Association for Information Systems
AIS Electronic Library (AISeL)

ICEB 2002 Proceedings | International Conference on Electronic Business (ICEB)

Winter 12-10-2002

Modeling Decision Systems via Uncertain Programming

Baoding Liu

Modeling Decision Systems via Uncertain Programming

Baoding Liu

Uncertainty Theory Laboratory, Department of Mathematical Sciences

Tsinghua University, Beijing 100084, China

http://orsc.edu.cn/∼liu

Abstract—**By uncertain programming we mean the optimization theory in generally uncertain (random, fuzzy, rough, fuzzy random, etc.) environments. The main purpose of this paper is to present a brief review on uncertain programming models, and classify them into three broad classes: expected value model, chance-constrained programming and dependent-chance programming. This presentation is based on the book: B. Liu, *Theory and Practice of Uncertain Programming*, Phisica-Verlag, Heidelberg, 2002.**

Keywords—**stochastic programming, fuzzy programming, uncertain programming, hybrid intelligent algorithm**

I. Introduction

Real-life decisions are usually made in the state of uncertainty. How do we model optimization problems in uncertain environments? How do we solve these models? Uncertain programming theory attempts to answer these questions.

By uncertain programming we mean the optimization theory in uncertain environments. The main topics of uncertain programming are stochastic programming, fuzzy programming, rough programming, fuzzy random programming, random fuzzy programming, random rough programming, rough random programming, fuzzy rough programming, rough fuzzy programming, birandom programming, bifuzzy programming, birough programming, and multifold uncertain 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 *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 [8] as a means of handling uncertainty by specifying a confidence level at which it is desired that the stochastic constraint holds. In practice, there usually are multiple events in a complex stochastic decision system. Sometimes the decision-maker wishes to maximize the chance functions of satisfying these events. In order to model this type of problem, Liu [39] provided 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 parameters. Fuzzy programming has been used in different ways in the past. Liu and Liu [57] 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 possibility level. Thus we have a fuzzy CCP theory. Analogously, following the idea of stochastic DCP, Liu [46] provided the fuzzy DCP theory.

Rough set theory, initialized by Pawlak [71], has been proved to be an excellent mathematical tool dealing with vague description of objects. A fundamental assumption in rough set theory is that any object from a universe is perceived through available information, and such information may not be sufficient to characterize the object exactly. Liu [56] presented a concept of rough space, and defined a rough variable as a measurable function from a rough space to the real line. Expected value operator of rough variable and trust measure of rough event are also suggested. Rough programming is thus proposed.

More generally, Liu [48][56] laid a foundation for optimization theory in uncertain environments, and called such a theory *uncertain programming*.

Fuzzy random variable was developed by Kwakernaak [31][32], and defined as a measurable function from a probability space to a collection of fuzzy variables. Fuzzy random programming is the theory dealing with optimization problems in fuzzy random environments, and has been made in several ways. We will introduce the theoretical framework of fuzzy random EVM (Liu and Liu [61]), fuzzy random CCP (Liu [54]), and fuzzy random DCP (Liu [55]).

Random fuzzy variable was initialized by Liu [56], and defined as a function from a possibility space to a collection of random variables. That is, a random fuzzy variable is a fuzzy variable defined on the universal set of random variables. As a tool of handling random fuzzy decision problems, random fuzzy programming includes random fuzzy EVM (Liu and Liu [60]), random fuzzy CCP (Liu [56]), and random fuzzy DCP (Liu [58]).

In addition to fuzzy random variable and random fuzzy variable, we have also other types of multifold uncertain variables: random rough variable, rough random variable, fuzzy rough variable, rough fuzzy variable, birandom variable, bifuzzy variable, and birough variable. Thus we have random rough programming, rough random programming, fuzzy rough programming, rough fuzzy programming, birandom programming, bifuzzy programming, birough programming, and multifold (three-fold, four-fold, etc.) uncertain programming (Liu [56]).

II. Various Types of Uncertainty

Fuzzy variable ξ is defined as a function from a possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$ to the real line $\Re$. 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 $f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \le 0, j = 1, 2, \cdots, m$ is defined by

$$\mathrm{Pos}\left\{f_j(\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_n) \le 0,\ j = 1, 2, \cdots, m\right\}$$

$$= \sup_{x_1, x_2, \cdots, x_n \in \Re} \left\{ \min_{1 \le i \le n} \mu_{\tilde{a}_i}(x_i) \;\middle|\; \begin{array}{r} f_j(x_1, x_2, \cdots, x_n) \le 0 \\ j = 1, 2, \cdots, m \end{array} \right\}.$$

The Second International Conference on Electronic Business
Taipei, Taiwan, December 10-13, 2002

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

Liu and Liu [57] presented an *expected value operator* of fuzzy variable. The expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathrm{Cr}\{\xi \ge r\}\mathrm{d}r - \int_{-\infty}^0 \mathrm{Cr}\{\xi \le r\}\mathrm{d}r. \tag{1}$$

Fuzzy random variables are mathematical descriptions for fuzzy stochastic phenomena, and are defined in several ways. Kwakernaak [31][32] first introduced the notion of fuzzy random variable. This concept was then developed by Puri and Ralescu [75], Kruse and Meyer [30], and Liu and Liu [59]. Liu and Liu [59] presented a definition of scalar expected value operator and several types of law of large numbers for fuzzy random variables. Liu [54][55] provided a concept of primitive chance measure of fuzzy random event.

Liu [56] initialized the concept of random fuzzy variable which is a function from a possibility space to a collection of random variables. The primitive chance measure of random fuzzy event was defined by Liu [56] as a function from [0,1] to [0,1]. The expected value operator of random fuzzy variable was given by Liu and Liu [60].

Let Λ be a nonempty set, $\mathcal{A}$ be a σ-algebra of subsets of Λ, Δ be an element in $\mathcal{A}$, and π be a nonnegative, real-valued, additive set function. Then $(\Lambda, \Delta, \mathcal{A}, \pi)$ is called a rough space. A rough variable ξ is defined by Liu [56] as a measurable function from the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to the real line $\Re$. The lower and the upper approximations of the rough variable ξ are then defined as follows,

$$\underline{\xi} = \left\{\xi(\lambda) \;\middle|\; \lambda \in \Delta\right\}, \quad \overline{\xi} = \left\{\xi(\lambda) \;\middle|\; \lambda \in \Lambda\right\}. \tag{2}$$

Let $\boldsymbol{\xi}$ be a rough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the upper trust of the rough event characterized by $f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m$ is defined by

$$\mathrm{T\overline{r}}\left\{\begin{array}{l} f_j(\boldsymbol{\xi}) \le 0 \\ j = 1, 2, \cdots, m \end{array}\right\} = \frac{\pi\left\{\lambda \in \Lambda | f_j(\boldsymbol{\xi}(\lambda)) \le 0,\ j = 1, 2, \cdots, m\right\}}{\pi\left\{\Lambda\right\}}.$$

Let $\boldsymbol{\xi}$ be a rough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the lower trust of the rough event characterized by $f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m$ is defined by

$$\mathrm{T\underline{r}}\left\{\begin{array}{l} f_j(\boldsymbol{\xi}) \le 0 \\ j = 1, 2, \cdots, m \end{array}\right\} = \frac{\pi\left\{\lambda \in \Delta | f_j(\boldsymbol{\xi}(\lambda)) \le 0,\ j = 1, 2, \cdots, m\right\}}{\pi\left\{\Delta\right\}}.$$

If $\pi\{\Delta\} = 0$, then we define

$$\mathrm{T\underline{r}}\left\{f_j(\boldsymbol{\xi}) \le 0,\ j = 1, 2, \cdots, m\right\} \equiv \mathrm{T\overline{r}}\left\{f_j(\boldsymbol{\xi}) \le 0,\ j = 1, 2, \cdots, m\right\}.$$

Let $\boldsymbol{\xi}$ be a rough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the trust of the rough event is defined as the average value of the lower and upper trusts.

Let ξ be a rough variable on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$. The expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathrm{Tr}\{\xi \ge r\}\mathrm{d}r - \int_{-\infty}^0 \mathrm{Tr}\{\xi \le r\}\mathrm{d}r. \tag{3}$$

A random rough variable is defined by Liu [56] as a function ξ from a rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to a collection of random variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\lambda) = \Pr\left\{\xi(\lambda) \in B\right\} \tag{4}$$

is a measurable function of λ. Let ξ be a random rough variable defined on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$. Then its expected value $E[\xi]$ is defined by

$$\int_0^{\infty} \mathrm{Tr}\left\{\lambda \in \Lambda | E[\xi(\lambda)] \ge r\right\}\mathrm{d}r - \int_{-\infty}^0 \Pr\left\{\lambda \in \Lambda | E[\xi(\lambda)] \le r\right\}\mathrm{d}r.$$

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a random rough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of random rough event characterized by $f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned} &\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m\right\}(\alpha) \\ &\quad = \sup\left\{\beta \;\middle|\; \mathrm{Tr}\left\{\boldsymbol{\lambda} \in \Lambda \;\middle|\; \Pr\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\lambda})) \le 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \ge \beta\right\} \ge \alpha\right\}. \end{aligned} \tag{5}$$

A rough random variable is defined by Liu [56] as a function ξ from a probability space $(\Omega, \mathcal{A}, \Pr)$ to a collection of rough variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\omega) = \mathrm{Tr}\left\{\xi(\omega) \in B\right\} \tag{6}$$

is a measurable function of ω. Let ξ be a rough random variable defined on the probability space $(\Omega, \mathcal{A}, \Pr)$. Then its expected value $E[\xi]$ is defined by

$$\int_0^{\infty} \Pr\left\{\omega \in \Omega | E[\xi(\omega)] \ge r\right\}\mathrm{d}r - \int_{-\infty}^0 \Pr\left\{\omega \in \Omega | E[\xi(\omega)] \le r\right\}\mathrm{d}r.$$

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a rough random vector on the probability space $(\Omega, \mathcal{A}, \Pr)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of rough random event characterized by $f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned} &\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m\right\}(\alpha) \\ &\quad = \sup\left\{\beta \;\middle|\; \Pr\left\{\boldsymbol{\omega} \in \Omega \;\middle|\; \mathrm{Tr}\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\omega})) \le 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \ge \beta\right\} \ge \alpha\right\}. \end{aligned}$$

A fuzzy rough variable is defined by Liu [56] as a function ξ from a rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to a collection of fuzzy variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\lambda) = \mathrm{Pos}\left\{\xi(\lambda) \in B\right\} \tag{7}$$

is a measurable function of λ. Let ξ be a fuzzy rough variable defined on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$. Then its expected value $E[\xi]$ is defined by

$$\int_0^{\infty} \mathrm{Tr}\left\{\lambda \in \Lambda | E[\xi(\lambda)] \ge r\right\}\mathrm{d}r - \int_{-\infty}^0 \mathrm{Tr}\left\{\lambda \in \Lambda | E[\xi(\lambda)] \le r\right\}\mathrm{d}r.$$

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a fuzzy rough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of fuzzy rough event characterized by $f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned} &\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, m\right\}(\alpha) \\ &\quad = \sup\left\{\beta \;\middle|\; \mathrm{Tr}\left\{\boldsymbol{\lambda} \in \Lambda \;\middle|\; \mathrm{Pos}\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\lambda})) \le 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \ge \beta\right\} \ge \alpha\right\}. \end{aligned}$$

A rough fuzzy variable is defined by Liu [56] as a function from a possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$ to a collection of rough variables. Let ξ be a rough fuzzy variable defined on the possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$. The expected value $E[\xi]$ is defined by

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

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a rough fuzzy vector on the possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of rough fuzzy event characterized by $f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned}&\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m\right\}(\alpha)\\ &\quad = \sup\left\{\beta \;\Big|\; \mathrm{Pos}\left\{\boldsymbol{\theta} \in \Theta \;\Big|\; \mathrm{Tr}\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\theta})) \leq 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \geq \beta\right\} \geq \alpha\right\}.\end{aligned}$$

A birandom variable is a function ξ from a probability space $(\Omega, \mathcal{A}, \mathrm{Pr})$ to a collection of random variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\omega) = \Pr\{\xi(\omega) \in B\} \tag{8}$$

is a measurable function of ω. Let ξ be a birandom variable defined on the probability space $(\Omega, \mathcal{A}, \mathrm{Pr})$. Then its expected value $E[\xi]$ is defined by

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

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a birandom vector on the probability space $(\Omega, \mathcal{A}, \mathrm{Pr})$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of birandom event characterized by $f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned}&\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m\right\}(\alpha)\\ &\quad = \sup\left\{\beta \;\Big|\; \Pr\left\{\boldsymbol{\omega} \in \Omega \;\Big|\; \Pr\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\omega})) \leq 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \geq \beta\right\} \geq \alpha\right\}.\end{aligned}$$

A bifuzzy variable is defined by Liu [53] as a function from a possibility space to a collection of fuzzy variables. Let ξ be a bifuzzy variable defined on the possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$. The expected value $E[\xi]$ is defined by Liu [53] as

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

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a bifuzzy vector on the possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of bifuzzy event characterized by $f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined by Liu [53] as

$$\begin{aligned}&\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m\right\}(\alpha)\\ &\quad = \sup\left\{\beta \;\Big|\; \mathrm{Pos}\left\{\boldsymbol{\theta} \in \Theta \;\Big|\; \mathrm{Pos}\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\theta})) \leq 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \geq \beta\right\} \geq \alpha\right\}.\end{aligned}$$

A birough variable is defined by Liu [56] as a function ξ from a rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to a collection of rough variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\lambda) = \mathrm{Tr}\{\xi(\lambda) \in B\} \tag{9}$$

is a measurable function of λ. Let ξ be a birough variable on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$. Then its expected value $E[\xi]$ is defined by

$$\int_0^{\infty} \mathrm{Tr}\{\lambda \in \Lambda | E[\xi(\lambda)] \geq r\}\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Tr}\{\lambda \in \Lambda | E[\xi(\lambda)] \leq r\}\mathrm{d}r.$$

Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)$ be a birough vector on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$, and $f_j : \Re^n \to \Re$ be continuous functions, $j = 1, 2, \cdots, m$. Then the primitive chance of birough event characterized by $f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m$ is a function from $[0, 1]$ to $[0, 1]$, defined as

$$\begin{aligned}&\mathrm{Ch}\left\{f_j(\boldsymbol{\xi}) \leq 0, j = 1, 2, \cdots, m\right\}(\alpha)\\ &\quad = \sup\left\{\beta \;\Big|\; \mathrm{Tr}\left\{\boldsymbol{\lambda} \in \Lambda \;\Big|\; \mathrm{Tr}\left\{\begin{array}{c} f_j(\boldsymbol{\xi}(\boldsymbol{\lambda})) \leq 0 \\ j = 1, 2, \cdots, m \end{array}\right\} \geq \beta\right\} \geq \alpha\right\}.\end{aligned}$$

A trirandom variable is a function ξ from a probability space $(\Omega, \mathcal{A}, \mathrm{Pr})$ to a collection of birandom variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\omega) = \mathrm{Ch}\{\xi(\omega) \in B\} \tag{10}$$

is a measurable function of ω.

A trifuzzy variable is a function from a possibility space to a collection of bifuzzy variables.

A trirough variable is a function ξ from a rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to a collection of birough variables such that for any Borel set B of $\Re$,

$$\xi^*(B)(\lambda) = \mathrm{Ch}\{\xi(\lambda) \in B\} \tag{11}$$

is a measurable function of λ.

We may also define other three-fold uncertainty. For example, a fuzzy random rough variable is a function from a rough space to a collection of fuzzy random variables such that for any Borel set B of $\Re$, $\xi^*(B)(\lambda) = \mathrm{Ch}\{\xi(\lambda) \in B\}$ is a measurable function of λ.

III. Ranking Uncertain Variables

Let ξ and η be two uncertain variables. Different from the situation of real numbers, there does not exist a natural ordership in an uncertain world. Thus an important problem appearing in uncertain systems is how to rank uncertain variables. 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 uncertain variables. This criterion leads to expected value models.

(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 criterion leads to maximax chance-constrained programming.

(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 criterion leads to minimax chance-constrained programming.

(iv) We say $\xi > \eta$ if and only if $\mathrm{Ch}\{\xi \geq \bar{r}\} > \mathrm{Ch}\{\eta \geq \bar{r}\}$ for some predetermined level $\bar{r}$. This criterion leads to dependent-chance programming.

IV. Expected Value Models

The first type of uncertain programming is the expected value model (EVM) in which the underlying philosophy is based on selecting the decisions with maximum expected value of return.

We have stochastic, fuzzy, rough, fuzzy random, random fuzzy, fuzzy rough, rough fuzzy, random rough,

rough random, birandom, bifuzzy, and birough EVM. The general form of EVM is formulated as follows,

$$\begin{cases} \max \ E[f(\boldsymbol{x}, \boldsymbol{\xi})] \\ \textbf{subject to:} \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{12}$$

where $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is an uncertain vector, $f(\boldsymbol{x}, \boldsymbol{\xi})$ is the return function, $g_j(\boldsymbol{x}, \boldsymbol{\xi})$ are uncertain constraint functions for $j = 1, 2, \cdots, p$, and E denotes the expected value operator.

The expected value multiobjective programming (EVMOP) model has the following form,

$$\begin{cases} \max \left[E[f_1(\boldsymbol{x}, \boldsymbol{\xi})], E[f_2(\boldsymbol{x}, \boldsymbol{\xi})], \cdots, E[f_m(\boldsymbol{x}, \boldsymbol{\xi})]\right] \\ \textbf{subject to:} \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \quad j = 1, 2, \cdots, p \end{cases} \tag{13}$$

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

We can also formulate an uncertain decision system as an expected value goal programming (EVGP) model according to the priority structure and target levels set by the decision-maker,

$$\begin{cases} \min \displaystyle\sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \textbf{subject to:} \\ \qquad E[f_i(\boldsymbol{x}, \boldsymbol{\xi})] + d_i^- - d_i^+ = b_i, \quad i = 1, 2, \cdots, m \\ \qquad E[g_j(\boldsymbol{x}, \boldsymbol{\xi})] \le 0, \qquad\qquad j = 1, 2, \cdots, p \\ \qquad d_i^+, d_i^- \ge 0, \qquad\qquad\qquad i = 1, 2, \cdots, m \end{cases} \tag{14}$$

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, 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.

V. Maximax Chance-Constrained Programming

In this section, we provide a spectrum of maximax chance-constrained programming (CCP) models in which the underlying philosophy is based on selecting the alternative that provides the best optimistic return with a given confidence level. In order to do so, we may measure the uncertain return by its optimistic value.

We have stochastic, fuzzy, rough, fuzzy random, random fuzzy, fuzzy rough, rough fuzzy, random rough, rough random, birandom, bifuzzy, and birough CCP. A single-objective maximax CCP model may be written as follows,

$$\begin{cases} \max \overline{f} \\ \textbf{subject to:} \\ \qquad \text{Ch}\left\{f(\boldsymbol{x}, \boldsymbol{\xi}) \ge \overline{f}\right\} \ge \beta \\ \qquad \text{Ch}\left\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, p\right\} \ge \alpha \end{cases} \tag{15}$$

where α and β are predetermined confidence levels, and $\max \overline{f}$ will be the β-optimistic return.

Maximax chance-constrained multiobjective programming (CCMOP) may be written as follows,

$$\begin{cases} \max \left[\overline{f}_1, \overline{f}_2, \cdots, \overline{f}_m\right] \\ \textbf{subject to:} \\ \qquad \text{Ch}\left\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) \ge \overline{f}_i\right\} \ge \beta_i, \quad i = 1, 2, \cdots, m \\ \qquad \text{Ch}\left\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\right\} \ge \alpha_j, \quad j = 1, 2, \cdots, p. \end{cases} \tag{16}$$

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

$$\begin{cases} \min \displaystyle\sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \textbf{subject to:} \\ \qquad \text{Ch}\left\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \le d_i^+\right\} \ge \beta_i^+, \quad i = 1, 2, \cdots, m \\ \qquad \text{Ch}\left\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \le d_i^-\right\} \ge \beta_i^-, \quad i = 1, 2, \cdots, m \\ \qquad \text{Ch}\left\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\right\} \ge \alpha_j, \qquad\quad j = 1, 2, \cdots, p \\ \qquad d_i^+, d_i^- \ge 0, \qquad\qquad\qquad\quad i = 1, 2, \cdots, m \end{cases} \tag{17}$$

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, d_i^- is the β_i^--optimistic negative deviation from the target of goal i, 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, and p is the number of real constraints.

VI. Minimax Chance-Constrained Programming

Murphy's law states that "if anything can go wrong, it will". If you believe it, you may select the decision with the best of these worst returns. In order to do so, we may measure the uncertain return by its pessimistic value.

As opposed to optimistic models, we have a spectrum of minimax CCP models in which the underlying philosophy is based on selecting the alternative with the best pessimistic return with a given confidence level. A single-objective minimax CCP model may be written as follows,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \min\limits_{\overline{f}} \overline{f} \\ \textbf{subject to:} \\ \qquad \text{Ch}\left\{f(\boldsymbol{x}, \boldsymbol{\xi}) \le \overline{f}\right\} \ge \beta \\ \qquad \text{Ch}\left\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, p\right\} \ge \alpha \end{cases} \tag{18}$$

where $\min \overline{f}$ is the β-pessimistic return.

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

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

According to the priority structure and target levels, the minimax CCGP model is written as follows,

$$\begin{cases} \min\limits_{\boldsymbol{x}} \displaystyle\sum_{j=1}^{l} P_j \sum_{i=1}^{m} \left[u_{ij} \left(\max_{d_i^+} d_i^+ \vee 0\right) + v_{ij} \left(\max_{d_i^-} d_i^- \vee 0\right)\right] \\ \textbf{subject to:} \\ \qquad \text{Ch}\left\{f_i(\boldsymbol{x}, \boldsymbol{\xi}) - b_i \ge d_i^+\right\} \ge \beta_i^+, \quad i = 1, 2, \cdots, m \\ \qquad \text{Ch}\left\{b_i - f_i(\boldsymbol{x}, \boldsymbol{\xi}) \ge d_i^-\right\} \ge \beta_i^-, \quad i = 1, 2, \cdots, m \\ \qquad \text{Ch}\left\{g_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\right\} \ge \alpha_j, \qquad\quad j = 1, 2, \cdots, p \end{cases}$$

where $d_i^+ \vee 0$ is the β_i^+-pessimistic positive deviation from the target of goal i, and $d_i^- \vee 0$ is the β_i^--pessimistic negative deviation from the target of goal i.

VII. Dependent-Chance Programming

In order to model uncertain decision problems, we may employ dependent-chance programming (DCP) in which the underlying philosophy is based on selecting the decisions with maximum chance to meet the events. We thus have stochastic, fuzzy, rough, fuzzy random, random fuzzy, fuzzy rough, rough fuzzy, random rough, rough random, birandom, bifuzzy, and birough DCP.

A typical DCP in an uncertain environment is given as follows:

$$\begin{cases} \max \mathrm{Ch}\{h_k(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k=1,2,\cdots,q\} \\ \textbf{subject to:} \\ \qquad g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, \quad j=1,2,\cdots,p \end{cases} \tag{20}$$

where $\boldsymbol{x}$ is a decision vector, $\boldsymbol{\xi}$ is an uncertain vector, then event is characterized by $h_k(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k = 1,2,\cdots,q$, and the uncertain environment is constrained as $g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, j=1,2,\cdots,p$.

Since a complex decision system usually undertakes multiple tasks, there undoubtedly exist multiple potential objectives in the decision process. A dependent-chance multiobjective programming (DCMOP) in an uncertain environment has the following form,

$$\begin{cases} \max \begin{bmatrix} \mathrm{Ch}\{h_{1k}(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k=1,2,\cdots,q_1\} \\ \mathrm{Ch}\{h_{2k}(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k=1,2,\cdots,q_2\} \\ \cdots \\ \mathrm{Ch}\{h_{mk}(\boldsymbol{x},\boldsymbol{\xi}) \le 0, k=1,2,\cdots,q_m\} \end{bmatrix} \\ \textbf{subject to:} \\ \qquad g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, \quad j=1,2,\cdots,p. \end{cases} \tag{21}$$

We can also formulate an uncertain decision system as a dependent-chance goal programming (DCGP) according to the priority structure and target levels,

$$\begin{cases} \min \sum_{j=1}^{l} P_j \sum_{i=1}^{m} (u_{ij} d_i^+ + v_{ij} d_i^-) \\ \textbf{subject to:} \\ \mathrm{Ch}\left\{ \begin{array}{c} h_{ik}(\boldsymbol{x},\boldsymbol{\xi}) \le 0 \\ k=1,2,\cdots,q_i \end{array} \right\} + d_i^- - d_i^+ = b_i, \ i=1,2,\cdots,m \\ g_j(\boldsymbol{x},\boldsymbol{\xi}) \le 0, \ j=1,2,\cdots,p \\ d_i^+, d_i^- \ge 0, \ i=1,2,\cdots,m \end{cases}$$

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

VIII. Uncertain Dynamic Programming

Let us consider an N-stage decision system in which $\widehat{\boldsymbol{a}} = (\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_N)$ represents the state vector, $\widehat{\boldsymbol{x}} = (\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$ the decision vector, $(\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_N)$ the uncertain vector. We also assume that the state transition function is

$$\boldsymbol{a}_{i+1} = T(\boldsymbol{a}_i,\boldsymbol{x}_i,\boldsymbol{\xi}_i), \quad i=1,2,\cdots,N-1. \tag{22}$$

Expected Value Dynamic Programming

In order to maximize the discount expected return over the horizon, we may use the following expected value dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \max\limits_{E[g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N)] \le 0} E[r_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N)] \\ f_n(\boldsymbol{a}) = \max\limits_{E[g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)] \le 0} E[r_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) + \theta f_{n+1}(T(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)] \\ \quad n \le N-1 \end{cases}$$

where r_i are the return functions at the ith stages, $i=1,2,\cdots,N$, respectively, θ is a discount rate, $0 \le \theta \le 1$, and E denotes the expected value operator. This type of uncertain (especially stochastic) dynamic programming has been applied to a wide variety of problems, for example, inventory systems.

Chance-Constrained Dynamic Programming

In order to maximize the discount optimistic return over the horizon, we may use the following chance-constrained dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \max\limits_{\mathrm{Ch}\{g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \le 0\} \ge \alpha} \overline{r}_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \\ f_n(\boldsymbol{a}) = \max\limits_{\mathrm{Ch}\{g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0\} \ge \alpha} \{\overline{r}_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) + \theta f_{n+1}(T(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)\} \\ \quad n \le N-1 \end{cases}$$

where the functions $\overline{r}_i$ are defined by

$$\overline{r}_i(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) = \sup_{\overline{r}} \left\{ \overline{r} \;\middle|\; \mathrm{Ch}\{r_i(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) \ge \overline{r}\} \ge \beta \right\} \tag{23}$$

for $i=1,2,\cdots,N$. If we want to maximize the discount pessimistic return over the horizon, then we must define the functions $\overline{r}_i$ as

$$\overline{r}_i(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) = \inf_{\overline{r}} \left\{ \overline{r} \;\middle|\; \mathrm{Ch}\{r_i(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) \le \overline{r}\} \ge \beta \right\} \tag{24}$$

for $i=1,2,\cdots,N$.

Dependent-Chance Dynamic Programming

In order to maximize the discount chance over the horizon, we may employ the following dependent-chance dynamic programming,

$$\begin{cases} f_N(\boldsymbol{a}) = \max\limits_{g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \le 0} \mathrm{Ch}\{h_N(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_N) \le 0\} \\ f_n(\boldsymbol{a}) = \max\limits_{g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0} \{\mathrm{Ch}\{h_n(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n) \le 0\} + \theta f_{n+1}(T(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_n)\} \\ \quad n \le N-1 \end{cases}$$

where $h_i(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) \le 0$ are the events, and $g(\boldsymbol{a},\boldsymbol{x},\boldsymbol{\xi}_i) \le 0$ are the uncertain environmrnts at the ith stages, $i = 1,2,\cdots,N$, respectively.

IX. Uncertain Multilevel Programming

Assume that in a decentralized two-level decision system there is one leader and m followers. Let $\boldsymbol{x}$ and $\boldsymbol{y}_i$ be the control vectors of the leader and the ith followers, $i=1,2,\cdots,m$, respectively. We also assume that the objective functions of the leader and ith followers are $F(\boldsymbol{x},\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})$ and $f_i(\boldsymbol{x},\boldsymbol{y}_1,\cdots,\boldsymbol{y}_m,\boldsymbol{\xi})$, $i=1,2,\cdots,m$, respectively, where $\boldsymbol{\xi}$ is an uncertain vector.

Expected Value Multilevel Programming

Let the feasible set of control vector $\boldsymbol{x}$ of the leader be defined by the expected constraint

$$E[G(\boldsymbol{x},\boldsymbol{\xi})] \le 0 \tag{25}$$

where G is a vector-valued function and 0 is a vector with zero components. Then for each decision $\boldsymbol{x}$ chosen by the leader, the feasibility of control array $(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)$ of followers should be dependent on $\boldsymbol{x}$, and generally represented by the expected constraint,

$$E[g(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \le 0 \tag{26}$$

where g is a vector-valued function.

Assume that the leader first chooses his control vector $\boldsymbol{x}$, and the followers determine their control array $(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)$ after that. In order to maximize the expected return of the leader, we have the following expected value bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} E[F(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad E[G(\boldsymbol{x}, \boldsymbol{\xi})] \le 0 \\ \qquad \text{where each } \boldsymbol{y}_i (i = 1, 2, \cdots, m) \text{ solves} \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y_i}} E[f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \\ \text{subject to:} \\ \qquad E[g(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi})] \le 0. \end{cases} \end{cases} \tag{27}$$

A *Nash equilibrium* of followers is the feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ with respect to $\boldsymbol{x}$ if

$$\begin{aligned} &E[f_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \\ &\quad \le E[f_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i^*, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \end{aligned} \tag{28}$$

for any feasible $(\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \cdots, m$. Let $\boldsymbol{x}^*$ be a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ be a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. We call the array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ a *Stackelberg-Nash equilibrium* to the expected value bilevel programming (27) if and only if,

$$E[F(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m, \boldsymbol{\xi})] \le E[F(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi})] \tag{29}$$

for any feasible $\overline{\boldsymbol{x}}$ and the Nash equilibrium $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m)$ with respect to $\overline{\boldsymbol{x}}$.

Chance-Constrained Multilevel Programming

In order to maximize the optimistic return subject to the chance constraint, we may use the following chance-constrained bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \overline{F} \\ \text{subject to:} \\ \qquad \text{Ch}\{F(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \ge \overline{F}\} \ge \beta \\ \qquad \text{Ch}\{G(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \alpha \\ \qquad \text{where each } \boldsymbol{y}_i (i = 1, 2, \cdots, m) \text{ solves} \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y_i}} \overline{f}_i \\ \text{subject to:} \\ \qquad \text{Ch}\{f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \ge \overline{f}_i\} \ge \beta_i \\ \qquad \text{Ch}\{g(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0\} \ge \alpha_i \end{cases} \end{cases} \tag{30}$$

where $\alpha, \beta, \alpha_i, \beta_i,\ i = 1, 2, \cdots, m$ are predetermined confidence levels.

A *Nash equilibrium* of followers is the feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ with respect to $\boldsymbol{x}$ if

$$\begin{aligned} &\overline{f}_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*) \\ &\quad \le \overline{f}_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i^*, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*) \end{aligned} \tag{31}$$

for any feasible array $(\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \cdots, m$. Let $\boldsymbol{x}^*$ be a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ be a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. The array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ is called a *Stackelberg-Nash equilibrium* to the chance-constrained bilevel programming (30) if and only if,

$$\overline{F}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m) \le \overline{F}(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*) \tag{32}$$

for any feasible control vector $\overline{\boldsymbol{x}}$ and the Nash equilibrium $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m)$ with respect to $\overline{\boldsymbol{x}}$.

In order to maximize the pessimistic return, we have the following minimax chance-constrained bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \min\limits_{\overline{F}} \overline{F} \\ \text{subject to:} \\ \qquad \text{Ch}\{F(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le \overline{F}\} \ge \beta \\ \qquad \text{Ch}\{G(\boldsymbol{x}, \boldsymbol{\xi}) \le 0\} \ge \alpha \\ \qquad \text{where each } \boldsymbol{y}_i (i = 1, 2, \cdots, m) \text{ solves} \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y_i}} \min\limits_{\overline{f}_i} \overline{f}_i \\ \text{subject to:} \\ \qquad \text{Ch}\{f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le \overline{f}_i\} \ge \beta_i \\ \qquad \text{Ch}\{g(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0\} \ge \alpha_i. \end{cases} \end{cases} \tag{33}$$

Dependent-Chance Multilevel Programming

Let $H(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0$ and $h_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0$ be the events of the leader and ith followers, $i = 1, 2, \cdots, m$, respectively. In order to maximize the chance function of the leader, we have the following dependent-chance bilevel programming,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \text{Ch}\{H(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0\} \\ \text{subject to:} \\ \qquad G(\boldsymbol{x}, \boldsymbol{\xi}) \le 0 \\ \qquad \text{where each } \boldsymbol{y}_i (i = 1, 2, \cdots, m) \text{ solves} \\ \qquad \begin{cases} \max\limits_{\boldsymbol{y_i}} \text{Ch}\{h_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0\} \\ \text{subject to:} \\ \qquad g(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m, \boldsymbol{\xi}) \le 0. \end{cases} \end{cases} \tag{34}$$

The feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ is called a *Nash equilibrium* of followers with respect to $\boldsymbol{x}$ if

$$\begin{aligned} &\text{Ch}\{h_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi}) \le 0\} \\ &\quad \le \text{Ch}\{h_i(\boldsymbol{x}, \boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i^*, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi}) \le 0\} \end{aligned} \tag{35}$$

for any feasible array $(\boldsymbol{y}_1^*, \cdots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \cdots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \cdots, m$. Let $\boldsymbol{x}^*$ be a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ be a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. We call the array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*)$ a *Stackelberg-Nash equilibrium* to the dependent-chance bilevel programming (34) if and only if,

$$\begin{aligned} &\text{Ch}\{H(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m, \boldsymbol{\xi}) \le 0\} \\ &\quad \le \text{Ch}\{H(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \cdots, \boldsymbol{y}_m^*, \boldsymbol{\xi}) \le 0\} \end{aligned}$$

for any feasible control vector $\overline{\boldsymbol{x}}$ and the Nash equilibrium $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \cdots, \overline{\boldsymbol{y}}_m)$ with respect to $\overline{\boldsymbol{x}}$.

X. Hybrid Intelligent Algorithms

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:

$$\begin{aligned} &U_1 : \boldsymbol{x} \to E[f(\boldsymbol{x}, \boldsymbol{\xi})], \\ &U_2 : \boldsymbol{x} \to \text{Pos/Cr/Nec}\{f_j(\boldsymbol{x}, \boldsymbol{\xi}) \le 0, j = 1, 2, \cdots, p\}, \\ &U_3 : \boldsymbol{x} \to \max\left\{\overline{f} \;\middle|\; \text{Pos/Cr/Nec}\left\{f(\boldsymbol{x}, \boldsymbol{\xi}) \ge \overline{f}\right\} \ge \alpha\right\}. \end{aligned}$$

We may compute the uncertain functions by simulations. 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.**

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).

References

[1] Bellman, R.E., and Zadeh, L.A., Decision making in a fuzzy environment, *Management Science*, Vol. 17, 141–164, 1970.

[2] Bouchon-Meunier, B., Kreinovich, V., Lokshin, A., and Nguyen, H.T., On the formulation of optimization under elastic constraints (with control in mind), *Fuzzy Sets and Systems*, Vol. 81, 5–29, 1996.

[3] Buckley, J.J., Possibility and necessity in optimization, *Fuzzy Sets and Systems*, Vol. 25, 1–13, 1988.

[4] Buckley, J.J., Stochastic versus possibilistic programming, *Fuzzy Sets and Systems*, Vol. 34, 173–177, 1990.

[5] Buckley, J.J., Multiobjective possibilistic linear programming, *Fuzzy Sets and Systems*, Vol. 35, 23–28, 1990.

[6] Buckley, J.J., and Hayashi, Y., Fuzzy genetic algorithm and applications, *Fuzzy Sets and Systems*, Vol. 61, 129–136, 1994.

[7] Buckley, J.J., and Feuring, T., Evolutionary algorithm solution to fuzzy problems: Fuzzy linear programming, *Fuzzy Sets and Systems*, Vol. 109, No. 1, 35–53, 2000.

[8] Charnes, A. and Cooper, W.W., Chance-constrained programming, *Management Science*, Vol. 6, No. 1, 73–79, 1959.

[9] Charnes, A. and Cooper, W.W., *Management Models and Industrial Applications of Linear Programming,* Wiley, New York, 1961.

[10] Dubois, D., and Prade, H., *Fuzzy Sets and Systems, Theory and Applications*, Academic Press, New York, 1980.

[11] Dubois, D., and Prade, H., Fuzzy cardinality and the modeling of imprecise quantification, *Fuzzy Sets and Systems*, Vol. 16, 199–230, 1985.

[12] Dubois, D., and Prade, H., *Possibility Theory*, Plenum, New York, 1988.

[13] Dubois, D., and Prade, H., Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems*, Vol. 17, 191–200, 1990.

[14] Gao, J., and Liu, B., New primitive chance measures of fuzzy random event, *International Journal of Fuzzy Systems*, Vol. 3, No. 4, 527-531, 2001.

[15] Gen, M., Liu, B., and Ida, K., Evolution program for deterministic and stochastic optimizations, *European Journal of Operational Research*, Vol. 94, No. 3, 618–625, 1996.

[16] Inuiguchi, M., Ichihashi, H., and Kume Y., Relationships between modality constrained programming problems and various fuzzy mathematical programming problems, *Fuzzy Sets and Systems*. Vol. 49, 243–259, 1992.

[17] Inuiguchi, M., Ichihashi, H., and Kume, Y., Modality constrained programming problems: An unified approach to fuzzy mathematical programming problems in the setting of possibility theory, *Information Sciences*, Vol. 67, 93–126, 1993.

[18] Inuiguchi, M., and Ramík, J., Possibilistic linear programming: A brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem, *Fuzzy Sets and Systems*, Vol. 111, No. 1, 3–28, 2000.

[19] Iwamura, K. and Liu, B., A genetic algorithm for chance constrained programming, *Journal of Information & Optimization Sciences*, Vol. 17, No. 2, 40–47, 1996.

[20] Iwamura, K. and Liu, B., Chance constrained integer programming models for capital budgeting in fuzzy environments, *Journal of the Operational Research Society*, Vol. 49, No. 8, 854–860, 1998.

[21] Iwamura, K. and Liu, B., Stochastic operation models for open inventory networks, *Journal of Information & Optimization Sciences*, Vol.20, No. 3, 347-363, 1999.

[22] Iwamura, K. and Liu, B., Dependent-chance integer programming applied to capital budgeting, *Journal of the Operations Research Society of Japan*, Vol. 42, No. 2, 117–127, 1999.

[23] Kall, P. and Wallace, S.W., *Stochastic Programming,* Wiley, Chichester, 1994.

[24] Kaufman, A. and Gupta, M.M., *Introduction to Fuzzy Arithmetic: Theory and Applications*, Van Nostrand Reinhold, New York, 1985.

[25] Kaufman, A. and Gupta, M.M., *Fuzzy Mathematical Models in Engineering and Management Science*, 2nd ed., North-Holland, Amsterdam, 1991.

[26] Keown, A.J. and Martin, J.D., A chance constrained goal programming model for working capital management, *Engng Econ.*, Vol. 22, 153–174, 1977.

[27] Keown, A.J., A chance-constrained goal programming model for bank liquidity management, *Decision Sciences*, Vol. 9, 93–106, 1978.

[28] Keown, A.J. and Taylor, B.W., A chance-constrained integer goal programming model for capital budgeting in the production area, *Journal of the Operational Research Society*, Vol. 31, No. 7, 579–589, 1980.

[29] Klir, G.J. and Folger, T.A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[30] Kruse, R., and Meyer, K.D., *Statistics with Vague Data*, D. Reidel Publishing Company, Dordrecht, 1987.

[31] Kwakernaak, H., Fuzzy random variables I, *Information Sciences*, Vol. 15, 1–29, 1978.

[32] Kwakernaak H., Fuzzy random variables II, *Information Sciences*, Vol. 17, 153–178, 1979.

[33] Lai, Y.-J. and Hwang, C.-L., A new approach to some possibilistic linear programming problems, *Fuzzy Sets and Systems*, Vol. 49, 121–133, 1992.

[34] Lai, Y.-J. and Hwang, C.-L., *Fuzzy Multiple Objective Decision Making: Methods and Applications*, Springer-Verlag, New York, 1994.

[35] Lee, E.S., and Li, R.J., Fuzzy multiple objective programming and compromise programming with Pareto optimum, *Fuzzy Sets and Systems*, Vol. 53, 275–288, 1993.

[36] Lee, E.S., Fuzzy multiple level programming, *Applied Mathematics and Computation*, Vol. 120, 79–90, 2001.

[37] Liu, B. and Ku, C., Dependent-chance goal programming and an application, *Journal of Systems Engineering & Electronics*, Vol. 4, No. 2, 40–47, 1993.

[38] Liu, B., Dependent-chance goal programming and its genetic algorithm based approach, *Mathematical and Computer Modelling*, Vol. 24, No. 7, 43–52, 1996.

[39] Liu, B., Dependent-chance programming: A class of stochas-

tic programming, *Computers & Mathematics with Applications*, Vol. 34, No. 12, 89–104, 1997.
[40] Liu, B., and Iwamura, K., Modelling stochastic decision systems using dependent-chance programming, *European Journal of Operational Research*, Vol. 101, No. 1, 193–203, 1997.
[41] Liu, B., and Iwamura, K., Chance constrained programming with fuzzy parameters, *Fuzzy Sets and Systems*, Vol. 94, No. 2, 227–237, 1998.
[42] Liu, B., and Iwamura, K., A note on chance constrained programming with fuzzy coefficients, *Fuzzy Sets and Systems*, Vol. 100, Nos. 1–3, 229–233, 1998.
[43] Liu, B., Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms, *Computers & Mathematics with Applications*, Vol. 36, No. 7, 79–89, 1998.
[44] Liu, B., Minimax chance constrained programming models for fuzzy decision systems, *Information Sciences*, Vol. 112, Nos. 1–4, 25–38, 1998.
[45] Liu, B., and Zhao, R., *Stochastic Programming and Fuzzy Programming*, Tsinghua University Press, Beijing, 1998.
[46] Liu, B., Dependent-chance programming with fuzzy decisions, *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 3, 354–360, 1999.
[47] Liu, B., and Esogbue, A.O., *Decision Criteria and Optimal Inventory Processes*, Kluwer Academic Publishers, Boston, 1999.
[48] Liu, B., *Uncertain Programming*, Wiley, New York, 1999.
[49] Liu, B., Dependent-chance programming in fuzzy environments, *Fuzzy Sets and Systems*, Vol. 109, No. 1, 97–106, 2000.
[50] Liu, B., and Iwamura, K., Topological optimization models for communication network with multiple reliability goals, *Computers & Mathematics with Applications*, Vol. 39, 59–69, 2000.
[51] Liu, B., Uncertain programming: A unifying optimization theory in various uncertain environments, *Applied Mathematics and Computation*, Vol. 120, Nos. 1–3, 227–234, 2001.
[52] Liu, B., and Iwamura, K., Fuzzy programming with fuzzy decisions and fuzzy simulation-based genetic algorithm, *Fuzzy Sets and Systems*, Vol. 122, No. 2, 253–262, 2001.
[53] Liu, B., Toward fuzzy optimization without mathematical ambiguity, *Fuzzy Optimization and Decision Making*, Vol. 1, No. 1, 43-63, 2002.
[54] Liu, B., Fuzzy random chance-constrained programming, *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 5, 713–720, 2001.
[55] Liu, B., Fuzzy random dependent-chance programming, *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 5, 721–726, 2001.
[56] Liu, B., *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg, 2002.
[57] Liu, B., and Liu, Y.-K., Expected value of fuzzy variable and fuzzy expected value models, *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 4, 445-450, 2002.
[58] Liu, B., Random fuzzy dependent-chance programming and its hybrid intelligent algorithm, *Information Sciences*, Vol. 141, No. 3-4, 259-271, 2002.
[59] Liu, Y.-K., and Liu, B., Fuzzy random variables: A scalar expected value, *Fuzzy Optimization and Decision Making*, to be published.
[60] Liu, Y.-K., and Liu, B., Expected value operator of random fuzzy variable and random fuzzy expected value models, *International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems*, to be published.
[61] Liu, Y.-K., and Liu, B., Expected value models in fuzzy random decision systems with a hybrid intelligent algorithm, Technical Report, 2000.
[62] Luhandjula, M.K., Linear programming under randomness and fuzziness, *Fuzzy Sets and Systems*, Vol. 10, 45–55, 1983.
[63] Luhandjula, M.K., On possibilistic linear programming, *Fuzzy Sets and Systems*, Vol. 18, 15–30,1986.
[64] Luhandjula, M.K., Fuzzy optimization: An appraisal, *Fuzzy Sets and Systems*, Vol. 30, 257–282, 1989.
[65] Luhandjula, M.K., Fuzziness and randomness in an optimization framework, *Fuzzy Sets and Systems*, Vol. 77, 291–297, 1996.
[66] Luhandjula M.K. and M.M. Gupta, On fuzzy stochastic optimization, *Fuzzy Sets and Systems*, Vol. 81, 47–55, 1996.
[67] Maleki, H.R., Tata, M., and Mashinchi, M., Linear programming with fuzzy variables, *Fuzzy Sets and Systems*, Vol. 109, No. 1, 21–33, 2000.
[68] Mareschal, B., Stochastic multicriteria decision making and uncertainty, *European Journal of Operational Research,* Vol. 26, No. 1, 58–64, 1986.
[69] Nahmias, S., Fuzzy variables, *Fuzzy Sets and Systems*, Vol. 1, 97–110, 1978.
[70] Ostasiewicz, W., A new approach to fuzzy programming, *Fuzzy Sets and Systems*, Vol. 7, 139–152, 1982.
[71] Pawlak, Z., Rough sets, *International Journal of Information and Computer Sciences*, Vol. 11, No. 5, 341–356, 1982.
[72] Pawlak, Z., Rough sets and fuzzy sets, *Fuzzy sets and Systems*, Vol. 17, 99–102, 1985.
[73] Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.
[74] Pawlak, Z., and Slowinski, R., Rough set approach to multiattribute decision analysis, *European Journal of Operational Research*, Vol. 72, 443–459, 1994.
[75] Puri, M.L. and Ralescu, D., Fuzzy random variables, *Journal of Mathematical Analysis and Applications*, Vol. 114, 409–422, 1986.
[76] Ramík, J., Extension principle in fuzzy optimization, *Fuzzy Sets and Systems*, Vol. 19, 29–35, 1986.
[77] Ramík, J., and Rommelfanger H., Fuzzy mathematical programming based on some inequality relations, *Fuzzy Sets and Systems*, Vol. 81, 77–88, 1996.
[78] Roubens, M., and Teghem Jr. J., Comparison of methodologies for fuzzy and stochastic multi-objective programming, *Fuzzy Sets and Systems*, Vol. 42, 119–132, 1991.
[79] Saade, J.J., Maximization of a function over a fuzzy domain, *Fuzzy Sets and Systems*, Vol. 62, 55–70, 1994.
[80] Sakawa, M., and Yano, H., Feasibility and Pareto optimality for multiobjective nonlinear programming problems with fuzzy parameters, *Fuzzy Sets and Systems*, Vol. 43, 1–15, 1991.
[81] Slowinski, R. and Teghem, Jr. J., Fuzzy versus stochastic approaches to multicriteria linear programming under uncertainty, *Naval Research Logistics*, Vol. 35, 673–695, 1988.
[82] Slowinski, R., and Vanderpooten, D., A generalized definition of rough approximations based on similarity, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 2, 331–336, 2000.
[83] Tanaka, H., Guo, P., and Zimmermann, H.-J., Possibility distribution of fuzzy decision variables obtained from possibilistic linear programming problems, *Fuzzy Sets and Systems*, Vol. 113, 323–332, 2000.
[84] Wang G. and Z. Qiao, Linear programming with fuzzy random variable coefficients, *Fuzzy Sets and Systems*, Vol. 57, 295–311, 1993.
[85] Yager, R.R., Mathematical programming with fuzzy constraints and a preference on the objective, *Kybernetes*, Vol. 9, 285–291, 1979.
[86] Yager, R.R., Generalized probabilities of fuzzy events from fuzzy belief structures, *Information Sciences*, Vol. 28, 45–62, 1982.
[87] Yager, R.R., On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 18, 183–190, 1988.
[88] Yager, R.R., On the specificity of a possibility distribution, *Fuzzy Sets and Systems*, Vol. 50, 279–292, 1992.
[89] Yager, R.R., Decision making with fuzzy probability assessments, *IEEE Transactions on Fuzzy Systems*, Vol. 7, 462–466, 1999.
[90] Yazenin, A.V., Fuzzy and stochastic programming, *Fuzzy Sets and Systems*, Vol. 22, 171–180, 1987.
[91] Yazenin, A.V., On the problem of possibilistic optimization, *Fuzzy Sets and Systems*, Vol. 81, 133–140, 1996.
[92] Zadeh, L.A., Fuzzy sets, *Information and Control*, Vol. 8, 338–353, 1965.
[93] Zadeh, L.A., Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, 28–44, 1973.

[94] Zadeh, L.A., Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, Vol. 1, 3–28, 1978.

[95] Zhao, R., and Liu, B., Stochastic programming models for general redundancy optimization problems, *IEEE Transactions on Reliability*, to be published.

[96] Zimmermann, H.-J., Fuzzy mathematical programming, *Computers and Operations Research*, Vol. 10, 291–298, 1983.

[97] Zimmermann, H.-J., Applications of fuzzy set theory to mathematical programming, *Information Sciences*, Vol. 36, 29–58, 1985.

[98] Zimmermann, H.-J., *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, Boston, 1985.