Interactive Decision Making for Fuzzy Multiobjective 0-1 Programs through Genetic Algorithms with Double Strings

Toshihiro Shibano
System Products Division
Shinryo Corporation
Yokohama, 220-81, Japan
shibano@shinryo.super-nova.co.jp

Masatoshi Sakawa
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima, 739, Japan
sakawa@msl.sys.hiroshima-u.ac.jp

Abstract

Multiobjective 0-1 programming problems involving fuzzy numbers are formulated for reflecting the experts' vague or fuzzy understanding of the nature of the parameters in the problem-formulation process. Using the level sets of fuzzy numbers, the corresponding non-fuzzy programming problems together with an extended Pareto optimality concept are introduced. For deriving a satisficing solution for the decision maker from an extended Pareto optimal solution set, an interactive decision making method is presented along with an illustrative numerical example.

1. Introduction

Genetic Algorithms (GAs), initiated by Holland [1], his colleagues and his students at the University of Michigan, a new learning paradigm that models a natural evolution mechanism, have recently received a great deal of attention regarding their potential as optimization techniques for solving combinatorial optimization problems or other difficult problems with nonlinear multimodal function. As we look at recent applications of GAs to optimization problems, especially to various kind of single-objective combinatorial optimization problems and/or to other difficult optimization problems with nonlinear multimodal functions, we can see continuing advances [2, 3, 4].

Recently, as a natural extension of single-objective 0-1 programming problems, Sakawa et al. [5] have formulated multiobjective 0-1 programming problems by assuming that the DM may have a fuzzy goal for each of the objective functions. After eliciting linear membership functions through an interaction with the DM, the fuzzy decision of Bellman and Zadeh [6] is adopted for combining them. In order to derive a compromise solution for the DM by solving the formulated problem, a genetic algorithm with double strings [5] which generates only feasible solutions without using penalty functions for treating the constraints has been proposed. Also, through the combination of the desirable features of both the interactive fuzzy satisficing methods for continuous variables [7] and the genetic algorithm with double strings [5], an interactive fuzzy satisficing method to derive a satisficing solution for the DM to multiobjective 0-1 programming problems has been proposed [8].

Under these circumstances, in this paper, in contrast to the multiobjective 0-1 programming problems discussed thus far, by considering the experts' vague or fuzzy understanding of the nature of the parameters in the problem-formulation process, multiobjective 0-1 programming problems with fuzzy numbers are formulated. Using the α-level sets of fuzzy numbers, an α-Pareto optimality concept is introduced and an interactive decision making method through genetic algorithms with double strings [5, 4, 8] for deriving a satisficing solution for the DM efficiently from an α-Pareto optimal solution set is presented.

2. Problem Formulation

In general, a multiobjective 0-1 programming problem with k conflicting objective functions, $z_i(\boldsymbol{x}) = \boldsymbol{c}_i\boldsymbol{x}$, $i = 1, \dots, k$, is formulated as:

$$\left.\begin{array}{ll} \text{minimize} & (\boldsymbol{c}_1\boldsymbol{x}, \boldsymbol{c}_2\boldsymbol{x}, \dots, \boldsymbol{c}_k\boldsymbol{x})^T \\ \text{subject to} & A\boldsymbol{x} \le \boldsymbol{b} \\ & x_j \in \{0, 1\},\ j = 1, \dots, n \end{array}\right\} \quad (1)$$

where $\boldsymbol{c}_i = (c_{i1}, \dots, c_{in})$, $i = 1, \dots, k$, $\boldsymbol{x} = (x_1, \dots, x_n)^T$, $\boldsymbol{b} = (b_1, \dots, b_m)^T$, and $A = (a_{ij})$ is an $m \times n$ matrix. For simplicity, it is assumed here that each element of A and $\boldsymbol{b}$ is positive.

0-7803-3796-4/97/$10.00©1997IEEE

In practice, however, it would certainly be more appropriate to consider that the possible values of the parameters in the description of the objective functions and the constraints usually involve the ambiguity of the experts' understanding of the real system. For this reason, in this paper, we consider a multiobjective 0-1 programming problem with fuzzy numbers (MO0-1P-FN) formulated as:

$$\left.\begin{array}{ll} \text{minimize} & (\tilde{c}_1 \boldsymbol{x}, \tilde{c}_2 \boldsymbol{x}, \ldots, \tilde{c}_k \boldsymbol{x})^T \\ \text{subject to} & \tilde{A}\boldsymbol{x} \leq \tilde{\boldsymbol{b}} \\ & x_j \in \{0,1\},\ j = 1, \ldots, n \end{array}\right\} \quad (2)$$

where $\tilde{A}$ is an $m \times n$ matrix whose elements are fuzzy numbers, $\tilde{c}$ and $\tilde{\boldsymbol{b}}$ are respectively n and m dimensional vectors whose elements are fuzzy numbers. These fuzzy numbers, reflecting the experts' ambiguous understanding of the nature of the parameters in the problem-formulation process, are assumed to be characterized as fuzzy numbers. Furthermore, for simplicity, it is assumed here that all of the fuzzy numbers in $\tilde{A}$ and $\tilde{\boldsymbol{b}}$ are positive.

Observing that this problem involves fuzzy numbers both in the objective functions and the constraints, it is evident that the notion of Pareto optimality cannot be applied. Thus, it seems essential to extend the notion of usual Pareto optimality in some sense. For that purpose, we first introduce the α-level set of all of the vectors and matrices whose elements are fuzzy numbers.

Definition 1 (α-level set)

The α-level set of fuzzy parameters $\tilde{A}$, $\tilde{\boldsymbol{b}}$ and $\tilde{c}$ is defined as the ordinary set $(\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$ for which the degree of their membership functions exceeds the level α.

Now suppose that the decision maker (DM) decides that the degree of all of the membership functions of the fuzzy numbers involved in the MO0-1P-FN should be greater than or equal to some value α. Then for such a degree α, the MO0-1P-FN can be interpreted as a nonfuzzy multiobjective 0-1 programming (MO0-1P-FN$(A, \boldsymbol{b}, \boldsymbol{c})$) problem which depends on the coefficient vector $(A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$. Observe that there exists an infinite number of such MO0-1P-FN$(A, \boldsymbol{b}, \boldsymbol{c})$ depending on the coefficient vector $(A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$, and the values of $(A, \boldsymbol{b}, \boldsymbol{c})$ are arbitrary for any $(A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$ in the sense that the degree of all of the membership functions for the fuzzy numbers in the MO0-1P-FN exceeds the level α. However, if possible, it would be desirable for the DM to choose $(A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$ in the MO0-1P-FN$(A, \boldsymbol{b}, \boldsymbol{c})$ to minimize the objective functions under the constraints. From such a point of view, for a certain degree α, it seems to be quite natural to have the MO0-1P-FN as the following nonfuzzy α-multiobjective programming (α-MO0-1P) problem:

$$\left.\begin{array}{ll} \text{minimize} & (c_1 \boldsymbol{x}, c_2 \boldsymbol{x}, \ldots, c_k \boldsymbol{x})^T \\ \text{subject to} & A\boldsymbol{x} \leq \boldsymbol{b} \\ & x_j \in \{0,1\},\ j = 1, \ldots, n \\ & (A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha. \end{array}\right\} \quad (3)$$

In the followings, for simplicity, we denote the feasible region satisfying the constraints of the problem (3) with respect to $\boldsymbol{x}$ by $X(A, \boldsymbol{b})$. It should be emphasized here that in the problem (3), the parameters $(A, \boldsymbol{b}, \boldsymbol{c})$ are treated as decision variables rather than constants.

On the basis of the α-level sets of the fuzzy numbers, we can introduce the concept of an α-Pareto optimal solution to the problem (3) as a natural extension of the usual Pareto optimality concept.

Definition 2 (α-Pareto optimal solution)

$\boldsymbol{x}^* \in X(A^*, \boldsymbol{b}^*)$ is said to be an α-Pareto optimal solution to the problem (3) if and only if there does not exist another $\boldsymbol{x} \in X(A, \boldsymbol{b})$, $(A, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{\boldsymbol{b}}, \tilde{c})_\alpha$ such that $c_i \boldsymbol{x} \leq c_i \boldsymbol{x}^*$, $i = 1, \ldots, k$, with strict inequality holding for at least one i, where the corresponding values of parameters $(A^*, \boldsymbol{b}^*, \boldsymbol{c}^*)$ are called α-level optimal parameters.

Observe that α-Pareto optimal solutions and α-level optimal parameters can be obtained through a direct application of the usual scalarizing methods for generating Pareto optimal solutions by regarding the decision variables in the problem (3) as $(\boldsymbol{x}, A, \boldsymbol{b}, \boldsymbol{c})$. However, as can be immediately understood from the definition, in general, α-Pareto optimal solutions consist of an infinite number of points and some kinds of subjective judgment should be added to the quantitative analyses by the DM. Namely, the DM must select a compromise or satisficing solution from an α-Pareto optimal solution set based on a subjective value judgment.

3. Augmented Minimax Problems

To generate a candidate for the satisficing solution which is also α-Pareto optimal, in our interactive decision making method, the DM is asked to specify the degree α of the α-level set and the reference levels of achievement of the objective functions, called reference levels. To be more explicit, for the DM's degree α and reference levels $\bar{z}_i$, $i = 1, ..., k$, the corresponding α-Pareto optimal solution, which is, in the minimax sense, nearest to the requirement or better than that if the reference levels are attainable, is obtained by solving the following minimax problem in an objective function space:

$$\left.\begin{array}{ll}\text{minimize} & \max_{i=1,\ldots,k}\{c_i\boldsymbol{x}-\bar{z}_i\}\\ \text{subject to} & A\boldsymbol{x}\le\boldsymbol{b}\\ & x_j\in\{0,1\},\ j=1,\ldots,n\\ & (A,\boldsymbol{b},\boldsymbol{c})\in(\tilde{A},\tilde{\boldsymbol{b}},\tilde{\boldsymbol{c}})_\alpha.\end{array}\right\}\quad(4)$$

It must be noted here that, for generating α-Pareto optimal solutions by solving the minimax problem, if the uniqueness of the optimal solution $\boldsymbol{x}^*$ is not guaranteed, it is necessary to perform the α-Pareto optimality test of $\boldsymbol{x}^*$. To circumvent the necessity to perform the α-Pareto optimality test in the minimax problems, it is reasonable to use the following augmented minimax problem instead of the minimax problem (4):

$$\left.\begin{array}{ll}\text{minimize} & \max_{i=1,\ldots,k}\left\{(c_i\boldsymbol{x}-\bar{z}_i)+\rho\sum_{i=1}^{k}(c_i\boldsymbol{x}-\bar{z}_i)\right\}\\ \text{subject to} & A\boldsymbol{x}\le\boldsymbol{b}\\ & x_j\in\{0,1\},\ j=1,\ldots,n\\ & (A,\boldsymbol{b},\boldsymbol{c})\in(\tilde{A},\tilde{\boldsymbol{b}},\tilde{\boldsymbol{c}})_\alpha,\end{array}\right\}\quad(5)$$

where ρ is a sufficiently small positive number.

In this formulation, however, constraints are nonlinear because the parameters A, $\boldsymbol{b}$, and $\boldsymbol{c}$ are treated as decision variables. Fortunately, however, from the properties of the α-level set for the vectors and/or matrices of fuzzy numbers, it should be noted here that the feasible regions for A, $\boldsymbol{b}$, $\boldsymbol{c}_i$ can be denoted respectively by the closed intervals $[A^L_\alpha, A^R_\alpha]$, $[\boldsymbol{b}^L_\alpha, \boldsymbol{b}^R_\alpha]$, $[\boldsymbol{c}^L_{i\alpha}, \boldsymbol{c}^R_{i\alpha}]$, where y^L_α or y^R_α represents the left or right extreme point of the α-level set $\tilde{y}_\alpha$. Therefore, we can obtain an optimal solution of the problem (5) by solving the following 0-1 programming problem:

$$\left.\begin{array}{ll}\text{minimize} & \max_{i=1,\ldots,k}\left\{(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x}-\bar{z}_i)+\rho\sum_{i=1}^{k}(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x}-\bar{z}_i)\right\}\\ \text{subject to} & A^L_\alpha\boldsymbol{x}\le\boldsymbol{b}^R_\alpha\\ & x_j\in\{0,1\},\ j=1,\ldots,n.\end{array}\right\}\quad(6)$$

4. GAs with Double Strings

4.1. Coding and decoding

In this paper, as one possible approach to generate only feasible solutions, a double string as is shown in Figure 1 is adopted for representing an individual, where $s_{i(j)}\in\{1,0\}$, $i(j)\in\{1,\ldots,n\}$, and $i(j)\neq i(j')$ for $j\neq j'$. In a double string, regarding $i(j)$ and $s_{i(j)}$ as the index of an element in a solution vector and the value of the element respectively, a string $\mathbf{S}$ can be transformed into a solution $\boldsymbol{x}=(x_1,\ldots,x_n)$ as $x_{i(j)}=s_{i(j)}$, $j=1,\ldots,n$. Unfortunately, however, since this mapping may generate infeasible solutions, we propose the following decoding algorithm for eliminating infeasible solutions. In the algorithm, n, j, $i(j)$, $x_{i(j)}$ and $\boldsymbol{a}_{i(j)}$ denote respectively length of a string, a position in a string, an index of a variable, 0-1 value of a variable with index $i(j)$ decoded from a string and an $i(j)$th column vector of the coefficient matrix A.

$$\begin{array}{rl}\text{index of variable:} & \begin{pmatrix} i(1) & i(2) & \cdots & i(n)\\ s_{i(1)} & s_{i(2)} & \cdots & s_{i(n)}\end{pmatrix}\\ \text{0-1 value}\quad: & \end{array}$$

Figure 1. Double string

Step 1: Set $j=1$, $\boldsymbol{\Sigma}=\mathbf{0}$.
Step 2: If $s_{i(j)}=1$, set $j=j+1$ and go to step 3. Otherwise, i.e., if $s_{i(j)}=0$, set $j=j+1$ and go to step 4.
Step 3: If $\boldsymbol{\Sigma}+\boldsymbol{a}_{s(i)}\le\boldsymbol{b}$, set $x_{s(i)}=1$, $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}+\boldsymbol{a}_{i(j)}$ and go to step 4. Otherwise, set $x_{i(j)}=0$ and go to step 4.
Step 4 If $j>n$, stop and regard $\boldsymbol{x}=(x_1,\ldots,x_n)^T$ as phenotype of the individual represented by the double string. Otherwise, return to step 2.

4.2. Fitness and scaling

It seems quite natural to define the fitness function of each individual $\mathbf{S}$ by $f(\mathbf{S}) = C_{\max} - \max_{i=1,\ldots,k}\left\{(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x}-\bar{z}_i)+\rho\sum_{i=1}^{k}(\boldsymbol{c}^L_{i\alpha}\boldsymbol{x}-\bar{z}_i)\right\}$ where $\mathbf{S}$ and $\boldsymbol{x}$ respectively denote an individual represented by a double string and phenotype of $\mathbf{S}$. Furthermore, using the individual minimum $z_i^{\min}=\boldsymbol{c}_i\boldsymbol{x}^{io}=\min\{\boldsymbol{c}_i\boldsymbol{x}\mid A\boldsymbol{x}\le\boldsymbol{b},\ \boldsymbol{x}\in\{0,1\}^n\}$ together with $z_i^{\mathrm{m}}=\max(\boldsymbol{c}_i\boldsymbol{x}^{1o},\ldots,\boldsymbol{c}_i\boldsymbol{x}^{i-1,o},\boldsymbol{c}_i\boldsymbol{x}^{i+1,o},\ldots,\boldsymbol{c}_i\boldsymbol{x}^{ko})$, $i=1,\ldots,k$, $C_{\max}$ is set as $C_{\max}=\max_{i=1,\ldots,k}|z_i^{\min}-z_i^{\mathrm{m}}|$.

In a reproduction operator based on the ratio of fitness of each individual to the total fitness such as an expected value model, it is frequently pointed out that the probability of selection depends on the relative ratio of fitness of each individual. Thus, several scaling mechanisms have been introduced [2, 3]. In this paper, a linear scaling is adopted. In the linear scaling, fitness f_i of an individual is transformed into f_i' according to $f_i'=a\cdot f_i+b$, where the coefficients a and b are determined so that the mean fitness of the population f_{mean} becomes a fixed point and the maximal fitness of the population $f_{\max}$ becomes twice as large as the mean fitness.

4.3. Reproduction

Up to now, various reproduction methods have been proposed and considered [2, 3]. Using several multiobjective 0-1 programming test problems, the authors have already investigated the performance of each of the six reproduction operators, i.e., ranking selection, elitist ranking selection, expected value selection, elitist expected value selection, roulette wheel selection and elitist roulette wheel selection, and as a result confirmed that elitist expected value selection is relatively efficient [5]. For this reason, as a reproduction operator, elitist expected value selection is adopted here. Elitist expected value selection is a combination of elitism and expected value selection as mentioned below.

Elitism If the fitness of a string in the past populations is larger than that of every string in the current population, preserve this string into the current generation.

Expected value selection For a population consisting of N strings, the expected value of the number of the ith string $\mathbf{S}_i$ in the next population $N_i = (f(\mathbf{S}_i)/\sum f(\mathbf{S}_i)) \times N$ is calculated. Then, the integral part of N_i denotes the deterministic number of the string $\mathbf{S}_i$ preserved in the next population. While, the decimal part of N_i is regarded as probability for one of the string $\mathbf{S}_i$ to survive, i.e., $N - \sum N_i$ strings are determined on the basis of this probability.

4.4. Crossover and mutation

If a single-point or multi-point crossover operator is applied to individuals represented by double strings, an index $i(j)$ in an offspring may take the same number that an index $i(j')$ $(j \neq j')$ takes. Recall that the same violation occurs in solving traveling salesman problems or scheduling problems through genetic algorithms. One possible approach to circumvent such violation, a crossover method called partially matched crossover (PMX) is useful. The PMX was first proposed by Goldberg and Lingle [9] for tackling a blind traveling salesman problem. Although it enables us to generate desirable offsprings without changing the double string structure, it is necessary to revise some points of the procedures. Our revised procedures of the PMX can be illustrated as follows:

Step 1: For two individuals $\mathbf{S}_1$ and $\mathbf{S}_2$ represented by double strings, choose two crossover points.

Step 2: According to the PMX, reorder upper strings of $\mathbf{S}_1$ and $\mathbf{S}_2$ together with the corresponding lower strings which yields $\mathbf{S}'_1$ and $\mathbf{S}'_2$.

Step 3: Exchange lower substrings between two crossover points of $\mathbf{S}'_1$ and $\mathbf{S}'_2$ for obtaining the resulting offsprings $\mathbf{S}''_1$ and $\mathbf{S}''_2$ after the revised PMX for double strings.

It is well recognized that a mutation operator plays a role of a local random search in genetic algorithms. In this paper, for the lower string of a double string, mutation of bit-reverse type is adopted.

4.5. Termination conditions

Incorporating GAs with double strings [5] into interactive fuzzy satisficing methods for multiobjective programs with continuous variables [7], an approximate solution of desirable precision must be obtained in a proper time. For this reason, two parameters $I_{\min}$ and $I_{\max}$, which respectively denote the number of generations to be searched at least and at most, are introduced. Then the following termination conditions are imposed.

Step 1: Set the iteration (generation) index $t = 0$ and the parameter of the termination condition $\varepsilon > 0$.

Step 2: Carry out a series of procedures for search through GAs (reproduction, crossover and mutation).

Step 3: Calculate the mean fitness f_{mean} and the maximal fitness $f_{\max}$ of the population.

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

Step 5 If $t > I_{\max}$, stop. Otherwise, set $t = t + 1$ and return to step 2.

5. Interactive Decision Making Method

We are now ready to propose an interactive algorithm for deriving a satisficing solution for the DM to the MO0-1P-FN by incorporating GAs with double strings [5] into interactive fuzzy satisficing methods for continuous variables [7]. The steps marked with an asterisk involve interaction with the DM.

Step 1*: Ask the DM to select the initial values of α, and the initial reference levels $\bar{z}_i = 1$, $i = 1, \ldots, k$.

Step 2: For the degree α and the reference levels $\bar{z}_i = 1$, $i = 1, \ldots, k$, specified by the DM, solve the corresponding augmented minimax problem through GAs with double strings.

Step 3*: The DM is supplied with the corresponding α-Pareto optimal solution. If the DM is satisfied with the current objective function values of the α-Pareto optimal solution, stop. Otherwise, the DM must update the reference levels and/or the degree α by considering the current values of the objective functions and degree α, and return to Step 2.

It must be observed here that, in this interactive algorithm, the following GAs with double strings are utilized mainly in Step 2. However, observe that, in

Step 1*, for calculating $z_i^{\min}$, $i = 1, \ldots, k$, GAs with double strings are applied.
Step 1: Generate N individuals of length n represented by double strings at random.
Step 2: Evaluate each individual on the basis of phenotype (n dimensional vector) decoded from genotype (string) through fitness and scaling.
Step 3: Apply a reproduction operator (elitist expected value selection).
Step 4: Apply a crossover operator (revised PMX) to individuals according to the probability of crossover p_c.
Step 5: Apply a mutation operator to individuals according to the probability of mutation p_m.
Step 6: Repeat these procedures from step 1 to step 5 until termination conditions described above are satisfied. Then, regard an individual with the maximal fitness as an optimal individual.

It is significant to note here that, through some experiments for solving a relatively simple numerical example, such as two-objective one-dimensional knapsack problems with $20 \sim 50$ variables, where all strings in the initial population are randomly generated at each interaction, it is often observed that the calculated solutions for the updated reference levels are dominated by those calculated before updating. In order to overcome such an undesirable phenomenon implying that the calculated solutions for updated reference levels are not always Pareto optimal, the method of generating an initial population is modified to include the elitism selection [4, 5]. To be more specific, one of the strings in the initial population is equal to the (approximate) optimal solution obtained by the preceding interaction and the remainder consist of $N-1$ strings generated at random.

6. Numerical Example

To demonstrate the feasibility and efficiency of the proposed interactive decision making method, consider a three-objective 0-1 programming problem with 30 variables and 2 constraints involving fuzzy numbers. For simplicity, it is assumed here that all of the membership functions for the fuzzy numbers involved in this example are triangular ones.

The coefficients involved in this numerical example are randomly generated in the following way.

(1) Coefficients a_{ij} are randomly chosen as $0.0 < a_{ij} < 1000.0$. Coefficients c_{1j} are randomly chosen as $-1000.0 < c_{1j} < 0.0$. Half of coefficients c_{2j} are randomly chosen as $-1000.0 < c_{1j} < 0.0$ and the remaining half of coefficients c_{2j} are randomly chosen as $0.0 < c_{2j} < 1000$. Coefficients c_{3j} are randomly chosen as $0.0 < c_{3j} < 1000.0$.

(2) Using randomly chosen parameter P in $[0.25, 0.75]$, b_i is determined as $b_i = P \times \sum_{j=1}^{n} a_{ij}$

(3) 90% of the coefficients a_{ij}, b_i and c_{ij} determined in (1) are assumed to be triangular fuzzy numbers. Multiplying b_{i0} and $c_{ij0} < 0$ by randomly chosen values in $[1.0, 1.1]$, right extreme points b_{i0}^R and left extreme points $c_{ij0}^L < 0$ of their membership functions are determined. Similarly, multiplying a_{ij0} and $c_{ij0} > 0$ by randomly chosen values in $[0.9, 1.0]$, left extreme points a_{ij0}^L and $c_{ij0}^L > 0$ of their membership functions are determined.

For a numerical example generated in this way, at each interaction with the DM, the corresponding augmented minimax problem is solved through 30 runs of GAs with double strings for obtaining an α-Pareto optimal solution. The parameters of GAs are set as, a population size $= 50$, the probability of crossover $p_c = 0.9$, the probability of mutation $p_m = 0.02$, $\varepsilon = 0.05$, $I_{\max} = 500$ and $I_{\min} = 300$. The coefficient ρ of the augmented minimax problem is set as 0.0001. As in shown in Table 1, in this example, the numbers of α-Pareto optimal solutions obtained at each interaction with the DM through 30 runs of GAs are respectively 17, 14, 22, 28 and 30, from which it is observed that relatively preferable results are obtained.

In the whole interaction processes as shown in Table 1, at the first interaction with the DM, for the initial degree $\alpha = 1.0$ specified by the DM, by considering the calculated individual minimum of each objective function, the DM set the initial reference levels as $\bar{z}_1 = -10836.6$, $\bar{z}_2 = -10177$ and $\bar{z}_3 = 0$. For the degree α and the reference levels specified by the DM, the augmented minimax problem is solved and the DM is supplied with the corresponding objective function values of the α-Pareto optimal solution as is shown in Interaction 1 of Table 1. On the basis of such information, since the DM is not satisfied with the current objective function values, the DM updates the reference levels to $\bar{z}_1 = -8836$, $\bar{z}_2 = -10177$ and $\bar{z}_3 = 0$ for improving the satisfaction levels for z_2 and z_3 at the expense of z_1. For the updated reference objective values, the corresponding augmented minimax problem yields the objective function values of the α-Pareto optimal solution as is shown in Interaction 2 of Table 1. The same procedure continues in this manner until the DM is satisfied with the current values of the objective functions. In this example, after two times updating both the reference objective values $(\bar{z}_1, \bar{z}_2, \bar{z}_3)$ and the degree α, at the fifth interaction, the satisficing solution of the DM is derived and the whole interactive processes are summarized in Table 1.

Table 1. Simulation results

		c_1x	c_2x	c_3x	numbers
Interaction 1	obtained solution	−7392.4	−6744.3	3554.3	17
$\bar{z}_1 = -10836.6$		−7459.2	−6783.6	3567.0	4
$\bar{z}_2 = -10177.7$		−7304.1	−7231.7	3653.6	5
$\bar{z}_3 = 0$		−7179.5	−6765.0	3511.7	1
$\alpha = 1.0$		−7176.1	−6529.3	3371.1	2
		−7120.9	−6820.4	3676.7	1
	exact solution	−7392.4	−6744.3	3554.3	
Interaction 2	obtained solution	−6099.9	−7055.8	3145.3	14
$\bar{z}_1 = -8836$		−6166.7	−7095.1	3158.0	3
$\bar{z}_2 = -10177$		−6995.4	−6974.5	3234.3	11
$\bar{z}_3 = 0$		−6928.6	−6935.2	3221.6	2
$\alpha = 1.0$					
	exact solution	−6099.9	−7055.8	3145.3	
Interaction 3	obtained solution	−6336.8	−6423.7	2606.4	22
$\bar{z}_1 = -8836$		−6270.0	−6384.4	2593.7	3
$\bar{z}_2 = -9177$		−6704.1	−6625.5	2860.7	3
$\bar{z}_3 = 0$		−6252.6	−6364.6	2866.1	1
$\alpha = 1.0$		−5998.7	−6115.9	2838.2	1
	exact solution	−6336.8	−6423.7	2606.4	
Interaction 4	obtained solution	−6372.9	−6466.0	2587.3	28
$\bar{z}_1 = -8836$		−6306.1	−6426.4	2574.7	1
$\bar{z}_2 = -9177$		−6047.2	−6340.5	2718.7	1
$\bar{z}_3 = 0$					
$\alpha = 0.9$					
	exact solution	−6372.9	−6466.0	2587.3	
Interaction 5	obtained solution	−6409.0	−6508.3	2568.1	30
$\bar{z}_1 = -8836$					
$\bar{z}_2 = -9177$					
$\bar{z}_3 = 0$					
$\alpha = 0.8$					
	exact solution	−6409.0	−6508.3	2568.1	

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), Third, revised and extended edition (1996).

[4] M. Sakawa and M. Tanaka, *Genetic Algorithms.* Asakura Syoten, Tokyo (1995) (in Japanese).

[5] M. Sakawa, K. Kato, H. Sunada and T. Shibano, Fuzzy programming for multiobjective 0-1 programming problems through revised genetic algorithms, *European Journal of Operational Research* (in press).

[6] R.E. Bellman and L.A. Zadeh, Decision making in a fuzzy environment, *Management Science*, **17**, 141-164 (1970).

[7] M. Sakawa, *Fuzzy Sets and Interactive Multiobjective Optimization.* Plenum Press, New York (1993).

[8] M. Sakawa, K. Kato and T. Shibano, An interactive fuzzy satisficing method for multiobjective multidimensional 0-1 knapsack problems through genetic algorithms, in *1996 IEEE International Conference on Evolutionary Computation*, 243-246 (1996).

[9] D.E. Goldberg and R. Lingle, Alleles, loci, and the traveling salesman problem, in Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Publishers, New Jersey, 154-159 (1985). Third Edition, Kluwer Academic Publishers, Boston (1996).