, extended edition (1994).
4. J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Publishers, pp. 160–168 (1985).
5. M. Sakawa and M. Tanaka. Genetic Algorithms. Asakuma Publishing Co. (1995).
6. M. Sakawa, M. Inuiguchi, H. Sunada, and K. Sawada. Fuzzy multiobjective combinatorial optimization using improved genetic algorithms. J. of the Japan Fuzzy Society, **6**, No. 1, pp. 177–185 (1994).
7. M. Sakawa, K. Kato, H. Sunada, and Y. Sonoda. Interactive fuzzy satisfaction method by improved genetic algorithms for multiobjective 0-1 programming problems. J. of the Japan Fuzzy Society, **7**, No. 2, pp. 361–370 (1995).
8. M. Sakawa. Fuzzy Sets and Interactive Multiobjective Optimization. Plenum Press, New York (1993).
9. M. Sakawa, T. Ishii, and K. Nishisaki. Soft Optimization. Asakuma Publishing Co. (1995).
10. H.-J. Zimmermann. Fuzzy programming and linear programming with several objective functions. Fuzzy Sets and Systems, **1**, pp. 45–55 (1978).

AUTHORS (from left to right)

  

Masatoshi Sakawa received his B.E. and D.E. degrees in Applied Mathematics and Physics from Kyoto University in 1970 and 1975, respectively. In 1975 he joined Kobe University as a Research Assistant in Engineering Systems in the Department of Systems Engineering of the Faculty of Engineering, and became Associate Professor in 1981. He joined Iwate University as a Professor in the Faculty of Engineering in 1987 and in 1990 he became professor and chair of Statistics and Management Engineering at the Faculty of Engineering, Hiroshima University. His main research interests are on decision making methods in multiobjective and fuzzy systems. He is author of *Optimization of Linear Systems*, *Optimization of Nonlinear Systems*, *Fundamental Concepts of Fuzzy Theory and Applications, Fundamentals of Management Mathematical Systems*, and *Soft Optimization and Genetic Algorithms,* all in Japanese, and *Fuzzy Sets and Interactive Multiobjective Optimizatio*n, from Plenum Press.

Toshihiro Shibano received his B.E. in Electrical Engineering from the Muroran Institute of Technology in 1979, and joined Shinryou Corporation in the same year. His main research interests are in fuzzy control and fuzzy programming methods. He is a member of the Fuzzy Society of Japan, the Society of Instrumentation and Automatic Control, the Institute of Electrical Engineers and the Society of KUUKI CHOW and Sanitary Engineers.

Hidenobu Obata received his B.E. in Electrical Engineering systems from Hiroshima University in 1996, and entered the Graduate School in the same university, where he is pursuing his graduate studies. His research interests include multiobjective combinatorial programming using genetic algorithms.