

Minimax mean-variance models for fuzzy portfolio selection

Xiaoxia Huang

Published online: 19 September 2010
© Springer-Verlag 2010

Abstract This paper discusses fuzzy portfolio selection problem in the situation where each security return belongs to a certain class of fuzzy variables but the exact fuzzy variable cannot be given. Two credibility-based minimax mean-variance models are proposed. The crisp equivalents of the models to linear programming ones are given in three special cases. In addition, a general solution algorithm is also provided. To help understand the modeling idea and to illustrate the effectiveness of the proposed algorithm, one example is presented.

Keywords Fuzzy portfolio selection · Minimax model · Portfolio optimization · Fuzzy programming · Investment analysis

1 Introduction

Portfolio selection has been one of the hottest research topics in financial area since Markowitz proposed his famous mean-variance theory (Markowitz 1952, 1959). Markowitz's great contribution is that he initially proposed a mathematical way for analyzing portfolio selection problem. Since Markowitz, mean-variance idea has been remaining the most popular principle for portfolio selection. Traditionally, security returns were assumed to be random variables. A variety of mean-variance models have been constantly developed (e.g., Abdelaziz et al. 2007; Corazza and Favaretto 2007; Hirschberger et al. 2007; Liu et al. 2003). As an improvement of mean-variance models,

mean-semivariance models have been proposed employing semivariance as the measure of risk (Chow and Denning 1994; Grootveld and Hallerbach 1999; Markowitz 1959). Different from defining the deviation from the expected value as risk, the probability of a pre-set level of loss was considered as risk by Roy (1952), Elton and Gruber (1995), William (1997) and so on. Recently, Huang (2008a) proposed a new definition of risk and presented a new mean-risk model by maximizing the expected return under the constraint that the risk curve of the portfolio is below the confidence curve.

Since many non-probabilistic factors may affect the security market, in many situations security returns contain much fuzziness instead of randomness. With the introduction and development of fuzzy set theory, scholars began to study portfolio selection problem with fuzzy returns. The researches mainly focused on extending mean-variance selection idea to fuzzy environment and a variety of fuzzy mean-variance models have been developed (Arenas-Parra et al. 2001; Bilbao-Terol et al. 2006; Carlsson et al. 2002; Tanaka et al. 2000). In particular, Huang (2007a, 2009) argued that credibility is a more suitable measure than possibility for a fuzzy event and proposed (Huang 2007a) two credibility-based fuzzy mean-variance models. As an extension of using semivariance to measure portfolio risk, Huang (2008b) defined the semivariance of fuzzy variable and proposed fuzzy mean-semivariance models for fuzzy portfolio selection.

Since the security market is so sensitive to various economic and political factors, sometimes exact random variables or fuzzy variables cannot be given to describe the security returns. Thus people have to handle portfolio selection with more complex uncertainty, and different mean-variance models were proposed. When the security returns are regarded to be random but the expected return

X. Huang (✉)
School of Economics and Management, University of Science
and Technology Beijing, Beijing 100083, China
e-mail: hxiaoxia@manage.ustb.edu.cn

of each underlying random asset varies in an estimated interval, Deng et al. (2005) proposed a stochastic minimax model by maximizing the worst possible expected rates of returns on portfolios. When the security returns are regarded to contain fuzziness with random parameters, Ammar (2008) and Li and Xu (2009) employed fuzzy random programming method to select the optimal portfolio. When the security returns are regarded to be random but the expected returns of the securities are fuzzy, Huang (2007b, 2008c) proposed different random fuzzy mean-variance models. In this paper, we will discuss the portfolio selection problem in a different complex uncertain environment where each underlying security is predicted to belong to a class of fuzzy variables but the exact fuzzy variable cannot be given.

The rest of the paper is organized as follows: to help better understand the paper, the mean-variance selection idea proposed in Huang (2007a) will be reviewed first in Sect. 2. Necessary knowledge about fuzzy variables will also be presented in the section. Then two credibility-based minimax portfolio selection models will be proposed in Sect. 3, and the crisp equivalents of the new models will be given in some special cases in Sect. 4. In Sect. 5, a hybrid intelligent algorithm will be provided for solving the optimization problems in general cases. In Sect. 6, one numerical example will be presented to illustrate the effectiveness of the proposed algorithm. Finally, the concluding comments will be presented in Sect. 7.

2 Fuzzy mean-variance selection idea

In the field of fuzzy portfolio selection, many scholars have extended Markowitz's mean-variance models (e.g., Arenas-Parra et al. 2001; Carlsson et al. 2002; Bilbao-Terol et al. 2006; Lacagnina and Pecorella 2006; Zhang et al. 2007). In particular, Huang (2007a, 2009) proposed that credibility is a more suitable measure than possibility measure for fuzzy portfolio selection because credibility is self-dual while possibility is not. Using possibility measure, we find that two fuzzy events with different occurring chances may have the same possibility value 1. Furthermore, we find that whenever the possibility value of a portfolio return greater than a target value is lower than 1, the possibility value of the opposite event (i.e., the portfolio return less than or equal to the target value) is the maximum value of 1; or whenever the possibility value of a portfolio return less than or equal to a target value is lower than 1, the possibility value of the opposite event (i.e., the portfolio return greater than the target value) is the maximum value of 1. For example, for a portfolio return described by the triangular fuzzy variable $\xi = (0, 1, 2)$, though the occurring chances of the return being lower than 1 and than

1.5 are different, the possibility values of the two events are the same value of 1, i.e., $\text{Pos}\{\xi < 1\} = \text{Pos}\{\xi < 1.5\} = 1$. In addition, it can be calculated that the possibility value of the portfolio return not greater than 0.5 is 0.5, while the possibility value of the opposite event is 1, i.e., $\text{Pos}\{\xi \leq 0.5\} = 0.5$, while $\text{Pos}\{\xi > 0.5\} = 1$. These results are quite awkward and will confuse the decision maker. Thus, Huang measured portfolio risk and return based on credibility. Furthermore, Huang argued that it is easier and more direct to describe a security return by fuzzy variable than by fuzzy probability. As a counterpart of Markowitz's mean-variance models in stochastic environment, Huang proposed two new credibilistic mean-variance models (Huang 2007a). Other fuzzy portfolio selection models can be found in Huang (2009).

Let x_i be the investment proportion in the i th security, and ξ_i the fuzzy variable that represents the i th security return. The ξ_i is defined as $\xi_i = (p'_i + d_i - p_i)/p_i$, $i = 1, 2, \dots, n$, respectively, where p'_i is the estimated closing price of the security i in the future, p_i the closing price of the security i at present, and d_i the estimated dividend of the security i from now to the future time. The mean-variance selection idea is to regard the expected value of the fuzzy portfolio return as the investment return and the variance the investment risk. Thus, the investors should select the portfolio with the maximum expected return under the constraint that the variance of the portfolio return will not be greater than a preset tolerable level. The mathematical expression of the idea is as follows:

$$\begin{cases} \max E[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n] \\ \text{subject to:} \\ V[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n] \leq \gamma \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n \end{cases} \quad (1)$$

where γ is the maximum tolerable variance level, and E the expected value operator and V the variance operator based on credibility measure defined as follows:

Let ξ be a fuzzy variable with membership function μ . The credibility measure is defined as (Liu and Liu 2002)

$$\text{Cr}\{\xi \in A\} = \frac{1}{2} \left(\sup_{t \in A} \mu(t) + 1 - \sup_{t \in A^c} \mu(t) \right) \quad (2)$$

for any set A of real numbers. It is easy to see that credibility is self-dual.

The expected value of ξ is defined as (Liu and Liu 2002)

$$E[\xi] = \int_0^{\infty} \text{Cr}\{\xi \geq r\} dr - \int_{-\infty}^0 \text{Cr}\{\xi \leq r\} dr \quad (3)$$

provided that at least one of the two integrals is finite. If the fuzzy variable ξ has a finite expected value, then its variance is defined as (Liu and Liu 2002)

$$V[\xi] = E[(\xi - E[\xi])^2]. \tag{4}$$

3 Fuzzy minimax mean-variance models

As discussed in Sect. 1, there are various kinds of uncertainty in the security market. Sometimes, the market is so complex that the investors only know that the security returns belong to a certain class of fuzzy variables. Then how should the investors select their optimal portfolio? In this section, we will handle such a complex fuzzy portfolio selection problem.

Suppose each security return ξ_i is fuzzy and known to belong to a certain class of fuzzy variables Γ_i . Let β be the maximum variance level the investors can tolerate. Then, for the safety purpose of decision making, the investors should first require that the maximum variance of the portfolio return in all cases should not be greater than the pre-set β level and then pursue maximum expected return in the worst case. To express the idea mathematically, we have the model as follows:

$$\begin{cases} \max_{x_i, 1 \leq i \leq n} \min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n] \\ \text{subject to:} \\ \max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n] \leq \beta \\ \xi_i \in \Gamma_i, \quad i = 1, 2, \dots, n \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n, \end{cases} \tag{5}$$

where E denotes the expected value operator of the fuzzy variables, and V the variance operator. The formula $\max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ gives the maximum variance value (i.e., the maximum risk level) of the portfolio return in all cases. The constraint $\max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n] \leq \beta$ means that the variance value of the portfolio return in any cases will not be greater than the preset level β . It is clear that the portfolios satisfying this constraint are safe ones. The formula $\min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ finds the expected value in worst case, and the objective function $\max_{x_i, 1 \leq i \leq n} \min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ means that the optimal portfolio should be the one with maximum expected return in the worst case.

If the investors preset a threshold of expected value b , they should first require that the minimum expected return in all cases should not be less than this pre-set level, and then pursue the portfolio with minimum variance value in the worst case. The mathematical expression is as follows:

$$\begin{cases} \min_{x_i, 1 \leq i \leq n} \max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n] \\ \text{subject to:} \\ \min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n] \geq b \\ \xi_i \in \Gamma_i, \quad i = 1, 2, \dots, n \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n. \end{cases} \tag{6}$$

4 Crisp equivalents

To use traditional methods to solve the models (5) and (6), we need to convert the models into traditional programming models. Normally distributed fuzzy variable and triangular fuzzy variable are two popular types of fuzzy variables. Let us first review them briefly and then discuss the crisp equivalents of the proposed models when all the security returns can be described by either of the two special types of fuzzy variables.

Example 1. A fuzzy variable ξ is called a *normally distributed fuzzy variable* if it has a normal membership function

$$\mu(r) = 2 \left(1 + \exp\left(\frac{\pi|r - e|}{\sqrt{6}\sigma}\right) \right)^{-1}, \quad r \in R, \sigma > 0,$$

where e is the expected value of ξ and σ^2 the variance value. We denote it by $\xi \sim \mathcal{N}(e, \sigma)$. It has been proven (Li and Liu 2007) that a normally distributed fuzzy variable has a maximum entropy among all types of fuzzy variables with expected value e and variance value σ^2 .

A security return with maximum entropy distributes most dispersively, and it will be most difficult to predict whether a specific return value will occur. Thus, if only the expected and variance values of the security return can be given, a conservative investor can use the normally distributed fuzzy variable to describe this security return. Usually the intervals of the expected return and the variance value can be obtained. Then the investor can use a class of normally distributed fuzzy variable to describe the security return.

Theorem 1 Assume that each security return ξ_i can be described by a class of normally distributed fuzzy variables, i.e., $\xi_i \sim \mathcal{N}(e_i, \sigma_i)$, where $e_i \in [e_{li}, e_{ui}]$, $\sigma_i \in [\sigma_{li}, \sigma_{ui}]$ (see Fig. 1). Then Model (5) can be converted into the following model:

$$\begin{cases} \max x_1 e_{l1} + x_2 e_{l2} + \dots + x_n e_{ln} \\ \text{subject to:} \\ x_1 \sigma_{u1} + x_2 \sigma_{u2} + \dots + x_n \sigma_{un} \leq \sqrt{\beta} \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n. \end{cases} \tag{7}$$

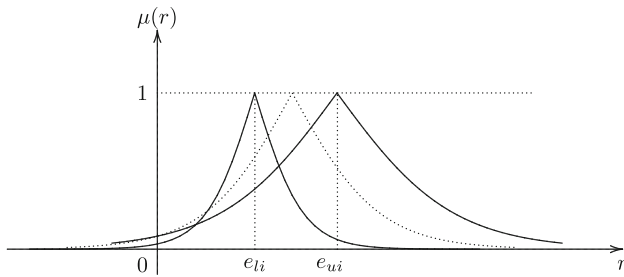


Fig. 1 Membership functions of a class of normally distributed fuzzy variables

Proof We know from Li and Liu (2007) that a weighted sum of normally distributed fuzzy variables is also normally distributed. That is, if fuzzy variables $\eta_1 \sim \mathcal{N}(e_1, \sigma_1)$ and $\eta_2 \sim \mathcal{N}(e_2, \sigma_2)$ are normally distributed fuzzy variables, then for any real numbers λ_1 and λ_2 , the fuzzy variable $\lambda_1\eta_1 + \lambda_2\eta_2$ is also a normally distributed fuzzy variable with expected value $\lambda_1e_1 + \lambda_2e_2$ and variance $(|\lambda_1|\sigma_1 + |\lambda_2|\sigma_2)^2$, whose membership function is

$$\mu(r) = 2 \left(1 + \exp \left(\frac{\pi |r - (\lambda_1 e_1 + \lambda_2 e_2)|}{\sqrt{6} (|\lambda_1| \sigma_1 + |\lambda_2| \sigma_2)} \right) \right)^{-1}$$

Therefore, $\max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ can be converted into $(x_1 \sigma_{u1} + x_2 \sigma_{u2} + \dots + x_n \sigma_{un})^2$ since $x_i \geq 0$ for $i = 1, 2, \dots, n$, which implies that only those securities whose returns are $\xi'_i \sim \mathcal{N}(e_i, \sigma_{ui}), e_i \in [e_{li}, e_{ui}], 1 \leq i \leq n$ need pass the constraint check. Then $\max_{x_i, 1 \leq i \leq n} \min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ now becomes $\max_{x_i, 1 \leq i \leq n} \min_{\xi'_i, 1 \leq i \leq n} E[x_1 \xi'_1 + x_2 \xi'_2 + \dots + x_n \xi'_n]$, and can be converted into $\max x_1 e_{l1} + x_2 e_{l2} + \dots + x_n e_{ln}$ since $x_i \geq 0$ for $i = 1, 2, \dots, n$. Thus, the theorem is obvious.

Example 2 A fuzzy variable ξ is called a *triangular fuzzy variable* if it has a triangular membership function which is given by

$$\mu(r) = \begin{cases} \frac{r-a}{b-a}, & \text{if } a \leq r \leq b \\ \frac{r-c}{b-c}, & \text{if } b \leq r \leq c \\ 0, & \text{otherwise.} \end{cases}$$

We denote it by $\xi = (a, b, c)$. Especially, if $b - a = c - b$, we call it the *symmetrical triangular fuzzy variable*. If the investor believes that the security returns can be described by the symmetrical triangular fuzzy variables but is not very sure about the lowest and highest values that the security returns may reach, he or she can use a class of symmetrical triangular fuzzy variables to describe the security returns.

Theorem 2 Assume that each security return ξ_i can be described by a class of symmetrical triangular fuzzy

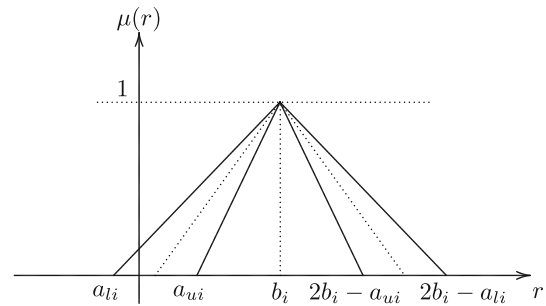


Fig. 2 Membership functions of a class of triangular fuzzy variables

variables $\xi_i = (a_i, b_i, 2b_i - a_i)$, where $a_i \in [a_{li}, a_{ui}], 1 \leq i \leq n$ (see Fig. 2). Then model (5) can be converted into the following model:

$$\begin{cases} \max x_1 b_1 + x_2 b_2 + \dots + x_n b_n \\ \text{subject to:} \\ x_1(b_1 - a_{l1}) + x_2(b_2 - a_{l2}) + \dots + x_n(b_n - a_{ln}) \leq \sqrt{6\beta} \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n. \end{cases} \tag{8}$$

Proof We know from Liu (2002) that a weighted sum of triangular fuzzy variables is also a triangular fuzzy variable. That is, if fuzzy variables $\eta_1 = (a_1, b_1, c_1)$ and $\eta_2 = (a_2, b_2, c_2)$ are triangular fuzzy variables, then for any real numbers $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, the fuzzy variable $\lambda_1\eta_1 + \lambda_2\eta_2 = (\lambda_1 a_1 + \lambda_2 a_2, \lambda_1 b_1 + \lambda_2 b_2, \lambda_1 c_1 + \lambda_2 c_2)$ is also a triangular fuzzy variable. It can be calculated that for a symmetrical triangular fuzzy variable $\eta = (a, b, c)$, its expected value is b and variance value is $(c - a)^2/24$. Therefore, $\max_{\xi_i, 1 \leq i \leq n} V[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ can be

converted into $(\sum_{i=1}^n x_i c_{ui} - \sum_{i=1}^n x_i a_{li})^2/24 = (\sum_{i=1}^n x_i b_i - \sum_{i=1}^n x_i a_{li})^2/6$, which implies that only those securities whose returns are $\xi'_i = (a_{li}, b_i, c_{ui}), 1 \leq i \leq n$ need pass the constraint check. Then $\max_{x_i, 1 \leq i \leq n} \min_{\xi_i, 1 \leq i \leq n} E[x_1 \xi_1 + x_2 \xi_2 + \dots + x_n \xi_n]$ now becomes $\max_{x_i, 1 \leq i \leq n} \min_{\xi'_i, 1 \leq i \leq n} E[x_1 \xi'_1 + x_2 \xi'_2 + \dots + x_n \xi'_n]$, and can be converted into $\max x_1 b_1 + x_2 b_2 + \dots + x_n b_n$. Thus, the theorem is obvious.

Theorem 3 Assume that each security return ξ_i can be described by a class of symmetrical triangular fuzzy variables $\xi_i = (a_i, a_i + t_i, a_i + 2t_i)$, where $a_i \in [a_{li}, a_{ui}], 1 \leq i \leq n$ (see Fig. 3). Then model (5) can be converted into the following model:

$$\begin{cases} \max x_1(a_{l1} + t_1) + x_2(a_{l2} + t_2) + \dots + x_n(a_{ln} + t_n) \\ \text{subject to:} \\ x_1 t_1 + x_2 t_2 + \dots + x_n t_n \leq \sqrt{6\beta} \\ x_1 + x_2 + \dots + x_n = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n. \end{cases} \tag{9}$$

Proof From the properties of the symmetrical triangular fuzzy variable, we know that the expected value of the fuzzy variable $\eta = (a, a + t, a + 2t)$ is $a + t$ and the variance value is $t^2/6$. Thus, the theorem is obvious.

5 Hybrid intelligent algorithm

Though in some special cases we can convert the proposed models into their crisp equivalents as shown in Sect. 4, however, in other cases, it is difficult to do so. Therefore, we need to find an algorithm to solve the proposed optimization problems in general cases. Genetic algorithm (GA) was proposed by Holland (1975). It does not require much mathematical analysis on the underlying optimization problems and has solved many complex optimization problems that are difficult to solve in traditional methods. In Huang (2007a), we have designed a fuzzy simulation based GA to solve the fuzzy mean-variance models in which fuzzy simulation is used to calculate the expected and variance values and GA is used for finding the optimal portfolio. In the algorithm, the solution time is mainly spent on the simulation procedures for calculating the expected and the variance values. Though the algorithm can be revised to solve the minimax models (5) and (6), however, when checking the constraints, we have to employ fuzzy simulation to calculate a sufficient number of expected values or the variance values in order to find the maximum variance value for Model (5) or the minimum expected value for Model (6). Thus, when using fuzzy simulation to calculate the expected value and the variance value, the algorithm will take a long time to find the optimal solution. Neural network (NN) is famous for approximating any nonlinear continuous functions over a closed bounded set (Leshno et al. 1993). In order to speed up the computational process, we employ an NN to approximate the expected and variance values, respectively. Then the trained NN is integrated into GA to find the optimal portfolio. In the following, we introduce in detail the NN training process and summarize the hybrid intelligent algorithm.

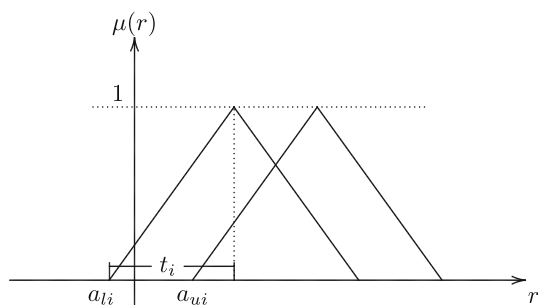


Fig. 3 Membership functions of a class of triangular fuzzy variables

5.1 Generating training data by fuzzy simulation

In order to train neurons to approximate the fuzzy expected value and variance value, one set of expected values and one set of variance values are first generated by fuzzy simulation as the training data sets. The detailed introduction for calculating expected and variance values by fuzzy simulation has been presented in Huang (2007a). Let $x = (x_1, x_2, \dots, x_n)$ represent the decision vector, where x_i are the investment proportions in the i th securities and n the number of the securities. Let $C = (c_1, c_2, \dots, c_n)$ denote the chromosomes, d_1 and d_2 the expected and variance values obtained by fuzzy simulation, respectively. The algorithm for generating training data is summarized as follows:

Step 1 Define the number of training data N in each training set.

Step 2 Set $j = 1$.

Step 3 Randomly generate a vector $C = (c_1, c_2, \dots, c_n)$ from the hypercube $[0, 1]^n$. Set

$$x_i = \frac{c_i}{c_1 + c_2 + \dots + c_n}, \quad i = 1, 2, \dots, n \tag{10}$$

which ensures that $x_1 + x_2 + \dots + x_n = 1$ always holds.

Step 4 Use fuzzy simulation to calculate the expected value $d_1 = E[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n]$ and variance value $d_2 = V[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n]$, respectively.

Step 5 If $j \leq N$, then $j = j + 1$ and go to step 3. Otherwise, stop.

5.2 Training NN

Determining network structure Before training the NN, we should determine the network structure, i.e., the number of hidden layers and the number of hidden neurons. Since in practice it is usually enough to train the NN with one hidden layer to approximate the uncertain function, in our paper, we consider the NN with one hidden layer connecting the input layer and the output layer in a feed-forward way. For example, for Model (5), we train one NN with n neurons in the input layer, h neurons in the hidden layer, and two neurons in the output layer to approximate the expected value function $E[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n]$ and the variance value function $V[x_1\xi_1 + x_2\xi_2 + \dots + x_n\xi_n]$, respectively. The number of hidden neurons h can be determined by the pruning algorithm (Castellano et al. 1997).

Training NN by backpropagation algorithm Let ω_{ij}^0 be the weight vector of the j th input neuron connected to the i th hidden neuron, ω_{ij}^1 the weight vector of the j th hidden neuron connected to the i th output neuron, and ρ the activation function of neurons in the hidden layer, e.g., $\rho(x) = \tanh(x)$. Then the outputs of the neurons in the hidden layer are

$$x_i^1 = \rho \left(\sum_{j=1}^n \omega_{ij}^0 x_j + \omega_{i0}^0 \right) = \tanh \left(\sum_{j=1}^n \omega_{ij}^0 x_j + \omega_{i0}^0 \right),$$

$$i = 1, 2, \dots, h, \quad (11)$$

and the outputs of the neurons in the output layer are

$$y_i = \sum_{j=1}^h \omega_{ij}^1 x_j^1 + \omega_{i0}^1, \quad i = 1, 2. \quad (12)$$

The aim of the NN training is to minimize the error between the target output produced by the NN and the actual output (in our case, the expected and variance values obtained by fuzzy simulation) by updating the weight values continuously.

Backpropagation algorithm is a well-known learning algorithm for NN training and is adopted in the paper. Assume that by fuzzy simulation we have obtained big enough training data (x, y_{1i}) and (x, y_{2i}) , $i = 1, 2, \dots, N$, for the expected and variance values, respectively. The weight values will be updated continuously in the training phase until the following error function

$$\text{Err}(\omega) = \frac{1}{2N} \sum_{k=1}^2 \sum_{i=1}^N \|y_{ki} - d_{k,i}\|^2 \quad (13)$$

is small enough, i.e.,

$$\text{Err}(\omega) \leq \varepsilon,$$

where ε is a preset small enough error rate.

To minimize the error function, we first randomly generate an initial weight vector ω for the neural network. Set $\Delta\omega_{ij}^1 = 0$ for $i = 1, 2$ and $j = 0, 1, 2, \dots, h$, and $\Delta\omega_{ij}^0 = 0$ for $i = 1, 2, \dots, h$ and $j = 0, 1, 2, \dots, n$. Then the weights are adjusted by an on-line learning process. Take the t th sample as an example. It is known from Eqs. (11) and (12) that the outputs of the hidden neurons are

$$x_{t,i}^1 = \tanh \left(\sum_{j=1}^n \omega_{ij}^0 x_{t,j} + \omega_{i0}^0 \right), \quad i = 1, 2, \dots, h,$$

and the outputs of the NN are

$$y_{t,i} = \sum_{j=1}^h \omega_{ij}^1 x_{t,j}^1 + \omega_{i0}^1, \quad i = 1, 2.$$

Then the error between the t th target output and actual output is

$$\text{Err}_t(\omega) = \frac{1}{2} \left((y_{1,t} - d_{1,t})^2 + (y_{2,t} - d_{2,t})^2 \right). \quad (14)$$

The weights of the NN will be adjusted continuously based on the error function $\text{Err}_t(\omega)$. For the hidden output weights ω_{ij}^1 , $i = 1, 2, j = 0, 1, 2, \dots, h$, we have

$$\Delta\omega_{ij}^1 \leftarrow -\alpha \frac{\partial \text{Err}_t}{\partial \omega_{ij}^1} + \eta \Delta\omega_{ij}^1, \quad \omega_{ij}^1 \leftarrow \omega_{ij}^1 + \Delta\omega_{ij}^1, \quad (15)$$

where $\frac{\partial \text{Err}_t}{\partial \omega_{ij}^1} = (y_{it} - d_{it})x_{t,i}^1$, and α and η are numbers between 0 and 1, e.g., $\alpha = 0.05$ and $\eta = 0.01$.

For the input-hidden weights ω_{ij}^0 , $i = 1, 2, \dots, h, j = 0, 1, 2, \dots, n$, we have

$$\Delta\omega_{ij}^0 \leftarrow -\alpha \frac{\partial \text{Err}_t}{\partial \omega_{ij}^0} + \eta \Delta\omega_{ij}^0, \quad \omega_{ij}^0 \leftarrow \omega_{ij}^0 + \Delta\omega_{ij}^0, \quad (16)$$

where $\frac{\partial \text{Err}_t}{\partial \omega_{ij}^0} = x_{t,j}(1 - (x_{t,i}^1)^2) \sum_{l=1}^2 (y_{lt} - d_{lt})\omega_{li}^1$.

When the NN has been trained over all the numbers of input-output training data, the average error is calculated according to the error function (13). Repeat the training process until the error is less than the predetermined precision ε or the maximum iteration times N' is met.

The algorithm for training the NN is summarized as follows:

- Step 1* Initialize weight vector ω , and set the parameters including error rate ε and α and η . Set $t = 0$, and $i = 1$.
- Step 2* $t \leftarrow t + 1$.
- Step 3* Adjust the weights according to (15) and (16).
- Step 4* Calculate the error Err_t according to (14).
- Step 5* If $t \leq N$, go to step 2.
- Step 6* Set $\text{Err} = \sum_{t=1}^N \text{Err}_t$.
- Step 7* If $\text{Err} > \varepsilon$, and $i < N'$, then $i \leftarrow i + 1$ and $t = 0$, and go to Step 2.
- Step 8* End.

5.3 Hybrid intelligent algorithm

Initialization After training NN, the trained NN is embedded into the GA to produce the hybrid intelligent algorithm. For the detailed introduction of the GA, the interested readers can refer to Huang (2007a). Here, we introduce in detail the initialization operation in which the feasibility checking and the computation of objective values are also introduced, and then summarize the hybrid intelligent algorithm.

When initializing chromosomes, for Model (5), first, randomly generate a point $\mathbf{C} = (c_1, c_2, \dots, c_n)$ from the hypercube $[0, 1]^n$. Next, randomly select fuzzy variables $\{\xi_{1k}, \xi_{2k}, \dots, \xi_{nk}\}$ from the corresponding class $\{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$, where $1 \leq k \leq N$ and N is a sufficiently large number. After that, use the trained NN to calculate the expected values $E[x_1\xi_{1k} + x_2\xi_{2k} + \dots + x_n\xi_{nk}]$ and the corresponding variance values $V[x_1\xi_{1k} + x_2\xi_{2k} + \dots + x_n\xi_{nk}]$. Among N couples of expected and variance values, find the maximum variance value denoted by V_{\max} and the minimum expected value whose corresponding variance value is V_{\max} . We denote this expected value by E_{\min} . Then the feasibility of the chromosome $\mathbf{C} = (c_1, c_2, \dots, c_n)$ is checked as follows:

If $V_{max} > \beta$ return 0;
return 1;

in which 1 means feasible, and 0 non-feasible.

For Model (6), first, randomly generate a point $\mathbf{C} = (c_1, c_2, \dots, c_n)$ from the hypercube $[0,1]^n$. Next, randomly select fuzzy variables $\{\xi_{1k}, \xi_{2k}, \dots, \xi_{nk}\}$ from the corresponding class $\{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$, where $1 \leq k \leq N$ and N is a sufficiently large number. After that, use the trained NN to calculate the expected values $E[x_1\xi_{1k} + x_2\xi_{2k} + \dots + x_n\xi_{nk}]$ and the corresponding variance values $V[x_1\xi_{1k} + x_2\xi_{2k} + \dots + x_n\xi_{nk}]$. Among N couples of expected and variance values, find the minimum expected value denoted by E_{min} and the maximum variance value whose corresponding expected value is E_{min} . We denote this variance value by V_{max} . Then the feasibility of the chromosome $\mathbf{C} = (c_1, c_2, \dots, c_n)$ is checked as follows:

If $E_{min} < b$ return 0;
return 1;

in which 1 means feasible, and 0 non-feasible.

If \mathbf{C} is checked to be feasible, it is taken as an initial chromosome. For Model (5), the expected value E_{min} is the objective value for this chromosome \mathbf{C} ; for Model (6), the variance value V_{max} is the objective value for this chromosome \mathbf{C} . If \mathbf{C} is checked to be infeasible, randomly generate another point \mathbf{C} from the hypercube $[0,1]^n$ until the point is proven to be feasible and taken as an initial chromosome. Repeat this process pop_size times, then initial feasible pop_size chromosomes $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{pop_size}$ are produced. In the meantime, we get the corresponding objective values for these initial feasible pop_size chromosomes.

Hybrid intelligent algorithm When the initial feasible chromosomes are obtained through initialization operation, the chromosomes will go through the operations of selection, crossover, and mutation. After finishing the whole round, the new population will be ready for its next round of selection, crossover, and mutation. The hybrid intelligent algorithm will continue until a given number of cyclic repetitions of the whole round is met (see Fig. 4). Summarization of the algorithm is as follows:

Step 1 Generate two training data sets for the expected and variance value functions by fuzzy simulation, respectively.

Step 2 Train an NN to approximate the expected and variance value functions, respectively, by back propagation algorithm.

Step 3 Initialize feasible pop_size chromosomes, in which the trained NN is used to calculate the constraint values of the chromosomes and check the feasibility of chromosomes.

Step 4 For all chromosomes use the trained NN to calculate the objective values.

Step 5 Give the rank order of the chromosomes according to the objective values, and compute the values of the rank-based evaluation function of the chromosomes.

Step 6 Compute the fitness of each chromosome according to the rank-based-evaluation function.

Step 7 Select the chromosomes by spinning the roulette wheel.

Step 8 Update the chromosomes by crossover and mutation operations, in which the trained NN is used to calculate the constraint values and to check the feasibility of the chromosomes.

Step 9 Repeat the fourth to the eighth steps.

Step 10 Terminate the repetition cycle when a fixed maximum number of generations is reached. Take the best chromosome as the solution for portfolio selection.

6 Numerical example

To help understand the optimization idea and to illustrate the effectiveness of the proposed algorithm, let us consider one numerical example which is performed on a HP personal computer with CPU 2.8 GHz and memory size 2.0 GB. The parameters are set as follows: the fuzzy simulation cycle is 3,000, the training data for the NN 3000, 15 hidden neurons, error rate $\varepsilon = 0.003$, maximum iteration times 3,000, the population size 30, the probability of crossover $P_c = 0.3$, the probability of mutation $P_m = 0.3$, the parameter a in the rank-based evaluation function is 0.05.

Example 3 According to the data in security markets and the experts' knowledge, usually the expected values of the security returns can be given. However, it is often difficult to predict the exact numbers of the other parameters of the security returns. From Shanghai Stock Market we have ten securities, among which five security returns belong to normally distributed fuzzy variables and the rest five ones belong to triangular fuzzy variables. The expected returns of the securities are given, but other parameters can only be provided in an interval. The data set is given in Table 1.

Suppose the investors accept 0.50 as the maximum tolerable variance level. Since the expected values of the security returns are known, according to Model (5), the minimax fuzzy portfolio selection model is given as follows:

Fig. 4 Hybrid intelligent algorithm

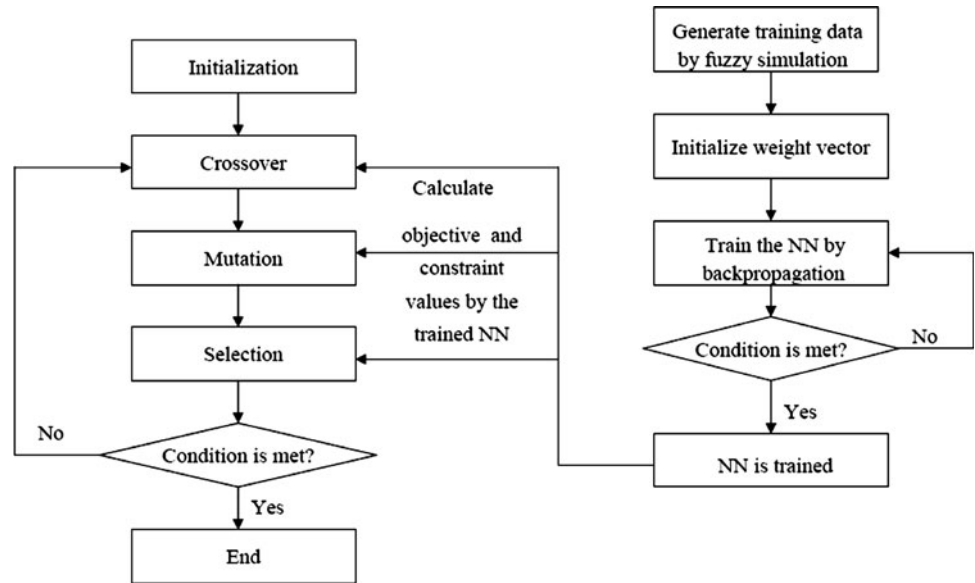


Table 1 Fuzzy returns of the ten securities

Security	Fuzzy return
1	$\xi_1 = (a_1, 1.40, 2.7 - a_1), a_1 \in [-0.2, 0.0]$
2	$\xi_2 = (a_2, 1.60, 3.1 - a_2), a_2 \in [-0.1, 0.3]$
3	$\xi_3 = (a_3, 1.80, 3.6 - a_3), a_3 \in [-0.4, -0.2]$
4	$\xi_4 = (a_4, 1.30, 2.6 - a_4), a_4 \in [-0.1, 0.1]$
5	$\xi_5 = (a_5, 2.30, 4.6 - a_5), a_5 \in [-0.6, 0.0]$
6	$\xi_6 \sim \mathcal{N}(2.0, \sigma_1), \sigma_1 \in [0.6, 0.8]$
7	$\xi_7 \sim \mathcal{N}(2.1, \sigma_2), \sigma_2 \in [0.7, 0.9]$
8	$\xi_8 \sim \mathcal{N}(1.5, \sigma_3), \sigma_3 \in [0.4, 0.6]$
9	$\xi_9 \sim \mathcal{N}(1.8, \sigma_4), \sigma_4 \in [0.5, 0.7]$
10	$\xi_{10} \sim \mathcal{N}(1.6, \sigma_5), \sigma_5 \in [0.5, 0.6]$

$$\begin{cases} \max E[\xi_1 x_1 + \xi_2 x_2 + \dots + \xi_{10} x_{10}] \\ \text{subject to:} \\ \max_{\xi_i, 1 \leq i \leq 10} V[\xi_1 x_1 + \xi_2 x_2 + \dots + \xi_{10} x_{10}] \leq 0.50 \\ x_1 + x_2 + \dots + x_{10} = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, 10. \end{cases} \quad (17)$$

In order to solve this model, we first generate by fuzzy simulation one training data set $\{x_{1i}, x_{2i}, \dots, x_{10i}, u_i | i = 1, 2, \dots, 3,000\}$ for the variance value function. An NN with one hidden layer is used. Employing pruning algorithm (Castellano et al. 1997), we set 15 hidden neurons in the hidden layer. Then the NN is trained to approximate the variance function. After training, we randomly generated 12 samples and the comparison of the variance values by fuzzy simulation and the trained NN is shown in Table 2, where "Difference" is the difference between the NN result and the simulation result. It can be seen that the maximum relative errors for the variance value do not exceed 1% which is considered acceptable.

Table 2 Comparisons of the values by the trained NN and the values by simulation

Variance by simulation	Difference	Relative error (%)
0.6216	-0.0059	0.95
0.7207	0.0061	0.85
0.5905	0.0042	0.71
0.5054	0.0012	0.24
0.6187	-0.0027	0.44
0.6632	-0.0063	0.95
0.5508	-0.0050	0.91
0.5711	0.0041	0.72
0.5363	-0.0044	0.82
0.5478	-0.0032	0.58
0.5589	0.0010	0.18
0.4896	0.0021	0.43

Next, we imbedded the trained NN into the GA to produce the hybrid intelligent algorithm. A run of the hybrid intelligent algorithm with 5,000 generations shows that among ten securities, in order to gain the maximum expected return in worst cases at the maximum variance value not greater than 0.50, the investors should assign their money according to Table 3. The corresponding maximum expected return is 1.8933. The solution time including NN training is about 12 min while the solution time by using the fuzzy simulation based GA without NN proposed in our previous paper (Huang 2007a) is about 4 h. By using NN, the time for calculating the variance values is greatly shortened. The solution time is mainly spent on the training of NN. Once the NN is trained, the optimal solution is found within only a few seconds.

Table 3 Allocation of money to the ten securities (%)

Security <i>i</i>	1	2	3	4	5	6	7	8	9	10
Allocation of money	1.17	2.74	0.00	0.00	2.59	63.44	1.09	3.32	10.87	14.78

Table 4 Different allocation plans (%)

Security <i>i</i>	1	2	3	4	5	6	7	8	9	10
Expected value	1.4	1.6	1.8	1.3	2.3	2.0	2.1	1.5	1.8	1.6
Maximum variance	0.43	0.48	0.81	0.33	1.40	0.64	0.81	0.36	0.49	0.36
Allocation 1	1.17	2.74	0.00	0.00	2.59	63.44	1.09	3.32	10.87	14.78
Allocation 2	0.00	0.00	1.34	0.00	10.18	19.71	66.28	0.00	2.49	0.00
Allocation 3	0.00	0.00	0.00	0.00	25.7	4.87	69.43	0.00	0.00	0.00

Allocation 1, 2, and 3 represent the allocation plans when the preset variance values are set at 0.5, 0.7, and 0.8, respectively

It is seen from Table 3 that most of the money is allocated to securities 6, 9, and 10. We calculate the expected and the maximum variance values of the individual ten securities and present them in Table 4. It is seen that security 6 has the third highest expected value with the maximum variance value bigger than 0.5 but lowest among the securities whose maximum variances are bigger than 0.5. Security 9 has the highest expected value among the securities whose maximum variance values are lower than 0.5 and security 10 has the second lowest maximum variance value with a not so low expected value. Thus, the allocation plan is reasonable. We increase the preset tolerable variance at 0.7. Since securities 5, 6, and 7 are securities with the highest expected value among all the ten securities, we expect that more money will be allocated to them with increase of the preset variance value. Since the maximum variance values of securities 5 and 7 are both higher than 0.7 with security 5 having the highest maximum variance value of 1.40, it is estimated that some other securities with low maximum variance may be included in the optimal portfolio. We run the algorithm and present the solution in Table 4. It is seen that the result is consistent with our expectation. We further increase the preset variance at 0.8 and expect that allocation of money to securities 5, 6 and 7 will be increased. The result shown in Table 4 confirms our expectation. Furthermore, it is seen that among securities 5, 6, and 7, when the preset variance value becomes bigger, allocation proportion on security 7 with both higher expected and variance values among the three securities becomes bigger. The result is also consistent with our judgement.

7 Conclusions

This paper proposes two credibility-based minimax mean-variance models for a fuzzy portfolio selection problem

where each security return belongs to a certain class of fuzzy variables but the exact fuzzy variable cannot be given. The proposed models are converted into linear programming models in three special cases. A neuron network imbedded genetic algorithm is provided for solving the proposed optimization problems in general cases. In the algorithm, the neuron network is used to approximate the expected value function and the variance value function so that the solution time is greatly shortened.

Acknowledgments This work was supported by National Natural Science Foundation of China Grant No.70871011, Program for New Century Excellent Talents in University, and the Fundamental Research Funds for the Central Universities.

References

Abdelaziz FB, Aouni B, Fayedh RE (2007) Multi-objective stochastic programming for portfolio selection. *Eur J Oper Res* 177:1811–1823

Ammar EE (2008) On solutions of fuzzy random multiobjective quadratic programming with applications in portfolio problem. *Inf Sci* 178:468–484

Arenas-Parra M, Bilbao-Terol A, Rodríguez-Uría MV (2001) A fuzzy goal programming approach to portfolio selection. *Eur J Oper Res* 133:287–297

Bilbao-Terol A, Pérez-Gladish B, Arenas-Parra M, Rodríguez-Uría MV (2006) Fuzzy compromise programming for portfolio selection. *Appl Math Comput* 173:251–264

Carlsson C, Fullér R, Majlender P (2002) A possibilistic approach to selecting portfolios with highest utility score. *Fuzzy Sets Syst* 131:13–21

Castellano G, Fanelli AM, Pelillo M (1997) An iterative pruning algorithm for feedforward neural networks. *IEEE Trans Neural Netw* 8:519–537

Chow K, Denning KC (1994) On variance and lower partial moment betas: the equivalence of systematic risk measures. *J Bus Financ Account* 21:231–241

Corazza M, Favaretto D (2007) On the existence of solutions to the quadratic mixed-integer mean-variance portfolio selection problem. *Eur J Oper Res* 176:1947–1960

Deng XT, Li ZF, Wang SY (2005) A minimax portfolio selection strategy with equilibrium. *Eur J Oper Res* 166:278–292

- Elton EJ, Gruber MJ (1995) *Modern portfolio theory and investment analysis*. Wiley, New York
- Grootveld H, Hallerbach W (1999) Variance vs downside risk: is there really that much difference?. *Eur J Oper Res* 114:304–319
- Hirschberger M, Qi Y, Steuer RE (2007) Randomly generating portfolio-selection covariance matrices with specified distributional characteristics. *Eur J Oper Res* 177:1610–1625
- Holland J (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor
- Huang X (2007a) Portfolio selection with fuzzy returns. *J Intell Fuzzy Syst* 18:383–390
- Huang X (2007b) Two new models for portfolio selection with stochastic returns taking fuzzy information. *Eur J Oper Res* 180:396–405
- Huang X (2008a) Portfolio selection with a new definition of risk. *Eur J Oper Res* 186:351–357
- Huang X (2008b) Mean-Semivariance Models for Fuzzy Portfolio Selection. *J Comput Appl Math* 217:1–8
- Huang X (2008c) Expected model for portfolio selection with random fuzzy returns. *Int J Gen Syst* 37:319–328
- Huang X (2009) A review of credibilistic portfolio selection. *Fuzzy Optim Decis Making* 8:263–281
- Lacagnina V, ecorella A (2006) A stochastic soft constraints fuzzy model for a portfolio selection problem. *Fuzzy Sets Syst* 157:1317–1327
- Leshno M, Ya V, Schochen S (1993) Multilayer feedforward networks with a nonpolynomial activation can approximate any functions. *Neural Netw* 6:861–867
- Li J, Xu JP (2009) A novel portfolio selection model in a hybrid uncertain environment. *Omega* 37:439–449
- Li X, Liu B (2007) Maximum entropy principle for fuzzy variable. *Int J Uncertain Fuzziness Knowl Based Syst* 15(Suppl):43–52
- Liu B (2002) *Theory and practice of uncertain programming*. Physica-Verlag, Heidelberg
- Liu B, Liu Y-K (2002) Expected value of fuzzy variable and fuzzy expected value models. *IEEE Trans Fuzzy Syst* 10:445–450
- Liu SC, Wang SY, Qiu WH (2003) A mean-variance-skewness model for portfolio selection with transaction costs. *Int J Syst Sci* 34:255–262
- Markowitz H (1952) Portfolio selection. *J Financ* 7:77–91
- Markowitz H (1959) *Portfolio selection: efficient diversification of investments*. Wiley, New York
- Roy AD (1952) Safety first and the holding of assets. *Econometrics* 20:431–449
- Tanaka H, Guo P, Türksen B (2000) Portfolio selection based on fuzzy probabilities and possibility distributions. *Fuzzy Sets Syst* 111:387–397
- Williams JO (1997) Maximizing the probability of achieving investment goals. *J Portf Manag* 46:77–81
- Zhang WG, Wang YL, Chen ZP, Nie ZK (2007) Possibilistic mean-variance models and efficient frontiers for portfolio selection problem. *Inf Sci* 177:2787–2801