Expert Systems with Applications 39 (2012) 10898–10903

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Stackelberg solutions for random fuzzy two-level linear programming through possibility-based probability model

Masatoshi Sakawa *, Takeshi Matsui

Faculty of Engineering, Hiroshima University, Higashi-Hiroshima 739-8527, Japan

ARTICLE INFO

Keywords:
Two-level linear programming
Random fuzzy variable
Stackelberg solutions
Possibility
Probability maximization

ABSTRACT

This paper considers computational methods for obtaining Stackelberg solutions to random fuzzy two-level linear programming problems. Assuming that the decision makers concerns about the probabilities that their own objective function values are smaller than or equal to certain target values, fuzzy goals of the decision makers for the probabilities are introduced. Using the possibility-based probability model to maximize the degrees of possibility with respect to the attained probability, the original random fuzzy two-level programming problems are reduced to deterministic ones. Extended concepts of Stackelberg solutions are introduced and computational methods are also presented. A numerical example is provided to illustrate the proposed method.

1. Introduction

In the real world, we often encounter situations where there are two or more decision makers in an organization with a hierarchical structure, and they make decisions in turn or at the same time so as to optimize their objective functions. Decision making problems in decentralized organizations are often modeled as Stackelberg games (Simaan and Cruz, 1973), and they are formulated as two-level mathematical programming problems (Shimizu et al., 1997; Sakawa and Nishizaki, 2009). In the context of two-level programming, the decision maker at the upper level first specifies a strategy, and then the decision maker at the lower level specifies a strategy so as to optimize the objective with full knowledge of the action of the decision maker at the upper level. In conventional multi-level mathematical programming models employing the solution concept of Stackelberg equilibrium, it is assumed that there is no communication among decision makers, or they do not make any binding agreement even if there exists such communication.

Computational methods for obtaining Stackelberg solutions to two-level linear programming problems are classified roughly into three categories: the vertex enumeration approach (Bialas and Karwan, 1984), the Kuhn–Tucker approach (Bard and Falk, 1982; Bard and Moore, 1990; Bialas and Karwan, 1984; Hansen et al., 1992), and the penalty function approach (White and Anandalingam, 1993). The subsequent works on two-level programming problems under noncooperative behavior of the decision makers have been appearing (Colson et al., 2005; Faisca et al., 2007; Gümüs and Floudas, 2001; Nishizaki and Sakawa, 2000; Nishizaki et al., 2003) including some applications to aluminium production process (Nicholls, 1996), pollution control policy determination (Amouzegar and Moshirvaziri, 1999), tax credits determination for biofuel producers (Dempe and Bard, 2001), pricing in competitive electricity markets (Fampa et al., 2008), supply chain planning (Roghanian et al., 2007) and so forth.

In order to deal with multiobjective problems (Sakawa, 1993, 2001) in hierarchical decision making, two-level multiobjective linear programming problems were formulated and a computational method for obtaining the corresponding Stackelberg solution was also developed (Nishizaki and Sakawa, 1999). Considering stochastic events related to hierarchical decision making situations, on the basis of stochastic programming models, two-level programming problems with random variables were formulated and algorithms for deriving the Stackelberg solutions were developed (Nishizaki et al., 2003). Furthermore, considering not only the randomness of parameters involved in objective functions and/or constraints but also the experts' ambiguous understanding of realized values of the random parameters, fuzzy random two-level linear programming problems were formulated, and computational methods for obtaining the corresponding Stackelberg solutions were also developed (Sakawa and Katagiri, 2012; Sakawa and Kato, 2009; Sakawa et al., in press; Sakawa et al., 2011).

From a viewpoint of ambiguity and randomness different from fuzzy random variables (Kwakernaak, 1978; Puri and Ralescu, 1986; Wang and Qiao, 1993), by considering the experts' ambiguous understanding of means and variances of random variables, a concept of random fuzzy variables was proposed, and

* Corresponding author.
E-mail address: sakawa@hiroshima-u.ac.jp (M. Sakawa).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.001

mathematical programming problems with random fuzzy variables were formulated together with the development of a simulation-based approximate solution method (Liu, 2002).

Under these circumstances, in this paper, assuming noncooperative behavior of the decision makers, we formulate random fuzzy two-level linear programming problems. To deal with the formulated two-level linear programming problems involving random fuzzy variables, we assume that the decision makers concerns about the probabilities that their own objective function values are smaller than or equal to certain target values. By considering the imprecise nature of the human judgments, we introduce the fuzzy goals of the decision makers for the probabilities. Then, assuming that the decision makers are willing to maximize the degrees of possibility with respect to the attained probability, we consider the possibility-based probability model for random fuzzy two-level programming problems. Extended concepts of Stackelberg solutions are introduced. Computational methods for obtaining approximate Stackelberg solutions through particle swarm optimization are also presented. An illustrative numerical example demonstrates the feasibility and efficiency of the proposed method.

2. Random fuzzy variables

In the framework of stochastic programming, it is implicitly assumed that the uncertain parameter which well represents the stochastic factor of real systems can be definitely expressed as a single random variable. However, from the expert's experimental point of view, the experts may think of a collection of random variables to be appropriate to express stochastic factors rather than only a single random variables. In this case, reflecting the expert's conviction degree that each of random variables properly represents the stochastic factor, it would be quite reasonable to assign the different degrees of possibility to each of random variables. For handling such an uncertain parameter, a random fuzzy variable was defined by Liu (2002) as a function from a possibility space to a collection of random variables, which is considered to be an extended concept of fuzzy variable (Nahmias, 1978). It should be noted here that the fuzzy variables can be viewed as another way of dealing with the imprecision which was originally represented by fuzzy sets. Although we can employ Liu's definition, for consistently discussing various concepts in relation to the fuzzy sets, we define the random fuzzy variables by extending not the fuzzy variables but the fuzzy sets.

Definition 1 (*Random fuzzy variable*). Let Γ be a collection of random variables. Then, a random fuzzy variable $\bar{\tilde{C}}$ is defined by its membership function

$$\mu_{\bar{\tilde{C}}} : \Gamma \to [0, 1]. \tag{1}$$

In Definition 1, the membership function $\mu_{\bar{\tilde{C}}}$ assigns each random variable $\bar{\gamma} \in \Gamma$ to a real number $\mu_{\bar{\tilde{C}}}(\bar{\gamma})$. It should be noted here that if Γ is defined as $\mathbb{R}$, then (1) becomes equivalent to the membership function of an ordinary fuzzy set. In this sense, a random fuzzy variable can be regarded as an extended concept of fuzzy sets. On the other hand, if Γ is defined as a singleton $\Gamma = \{\bar{\gamma}\}$ and $\mu_{\bar{\tilde{C}}}(\bar{\gamma}) = 1$, then the corresponding random fuzzy variable $\bar{\tilde{C}}$ can be viewed as an ordinary random variable.

When taking account of the imprecise nature of the realized values of random variables, it would be appropriate to employ the concept of fuzzy random variables. However, it should be emphasized here that if mean and/or variance of random variables are specified by the expert as a set of real values or fuzzy sets, such uncertain parameters can be represented by not fuzzy random variables but random fuzzy variables.

As a simple example of random fuzzy variables, we consider a Gaussian random variable whose mean value is not definitely specified as a constant. For example, when some random parameter $\bar{\gamma}$ is represented by the Gaussian random variable $N(s_i, 10^2)$ where the expert identifies a set $\{s_1, s_2, s_3\}$ of possible mean values as $(s_1, s_2, s_3) = (90, 100, 110)$, if the membership function $\mu_{\bar{\tilde{C}}}$ is defined by

$$\mu_{\bar{\tilde{C}}}(\bar{\gamma}) = \begin{cases} 0.5 & \text{if } \bar{\gamma} \sim N(90, 10^2), \\ 0.7 & \text{if } \bar{\gamma} \sim N(100, 10^2), \\ 0.3 & \text{if } \bar{\gamma} \sim N(110, 10^2), \\ 0 & \text{otherwise}, \end{cases}$$

then $\bar{\tilde{C}}$ is a random fuzzy variable. More generally, when the mean values are expressed as fuzzy sets or fuzzy numbers, the corresponding random variable with the fuzzy mean is represented by a random fuzzy variable.

3. Problem formulation and transformed problems

Random fuzzy two-level linear programming problems are generally formulated as

$$\left.\begin{array}{l} \underset{\boldsymbol{x}_1}{\text{minimize}}\, z_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = \bar{\tilde{\boldsymbol{C}}}_{11}\boldsymbol{x}_1 + \bar{\tilde{\boldsymbol{C}}}_{12}\boldsymbol{x}_2 \\ \text{where } \boldsymbol{x}_2 \text{ solves} \\ \underset{\boldsymbol{x}_2}{\text{minimize}}\, z_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = \bar{\tilde{\boldsymbol{C}}}_{21}\boldsymbol{x}_1 + \bar{\tilde{\boldsymbol{C}}}_{22}\boldsymbol{x}_2 \\ \text{subject to} \quad A_1\boldsymbol{x}_1 + A_2\boldsymbol{x}_2 \leqslant \boldsymbol{b} \\ \qquad\qquad \boldsymbol{x}_1 \geqslant \boldsymbol{0},\ \boldsymbol{x}_2 \geqslant \boldsymbol{0}, \end{array}\right\} \tag{2}$$

where $\boldsymbol{x}_1$ is an n_1 dimensional decision variable column vector for the decision maker at the upper level (DM1), $\boldsymbol{x}_2$ is an n_2 dimensional decision variable column vector for the decision maker at the lower level (DM2), A_j, $j = 1, 2$ are $m \times n_j$ coefficient matrices, and $\boldsymbol{b}$ is an m dimensional column vector, and $z_l(\boldsymbol{x}_1, \boldsymbol{x}_2)$, $l = 1, 2$ are the objective functions for DMl, $l = 1, 2$, respectively.

Observing that the real data with uncertainty are often distributed normally, from the practical point of view, we assume that each of $\bar{\tilde{C}}_{ljk}$, $k = 1, 2, \ldots, n_j$ of $\bar{\tilde{\boldsymbol{C}}}_{lj}$, $l = 1, 2$, $j = 1, 2$ is the Gaussian random variable with fuzzy mean value $\tilde{M}_{ljk}$ which is represented by an L–R fuzzy number characterized by the membership function

$$\mu_{\tilde{M}_{ljk}}(\tau) = \begin{cases} L\left(\frac{m_{ljk} - \tau}{\alpha_{ljk}}\right) & \text{if } m_{ljk} \geqslant \tau \\ R\left(\frac{\tau - m_{ljk}}{\beta_{ljk}}\right) & \text{if } m_{ljk} < \tau, \end{cases} \tag{3}$$

where the shape functions L and R are nonincreasing continuous functions from $[0, \infty)$ to $[0, 1]$, m_{ljk} is the mean value, and α_{ljk} and β_{ljk} are positive numbers which represent left and right spreads. Fig. 1 illustrates an example of the membership function $\mu_{\tilde{M}_{ljk}}(\tau)$.

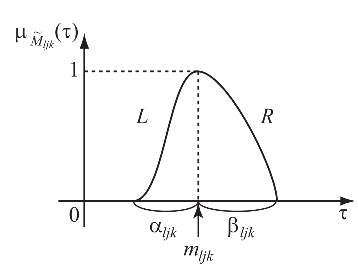

Fig. 1. An example of the membership function $\mu_{\tilde{M}_{ljk}}(\tau)$.

Let Γ be a collection of all possible Gaussian random variables $N(s,\sigma^2)$ where $s \in (-\infty,\infty)$ and $\sigma^2 \in (0,\infty)$. Then, $\tilde{\bar{C}}_{ljk}$ is expressed as a random fuzzy variable with the membership function

$$\mu_{\tilde{\bar{C}}_{ljk}}(\bar{\gamma}_{ljk}) = \left\{ \mu_{\tilde{M}_{ljk}}(s_{ljk}) | \bar{\gamma}_{ljk} \sim N(s_{ljk}, \sigma^2_{ljk}) \right\}, \quad \forall \bar{\gamma}_{ljk} \in \Gamma. \tag{4}$$

In view of (4), through the extension principle by Zadeh, the membership function of a random fuzzy variable corresponding to each of objective functions $z_l(\boldsymbol{x}_1, \boldsymbol{x}_2)$, $l = 1, 2$ is given as

$$\begin{aligned} \mu_{\tilde{\bar{\boldsymbol{C}}}_l \boldsymbol{x}}(\bar{u}_l) &= \sup_{\bar{\gamma}_l} \left\{ \min_{1 \leqslant k \leqslant n_j, j=1,2} \mu_{\tilde{\bar{C}}_{ljk}}(\bar{\gamma}_{ljk}) \middle| \bar{u}_l = \sum_{j=1}^{2} \sum_{k=1}^{n_j} \bar{\gamma}_{ljk} x_{jk} \right\} \\ &= \sup_{\boldsymbol{s}_l} \left\{ \min_{1 \leqslant k \leqslant n_j, j=1,2} \mu_{\tilde{M}_{ljk}}(s_{ljk}) \middle| \bar{u}_l \sim N\left(\sum_{j=1}^{2} \sum_{k=1}^{n_j} s_{ljk} x_{jk}, V_l(\boldsymbol{x}) \right) \right\}, \end{aligned} \tag{5}$$

where $\bar{\boldsymbol{\gamma}}_l = (\bar{\gamma}_{l11}, \ldots, \bar{\gamma}_{l1n_1}, \bar{\gamma}_{l21}, \ldots, \bar{\gamma}_{l2n_2})$, $\boldsymbol{s}_l = (s_{l11}, \ldots, s_{l1n_1}, s_{121}, \ldots, s_{l2n_2})$, and

$$V_l(\boldsymbol{x}) = \sum_{j=1}^{2} \sum_{k=1}^{n_j} \sigma^2_{ljk} x^2_{jk}.$$

Now assuming that the decision makers (DMs) concerns about the probabilities that their own objective function values $\tilde{\bar{\boldsymbol{C}}}_l \boldsymbol{x}$ are smaller than or equal to certain target variables f_l, $l = 1, 2$, we introduce the probabilities

$$\tilde{P}_l = P\{\omega | \tilde{\bar{\boldsymbol{C}}}(\omega) \boldsymbol{x} \leqslant f_l\}, \quad l = 1, 2 \tag{6}$$

which are expressed as fuzzy sets $\tilde{P}_l$ with the membership functions

$$\mu_{\tilde{P}_l}(p_l) = \sup_{\bar{u}_l} \{ \mu_{\tilde{\bar{\boldsymbol{C}}}_l \boldsymbol{x}}(\bar{u}_l) | \tilde{p}_l = P(\omega | \bar{u}_l(\omega) \leqslant f_l) \}.$$

Considering the imprecise nature of the DMs' judgments for the probabilities $\tilde{P}_l$ with respect to the random fuzzy objective function values $\tilde{\bar{\boldsymbol{C}}}_l \boldsymbol{x}$, $l = 1, 2$, we introduce the fuzzy goals $\tilde{G}_l$, $l = 1, 2$ such as "$\tilde{P}_l$ should be greater than or equal to a certain value." Such fuzzy goals $\tilde{G}_l$, $l = 1, 2$ can be quantified by eliciting corresponding membership functions

$$\mu_{\tilde{G}_l}(p) = \begin{cases} 0 & \text{if } p \leqslant p^0_l \\ g_l(p) & \text{if } p^0_l \leqslant p \leqslant p^1_l, \quad l = 1, 2 \\ 1 & \text{if } p^1_l \leqslant p, \end{cases} \tag{7}$$

where $g_l(p)$, $l = 1, 2$ are nondecreasing functions. Fig. 2 illustrates a possible shape of the membership function for the fuzzy goal $\tilde{G}_l$.

Recalling that the membership function $\mu_{\tilde{P}_l}(\cdot)$ is regarded as a possibility distribution, the degree of possibility that the probability $\tilde{P}_l$ attains the fuzzy goal $\tilde{G}_l$ is expressed as

$$\Pi_{\tilde{P}_l}(\tilde{G}_l) = \sup_{p_l} \min\{ \mu_{\tilde{P}_l}(p_l), \mu_{\tilde{G}_l}(p_l) \}, \quad l = 1, 2. \tag{8}$$

Fig. 3 illustrates the degree of possibility $\Pi_{\tilde{P}_l}(\tilde{G}_l)$.

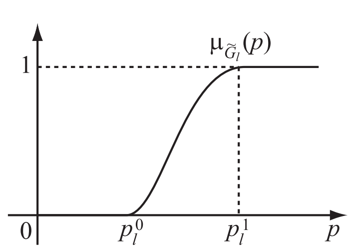

Fig. 2. An example of a membership function $\mu_{\tilde{G}_l}(y)$ of a fuzzy goal $\tilde{G}_l$.

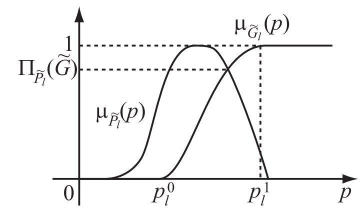

Fig. 3. The degree of possibility $\Pi_{\tilde{P}_l}(\tilde{G}_l)$.

Through the use of (8), the original random fuzzy two-level linear programming problem (2) can be interpreted as

$$\left. \begin{array}{ll} \underset{\text{for DM1}}{\text{maximize}} & Z_1^{\Pi,P}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \Pi_{\tilde{P}_1}(G_1) \\ \text{where } \boldsymbol{x}_2 \text{ solves} & \\ \quad \underset{\text{for DM2}}{\text{maximize}} & Z_2^{\Pi,P}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \Pi_{\tilde{P}_2}(G_2) \\ \quad \text{subject to} & A_1 \boldsymbol{x}_1 + A_2 \boldsymbol{x}_2 \leqslant \boldsymbol{b} \\ & \boldsymbol{x}_1 \geqslant \boldsymbol{0},\ \boldsymbol{x}_2 \geqslant \boldsymbol{0}. \end{array} \right\} \tag{9}$$

When the DMs would like to minimize their own objective function values under the condition that the degrees of possibility with respect to the attained probabilities are greater than or equal to certain permissible levels, we consider the two-level programming problem

$$\left. \begin{array}{ll} \underset{\text{for DM1}}{\text{maximize}} & h_1 \\ \text{where } \boldsymbol{x}_2 \text{ solves} & \\ \quad \underset{\text{for DM2}}{\text{maximize}} & h_2 \\ \quad \text{subject to} & \Pi_{\tilde{P}_1}(\tilde{G}_1) \geqslant h_1 \\ & \Pi_{\tilde{P}_2}(\tilde{G}_2) \geqslant h_2 \\ & A_1 \boldsymbol{x}_1 + A_2 \boldsymbol{x}_2 \leqslant \boldsymbol{b} \\ & \boldsymbol{x}_1 \geqslant \boldsymbol{0},\ \boldsymbol{x}_2 \geqslant \boldsymbol{0}. \end{array} \right\} \tag{10}$$

where h_1 and h_2 are permissible possibility levels specified by the DMs.

From (8), the constraints $\Pi_{\tilde{P}_l}(\tilde{G}_l) \geqslant h_l$, $l = 1, 2$ in (10) is equivalently replaced by the condition that there exists a p such that $\mu_{\tilde{P}_l}(p_l) \geqslant h_l$ and $\mu_{\tilde{G}_l}(p_l) \geqslant h_l$, namely,

$$\sup_{\boldsymbol{s}_l} \min_{1 \leqslant k \leqslant n_j, j=1,2} \left\{ \mu_{\tilde{M}_{ljk}}(s_{ljk}) \middle| p_l = P(\omega | \bar{u}_l(\omega) \leqslant f_l), \bar{u}_l \sim N\left(\sum_{j=1}^{2} \sum_{k=1}^{n_j} s_{ljk} x_{jk}, V_l(\boldsymbol{x}) \right) \right\} \geqslant h_l \tag{11}$$

and $p_l \geqslant \mu^{\star}_{\tilde{G}_l}(h_l)$, $l = 1, 2$, where $\mu^{\star}_{\tilde{G}_l}(h_l)$ are pseudo inverse functions defined as $\mu^{\star}_{\tilde{G}_l}(h_l) = \inf\{ p_l | \mu_{\tilde{G}_l}(p_l) \geqslant h_l \}$, $l = 1,2$. This implies that there exists a vector $(p_l, \boldsymbol{s}_l, \bar{u}_l)$, $l = 1, 2$ such that

$$\min_{1 \leqslant k \leqslant n_j, j=1,2} \mu_{\tilde{M}_{ljk}}(s_{ljk}) \geqslant h_l, l = 1, 2$$

$$\bar{u}_l \sim N\left(\sum_{j=1}^{2} \sum_{k=1}^{n_j} s_{ljk} x_{jk}, V_l(\boldsymbol{x}) \right), \quad P(\omega | \bar{u}_l(\omega) \leqslant f_l), p \geqslant \mu^{\star}_{\tilde{G}_l}(h_l),$$

which can be equivalently transformed into the condition that there exists a vector $(\boldsymbol{s}_l, \bar{u}_l)$ such that

$$\begin{aligned} &\mu_{\tilde{M}_{ljk}}(s_{ljk}) \geqslant h_l, l, j = 1, 2, k = 1, \ldots, n, \\ &\bar{u}_l \sim N\left(\sum_{j=1}^{2} \sum_{k=1}^{n_j} s_{ljk} x_{jk}, V_l(\boldsymbol{x}) \right), \quad P(\omega | \bar{u}_l(\omega) \leqslant f_l) \geqslant \mu^{\star}_{\tilde{G}_l}(h_l). \end{aligned} \tag{12}$$

In view of (3), it follows that

$$\mu_{\tilde{M}_{ljk}}(s_{ljk}) \geqslant h_l \iff s_{ljk} \in [m_{ljk} - L^\star(h_l)\alpha_l, m_{ljk} + R^\star(h_l)\beta_l],$$

where $L^\star(h_l)$ and $R^\star(h_l)$ are pseudo inverse functions defined as

$$L^\star(h_l) = \sup\{t|L(t) > h_l, r \geqslant 0\} \quad (0 \leqslant h_l \leqslant 1)$$

and

$$R^\star(h_l) = \sup\{t|R(t) > h_l, r \geqslant 0\} \quad (0 \leqslant h_l \leqslant 1), \quad l = 1, 2.$$

Hence, (12) is rewritten as the equivalent condition that there exists a $\bar{u}_l$ such that

$$\begin{aligned} &P(\omega|\bar{u}_l(\omega) \leqslant f_l), p \geqslant \mu^\star_{\tilde{G}_l}(h_l), \\ &\bar{u}_l \sim N\left(\sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{ljk}, V_l(\boldsymbol{x})\right). \end{aligned} \tag{13}$$

Since $\mathrm{P}(\omega|\bar{u}_l(\omega) \leqslant f_l)$ is transformed into

$$P\left(\omega\left|\frac{\bar{u}_l - \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk}}{\sqrt{V_l(\boldsymbol{x})}} \leqslant \frac{f_l - \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk}}{\sqrt{V_l(\boldsymbol{x})}}\right.\right)$$

in consideration of

$$\frac{\bar{u}_l - \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk}}{\sqrt{V_l(\boldsymbol{x})}} \sim N(0,1),$$

(13) is equivalently transformed as

$$\Phi\left(\frac{f_l - \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk}}{\sqrt{V_l(\boldsymbol{x})}}\right) \geqslant \mu^\star_{\tilde{G}_l}(h_l), \tag{14}$$

where Φ is a probability distribution function of the standard Gaussian random variable $N(0,1)$.

From the monotone increasingness of Φ, (14) is rewritten as

$$\sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk} + \Phi^{-1}(\mu^\star_{\tilde{G}_l}(h_l))\sqrt{V_l(\boldsymbol{x})} \leqslant f_l, \tag{15}$$

where Φ^{-1} is the inverse function of Φ.

In this way, from (11), (12), (14)–(16), it holds that

$$\begin{aligned} \Pi_{\tilde{P}_l}(\tilde{G}_l) \geqslant h_l \iff &\sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{ljk} - L^\star(h_l)\alpha_{ljk}\}x_{jk} \\ &+ \Phi^{-1}\left(\mu^\star_{\tilde{G}_l}(h_l)\right)\sqrt{V_l(\boldsymbol{x})} \leqslant f_l. \end{aligned} \tag{16}$$

Consequently, (10) is equivalently transformed into

$$\left.\begin{aligned} &\underset{\boldsymbol{x}_1}{\text{maximize}}\ h_1 \\ &\text{where } \boldsymbol{x}_2 \text{ solves} \\ &\underset{\boldsymbol{x}_2}{\text{maximize}}\ h_2 \\ &\text{subject to} \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{1jk} - L^\star(h_1)\alpha_{1jk}\}x_{1jk} + \Phi^{-1}\left(\mu^\star_{\tilde{G}_1}(h_1)\right)\sqrt{V_1(\boldsymbol{x})} \leqslant f_1 \\ &\qquad \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{2jk} - L^\star(h_2)\alpha_{2jk}\}x_{2jk} + \Phi^{-1}\left(\mu^\star_{\tilde{G}_2}(h_2)\right)\sqrt{V_2(\boldsymbol{x})} \leqslant f_2 \\ &\qquad A_1\boldsymbol{x}_1 + A_2\boldsymbol{x}_2 \leqslant \boldsymbol{b} \\ &\qquad \boldsymbol{x}_1 \geqslant \boldsymbol{0}, \boldsymbol{x}_2 \geqslant \boldsymbol{0}. \end{aligned}\right\} \tag{17}$$

4. Computational methods for obtaining Stackelberg solutions

We can now formulate the following problem for obtaining Stackelberg solutions.

$$\left.\begin{aligned} &\underset{\boldsymbol{x}_1}{\text{maximize}}\ h_1 \\ &\text{where } \boldsymbol{x}_2 \text{ solves} \\ &\underset{\boldsymbol{x}_2}{\text{maximize}}\ h_2 \\ &\text{subject to} \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{1jk} - L^\star(h_1)\alpha_{1jk}\}x_{jk} + \Phi^{-1}\left(\mu^\star_{\tilde{G}_1}(h_1)\right)\sqrt{V_1(\boldsymbol{x})} \leqslant f_1 \\ &\qquad \sum_{j=1}^{2}\sum_{k=1}^{n_j}\{m_{2jk} - L^\star(h_2)\alpha_{2jk}\}x_{jk} + \Phi^{-1}\left(\mu^\star_{\tilde{G}_2}(h_2)\right)\sqrt{V_2(\boldsymbol{x})} \leqslant f_2 \\ &\qquad A_1\boldsymbol{x}_1 + A_2\boldsymbol{x}_2 \leqslant \boldsymbol{b} \\ &\qquad \boldsymbol{x}_1 \geqslant \boldsymbol{0}, \boldsymbol{x}_2 \geqslant \boldsymbol{0}. \end{aligned}\right\} \tag{18}$$

Then we can introduce the following extended concepts of Stackelberg solution for the original random fuzzy two-level linear programming problem (2), where, for notational convenience, the feasible region of (18) is denoted by X_{PP}.

Definition 2. [PP-Stackelberg solution] A feasible solution $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) \in X_{PP}$ is called an PP-Stackelberg solution, meaning a Stackelberg solution through a possibility-based probability model, if $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*)$ is an optimal solution to (18).

In the context of noncooperative two-level programming problem (18), recall that DM1 first makes a decision, and then DM2 makes a decision so as to minimize the objective function with full knowledge of the decision of DM1. Namely, after DM1 specifies a decision, DM2 solves the corresponding programming problem, and chooses an optimal solution as a rational response. Assuming that DM2 chooses the rational response, DM1 also makes a decision such that the objective function of self is minimized. Then, the solution defined as such a procedure is a Stackelberg solution.

To be more explicit, a Stackelberg solution to the two-level programming problem (18) is defined as:

$$\left\{(\boldsymbol{x}_1, \boldsymbol{x}_2)\left|(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \arg\min_{(\boldsymbol{x}_1, \boldsymbol{x}_2) \in IR} Z_1^{\Pi,P}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right.\right\} \tag{19}$$

where

$$IR = \{(\boldsymbol{x}_1, \boldsymbol{x}_2)|(\boldsymbol{x}_1, \boldsymbol{x}_2) \in S,\ \boldsymbol{x}_1 \in R(\boldsymbol{x}_1)\} \tag{20}$$

is a inducible region, and

$$R(\boldsymbol{x}_1) = \left\{\boldsymbol{x}_2 \geqslant \boldsymbol{0}\left|\boldsymbol{x}_2 \in \arg\min_{\boldsymbol{x}_2 \in S(\boldsymbol{x}_1)} Z_2^{\Pi,P}(\boldsymbol{x}_1, \boldsymbol{x}_2)\right.\right\} \tag{21}$$

is a set of rational responses of DM2.

Observing that (18) is a two-level nonconvex nonlinear programming problem, in the following, we consider an approximate solution method for obtaining PP-Stackelberg solutions through particle swarm optimization for nonlinear programming (PSONLP) (Matsui et al., 2008).

To develop computational methods for obtaining Stackelberg solutions to the two-level programming problem (18) through PSONLP, noting that Stackelberg solutions always lie on the inducible region, it would be effective to explore in the vicinity of the inducible region intensively. Since the best position of the swarm is always involved in the search direction determination scheme of PSONLP, for the upper level decision variable vector of the best position of the swarm, obtaining the corresponding rational responses for a specified number of search generations, the vector to the inducible region is involved in the search direction. As a result, it becomes possible to explore around the inducible region effectively.

We can now present the following basic structure of the computational method for obtaining PP-Stackelberg solutions.

The computational method for obtaining PP-Stackelberg solutions

Step 1: Generate N initial feasible search positions $\left(\boldsymbol{x}_{i,1}^0, \boldsymbol{x}_{i,2}^0\right)$, $i = 1,\ldots,N$ for the upper and lower level decision variable vector $(\boldsymbol{x}_1,\boldsymbol{x}_2)$ using the homomorphous mapping. To be more specific, map N points generated randomly in the n dimensional hypercube $[-1, 1]^n$ to the feasible region using the homomorphous mapping, and let these points be initial search positions $\left(\boldsymbol{x}_{i,1}^0, \boldsymbol{x}_{i,2}^0\right)$, $i = 1,\ldots,N$.

Step 2: For the generated positions $\boldsymbol{x}_{i,1}^0$, using PSONLP, obtain the corresponding approximate rational responses $\hat{\boldsymbol{x}}_{i,2}^0$, and let the best positions of each particle be $\boldsymbol{p}_i^0 := \left(\boldsymbol{x}_{i,1}^0, \hat{\boldsymbol{x}}_{i,2}^0\right)$. Let $\boldsymbol{p}_g^0$ be the best position among $\boldsymbol{p}_i^0$ evaluated by the upper level objective function value.

Step 3: Let t:=0.

Step 4: Let $\boldsymbol{p}_i^{t+1} := \boldsymbol{p}_i^t$, $i = 1,\ldots,N$, and $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_g^t$.

Step 5: Following the search procedure of PSONLP, update the best search position of the particle. If $t := 100T_{up}$, $T_{up} = 1,\ldots,T_{\max}/100$, go to Step 5–1.

Step 5–1: For the upper level decision variables of the current best position $\boldsymbol{p}_i^t$ of the swarm and the current best position $\boldsymbol{p}_g^t$ of the particle, using PSONLP, obtain the corresponding approximate rational responses and update the best search position of the particle.

Step 6: If $t = T_{\max}$, then $\boldsymbol{p}_g^t$ is regarded as an approximate Stackelberg solution, and the algorithm is terminated. Otherwise, let t:=t + 1, and return to Step 4.

5. Numerical example

To demonstrate the feasibility and efficiency of the proposed method, as a numerical example of (2), consider the random fuzzy two-level linear programming problem

$$\left.\begin{aligned}
&\underset{\text{for DM1}}{\text{minimize}}\, z_1(\boldsymbol{x}_1,\boldsymbol{x}_2) = \bar{\tilde{\boldsymbol{C}}}_{11}\boldsymbol{x}_1 + \bar{\tilde{\boldsymbol{C}}}_{12}\boldsymbol{x}_2\\
&\text{where } \boldsymbol{x}_2 \text{ solves}\\
&\underset{\text{for DM2}}{\text{minimize}}\, z_2(\boldsymbol{x}_1,\boldsymbol{x}_2) = \bar{\tilde{\boldsymbol{C}}}_{21}\boldsymbol{x}_1 + \bar{\tilde{\boldsymbol{C}}}_{22}\boldsymbol{x}_2\\
&\text{subject to } \boldsymbol{a}_{11}\boldsymbol{x}_1 + \boldsymbol{a}_{12}\boldsymbol{x}_2 \leqslant b_1\\
&\qquad \boldsymbol{a}_{21}\boldsymbol{x}_1 + \boldsymbol{a}_{22}\boldsymbol{x}_2 \leqslant b_2\\
&\qquad \boldsymbol{a}_{31}\boldsymbol{x}_1 + \boldsymbol{a}_{32}\boldsymbol{x}_2 \geqslant b_3\\
&\qquad \boldsymbol{x}_1 = (x_{11},x_{12},x_{13})^T \geqslant \boldsymbol{0}\\
&\qquad \boldsymbol{x}_2 = (x_{21},x_{22},x_{23})^T \geqslant \boldsymbol{0}.
\end{aligned}\right\} \qquad (22)$$

Table 1 shows values of coefficients of constraints $\boldsymbol{a}_i$, i = 1, 2, 3 and b_i, i = 1, 2, 3 and Table 2 shows values of parameters of random fuzzy variables m_{ljk}, α_{ljk} and σ_{ljk}^2, l = 1, 2, j = 1, 2, k = 1,...,6, where triangular fuzzy numbers are assumed for $\mu_{\tilde{M}_{ljk}}(\tau)$.

Although the membership function does not always need to be linear, for the sake of simplicity, assume that the DMs subjectively determine the linear membership functions

$$\mu_{\tilde{G}_l}(p) = \begin{cases} 0 & \text{if } p \leqslant p_l^0\\ \frac{p-0.2}{0.7} & \text{if } p_l^0 \leqslant p \leqslant p_l^1, \quad l = 1,2\\ 1 & \text{if } p_l^1 \leqslant p, \end{cases}$$

Table 1
Values of coefficients in constraints.

	a_{l11}	a_{l12}	a_{l13}	a_{l21}	a_{l22}	a_{l23}	$\boldsymbol{b}$
$\boldsymbol{a}_1$	3	2	4	3	4	3	90
$\boldsymbol{a}_2$	4	4	4	5	3	5	120
$\boldsymbol{a}_3$	3	3	5	5	4	4	110

Table 2
Values of m_{ljk}, α_{ljk} and σ_{ljk}^2.

	$\bar{\tilde{c}}_{l11}$	$\bar{\tilde{c}}_{l12}$	$\bar{\tilde{c}}_{l13}$	$\bar{\tilde{c}}_{l21}$	$\bar{\tilde{c}}_{l22}$	$\bar{\tilde{c}}_{l23}$
m_{1jk}	−1.90	−2.60	−3.30	−2.60	−2.80	−2.30
m_{2jk}	−1.90	−2.40	−1.50	−1.90	−2.40	−2.00
α_{1jk}	0.80	0.20	0.50	0.40	0.60	0.30
α_{2jk}	0.20	0.80	0.30	0.40	0.70	0.60
σ_{1jk}^2	2.30	2.90	3.00	0.50	2.70	2.10
σ_{2jk}^2	2.70	2.60	2.20	3.70	3.00	3.20

for the probabilities $\widetilde{P}_1$ and $\widetilde{P}_2$ by assessing $p_1^0 = 0.2$, $p_2^0 = 0.2$, $p_1^1 = 0.9$ and $p_2^1 = 0.9$. Also assume that the DMs specify the target values as $f_1 = -61.0$ and $f_2 = -44.8$.

The parameter values of particle swarm optimization for nonlinear programming (PSONLP) are set as swarm size N = 50 and maximal search generation number $T_{\max}$ = 5000.

Through the use of the proposed computational method for obtaining approximate PP-Stackelberg solutions, we can obtain an approximate PP-Stackelberg solution

$$\begin{aligned}(\boldsymbol{x}_1,\boldsymbol{x}_2) &= (x_{11},x_{12},x_{13},x_{21},x_{22},x_{23})^T\\ &= (0.272650, 14.869833, 7.646532, 1.425490, 5.281371, 1.151432),\end{aligned}$$

where

$$h_1 = 0.581806$$

and

$$h_2 = 0.564814.$$

6. Conclusion

In this paper, computational methods for obtaining Stackelberg solutions for random fuzzy two-level linear programming problems have been presented. Through the introduction of the possibility-based probability model, the original random fuzzy two-level programming problems were reduced to deterministic two-level programming ones. For the transformed problems, extended concepts of Stackelberg solutions were introduced. Computational methods for obtaining approximate extended Stackelberg solutions through particle swarm optimization for nonlinear programming were also presented. An illustrative numerical example demonstrated the feasibility and efficiency of the proposed method. Extensions to other stochastic programming models will be considered elsewhere. Further considerations from the view point of random fuzzy cooperative two-level programming will be reported in the near future.

References

Amouzegar, M. A., & Moshirvaziri, K. (1999). Determining optimal pollution control policies: An application of bilevel programming. *European Journal of Operational Research, 119*(1), 100–120.

Bard, J. F., & Falk, J. E. (1982). An explicit solution to the multi-level programming problem. *Computers and Operations Research, 9*(1), 77–100.

Bard, J. F., & Moore, J. T. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing, 11*(2), 281–292.

Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming. *Management Science, 30*(8), 1004–1020.

Colson, B., Marcotte, P., & Savard, G. (2005). A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Computational Optimization and Applications, 30*(3), 211–227.

Dempe, S., & Bard, J. F. (2001). Bundle trust-region algorithm for bilinear bilevel programming. *Journal of Optimization Theory and Applications, 110*(2), 265–288.

Faisca, N. P., Dua, V., Rustem, B., Saraiva, P. M., & Pistikopoulos, E. N. (2007). Parametric global optimisation for bilevel programming. *Journal of Global Optimization, 38*(4), 609–623.

Fampa, M., Barroso, L. A., Candal, D., & Simonetti, L. (2008). Bilevel optimization applied to strategic pricing in competitive electricity markets. *Computational Optimization and Applications, 39*(2), 121–142.

Gümüs, Z. H., & Floudas, C. A. (2001). Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization, 20*(1), 1–31.

Hansen, P., Jaumard, B., & Savard, G. (1992). New branch-and-bound rules for liner bilevel programming. *SIAM Journal on Scientific and Statistical Computing, 13*(5), 1194–1217.

Kwakernaak, H. (1978). Fuzzy random variables – I. Definitions and theorems. *Information Sciences, 15*(1), 1–29.

Liu, B. (2002). Random fuzzy dependent-chance programming and its hybrid intelligent algorithm. *Information Sciences, 141*(3-4), 259–271.

Matsui, T., Sakawa, M., Kato, K., Uno, T., & Tamada, K. (2008). Particle swarm optimization for interactive fuzzy multiobjective nonlinear programming. *Scientiae Mathematicae Japonicae, 68*(1), 103–115.

Nahmias, S. (1978). Fuzzy variables. *Fuzzy Sets and Systems, 1*(2), 97–110.

Nicholls, M. G. (1996). The applications of non-linear bi-level programming to the aluminium industry. *Journal of Global Optimization, 8*(3), 245–261.

Nishizaki, I., & Sakawa, M. (1999). Stackelberg solutions to multiobjective two-level linear programming problems. *Journal of Optimization Theory and Applications, 103*(1), 161–182.

Nishizaki, I., & Sakawa, M. (2000). Computational methods through genetic algorithms for obtaining Stackelberg solutions to two-level mixed zero-one programming problems. *Cybernetics and Systems: An International Journal, 31*(2), 203–221.

Nishizaki, I., Sakawa, M., & Katagiri, H. (2003). Stackelberg solutions to multiobjective two-level linear programming problems with random variable coefficients. *Central European Journal of Operations Research, 11*(3), 281–296.

Puri, M. L., & Ralescu, D. A. (1986). Fuzzy random variables. *Journal of Mathematical Analysis and Applications, 114*(2), 409–422.

Roghanian, E., Sadjadi, S. J., & Aryanezhad, M. B. (2007). A probabilistic bi-level linear multi-objective programming problem to supply chain planning. *Applied Mathematics and Computation, 188*(1), 786–800.

Sakawa, M. (1993). *Fuzzy sets and interactive multiobjective optimization*. New York: Plenum Press.

Sakawa, M. (2001). *Genetic algorithms and fuzzy multiobjective optimization*. Boston: Kluwer Academic Publishers.

Sakawa, M., & Katagiri, H. (2012). Stackelberg solutions for fuzzy random two-level linear programming through level sets and fractile criterion optimization. *Central European Journal of Operations Research, 20*(1), 101–117.

Sakawa, M., & Kato, K. (2009). Fuzzy random noncooperative two-level linear programming through absolute deviation minimization using possibility and necessity. International Institute for Applied Systems Analysis (IIASA), Interim Report IR-09-021.

Sakawa, M., Katagiri, H., & Matsui, T. (in press). Stackelberg solutions for fuzzy random bilevel linear programming through level sets and probability maximization. Operational Research: An International Journal (in press).

Sakawa, M., & Nishizaki, I. (2009). *Cooperative and noncooperative multi-level programming*. New York: Springer.

Sakawa, M., Nishizaki, I., & Katagiri, H. (2011). *Fuzzy stochastic multiobjective programming*. New York: Springer.

Shimizu, K., Ishizuka, Y., & Bard, J. F. (1997). *Nondifferentiable and two-level mathematical programming*. Boston: Kluwer Academic Publishers.

Simaan, M., & Cruz, J. B. Jr., (1973). On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications, 11*(5), 533–555.

Wang, G-Y., & Qiao, Z. (1993). Linear programming with fuzzy random variable coefficients. *Fuzzy Sets and Systems, 57*(3), 295–311.

White, D. J., & Anandalingam, G. (1993). A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization, 3*(4), 397–419.