Fuzzy Optimization and Decision Making, 1, 43–63, 2002

Toward Fuzzy Optimization without Mathematical Ambiguity

BAODING LIU
Uncertain Systems Laboratory, Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
http://orsc.edu.cn/~liu

Abstract. Fuzzy programming has been discussed widely in literature and applied in such various disciplines as operations research, economic management, business administration, and engineering. The main purpose of this paper is to present a brief review on fuzzy programming models, and classify them into three broad classes: expected value model, chance-constrained programming and dependent-chance programming. In order to solve general fuzzy programming models, a hybrid intelligent algorithm is also documented. Finally, some related topics are discussed.

Keywords: fuzzy programming, fuzzy simulation, genetic algorithm, neural network

1. Introduction

Decisions are usually made in fuzzy environments. Before discussing fuzzy optimization theory, let us recall stochastic programming. With the requirement of considering randomness, appropriate formulations of stochastic programming have been developed to suit the different purposes of management. The first type of stochastic programming is the so-called *expected value model* (EVM) which optimizes the expected objective functions subject to some expected constraints. The second, *chance-constrained programming* (CCP), was pioneered by Charnes and Cooper (1959) as a means of handling uncertainty by specifying a confidence level at which it is desired that the stochastic constraint holds. Sometimes a complex stochastic decision system undertakes multiple tasks, called events, and the decision-maker wishes to maximize the chance functions of satisfying these events in an uncertain environment. In order to model this type of problems, Liu (1997) initiated a theoretical framework of the third type of stochastic programming called *dependent-chance programming* (DCP).

Fuzzy programming offers a powerful means of handling optimization problems with fuzzy factors. Fuzzy programming has been used in different ways in the past, for example, Ostasiewicz (1982), Buckley (1988, 1990a, 1990b), Inuiguchi et al. (1993), Inuiguchi and Ramík (2000), Ramík and Rommelfanger (1996), Tanaka et al. (2000). A detailed survey on fuzzy optimization was made in 1989 by Luhandjula (1989). The readers may also consult the books such as Zimmermann (1985b) and Liu (1999b). Especially, Liu and Liu (2001a) presented a concept of expected value operator of fuzzy variable and provided a spectrum of fuzzy EVM. Following the idea of stochastic CCP, in a fuzzy decision system we assume that the fuzzy constraints will hold with a given confidence level, then we may construct fuzzy CCP models. Several papers (see Luhandjula (1983, 1986), Yazenin (1996), Zimmermann (1983, 1985a)) had considered fuzzy linear programming or fuzzy multiobjective linear programming problems, and proposed a series of ideas of translating the original chance constraints into crisp equivalents. However, with the development of more effective computer and computational intelligence, many new complex optimization problems can be processed by digital computers. Thus Liu and Iwamura (1998a, 1998b) and Liu (1998) constructed a general theoretical framework of CCP without a linearity assumption. Following the idea of stochastic DCP, Liu (2000) provided the DCP theory in fuzzy environments. Traditionally, mathematical programming models produce crisp decision vectors such that some objectives achieve the optimal values. However, for practical purposes, we should provide a fuzzy decision rather than a crisp one. Bouchon-Meunier et al. (1996) surveyed various approaches to maximizing a numerical function over a fuzzy set. Buckley and Hayashi (1994) presented a fuzzy genetic algorithm for maximizing a real-valued function by selecting an optimal fuzzy set. More generally, Liu and Iwamura (2001) provided a spectrum of CCP with fuzzy decisions, and Liu (1999a) constructed the framework of DCP with fuzzy decisions. A series of hybrid intelligent algorithms has also been designed for solving these fuzzy programming models.

Assume that $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is a fuzzy vector, $f(\boldsymbol{x},\boldsymbol{\xi})$ is a return function, and $g_j(\boldsymbol{x},\boldsymbol{\xi})$ are constraint functions, $j=1, 2,\cdots, p$. Let us examine the following "fuzzy programming",

$$\begin{cases} \max\ f(\boldsymbol{x},\boldsymbol{\xi}) \\ \text{subject to :} \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \le \boldsymbol{0}, \quad j = 1,2,\cdots,p. \end{cases} \tag{1}$$

Similar to stochastic programming, the model (1) is not well-defined mathematically because (i) we cannot maximize the fuzzy return function $f(\boldsymbol{x},\boldsymbol{\xi})$ (just like that we cannot maximize a random return function), and (ii) the constraints $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j=1, 2, \cdots, p$ do not produce a crisp feasible set.

Unfortunately, the form of fuzzy programming like (1) appears frequently in the literature. Fuzzy programming is a class of mathematical models. Everyone people should have the same understanding of the same mathematical model. In other words, a mathematical model must have an unambiguous explanation. The form (1) is considered to be not well defined unless there is a precise interpretation.

The main purpose of this paper is to introduce some fuzzy programming models without mathematical ambiguity, and classify them into three catagories: EVM, CCP and DCP. The paper is organized as follows. Sections 2, 3 and 4 introduce basic concepts in the area of fuzzy optimization, such as possibility space, fuzzy variable, possibility, necessity, credibility, and expected value operator. In Section 5 we present the general EVM. In Section 6 CCP models are documented. Section 7 deals with DCP theory. Section 8 extends fuzzy programming with possibility measure to that with necessity and credibility measures. Section 9 explains the technique of fuzzy simulations. Section 10 provides a hybrid intelligent algorithm for solving general fuzzy programming models. Some related topics are documented in Section 11.

2. Possibility Space and Fuzzy Variables

In order to provide an axiomatic Btheory to describe fuzziness, Nahmias (1978) suggested a theoretical framework. Let Θ be a nonempty set, and $B(\Theta)$ be the power set of Θ. For each $A \in B(\Theta)$, there is a nonnegative number $\mathrm{Pos}\{A\}$, called its possibility, such that

(i) $\mathrm{Pos}\{0\} = 0$, $\mathrm{Pos}\{\Theta\} = 1$; and
(ii) $\mathrm{Pos}\{\cup_k A_k\} = \sup_k \mathrm{Pos}\{A_k\}$ for any arbitrary collection $\{A_k\}$ in $B(\Theta)$.

The triplet $(\Theta, B(\Theta), \mathrm{Pos})$ is called a *possibility space*, and the function Pos is referred to as a possibility measure.

Fuzzy variable has been defined in many ways. Here a fuzzy variable $\boldsymbol{\xi}$ is defined as a function from a possibility space $(\Theta, B(\Theta), \mathrm{Pos})$ to the real line $\Re$. Then its membership function is derived from the possibility measure Pos by $\mu(x) = \mathbf{Pos}\{\theta \in \Theta | \boldsymbol{\xi}(\theta) = x\}$.

For any fuzzy variable $\boldsymbol{\xi}$ with membership function μ, since $\boldsymbol{\xi}$ is a function from $\boldsymbol{\Theta}$ to the real line $\Re$, we have $\boldsymbol{\Theta} = \cup_{x \in \Re}\{\theta \in \boldsymbol{\Theta} | \boldsymbol{\xi}(\theta) = x\}$. It follows from (ii) that

$$\sup_x \mu(x) = \sup_x \mathrm{Pos}\{\theta \in \boldsymbol{\Theta} \mid \boldsymbol{\xi}(\theta) = x\} = \mathrm{Pos}\{\boldsymbol{\Theta}\} = 1.$$

That is, any fuzzy variables defined as above are normalized.

Suppose that $(\boldsymbol{\Theta}_i, B(\boldsymbol{\Theta}_i), \mathrm{Pos}_i)$ are possibility spaces, $i = 1, 2, \cdots, m$.

Write

$$\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 \times \boldsymbol{\Theta}_2 \times \cdots \times \boldsymbol{\Theta}_m. \tag{2}$$

For any $A \in B(\boldsymbol{\Theta})$, we define a possibility measure as follows,

$$\mathrm{Pos}\{A\} = \sup_{(\theta_1, \theta_2, \cdots, \theta_m) \in A} \mathrm{Pos}_1\{\theta_1\} \wedge \mathrm{Pos}_2\{\theta_2\} \wedge \cdots \wedge \mathrm{Pos}_m\{\theta_m\}. \tag{3}$$

Is the triplet $(\boldsymbol{\Theta}, B(\boldsymbol{\Theta}), \mathrm{Pos})$ a possibility space? It is obvious that $\mathrm{Pos}\{0\} = 0$ and $\mathrm{Pos}\{\boldsymbol{\Theta}\} = 1$. In addition, for any arbitrary collection $\{A_k\}$ in $B(\boldsymbol{\Theta})$, we have

$$\begin{aligned}\mathrm{Pos}\{\cup_k A_k\} &= \sup_{(\theta_1, \theta_2, \cdots, \theta_m) \in \cup_k A_k} \min_{1 \le i \le m} \mathrm{Pos}_i\{\theta_i\} = \sup_k \left\{ \sup_{(\theta_1, \theta_2, \cdots, \theta_m) \in A_k} \min_{1 \le i \le m} \mathrm{Pos}_i\{\theta_i\} \right\} \\ &= \sup_k \{\mathrm{Pos}\{A_k\}\}.\end{aligned}$$

Hence $(\boldsymbol{\Theta}, B(\boldsymbol{\Theta}), \mathrm{Pos})$ is proved to be a possibility space, and is called a *product possibility space*.

3. Possibility, Necessity and Credibility

Possibility theory was proposed by Zadeh (1978), and developed by many researchers such as Dubois and Prade (1988). Let $\tilde{a}$ and $\tilde{b}$ be fuzzy variables defined on the possibility spaces (Θ_1, $\mathcal{B}(\Theta_1)$, Pos_1) and (Θ_2, $\mathcal{B}(\Theta_2)$, Pos_2), respectively. Then $\tilde{a} \leq \tilde{b}$ is a fuzzy event defined on the product possibility space (Θ, $\mathcal{B}(\Theta)$, Pos), whose possibility is

$$\begin{aligned}\text{Pos}\left\{\tilde{a} \leq \tilde{b}\right\} &= \sup_{\theta_1 \in \Theta_1, \theta_2 \in \Theta_2} \text{Pos}\left\{\{(\theta_1, \theta_2)\} \mid \tilde{a}(\theta_1) \leq \tilde{b}(\theta_2)\right\} \\ &= \sup_{\theta_1 \in \Theta_1, \theta_2 \in \Theta_2} \left\{\text{Pos}_1\{\theta_1\} \wedge \text{Pos}_2\{\theta_2\} \mid \tilde{a}(\theta_1) \leq \tilde{b}(\theta_2)\right\} \\ &= \sup_{x, y \in \Re} \left\{\mu_{\tilde{a}}(x) \wedge \mu_{\tilde{b}}(y) \mid x \leq y\right\}.\end{aligned}$$

This means that the possibility of $\tilde{a} \leq \tilde{b}$ is the largest possibility that there exists at least one pair of values x, $y \in \Re$ such that $x \leq y$, and the values of $\tilde{a}$ and $\tilde{b}$ are x and y, respectively. More generally, let $\tilde{a}_1$, $\tilde{a}_2, \cdots, \tilde{a}_n$ be fuzzy variables, and f_j: $\Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the possibility of the fuzzy event characterized by $f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0$, $j = 1, 2, \cdots, m$ is defined by

$$\begin{aligned}&\text{Pos}\left\{f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0,\ j = 1, 2, \cdots, m\right\} \\ &\quad = \sup_{x_1, x_2, \cdots, x_n \in \Re} \left\{ \min_{1 \leq i \leq n} \mu_{\tilde{a}_i}(x_i) \,\middle|\, \begin{array}{c} f_j(x_1, x_2, \cdots, x_n) \leq 0 \\ j = 1, 2, \cdots, m \end{array} \right\}. \end{aligned} \tag{4}$$

The *necessity* of a fuzzy event is defined as the impossibility of the opposite event. Thus a necessity measure is the dual of possibility measure. Since the opposite event of $f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0$, $j = 1, 2, \cdots, m$ is $f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) > 0$, $\exists\, j \in \{1, 2, \cdots, m\}$, the necessity is defined by

$$\begin{aligned}&\text{Nec}\left\{f_i(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0,\ j = 1, 2, \cdots, m\right\} \\ &\quad = 1 - \sup_{x_1, x_2, \cdots, x_n \in \Re} \left\{ \min_{1 \leq i \leq n} \mu_{\tilde{a}_i}(x_i) \,\middle|\, \begin{array}{c} f_j(x_1, x_2, \cdots, x_n) > 0 \\ \exists\, j \in \{1, 2, \cdots, m\} \end{array} \right\}. \end{aligned} \tag{5}$$

The *credibility* of a fuzzy event is defined as the average of its possibility and necessity, i.e., $\text{Cr}\{\cdot\} = \frac{1}{2}(\text{Pos}\{\cdot\} + \text{Nec}\{\cdot\})$.

It is easy to prove the following relationship among possibility, necessity and credibility,

$$\begin{aligned}&\text{Pos}\left\{f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0, j = 1, 2, \cdots, m\right\} \\ &\quad \geq \text{Cr}\left\{f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0, j = 1, 2, \cdots, m\right\} \\ &\quad \geq \text{Nec}\left\{f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \leq 0, j = 1, 2, \cdots, m\right\}. \end{aligned} \tag{6}$$

4. Expected Value Operator

Liu and Liu (2001a) presented an *expected value operator* of fuzzy variable. Let ξ be a fuzzy variable on the possibility space ($\mathbf{\Theta}$, $\mathcal{B}(\mathbf{\Theta})$, Pos). The expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathrm{Cr}\{\xi \geq r\}\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Cr}\{\xi \leq r\}\mathrm{d}r. \tag{7}$$

This definition is not only applicable to continuous case but also discrete case. When ξ is a discrete fuzzy variable whose membership function is given by

$$\mu(x) = \begin{cases} \mu_1, & \text{if } x = a_1 \\ \mu_2, & \text{if } x = a_2 \\ \cdots \\ \mu_m, & \text{if } x = a_m. \end{cases}$$

Without loss of generality, we also assume that $a_1 \leq a_2 \leq \cdots \leq a_m$. It follows from (7) that the expected value of ξ is

$$E[\xi] = \sum_{i=1}^{m} w_i a_i \tag{8}$$

where the weights w_i, $i = 2,3,\cdots, m$ are given by

$$\begin{aligned} w_1 &= \tfrac{1}{2}(\mu_1 + \max_{1\leq j\leq m} \mu_j - \max_{1<m}\mu_j), \\ w_i &= \tfrac{1}{2}\left(\max_{1\leq j\leq i} \mu_j - \max_{1\leq j<i} \mu_j + \max_{i\leq j\leq m} \mu_j - \max_{i<j\leq m} \mu_j\right), \quad 2 \leq i \leq m-1, \\ w_m &= \tfrac{1}{2}\left(\max_{1\leq j\leq m} \mu_j - \max_{1\leq j<m} \mu_j + \mu_m\right). \end{aligned}$$

The definition of expected value operator is reasonable because (i) since the dual of Cr is itself, the expected value $E[\xi]$ is exactly a type of Choquet integral which is a generalization of mathematical expectation; (ii) if the fuzzy variable ξ is replaced with a random variable (whose density function is ϕ) and Cr is replaced with the probability measure Pr (whose dual is itself!), then we have

$$\int_0^{+\infty} \Pr\{\xi \geq r\}\mathrm{d}r - \int_{-\infty}^{0} \Pr\{\xi \leq r\}\mathrm{d}r = \int_{-\infty}^{+\infty} x\phi(x)\mathrm{d}x$$

which is exactly the expected value of random variable ξ. This means that the form of expected value of fuzzy variable is identical to that of random variable; and (iii) if ξ and η are fuzzy variables with finite expected values, then $E[a\xi + b\eta] = aE[\xi] + bE[\eta]$ for any real numbers a and b (Liu and Liu (2001b)).

5. Expected Value Models

In order to make unambiguous fuzzy programming models, Liu and Liu (2001a) presented a series of fuzzy EVM, in which the underlying philosophy is based on selecting the decision with maximum expected return. The general form of EVM is as follows,

$$\begin{cases} \max\ E[f(\boldsymbol{x},\boldsymbol{\xi})] \\ \text{subject to :} \\ E[g_j(\boldsymbol{x},\boldsymbol{\xi})] \le 0, \quad j = 1,2,\cdots,p \end{cases} \tag{9}$$

where x is a decision vector, $\boldsymbol{\xi}$ is a fuzzy vector, $f(\boldsymbol{x},\boldsymbol{\xi})$ is the return function, $g_j(\boldsymbol{x},\boldsymbol{\xi})$ are fuzzy constraint functions for $j = 1, 2, \cdots, p$.

As an extension of single-objective programming models, fuzzy expected value multi-objective programming (EVMOP) has the following general form:

$$\begin{cases} \max\ E[f_1(\boldsymbol{x},\boldsymbol{\xi})], E[f_2(\boldsymbol{x},\boldsymbol{\xi})],\cdots,E[f_m(\boldsymbol{x},\boldsymbol{\xi})]] \\ \text{subject to :} \\ E[g_j(\boldsymbol{x},\boldsymbol{\xi})] \le 0, \quad j = 1,2,\cdots,p \end{cases} \tag{10}$$

where $f_i(\boldsymbol{x},\boldsymbol{\xi})$ are return functions for $i = 1, 2, \cdots, m$.

When some management targets are given, the objective function may minimize the deviations, positive, negative, or both, with a certain priority structure. Thus we can also formulate a fuzzy decision system as an expected value goal programming (EVGP) according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij}d_i^+ + v_{ij}d_i^-) \\ \text{subject to :} \\ E[f_i(\boldsymbol{x},\boldsymbol{\xi})] + d_i^- - d_i^+ = b_i, \quad i = 1,2,\cdots,m \\ E[g_j(\boldsymbol{x},\boldsymbol{\xi})] \le 0, \qquad j = 1,2,\cdots,p \\ d_i^+, d_i^- \ge 0, \qquad i = 1,2,\cdots,m \end{cases} \tag{11}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is positive deviation from the target of goal i, d_i^- is negative deviation from the target of goal i, f_i is a function in goal constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of real constraints.

6. Chance-Constrained Programming

In practice, we are not always concerned with maximizing expected profit or minimizing expected cost. Thus EVM is not always valid. Since the fuzzy constraints $g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $j = 1, 2, \cdots, p$ do not define a crisp feasible set, a natural idea is to provide a possibility (called confidence level) α at which it is desired that the fuzzy constraints hold. Thus we have a chance constraint as follows,

$$\text{Pos}\left\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, p\right\} \geq \alpha. \tag{12}$$

In addition, no matter what types of fuzzy vector $\boldsymbol{\xi}$ and functional form f are, for each given decision $\boldsymbol{x}$, $f(\boldsymbol{x},\boldsymbol{\xi})$ is a fuzzy variable. Analogous to the stochastic programming, we are interested in two critical values — *optimistic value* and *pessimistic value* — with a given confidence level.

Let $\beta \in (0, 1]$. The number $f_{sup} = \sup\{r \mid \text{Pos}\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq r\} \geq \beta\}$ is called the β-optimistic value to the return function $f(\boldsymbol{x},\boldsymbol{\xi})$. This means that the return function $f(\boldsymbol{x},\boldsymbol{\xi})$ will reach the β-optimistic value f_{sup} with possibility β.

On the other hand, Murphy's law states that "if anything can go wrong, it will". If you believe it, you may select the action with the best of these worst returns. In order to do so, we may measure the return function by its β-pessimistic value, $f_{inf} = \inf\{r \mid \text{Pos}\{f(\boldsymbol{x},\boldsymbol{\xi}) \leq r\} \geq \beta\}$. This means that the return $f(\boldsymbol{x},\boldsymbol{\xi})$ will be below the β-pessimistic value f_{inf} with possibility β.

6.1. Maximax Chance-Constrained Programming

Following the idea of stochastic CCP, Liu and Iwamura (1998a, 1998b) suggested a spectrum of fuzzy CCP models. When we want to maximize the β-optimistic value, we have the following fuzzy CCP model,

$$\begin{cases} \max \bar{f} \\ \text{subject to :} \\ \text{Pos}\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}\} \geq \beta \\ \text{Pos}\left\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, p\right\} \geq \alpha \end{cases} \tag{13}$$

where α and β are the predetermined confidence levels. The CCP model (13) is called a maximax model because it is equivalent to the maximax form

$$\begin{cases} \max\limits_{x} \max\limits_{\bar{f}} \bar{f} \\ \text{subject to :} \\ \text{Pos}\left\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}\right\} \geq \beta \\ \text{Pos}\left\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1,2,\cdots,p\right\} \geq \alpha \end{cases}$$

where $\max \bar{f}$ is the β-optimistic value to the return function $f(\boldsymbol{x},\boldsymbol{\xi})$.

If there are multiple objectives, then we have the following chance-constrained multiobjective programming (CCMOP) model,

$$\begin{cases} \max[\bar{f}_1,\bar{f}_2,\cdots,\bar{f}_m] \\ \text{subject to :} \\ \text{Pos}\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}_i\} \geq \beta_i, \quad i = 1,2,\cdots,m \\ \text{Pos}\left\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\right\} \geq \alpha_j, \quad j = 1,2,\cdots,p \end{cases} \tag{14}$$

where β_i are confidence levels. The fuzzy CCMOP (14) is equivalent to the maximax form

$$\begin{cases} \max\limits_{x} \left[\max\limits_{\bar{f}_1}\bar{f}_1, \max\limits_{\bar{f}_2}\bar{f}_2,\cdots, \max\limits_{\bar{f}_m}\bar{f}_m\right] \\ \text{subject to :} \\ \text{Pos}\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}_i\} \geq \beta_i, \quad i = 1,2,\cdots,m \\ \text{Pos}\left\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\right\} \geq \alpha_j, \quad j = 1,2,\cdots,p \end{cases}$$

where $\max \bar{f}_i$ are the β_i-optimistic values to the return functions $f_i(\boldsymbol{x},\boldsymbol{\xi})$, $i = 1,2,\cdots,m$, respectively.

We can also formulate the fuzzy decision system as a minimin chance-constrained goal programming (CCGP) according to the priority structure and target levels set by the decision-maker:

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{subject to :} \\ \text{Pos}\,\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+, \quad i = 1,2,\cdots,m \\ \text{Pos}\,\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \le d_i^-\} \ge \beta_i^-, \quad i = 1,2,\cdots,m \\ \text{Pos}\,\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0\} \ge \alpha_j, \qquad j = 1,2,\cdots,p \\ d_i^+, d_i^- \ge 0, \qquad i = 1,2,\cdots,m \end{cases} \tag{15}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the β_i^+ -optimistic positive deviation from the target of goal i, defined as

$$\min \left\{ d \vee 0 \mid \text{Pos}\,\{ f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \le d\} \ge \beta_i^+ \right\}, \tag{16}$$

d_i^- is the β_i^--optimistic negative deviation from the target of goal i, defined as

$$\min \left\{ d \vee 0 \mid \text{Pos}\,\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \le d\} \ge \beta_i^- \right\}, \tag{17}$$

f_i is a function in goal constraints, g_j is a function in real constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, p is the number of real constraints.

If the fuzzy vector $\boldsymbol{\xi}$ degenerates to the crisp case, then two posibilities Pos$\{f_i(x,\boldsymbol{\xi}) - b_i \le d_i^+\}$ and Pos $\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \le d_i^-\}$ should be always 1 provided that β_i^+, $\beta_i^- > 0$, and the goal constraints

$$\text{Pos}\,\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \le d_i^+\} \ge \beta_i^+,\ d_i^+ \ge 0,$$

$$\text{Pos}\,\left\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \le d_i^-\right\} \ge \beta_i^-, d_i^- \ge 0$$

become $d_i^- = [f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i] \vee 0$, $d_i^- = [b_i - f_i(\boldsymbol{x},\boldsymbol{\xi})] \vee 0$. This coincides with the crisp goal programming.

6.2. Minimax Chance-Constrained Programming

In fact, maximax CCP models are essentially a type of optimistic models which maximize the maximum possible return. This section introduces a spectrum of minimax CCP models constructed by Liu (1998), which will select the alternative that provides the best of the worst possible return.

If we want to maximize the β-pessimistic value, then we have the following minimax CCP model,

$$\begin{cases} \max\limits_{x} \min\limits_{\bar{f}} \bar{f} \\ \text{subject to :} \\ \text{Pos}\,\{f(\boldsymbol{x},\boldsymbol{\xi}) \leq \bar{f}\} \geq \beta \\ \text{Pos}\,\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1,2,\cdots,p\} \geq \alpha \end{cases} \tag{18}$$

where $\min \bar{f}$ is the β-pessimistic value to the return function $f(\boldsymbol{x},\boldsymbol{\xi})$.

If there are multiple objectives, we may employ the following minimax CCMOP model,

$$\begin{cases} \max\limits_{x} \left[\min\limits_{\bar{f}_1} \bar{f}_1, \min\limits_{\bar{f}_2} \bar{f}_2, \cdots, \min\limits_{\bar{f}_m} \bar{f}_m \right] \\ \text{subject to :} \\ \text{Pos}\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq \bar{f}_i\} \geq \beta_i, \quad i = 1,2,\cdots,m \\ \text{Pos}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq \boldsymbol{0}\} \geq \alpha_j, \quad j = 1,2,\cdots,p \end{cases} \tag{19}$$

where α_j and β_i are confidence levels, and $\min \bar{f}_i$ are the β_i-pessimistic values to the return functions $f_i(\boldsymbol{x},\boldsymbol{\xi})$, $i = 1, 2, \cdots, m$, respectively.

According to the priority structure and target levels set by the decision-maker, the minimax CCGP model is written as follows,

$$\begin{cases} \min\limits_{x} \sum\limits_{j=1}^{l} P_j \sum\limits_{i=1}^{m} \left[u_{ij} \left(\max\limits_{d_i^+} d_i^+ \vee 0 \right) + v_{ij} \left(\max\limits_{d_i^-} d_i^- \vee 0 \right) \right] \\ \text{subject to :} \\ \text{Pos}\,\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \geq d_i^+\} \geq \beta_i^+, \qquad i = 1,2,\cdots,m \\ \text{Pos}\,\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) \geq d_i^-\} \geq \beta_i^-, \qquad i = 1,2,\cdots,m \\ \text{Pos}\,\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq \alpha_j, \qquad j = 1,2,\cdots,p \end{cases} \tag{20}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, $d_i^+ \vee 0$ is the β_i^+-pessimistic positive deviation from the target of goal i, defined as

$$\max\left\{d \vee 0 \mid \text{Pos}\left\{f_i(\boldsymbol{x},\boldsymbol{\xi}) - b_i \geq d\right\} \geq \beta_i^+\right\}, \tag{21}$$

$d_i^- \vee 0$ is the β_i^- -pessimistic negative deviation from the target of goal i with possibility β_i^-, defined as

$$\max\left\{d \vee 0 \mid \text{Pos}\left\{b_i - f_i(\boldsymbol{x},\boldsymbol{\xi}) - \geq d\right\} \geq \beta_i^-\right\}, \tag{22}$$

b_i is the target value according to goal i, l is the number of priorities, and m is the number of goal constraints.

In a minimax CCGP model, there is no nonnegativity condition $d_i^+, d_i^- \geq 0$ because it clashes with the goal constraints. In order to overcome this problem, we have to replace d_i^+ and d_i^- with $d_i^+ \vee 0$ and $d_i^- \vee 0$, respectively, in the objective function.

7. Dependent-Chance Programming

Following the idea of stochastic DCP, Liu (1999a) provided a series of DCP models in fuzzy environments, in which the underlying philosophy is based on selecting the decision with maximum possibility to meet the event.

Uncertain environment, event and chance function are key elements in the area of fuzzy DCP. DCP theory breaks the concept of feasible set and replaces it with an uncertain environment. By *uncertain environment* (in this case the fuzzy environment) we mean the fuzzy constraints represented by

$$g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, \quad j = 1, 2, \cdots, p \tag{23}$$

where x is a decision vector, and $\boldsymbol{\xi}$ is a fuzzy vector.

A complex decision system usually undertakes multiple tasks. By *events* we mean these tasks. An event $\mathcal{E}$ is generally characterized by

$$h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, \quad k = 1, 2, \cdots, q. \tag{24}$$

Note that any event is uncertain if it is in an uncertain environment.

The *chance function* of an event $\mathcal{E}$ characterized by (24) is defined as the possibility measure of the event $\mathcal{E}$, i.e.,

$$f(x) = \text{Pos}\left\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1, 2, \cdots, q\right\} \tag{25}$$

subject to the uncertain environment (23).

Sometimes, the fuzzy parameter $\boldsymbol{\xi}$ may vanish in the event $h_k(\boldsymbol{x},\boldsymbol{\xi}) \le 0$, $k = 1, 2,\cdots, q$. However, it is still an uncertain environment. An fundamental law in DCP theory is that *any event is uncertain if it is in an uncertain environment*.

How do we compute the chance function of an event $\boldsymbol{\varepsilon}$ in an uncertain environment? In order to answer this problem, let us recall some basic concepts defined in the area of DCP.

The *support* $\mathcal{E}^*$ is defined as the set consisting of all nondegenerate decision variables of functions $h_k(\boldsymbol{x},\boldsymbol{\xi})$, $k = 1, 2,\cdots, q$.

The jth constraint $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0$ is called an *active constraint* of the event $\mathcal{E}$ if the set of nondegenerate decision variables of $g_j(\boldsymbol{x},\boldsymbol{\xi})$ and the support $\mathcal{E}^*$ have nonempty intersection; otherwise it is inactive.

The *dependent support* $\mathcal{E}^{**}$ is defined as the set consisting of all nondegenerate decision variables of $h_k(\boldsymbol{x},\boldsymbol{\xi})$, $k = 1, 2,\cdots, q$ and $g_j(\boldsymbol{x},\boldsymbol{\xi})$ in the active constraints to the event $\mathcal{E}$.

The jth constraint $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0$ is called a *dependent constraint* of the event $\boldsymbol{\varepsilon}$if the set of nondegenerate decision variables of $g_j(\boldsymbol{x},\boldsymbol{\xi})$ and the dependent support $\mathcal{E}^{**}$ have nonempty intersection; otherwise it is independent.

For each decision $\boldsymbol{x}$ and realization $\boldsymbol{\xi}$, an event $\boldsymbol{\varepsilon}$ is said to be consistent in the uncertain environment if the following two conditions hold: (i) $h_k(\boldsymbol{x},\boldsymbol{\xi}) \le 0$, $k = 1,2,\cdots, q$, and (ii) $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j \in J$, where J is the index set of all dependent constraints. An event can be met by a decision provided that the decision meets both the event itself and the dependent constraints. We conclude it with the following principle of uncertainty.

Principle of Uncertainty: The chance of a fuzzy event is the possibility (or necessity, credibility) that the event is consistent in the uncertain environment.

Assume that there are m events $\boldsymbol{\varepsilon}_i$ characterized by $h_{ik}(\boldsymbol{x},\boldsymbol{\xi}) \le 0$, $k = 1, 2,\cdots, q_i$ for $i = 1, 2,\cdots, m$ in the uncertain environment $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0$, $j = 1,2,\cdots, p$. The principle of uncertainty implies that the chance function of the ith event $\boldsymbol{\varepsilon}_i$ in the uncertain environment is

$$f_i(\boldsymbol{x}) = \mathrm{Pos}\left\{\begin{array}{c} h_{ik}(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k = 1, 2, \cdots, q_i \\ \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j \in J_i \end{array}\right\} \tag{26}$$

where J_i are defined by

$$J_i = j \in \{1, 2, \cdots, p\} \mid g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0 \textit{ is a dependent constraint of } \boldsymbol{\xi}\}$$

for $\quad i = 1, 2, \ldots, m$.

A typical formulation of DCP in a fuzzy environment is given as follows:

$$\begin{cases} \max\ \mathrm{Pos}\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k = 1, 2,\cdots, q\} \\ \\ \text{subject to :} \\ \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{27}$$

where $\boldsymbol{x}$ is an n-dimensional decision vector, $\boldsymbol{\xi}$ is a fuzzy vector, the event $\boldsymbol{\varepsilon}$ is characterized by $h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $k = 1, 2, \cdots, q$, and the uncertain environment is $g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $j = 1, 2, \cdots, p$.

The fuzzy DCP (27) reads as "maximizing the possibility of the fuzzy event $h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $k = 1, 2, \cdots, q$ subject to the uncertain environment $g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $j = 1, 2, \cdots, p$."

Since a complex decision system usually undertakes multiple tasks, there undoubtedly exist multiple potential objectives (some of them are chance functions) in the decision process. A typical formulation of fuzzy dependent-chance multiobjective programming (DCMOP) is given as follows,

$$\begin{cases} \max \begin{bmatrix} \text{Pos}\{h_{1k}(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1, 2, \cdots, q_1\} \\ \text{Pos}\{h_{2k}(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1, 2, \cdots, q_2\} \\ \cdots \\ \text{Pos}\{h_{mk}(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k = 1, 2, \cdots, q_m\} \end{bmatrix} \\ \text{subject to:} \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{28}$$

where $h_{ik}(\boldsymbol{x},\boldsymbol{\xi}) \leq 0$, $k = 1, 2, \cdots, q_i$ represent events $\mathcal{E}_i$ for $i = 1, 2, \cdots, m$, respectively.

Dependent-chance goal programming (DCGP) in fuzzy environment may be considered as an extension of goal programming in a complex uncertain decision system. We can formulate a fuzzy decision system as a DCGP according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \text{subject to:} \\ \text{Pos}\left\{ \begin{array}{l} h_{ik}(\boldsymbol{x},\boldsymbol{\xi}) \leq 0 \\ k = 1, 2, \cdots, q_i \end{array} \right\} + d_i^- - d_i^+ = b_i, \quad i = 1, 2, \cdots, m \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, \qquad j = 1, 2, \cdots, p \\ d_i^+, d_i^- \geq 0, \qquad i = 1, 2, \cdots, m \end{cases} \tag{29}$$

where P_j is the preemptive priority factor which expresses the relative importance of various goals, $P_j \gg P_{j+1}$, for all j, u_{ij} is the weighting factor corresponding to positive deviation for goal i with priority j assigned, v_{ij} is the weighting factor corresponding to negative deviation for goal i with priority j assigned, d_i^+ is the positive deviation from the target of goal i, d_i^- is the negative deviation from the target of goal i, g_j is a function in system constraints, b_i is the target value according to goal i, l is the number of priorities, m is the number of goal constraints, and p is the number of system constraints.

Remark. It is clear that the modality constrained programming is a special case of DCP model when the uncertain environment reduces to crisp constraints.

8. Varieties of Fuzzy Programming Model

Let ξ and η be two fuzzy variables defined on the possibility space $(\Theta, B(\Theta), \text{Pos})$. There are many ways to rank them. For example, we say $\xi \geq \eta$ if and only if $\xi(\theta) \geq \eta(\theta)$ for all $\theta \in \Theta$. However, in practice, this way is obviously not appreciate. The following ranking methods are suggested.

(i) We say $\xi > \eta$ if and only if $E[\xi] > E[\eta]$, where E is the expected value operator of fuzzy variable. This leads to fuzzy EVM.
(ii) We say $\xi > \eta$ if and only if, for some predetermined confidence level $\alpha \in (0,1]$, we have $\xi_{\sup}(\alpha) > \eta_{\sup}(\alpha)$, where $\xi_{\sup}(\alpha)$ and $\eta_{\sup}(\alpha)$ are the α-optimistic values of ξ and η, respectively. This leads to fuzzy CCP.
(iii) We say $\xi > \eta$ if and only if, for some predetermined confidence level $\alpha \in (0,1]$, we have $\xi_{\inf}(\alpha) > \eta_{\inf}(\alpha)$, where $\xi_{\inf}(\alpha)$ and $\eta_{\inf}(\alpha)$ are the α-pessimistic values of ξ and η, respectively. This leads to fuzzy minimax CCP.
(iv) We say $\xi > \eta$ if and only if $\text{Pos}\,\{\xi \geq \bar{r}\} > \text{Pos}\,\{\eta \geq \bar{r}\}$ for some predetermined level $\bar{r}$. This leadsn to fuzzy DCP.

The possibility measure "Pos" has been employed in fuzzy programming. In fact, it may be replaced with the necessity measure "Nec" or credibility measure "Cr", thus producing varieties of fuzzy CCP and DCP models. For example,

$$\begin{cases} \max \bar{f} \\ \text{subject to :} \\ \text{Nec}\,\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}\} \geq \beta \\ \text{Nec}\,\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0,\ j = 1,2,\cdots,p\} \geq \alpha, \end{cases} \tag{30}$$

$$\begin{cases} \max \overline{f} \\ \text{subject to :} \\ \mathrm{Cr}\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \overline{f}\} \geq \beta \\ \mathrm{Cr}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0,\ j = 1,2,\cdots,p\} \geq \alpha. \end{cases} \tag{31}$$

In addition, it is clear that both minimax and maximax models are extreme cases. In fact, we can also define the CCP model in the following form,

$$\begin{cases} \max\ \lambda f_{\inf} + (1-\lambda) f_{\sup} \\ \text{subject to :} \\ \mathrm{Pos/Nec/Cr}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0,\ j = 1,2,\cdots,p\} \geq \alpha \end{cases} \tag{32}$$

where f_{inf} and f_{sup} are the β-pessimistic and β-optimistic values to the return function $f(\boldsymbol{x},\boldsymbol{\xi})$, λ is a given number between 0 and 1. Note that when $\lambda = 0$, it is a maximax model; when $\lambda = 1$, it is a minimax model.

9. Fuzzy Simulation

In order to solve general fuzzy programming models, we have to design some fuzzy simulation techniques to handle uncertain functions, such as critical value (Liu and Iwamura (1998a, 1998b)), chance function (Liu (1999a, 2000)), and expected value (Liu and Liu (2001a)).

9.1. Computing expected value

Let $\boldsymbol{\xi}$ be a fuzzy vector with membership function μ, and f be a real-valued continuous function. In order to solve fuzzy EVM, we must compute the expected value $E[f(\boldsymbol{x},\boldsymbol{\xi})]$ for each given decision $\boldsymbol{x}$. We can estimate the expected value $E[f(\boldsymbol{x},\boldsymbol{\xi})]$ by the following procedure.

Step 1: Sample N points $\boldsymbol{u}_i$ uniformly from ε-level set of $\boldsymbol{\xi}$ for $i = 1, 2,\cdots, N$, where N is a large integer representing the sample size, and ε is a sufficiently small number.
Step 2: Rearrange $\boldsymbol{u}_1, \boldsymbol{u}_2,\cdots,\boldsymbol{u}_N$ such that $a_1 \leq a_2 \leq \cdots \leq a_N$, where $a_i = f(\boldsymbol{x},\boldsymbol{u}_i)$ for $i = 1, 2,\cdots, N$.
Step 3: Calculate the expected value $E[f(\boldsymbol{x},\boldsymbol{\xi})]$ according to formula (8), where $\mu_i = \mu(\boldsymbol{u}_i)$ for $i = 1, 2,\cdots, N$.

9.2. *Checking Chance Constraints*

Although chance constraints can be represented in an explicit form for some special cases, we need numerical methods for general cases. In order to check the fuzzy chance constraints like

$$\mathrm{Pos}\{g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0,\ j=1,2,\cdots,p\} \geq \alpha, \tag{33}$$

we first generate a crisp vector $\boldsymbol{u}$ from the α-level set of fuzzy vector $\boldsymbol{\xi}$. If the α-level set of the fuzzy vector $\boldsymbol{\xi}$ is too complex to determine, we can sample a vector $\boldsymbol{u}$ from a hypercube containing the α-level set and then accept or reject it, depending on whether $\mu(\boldsymbol{u}) \geq \alpha$ or not. If $g_j(\boldsymbol{x},\boldsymbol{u}) \leq 0, j = 1, 2, \cdots, p$ then the chance constraint holds.

Step 1: Randomly generate $\boldsymbol{u}$ from the α-level set of fuzzy vector $\boldsymbol{\xi}$.
Step 2: If $g_j(\boldsymbol{x},\boldsymbol{u}) \leq 0, j = 1, 2, \cdots, p$, return "feasible".
Step 3: Repeat the first and second steps for N times.
Step 4: Return "infeasible".

9.3. *Finding Critical Values*

For a fuzzy objective constraint with a fuzzy vector $\boldsymbol{\xi}$,

$$\mathrm{Pos}\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}\} \geq \beta, \tag{34}$$

we should find the optimistic value $\bar{f}$ for a given decision vector $\boldsymbol{x}$. First we set $\bar{f} = -\infty$. Then we generate a crisp vector $\boldsymbol{u}$ from the β-level set of fuzzy vector $\boldsymbol{\xi}$. We set $\bar{f} = f(\boldsymbol{x},\boldsymbol{u})$ provided that $\bar{f} < f(\boldsymbol{x},\boldsymbol{u})$. Repeat this process N times. The value $\bar{f}$ is regarded as the objective value at the point x.

Step 1: Set $\bar{f} = -\infty$.
Step 2: Randomly generate $\boldsymbol{u}$ from the β-level set of fuzzy vector $\boldsymbol{\xi}$.
Step 3: If $\bar{f} = f(\boldsymbol{x}, \boldsymbol{u})$, then we set $\bar{f} = f(\boldsymbol{x}, \boldsymbol{u})$.
Step 4: Repeat the second and third steps for N times.
Step 5: Return $\bar{f}$.

9.4. *Computing Chance Functions*

We will show how to compute the chance functions by the technique of fuzzy simulation. It follows from the principle of uncertainty that the chance function is

$$f(\boldsymbol{x}) = \mathrm{Pos}\left\{\begin{array}{c} h_k(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, k=1,2,\cdots,q \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0,\ j \in J \end{array}\right\}.$$

For each fixed decision $\boldsymbol{x}$, we set $f(\boldsymbol{x}) = 0$ at first. In practice, we are not interested in the decision vectors with too low possibility. Thus we can predetermine a level, say α, Then we randomly generate a crisp vector $\boldsymbol{u}$ from the α-level set of fuzzy vector $\boldsymbol{\xi}$. If $h_k(\boldsymbol{x}, \boldsymbol{u}) \leq 0$, $k = 1, 2, \cdots, q$, $g_j(\boldsymbol{x}, \boldsymbol{u}) \leq 0$, $j \in J$ and $f(\boldsymbol{x}) < \mu(\boldsymbol{u})$, then we set $f(\boldsymbol{x})$ is regarded as its estimation.

Step 1: Set $f(\boldsymbol{x}) = \alpha$, where α is a lower estimation of $f(\boldsymbol{x})$.
Step 2: Randomly generate u from the α-level set a fuzzy vector $\boldsymbol{\xi}$.
Step 3: If $h_k(\boldsymbol{x}, \boldsymbol{u}) \leq 0$, $k = 1,2, \cdots, q$, $g_j(\boldsymbol{y}, \boldsymbol{u}) \leq 0$, $j \in J$ and $f(\boldsymbol{x}) < \mu(\boldsymbol{u})$, then we set $f(\boldsymbol{x}) = \mu(\boldsymbol{u})$.
Step 4: Repeat the second and third steps for N times.
Step 5: Return $f(\boldsymbol{x})$.

10. Hybrid Intelligent Algorithms

Numerous intelligent algorithms — simulations, neural networks (NN), genetic algorithms (GA), simulated annealing (SA), tabu search (TS)—have been developed to solve problems. There are also numerous algorithms designed for solving fuzzy optimization problems, for example, Buckley and Hayashi (1994), Buckley and Feuring (2000). In order to solve general fuzzy programming models, a natural idea is to integrate these algorithms to produce more powerful and effective *hybrid intelligent algorithms* (Liu, 1999b).The readers may get some source files of hybrid intelligent algorithms at http://orsc.edu.cn/~liu/Uncertain_Programming.

This section will introduce the general principle of designing hybrid intelligent algorithms. Essentially, there are three types of uncertain function arising in the area of fuzzy programming:

$$U_1 : \boldsymbol{x} \to E[f(\boldsymbol{x},\boldsymbol{\xi})],$$

$$U_2 : \boldsymbol{x} \to \mathrm{Pos/Cr/Nec}\{f_j(\boldsymbol{x},\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, p\},$$

$$U_3 : \boldsymbol{x} \to \max\{\bar{f} \mid \mathrm{Pos/Cr/Nec}\{f(\boldsymbol{x},\boldsymbol{\xi}) \geq \bar{f}\} \geq \alpha\}.$$

We may compute the uncertain functions by fuzzy simulation. However, fuzzy simulation is obviously a time-consuming process. In order to speed up the process, we may generate input-output data for each type of uncertain function. Then we train a feedforward NN to approximate the uncertain function using the generated training data. For solving general fuzzy programming models, we may embed the trained NN into GA, thus producing a hybrid intelligent algorithm. The general procedure of hybrid intelligent algorithm is listed as follows,

Step 1: Generate training input-output data for uncertain functions by fuzzy simulations.
Step 2: Train a neural network to approximate the uncertain functions according to the generated training data.

Step 3: Initialize *pop_size* chromosomes whose feasibility may be checked by the trained neural network.
Step 4: Update the chromosomes by crossover and mutation operations in which the feasibility of offspring may be checked by the trained neural networks.
Step 5: Calculate the objective values for all chromosomes by the trained neural networks.
Step 6: Compute the fitness of each chromosome according to the objective values.
Step 7: Select the chromosomes by spinning the roulette wheel.
Step 8: Repeat the fourth to seventh steps for a given number of cycles.
Step 9: Report the best chromosome as the optimal solution.

11. Related Topics

Fuzzy random variables are mathematical descriptions for fuzzy stochastic phenomena, and are defined in several ways. Kwakernaak (1978, 1979) first introduced the notion of fuzzy random variable. This concept was then developed by Puri and Ralescu (1986), Kruse and Meyer (1987), and Liu and Liu (2001b). Perhaps the definition given by Liu and Liu (2001b) is the most appropriate for fuzzy random optimization. Let $(\Omega, \mathcal{A}, \Pr)$ be a probability space, and $\mathcal{F}$ be a collection of fuzzy variables. A fuzzy random variable is a function ξ: $\Omega \to \mathcal{F}$ such that for any Borel set B of $\Re$, $\xi^*(B)(\omega) = \text{Pos}\{\xi(\omega)\} \in B\}$ is a measurable function of ω. The *primitive chance measure* of fuzzy random event $\xi \leq 0$ is defined by Liu (2001b) as a function from [0,1] to [0,1], i.e.,

$$\text{Ch}\{\xi \leq 0\}(\alpha) = \sup\{\beta \mid \Pr\{\omega \in \Omega \mid \text{Pos}\{\xi(\omega) \leq 0\} \geq \beta\} \geq \alpha\}.$$

We thus have fuzzy random CCP (Liu (2001a)), and fuzzy random DCP (Liu (2001b)). The *expected value* of ξ is defined by Liu and Liu (2001b) as

$$E[\xi] = \int_0^\infty \Pr\{\omega \in \Omega \mid E[\xi(\omega)] \geq r\}\mathrm{d}r - \int_{-\infty}^0 \Pr\{\omega \in \Omega \mid E[\xi(\omega)] \leq r\}\mathrm{d}r.$$

We thus have the theoretical framework of fuzzy random EVM (Liu and Liu (2001d)).

Random fuzzy variable was introduced by Liu (2001c), and defined as a function ξ from a possibility space to a collection of random variables. The *primitive chance* of random fuzzy event $\xi \leq 0$ is defined by Liu (2001c) as a function from [0,1] to [0,1], i.e.,

$$\text{Ch}\{\xi \leq 0\}(\alpha) = \sup\{\beta \mid \text{Pos}\{\theta \in \Theta \mid \Pr\{\xi(\theta) \leq 0\} \geq \beta\} \geq \alpha\}.$$

We thus have random fuzzy CCP (Liu (2001c)), and random fuzzy DCP (Liu (2001d)). The *expected value* $E[\xi]$ is defined by Liu and Liu (2001c) as

$$E[\xi] = \int_0^\infty \text{Cr}\{\theta \in \Theta \mid E[\xi(\theta)] \geq r\}\mathrm{d}r - \int_{-\infty}^0 \text{Cr}\{\theta \in \Theta \mid E[\xi(\theta)] \leq r\}\mathrm{d}r.$$

We thus have random fuzzy EVM (Liu and Liu (2001c)).

We may also define a *bifuzzy variable* as a function ξ from a possibility space to a collection of fuzzy variables. Then its *expected value* $E[\xi]$ is defined as

$$E[\xi] = \int_0^{\infty} \mathrm{Cr}\{\theta \in \Theta \mid E[\xi(\theta)] \geq r\}\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Cr}\{\theta \in \Theta \mid E[\boldsymbol{\xi}(\theta)] \leq r\}\mathrm{d}r.$$

The *primitive chance* of bifuzzy event $\xi \leq 0$ is defined as a function from [0,1] to [0,1], i.e.,

$$\mathrm{Ch}\{\xi \leq 0(\alpha) = \sup\{\beta \mid \mathrm{Pos}\{\theta \in \Theta \mid \mathrm{Pos}\{\xi(\theta) \leq 0\} \geq \beta\} \geq \alpha\}.$$

More generally, Liu (1999b) laid a foundation for optimization theory in uncertain (stochastic, fuzzy, fuzzy random, random fuzzy, etc.) environments, and called such a theory *uncertain programming*.

Fuzzy programming is a growing subject and there are numerous problems remaining to be solved. From the theoretical viewpoint, different versions of fuzzy programming will be made in fuzzy environments and more general mathematical analysis is needed. We have constructed single-objective programming, multiobjective programming and goal programming in fuzzy environments. We should also discuss multilevel programming and dynamic programming in fuzzy environments. For the mathematical properties of fuzzy programming, we should consider crisp equivalents, sensitivity analysis, upper/lower bound, dual theorems, and optimality conditions.

From the computational viewpoint, we should design more effective computer algorithms. A series of hybrid intelligent algorithms has been designed. Parallel computation is also worth mentioning. In addition, we may design some classical algorithms for special-structured fuzzy programming in the light of mathematical properties.

From the applied viewpoint, the most useful lines of advance are likely to be generated in the different applied fields such as (i) capital budgeting that is concerned with maximizing the total net profit subject to a budget constraint by selecting an appropriate combination of projects, in which profit and demands are assumed to be fuzzy; (ii) redundancy optimization that is to determine the optimal number of redundant elements for each component so as to maximize the system performances, in which the lifetimes are assumed to be fuzzy; (iii) parallel machine scheduling that is concerned with finding an efficient schedule during an uninterrupted period of time for a set of machines to process a set of jobs, in which the processing time and due date may be regarded as fuzzy variables; (iv) facility location-allocation that is to find locations for new facilities such that the conveying cost from facilities to customers is minimized, in which the demands and transportation costs are fuzzy variables; (v) critical path problem that is to choose a critical path through the network with the longest completion time, in which the lengths of arcs are assumed to be fuzzy; (vi) vehicle routing problem that is concerned with finding efficient routes, beginning and ending at a central depot, for a fleet of vehicles to serve a number of customers with demands for some commodity, in which the demands, travel times and time windows are assumed to be fuzzy; and (vii) inventory network, queuing system, manufacturing system, finance, energy system, pattern recognition, quality control, and risk analysis.

Acknowledgments

This work was supported by National Natural Science Foundation of China Grant No.69804006, and Sino-French Joint Laboratory for Research in Computer Science, Control and Applied Mathematics(LIAMA). The author would like to thank Prof. S.-C. Fang and Prof. Henry L.W. Nuttle for their valuable comments and suggestions.

References

Bouchon-Meunier B.,V. Kreinovich, A. Lokshin and H.T. Nguyen. (1996). "On the Formulation of Optimization Under Elastic Constraints (With Control in Mind)," *Fuzzy Sets and Systems* 81, 5–29.

Buckley, J.J. (1988). "Possibility and Necessity in Optimization," *Fuzzy Sets and Systems* 25, 1–13.

Buckley, J.J. (1990a). "Stochastic Versus Possibilistic Programming," *Fuzzy Sets and Systems* 34, 173–177.

Buckley, J.J. (1990b). "Multiobjective Possibilistic Linear Programming," *Fuzzy Sets and Systems* 35, 23–28.

Buckley, J.J., and Y. Hayashi. (1994). "Fuzzy Genetic Algorithm and Applications," *Fuzzy Sets and Systems* 61, 129–136.

Buckley, J.J., and T. Feuring. (2000). "Evolutionary Algorithm Solution to Fuzzy Problems: Fuzzy Linear Programming," *Fuzzy Sets and Systems* 109, 35–53.

Charnes, A. and W.W. Cooper. (1959). "Chance-constrained Programming," *Management Science* 6, 73–79.

Dubois, D. and H. Prade. (1988). *Possibility Theory.* New York: Plenum.

Inuiguchi, M., H. Ichihashi, and Y. Kume. (1993). "Modality Constrained Programming Problems: An Unified Approach to Fuzzy Mathematical Programming Problems in the Setting of Possibility Theory," *Information Science* 67, 93–126.

Inuiguchi, M., and J. Ramík. (2000). "Possibilistic Linear Programming: A Brief Review of Fuzzy Mathematical Programming and a Comparison with Stochastic Programming in Portfolio Selection Problem," *Fuzzy Sets and Systems* 111, 3–28.

Kruse, R., and K.D. Meyer. (1987). *Statistics with Vague Data*. Dordrecht: D. Reidel Publishing Company.

Kwakernaak, H. (1978). "Fuzzy Random Variables I," *Information Sciences* 15, 1–29.

Kwakernaak, H. (1979). "Fuzzy Random Variables II," *Information Sciences* 17, 153–178.

Liu, B. (1997). "Dependent-chance Programming: A Class of Stochastic Optimization," *Computers and Mathematics with Applications* 34, 89–104.

Liu, B. and K. Iwamura. (1998a). "Chance-constrained Programming with Fuzzy Parameters," *Fuzzy Sets and Systems* 94, 227–237.

Liu, B. and K. Iwamura. (1998b). "A Note on Chance-constrained Programming with Fuzzy Coefficients," *Fuzzy Sets and Systems* 100, 229–233.

Liu, B. (1998). "Minimax Chance-constrained Programming Models for Fuzzy Decision Systems," *Information Sciences* 112, 25–38.

Liu, B. (1999a). "Dependent-chance Programming with Fuzzy Decisions," *IEEE Transactions on Fuzzy Systems* 7, 354–360.

Liu, B. (1999b). *Uncertain Programming*. New York: John Wiley and Sons.

Liu, B. (2000). "Dependent-chance Programming in Fuzzy Environments," *Fuzzy Sets and Systems* 109, 95–104.

Liu, B. and K. Iwamura. (2001). "Fuzzy Programming with Fuzzy Decisions and Fuzzy Simulation-based Genetic Algorithm," *Fuzzy Sets and Systems* 122, 69–77.

Liu, B. (2001a). Fuzzy Random Chance-constrained Programming. *IEEE Transactions on Fuzzy Systems* Vol. 9, No. 5, 713–720.
Liu, B. (2001b). Fuzzy Random Dependent-chance Programming. *IEEE Transactions on Fuzzy Systems* Vol. 9, No. 5, 721–726.
Liu, B. (2001c). Random Fuzzy Variables and Random Fuzzy Programming. To appear in *Fuzzy Sets and Systems.*
Liu, B. (2001d). Random Fuzzy Dependent-chance Programming and its Hybrid Intelligent Algorithm. Technical Report.
Liu, B., and Y.-K. Liu. (2001a). Expected Value of Fuzzy Variable and Fuzzy Expected Value Models. To appear in *IEEE Transactions on Fuzzy Systems.*
Liu, Y.-K., and B. Liu. (2001b). Fuzzy Random Variables: A Scalar Expected Value. Technical Report.
Liu, Y.-K., and B. Liu. (2001c). Expected Value Operator of Random Fuzzy Variable and Random Fuzzy Expected Value Models. Technical Report.
Liu, Y.-K., and B. Liu. (2001d). Expected Value Models in Fuzzy Random Decision Systems with a Hybrid Intelligent Algorithm. Technical Report.
Luhandjula, M. K. (1983). "Linear Programming Under Randomness and Fuzziness," *Fuzzy Sets and Systems* 10, 45–55.
Luhandjula, M. K. (1986). "On Possibilistic Linear Programming," *Fuzzy Sets and Systems* 18, 15–30.
Luhandjula, M. K. (1989). "Fuzzy Optimization: An Appraisal," *Fuzzy Sets and Systems* 30, 257–282.
Nahmias, S. (1978). "Fuzzy Variables," *Fuzzy Sets and Systems* 1, 97–110.
Ostasiewicz, W. (1982). "A New Approach to Fuzzy Programming," *Fuzzy Sets and Systems* 7, 139–152.
Puri, M.L., and D. Ralescu. (1986). "Fuzzy Random Variables," *Journal of Mathematical Analysis and Applications* 114, 409–422.
Ramík, J., and H. Rommelfanger. (1996). "Fuzzy Mathematical Programming Based on some Inequality Relations," *Fuzzy Sets and Systems* 81, 77–88.
Tanaka, H., P. Guo, and H.-J. Zimmermann. (2000). "Possibility Distribution of Fuzzy Decision Variables Obtained from Possibilistic Linear Programming Problems," *Fuzzy Sets and Systems* 113, 323–332.
Yazenin, A.V. (1996). "On the Problem of Possibilistic Optimization," *Fuzzy Sets and Systems* 81, 133–140.
Zadeh, L.A. (1978). "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems* 1, 3–28.
Zimmermann, H.-J. (1983). "Fuzzy Mathematical Programming," *Computers and Operations Research* 10, 291–298.
Zimmermann, H.-J. (1985a). "Applications of Fuzzy Set Theory to Mathematical Programming," *Information Science* 36, 29–58.
Zimmermann, H.-J. (1985b). *Fuzzy Set Theory and its Applications*. Boston: Kluwer Academic Publishers.