1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia. Editors, L.C. Jain and R.K.Jain

Large scale fuzzy multiobjective 0-1 programs through genetic algorithms with decomposition procedures

†Kosuke KATO and ‡Masatoshi SAKAWA
Department of Industrial and Systems Engineering, Hiroshima University, JAPAN

***Abstract* — In this paper, we focus on large scale fuzzy multiobjective 0-1 programming problems. For fuzzy multiobjective programming problems, we introduce extended concepts of the ordinary Pareto optimal solution. The problem to find a satisficing solution for the decision maker from the extended Pareto optimal solution set is a kind of combinatorial optimization problem, so we apply genetic algorithms to solve it. Since the problems considered in this paper are large scale and a large scale problem has often a special structure called block angular structure, we apply genetic algorithms revised by the authors to utilize the special structure.**

***KeyWords* — Large scale 0-1 programs, genetic algorithms, triple strings, decomposition procedures**

1. Introduction

In general, it is difficult to solve large scale problems exactly owing to various kinds of restrictions. However, most of large scale problems arising in applications have special structures that can be exploited. Block angular structure to constraints is one of well-known special structures. For linear and nonlinear programming problems with block angular structure some decomposition techniques have been proposed [1], while neither exact algorithms nor decomposition ones in practice are not established for large scale programming problems with discrete variables.

In recent years, many researchers have took interest in genetic algorithms (GA) as an optimization technique and various researches about the application of GA to mathematical programming problems have been reported [2].

For fuzzy multiobjective 0-1 programming problems, M. Sakawa et al. proposed a genetic algorithm using double string representation where each individual (genotype) transformed into a feasible solution (phenotype) and showed its efficiency [3], [4]. For the purpose of deriving solutions efficiently by utilizing the special structure of the problem, K. Kato et al. proposed a genetic algorithm using triple string representation as the extension of double string representation [5], [6]. In these papers [5], [6], however, the proposed method was applied to a few problems of small size.

Under these circumstances, in the present paper, we investigate the performance of a genetic algorithm with decomposition procedures using triple string representation for large scale fuzzy multiobjective 0-1 programming problems with block angular structure.

2. Problem formulation and solution concept

Let us consider a block angular multiobjective 0-1 programming problem as follows:

$$\left.\begin{array}{lrl} \text{minimize} & c_1 x = & c_{11}x_1 + \cdots + c_{1p}x_p \\ \vdots & & \vdots \\ \text{minimize} & c_k x = & c_{k1}x_1 + \cdots + c_{kp}x_p \\ \text{subject to} & & A_1x_1 + \cdots + A_px_p \le b_0 \\ & & B_1x_1 \qquad\qquad\quad \le b_1 \\ & & \qquad \ddots \qquad\qquad \vdots \\ & & \qquad\qquad\quad B_px_p \le b_p \\ & & x_j \in \{0,1\}^{n_j},\ j = 1,\ldots,p \end{array}\right\} \quad (1)$$

where c_{ij}'s, $i = 1,\ldots,k$, $j = 1,\ldots,p$, are n_j dimensional cost factor row vectors, x_j's, $j = 1,\ldots,p$, are n_j dimensional column vectors of 0-1 decision variables, $A_1x_1 + \cdots + A_px_p \le b_0$ denotes m_0 dimensional coupling constraints, A_j's, $j = 1,\ldots,p$, are $m_0 \times n_j$ coefficient matrices. The inequalities $B_jx_j \le b_j$, $j = 1,\ldots,p$, are m_j dimensional block constraints with respect to x_j and B_j's, $j = 1,\ldots,p$, are $m_j \times n_j$ coefficient matrices. In this paper, it is assumed that each element of A_j, B_j and b_j is positive respectively. For simplicity in notation, define the following vectors and matrices.

$$c_i = (c_{i1},\ldots,c_{ip}),\ \ c = (c_1,\ldots,c_k),$$
$$x = (x_1,\ldots,x_p),\ \ A = (A_1,\ldots,A_p),$$
$$B = \begin{bmatrix} B_1 & & O \\ & \ddots & \\ O & & B_p \end{bmatrix},\ \ b = (b_0^T, b_1^T, \ldots, b_p^T)^T.$$

For such multiobjective programming problems, instead of optimal solutions for ordinary single-objective optimization problems, Pareto optimal solutions such that the value of a certain objective function is never improved without the value of another objective function getting worse are de-

‡ Corresponding author
Tel:+81-824-24-7694, Fax:+81-824-24-7694
E-mail: sakawa@msl.sys.hiroshima-u.ac.jp
† Presenter, 4-1, Kagamiyama 1-chome, Higashi-Hiroshima, 739, JAPAN
E-mail: kato@msl.sys.hiroshima-u.ac.jp

0-7803-4316-6/98/$10.00 ©1998 IEEE

fined.

Definition 1 (Pareto optimal solution)
$x^* \in X$ is said to be a Pareto optimal solution if and only if there does not exist another $x \in X$ such that $c_i x \leq c_i x^*$, $i = 1, \ldots, k$ and $c_j x < c_j x^*$ for at least one j, $1 \leq j \leq k$.

In practice, it would be more appropriate to consider that the possible values of the parameters in the above multiobjective 0-1 programming problem usually involve the ambiguity of the experts' understanding of the real system. Thus, we consider the following multiobjective block angular 0-1 programming problems with fuzzy parameters as follows.

$$\left.\begin{array}{lrl}
\text{minimize} & c_1 x = & \tilde{c}_{11} x_1 + \cdots + \tilde{c}_{1p} x_p \\
\vdots & & \vdots \\
\text{minimize} & c_k x = & \tilde{c}_{l1} x_1 + \cdots + \tilde{c}_{lp} x_p \\
\text{subject to} & & \tilde{A}_1 x_1 + \cdots + \tilde{A}_p x_p \leq \tilde{b}_0 \\
& & \tilde{B}_1 x_1 \qquad\qquad \leq \tilde{b}_1 \\
& & \ddots \qquad \vdots \\
& & \tilde{B}_p x_p \leq \tilde{b}_p \\
& & x_j \in \{0,1\}^{n_j},\ j = 1, \ldots, p
\end{array}\right\} \quad (2)$$

where a vector $\tilde{y}$ means that its elements are fuzzy numbers.

Now suppose that the decision maker considers that the degree of all of the membership functions of the fuzzy numbers involved in the multiobjective 0-1 programming problem with fuzzy parameters should be greater than or equal to a fixed value α, i.e., (A, B, b, c) should be an element in the α-level set $(\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha$. For the above α and the multiobjective 0-1 programming problem with fuzzy parameters, the following nonfuzzy α-multiobjective 0-1 programming problem is defined as:

$$\left.\begin{array}{lrl}
\text{minimize} & c_1 x = & c_{11} x_1 + \cdots + c_{1p} x_p \\
\vdots & & \vdots \\
\text{minimize} & c_k x = & c_{k1} x_1 + \cdots + c_{kp} x_p \\
\text{subject to} & & A_1 x_1 + \cdots + A_p x_p \leq b_0 \\
& & B_1 x_1 \qquad\qquad \leq b_1 \\
& & \ddots \qquad \vdots \\
& & B_p x_p \leq b_p \\
& & x_j \in \{0,1\}^{n_j},\ j = 1, \ldots, p \\
& & (A, B, b, c) \in (\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha
\end{array}\right\} \quad (3)$$

where $X(A, B, b)$ denotes the feasible region of the above α-multiobjective 0-1 programming problem and it should be emphasized that the parameters (A, B, b, c) are treated as decision variables rather than constants.

Then, it is evident that the notion of the ordinary Pareto optimality cannot be applied directly. Thus, on the basis of the α-level sets of the fuzzy parameters, the concept of an α-Pareto optimal solution to the α-multiobjective 0-1 programming problem (3) as a natural extension of the Pareto optimality is defined [7].

Definition 2 (α-Pareto optimal solution)
$x^* \in X(A^*, B^*, b^*)$ is said to be an α-Pareto optimal solution to the α-multiobjective 0-1 programming problem if and only if there does not exist another $x \in X(A, B, b)$, $(A, B, b, c) \in (\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha$ such that $c_i x \leq c_i x^*$, $i = 1, \ldots, k$, and $c_j(x) < c_j x^*$ for at least one j, $1 \leq j \leq k$, where the corresponding values of parameters (A^*, B^*, b^*, c^*) are called α-level optimal parameters.

2.1. *Fuzzy goals*

For the α-multiobjective 0-1 programming problem (3), considering the vague nature of the decision maker's judgements, it is quite natural to assume that the decision maker may have imprecise or fuzzy goals for the objective functions in the α-multiobjective 0-1 programming problem. In general, goals stated by the decision maker may be to achieve "$c_i x$ should be in the vicinity of a fixed value" (called fuzzy equal) or "$c_i x$ should be substantially less than or equal to a fixed value" (called fuzzy min) or "$c_i x$ should be substantially greater than or equal to a fixed value" (called fuzzy max). Such a generalized α-multiobjective 0-1 programming problem may now be expressed as

$$\left.\begin{array}{rl}
\text{fuzzy min} & c_i x \quad i \in I_1 \\
\text{fuzzy max} & c_i x \quad i \in I_2 \\
\text{fuzzy equal} & c_i x \quad i \in I_3 \\
\text{subject to} & x \in X(A, B, b) \\
& (A, B, b, c) \in (\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha
\end{array}\right\} \quad (4)$$

where $I_1 \cup I_2 \cup I_3 = \{1, \ldots, k\}$, $I_i \cap I_j = \emptyset$, $i, j = 1, 2, 3$, $i \neq j$.

Assuming each of fuzzy goals is quantified by a membership function $\mu_i(c_i x)$, $i = 1, \ldots, k$, as shown in Fig. 1, the problem (4) can be written as the following fuzzy α-multiobjective decision making problem:

$$\left.\begin{array}{rl}
\text{maximize} & (\mu_1(c_1 x), \ldots, \mu_k(c_k x)) \\
\text{subject to} & x \in X(A, B, b) \\
& (A, B, b, c) \in (\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha
\end{array}\right\} \quad (5)$$

For the above problem, the concept of M-α-Pareto optimality is defined [7].

Definition 3 (M-α-Pareto optimal solution)
$x^* \in X(A^*, B^*, b^*)$ is said to be an M-α-Pareto optimal solution to the α-multiobjective 0-1 programming problem if and only if there does not exist another $x \in X(A, B, b)$, $(A, B, b, c) \in (\tilde{A}, \tilde{B}, \tilde{b}, \tilde{c})_\alpha$ such that $\mu_i(c_i x) \leq \mu_i(c_i x^*)$, $i = 1, \ldots, k$, and $\mu_j(c_j x) < \mu_j(c_j x^*)$ for at least one j $(1 \leq j \leq k)$, where the corresponding values of parameters (A^*, B^*, b^*, c^*) are called M-α-level optimal parameters.

Note that M-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 (5)

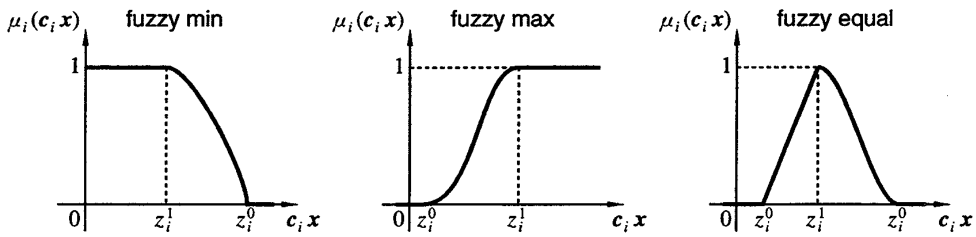

Figure 1. Membership functions of fuzzy goals

as $(\boldsymbol{x}, A, B, \boldsymbol{b}, C)$.

Having elicited the membership functions $\mu_i(\boldsymbol{c}_i\boldsymbol{x})$, $i = 1, \ldots, k$, from the DM for each of the objective functions $\boldsymbol{c}_i\boldsymbol{x}$, $i = 1, \ldots, k$, if a general aggregation function $\mu_D(\mu_1(\boldsymbol{c}_1\boldsymbol{x}), \ldots, \mu_k(\boldsymbol{c}_k\boldsymbol{x}), \alpha)$ is introduced, the decision making problem considering k conflicting objective functions can be formally defined as follows [7].

$$
\begin{array}{ll}
\text{maximize} & \mu_D(\mu_1(\boldsymbol{c}_1\boldsymbol{x}), \ldots, \mu_k(\boldsymbol{c}_k\boldsymbol{x}), \alpha) \\
\text{subject to} & \boldsymbol{x} \in X(A, B, \boldsymbol{b}) \\
 & (A, B, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{B}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}})_\alpha \\
 & \alpha \in [0, 1]
\end{array} \qquad (6)
$$

If the form of the function $\mu_D(\cdot)$ can be identified explicitly, this problem can be converted into an ordinary single-objective programming problem. However, it is so difficult to identify $\mu_D(\cdot)$ globally and explicitly that an interactive decision making is introduced in order to find a satisficing solution for the DM among from the M-α-Pareto optimal solution set.

3. Augmented minimax problems

Assume that the reference membership levels $\bar{\mu}_i$, $i = 1, \ldots, k$, reflecting the aspiration level of the DM for each membership function $\mu_i(\boldsymbol{c}_i\boldsymbol{x})$ is subjectively specified by the DM. Then, if the reference membership levels are attainable, an M-α-Pareto optimal solution which gives better values of the membership functions for fuzzy goals than the reference membership levels can be obtained. While, if not, it is desirable to obtain an M-α Pareto optimal solution which is nearest to the reference membership levels in the minimax sense.

Such an M-α-Pareto optimal solution can be obtained by solving the following augmented minimax problem [7].

$$
\left.
\begin{array}{ll}
\text{minimize} & \displaystyle\max_{i=1,\ldots,k} \{\bar{\mu}_i - \mu_i(\boldsymbol{c}_i\boldsymbol{x}) \\
 & \displaystyle\qquad + \rho\sum_{j=1}^{k}(\bar{\mu}_i - \mu_i(\boldsymbol{c}_i\boldsymbol{x}))\} \\
\text{subject to} & \boldsymbol{x} \in X(A, B, \boldsymbol{b}) \\
 & (A, B, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{B}, \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{c}})_\alpha
\end{array}
\right\} \qquad (7)
$$

where ρ is a sufficiently small positive number. By solving the above augmented minimax problem (7), an M-α-Pareto optimal solution which is nearest to the reference membership levels in the minimax sense can be obtained regardless of its uniqueness [7].

From the properties of the α-level set for $\tilde{A}$, $\tilde{B}$, $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{c}}$, the feasible regions for A, B, $\boldsymbol{b}$ and $\boldsymbol{c}$ can be expressed respectively by closed intervals $[A^L_\alpha, A^R_\alpha]$, $[B^L_\alpha, B^R_\alpha]$, $[\boldsymbol{b}^L_\alpha, \boldsymbol{b}^R_\alpha]$ and $[\boldsymbol{c}^L_\alpha, \boldsymbol{c}^R_\alpha]$.

Now, define $\mu_{iR}(\cdot)$ and $\mu_{iL}(\cdot)$ as follows (Fig. 2).

$$
\mu_{iR}(\boldsymbol{c}_i\boldsymbol{x}) = \begin{cases} 1 & , \ \boldsymbol{c}_i\boldsymbol{x} \le z_i^1 \\ \mu_i(\boldsymbol{c}_i\boldsymbol{x}) & , \ \boldsymbol{c}_i\boldsymbol{x} > z_i^1 \end{cases} \qquad (8)
$$

$$
\mu_{iL}(\boldsymbol{c}_i\boldsymbol{x}) = \begin{cases} \mu_i(\boldsymbol{c}_i\boldsymbol{x}) & , \ \boldsymbol{c}_i\boldsymbol{x} < z_i^1 \\ 1 & , \ \boldsymbol{c}_i\boldsymbol{x} \ge z_i^1 \end{cases} \qquad (9)
$$

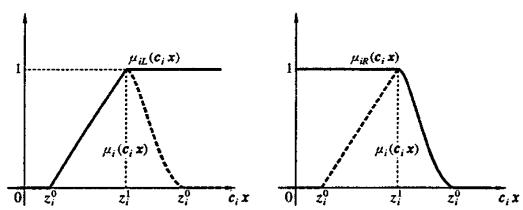

Figure 2. $\mu_{iL}(\cdot)$ and $\mu_{iR}(\cdot)$

Since the DM can be choose the most desirable parameter values $(A, B, \boldsymbol{b}, \boldsymbol{c}) \in (\tilde{A}, \tilde{B}, \tilde{\boldsymbol{b}}, \tilde{C})_\alpha$, the membership value of the ith objective function $M_i(\boldsymbol{x})$ for a solution $\boldsymbol{x}$ is considered to be defined as:

$$
M_i(\boldsymbol{x}) = \begin{cases} \mu_i(\boldsymbol{c}_i^L\boldsymbol{x}) & , i \in I_1 \\ \mu_i(\boldsymbol{c}_i^R\boldsymbol{x}) & , i \in I_2 \\ \min\{\mu_{iR}(\boldsymbol{c}_i^L\boldsymbol{x}), \mu_{iL}(\boldsymbol{c}_i^R\boldsymbol{x})\} & , i \in I_3 \end{cases} \qquad (10)
$$

Then, the problem (7) can be rewritten as follows.

$$
\left.
\begin{array}{ll}
\text{minimize} & \displaystyle\max_{i=1,\ldots,k} \{(\bar{\mu}_i - M_i(\boldsymbol{x})) \\
 & \displaystyle\qquad + \rho\sum_{j=1}^{k}(\bar{\mu}_i - M_i(\boldsymbol{x}))\} \\
\text{subject to} & A^L_{1\alpha}\boldsymbol{x}_1 + \cdots + A^L_{p\alpha}\boldsymbol{x}_p \le \boldsymbol{b}^R_{0\alpha} \\
 & B^L_{1\alpha}\boldsymbol{x}_1 \qquad\qquad\quad \le \boldsymbol{b}^R_{1\alpha} \\
 & \qquad\qquad \ddots \qquad\qquad \vdots \\
 & \qquad\qquad\qquad B^L_{p\alpha}\boldsymbol{x}_p \le \boldsymbol{b}^R_{p\alpha} \\
 & \boldsymbol{x}_j \in \{0, 1\}^{n_j}, \ j = 1, \ldots, p
\end{array}
\right\} \qquad (11)
$$

In an interactive fuzzy satisficing method [6], the decision maker updates the reference membership levels, the degree α or both repeatedly until he is satisfied with the α-Pareto optimal solution for the reference levels and the degree α by solving the corresponding minimax problem (11). The interactive algorithm to obtain such satisficing solution for the decision maker can be constructed as follows.

Step 0 Calculate the minimum and the maximum of each objective function under the given constraints for $\alpha = 0$ and $\alpha = 1$.

Step 1 Considering the minimum and the maximum of each objective function, the decision maker subjectively specifies the membership function for each objective function.

Step 2 Ask the decision maker to select the initial value of α $(0 \le \alpha \le 1)$ and the initial reference membership levels $\bar{\mu}_i$ $(i = 1, \ldots, l)$.

Step 3 Calculate a Pareto optimal solution for the above reference levels by solving the problem (11).

Step 4 If the decision maker is satisfied with the current values of objective functions given by the current optimal solution, stop. Otherwise, ask the decision maker to update reference membership levels by taking account of the current values of membership functions and objective functions and return to step 3.

Since the problem (11) is an ordinary block angular 0-1 programming problem of knapsack type, a genetic algorithm with decomposition procedures using triple string representation [5], [6] is applicable.

4. Genetic algorithms with decomposition procedures

4.1. Coding and decoding

For multiobjective 0-1 programming problems of knapsack type, i.e., where all coefficients and all right side constants in the constraints are nonnegative, a genetic algorithm using double string representation as shown in Fig. 3 was proposed by M. Sakawa et al. [3], [4]. where $s(i)$ corresponds to

Indices	$s(1)$	$s(2)$	$\cdots$	$s(i)$	$\cdots$	$s(n)$
0-1 values	$g_{s(1)}$	$g_{s(2)}$	$\cdots$	$g_{s(i)}$	$\cdots$	$g_{s(n)}$

Figure 3. Double string

the index j of a variable x_j and $x_{s(i)}$ denotes the value of the variable x_j.

In order to guarantee the feasibility of a solution (phenotype) generated from an individual (genotype), M. Sakawa et al. [3], [4] proposed a decoding algorithm for a double string. In the decoding algorithm, starting from the left edge of the string, a variable $x_{s(i)}$ corresponding to each index $s(i)$, $i = 1, 2, \ldots,$, is fixed to $g_{s(i)}$, i.e., $x_{s(i)} = g_{s(i)}$ until all constraints are satisfied. For remaining variables, if the constraints are broken when $g_{s(i)} = 1$, the value of the variable with the index $s(i)$ is fixed to 0 by force. By doing so, only feasible solutions will be generated.

In view of the special structure of the problem (11), it seems to be quite reasonable to define an individual $\mathbf{S}$ as an aggregation of p subindividuals $\mathbf{s}^j$, $j = 1, \ldots, p$, corresponding to the block constraint $B_j \boldsymbol{x}_j \le \boldsymbol{b}_j$ (Fig. 4).

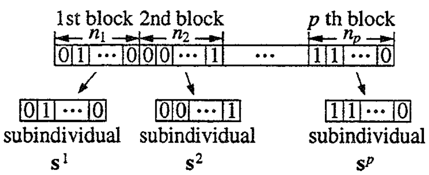

Figure 4. Division of an individual $\mathbf{S}$ into p subindividuals $\mathbf{s}^j$

In [5], [6], each subindividual $\mathbf{s}^j$ is presented by a triple string as shown in Fig. 5. To be

	r^j			
$\mathbf{s}^j =$	$\nu^j(1)$	$\nu^j(2)$	$\cdots$	$\nu^j(n_j)$
	$g^j_{\nu^j(1)}$	$g^j_{\nu^j(2)}$	$\cdots$	$g^j_{\nu^j(n_j)}$

Figure 5. Triple string

more explicit, in a triple string which represents a subindividual corresponding to the jth block, r^j $(\in \{1, \ldots, p\})$ represents the priority of the jth block, $\nu^j(k)$ $(\in \{1, \ldots, n_j\})$ denotes an index of a variable in phenotype and $g^j_{\nu^j(k)}$ is a 0-1 value variable.

Decoding this string (genotype) by means of the following algorithm, the resulting solution (phenotype) becomes always feasible. In the algorithm, n_j denotes the number of variables in the jth block $(j = 1, \ldots, p)$, $\boldsymbol{\alpha}^j_{\nu^j(k)}$ is the $\nu^j(k)$th column vector in the jth coupling constraint coefficient matrix A_j and $\boldsymbol{\beta}^j_{\nu^j(k)}$ is the $\nu^j(k)$th column vector in the jth block constraint coefficient matrix B_j.

Step 1 Set $i = 1$, $\boldsymbol{\Sigma} = \mathbf{0}$ and proceed to step 2.

Step 2 Find out such a block as $i = r^j$ and proceed to step 3.

Step 3 For the above block j, set $k = 1$, $\boldsymbol{\sigma} = \mathbf{0}$ and repeat the following procedures.

(a) If $g^j_{\nu^j(k)} = 1$, set $k = k + 1$ and go to (b). Otherwise, i.e., if $g^j_{\nu^j(k)} = 0$, set $k = k + 1$ and go to (c).

(b) If $\boldsymbol{\Sigma} + \boldsymbol{\alpha}^j_{\nu^j(k)} \le \boldsymbol{b}_0$ and $\boldsymbol{\sigma} + \boldsymbol{\beta}^j_{\nu^j(k)} \le \boldsymbol{b}_j$, set $x^j_{\nu^j(k)} = 1$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma} + \boldsymbol{\alpha}^j_{\nu^j(k)}$, $\boldsymbol{\sigma} = \boldsymbol{\sigma} + \boldsymbol{\beta}^j_{\nu^j(k)}$, and go to (c). Otherwise, set $x^j_{\nu^j(k)} = 0$ and go to (c).

(c) If $k > n_j$, go to step 4 and regard $\boldsymbol{x}^j = (x_1^j, \ldots, x_{n_j}^j)^T$ as phenotype of a subindividual $\mathbf{s}^j$ represented by triple string. Otherwise, return to (a).

Step 4 If $i = p$, stop and regard $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^p)^T$ as phenotype of an individual $\mathbf{S}$. Otherwise, set $i = i + 1$ and return to step 2.

4.2. Fitness

Fitness f_i of each individual $\mathbf{S}_i$ is defined as

$$f_i = 1.0 + k\rho - \max_{i=1,\ldots,k}\Big\{(\bar{\mu}_i - M_i(\boldsymbol{x})) + \rho \sum_{j=1}^{k} (\bar{\mu}_i - M_i(\boldsymbol{x}))\Big\}.$$

As a way of scaling of fitness, the linear scaling $f_i' = a \cdot f_i + b$ is adopted, where the constants a and b are determined so that the mean fitness f_{mean} will be a fixed point ($f_{\text{mean}} = a \cdot f_{\text{mean}} + b$) and the max fitness f_{max} will be mapped to the twice value of the mean fitness ($2 \cdot f_{\text{mean}} = a \cdot f_{\text{max}} + b$).

4.3. Reproduction

Various kinds of reproduction methods have been proposed. Among them, M. Sakawa et al. have already investigated the performance of each of 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, it was confirmed that elitist expected value selection is relatively efficient for multiobjective 0-1 programming problems incorporating the fuzzy goals of the decision maker [3]. In this paper, according to [3], as a reproduction operator, elitist expected value selection is adopted.

4.4. Crossover

If a single-point crossover or multi-point crossover is directly applied to individuals of triple string type, an index $\nu^j(k)$ in the j th subindividual of an offspring may take the same number that an index $\nu^j(k')$ ($k \neq k'$) takes. The same violation occurs in solving traveling salesman problem or scheduling problem through genetic algorithm as well. In order to avoid this violation, a crossover method called partially matched crossover (PMX) was modified to be suitable for double strings [3], [4]. In this paper, PMX is applied as usual for upper string, whereas, for a couple of middle string and lower string, PMX for double string [3], [4] is applied to every subindividual as in [5], [6].

4.5. Mutation and inversion

It is considered that mutation plays a role of local random search in genetic algorithm. In this paper, only for the lower string of triple string, mutation of bit-reverse type is adopted and applied to every subindividual.

Furthermore, In this paper, for the middle string and for the upper string of the triple string, inversion defined by the following algorithm is adopted.

Step 1 After determining two inversion points h and k ($h < k$), pick out the part of the string from h to k.

Step 2 Arrange the substring in reverse order.

Step 3 Put the arranged substring back in the string.

4.6. The whole algorithm

When applying a genetic algorithm to the problem (11), an approximate optimal solution of desirable precision should be obtained in proper time. For this reason, the following parameters, the minimal search generation I_{min}, the maximal search generation I_{max}, the convergence criterion ε are introduced.

The outline of the genetic algorithm with decomposition procedures proposed by the authors [5], [6], is shown as follows.

Step 1 Set an iteration index (generation) $t = 0$ and determine the probability of crossover p_c, the probability of mutation p_m, the probability of inversion p_i, the minimal search generation I_{min}, the maximal search generation I_{max} and the convergence criterion ε.

Step 2 Generate N individuals whose subindividuals are of triple string type at random.

Step 3 Evaluate each individual (subindividual) on the basis of phenotype obtained by the proposed decoding algorithm and calculate the mean fitness f_{mean} and the maximal fitness f_{max} of the population. If $t > I_{\text{min}}$ and $(f_{\text{max}} - f_{\text{mean}})/f_{\text{max}} < \varepsilon$, or, if $t > I_{\text{max}}$, regard an individual with the maximal fitness as an optimal individual and terminate this program. Otherwise, set $t = t + 1$ and proceed to step 4.

Step 4† Apply reproduction to every subindividual.

Step 5† Apply crossover for a couple of middle string and lower string to every subindividual according to the probability of crossover p_c.

Step 6† Apply mutation to every subindividual according to the probability of mutation p_m.

Step 7† Apply inversion to every subindividual according to p_m.

Step 8 Apply crossover for upper string according to p_c.

Step 9 Apply inversion for upper string according to p_i and return to step 3.

1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia. Editors, L.C. Jain and R.K.Jain

In the above genetic algorithm, several operations in the steps with † can be applied to every subindividual independently. As a result, it is possible to reduce the amount of working memory needed to solve a problem.

5. Numerical experiments

In the paper [5], [6], the genetic algorithm using triple string representation with decomposition procedures (TGA) were applied only to block angular 0-1 programming problems with 20 and 25 variables.

In the present paper, we attempt to apply it to several problems of larger size (with 30, 50, 70 and 100 variables). For comparison, we also apply a genetic algorithm using double string without decomposition procedures (DGA) [3], [4] to the same problems.

In the following numerical experiments, the corresponding problem was solved through 5 runs of TGA and DGA at each interaction with the decision maker. The parameters used in GA were set as, the population size = 100, the probability of crossover $p_c = 0.7$, the probability of mutation $p_m = 0.005$, the probability of inversion $p_i = 0.01$, the convergence criterion $\varepsilon = 0.05$, $I_{\max} = 1000$ and $I_{\min} = 200$.

First, we consider a multiobjective block angular 0-1 programming problem with fuzzy parameters (3 objectives, 3 blocks, 30(=12+8+10) variables and 3 coupling constraints).

Table 1, 2 shows the interaction processes with the DM by TGA and DGA respectively.

At first, the minimax problem was solved for the initial reference membership levels and α (1.0, 1.0, 1.0, 1.0), and the decision maker was supplied with the corresponding Pareto optimal solution and both the membership function values and the objective function values as is shown in the first interaction of Table 1, 2. On the basis of such information, since the decision maker was not satisfied with the current membership function values or objective function values , the decision maker updated the reference membership values to $\bar{\mu}_1 = 0.8$, $\bar{\mu}_2 = 1.0$, $\bar{\mu}_3 = 1.0$ and $\alpha = 1.0$ for improving the satisfaction levels for z_2 and z_3 at the expense of z_1. For the updated reference membership values, the corresponding minimax problem yielded the Pareto optimal solution and both the membership function values and the objective function values as shown in the second interaction of Table 1, 2. The same procedure continues in this manner until the decision maker was satisfied with the current values of the membership functions and the objective functions. At the fourth interaction, the satisficing solution for the decision maker was derived (Table 1, 2).

In this example, the average processing time (APT) through all interaction processes and the average relative error (ARE)

$$\text{ARE} = \frac{\text{Best value by GA} - \text{Exact optimal value}}{\text{Exact optimal value}}$$

are shown in Table 3.

Table 3. Average processing time (APT) and average relative error (ARE) for a problem with 30 variables

	APT (sec)	ARE (×100%)
TGA	211.010	0.1122
DGA	242.654	0.1122

Next, we consider a multiobjective block angular 0-1 programming problem with fuzzy parameters (3 objectives, 5 blocks, 50(=12+8+10+7+13) variables and 5 coupling constraints).

Table 4 shows the average processing time (APT) and the average relative error (ARE) for the problem involving 50 variables. Figure 6 shows a graph of processing times for typical problems with 30, 50, 70 and 100 variables. From all results shown in above, in general, the genetic algorithm using triple string representation with decomposition procedures are more efficient and effective than the genetic algorithm using double string.

Table 4. Average processing time (APT) and average relative error (ARE) for a problem with 50 variables

	APT (sec)	ARE (×100%)
TGA	315.366	1.299
DGA	383.333	1.472

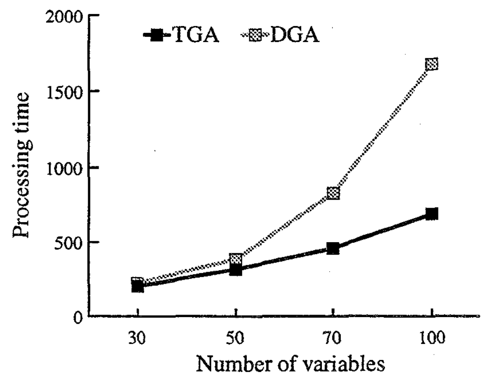

Figure 6. Processing time for typical problems with 30, 50, 70 and 100 variables

6. Conclusions

In this paper, an interactive fuzzy satisficing method through genetic algorithms for large scale

Table 1. Interaction process (TGA, 5 trials)

Interaction	z_1	z_2	z_3	μ_1	μ_2	μ_3	Number of solutions
1st	−6178	2916	−71	0.6525	0.6737	0.6554	4
(1.0, 1.0, 1.0, 1.0)	−6179	3217	−215	0.6526	0.6486	0.6723	1
(Optimum)	−6178	2916	−71	0.6525	0.6737	0.6554	
2nd	−4892	2123	−840	0.5356	0.7398	0.7459	5
(0.8, 1.0, 1.0, 1.0)							
(Optimum)	−4892	2123	−840	0.5356	0.7398	0.7459	
3rd	−4979	2143	−682	0.5435	0.7381	0.7273	5
(0.8, 1.0, 0.9, 1.0)							
(Optimum)	−4979	2143	−682	0.5435	0.7381	0.7273	
4 th	−4989.9	2084.1	−27.5	0.5445	0.7430	0.6503	5
(0.8, 1.0, 0.9, 0.7)							
(Optimum)	−4989.9	2084.1	−27.5	0.5445	0.7430	0.6503	

Table 2. Interaction process (DGA, 5 trials)

Interaction	z_1	z_2	z_3	μ_1	μ_2	μ_3	Number of solutions
1st	−6178	2916	−71	0.6525	0.6737	0.6554	4
(1.0, 1.0, 1.0, 1.0)	−6179	3217	−215	0.6526	0.6486	0.6723	1
(Optimum)	−6178	2916	−71	0.6525	0.6737	0.6554	
2nd	−4892	2123	−840	0.5356	0.7398	0.7459	5
(0.8, 1.0, 1.0, 1.0)							
(Optimum)	−4892	2123	−840	0.5356	0.7398	0.7459	
3rd	−4979	2143	−682	0.5435	0.7381	0.7273	5
(0.8, 1.0, 0.9, 1.0)							
(Optimum)	−4979	2143	−682	0.5435	0.7381	0.7273	
4 th	−4989.9	2084.1	−27.5	0.5445	0.7430	0.6503	5
(0.8, 1.0, 0.9, 0.7)							
(Optimum)	−4989.9	2084.1	−27.5	0.5445	0.7430	0.6503	

fuzzy multiobjective 0-1 programming problems and a genetic algorithm with decomposition procedures were explained roughly. In order to investigate the performance of the genetic algorithm, it was applied to several large scale fuzzy multiobjective 0-1 programming problems and compared with other solution methods about processing time and solution precision. The results of the numerical experiments showed the efficiency and effectiveness of the genetic algorithm with decomposition procedures.

References

1. L.S. Lasdon, "Optimization Theory for Large Systems", Macmillan, New York, 1970.
2. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, 1992, Second extended edition, 1994, Third revised and extended edition, 1996.
3. M. Sakawa, M. Inuiguchi, H. Sunada and K. Sawada, "Fuzzy multiobjective combinatorial optimization through revised genetic algorithms," Journal of Japan Society for Fuzzy Theory and Systems, Vol. 6, No. 1, pp. 177-185, 1994 (in Japanese).
4. M. Sakawa, K. Kato, H. Sunada and Y. Enda, "An interactive fuzzy satisficing method for multiobjective 0-1 programming problems through revised genetic algorithms," Journal of Japan Society for Fuzzy Theory and Systems, Vol. 7, No. 2, pp. 361-370, 1995 (in Japanese).
5. K. Kato and M. Sakawa, "Genetic Algorithms with Decomposition Procedures for Fuzzy Multiobjective 0-1 Programming Problems with Block Angular Structure," Proceedings of 1996 IEEE International Conference on Evolutionary Computation, pp. 706-709, 1996.
6. K. Kato and M. Sakawa, "An interactive fuzzy satisficing method for multiobjective block angular 0-1 programming problems involving fuzzy parameters through genetic algorithms with decomposition procedures," Proceedings of Seventh International Fuzzy Systems Association World Congress, Vol. 3, pp. 9-14, 1997.
7. M. Sakawa, "*Fuzzy Sets and Interactive Multiobjective Optimization*," Plenum Press , New York, 1993.