Computers & Industrial Engineering 89 (2015) 235–240

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Uncertain multilevel programming: Algorithm and applications

Baoding Liu [a], Kai Yao [b,*]

[a] Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[b] School of Management, University of Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Available online 12 October 2014

Keywords:
Uncertainty theory
Uncertain programming
Multilevel programming
Stackelberg–Nash equilibrium
Genetic algorithm

ABSTRACT

Multilevel programming is used to model a decentralized planning problem with multiple decision makers in a hierarchical system. This paper aims at providing an uncertain multilevel programming model that is a type of multilevel programming involving uncertain variables. Besides, a genetic algorithm is employed to solve the model. As an illustration, the uncertain multilevel programming model is applied to a product control problem.

1. Introduction

Multilevel programming was first proposed by Bracken and McGill (1973) to model a decentralized noncooperative decision system with one leader and multiple followers of equal status in 1973. It finds many applications in daily life such as strategic-force planning (Bracken & McGill, 1974), resource allocation (Aiyoshi & Shimizu, 1981), and water regulation (Anandalingam & Apprey, 1991). In 1990, Ben-Ayed and Blair (1990) showed that multilevel programming is an NP-hard problem. In order to solve the model numerically, many algorithms have been proposed such as extreme point algorithm (Candler & Towersley, 1982), *k*th best algorithm (Bialas & Karwan, 1984), branch and bound algorithm (Bard & Falk, 1982), descent method (Savard & Gauvin, 1994), and intelligent algorithm (Liu, 1998).

However, in many cases, the parameters in the multilevel programming are indeterminate. Multilevel programming involving random variable was first proposed by Patriksson and Wynter (1999) in 1999. In addition, Gao, Liu, and Gen (2004) proposed some new stochastic multilevel programming models in 2004. Multilevel programming involving fuzzy set was first proposed by Lai (1996) in 1996, and then developed by Shih, Lai, and Lee (1996), and Lee (2001). Especially, Gao and Liu (2005) proposed a new fuzzy multilevel programming model, and defined a Stackelberg–Nash equilibrium.

As we know, a premise of applying probability theory is that the obtained probability distribution is close enough to the true frequency. In order to get it, we should have enough samples. But due to economical or technical difficulties, we sometimes have no samples. In this case, we have to invite some domain experts to evaluate the belief degree that each event happens. However, a lot of surveys showed that human beings usually estimate a much wider range of values than the object actually takes (Liu, 2015). This conservatism of human beings makes the belief degrees deviate far from the frequency. As a result, the belief degree cannot be treated as probability distribution, otherwise some counterintuitive phenomena may happen (Liu, 2012). In order to deal with the belief degree mathematically, an uncertainty theory was founded by Liu (2007) in 2007, and refined by Liu (2010) in 2010. A concept of uncertain variable is used to model uncertain quantity, and belief degree is regarded as its uncertainty distribution. As a type of mathematical programming involving uncertain variables, uncertain programming was founded by Liu (2009) in 2009. So far, uncertain programming has been applied to many fields such as project scheduling, vehicle routing, facility location, and system design.

In this paper, we will propose a framework of uncertain multilevel programming. The rest of the paper is organized as follows. In Section 2, we review some concepts and theorems in uncertainty theory. In Section 3, we introduce the basic form of uncertain programming. The uncertain multilevel programming is proposed in Section 4, and its equivalent model is obtained and a genetic algorithm to solve the model is introduced in Section 5. In order to illustrate the efficiency of the algorithm, an example of production control is proposed in Section 6. At last, some remarks are made in Section 7.

2. Preliminary

In order to model human's belief degree, an uncertainty theory was founded by Liu (2007) in 2007 and refined by Liu (2010) in

* Corresponding author.
E-mail addresses: liu@tsinghua.edu.cn (B. Liu), yaokai@ucas.ac.cn (K. Yao).

http://dx.doi.org/10.1016/j.cie.2014.09.029
0360-8352/

2010 as a branch of axiomatic mathematics. Nowadays, it has been widely applied to mathematical programming, and has brought out a branch of uncertain programming (Liu, 2009) which is a spectrum of mathematical programming involving uncertain variables. So far, uncertain programming has been applied to shortest path problem (Gao, 2011), facility location problem (Gao, 2012; Wen, Qin, & Kang, 2014), employment contract model (Mu, Lan, & Tang, 2013), inventory problem (Qin & Kar, 2013), spanning tree (Zhang, Wang, & Zhou, 2013), and so on.

The basic concept of uncertainty theory is uncertain measure, which is used to indicate the belief degree of each event.

Definition 1 Liu, 2007. Let Γ be a nonempty set, and $\mathcal{L}$ be a σ-algebra on Γ. A set function $\mathcal{M}$ is called an uncertain measure if it satisfies the following axioms,

Axiom 1: (Normality Axiom) $\mathcal{M}\{\Gamma\} = 1$;
Axiom 2: (Duality Axiom) $\mathcal{M}\{\Lambda\} + \mathcal{M}\{\Lambda^c\} = 1$ for any $\Lambda \in \mathcal{L}$;
Axiom 3: (Subadditivity Axiom) For every sequence of $\{\Lambda_i\} \in \mathcal{L}$, we have

$$\mathcal{M}\left\{\bigcup_{i=1}^{\infty}\Lambda_i\right\} \leqslant \sum_{i=1}^{\infty}\mathcal{M}\{\Lambda_i\}.$$

In this case, the triple $(\Gamma, \mathcal{L}, \mathcal{M})$ is called an uncertainty space.

Besides, a product axiom was given by Liu (2009) for the operation of uncertain variables in 2009.

Axiom 4: (Product Axiom) Let $(\Gamma_k, \mathcal{L}_k, \mathcal{M}_k)$ be uncertainty spaces for $k = 1, 2, \ldots$ Then the product uncertain measure $\mathcal{M}$ is an uncertain measure satisfying

$$\mathcal{M}\left\{\prod_{i=1}^{\infty}\Lambda_k\right\} = \bigwedge_{k=1}^{\infty}\mathcal{M}_k\{\Lambda_k\}$$

where Λ_k are arbitrarily chosen events from $\mathcal{L}_k$ for $k = 1, 2, \ldots$, respectively.

Uncertain variable is used to represent quantities in uncertainty. Essentially, it is a measurable function on an uncertainty space.

Definition 2 Liu, 2007. Let $(\Gamma, \mathcal{L}, \mathcal{M})$ be an uncertainty space. An uncertain variable ξ is a measurable function from Γ to the set of real numbers, i.e., for any Borel set B of real numbers, the set

$$\{\xi \in B\} = \{\gamma \in \Gamma \,|\, \xi(\gamma) \in B\}$$

is an event.

Definition 3 Liu, 2009. The uncertain variables $\xi_1, \xi_2, \ldots, \xi_n$ are said to be independent if

$$\mathcal{M}\left\{\bigcap_{i=1}^{n}(\xi_i \in B_i)\right\} = \bigwedge_{i=1}^{n}\mathcal{M}\{\xi_i \in B_i\}$$

for any Borel sets $B_1, B_2, \ldots, B_n$ of real numbers.

In order to describe an uncertain variable in practice, a concept of uncertainty distribution is defined below.

Definition 4 Liu, 2007. The uncertainty distribution Φ of an uncertain variable ξ is defined by

$$\Phi(x) = \mathcal{M}\{\xi \leqslant x\}$$

for any real number x.

If an uncertainty distribution has an inverse function, then the inverse function is called an inverse uncertainty distribution. In this case, the uncertainty distribution is called regular. Inverse uncertainty distributions play an important role in the operation of uncertain variables. Let $\xi_1, \xi_2, \ldots, \xi_n$ be independent uncertain variables with uncertainty distributions $\Phi_1, \Phi_2, \ldots, \Phi_n$, respectively. Liu (2010) showed that if the function $f(x_1, x_2, \ldots, x_n)$ is strictly increasing with respect to $x_1, x_2, \ldots, x_m$ and strictly decreasing with respect to $x_{m+1}, x_{m+2}, \ldots, x_n$, then $\xi = f(\xi_1, \xi_2, \ldots, \xi_n)$ is an uncertain variable with an inverse uncertainty distribution

$$\Psi^{-1}(r) = f(\Phi_1^{-1}(r), \ldots, \Phi_m^{-1}(r), \Phi_{m+1}^{-1}(1-r), \ldots, \Phi_n^{-1}(1-r)).$$

The expected value of an uncertain variable is an average of the uncertain variable in the sense of uncertain measure.

Definition 5 Liu, 2007. The expected value of an uncertain variable ξ is defined by

$$E[\xi] = \int_0^{+\infty}\mathcal{M}\{\xi \geqslant x\}\mathrm{d}x - \int_{-\infty}^{0}\mathcal{M}\{\xi \leqslant x\}\mathrm{d}x$$

provided that at least one of the two integrals is finite.

Assuming that ξ has an uncertainty distribution Φ, Liu (2007) proved

$$E[\xi] = \int_0^{+\infty}(1-\Phi(x))\mathrm{d}x - \int_{-\infty}^{0}\Phi(x)\mathrm{d}x.$$

Furthermore, Liu and Ha (2010) proved that the uncertain variable $\xi = f(\xi_1, \xi_2, \ldots, \xi_n)$ has an expected value

$$E[\xi] = \int_0^1 f(\Phi_1^{-1}(r), \ldots, \Phi_m^{-1}(r), \Phi_{m+1}^{-1}(1-r), \ldots, \Phi_n^{-1}(1-r))\mathrm{d}r.$$

Here, the function f and the uncertain variables $\xi_1, \xi_2, \ldots, \xi_n$ are as aforementioned.

3. Uncertain programming – basic form

Assume that $\boldsymbol{x}$ is a decision vector, and ξ is an uncertain vector. Since an uncertain objective function $f(\boldsymbol{x}, \xi)$ cannot be directly maximized, we may maximize its expected value, i.e.,

$$\max_{\boldsymbol{x}} E\,[f(\boldsymbol{x}, \xi)].$$

In addition, since the uncertain constraints $g_j(\boldsymbol{x}, \xi) \leqslant 0$, $j = 1, 2, \ldots, p$ do not define a crisp feasible set, it is naturally desired that the uncertain constraints hold with confidence levels $\alpha_1, \alpha_2, \ldots, \alpha_p$. Then we have a set of chance constraints,

$$\mathcal{M}\{g_j(\boldsymbol{x}, \xi) \leqslant 0\} \geqslant \alpha_j, \quad j = 1, 2, \ldots, p.$$

In order to obtain a decision with maximum expected objective value subject to a set of chance constraints, Liu (2009) proposed the following uncertain programming model,

$$\begin{cases} \max\limits_{\boldsymbol{x}} E\,[f(\boldsymbol{x}, \xi)] \\ \text{subject to :} \\ \qquad \mathcal{M}\{g_j(\boldsymbol{x}, \xi) \leqslant 0\} \geqslant \alpha_j, \quad j = 1, 2, \ldots, p. \end{cases} \tag{1}$$

Definition 6. A vector $\boldsymbol{x}$ is called a feasible solution to the uncertain programming model (1) if

$$\mathcal{M}\{g_j(\boldsymbol{x}, \xi) \leqslant 0\} \geqslant \alpha_j$$

for $j = 1, 2, \ldots, p$.

Definition 7. A feasible solution $\boldsymbol{x}^*$ is called an optimal solution to the uncertain programming model (1) if

$$E\,[f(\boldsymbol{x},\xi)] \leqslant E\,[f(\boldsymbol{x}^*,\xi)]$$

for any feasible solution $\boldsymbol{x}$.

Assume that $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ where $\xi_1, \xi_2, \ldots, \xi_n$ are independent uncertain variables with uncertainty distributions $\Phi_1, \Phi_2, \ldots, \Phi_n$, respectively. Without loss of generality, we also assume that f is monotone increasing with respect to $\xi_1, \xi_2, \ldots, \xi_k$, and strictly decreasing with respect to $\xi_{k+1}, \xi_{k+2}, \ldots, \xi_n$, and g_j is strictly increasing with respect to $\xi_1, \xi_2, \ldots, \xi_{k_j}$, and strictly decreasing with respect to $\xi_{k_j+1}, \xi_{k_j+2}, \ldots, \xi_n$ for $j = 1, 2, \ldots, p$. Then the uncertain programming model (1) is equivalent to a crisp model as follows,

$$\begin{cases} \max\limits_{\boldsymbol{x}} \int_0^1 f\left(\boldsymbol{x}, \Phi_1^{-1}(r), \ldots, \Phi_k^{-1}(r), \Phi_{k+1}^{-1}(1-r), \ldots, \Phi_n^{-1}(1-r)\right)\mathrm{d}r \\ \text{subject to}: \\ \quad g_j\left(\boldsymbol{x}, \Phi_1^{-1}(\alpha_j), \ldots, \Phi_{k_j}^{-1}(\alpha_j), \Phi_{k_j+1}^{-1}(1-\alpha_j), \ldots, \Phi_n^{-1}(1-\alpha_j)\right) \leqslant 0, \\ \quad j = 1, 2, \ldots, p. \end{cases}$$

4. Uncertain multilevel programming

Now, consider a decentralized two-level decision system with one leader and m followers as shown in Fig. 1. Let $\boldsymbol{x}$ be the control vector of the leader, and $\boldsymbol{y}_i$ be that of the ith followers, $i = 1, 2, \ldots, m$, respectively. Assume that the objective function of the leader is $F(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi)$, where ξ is an uncertain vector. Since the objective function is an uncertain variable, it cannot be directly maximized. Instead, we maximize its expected value, i.e.,

$$\max_{\boldsymbol{x}} E[F(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi)].$$

Assume that the objective functions of the ith followers are $f_i(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi), i = 1, 2, \ldots, m,$ respectively. Similarly, we have

$$\max_{\boldsymbol{x}} E\,[f_i(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi)], \quad i = 1, 2, \ldots, m.$$

Assume that the constraint of the leader is

$$G(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leq 0 \tag{2}$$

where G is a vector-valued function and 0 is a zero vector. Since $G(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi)$ is an uncertain variable, the inequality (2) generally does not hold identically. Instead, we hope the inequality (2) holds with a given confidence level α. Then the feasible set of the leader's control vector $\boldsymbol{x}$ is defined by the chance constraint

$$\mathcal{M}\{G(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leqslant 0\} \geqslant \alpha.$$

For each decision $\boldsymbol{x}$ chosen by the leader, the feasibility of control vectors $\boldsymbol{y}_i$ of the ith followers should be dependent on not only $\boldsymbol{x}$ but also $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i+1}, \ldots, \boldsymbol{y}_m$. Assume that the constraints of the ith followers are $g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leqslant 0$, where g_i are vector-valued functions, $i = 1, 2, \ldots, m$, respectively. Similarly, it is represented by the chance constraints

$$\mathcal{M}\{g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leqslant 0\} \geqslant \alpha_i$$

where α_i are given confidence levels for $i = 1, 2, \ldots, m$.

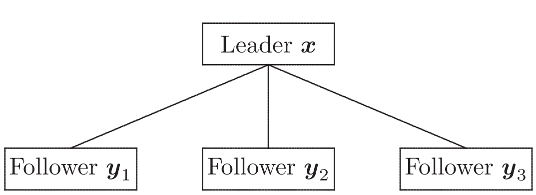

Fig. 1. A decentralized decision system.

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

$$\begin{cases} \max\limits_{\boldsymbol{x}} E[F(\boldsymbol{x}, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*, \xi)] \\ \text{subject to}: \\ \quad \mathcal{M}\{G(\boldsymbol{x}, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*, \xi) \leqslant 0\} \geqslant \alpha \\ \quad (\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*) \text{ solves problems } (i = 1, 2, \ldots, m) \\ \quad \begin{cases} \max\limits_{\boldsymbol{y}_i} E[f_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi)] \\ \text{subject to}: \\ \quad \mathcal{M}\{g_i(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leqslant 0\} \geqslant \alpha_i. \end{cases} \end{cases} \tag{3}$$

Definition 8. Let $\boldsymbol{x}$ be a feasible control vector of the leader. A Nash equilibrium of followers is the feasible array $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*)$ with respect to $\boldsymbol{x}$ if

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

for any feasible array $(\boldsymbol{y}_1^*, \ldots, \boldsymbol{y}_{i-1}^*, \boldsymbol{y}_i, \boldsymbol{y}_{i+1}^*, \ldots, \boldsymbol{y}_m^*)$ and $i = 1, 2, \ldots, m$.

Definition 9. Suppose that $\boldsymbol{x}^*$ is a feasible control vector of the leader and $(\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*)$ is a Nash equilibrium of followers with respect to $\boldsymbol{x}^*$. We call the array $(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*)$ a Stackelberg–Nash equilibrium to the uncertain multilevel programming (3) if

$$E[F(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2, \ldots, \overline{\boldsymbol{y}}_m, \xi)] \leqslant E[F(\boldsymbol{x}^*, \boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_m^*, \xi)]$$

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

5. Equivalent crisp model

From the mathematical viewpoint, there is no difference between deterministic mathematical programming and uncertain programming except for the fact that there exist uncertain variables in the latter. In fact, the uncertain multilevel programming model (3) is equivalent to a deterministic multilevel programming model.

Let $\xi_1, \xi_2, \ldots, \xi_n$ be independent uncertain variables with uncertainty distributions $\Phi_1, \Phi_2, \ldots, \Phi_n$, respectively. Without loss of generality, we assume that F is a real function, and strictly increasing with respect to $\xi_1, \xi_2, \ldots, \xi_k$, and strictly decreasing with respect to $\xi_{k+1}, \xi_{k+2}, \ldots, \xi_n$. Then we have

$$\begin{aligned} E\,[F(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi_1, \ldots, \xi_n)] = \int_0^1 F\Big(&\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \Phi_1^{-1}(r), \ldots, \\ &\Phi_k^{-1}(r), \Phi_{k+1}^{-1}(1-r), \ldots, \Phi_n^{-1}(1-r)\Big)\mathrm{d}r. \end{aligned}$$

Assume f_i is a real function, and strictly increasing with respect to $\xi_1, \xi_2, \ldots, \xi_{k_i}$, and strictly decreasing with respect to $\xi_{k_i+1}, \xi_{k_i+2}, \ldots, \xi_n$ for $i = 1, 2, \ldots, m$. Then

$$\begin{aligned} E[f_i(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \xi_1, \ldots, \xi_n)] = \int_0^1 f_i\Big(&\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \Phi_1^{-1}(r), \ldots, \\ &\Phi_{k_i}^{-1}(r), \Phi_{k_i+1}^{-1}(1-r), \ldots, \Phi_n^{-1}(1-r)\Big)\mathrm{d}r. \end{aligned}$$

Assume G is a real function, and strictly increasing with respect to $\xi_1, \xi_2, \ldots, \xi_s$, and strictly decreasing with respect to $\xi_{s+1}, \xi_{s+2}, \ldots, \xi_n$. Then $\mathcal{M}\{G(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m, \xi) \leqslant 0\} \geqslant \alpha$ is equivalent to

$$G\left(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_m,\Phi_1^{-1}(\alpha),\ldots,\Phi_s^{-1}(\alpha),\Phi_{s+1}^{-1}(1-\alpha),\ldots,\Phi_n^{-1}(1-\alpha)\right)\leqslant 0.$$

Assume g_i is a real function, and strictly increasing with respect to $\xi_1,\ \xi_2,\ \ldots,\ \xi_{s_i}$, and strictly decreasing with respect to $\xi_{s_i+1},\ \xi_{s_i+2},\ \ldots,\ \xi_n$ for $i=1,\ 2,\ \ldots,\ m$. Then $\mathcal{M}\{g_i(\boldsymbol{x},\ \boldsymbol{y}_1,\ \ldots,\boldsymbol{y}_m,\ \xi)\leqslant 0\}\geqslant \alpha_i$ is equivalent to

$$g_i\left(\boldsymbol{x},\boldsymbol{y}_1,\ \ldots,\boldsymbol{y}_m,\Phi_1^{-1}(\alpha_i),\ldots,\Phi_{s_i}^{-1}(\alpha_i),\Phi_{s_i+1}^{-1}(1-\alpha_i),\ldots,\Phi_n^{-1}(1-\alpha_i)\right)\leqslant 0.$$

Thus the uncertain multilevel programming model (3) is equivalent to

$$\begin{cases}\max\limits_{\boldsymbol{x}} \int_0^1 F\left(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m,\Phi_1^{-1}(r),\ldots,\Phi_k^{-1}(r),\Phi_{k+1}^{-1}(1-r),\ldots,\Phi_n^{-1}(1-r)\right)\mathrm{d}r\\ \text{subject to}:\\ \quad G\left(\boldsymbol{x},\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\ldots,\boldsymbol{y}_m^*,\Phi_1^{-1}(\alpha),\ldots,\Phi_s^{-1}(\alpha),\Phi_{s+1}^{-1}(1-\alpha),\ldots,\Phi_n^{-1}(1-\alpha)\right)\leqslant 0\\ \quad (\boldsymbol{y}_1^*,\boldsymbol{y}_2^*,\ldots,\boldsymbol{y}_m^*)\ \text{solves problems}\ (i=1,2,\ldots,m)\\ \quad\begin{cases}\max\limits_{\boldsymbol{y}_i} \int_0^1 f_i\left(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m,\Phi_1^{-1}(r),\ldots,\Phi_{k_i}^{-1}(r),\Phi_{k_i+1}^{-1}(1-r),\ldots,\Phi_n^{-1}(1-r)\right)\mathrm{d}r\\ \text{subject to}:\\ \quad g_i\left(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m,\Phi_1^{-1}(\alpha_i),\ldots,\Phi_{s_i}^{-1}(\alpha_i),\Phi_{s_i+1}^{-1}(1-\alpha_i),\ldots,\Phi_n^{-1}(1-\alpha_i)\right)\leqslant 0.\end{cases}\end{cases} \tag{4}$$

In order to solve uncertain programming models (3), we just need find a numerical method for solving the deterministic mathematical programming (4). So far, many algorithms have been proposed such as extreme point algorithm (Candler & Towersley, 1982), kth best algorithm (Bialas & Karwan, 1984), branch and bound algorithm (Bard & Falk, 1982), descent method (Savard & Gauvin, 1994), and genetic algorithm (Liu, 1998).

Here, we introduce the genetic algorithm to solve multilevel programming by Liu (1998):

Step 0: Input parameters such as population size, crossover probability and mutation probability.
Step 1: Initialize chromosomes randomly in the feasible set.
Step 2: Update the chromosomes by the crossover and mutation operations.
Step 3: For each chromosome, determine the Nash equilibrium of the followers via genetic algorithm.
Step 4: Calculate the objective values of the leader for each chromosomes with respect to the Nash equilibrium.
Step 5: Compute the fitness of each chromosome based on the objective values.
Step 6: Select the chromosomes by spinning the roulette wheel.
Step 7: Repeat the second to sixth steps for a given number of cycles.
Step 8: Report the best chromosome as the optimal solution.

In order to illustrate the effectiveness of genetic algorithm in solving uncertain multilevel programming model, we give two numerical examples.

Example 1. Assume there is one leader and one follower in the uncertain multilevel programming model whose control vectors are $\boldsymbol{x}=(x_1,x_2)$ and $\boldsymbol{y}=(y_1,y_2)$, respectively. Suppose that the uncertain multilevel programming model is formulated as follows,

$$\begin{cases}\max E[\xi_1 x_1 \sin y_1^* + \xi_2 y_2^* \sin x_2]\\ \text{subject to}:\\ \quad x_1+x_2\leqslant \pi, x_1\geqslant 0, x_2\geqslant 0\\ \quad (y_1^*,y_2^*)\ \text{solves the problem}\\ \quad\begin{cases}\max E[\xi_1 x_1\cos y_1 - \xi_2 y_2 \cos x_2]\\ \text{subject to}:\\ \quad 0\leqslant y_1\leqslant x_1, 0\leqslant y_2\leqslant x_2\end{cases}\end{cases} \tag{5}$$

where $\xi_1\sim\mathcal{N}(1,3)$ and $\xi_2\sim\mathcal{N}(2,4)$ are normal uncertain variables. Set the population size as 30, the probability of crossover as 0.2, and the probability of mutation as 0.1. A run of genetic algorithm with 100 generations shows that the Stackelberg–Nash equilibrium is

$$\boldsymbol{x}=(0.9339, 2.0304),\quad \boldsymbol{y}=(0, 2.0304),$$

and the objective values of the leader and the follower are 3.6394 and 2.7354, respectively. Note that the Stackelberg–Nash equilibrium is not unique. For example, another Stackelberg–Nash equilibrium is $\boldsymbol{x}=(0.7113, 2.0342)$, $\boldsymbol{y}=(0, 2.0342)$ with the objective values 3.6393 and 2.5300 for the leader and the follower, respectively.

In addition, in order to illustrate the robustness of the genetic algorithm for uncertain programming model, a further study is carried out. When the population size (*pop_size*), probability of crossover (P_c) and probability of mutation (P_m) vary, the obtained optimal solution via genetic algorithm also varies, and the values are shown in Table 1. The percent error, i.e.,

$$\frac{\text{maximum objective value} - \text{minimum objective value}}{(\text{maximum objective value}+\text{minimum objective value})/2}\times 100\%$$

is just only

$$\frac{3.6394-3.6393}{(3.6394+3.6393)/2}\times 100\% = 0.0027\%,$$

and that means the genetic algorithm is robust to the parameters, and can solve this uncertain multilevel programming model effectively.

Example 2. Assume there is one leader and two followers in the uncertain multilevel programming model whose control vectors are $\boldsymbol{x}=(x_1,x_2)$, $\boldsymbol{y}_1=(y_{11},\ y_{12})$ and $\boldsymbol{y}_2=(y_{21},\ y_{22})$, respectively. Suppose that the uncertain multilevel programming model is formulated as follows,

$$\begin{cases}\max E[x_1(y_{11}^*+y_{12}^*)/\xi_1 + x_2(y_{21}^*+y_{22}^*)/\xi_2]\\ \text{subject to}:\\ \quad \mathcal{M}\{x_1^2+x_2^2\leqslant \xi_1^2+\xi_2^2\}\geqslant 0.9\\ \quad x_1\geqslant 0, x_2\geqslant 0\\ \quad (y_{11}^*,y_{12}^*,y_{21}^*,y_{22}^*)\ \text{solves the problems}\\ \quad\begin{cases}\max E[\xi_1 y_{11}+\xi_2 y_{12}]\\ \text{subject to}:\\ \quad \mathcal{M}\{2y_{11}+y_{12}\leqslant \xi_1+x_1\}\geq 0.9\\ \quad y_{11}\geq 0, y_{12}\geq 0\end{cases}\\ \quad\begin{cases}\max E[\xi_1 y_{21}+\xi_2 y_{22}]\\ \text{subject to}:\\ \quad \mathcal{M}\{y_{21}+2y_{22}\leqslant \xi_2+x_2\}\geq 0.9\\ \quad y_{21}\geq 0, y_{22}\geq 0\end{cases}\end{cases} \tag{6}$$

where $\xi_1\sim\mathcal{L}(1,3)$ and $\xi_2\sim\mathcal{L}(2,4)$ are linear uncertain variables. Set the population size as 30, the probability of crossover as 0.2, and the probability of mutation as 0.1. A run of genetic algorithm with 100 generations shows that the Stackelberg–Nash equilibrium is

$$\boldsymbol{x}=(4.3643, 1.7168),\quad \boldsymbol{y}_1=(0, 7.1643),\quad \boldsymbol{y}_2=(5.5168, 0)$$

and the optimal objective values of the leader and the two followers are 20.4579, 21.4929, and 11.0337, respectively.

Table 2 shows the different objective values when the parameters in the genetic algorithm vary. The percent error is

$$\frac{20.6248-20.4119}{(20.6248+20.4119)/2}\times 100\% = 1.04\%,$$

and it also illustrates that the genetic algorithm is robust, and plays an effective role in solving uncertain multilevel programming model.

Table 1
Comparison of solutions in Example 1.

No.	pop_size	P_c	P_m	Stackelberg–Nash equilibrium		Objective value	
				Leader	Follower	Leader	Follower
1	30	0.2	0.1	(0.9339, 2.0304)	(0, 2.0304)	3.6394	2.7354
2	30	0.2	0.2	(0.4962, 2.0287)	(0, 2.0287)	3.6394	2.2896
3	30	0.1	0.2	(0.7150, 2.0230)	(0, 2.0230)	3.6393	2.4830
4	50	0.1	0.2	(1.0138, 2.0287)	(0, 2.0287)	3.6394	2.8074
5	50	0.2	0.1	(0.7889, 2.0357)	(0, 2.0357)	3.6393	2.6143

Table 2
Comparison of solutions in Example 2.

No.	pop_size	P_c	P_m	Stackelberg–Nash equilibrium			Objective value		
				Leader	Follower 1	Follower 2	Leader	Follower 1	Follower 2
1	30	0.2	0.1	(4.3643, 1.7168)	(0, 7.1643)	(5.5168, 0)	20.4579	21.4929	11.0337
2	30	0.2	0.2	(4.1299, 2.2653)	(0, 6.9299)	(6.0653, 0)	20.4828	20.7897	12.1306
3	30	0.1	0.2	(4.2945, 1.9511)	(0, 7.0945)	(5.7511, 0)	20.6248	21.2835	11.5022
4	50	0.1	0.2	(4.2254, 2.0876)	(0, 7.0254)	(5.8876, 0)	20.5656	21.0761	11.7751
5	50	0.2	0.1	(4.3420, 1.7544)	(0, 7.1420)	(5.5544, 0)	20.4119	21.4261	11.1089

6. Applications

In this section, we apply the uncertain multilevel programming to a product control problem. Consider an enterprise with a center and two factories. Assume the center supplies two types of resources to the factories and sell the products to the markets, and the factories produce two types of products with the resources. The center makes a decision on the allocation of the resources to maximize the profit in the markets, and each factory desires to guarantee the efficiency and quality. The notations are introduced as follows,

x_{ij}	amount of resource i allocated to factory j, $i,j=1,2$
y_{ij}	amount of product i produced by factory j, $i,j=1,2$
Y_i	total amount of product i, i.e., $Y_i=y_{i1}+y_{i2},\ i=1,2$
ξ_1,ξ_2	unit-prices of product 1 and product 2 which are uncertain variables, $\xi_1\sim\mathcal{L}(195,205), \xi_2\sim\mathcal{L}(145,175)$
$F(Y_1,Y_2)$	total profit of marketing $F(Y_1,Y_2)=\xi_1Y_1+\xi_2Y_2$
$f_j(y_{1j},y_{2j})$	objective function of factory j, $f_1(y_{11},y_{21})=(y_{11}-4.0)^2+(y_{21}-13.0)^2$, $f_2(y_{12},y_{22})=(y_{12}-35.0)^2+(y_{22}-2.0)^2$.

Then an uncertain multilevel programming model of product control is given as follows,

$$\begin{cases}\max\limits_{x_{11},x_{12},x_{21},x_{22}} E[\xi_1(y_{11}^*+y_{12}^*)+\xi_2(y_{21}^*+y_{22}^*)]\\ \text{subject to:}\\ \qquad x_{11}+x_{12}+x_{21}+x_{22}\leqslant 40\\ \qquad 0\leqslant x_{11}\leqslant 10,\quad 0\leqslant x_{12}\leqslant 15\\ \qquad 0\leqslant x_{21}\leqslant 5,\quad 0\leqslant x_{22}\leqslant 20\\ \qquad (y_{11}^*,y_{21}^*,y_{12}^*,y_{22}^*)\text{ solves the problems}\\ \qquad \begin{cases}\min\limits_{y_{11},y_{21}}(y_{11}-4.0)^2+(y_{21}-13.0)^2\\ \text{subject to:}\\ \qquad 4y_{11}+7y_{21}\leqslant 10x_{11},\quad 6y_{11}+3y_{21}\leqslant 10x_{21}\\ \qquad 0\leqslant y_{11},y_{21}\leq 20\end{cases}\\ \qquad \begin{cases}\min\limits_{y_{12},y_{22}}(y_{12}-35.0)^2+(y_{22}-2.0)^2\\ \text{subject to:}\\ \qquad 4y_{12}+5y_{22}\leqslant 10x_{12},\quad 6y_{12}+7y_{22}\leqslant 10x_{22}\\ \qquad 0\leqslant y_{12},y_{22}\leq 40.\end{cases}\end{cases} \tag{7}$$

The model (7) has a Stackelberg–Nash equilibrium $\boldsymbol{x}=(7.9356, 11.6823, 4.1880, 16.1941)$, $\boldsymbol{y}_1=(1.8363, 10.2873)$, $\boldsymbol{y}_2=(26.9902, 0)$, and an objective value 6473 by the genetic algorithm. So the center supplies 7.9356 amounts of source 1 and 11.6823 amounts of source 2 to factory 1, and supplies 4.1880 amounts of source 1 and 16.1941 amounts of source 2 to factory 2. In addition, the optimal amounts of the product 1 and product 2 for factory 1 are 1.8363 and 10.2873, and the optimal amounts of the product 1 and product 2 for factory 2 are 26.9902 and 0, respectively. In this case, the supply center has a profit 6473.

7. Conclusions

This paper proposed an uncertain multilevel programming model that is a type of multilevel programming involving uncertain variables. It was transformed into a crisp multilevel model, and genetic algorithm was employed to solve it. The efficiency of the algorithm was illustrated by some numerical examples. Finally, the uncertain multilevel programming was applied to a product control problem. Further researches may cover modified intelligent algorithms for uncertain multilevel programming model, and applications of the model in various areas.

Acknowledgement

This work was supported by National Natural Science Foundation of China Grants No. 61273044 and No. 61403360.

References

Aiyoshi, E., & Shimizu, K. (1981). Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Transactions on Systems, Man, and Cybernetics, 11*(6), 444–449.
Anandalingam, G., & Apprey, V. (1991). Multi-level programming and conflict resolution. *European Journal of Operational Research, 51*, 233–247.
Bard, J., & Falk, J. (1982). An explicit solution to the multi-level programming problem. *Computers and Operations Research, 9*, 77–100.
Ben-Ayed, O., & Blair, C. E. (1990). Computational difficulties of bilevel linear programming. *Operations Research, 38*, 556–560.
Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming. *Management Science, 30*, 1004–1020.
Bracken, J., & McGill, J. M. (1973). Mathematical programs with optimization problems in the constraints. *Operational Research, 21*, 37–44.
Bracken, J., & McGill, J. M. (1974). Defense applications of mathematical programs with optimization problem in the constraints. *Operations Research, 22*, 1086–1096.
Candler, W., & Towersley, R. (1982). A linear two-level programming problem. *Computers and Operations Research, 9*, 59–76.

Gao, Y. (2011). Shortest path problem with uncertain arc lengths. *Computers and Mathematics with Applications, 62*(6), 2591–2600.

Gao, Y. (2012). Uncertain models for single facility location problems on networks. *Applied Mathematical Modelling, 36*(6), 2592–2599.

Gao, J., & Liu, B. (2005). Fuzzy multilevel programming with a hybrid intelligent algorithm. *Computers & Mathematics with Applications, 49*(9-10), 1539–1548.

Gao, J., Liu, B., & Gen, M. (2004). A hybrid intelligent algorithm for stochastic multilevel programming, IEEJ Transaction on Electronics. *Information and Systems, 124*(10), 1991–1998.

Lai, Y. J. (1996). Hierachical optimization: A satisfactory solution. *Fuzzy Sets and Systems, 77*, 321–335.

Lee, E. S. (2001). Fuzzy multiple level programming. *Applied Mathematics and Computation, 120*, 79–90.

Liu, B. (1998). Stackelberg–Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computers & Mathematics with Applications, 36*(7), 79–89.

Liu, B. (2007). *Uncertainty theory* (2nd ed.). Berlin: Springer Verlag.

Liu, B. (2009). Some research problems in uncertainty theory. *Journal of Uncertain Systems, 3*(1), 3–10.

Liu, B. (2009). *Theory and practice of uncertain programming* (2nd ed.). Berlin: Springer Verlag.

Liu, B. (2010). *Uncertainty theory: A branch of mathematics for modeling human uncertainty*. Berlin: Springer-Verlag.

Liu, B. (2012). Why is there a need for uncertainty theory? *Journal of Uncertain Systems, 6*(1), 3–10.

Liu, B. (2015). *Uncertainty theory* (4th ed.). Berlin: Springer-Verlag.

Liu, Y. H., & Ha, M. H. (2010). Expected value of function of uncertain variables. *Journal of Uncertain Systems, 4*(3), 181–186.

Mu, R., Lan, Y., & Tang, W. (2013). An uncertain contract model for rural migrant worker's employment problems. *Fuzzy Optimization and Decision Making, 12*(1), 29–39.

Patriksson, M., & Wynter, L. (1999). Stochastic mathematical programs with equilibrium constraints. *Operations Research Letters, 25*, 159–167.

Qin, Z., & Kar, S. (2013). Single-period inventory problem under uncertain environment. *Applied Mathematics and Computation, 219*(18), 9630–9638.

Savard, G., & Gauvin, J. (1994). The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters, 15*, 275–282.

Shih, H. S., Lai, Y. J., & Lee, E. S. (1996). Fuzzy approach for multi-level programming problems. *Computer & Operations Research, 23*, 73–91.

Wen, M., Qin, Z., & Kang, R. (2014). The α-cost minimization model for capacitated facility location–allocation problem with uncertain demands. *Fuzzy Optimization and Decision Making, 13*(3), 345–356. doi: http://dx.doi.org/10.1007/s10700-014-9179-z.

Zhang, X., Wang, Q., & Zhou, J. (2013). Two uncertain programming models for inverse minimum spanning tree problem. *Industrial Engineering & Management Systems, 12*(1), 9–15.