Control and Cybernetics

vol. **33** (2004) No. 4

A generalized varying-domain optimization method for fuzzy goal programming with priorities based on a genetic algorithm

by

Li Shaoyuan[1], Hu Chaofang and Teng Changjun

Institute of Automation
Shanghai Jiao Tong University
Shanghai, 200030, P.R. China
[1]e-mail: syli@sjtu.edu.cn

Abstract: This paper proposes a generalized domain optimization method for fuzzy goal programming with different priorities. According to the three possible styles of the objective function, the domain optimization method and its generalization are correspondingly proposed. This method can generate the results consistent with the decision-maker's priority expectations, according to which the goal with higher priority may have higher level of satisfaction. However, the reformulated optimization problem may be nonconvex for the reason of the nature of the original problem and the introduction of the varying-domain optimization method. It is possible to obtain a local optimal solution for nonconvex programming by the SQP algorithm. In order to get the global solution of the new programming problem, the co-evolutionary genetic algorithm, called GENOCOP III, is used instead of the SQP method. In this way the decision-maker can get the optimum of the optimization problem. We demonstrate the power of this proposed method based on genetic algorithm by illustrative examples.

Keywords: fuzzy goal programming, priorities, SQP, genetic algorithm, GENOCOP III.

1. Introduction

Goal programming (GP) is a useful method for decision-makers to consider simultaneously many goals in order to find a satisfactory solution. This method was introduced by Charnes and Cooper (1961). The first application quickly demonstrated its interest in a number of areas, and later numerous variants and a number of impressive applications followed. As a robust tool for multi-objective decision-making (MODM) problem, GP has been studied extensively for the past 35 years.

In the multi-criterion setting the special characteristic of GP models is the way the decision criteria is dealt with. Instead of the direct evaluation of the criteria outcomes, GP models explicitly introduce the desired target value and goal for each criterion, and optimize the deviations of the criteria outcomes from these goals. The solution depends on the metrics used for the deviations and as well as the method of weighting of the different goals. There are two common weighting methods. The first one is the fixed order of goals. In practice, this is implemented by searching a lexicographic minimum of the ordered deviation vector (Ijiri, 1965). The second one is the use of weights on goals and the minimization of the weighted sum of goal deviations (Charnes, Cooper, 1977). Sometimes, also the minimization of the maximum deviation is used as suggested by Flavell (1976). The GP approach to the multi-criterion problems has received increasing interest due to its modeling flexibility and conceptual simplicity (Hämäläinem, Mantysaari, 2002).

However, it is difficult for the decision-maker to determine precisely the goal value of each objective, since possibly only some partial information is known. To incorporate uncertainty and imprecision in the model formulation, the fuzzy set theory, initially proposed by Zadeh (1965) was introduced in the field of conventional decision-making (DM) problems, where aspiration levels of objectives are assigned in an imprecise manner. According to the fuzzy-set-based theory, the inaccurate objectives and constraints are represented by the associated membership functions (Bellman, Zadeh, 1970), for instance, the triangle-like or trapezoid-like membership functions. We call the inaccurate objectives and constraints fuzzy objectives and constraints. The concept of fuzzy programming (FP) on a general level was first proposed by Tanaka et al. (1974) in the framework of fuzzy decision of Bellman and Zadeh (1970). After that, the FP approach was applied to linear programming (LP) with several objectives (Zimmermann, 1978). Introducing fuzzy uncertainty and imprecision into the GP problems, Narasimhan (1980) initially proposed fuzzy goal programming (FGP), which gave rise to some related research (Rao, Tiwari, Mohanty, 1988; Rao, Sundararaju, Prakash 1992; Rubin, Narasimhan, 1984).

GP and FP are two approaches to solving the vector optimization problem by reducing it to a single (or sequential) one. Both of them need an aspiration level for each goal. These aspiration levels are determined either by the decision-maker or the decision analyst. In addition to the aspiration levels of the goals, FP needs admissible violation constant (or tolerance) for each goal. A larger violation of a goal indicates lower importance of this goal. It can be proved that every fuzzy linear programming problem has an equivalent weighted linear goal programming problem where the weights are the reciprocals of the admissible violation constants (Mohamed, 1997). In general, every FP is a GP with some weights assigned to the deviational variables in the objective function, where the FP has fuzziness in the aspiration levels, i.e. the problem is to get a solution that makes the objectives as close as possible to a specific goal within a certain limit. In this paper, we use the FP method to solve the GP problem with different priorities, and the results of the examples show the capacity of this method. Applying fuzzy set theory to GP has advantages of allowing for vague aspirations of the decision-makers, which can be qualified by some natural language terms. This approach has a more flexible framework, which allows the decision-makers to represent the information in a more direct way when they are unable to express it precisely. Thus, the burden of quantifying a qualitative concept is largely eliminated.

In a GP problem, it is practical to consider that there are different priorities of the goals, and a lot of research has been done on this problem. The priority structure is assigned by using different methods, such as fuzzy analytical hierarchy process (Saaty, 1978). Conventional delaminating FGP method (Chen, 1994; Tiwari, Mohanty, Rao, 1987; Tiwari, Dharmar, Rao, 1986) is used when the decision-makers has a priority order toward different goals. This approach categorizes fuzzy goals into k priority levels according to the decision-maker, where k is less than the number of the fuzzy goals. The k subproblems are solved in sequence, and the desirable membership values of the fuzzy goals belonging to the first priority levels are achieved foremost. Then these membership values are regarded as additional constraints of the inferior levels. But this method has low computation efficiency. Liang-Husan Chen and Feng-Chou Tsai (2001) proposed an approach using an additive model to solve this problem, and when the goals are assigned to different levels, the interrelations between the different memberships of the goals are added as crisp constraints. The reason that we do not apply this method is that: (1) the added constraints are too strict for solving the optimization problem and there may be no feasible solutions when the decision-maker requires a highly desirable value for a fuzzy goal, while considering the priority order; (2) in reality, the decision-maker has a limited ability to determine priorities and aspiration levels for goals or only a vague or imprecise knowledge about trade-off relationship among goals (Rasmy, 2002). Thus, priorities and goals are often fixed arbitrarily. Consequently, this model solution may be wrong. The interactive method has been introduced into this problem and studied by many scholars (Kato, Sakawa, 1998; Sakawa, Yauchi, 2001; Sakawa, Yano, 1989; Sakawa, Kato, Nishizaki, 2003). But it is not a practical way since the decision-maker must stand by and introduce important information on preferences at each step of the optimization process. The genetic algorithms (GAs) have also been also introduced into this problem area (Kato, Sakawa, 1998; Sakawa, Yauchi, 2001), though they are associated with a heavy burden of computations.

In this paper, the domain optimization method is also generalized. According to the other two possible styles of the objective functions, two other domain optimization methods are correspondingly proposed. However, the reformulated optimization problem may be nonconvex for the reason of the nature of the original problem and the introduction of the varying-domain optimization method. It is possible to obtain a local optimal solution for nonconvex programming by the SQP algorithm. In order to get the global solution of the new programming problem, the co-evolutionary genetic algorithm for numerical optimization of constrained problems, called GENOCOP III (Michalewicz, 1996; Michalewicz, Nazhiyath, 1995; Michalewicz, Schoenauer, 1996), is used instead of the SQP method. Section 2 describes the formulation of FGP with different priorities, Section 3 gives the generalized domain optimization method, GENOCOP III is introduced in Section 4, and the optimization algorithm is proposed in Section 5. Section 6 verifies the efficiency of the varying-domain optimization method through illustrative examples, a comparison of optimal solutions between GENOCOP III and SQP is provided there, and the conclusion is made in the last section.

2. The FGP with different priorities

2.1. The formulation of FGP

Denote by $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T \in R^n$ the decision vector and by $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})...,\ f_m(\boldsymbol{x}))$ the objective functions, while $G(\boldsymbol{x})$ is the system constraint. A goal value f_i^*, $(i = 1, 2, ..., m)$ is given for each objective and the GP model is formulated as follows:

$$\begin{aligned} & f_i(\boldsymbol{x}) \to f_i^*, \quad (i = 1, 2, ..., m) \\ & \boldsymbol{x} \in G(\boldsymbol{x}) \in R^n. \end{aligned} \tag{1}$$

Using the concept of fuzzy sets, the membership function of the objective functions can be defined based on the following steps given by Zimmermann (1978):

Step 1: For the following optimization model,

$$\begin{aligned} & \max f_i(\boldsymbol{x}), \quad i = 1, 2, ..., m \\ & \text{s.t. } \boldsymbol{x} \in G(\boldsymbol{x}) \in R^n \end{aligned} \tag{2}$$

let $\boldsymbol{x}^*$ be the optimal solution of the objective function $f_i(x)$, i.e. $f_i(\boldsymbol{x}^*) = f_i^{\max}$.

Step 2: Find $f_i(\boldsymbol{x}^*) = f_i^{\min}$, $\forall i$ with the same procedure as in Step 1.

Step 3: The definition of the membership functions $\mu_{f_i}(\boldsymbol{x})$, $(i = 1, 2, ..., m)$ is now as follows:

$$\mu_{f_i}(\boldsymbol{x}) = \begin{cases} \dfrac{f_i^{\max} - f_i(\boldsymbol{x})}{f_i^{\max} - f_i^*}, & f_i^* < f_i(\boldsymbol{x}) \le f_i^{\max} \\ 1, & f_i(\boldsymbol{x}) = f_i^* \\ \dfrac{f_i(\boldsymbol{x}) - f_i^{\min}}{f_i^* - f_i^{\min}}, & f_i^{\min} \le f_i(\boldsymbol{x}) < f_i^* \ . \end{cases} \tag{3}$$

The corresponding graph of $\mu_{f_i}(\boldsymbol{x})$ is showen as Fig. 1.

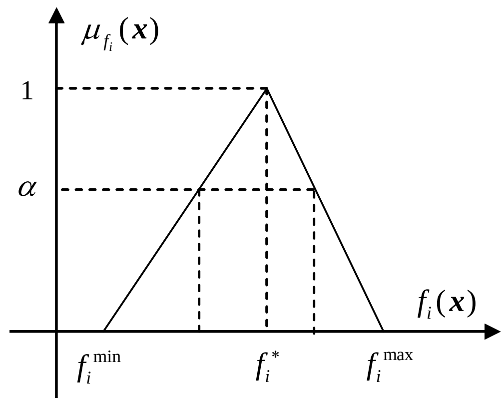

Figure 1. The membership function of $\mu_{f_i}(\boldsymbol{x})$

Denote by $F_i(\boldsymbol{x})$, $(i = 1, 2, ..., m)$ the corresponding set that has the following form:

$$F_i(\boldsymbol{x}) = \{\boldsymbol{x} | f_i^{\min} \leq f_i(\boldsymbol{x}) \leq f_i^{\max},\ \boldsymbol{x} \in R^n\}, \quad (i = 1, 2, ..., m).$$

Then the α-level sets of $F_i^\alpha(\boldsymbol{x})$, $(i = 1, 2, ..., m)$, are defined as:

$$F_i^\alpha(\boldsymbol{x}) = \{\boldsymbol{x} | \alpha \leq \mu_{f_i}(\boldsymbol{x}),\ 0 < \alpha \leq 1,\ \boldsymbol{x} \in F_i(\boldsymbol{x})\}, \quad (i = 1, 2, ..., m) \tag{4}$$

and from them we can get the α-level set of $F_i^\alpha(\boldsymbol{x})$:

$$F(\alpha, \boldsymbol{x}) = F^\alpha(\boldsymbol{x}) = F_1^\alpha \cap F_2^\alpha \cap ... \cap F_m^\alpha\ . \tag{5}$$

The FGP model proposed by Yang et al. (1991) is expressed as:

$$\left.\begin{array}{l} \text{Find } \boldsymbol{x}^* \text{ such that} \\ \max \alpha \\ \text{subject to } \boldsymbol{x}^* \in F(\alpha, \boldsymbol{x}) \cap G(\boldsymbol{x}) \end{array}\right\} \tag{6}$$

where $G(\boldsymbol{x})$ is system constraint from the original programming problem.

2.2. The formulation of FGP with different priorities

In the GP problems, the decision-maker usually has a preemptive priority for achieving goals. That is, some goals have a higher priority for their achievement than the other under system constraints. Suppose that the priority of objective $f_i(\boldsymbol{x})$ is higher than that of objective $f_{i-1}(\boldsymbol{x})$ for all $\boldsymbol{x}$, which is denoted as

$$f_{i-1}(\boldsymbol{x}) \prec f_i(\boldsymbol{x})\ . \tag{7}$$

If these two or more objectives have the same priority, we denote this as

$$f_{i-1}(\boldsymbol{x}) \sim f_i(\boldsymbol{x}). \tag{8}$$

For the convenience of expression, we assume that for the objectives, we have

$$f_{i-1}(\boldsymbol{x}) \prec f_i(\boldsymbol{x}),\ i = 2, 3, ..., m\ . \tag{9}$$

It is reasonable for us to hope that objectives with higher priorities will also have higher degree of satisfaction. If $\boldsymbol{x}^*$ is a solution to the fuzzy goal programming problem with multiple priority, then by using the concept of α-level membership functions, the additional conditions of priority can be described as follows (Chen, Tsai, 2001):

$$\mu_{f_1}(\boldsymbol{x}^*) \leq \mu_{f_2}(\boldsymbol{x}^*) \leq\ ...\ \leq \mu_{f_m}(\boldsymbol{x}^*)\ . \tag{10}$$

In Chen and Tsai (2001), the FGP with different priorities is formulated as follows:

$$\left.\begin{array}{l} \text{Find } \boldsymbol{x}^* \text{ such that} \\ \max\ \displaystyle\sum_{i=1}^{m} \mu_{f_i}(\boldsymbol{x}) \\ \text{s.t.} \begin{cases} \boldsymbol{x} \in F^{\alpha}(\boldsymbol{x}) \\ \text{the system constraints} \\ \mu_{f_1}(\boldsymbol{x}) \leq \mu_{f_2}(\boldsymbol{x}) \leq\ ...\ \leq \mu_{f_m}(\boldsymbol{x}) \end{cases} \end{array}\right\}. \tag{11}$$

In the expression (11), the added constraints expressed as (10) maybe too strict for solving the optimization problem and there may be no feasible solutions when the decision-maker requires a high desirable achievement degree for a fuzzy goal, when considering the priority order.

Here we use another model to formulate the FGP with different priorities. First comes the preference of solutions. Given two solutions $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$ of this problem which satisfy

$$\begin{array}{l} \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \in F(\alpha, \boldsymbol{x}) \cap G(\alpha, \boldsymbol{x}),\ (0 \leq \alpha \leq 1) \\ \mu_{f_1}(\boldsymbol{x}^{(1)}) \leq \mu_{f_2}(\boldsymbol{x}^{(1)} \leq ... \leq \mu_{f_m}(\boldsymbol{x}^{(1)}) \\ \mu_{f_1}(\boldsymbol{x}^{(2)}) \leq \mu_{f_2}(\boldsymbol{x}^{(2)}) \leq ... \leq \mu_{f_m}(\boldsymbol{x}^{(2)}) \end{array} \tag{12}$$

we say that $\boldsymbol{x}^{(1)}$ is more preferable than $\boldsymbol{x}^{(2)}$ if there exists $k \in [1, n]$, which satisfies

$$\left.\begin{array}{l} \mu_{f_i}(\boldsymbol{x}^{(1)}) = \mu_{f_i}(\boldsymbol{x}^{(2)}), \quad (\text{for all } i > k) \\ \mu_{f_k}(\boldsymbol{x}^{(1)}) > \mu_{f_k}(\boldsymbol{x}^{(2)}) \end{array}\right\}. \tag{13}$$

So the FGP with different priorities is formulated as:

$$\left.\begin{array}{l} \text{Find } \boldsymbol{x}^* \text{ such that} \\ \max \alpha \\ \text{s.t.} \begin{cases} \forall \boldsymbol{x}^* \in F(\alpha, \boldsymbol{x}) \cap G(\boldsymbol{x}), \ (0 \le \alpha \le 1) \\ \mu_{f_1}(\boldsymbol{x}^*) \le \mu_{f_2}(\boldsymbol{x}^*) \le \dots \le \mu_{f_m}(\boldsymbol{x}^*) \end{cases} \end{array}\right\} . \tag{14}$$

Then comes the same drawback as described by (11). In the next section we introduce a new domain optimization method to solve this problem.

3. The domain optimization method and its generalization

3.1. The domain optimization method

For FGP with different priorities as expressed in (14), with objective $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_m(\boldsymbol{x}))$ and their corresponding goal values f_i^*, $(i = 1, 2, ..., m)$ and the membership functions defined as (3) (shown in Fig. 1), we do not use the constant domain $[f_i^{\min}, f_i^{\max}]$ for every objective f_i $(i = 1, 2, ..., m)$, but introduce new variables and use the varying domain $[\beta_i^{\min}, \beta_i^{\max}]$. In fact, by rewriting the objective functions $f_i(\boldsymbol{x})$ with their goal values f_i^* as $f_i(\boldsymbol{x}) - f_i^* \to f_i'(\boldsymbol{x})$, we can revise the objectives so that they have the same goal value 0. Because selection of domains is a very flexible process, we multiply them with some factors and simply get the same symmetric domain $[-1, 1]$ for each objective.

For example

$$f_1(\boldsymbol{x}) \to 2 \quad \text{and} \quad f_1(\boldsymbol{x}) \in [0, 4], \quad \text{then} \quad f_1'(\boldsymbol{x}) = \frac{1}{2}(f_1(\boldsymbol{x}) - 2)$$

and

$$f_2(\boldsymbol{x}) \to 4 \quad \text{and} \quad f_2(\boldsymbol{x}) \in [1, 7], \quad \text{then} \quad f_2'(\boldsymbol{x}) = \frac{1}{3}(f_2(\boldsymbol{x}) - 4) \ .$$

The new objectives $f_1'(\boldsymbol{x})$ and $f_2'(\boldsymbol{x})$ have the goal 0 and the same symmetric domain $[-1, 1]$. In this way we can normalize the objective functions. It can be easily proved that $\mu_{f_i}(\boldsymbol{x})$ and $\mu_{f_i'}(\boldsymbol{x})$ have the same value for the same $\boldsymbol{x}$.

After the modification of the objectives, we can select $[-\beta_i, \beta_i]$, $\beta_i \in [0, 1]$, $i = 1, 2, ..., m$ as the varying domains.

Since $f_1(\boldsymbol{x})$ is the least important objective, let $\alpha = \mu_f(\boldsymbol{x}) = \mu_{f_1}(\boldsymbol{x}) = \mu_{f_1'}(\boldsymbol{x})$ and $\beta_1 = 1$. Now we consider getting a solution $\boldsymbol{x}^*$ that maximizes α and satisfies the condition $\mu_{f_1}(\boldsymbol{x}^*) \le \mu_{f_2}(\boldsymbol{x}^*) \le \dots \le \mu_{f_m}(\boldsymbol{x}^*)$ in a probability perspective.

Enlightened by the epistemic utility function introduced in Goodrich et al. (1998), we introduce a variable γ and rewrite the problem (14) as follows:

Find $\boldsymbol{x}^*$ such that

$$\left.\begin{array}{l} \max(\alpha-\lambda\cdot\gamma) \\ \text{s.t.}\left\{\begin{array}{l} \alpha\beta_i-\beta_i\leq f_i'(\boldsymbol{x})\leq\beta_i-\alpha\beta_i \\ \beta_i-\beta_{i-1}\leq\gamma,\quad \beta_i\in[0,1],\quad (i=2,3,...,m) \\ \forall\boldsymbol{x}\in G(\boldsymbol{x}),\quad (0\leq\alpha\leq1),\quad 0<\lambda \end{array}\right. \end{array}\right\} \tag{15}$$

where $\alpha\beta_i-\beta_i\leq f_i'(\boldsymbol{x})\leq\beta_i-\alpha\beta_i$, $(i=2,3,...,m)$ is the transformation of $\boldsymbol{x}\in F(\alpha,\boldsymbol{x})$.

We solve the above optimization problem with some optimization method and obtain a solution. If in the solution we get $\gamma^*\leq 0$ then we have

$$\beta_i^*\leq\beta_{i-1}^*,\quad (i=2,...,m)$$

and

$$\alpha^*\beta_i^*-\beta_i^*\leq f_i'(\boldsymbol{x}^*)\leq\beta_i^*-\alpha^*\beta_i^*,\quad (i=1,2,3,...,m)$$

as shown in Fig. 2.

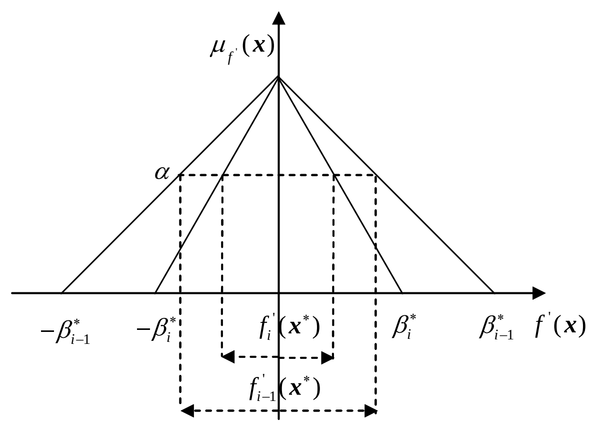

Figure 2. Two-side domain optimization

It can be easily seen from the domain illustration that the final point $f_i(\boldsymbol{x}^*)$ might be in a smaller interval than that of $f_{i-1}(\boldsymbol{x}^*)$ and is nearer to the goal value 0. We call this "domain optimization". If the values of $f_{i-1}(\boldsymbol{x}^*)$ and $f_i(\boldsymbol{x}^*)$ are random in their final domains, then we satisfy the priority request of

$$\mu_{f_{i-1}}(\boldsymbol{x}^*)\leq\mu_{f_i}(\boldsymbol{x}^*),\ i=2,...,m \tag{16}$$

in a probability perspective.

To get an initial feasible solution for a certain iterative algorithm meant to solve the optimization problem (15), we can choose $\gamma^0 > 0$. But if $\gamma^* > 0$ in the solution we obtain, then the priority order cannot be preserved during the solving process, and we assign the parameter λ a larger value. If, ultimately, $\gamma^* > 0$, the priority structure proposed by the decision-maker may have turned out unreasonable.

Denote

$$\underline{\mu}_{f_i}(\boldsymbol{x}) = \inf(\mu_{f_i}(\boldsymbol{x})) \leq \mu_{f_i}(\boldsymbol{x}^*), \quad f(\boldsymbol{x}) \in [-\alpha^*\beta_i^* - \beta_i^*, \beta_i^* - \alpha^*\beta_i^*] \,.$$

Then it is obvious that

$$\underline{\mu}_{f_i}(\boldsymbol{x}) = 1 - (\beta_i^* - \alpha^*\beta_i^*) = 1 - (1-\alpha^*)\beta_i^* \quad (i = 1, 2, ..., m) \,.$$

Since

$$\beta_i^* \leq \beta_{i-1}^*, \ (i = 2, 3, ..., m)$$

then (16) can be achieved. Consequently, we can get

$$\underline{\mu}_{f_1}(\boldsymbol{x}) \leq \underline{\mu}_{f_2}(\boldsymbol{x}) \leq ... \leq \underline{\mu}_{f_m}(\boldsymbol{x}) \,. \tag{17}$$

In this paper, we do not apply the strict constraints expressed as (10) but only satisfy the expression (17). Thus we allow a certain tolerance of the priority order condition, the solving process may be more flexible and the solution for each objective may feature a higher membership.

The parameter λ in (15), as the result of a satisfactory decision (Goodrich, Stirling, Frost, 1998), shows the balance between optimization and priority order of the objectives. If $\lambda \to 0$, then the final solution we obtain can approximate or reach the goal very well, but the priority order might be seriously violated. If $\lambda \to \infty$, then we will get the solution that satisfies the priority order very well but some of the objectives may deviate significantly from their goal values.

3.2. The generalization of the domain optimization method

Actually, in distinction from the expression (3), the membership functions of the objectives may have the following forms:

$$\mu_{f_i}(\boldsymbol{x}) = \begin{cases} 1 & f_i(\boldsymbol{x}) \leq f_i^* \\ \dfrac{U_i - f_i(\boldsymbol{x})}{U_i - f_i^*} & f_i^* \leq f_i(\boldsymbol{x}) \leq U_i \\ 0 & f_i(\boldsymbol{x}) \geq U_i \end{cases} \tag{18.1}$$

or

$$\mu_{f_i}(\boldsymbol{x}) = \begin{cases} 1 & f_i(\boldsymbol{x}) \geq f_i^* \\ \dfrac{f_i(\boldsymbol{x}) - L_i}{f_i^* - L_i} & L_i \leq f_i(\boldsymbol{x}) \leq f_i^* \\ 0 & f_i(\boldsymbol{x}) \geq L_i \end{cases} \,. \tag{18.2}$$

The shape of $\mu_{f_i}(\boldsymbol{x})$ is shown in Fig. 3.

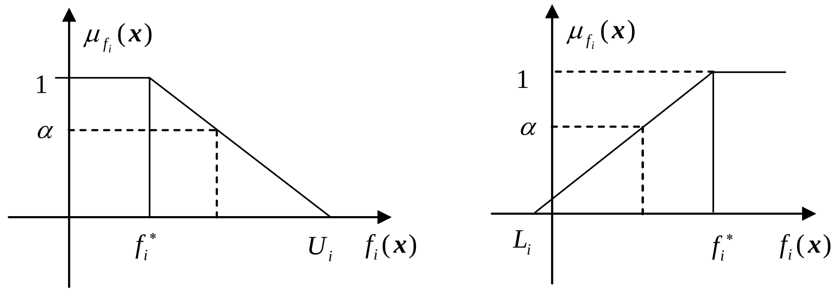

Figure 3. The membership function of $\mu_{f_i}(\boldsymbol{x})$

Considering that the membership functions of the objectives may take one of the above two forms, or the hybrid forms involving (3), (18.1), (18.2), it is necessary that we generalize the domain optimization method.

3.2.1. The single-side domain optimization method

When the goal value f_i^* is the minimum of $f_i(\boldsymbol{x})$ (a usual case in the multi-objective minimization problem), or less than $f_i^{\min}$, i.e., the objective functions all have the membership functions like (18.1) (since the maximization of $f_i(\boldsymbol{x})$ equals the minimization of $-f_i(\boldsymbol{x})$, the case like (18.2) can be converted to (18.1)), then the *single-side domain optimization method* is proposed as follows:

$$\left.\begin{array}{l}\text{Find } \boldsymbol{x}^* \text{ such that} \\ \max(\alpha - \lambda \cdot \gamma) \\ \text{s.t.} \begin{cases} 0 \le f_i'(\boldsymbol{x}) \le \beta_i - \alpha\beta_i \\ \beta_i - \beta_{i-1} \le \gamma, \quad \beta_i \in [0,1], \quad (i = 2, 3, ..., m) \\ \forall \boldsymbol{x} \in F(\alpha, \boldsymbol{x}) \cap G(\boldsymbol{x}), \ (0 \le \alpha \le 1), \ \lambda > 0 \end{cases}\end{array}\right\}. \tag{19}$$

Like in the two-side domain optimization method, if in the solution we get $\gamma^* \le 0$, we also have

$$\beta_i^* \le \beta_{i-1}^* \ (i = 2, 3, ..., m)$$

and

$$0 \le f_i'(\boldsymbol{x}^*) \le \beta_i^* - \alpha^* \beta_i^* \quad (i = 1, 2, 3, ..., m)$$

as shown in Fig. 4.

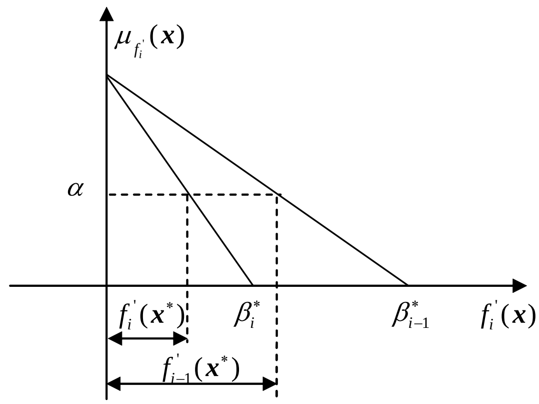

Figure 4. Single-side domain optimization

3.2.2. The hybrid-side domain optimization method

When a hybrid case appears, i.e., some goal values of the objective functions lie in the domain, $[f_i^{\min}, f_i^{\max}]$, while the other ones follow the forms of (18.1) and/or (18.2), we propose the hybrid-side domain optimization method as follows:

$$\left.\begin{array}{l} \text{Find } \boldsymbol{x}^* \text{ such that} \\ \max(\alpha - \lambda \cdot \gamma \\ \text{s.t.} \left\{ \begin{array}{l} 0 \le f_i'(\boldsymbol{x}) \le \beta_i - \alpha\beta_i, \quad i \in I_1 \\ \alpha\beta_i - \beta_i \le f_i'(\boldsymbol{x}) \le \beta_i - \alpha\beta, \quad i \in I_2 \\ \beta_i - \beta_{i-1} \le \gamma,\ \beta_i \in [0,1],\ (i = 2,3,...,m) \\ \forall \boldsymbol{x} \in G(\boldsymbol{x}),\ (0 \le \alpha \le 1),\ \lambda > 0 \end{array} \right. \end{array} \right\} \tag{20}$$

where I_1 is the set of the goal values of the objective functions conforming to (18.1) and/or (18.2), and I_2 is the set of the goal values from the domain $[f_i^{\min}, f_i^{\max}]$.

Like before, if in the solution we get $\gamma^* \le 0$, then we have

$$\beta_i^* \le \beta_{i-1}^*,\ (i = 2,3,...,m)$$

as shown in Fig. 5.

One of the difficulties associated with the optimization problems (15), (19) or (20) is that the respective problem may be nonconvex, especially when some of the constraints are nonconvex functions. Then the optimization problem (15), (19) or (20) is a complex nonconvex programming problem and the standard solving methods such as SQP may only get a local optimum. In order to

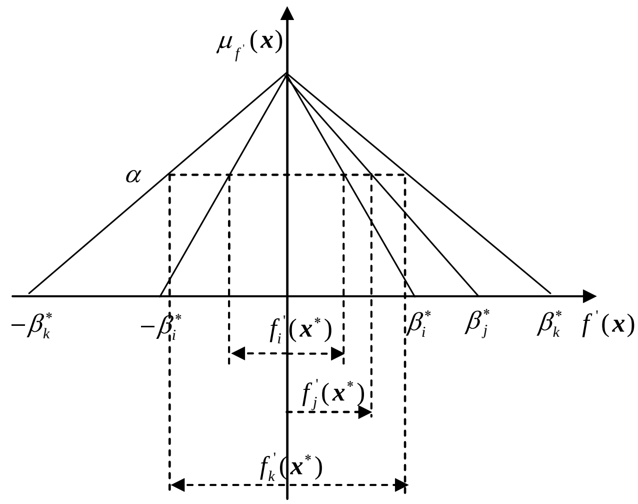

Figure 5. Hybrid-side domain optimization method

obtain the global optimum, GENOCOP III, a GA is introduced to solve the optimization problems (15), (19) and (20).

4. The optimization algorithms based on GENOCOP III

4.1. Genetic algorithm - GENOCOP III

Because the optimization problem (15), (19) or (20) is nonlinear, the conventional solving method is the Sequential Quadratic Programming (SQP) (Chen, 1994). SQP is an iterative procedure, which solves a Quadratic Programming (QP) problem at each iteration. As a classical method for constrained nonlinear optimization, it is based on Kuhn-Tucker (KT) conditions. When the optimization problem is convex, Kuhn-Tucker (KT) equation is the sufficient and necessary condition of the extremum problems with constraints. However, KT equation is only the necessary condition for the nonconvex programming problem, and the result may be a local solution. If the reformulated optimization problem is nonconvex for the reason of the original problem and the varying-domain optimization method, then the algorithms to get the global optimum, proposed in this paper, are GAs.

4.1.1. Overview of GAs

Genetic algorithms (GAs) proposed by Holland (1975) are very efficient global optimization methods, and belong to the family of the optimization techniques that are inspired by the mechanism of evolution and natural genetics. From the optimization point of view, they represent the random search techniques with a better kind of post search activities. They are robust in nature and applicable to a wide range of problems. GAs can converge in cases for which classical solutions come up with the problem of instability or do not converge at all. They have been extensively used in a wide variety of applications, such as manufacturing (Jensen, 2003; Pongcharoen, Hicks, Braiden, 2004; Wu et al., 2004), network optimization (Chou, Premkumar, Chu, 2001; Buczak, Wang, 2001), economy (Arifovic, 199; Ceylan, Öztürk, 2004), etc.

GAs are basically composed of three main operators: reproduction, crossover and mutation. Their principle is outlined in Fig. 6. Starting from an initial population, each iteration or generation consists in choosing some pairs of parent chromosomes for "mating". The reproduction process generates the offspring through crossing-over and mutations. The new generation of the offspring chromosomes obtained in this way offers solutions, whose adaptation to the problem considered is better. The algorithm stops as soon as the pertinence of solutions ceases to improve.

Initializing: randomly create a population
Evaluating: evaluate the pertinence of all the individuals
While the process is envolving:
    Select a subset of the population
    Recombine the parents' genes of the subset
    Mutate some genes of the subset
    Evaluate the subset and **replace** the original population with it
End

Figure 6. The principles of the genetic algoritm

4.1.2. GENOCOP III

As a co-evolutionary genetic algorithm, GENOCOP was proposed to solve Constrainted Optimization Problems (COPs) by Michalewicz (1996) and Michalewicz, Schoenauer (1996). GENOCOP III, according to the idea of repair algorithms, unlike the methods based on penalty function, is a revised version of GENOCOP. It is a very effective method especially for handing the general nonlinear programming problems with nonlinear constraints (Michalewicz, Nazhiyath, 1995). GENOCOP III incorporates the original GENOCOP system for solving the linear constraints, but extends it by maintaining two separate populations, where a development in one population influences evaluations of individuals in the other population. The first population, P_s, consists of the so-called search points that satisfy the linear constraints of the problem (as in the original GENOCOP system). The second population, P_r, consists of the so-called reference points that satisfy all of the constraints of the problem.

In GENOCOP III, for the purpose of initialization, an initial reference point is assumed to be generated randomly from individuals satisfying the lower and upper bounds. If GENOCOP III has difficulties in locating such a reference point, the user is prompted for it. In case where the ratio ρ between the sizes of the feasible and the whole search spaces is very small, it may happen that the initial set of reference points consist of a multiple copies of a single feasible point.

GENOCOP III uses the objective function for evaluation of fully feasible individuals (reference points) only, so the evaluation function is not distorted as in the penalty based methods. The infeasible search points are repaired for evaluation. Suppose that a search point $\boldsymbol{s}$ is not fully feasible, then the repair process works as follows:

1. Select one reference point $\boldsymbol{r} \in P_{\boldsymbol{r}}$.
2. Create random points $\boldsymbol{z}$ from a segment between $\boldsymbol{s}$ and $\boldsymbol{r}$ by generating random numbers from the range $< 0, 1 >$: $z = a\boldsymbol{s} + (1 - a)\boldsymbol{r}$.
3. Once a feasible $\boldsymbol{z}$ is found and the evaluation of $\boldsymbol{z}$ is better than that of $\boldsymbol{r}$, then replace $\boldsymbol{z}$ by $\boldsymbol{r}$ as a new reference point. Also replace $\boldsymbol{s}$ by $\boldsymbol{z}$ with some probability of replacement p_r.

In Sakawa, Yauchi (2001), the revised GENOCOP III is proposed to improve the computational efficiency and to find the initial feasible solution. If it is hard to find an initial feasible solution, information from Sakawa and Yauchi (2001) may be of use.

Note: The preliminary version of GENOCOP III is available from ftp.uncc.edu/coe/evol/genocopIII.tar.Z.

4.2. Optimization algorithm based on GENOCOP III

Based on the generalized domain optimization method and GENOCOP III, we propose the following algorithm for goal programming with different priorities:

Step 1: Modify the objectives so that they have the uniform goal value 0 and domains $[-1, 1]$ and/or $[0, 1]$.

Step 2: According to the priority order, form the optimization problem expressed as (15), (19) or (20).

Step 3: Using GENOCOP III solve the optimization problem formulated in Step 2.

Step 4: If $\gamma^* > 0$, then assign the parameter λ a larger value, go to Step 3.

5. Numerical examples

In this section, the results of SQP and GENOCOP III in solving the following numerical examples via transformation of the varying-domain optimization method are given. The comparison between the optimal solutions by these algorithms is provided. The results show not only the efficiency of the varying-domain optimization method proposed in this paper, but also the superiority of GENOCOP III to SQP for nonconvex optimization problem.

EXAMPLE 5.1 *(Sakawa, Yano, 1989)*

$$\begin{aligned}
&\min f_1(\boldsymbol{x}) = (x_1+5)^2 + 4x_2^2 + 2(x_3-50)^2\\
&\min f_2(\boldsymbol{x}) = 2(x_1-45)^2 + (x_2+15)^2 + 3(x_3+20)^2\\
&\max f_3(\boldsymbol{x}) = 3(x_1+20)^2 + 5(x_2-45)^2 + (x_2+15)^2\\
&\text{s.t.}\ \ x_1^2 + x_2^2 + x_3^2 \le 100\\
&\qquad 0 \le x_i \le 10, \quad i = 1,2,3.
\end{aligned}$$

The priority structure is:

$$f_3(\boldsymbol{x}) \succ f_1(\boldsymbol{x}) \succ f_2(\boldsymbol{x})$$

Step 1: Calculate the respective minimum or maximum of the objective functions, the intervals are [3225, 5433], [3875, 7002] and [7550, 13078], and therefore transformation of the objective functions can be expressed as

$$\begin{aligned}
f_1'(\boldsymbol{x}) &= \frac{f_1(\boldsymbol{x}) - 3225}{2208} \in [0,1]\\
f_2'(\boldsymbol{x}) &= \frac{f_2(\boldsymbol{x}) - 3875}{3127} \in [0,1]\\
f_3'(\boldsymbol{x}) &= \frac{f_3(\boldsymbol{x}) - 13078}{5528} \in [-1,0].
\end{aligned}$$

Step 2: Reformulate the optimization expression using the single-side domain optimization method:

$$\max(\alpha - \lambda\cdot\gamma)$$
$$\text{s.t.}\begin{cases}
0 \le f_1'(\boldsymbol{x}) \le (1-\alpha)\beta_1\\
0 \le f_2'(\boldsymbol{x}) \le (1-\alpha)\\
-(1-\alpha)\beta_3 \le f_3'(\boldsymbol{x}) \le 0\\
\beta_1 \le 1+\gamma\\
\beta_3 - \beta_1 \le \gamma\\
x_1^2 + x_2^2 + x_3^2 \le 100\\
0 \le \alpha,\ \beta_2, \beta_3 \le 1\\
-1 \le \gamma \le 1\\
0 \le x_i \le 10,\ i = 1,2,3.
\end{cases}$$

Step 3: Use SQP and GENOCOP III to solve above programming problem. For different values of the parameter λ, the respective solutions are obtained. The results of SQP are given in Table 1. The results of GENOCOP III are shown in Table 2.

From Table 1 and Table 2, we can see that the results of SQP and GENOCOP III satisfy the requirements of the decision-maker.

Table 1. Results for different λ values using SQP

λ	x_1	x_2	x_3	μ_1	μ_2	μ_3
0.5	6.9182	0.0000	7.2207	0.7386	0.5288	0.9484
1	6.9182	0.0000	7.2207	0.7386	0.5288	0.9484
2	6.9182	0.0000	7.2207	0.7386	0.5288	0.9484

Table 2. Results for different λ values using GENOCOP III

λ	x_1	x_2	x_3	μ_1	μ_2	μ_3
0.5	7.4689	0.0711	6.5413	0.6794	0.5898	0.9535
1	7.4166	0.1391	6.3846	0.6676	0.5946	0.9452
2	6.9549	0.0510	6.5535	0.6861	0.5645	0.9400

REMARK 5.1 *Since the nonconvex nonlinear character of this problem is not obvious, both SQP and GENOCOP III can be used to solve it. And the satisfactory results are also obtained. However, SQP is not sensitive to the adjustment of λ because the solutions are very similar and they may be local. On the contrary, GENOCOP III not only satisfies the priority order $f_3(\boldsymbol{x}) \succ f_1(\boldsymbol{x}) \succ f_2(\boldsymbol{x})$, but also is very sensitive to the adjustment of λ according to the requirements, and the satisfactory solutions can be obtained successfully.*

EXAMPLE 5.2 *(Sakawa and Yauchi, 2001)*

$$\begin{aligned}
\min\ & f_1(\boldsymbol{x}) = 7x_1^2 - x_2^2 + x_1x_2 - 14x_1 - 16x_2 + 8(x_3-10)^2 + 4(x_4-5)^2 \\
& +(x_5-3)^2 + 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + x_{10}^2 + 45 \\
\min\ & f_2(\boldsymbol{x}) = (x_1-5)^2 + 5(x_2-12)^2 + 0.5x_3^4 + 3(x_4-11)^2 + 0.2x_5^5 + 7x_6^2 \\
& +0.1x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 + x_8^2 + 3(x_9-5)^2 + (x_{10}-5)^2 \\
\min\ & f_3(\boldsymbol{x}) = x_1^3 + (x_2-5)^2 + 3(x_3-9)^2 - 12x_3 + 2x_4^3 + 4x_5^2 + (x_6-5)^2 \\
& +6x_7^2 + 3(x_7-2)x_8^2 - x_9x_{10} + 4x_9^3 + 5x_1 - 8x_1x_7
\end{aligned}$$

The system constraints are:

$$\begin{aligned}
& -3(x_1-2)^2 - 4(x_2-3)^2 - 2x_3^2 + 7x_4 - 2x_5x_6x_8 + 120 \geq 0 \\
& -5x_1^2 - 8x_2 - (x_3-6)^2 + 2x_4 + 40 \geq 0 \\
& -x_1^2 - 2(x_2-2)^2 + 2x_1x_2 - 14x_5 - 6x_5x_6 \geq 0 \\
& -0.5(x_1-8)^2 - 2(x_2-4)^2 - 3x_5^2 + x_5x_8 + 30 \geq 0 \\
& 3x_1 - 6x_2 - 12(x_9-8)^2 + 7x_{10} \geq 0 \\
& 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 105 \\
& 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
& -8x_1 + 2x_2 + 5x_9 - 2x_{10} \leq 12 \\
& -5.0 \leq x_i \leq 10, \quad i = 1, ..., 10.
\end{aligned}$$

The priority structure is:

$$f_2(\boldsymbol{x}) \succ f_3(\boldsymbol{x}) \succ f_1(\boldsymbol{x}).$$

Note: Due to the semi-positive definiteness of the criterion function's Hessian matrix, this example is obviously a nonconvex multi-objective problem. The KT conditions and the SQP are improper for soving this problem. Consequently, GENOCOP III is used to find the global solution. Also the computations for each objective will be conduct with this algorithm.

Step 1: Calculate the respective minimum and maximum of the objective functions by GENOCOP III, to obtain the interval of each function: [89, 3437], [314, 7507] and [307, 9000]. Therefore, define the following membership functions:

$$f_1'(\boldsymbol{x}) = \frac{f_1(\boldsymbol{x}) - 89}{3348}$$
$$f_2'(\boldsymbol{x}) = \frac{f_2(\boldsymbol{x}) - 314}{7193}$$
$$f_3'(\boldsymbol{x}) = \frac{f_3(\boldsymbol{x}) - 307}{8693}\,.$$

The initial feasible solution is chosen as $(x_1, ..., x_{10}) = (2, 2, 8, 6, 1, -2, 1, 6, 8, 8)$.

Step 2: Formulate the optimization problem using the single-side domain optimization version:

$$\max(\alpha - \lambda \cdot \gamma)$$
$$\text{s.t.} \begin{cases} 0 \le f_1'(\boldsymbol{x}) \le 1 - \alpha \\ 0 \le f_2'(\boldsymbol{x}) \le (1-\alpha)\beta_2 \\ 0 \le f_3'(\boldsymbol{x}) \le (1-\alpha)\beta_3 \\ \beta_3 \le 1 + \gamma \\ \beta_2 - \beta_3 \le \gamma \\ \text{system constraints} \\ 0 \le \alpha, \quad \beta_2, \beta_3 \le 1 \\ -1 \le \gamma \le 1 \\ -5 \le x_i \le 10, \quad 1 \le i \le 10\,. \end{cases}$$

Step 3: Use GENOCOP III to solve the above programming problem again. When $\lambda = 1$, we get

$$\begin{cases} \boldsymbol{x} = (1.9421\ 2.8821\ 5.3164\ 5.6807\ 0.2245\ 1.2515\ 1.1823\ 6.1321\ 6.1178\ 7.7145) \\ f_i' = (0.1057\ 0.0872\ 0.0965) \\ \mu_{f_i}(\boldsymbol{x}) = (0.8943\ 0.9128\ 0.9035)\,. \end{cases}$$

It can be seen that the priority order $f_2(\boldsymbol{x}) \succ f_3(\boldsymbol{x}) \succ f_1(\boldsymbol{x})$ is satisfied.

Remark 5.2 *Because of the obviously nonconvex nonlinear nature, SQP cannot be used to find the final solution. This particular problem was solved using only GENOCOP III. The satisfactory results were obtained, and the priority requirement was realized. It appears, therefore, that GENOCOP III is suitable to deal with strongly nonconvex nonlinear optimization with nonlinear constraints.*

6. Conclusions

In this paper, the GP problem with different priorities is solved through the generalized domain optimization method and GENOCOP III genetic algorithm is used to solve the nonconvex programming problem formulated on the basis of the original problem and the generalized domain optimization method. According to different positions of the goal values, three domain optimization versions are presented to overcome the respective problem. The method proposed in this paper can solve GP (multi-objective programming) problems with priorities and complex objective functions as well as constraints. The optimization results of the examples show the efficiency of generalized domain optimization method and GENOCOP III, when used instead of SQP. It can be used in complicated real-word decision-making problems and offers a promising prospect.

Acknowledgements

This work was supported by the National Nature Science Foundation of China (Grant: 60474051) and the High Technology Research and Development Program of China (Grant : 2002AA412130). The authors are grateful to anonymous reviewers for their valuable recommendations.

References

Arifovic, J. (1995) Genetic algorithms and inflationary economies. *Journal of Monetary Economics* **36**, 219-243.

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

Buczak, A.L. and Wang, H. (2001) Genetic algorithm convergence study for sensor network optimization. *Information Sciences* **133**, 267-282.

Ceylan, H. and Öztürk, H.K. (2004) Estimating energy demand of Turkey based on economic indicators using genetic algorithm approach. *Energy Conversion and Management* **45**, 2525-2537.

Charnes, A. and Cooper, W.W. (1961) *Management Models and Industrial Applications of Linear Programming.* John Wiley and Sons, New York.

Charnes, A. and Cooper, W.W. (1977) Goal programming and multi-objective optimization. *European Journal of Operational Research* **1** (1), 39-45.

CHEN, H.K. (1994) A note on a fuzzy goal programming algorithms by Tiwari Dharmar and Rao. *Fuzzy Sets and Systems* **62**, 287-290.

CHEN, L.H. and TSAI, F.C. (2001) Fuzzy goal programming with different importance and priorities. *European Journal of Operational Research* **133**, 548-556.

CHOU, H.H., PREMKUMAR, G. and CHU, C.H. (2001) Genetic algorithms for communications network design - an empirical study of the factors that influence performance. *IEEE Transactions on evolutionary comutation* **5** (3), 236-249.

FLAVELL, R.B. (1976) A new goal programming formulation. *Omega* **4**, 731-732.

GOODRICH, M.A., STIRLING, W.C. and FROST, R.L. (1998) A theory of satisficing decisions and control. *IEEE Transactions on Systems, Man and Cybernetics-Part A* **28** (6), 763-779.

HÄMÄLÄINEN, R.P. and MANTYSAARI, J. (2002) Dynamic multi-objective heating optimization. *European Journal of Operational Research* **142**, 1-15.

HOLLAND, J. H. (1975) *Adaptation in Natural and Artificial Systems.* Univ. of Michigan Press, Ann Arbor, MI.

IJIRI, Y. (1965) *Management Goals and Accounting for Control.* North-Holland, Amsterdam.

JENSEN, M.T. (2003) Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation* **7** (5), 275-288.

KATO, K. and SAKAWA, M. (1998) An interactive fuzzy satisficing method for large scale multi-objective 0-1 programming problems with fuzzy parameters through genetic algorithms. *European Journal of Operational Research* **107**, 590-598.

MICHALEWICZ, Z. (1996) *Genetic Algorithms + Data Structure = Evolution Programs*, third, revised and extended edition. Springer-Verlag, New York.

MICHALEWICZ, Z. and NAZHIYATH, G. (1995) Genocop III: A co-evolutionary algorithms for Numerical Optimization Problems with Nonlinear Constraints. *IEEE Proc. Second Int. Conf. Evolutionary Comput., Perth, Australia*, 647-651.

MICHALEWICZ, Z. and SCHOENAUER, M. (1996) Evolutionary Algorithm for Constrained Parameter Optimization problems. *Evolutionary Comput.* **4** (1), 1-32.

MOHAMED, R.H. (1997) The relationship between goal programming and fuzzy programming. *Fuzzy Sets and Systems* **89**, 215-222.

NARASIMHAN, R. (1980) Goal programming in a fuzzy environment. *Decisions and Sciences* **11**, 325-336.

Pongcharoen, P., Hicks, C. and Braiden, P.M. (2004) The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure. *European Journal of Operational Research* **152** (1), 215-225.

Rao, S.S., Sundararaju, K., Prakash, B.G. and Balakrishna, C. (1992) Fuzzy goal programming approach for structural optimization. *AIAA Journal* **30** (5), 1425-1432.

Rao, J.R., Tiwari, R.N. and Mohanty, B.K. (1988) A preference structure on aspiration levels in a goal programming problem-a fuzzy approach. *Fuzzy Sets and Systems* **25**, 175-182.

Rasmy, M.H., Sang, M.L., Abd B-Wahed, W.F., Ragab, A.M., El-Sherbiny, M.M. (2002) An expert system for multi-objective decision making: application of fuzzy linguistic preferences and goal programming. *Fuzzy Sets and Systems* **127**, 209-220.

Rubin, P.A. and Narasimhan, R. (1984) Fuzzy goal programming with nested priorities. *Fuzzy Sets and Systems* **14**, 115-129.

Saaty, T.L. (1978) Exploring the interface between hierarchies multi-objectives and fuzzy sets. *Fuzzy Sets and Systems* **1**, 57-68.

Sakawa, M., Kato, K., Nishizaki, I. (2003) An interactive fuzzy satisficing method for multi-objective stochastic linear programming problems through an expectation model. *European Journal of Operational Research* **145**, 665-672

Sakawa, M. and Yano, H. (1989) An interactive fuzzy satisficing method for multi-objective nonlinear programming with fuzzy members. *Fuzzy Sets and Systems* **30** (3), 221-238.

Sakawa, M. and Yauchi, K. (2001) An interactive fuzzy satisficing method for multi-objective non-convex programming problems with fuzzy members through co-evolutionary genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics-Part A* **31** (3), 459-467.

Tanaka, H., Okuda, T. and Asai, K. (1974) On fuzzy mathematical programming. *Journal of Cybernetics* **3**, 37-46.

Tiwari, R.N., Dharmar, S. and Rao, J.R. (1986) Priority structure in fuzzy goal programming. *Fuzzy Sets and Systems* **19**, 251-259.

Tiwari, R.N., Mohanty, B.K. and Rao, J.R. (1987) Fuzzy goal programming - an additive model. *Fuzzy Sets and Systems* **24**, 27-34.

Wu, A.S., Yu, H., Jin, S., Lin, K-C. and Schiavone, G. (2004) An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Transactions on Parallel and Distributed Systems* **15** (9), 824-834.

Yang, T., Ignizio, J.P. and Kim, H.J. (1991) Fuzzy programming with nonlinear membership functions: piecewise linear approximation. *Fuzzy Sets and Systems* **11**, 39-53.

Zadeh, L.A. (1965) Fuzzy sets. *Information and Control* **8**, 338-353

Zimmermann, H.J. (1978) Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems* **1**, 45-55.