An Interactive Fuzzy Satisficing Method for Multiobjective Multidimensional 0-1 Knapsack Problems through Genetic Algorithms

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

Kosuke KATO
Faculty of Engineering
Hiroshima University
Higashi-Hiroshima, Japan, 739
kato@msl.sys.hiroshima-u.ac.jp

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

Abstract— **In this paper, an interactive fuzzy satisficing method for multiobjective multidimensional 0-1 knapsack problems is proposed by incorporating the desirable features of both the interactive fuzzy programming methods and genetic algorithms. By considering the vague nature of human judgements, fuzzy goals of the decision maker (DM) for objective functions are quantified by eliciting linear membership functions. If the DM specifies a reference membership level for each of the membership functions, the corresponding (local) Pareto optimal solution can be obtained by solving the formulated minimax problem through a genetic algorithm with double strings. For obtaining an optimal solution not dominated by the solutions before interaction, the algorithm is revised by introducing some new mechanism for forming an initial population. Illustrative numerical examples demonstrate both feasibility and effectiveness of the proposed method.**

KeyWords— **Multiobjective multidimensional 0-1 knapsack problems, interactive fuzzy satisficing methods, genetic algorithms, double strings**

I. Introduction

Genetic algorithms [1], a new learning paradigm that models a natural evolution mechanism, have recently received a great deal of attention regarding their potential as optimization techniques to solve combinatorial optimization problems [2; 3]. For multiobjective 0-1 programming problems, to generates only feasible solutions without using penalty functions, the authors have proposed a genetic algorithm with double strings [5]. Moreover, incorporating fuzzy goals of the decision maker (DM) for objective functions together with fuzzy decision, a compromise solution for the DM can be derived efficiently through the proposed genetic algorithm. There remains, however, such a problem that no interaction with the DM is considered once the membership functions have been determined.

In this paper, an interactive fuzzy satisficing method for multiobjective multidimensional 0-1 knapsack problem is proposed by incorporating the advantages of both genetic algorithms with double strings [5] and interactive fuzzy programming [4]. The basic idea behind interactive fuzzy programming is to derive a satisficing solution for the DM from a set of Pareto optimal solutions efficiently by updating reference membership levels. Unfortunately, however, it is significant to realize that a simple hybrid between the interactive method and the genetic algorithm suffers from a lot of possibility to generate optimal solutions dominated by those obtained before interaction. With this observation in mind, we propose an interactive fuzzy satisficing methods by modifying the generation method of an initial population in the genetic algorithms with double strings.

II. Interactive Fuzzy Programming through Genetic Algorithm

In general, a multiobjective multidimensional 0-1 knapsack problem with l distinct objective functions $z_i(\boldsymbol{x}) = \boldsymbol{c}_i\boldsymbol{x}$ $(i = 1, \ldots, l)$ is formulated as follows:

$$\left.\begin{array}{ll} \text{minimize} & (z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), \ldots, z_l(\boldsymbol{x}))^T \\ \text{subject to} & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{array}\right\} \quad (1)$$

where $\boldsymbol{c}_i = (c_{i1}, \ldots, c_{in})$ $(i = 1, \ldots, l)$, $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $\boldsymbol{b} = (b_1, \ldots, b_m)^T$, $A = (a_{jk})$ $(j = 1, \ldots, m; k = 1, \ldots, n)$ is an $m \times n$ matrix. Furthermore, it is assumed that each element of $\boldsymbol{c}_i$ is a real number and each element of A and $\boldsymbol{b}$ is a positive integer.

For such a multiobjective 0-1 programming problem, considering the vague nature of the DM's judgements, it is quite natural to assume that the DM may have a fuzzy goal such as "$z_i(\boldsymbol{x})$ should be substantially less than or equal to a fixed value". Such a fuzzy goal of the DM can be quantified by eliciting a membership function.

In this paper, for simplicity, the linear membership function

$$\mu_i(z_i(\boldsymbol{x})) = \begin{cases} 0 & ; \quad z_i(\boldsymbol{x}) > z_i^0 \\ \dfrac{z_i(\boldsymbol{x}) - z_i^0}{z_i^1 - z_i^0} & ; \quad z_i^1 < z_i(\boldsymbol{x}) \leqq z_i^0 \\ 1 & ; \quad z_i(\boldsymbol{x}) \leqq z_i^1 \end{cases} \quad (2)$$

is adopted, where z_i^0 and z_i^1 denotes values of the objective function $z_i(\boldsymbol{x})$ whose degree of membership function are 0 and 1, respectively. These values are subjectively determined through interaction with the DM.

Having elicited the linear membership functions $\mu_i(z_i(\boldsymbol{x}))$ $(i = 1, \ldots, k)$ from the DM for each of the objective function $z_i(\boldsymbol{x})$ $(i = 1, \ldots, k)$, if we introduce a general conjunctive function

$$\mu_D(\boldsymbol{x}) = \mu_D(\mu_1(z_1(\boldsymbol{x})), \ldots, \mu_k(z_k(\boldsymbol{x}))), \quad (3)$$

the problem (1) can be written as the following fuzzy multiobjective decision making problem:

$$\underset{x \in X}{\text{maximize}}\, \mu_D(\boldsymbol{x}), \quad (4)$$

where X is the constrained set of the problem (1) and the value of the conjunctive function $\mu_D(\boldsymbol{x})$ is interpreted as degree of satisfaction of the DM for the whole of k fuzzy goals. As the conjunctive function, if we adopt the well-known fuzzy decision of Bellman and Zadeh or minimum operator, the multiobjective 0-1 programming problem (1) can be interpreted as:

$$\left.\begin{array}{ll} \text{maximize} & \min_{i=1,\dots,l} \{\mu_i(z_i(\boldsymbol{x}))\} \\ \text{subject to} & A\boldsymbol{x} \leqq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{array}\right\} \quad (5)$$

In this formulation, however, there still remains a series problem that no interaction with the DM is considered once membership functions have been determined.

To cope with this problem, in this paper, an interactive method for multiobjective programming problems with continuous variables [4] is introduced. In this method, after determining a membership function for each of the objective functions, the DM is asked to specify a reference point $\bar{\boldsymbol{\mu}} = (\bar{\mu}_1, \dots, \bar{\mu}_l)^T$ which reflects an aspiration level of the DM for each of the membership functions. The corresponding Pareto optimal solution, which is closest to the reference point or better than that if the reference point is attainable in the minimax sense, can be obtained by solving the minimax problem [4]

$$\left.\begin{array}{ll} \text{minimize} & \max_{i=1,\dots,l} \{\bar{\mu}_i - \mu_i(z_i(\boldsymbol{x}))\} \\ \text{subject to} & A\boldsymbol{x} \leqq \boldsymbol{b} \\ & \boldsymbol{x} \in \{0,1\}^n \end{array}\right\} \quad (6)$$

where $\bar{\mu}_i$'s are called reference membership levels. Incorporating genetic algorithms with double strings into this interactive fuzzy programming method, it becomes possible to introduce the following interactive algorithm for deriving a satisficing solution of the DM.

Step 1 Set initial reference membership levels (if it is difficult to determine these values, set them to 1).

Step 2 Generate N individuals of length n represented by double strings at random.

Step 3 Evaluate each individual on the basis of phenotype (n dimensional vector) decoded from genotype (string).

Step 4 Apply reproduction operator.

Step 5 Apply crossover operator to individuals according to crossover rate p_c.

Step 6 Apply mutation operator to individuals according to mutation rate p_m.

Step 7 Repeat these procedures from step 3 to step 6 until termination conditions are satisfied. Then, regard an individual with the maximal fitness as an optimal individual and proceed to step 8.

Step 8 If the DM is satisfied with the current values of membership functions and objective functions given by the current optimal individual, stop. Otherwise, ask the DM to update reference membership levels by taking account of the current values of membership functions and objective functions and return to step 2.

III. Genetic Algorithms with Double Strings

A. Coding and decoding

Usually, an individual in genetic algorithms is represented by a 0-1 alphabetic string. This representation, however, may weaken ability of genetic algorithms since an individual whose phenotype is feasible is scarcely generated under this representation. In this paper, as one possible way to generate only feasible solutions, a double string as is shown in Fig. 1 is adopted [5].

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

Fig. 1. Double string

Decoding this string (genotype) by means of the following algorithm, the resulting solution (phenotype) becomes always feasible. In the algorithm, n, i, $s(i)$, $x_{s(i)}$ and $\boldsymbol{a}_{s(i)}$ denote respectively length of a string, a position in a string, an index of a variable, 0-1 value of a variable with index $s(i)$ decoded from a string and a column vector in the constraint coefficient matrix A.

Step 1 Set $i = 1$, $\boldsymbol{\Sigma} = \boldsymbol{0}$.

Step 2 If $g_{s(i)} = 1$, set $i = i + 1$ and go to step 3. Otherwise, i.e., if $g_{s(i)} = 0$, set $i = i + 1$ and go to step 4.

Step 3 If $\boldsymbol{\Sigma} + \boldsymbol{a}_{s(i)} \leqq \boldsymbol{b}$, set $x_{s(i)} = 1$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma} + \boldsymbol{a}_{s(i)}$ and go to step 4. Otherwise, set $x_{s(i)} = 0$ and go to step 4.

Step 4 If $i > n$, stop and regard $\boldsymbol{x} = (x_1, \dots, x_n)^T$ as phenotype of the individual represented by the double string. Otherwise, return to step 2.

B. Fitness and scaling

It seems quite natural to define the fitness function of each individual **S** by

$$f(\mathbf{S}) = 1 - \max_{i=1,\dots,l} \{\bar{\mu}_i - \mu_i(z_i(\boldsymbol{x}))\} \quad (7)$$

where S and $\boldsymbol{x}$ denote an individual represented by double string and phenotype of **S** respectively.

In reproduction operator based on the ratio of fitness of each individual to the total fitness such as expected value model, it is a problem that probability of selection depends on the relative ratio of fitness of each individual. Thus, linear scaling is adopted.

Linear scaling Fitness f_i of an individual is transformed into f_i' as follows:

$$f_i' = a \cdot f_i + b.$$

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.

C. Reproduction

Up to now, various reproduction methods have been proposed and considered [2; 3]. The authors 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 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.

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 i th string $\boldsymbol{s}_i$ in the next population

$$N_i = \left(f(\boldsymbol{s}_i)/\sum f(\boldsymbol{s}_i)\right) \times N$$

is calculated. Then, the integral part of N_i denotes the deterministic number of the string $\boldsymbol{s}_i$ preserved in the next population. While, the decimal part of N_i is regarded as probability for one of the string $\boldsymbol{s}_i$ to survive, i.e., $N - \sum N_i$ strings are determined on the basis of this probability.

D. Crossover

If a single-point crossover or multi-point crossover is applied to individuals of double string type, an index $s(k)$ in an offspring may take the same number that an index $s(k')$ $(k \neq k')$ takes. The same violation occurs in solving traveling salesman problem or scheduling problem through genetic algorithm as well. For the purpose of avoiding this violation, a crossover method called partially matched crossover (PMX) [2] is adopted.

E. Mutation

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

F. Convergence conditions

Applying genetic algorithms to an interactive multiobjective 0-1 programming problem, an approximate solution of desirable precision must be obtained in proper time. For this reason, two parameters $I_{\min}$ which denotes how many generations will have to be searched at least and $I_{\max}$ which does at most are introduced. Moreover the following condition of convergence is imposed.

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

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

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.

G. Numerical experiments

As a numerical example, consider a two-objective one-dimensional knapsack problem with 20 variables incorporating fuzzy goals of the DM,. The coefficients of the problem are determined at random. Concerning the fuzzy goals of the DM, the values of z_i^0 and z_i^1 are set to be their individual maximum and minimum respectively. The parameters of GA are set as, population size = 50, the crossover ratio $p_c = 0.9$, the mutation ratio $p_m = 0.02$, $\varepsilon = 0.05$, $I_{\max} = 1000$ and $I_{\min} = 100$. Moreover, suppose the DM updates the reference membership levels as $(1.0, 1.0) \to (0.9, 1.0) \to (0.85, 1.0)$ through interactions.

The results show the difference between the mean fitness of approximate optimal solutions through GA and the fitness of the true optimal solution is smaller than 1% after all interactions. Consequently, it is concluded that an approximate optimal solution of high precision is obtained through GA. However, unfortunately, some solutions calculated through GA for the updated reference membership levels are dominated, i.e., all objective function values of the solution are inferior to those of solutions before interaction.

IV. Modification of Genetic Algorithms

In the results of the above simulations, it is observed that the calculated solutions for updated reference membership levels are dominated by those calculated before updating. In order to overcome such an undesirable phenomenon that the calculated solutions for updated reference membership levels are not always Pareto optimal, the method of generating an initial population is modified as described in the following section.

A. Methods of generating initial population in interaction

In the experiments in the previous section, all strings included in the initial population were generated at random every interaction. Here, the following method of generating initial population is proposed.

Revised Method 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.

As a result, expected value selection and elitism selection are simultaneously adopted, and hence it is expected that the optimal solution after interaction will not be dominated.

B. Numerical experiments

As a numerical example, consider the fuzzy two objective knapsack problem with 20 variables discussed above. The parameters of GA are set as, population size = 50, the crossover ratio $p_c = 0.9$, the mutation ratio $p_m = 0.02$, $\varepsilon = 0.05$, $I_{\max} = 1000$ and $I_{\min} = 50$.

Table 1. Results when the reference membership level (1.0,1.0) was updated to (0.9,1.0).

The first interaction				The second interaction					
$z_1(x)$	$z_2(x)$	μ_1	μ_2	$z_1(x)$	$z_2(x)$	μ_1	μ_2	Number of solutions	Average generation number to converge
4628	2104	0.7280	0.7481	4336	1772	0.6821	0.7879	100	85.30
4667	2498	0.7341	0.7010	4336	1772	0.6821	0.7879	98	74.09
				4298	2102	0.6761	0.7484	1	
				4176	2033	0.6569	0.7566	1	
4604	2029	0.7242	0.7571	4336	1772	0.6821	0.7879	99	71.47
				4604	2029	0.7242	0.7571	1	
4599	2181	0.7234	0.7389	4336	1772	0.6821	0.7879	100	75.49
4500	2471	0.7078	0.7042	4336	1772	0.6821	0.7879	98	87.90
				4303	1830	0.6769	0.7809	1	
				4298	2102	0.6761	0.7484	1	

Table 2. Results when the reference membership level (0.9,1.0) was updated to (0.85,1.0).

The second iteration				The third iteration					
$z_1(x)$	$z_2(x)$	μ_1	μ_2	$z_1(x)$	$z_2(x)$	μ_1	μ_2	Number of solutions	Average generation number to converge
4336	1772	0.6821	0.7879	4336	1772	0.6821	0.7879	100	55.77
4327	1905	0.6806	0.7720	4336	1772	0.6821	0.7879	100	61.64
4604	2029	0.7242	0.7571	4336	1772	0.6821	0.7879	100	58.54
4298	2102	0.6761	0.7484	4336	1772	0.6821	0.7879	100	59.40
4176	2033	0.6569	0.7566	4336	1772	0.6821	0.7879	100	60.84
				4035	1573	0.6347	0.8117	1	
4103	1890	0.6454	0.7738	4336	1772	0.6821	0.7879	100	57.47

The results of the first interaction, in which all reference membership levels were set to 1 and the initial population was generated at random, are shown at the column for the first interaction in Table 1. While, the results of the second interaction, in which the reference membership levels were updated from $(1.0, 1.0)$ to $(0.9, 1.0)$ and the initial population was generated based on the revised method are shown at the column for the second interaction in Table 1. Moreover, the results of the second interaction and that of the third interaction, in which the reference membership levels were updated from $(0.9, 1.0)$ to $(0.85, 1.0)$ and the initial population was generated based on the revised method are shown at the corresponding columns in Table 2. Note that the first objective function $z_1(\boldsymbol{x})$ must be maximized and the second objective function $z_2(\boldsymbol{x})$ must be minimized.

As can be seen in the results of Tables 1 and 2, no solution after an interaction was dominated by that before an interaction and then all solutions obtained were Pareto optimal solutions. From the result, it is gathered that the proposed method which leaves an approximate optimal solution is efficient.

V. Conclusion

In this paper, an interactive fuzzy satisficing method, a hybrid between the interactive method and genetic algorithm, was proposed for multiobjective multidimensional 0-1 knapsack problems. With regard to genetic algorithms, a genetic algorithm with double strings which generates only feasible solutions was used. In case of combining the genetic algorithm to an interactive method simply, values of mean relative difference between approximate optimal solutions obtained through GA and strictly optimal solution were smaller than 1%, while some of approximate optimal solutions after an interaction were dominated by those before the interaction. Hence, a method of generating initial population was proposed. The results that the solutions after an interaction were not dominated by those before the interaction implies effectiveness of the proposed method.

References

[1] J. H. Holland: *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[2] D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.

[3] Z. Michalewicz: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992, Second, extended edition, 1994.

[4] M. Sakawa: *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, 1993.

[5] M. Sakawa et al.: "Fuzzy multiobjective combinatorial optimization through revised genetic algorithms", Journal of Japan Society for Fuzzy Theory and Systems, **6**, pp. 177-186, 1994 (in Japanese).