Scientiae Mathematicae Japonicae Online, **Vol. 10**, (2004), 99–107

AN INTERACTIVE FUZZY SATISFICING METHOD FOR MULTIOBJECTIVE STOCHASTIC INTEGER PROGRAMMING PROBLEMS THROUGH VARIANCE MINIMIZATION MODEL

C. Perkgoz, K. Kato, H. Katagiri and M. Sakawa

Received August 27, 2003

Abstract. In this paper, we focus on multiobjective integer programming problems with random variable coefficients in objective functions and/or constraints. For such multiobjective problems, after reformulation of them on the basis of an expectation optimization model and a variance minimization model for the chance constrained programming, incorporating fuzzy goals of the decision maker for the objective functions, we propose an interactive fuzzy satisficing method to derive a satisficing solution for the decision maker as a fusion of the stochastic programming and the fuzzy one.

1 Introduction In the real world, we often encounter the situation that we have to make a decision under uncertainty because it is difficult to get all the information needed for decision making. For such decision making problems involving uncertainty, stochastic programming and fuzzy programming are two typical approaches.

Stochastic programming techniques have been developed in various ways as an optimization method based on the probability theory (e.g., two stage problem by G.B. Dantzig [2], chance constrained programming by A. Charnes and W.W. Cooper [1]). In particular, for multiobjective linear programming problems involving random variable coefficients, Stancu-Minasian [6] handled the minimum risk approach, while Teghem et al. [12] presented an interactive method.

On the other hand, fuzzy mathematical programming representing the ambiguity in a decision making situation by fuzzy concepts has attracted attention of many researchers. Fuzzy multiobjective linear programming have been developed by numerous researchers, and a lot of successful applications have been appearing [7].

As a hybrid of the stochastic approach and fuzzy approach, in particular, Sakawa et al. [11] proposed an interactive fuzzy satisficing method based on the expectation optimization model for multiobjective stochastic linear programming problems. Furthermore, they have extended it to the simple recourse model [10] and the variance minimization model [9]. However, in these literatures, they dealt with only the case that decision variables are continuous.

In this paper, focusing on multiobjective stochastic integer programming problems, we present an interactive fuzzy satisficing method based on an expectation optimization model and a variance minimization model. In order to consider the nonlinearity of problems solved in the interactive fuzzy satisficing method and to cope with large-scale problems, we adopt genetic algorithms as a solution method.

2003 *Mathematics Subject Classification.* Primary 90C15, 90C59; Secondary 90C10, 90C29.
Key words and phrases. multiobjective integer programming problem, stochastic programming, variance minimization, genetic algorithm.

2 Multiobjective stochastic integer programming problems In this paper, we focus on the following multiobjective stochastic integer programming problem in which parameters in objective functions and the right-hand side of constraints are random variables.

$$\left.\begin{array}{rl} \text{minimize} & z_l(\boldsymbol{x},\omega) = \boldsymbol{c}_l(\omega)\boldsymbol{x},\ \ l=1,\ldots,k \\ \text{subject to} & A\boldsymbol{x} \le \boldsymbol{b}(\omega) \\ & x_j \in \{0,1,\ldots,\nu_j\},\ \ j=1,\ldots,n \end{array}\right\} \tag{1}$$

where $\boldsymbol{x}$ is an n dimensional integer decision variable column vector, and A is an $m \times n$ coefficient matrix. $\boldsymbol{c}_l(\omega)$, $l = 1,\ldots,k$ are n dimensional random variable row vectors with mean $\bar{\boldsymbol{c}}_l$ and covariance matrix $V_l = (v^l_{jh}) = (\text{Cov}\{c_{lj}(\omega), c_{lh}(\omega)\})$, $j = 1,\ldots,n$, $h = 1,\ldots,n$, and $b_i(\omega)$, $i = 1,\ldots,m$ are random variables which are independent of each other, and the distribution function of each of them are also assumed to be continuous and strictly increasing.

For instance, there may exist a project selection problem to optimize not only gross profits but total labor costs under constraints such as the usable amount of manpower is limited, where profit coefficients of projects, labor cost coefficients of projects and maximal usable amount of manpower depend on business conditions.

3 Chance constrained programming Since (1) contains random variable coefficients, solution methods for ordinary mathematical programming problems cannot be directly applied. Consequently, we deal with the constraints in (1) as chance constrained conditions which mean that the constraints need to be satisfied with a certain probability (satisficing level) and over. Replacing the constraints in (1) by chance constrained conditions with satisficing levels β_i, the problem can can be converted as:

$$\left.\begin{array}{rl} \text{minimize} & z_l(\boldsymbol{x},\omega) = \boldsymbol{c}_l(\omega)\boldsymbol{x},\ \ l=1,\ldots,k \\ \text{subject to} & \Pr\{\boldsymbol{a}_i\boldsymbol{x} \le b_i(\omega)\} \ge \beta_i,\ \ i=1,\ldots,m \\ & x_j \in \{0,1,\ldots,\nu_j\},\ \ j=1,\ldots,n \end{array}\right\} \tag{2}$$

where $\boldsymbol{a}_i$ is the ith row vector of A and $b_i(\omega)$ is the ith element of $\boldsymbol{b}(\omega)$.

Using continuous and strictly increasing distribution functions $F_i(r) = \Pr\{b_i(\omega) \le r\}$ of random variables $b_i(\omega)$, $i = 1,\ldots,m$, the i th constraint in (2) can be rewritten as:

$$\begin{aligned} \Pr\{\boldsymbol{a}_i\boldsymbol{x} \le b_i(\omega)\} \ge \beta_i &\Leftrightarrow 1 - \Pr\{b_i(\omega) \le \boldsymbol{a}_i\boldsymbol{x}\} \ge \beta_i \\ &\Leftrightarrow 1 - F_i(\boldsymbol{a}_i\boldsymbol{x}) \ge \beta_i \\ &\Leftrightarrow F_i(\boldsymbol{a}_i\boldsymbol{x}) \le 1 - \beta_i \\ &\Leftrightarrow \boldsymbol{a}_i\boldsymbol{x} \le F_i^{-1}(1-\beta_i). \end{aligned} \tag{3}$$

Letting $\hat{b}_i = F_i^{-1}(1-\beta_i)$ in (3), (2) can be transformed into the following problem:

$$\left.\begin{array}{rl} \text{minimize} & z_l(\boldsymbol{x},\omega) = \boldsymbol{c}_l(\omega)\boldsymbol{x},\ \ l=1,\ldots,k \\ \text{subject to} & A\boldsymbol{x} \le \hat{\boldsymbol{b}} \\ & x_j \in \{0,1,\ldots,\nu_j\},\ \ j=1,\ldots,n \end{array}\right\} \tag{4}$$

where $\hat{\boldsymbol{b}} = (\hat{b}_1,\ldots,\hat{b}_m)^T$, and we denote the feasible region of (4) by X.

4 Expectation optimization model Let us consider the expectation optimization model, which is the simplest one in the chance constrained programming. In the model, we substitute the minimization of expectations of objective functions for the minimization of objective functions in (4). Then, the problem can be rewritten as:

$$\underset{\boldsymbol{x}\in X}{\text{minimize}}\quad \bar{z}_l(\boldsymbol{x}) = \text{E}\{z_l(\boldsymbol{x},\omega)\} = \bar{\boldsymbol{c}}_l\boldsymbol{x},\ \ l=1,\ldots,k \tag{5}$$

Since the above problem is an ordinary multiobjective integer programming problem, an interactive fuzzy satisficing method for multiobjective integer programming problems using genetic algorithms [8] is directly applicable in order to obtain a satisficing solution for the decision maker.

5 Variance minimization model Since objective functions regarded as random variables in (4) are reduced to their expectations in the expectation optimization model, the requirement of the decision maker for risk is not reflected in the obtained solution. ¿From this viewpoint, in this section, we consider the variance minimization model. In the model, we substitute the minimization of variances of objective functions $z_l(\boldsymbol{x}, \omega)$, $l = 1, \ldots, k$ in (4) for the minimization of them. Then, the problem can be rewritten as:

$$\text{(6)} \qquad \underset{\boldsymbol{x} \in X}{\text{minimize}} \quad z_l'(\boldsymbol{x}) = \text{Var}\{z_l(\boldsymbol{x}, \omega)\} = \boldsymbol{x}^T V_l \boldsymbol{x}, \ \ l = 1, \ldots, k$$

Since (6) is a multiobjective quadratic integer programming problem, a satisficing solution for the decision maker can be obtained through an interactive fuzzy satisficing method for multiobjective integer programming problems using genetic algorithms [8] with some modifications.

Using the variance minimization model, the obtained solution might be too bad in the sense of the expectation of objective functions, while it accomplishes the minimization in the sense of the variance. In order to take the requirement of the decision maker for expectations of objective functions into account, we consider the following revised variance minimization model incorporating constraints that the expectation of each objective function, $\bar{z}_l(\boldsymbol{x})$ must be less than or equal to a certain permissible level γ_l, $l = 1, \ldots, k$.

$$\text{(7)} \qquad \left.\begin{array}{rl} \text{minimize} & z_1'(\boldsymbol{x}) = \text{Var}\{z_1(\boldsymbol{x}, \omega)\} = \boldsymbol{x}^T V_1 \boldsymbol{x}, \ \ l = 1, \ldots, k \\ \text{subject to} & A\boldsymbol{x} \leq \hat{\boldsymbol{b}} \\ & \bar{C}\boldsymbol{x} \leq \boldsymbol{\gamma} \\ & x_j \in \{0, 1, \ldots, \nu_j\}, \ \ j = 1, \ldots, n \end{array}\right\}$$

where $\bar{C} = (\bar{\boldsymbol{c}}_1^T, \ldots, \bar{\boldsymbol{c}}_k^T)^T$ and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)^T$, and we denote the feasible region of (7) by X'.

Since the above problem is a multiobjective quadratic integer programming problem like (6), we can apply an interactive fuzzy satisficing method for multiobjective integer programming problems using genetic algorithms [8] with some modifications.

An interactive fuzzy satisficing method for the revised variation optimization model

Step 1: Ask the decision maker to subjectively determine satisficing levels β_i, $i = 1, \ldots, m$ for constraints in (2).

Step 2: Calculate the minimum $\bar{z}_l^{\min}$ and the maximum $\bar{z}_l^{\max}$ of the expectation of each objective function in (4), $\text{E}[z_l(\boldsymbol{x}, \omega)] = \bar{z}_l(\boldsymbol{x})$, $l = 1, \ldots, k$ by solving the following problems

$$\text{(8)} \qquad \underset{\boldsymbol{x} \in X}{\text{minimize}} \ \ \bar{z}_l(\boldsymbol{x}) = \bar{\boldsymbol{c}}_l \boldsymbol{x}, \ \ l = 1, \ldots, k,$$

$$\text{(9)} \qquad \underset{\boldsymbol{x} \in X}{\text{maximize}} \ \ \bar{z}_l(\boldsymbol{x}) = \bar{\boldsymbol{c}}_l \boldsymbol{x}, \ \ l = 1, \ldots, k.$$

Then, ask the decision maker to subjectively specify permissible levels γ_l, $l = 1, \ldots, k$ for objective functions in consideration of $\bar{z}_l^{\min}$ and $\bar{z}_l^{\max}$. Note that problems (8) and (9) are ordinary single-objective linear integer programming problems, a genetic algorithm with double strings based on using linear programming relaxation based on reference solution updating (GADSLPRRSU) [8] is directly applicable.

Step 3: Calculate the individual minimum $z'_{l,\min}$ of $z'_l(\boldsymbol{x})$, $l = 1, \ldots, k$ in (7) by solving problems to minimize the variance of each objective function in (6) under chance constraints corresponding to those satisficing levels by GADSLPRRSU [8] with some modifications.

$$\text{(10)} \qquad \underset{\boldsymbol{x} \in X'}{\text{minimize}} \;\; z'_l(\boldsymbol{x}) = \boldsymbol{x}^T V_l \boldsymbol{x}, \;\; l = 1, \ldots, k$$

Step 4: Ask the decision maker to subjectively specify membership functions $\mu_l(z'_l(\boldsymbol{x}))$, $l = 1, \ldots, k$ which quantify fuzzy goals for objective functions based on minimal values $\bar{z}_{l,\min}$ calculated in step 2.

Step 5: Ask the decision maker to set initial reference membership levels $\bar{\mu}_l$, $l = 1, \ldots, k$ (usually, $\bar{\mu}_l = 1$, $l = 1, \ldots, k$).

Step 6: Solve an augmented minimax problem (11) for current reference membership levels $\bar{\mu}_l$, $l = 1, \ldots, k$ by GADSLPRRSU [8] with some modifications.

$$\text{(11)} \qquad \underset{\boldsymbol{x} \in X'}{\text{minimize}} \quad \max_{l=1,\ldots,k} \{(\bar{\mu}_l - \mu_l(z'_l(\boldsymbol{x}))) + \rho \sum_{i=1}^{k} (\bar{\mu}_i - \mu_i(z'_i(\boldsymbol{x})))\}$$

Step 7: If the decision maker is satisfied with the solution to (11), the interactive process is terminated. Otherwise, ask the decision maker to update reference membership levels $\bar{\mu}_l$, $l = 1, \ldots, k$ in consideration of current membership function values or objective function values, and go to step 5.

6 Genetic Algorithm with Double Strings Using Linear Programming Relaxation Based on Reference Solution Updating (GADSLPRRSU) Since problems (8), (9), (10), (11) solved in the interactive fuzzy satisficing method mentioned above are 0-1 programming problems, it is difficult to solve them for large-scale problems by enumeration-based method. Thereby, some efficient approximate solution method is required for practical use. As approximate solution methods for discrete optimization problems, because of versatility and ease of implementation, metaheuristics such as genetic algorithms, simulated annealing and tabu search are thought to be dominant. Because there was reported in [4] that genetic algorithms are applied to multiobjective programming problems most frequently among metaheuristics, and M. Sakawa et al. showed the effectiveness of a genetic algorithm with double strings based on using linear programming relaxation based on reference solution updating (GADSLPRRSU) [8] for linear integer programming problems defined as (12), we adopted GADSLPRRSU as a solution method in the interactive fuzzy satisficing method.

$$\text{(12)} \qquad \left.\begin{array}{rl} \text{minimize} & \boldsymbol{c}\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} \leq \boldsymbol{b} \\ & x_j \in \{0, 1, \ldots, \nu_j\}, \;\; j = 1, \ldots, n \end{array}\right\}$$

where $A = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n]$ is an $m \times n$ coefficient matrix, $\boldsymbol{b} = (b_1, \ldots, b_m)^T$ is an m dimensional column vector and $\boldsymbol{c} = (c_1, \ldots, c_n)$ is an n dimensional row vector.

Individual **S** :

$s(1)$	$s(2)$	$\cdots$	$s(n)$
$g_{s(1)}$	$g_{s(2)}$	$\cdots$	$g_{s(n)}$

Figure 1: Double string

6.1 Individual Representation An individual representation by double strings shown in Fig. 1 is adopted in GADSLPRRSU. In the figure, each of $s(j)$, $j = 1, \dots, n$ is the index of an element in a solution vector and each of $g_{s(j)} \in \{0, 1, \dots, \nu_{s(j)}\}$, $j = 1, \dots, n$ is the value of the element, respectively.

6.2 Decoding Algorithm In [8], a decoding algorithm of double strings for linear integer programming problems, which generates a feasible solution from a double string. is constructed as follows. In the algorithm, a feasible solution $\boldsymbol{x}^*$, called a reference solution, is used as the origin of decoding.

Decoding algorithm using a reference solution

In this algorithm, it is assumed that a feasible solution $\boldsymbol{x}^*$ to (12) is obtained in advance. Let n and N be the number of variables and the number of individuals in the population, respectively. Also, $\boldsymbol{b}^+$ means a column vector of positive right-hand side constants, and the corresponding coefficient matrix is denoted by $A^+ = (\boldsymbol{p}_1^+, \dots, \boldsymbol{p}_n^+)$.

Step 1: Let $j := 1$ and $\mathbf{psum} := \mathbf{0}$.

Step 2: If $g_{s(j)} = 0$, set $q_{s(j)} := 0$ and $j := j + 1$, and go to step 4. If $g_{s(j)} \neq 0$, go to step 3.

Step 3: If $\mathbf{psum} + \boldsymbol{p}^+_{s(j)} \cdot g_{s(j)} \leq \boldsymbol{b}^+$, set $q_{s(j)} := g_{s(j)}$, $\mathbf{psum} := \mathbf{psum} + \boldsymbol{p}^+_{s(j)} \cdot g_{s(j)}$ and $j := j + 1$, and go to step 4. Otherwise, set $q_{s(j)} := 0$ and $j := j + 1$, and go to step 4.

Step 4: If $j > n$, go to step 5. If $j \leq n$, go to step 2.

Step 5: Let $j := 1$, $l := 0$ and $\mathbf{sum} := \mathbf{0}$.

Step 6: If $g_{s(j)} = 0$, set $j := j + 1$ and go to step 8. If $g_{s(j)} \neq 0$, set $\mathbf{sum} := \mathbf{sum} + \boldsymbol{p}_{s(j)} \cdot g_{s(j)}$ and go to step 7.

Step 7: If $\mathbf{sum} \leq \boldsymbol{b}$, set $l := j$, $j := j + 1$, and go to step 8. Otherwise, set $j := j + 1$ and go to step 8.

Step 8: If $j > n$, go to step 9. If $j \leq n$, go to step 6.

Step 9: If $l > 0$, go to step 10. If not, go to step 11.

Step 10: For $x_{s(j)}$ satisfying $1 \leq j \leq l$, let $x_{s(j)} := g_{s(j)}$. For $x_{s(j)}$ satisfying $l + 1 \leq j \leq n$, let $x_{s(j)} := 0$, and stop.

Step 11: Let $\mathbf{sum} := \sum_{k=1}^{n} \boldsymbol{p}_{s(k)} \cdot x^*_{s(k)}$ and $j := 1$.

Step 12: If $g_{s(j)} = x^*_{s(j)}$, let $x_{s(j)} := g_{s(j)}$ and $j := j + 1$, and go to step 16. Otherwise, go to step 13.

Step 13: If $\mathbf{sum} - \boldsymbol{p}_{s(j)} \cdot x^*_{s(j)} + \boldsymbol{p}_{s(j)} \cdot g_{s(j)} \leq \boldsymbol{b}$, set $\mathbf{sum} := \mathbf{sum} - \boldsymbol{p}_{s(j)} \cdot x^*_{s(j)} + \boldsymbol{p}_{s(j)} \cdot g_{s(j)}$ and $x_{s(j)} := g_{s(j)}$, and go to step 16. Otherwise, go to step 14.

Step 14: Let $t_{s(j)} := \lfloor 0.5 \cdot (x^*_{s(j)} + g_{s(j)}) \rfloor$ and go to step 15.

Step 15: If $\mathbf{sum} - \boldsymbol{p}_{s(j)} \cdot x^*_{s(j)} + \boldsymbol{p}_{s(j)} \cdot t_{s(j)} \leq \boldsymbol{b}$, set $\mathbf{sum} := \mathbf{sum} - \boldsymbol{p}_{s(j)} \cdot x^*_{s(j)} + \boldsymbol{p}_{s(j)} \cdot t_{s(j)}$, $g_{s(j)} := t_{s(j)}$ and $x_{s(j)} := t_{s(j)}$, and go to step 16. Otherwise, set $x_{s(j)} := x^*_{s(j)}$ and go to step 16.

Step 16: If $j > n$, stop. Otherwise, return to step 12.

The proposed decoding algorithm can make the correspondence of any individual to a feasible solution. Because solutions obtained the decoding algorithm using a reference solution tend to concentrate around the reference solution, the reference solution updating procedure is adopted.

6.3 Usage of Linear Programming Relaxation In order to find an approximate optimal solution with high accuracy in reasonable time, we need some schemes such as the restriction of the search space to a promising region, the generation of individuals near the optimal solution and so forth. ¿From the point of view, the information about an optimal solution to the corresponding linear programming relaxation problem

$$
\left.\begin{array}{rl}
\text{minimize} & \boldsymbol{c}\boldsymbol{x} \\
\text{subject to} & A\boldsymbol{x} \leq \boldsymbol{b} \\
& 0 \leq x_j \leq \nu_j, \;\; j = 1, \ldots, n
\end{array}\right\} \tag{13}
$$

is used in the generation of the initial population and the mutation [8].

6.4 Computational procedures of GADSLPRRSU

Step 0: Determine values of the parameters used in the genetic algorithm.

Step 1: Generate the initial population consisting of N individuals based on the information of a solution to the continuous relaxation problem (13).

Step 2: Decode each individual (genotype) in the current population and calculate its fitness based on the corresponding solution (phenotype).

Step 3: If the termination condition is fulfilled, stop. Otherwise, let $t := t + 1$ and go to step 4.

Step 4: Apply reproduction operator using elitist expected value selection after linear scaling.

Step 5: Apply crossover operator, called PMX (Partially Matched Crossover) for double string.

Step 6: Apply mutation based on the information of a solution to the continuous relaxation problem (13).

Step 7: Apply inversion operator. Go to step 2.

Since most of procedures are the same as those of GADSLPRRSU [8] except relaxation problems and fitness function, we omit the details of them.

6.5 Some modifications With respect to relaxation problems in (10) and (11), since those objective functions are not linear, we need to solve corresponding relaxation problems by appropriate nonlinear programming techniques. In this paper, we use GENOCOPIII proposed by Z. Michalewicz et al. [5].

Furthermore, fitness functions for problems (8), (9), (10), (11) are revised as follows.

Problem (8)

$$f(\mathbf{s}) = \frac{\bar{\boldsymbol{c}}_l \boldsymbol{x} - \sum_{j \in J^+_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j}{\sum_{j \in J^-_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j - \sum_{j \in J^+_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j} \tag{14}$$

Problem (9)

$$f(\mathbf{s}) = \frac{\bar{\boldsymbol{c}}_l \boldsymbol{x} - \sum_{j \in J^-_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j}{\sum_{j \in J^+_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j - \sum_{j \in J^-_{\bar{\boldsymbol{c}}_l}} \bar{c}_{lj} \cdot \nu_j} \tag{15}$$

Problem (10)

$$f(\mathbf{s}) = \frac{\boldsymbol{x}^T V_l \boldsymbol{x} - \sum_{i,j \in IJ^+_{V_l}} v^l_{ij} \cdot \nu_i \nu_j}{- \sum_{i,j \in IJ^+_{V_l}} v_{ij} \cdot \nu_i \nu_j} \tag{16}$$

Problem (11)

$$f(\mathbf{s}) = 1 - \max_{l=1,\dots,k} \{\bar{\mu}_l - \mu_l(z'_l(\boldsymbol{x}))\} \tag{17}$$

where $J^+_{\bar{\mathbf{c}}^l_i} = \{j \mid \bar{c}^l_{ij} > 0, 1 \le j \le n\}$, $J^-_{\bar{\mathbf{c}}^l_i} = \{j \mid \bar{c}^l_{ij} < 0, 1 \le j \le n\}$, $IJ^-_{V_l} = \{i, j \mid v^l_{ij} < 0, 1 \le i, j \le n\}$, and v^l_{ij} is the (i, j) element of V_l.

7 Numerical experiment To demonstrate the feasibility of the proposed method, consider the following multiobjective linear programming problem involving random variable coefficients (3 objectives, 100 variables, 10 constraints).

$$\left.\begin{array}{rl} \text{minimize} & z_l(\boldsymbol{x}, \omega) = \boldsymbol{c}_l(\omega)\boldsymbol{x}, \;\; l = 1, 2, 3 \\ \text{subject to} & \boldsymbol{a}_i \boldsymbol{x} \le b_i(\omega), \;\; i = 1, \dots, 10 \\ & x_j \in \{0, 1, \dots, \nu_j\}, \;\; j = 1, \dots, 100 \end{array}\right\} \tag{18}$$

In this problem, each element of A in the numerical example was selected at random from $\{-10, 10\}$. $b_1(\omega), \dots, b_{10}(\omega)$ are Gaussian random variables $N(1868, 40^2)$, $N(1244, 30^2)$, $N(2292, 50^2)$, $N(656, 20^2)$, $N(2056, 10^2)$, $N(1156, 40^2)$, $N(632, 30^2)$, $N(1968, 50^2)$, $N(1260, 20^2)$, $N(516, 10^2)$, where $N(m, s^2)$ stands for a Gaussian random variable having mean m and variance s^2, and mean values are determined by the following equation

$$\bar{b}_i = \sum_{j \in J^-_{\boldsymbol{a}_i}} a_{ij} + \delta \times \left(\sum_{j \in J^+_{\boldsymbol{a}_i}} a_{ij} - \sum_{j \in J^-_{\boldsymbol{a}_i}} a_{ij} \right)$$

where the positive constant $\delta \in [0,1]$ denotes the degree of looseness of constraints. To be more specific, the constraints become looser as δ increases to 1, while the constraints become tighter as it decreases to 0. In this paper, we set $\gamma = 0.6$. On the other hand, $\boldsymbol{c}_1(\omega)$, $\boldsymbol{c}_2(\omega)$ and $\boldsymbol{c}_3(\omega)$ are vectors of Gaussian random variables with finite mean vectors elements of which are determined randomly from $\{0,16\}$, $\{-8,8\}$ and $\{-16,0\}$, and positive definite covariance matrices.

First, according to step 1 of the interactive fuzzy satisficing method, the decision maker determines the satisficing levels β_i, $i = 1,\ldots,10$ for each of the constraints in (18). The hypothetical decision maker in this example specifies the satisficing levels as $(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5,\beta_6,\beta_7,\beta_8,\beta_9,\beta_{10})^T = (0.85,0.95,0.90,0.85,0.95,0.90,0.85,0.90,0.85,0.95)^T$.

Second, according to step 2, the individual minimum $\bar{z}_{l,\min}$ and maximum $\bar{z}_{l,\max}$ of objective functions $\mathrm{E}[z_l(\boldsymbol{x},\omega)]$, $l = 1,\ldots,3$, are calculated under the chance constrained conditions corresponding to the satisficing levels. Each value is obtained by solving (8) and (9) through GADSLPRRSU as $\bar{z}_{1,\min} = 0$, $\bar{z}_{1,\max} = 16505$, $\bar{z}_{2,\min} = -4209$, $\bar{z}_{2,\max} = 4220$, $\bar{z}_{3,\min} = -14100$, $\bar{z}_{3,\max} = 0$. Based on these values, the decision maker subjectively specifies permissible levels $\gamma_1 = 8000$, $\gamma_2 = 0$, $\gamma_3 = -7000$.

Third, according to step 3, the individual minimum $z'_{l,\min}$ of $z'_l(\boldsymbol{x})$, $l = 1,\ldots,3$ in (7) are calculated by using the modified GADSLPRRSU as $z'_{1,\min} = 12938.2$, $z'_{2,\min} = 460.2$, $z'_{3,\min} = 6617.0$.

In step 4, The decision maker subjectively determines membership functions to quantify fuzzy goals for objective functions. Here, the following linear membership function is adopted.

$$\mu'_l(z'_l(\boldsymbol{x})) = \frac{z'_l(\boldsymbol{x}) - z'_{l,0}}{z'_{l,1} - z'_{l,0}}$$

According to step 5, the decision maker specifies the initial reference membership levels $(\bar{\mu}_1,\bar{\mu}_2,\bar{\mu}_3)$ as $(1.00,1.00,1.00)$.

Next, according to step 6, in order to find the optimal solution to the augmented minimax problem (11) for $(\bar{\mu}_1,\bar{\mu}_2,\bar{\mu}_3) = (1.00,1.00,1.00)$, the modified GADSLPRRSU is used. The obtained solution is shown at the second column in Table 1. According to step 7, the hypothetical decision maker cannot be satisfied with this solution, particularly, he wants to improve $\mu_2(\cdot)$, $\mu_3(\cdot)$ at the sacrifice of $\mu_1(\cdot)$. Thus, the decision maker updates the reference membership levels to (0.90, 1.00, 1.00) and return to step 6. By repetition of such interaction with the decision maker, in this example, a satisficing solution is obtained at the third interaction.

8 Conclusion In this paper, we focused on multiobjective integer programming problems involving random variable coefficients. After the formulation of the expectation optimization model and the variance minimization model, we introduced fuzzy goals to consider the ambiguous or fuzzy judgments of the decision maker and proposed an interactive fuzzy satisficing method to derive a satisficing solution for the decision maker. An illustrative numerical example demonstrated the feasibility of the proposed method. Extensions of the proposed method to other chance constrained condition problems such as the probability maximization model, the fractile criterion optimization model and so forth are now under investigation and will be reported elsewhere.

References

[1] A. Charnes and W.W. Cooper, Chance constrained programming, Management Science, Vol. 6, pp. 73-79, 1959.

Table 1: Process of interaction.

Interaction	1st	2nd	3rd
$\bar{\mu}_1$	1.000	0.900	0.900
$\bar{\mu}_2$	1.000	1.000	0.950
$\bar{\mu}_3$	1.000	1.000	1.000
$\mu_1(z'_1(\boldsymbol{x}))$	0.7212	0.6524	0.6741
$\mu_2(z'_2(\boldsymbol{x}))$	0.7207	0.7542	0.7325
$\mu_3(z'_3(\boldsymbol{x}))$	0.7211	0.7325	0.7775
$\bar{z}_1(\boldsymbol{x})$	7177	-88	-7000
$\bar{z}_2(\boldsymbol{x})$	7563	-35	-7000
$\bar{z}_3(\boldsymbol{x})$	7321	-3	-7000

[2] G.B. Dantzig, Linear programming under uncertainty, Management Science, Vol. 1, pp. 3-4, 1955.

[3] H. Ishii, Stochastic optimization, M. Iri and H. Konno (eds.): Applications of Mathematical Programming (Theoretical Topics), Sangyo Tosho, 1982, pp. 1-40.

[4] D.F. Jones, S.K. Mirrazavi and M. Tamiz, Multi-objective meta-heuristics: An overview of the current state-of-the-art, *European Journal of Operational Research*, Vol. 137, No. 1, pp. 1–9, 2002.

[5] Z. Michalewicz and G. Nazhiyath, GenocopIII: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints, Proceedings of the Second IEEE International Conference on Evolutionary Computation, pp. 647–651, 1995.

[6] I.M. Stancu-Minasian, Overview of different approaches for solving stochastic programming problems with multiple objective functions, In R. Slowinski and J. Teghem (eds.): Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty, Kluwer Academic Publishers, 1990, pp. 71-101.

[7] M. Sakawa, Fuzzy Sets and Interactive Multiobjective Optimization, Plenum Press, 1993.

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

[9] M. Sakawa, K. Kato and H. Katagiri, An interactive fuzzy satisficing method through a variance minimization model for multiobjective linear programming problems involving random variable coefficients, KES'2002 Part 2, pp. 1222-1226, 2002.

[10] M. Sakawa, K. Kato, I. Nishizaki and K. Wasada, An interactive fuzzy satisficing method for multiobjective stochastic linear programs through simple recourse model, Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp. 59-64, 2001.

[11] M. Sakawa, K. Kato, I. Nishizaki and M. Yoshioka, Interactive decision making for fuzzy multiobjective linear programming problems involving random variable coefficients, Proceedings of The Fourth Asian Fuzzy Systems Symposium, Vol. 1, pp. 392-397, 2000.

[12] J. Teghem Jr., D. Dufrane, M. Thauvoye and P. Kunsch, STRANGE: an interactive method for multi-objective linear programming under uncertainty, European Journal of Operational Research, Vol. 26, pp. 65-82, 1986.

Graduate School of Engineering, Hiroshima University
1-4-1, Kagamiyama, Higashi-Hiroshima, 739-8527 Japan
Cahit Perkgoz
TEL: 0824-24-7693 FAX: 0824-22-7195
E-mail: cperkgoz@msl.sys.hiroshima-u.ac.jp