Fuzzy Random Dependent-Chance Programming Models of Loan Portfolio

Dongjing Pan
Department of Computer Science and Technology, Dezhou University, Dezhou Shandong 253023, China
E-mail: pdj1970@163.com

Abstract—The environment of loan in bank is very complex, there are not only random factors but also fuzzy factors, so the return rates of loan often have fuzzy random characteristic. Mean chance is a measure of fuzzy random variable. This paper proposes two fuzzy random dependent-chance programming models of loan portfolio, one is minimize the mean chance of a bad outcome under the certain expected return rate, one is maximize the mean chance of the prospective return rate under the certain expected return rate. Hybrid intelligent algorithms are employed to solve the models. Finally, two numerical examples are given to show the validity and feasibility of the models and algorithms.

Index Terms—dependent-chance programming, loan portfolio, mean chance, fuzzy random

I. INTRODUCTION

In order to distribute risk, the bank puts loan in the different projects, which is loan portfolio. Essentially, it is portfolio selection. Loan portfolio is that the bank should decide how to allocate the certain capital in proportion so as to obtain the maximal return rates and the minimal risk. Since Markowitz[1] initialized the mean-variance model of portfolio selection, many scholars propose many different methods to solve portfolio problem. Tang[2] gave a kind of probability criterion portfolio investment model, in the model, the objective is to maximize the probability of the prospective return rare. Under the constraint of certain return rate, Sheedy[3] establishes the asset allocation decision model when the risk changes. Ning[4] gives chance programming model of loan portfolio when the return rate is fuzzy. Dietzsch and Petye[27] proposed a internal credit risks model about SME loan. Huang[16] measured portfolio risk by the variance based on credibility and proposed two new credibility-based fuzzy mean-variance models. Tanaka and Guo[15] quantified mean and variance of a portfolio through fuzzy probability and possibility distributions. These models’ objective is mainly maximize the return rates under the constraint of certain risk, or minimize the risk under the constraint of certain return rate. Risk is primarily mathematically defined in three ways: variance, semivariance and a probability of a bad outcome. Based on Markowitz’s mean-variance model, a large numbe of extensions have been proposed[5,6,7,8]. Semivariance is another measure of risk proposed by Markowitz[9], semivariance is an important improvement of variance because it only measures the investment return below the expected value. Many models have been built to minimize the semivariance from different angles[10,11]. The third popular definition of risk is a probability of a bad outcome initially by Roy[12]. Much research has been undertaken to find ways of minimizing the probability of the bad outcome[13,14].

However, the above studies mainly focused in two directions: stochastic environment and fuzzy environment. But the investment environment is so complex, sometimes we have to deal with the uncertainty of both fuzziness and randomness simultaneously. For example, the loan return rate can be regarded to be triangle fuzzy variable \((\rho - 0.1, \rho, \rho + 0.1)\), and \(\rho\) is random variable. Thus we have to face “fuzzy return rates with random parameters”. To deal with this type of uncertainty, this paper proposes that return rates be regarded as fuzzy random variable. Huang[23] gave a new optimal model of portfolio selection with random fuzzy returns, the paper proposed the primitive chance measure of risk, but the primitive chance measure only measures the maximum possibility of a random fuzzy or fuzzy random event occurs under a given probability level, and she did not research optimal model in fuzzy random environment. So in this paper, we consider the loan portfolio problem in fuzzy random environment, and because the mean chance measures the mean or expected possibility of the fuzzy random event, it can show the possibility of the fuzzy random event more extensive than primitive chance, so we use the mean chance of a bad outcome to measure the risk. Base on mean chance, this paper proposes two new dependent-chance programming models, one is minimize the mean chance of a bad outcome under the certain expected return rate, another is maximize the mean chance of the prospective return rate under the certain expected return rate, and designs hybrid intelligent algorithms to solve the models.

The rest of the paper is organized as follows. For better understanding of the paper, some basic knowledge about fuzzy random variables is introduced in section 2. In section 3, we propose two new dependent-chance programming models based on mean chance. In order to give a general algorithm for the models, hybrid intelligent
algorithms integrating fuzzy random simulation, neural network and genetic algorithm are designed in section 4. In section 5, two numerical examples are given to show the new models and the efficiency of the algorithms. Finally, a brief summary of this paper is given in section 6.

II PRELIMINARIES

Fuzzy random variable is a math description of fuzzy random phenomenon, it has different math definitions, it was first introduced by Kwakernaak[17,18], then Puri and Ralescu[19], Liu and Liu[20] gave the different measure of fuzzy random variable. And according to the need of different theory, many scholars gave the different mathematical definitions and different measures of fuzzy random variable. In this paper, we use the definitions of fuzzy random variable given by Liu and Liu[20]. Roughly speaking, a fuzzy random variable is a measurable function from a probability space to a collection of fuzzy variables. The primitive chance measure, mean chance measure, fuzzy chance measure of a fuzzy random event have been defined by Liu[21], and the concepts of expected value operator of fuzzy random variable was also presented by Liu[24]. Fuzzy random theory play an important role in solving optimization problems involving both fuzziness and randomness. In this paper, we will employ the fuzzy random theory to solve the loan portfolio problem in a fuzzy random environment.

In order to better understanding this paper, some concepts of probability, possibility, necessity and credibility measure were first briefly reviewed, and then we introduce the concept of a fuzzy random variable and the expected value, primitive chance measure, mean chance measure of a fuzzy random variable.

Definition 1 Let \( \Omega \) be a nonempty set, and \( A \) a \( \sigma \)-algebra of subsets of \( \Omega \). The set function \( \mathcal{P} \) is called a probability measure if:

1. \( \mathcal{P}\{\Omega\}=1 \);
2. \( \mathcal{P}\{A\}\geq 0 \) for any \( A \in \mathcal{A} \);
3. \( \mathcal{P}\{\bigcup_{i=1}^{n}A_{i}\}=\sum_{i=1}^{n}\mathcal{P}\{A_{i}\} \).

Then the triplet \( (\Omega,\mathcal{A},\mathcal{P}) \) is called a probability space.

Definition 2 Let \( \Theta \) be a nonempty set, and \( P(\Theta) \) the power set of \( \Theta \), if for each \( A \in P(\Theta) \), there is a nonnegative number \( \text{Pos}(A) \), called possibility, such that

1. \( \text{Pos}\{\emptyset\}=0, \text{Pos}\{\Theta\}=1 \);
2. \( \text{Pos}\{\bigcup_{i=1}^{n}A_{i}\}=\sup\text{Pos}\{A_{i}\} \) for any arbitrary collection \( \{A_{i}\} \) in \( P(\Theta) \).

Then the triplet \( (\Theta, P(\Theta), \text{Pos}) \) is called a possibility space.

Definition 3 Let \( \xi \) be a fuzzy variable on a possibility space \( (\Omega, P(\Omega), \text{Pos}) \) with membership function \( \mu \), and \( r \) a real number. The possibility, necessity, and credibility of a fuzzy event, characterized by \( \xi \leq r \), is defined by

\[
\text{Pos}(\xi \leq r) = \sup_{u \leq r} \mu(u),
\]

\[
\text{Nec}(\xi \leq r) = 1 - \text{Pos}(\xi > r) = 1 - \sup_{u > r} \mu(u),
\]

\[
\text{Cr}(\xi \leq r) = \frac{1}{2} (\text{Pos}(\xi \leq r) + \text{Nec}(\xi \leq r)).
\]

The expected value of a fuzzy variable is defined by

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

In order to avoid the action of \( \infty - \infty \), at least one of the two integrals of above formula is finite.

Definition 4 (Liu and Liu[20]) A fuzzy random variable \( \xi \) is a measurable function from a probability space \( (\Omega, \mathcal{A}, \mathcal{P}) \) to a collection of fuzzy variables.

Example 1 Let \( \xi = (\rho - 0.5, \rho, \rho + 1.5) \), and \( \rho \sim \exp(1) \), then \( \xi \) is a fuzzy random variable.

Example 2 Let \( (\Omega, \mathcal{A}, \mathcal{P}) \) be probability space, if \( \Omega = (\omega_1,\omega_2,\cdots,\omega_m) \) and \( \eta_1,\eta_2,\cdots,\eta_m \) are fuzzy variables. Then the function

\[
\xi(\omega) = \begin{cases} 
\eta_1, & \text{if } \omega = \omega_1 \\
\eta_2, & \text{if } \omega = \omega_2 \\
\vdots, & \vdots \\
\eta_m, & \text{if } \omega = \omega_n 
\end{cases}
\]

is a fuzzy random variable.

Definition 5 (Liu and Liu[20]) Let \( \xi \) be a fuzzy random variable defined in probability space \( (\Omega, \mathcal{A}, \mathcal{P}) \). The expected value of \( \xi \) is defined by

\[
E[\xi] = \int_{0}^{\infty} \mathcal{P}(\omega \in \Omega | E[\xi(\omega)] \geq r)dr
\]

\[-\int_{\infty}^{0} \mathcal{P}(\omega \in \Omega | E[\xi(\omega)] \leq r)dr.
\]

In order to avoid the action of \( \infty - \infty \), at least one of the two integrals of above formula is finite.

Definition 6 (Liu[21], Gao and Liu[22]) Let \( \xi = (\xi_1,\xi_2,\cdots,\xi_n) \) be fuzzy random vector that is defined in probability space \( (\Omega, \mathcal{A}, \mathcal{P}) \), \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is measurable function. Then the primitive chance of a fuzzy random event characterized by \( f(\xi) \leq 0 \) is a function from \([0,1]\) to \([0,1]\), defined as

\[
\text{Ch}(f(\xi) \leq 0)(\alpha) = \sup \beta \mathcal{P}(\omega \in \Omega | \mathcal{C}(f(\xi(\omega)) \leq 0) \geq \beta) \geq \alpha
\]
We call \( \text{Ch}[f(\xi) \leq 0](\alpha) \) \( \alpha \) primitive chance of the fuzzy random event \( f(\xi) \leq 0 \).

**Theorem 1** (Gao and Liu[22]) Let \( \xi \) be fuzzy random vector that is defined in probability space \((\Omega, A, \text{Pr})\), \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is measurable function, then \( \text{Ch}[f(\xi) \leq 0](\alpha) \) is a decreasing function of \( \alpha \).

**Definition 7** (Liu[24]) Let \( \xi = (\xi_1, \xi_2, \ldots, \xi_n) \) be fuzzy random vector that is defined in probability space \((\Omega, A, \text{Pr})\), \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is measurable function. Then the mean chance of a fuzzy random event characterized by \( f(\xi) \leq 0 \) is defined as

\[
\text{Ch}^a(\{f(\xi) \leq 0\}) = \int_0^\alpha \text{Ch}[f(\xi) \leq 0](\alpha) d\alpha
\]

The value of the primitive chance at \( \alpha \) measures the maximum possibility of a fuzzy random event occurs under a given probability level \( \alpha \), while the mean chance measures the mean or expected possibility of the fuzzy random event[26]. The geometric meaning of mean chance is shown in Fig.1, mean chance equals to the area encircled by the curve and the coordinate axis.

![Geometric meaning of mean chance](image)

**III TWO NEW DEPENDENT-CHANCE PROGRAMMING MODELS OF LOAN PORTFOLIO**

Supposing the bank will loan for \( n \) projects, let \( \mathbf{x}_i \) represent the loan proportion for the \( i \)th project, \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) is decision vector, \( \xi = (\xi_1, \xi_2, \ldots, \xi_n) \) is the vector that is composed of return rates of \( n \) kinds of loan, \( \xi_i \) represents the \( i \)th return rate, it is a fuzzy random variable, \( R_0 \) is the preset bad outcome return rate. In order to avoid risk, we can minimize the mean chance of the return rates less than the preset bad outcome \( R_0 \) under the constraint of expected return rates no less than \( \mu \), so the following model can be given:

\[
\begin{align*}
\min & \quad \text{Ch}^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \\
\text{s.t.} & \quad E[\sum_{i=1}^{n} x_i \xi_i] \geq \mu \\
& \quad \sum_{i=1}^{n} x_i = 1 \\
& \quad x_i \geq 0, \ i = 1, 2, \cdots, n
\end{align*}
\]

If we set the prospective return rate is \( R_i \), the target is to maximize the mean chance of the return rates more than \( R_i \) under the constraint of expected return rates no less than \( \mu \), we can get the following model:

\[
\begin{align*}
\max & \quad \text{Ch}^a(\sum_{i=1}^{n} x_i \xi_i \geq R_i) \\
\text{s.t.} & \quad E[\sum_{i=1}^{n} x_i \xi_i] \geq \mu \\
& \quad \sum_{i=1}^{n} x_i = 1 \\
& \quad x_i \geq 0, \ i = 1, 2, \cdots, n
\end{align*}
\]

Because the return rates of loan are fuzzy random variables, it is hard to find out crisp equivalents of the above models, so hybrid intelligent algorithms are employed to solve the models.

**IV HYBRID INTELLIGENT ALGORITHM**

Now we mainly take model(1) for example to illustrate the solving process. Since return rates are fuzzy random variables, it is difficult to solve model(1) in traditional ways. To provide a general solution to the model (1), we design a hybrid intelligent algorithm integrating genetic algorithm(GA), fuzzy random simulation and neural network(NN). Fuzzy random simulation is applied to compute the objective values of mean chance \( \text{Ch}^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \) and the expected return rate \( E[\sum_{i=1}^{n} x_i \xi_i] \), GA is employed to find the optimal solution. In order to reduce the computational work, neural network is trained to approximate the objective values of mean chance \( \text{Ch}^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \) and the expected return rates \( E[\sum_{i=1}^{n} x_i \xi_i] \).

**A Fuzzy random simulation:**

We should utilize fuzzy random simulation to estimate the uncertain functions[24]:

...
\[ U_1 : x \rightarrow Ch^a \left( \sum_{i=1}^{n} x_i \xi_i \leq R_0 \right) \]

\[ U_2 : x \rightarrow E[\sum_{i=1}^{n} x_i \xi_i] \]

Fuzzy random simulation for \( U_1(x) \) : we first compute primitive chance \( Ch(\sum_{i=1}^{n} x_i \xi_i \leq R_0)(\alpha) \) through step 1 to step 4.

Step 1 Generate \( \omega_1, \omega_2, \ldots, \omega_m \) from \( \Omega \) according to the probability measure \( Pr \).

Step 2 Compute the credibility \( \beta_k = Cr(\sum_{i=1}^{n} x_i \xi_i(\omega_k) \leq R_0), \quad k = 1, 2, \ldots, m \), respectively, by fuzzy simulation.

Step 3 Set \( m' \) as the integer part of \( am \).

Step 4 Return the \( m' \) th largest element in sequence \( \{\beta_1, \beta_2, \ldots, \beta_m\} \).

Let \( \alpha \) change from 0 to 1, then \( Ch^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \) can be computed through the following formula

\[ Ch^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) = \int_{0}^{1} Ch(\sum_{i=1}^{n} x_i \xi_i \leq R_0)(\alpha)d\alpha \]

In the above step 2, the fuzzy simulation process of \( Cr(f(\xi) \leq R_0) \) is described as follows:

Generate \( \theta_k \) in \( \Theta \) evenly and make \( Pos(\theta_k) \geq \varepsilon \) let \( v_k = Pos(\theta_k), \quad k = 1, 2, \ldots, N, \quad \varepsilon \) is a small number enough, the credibility of \( Cr(f(\xi) \leq R_0) \) can be estimated by the following formula

\[ L = \frac{1}{2} (\max_{1 \leq k \leq N} \{v_k \mid f(\xi(\theta_k)) \leq R_0\}) + \min_{1 \leq k \leq N} (1 - v_k \mid f(\xi(\theta_k)) > R_0)) \]

fuzzy simulation value of \( Cr(f(\xi) \leq R_0) \).

Fuzzy random simulation for \( U_2(x) \) is described as follows:

Step 1 Set \( e = 0 \).

Step 2 Generate \( \omega \) from \( \Omega \) according to the probability measure \( Pr \).

Step 3 \( e \leftarrow e + E[\sum_{i=1}^{n} x_i \xi_i(\omega)], \quad E[\sum_{i=1}^{n} x_i \xi_i(\omega)] \) can be computed by fuzzy simulation.

Step 4 Repeat the second to third steps for \( N \) times.

Step 5 \( E[\sum_{i=1}^{n} x_i \xi_i] \leftarrow e / N \).

In the above step 3, the fuzzy simulation process of expected value of \( E[f(\xi)] \) is as following step 1 to step 8.

Step 1 Set \( g = 0 \).

Step 2 Generate \( \theta_k \) evenly in \( \Theta \) and make \( Pos(\theta_k) \geq \varepsilon \) let \( v_k = Pos(\theta_k), \quad k = 1, 2, \ldots, N, \quad \varepsilon \) is a small number enough.

Step 3 Let \( a = f(\xi(\theta_1)) \wedge \cdots \wedge f(\xi(\theta_k)), \quad b = f(\xi(\theta_1)) \lor \cdots \lor f(\xi(\theta_k)) \).

Step 4 Generate \( r \) evenly in \([a, b]\).

Step 5 If \( r \geq 0 \), then \( g \leftarrow g + Cr(f(\xi) \geq r) \).

Step 6 If \( r < 0 \), then \( g \leftarrow g + Cr(f(\xi) \leq r) \).

Step 7 Repeat the fourth to sixth steps for \( N \) times.

Step 8 \( E[f(\xi)] = a \lor 0 + b \lor 0 + g \cdot (b - a) / N \).

B Train NN

We use BPA back propagation algorithm to train NN to approximate the objective value of mean chance

\[ Ch^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \] and the expected return rates \( E[\sum_{i=1}^{n} x_i \xi_i] \) [24]. First, generate training data set, one training data is expressed as \( \{x_1, x_2, \ldots, x_n, U_1, U_2\} \), where

\[ U_1 = Ch^a(\sum_{i=1}^{n} x_i \xi_i \leq R_0) \] and

\[ U_2 = E[\sum_{i=1}^{n} x_i \xi_i], \quad U_1, U_2 \] can be computed by fuzzy random simulation. When generating input data \( \{x_1, x_2, \ldots, x_n\} \), we set \( x_i = x_i / (x_1 + x_2 + \cdots + x_n), \quad i = 1, 2, \ldots, n \). Which ensure that \( \sum_{i=1}^{n} x_i = 1 \) always holds. Then use BPA back propagation algorithm to train NN. The training purpose is to find the most suitable weights \( \beta \) that can minimize the error between the output of NN and \( U_1, U_2 \). It is usually enough to train the NN with one hidden layer. In the paper, the NN has one hidden layer connecting the input layer and the output layer in a feed-forward way and has two neurons in the output layer.

Supposing the NN has \( l \) neurons in the input layer, \( p \) neurons in the hidden layer and \( m \) neurons in the output layer. Now, there are \( N \) samples

\[ \{x_{k,1}, x_{k,2}, \ldots, x_{k,n}; d_{k,1}, d_{k,2}, \ldots, d_{k,m}\}, \quad k = 1, 2, \ldots, N \).

When the \( k \)-th sample is used, the outputs of the hidden neurons are
The crossover until it is feasible. If \( n < 2 \), it is a parent, \( \sigma \) in \( \omega_n \) are \( \exists \) in \( \omega_n \) from the open interval \( r \), which ensures that \( \sum_i x_i = 1 \) always holds. Then check their feasibility by NN, if \( E[\sum_i x_i \xi] \geq \mu \), it is a feasible chromosome.

Selection process: We select chromosomes by spinning the roulette wheel such that the better chromosomes will have. The selection process is as follows:[24]

Firstly, If there are \( pos \_size \) chromosomes \( V_1, V_2, \ldots, V_{pop \_size} \) at the current generation, we can order these chromosomes from good to bad, the better the chromosomes is, the smaller the ordinal number it has. Let a parameter \( a \in (0, 1) \) in the genetic system be given, we can define the rank-based evaluation function as follows

\[
eval(V_i) = a(1 - a)^{i - 1}, \quad i = 1, 2, \ldots, pop \_size
\]

Note that \( i = 1 \) means the best chromosome, \( i = pop \_size \) means the worst one.

Secondly, calculate the cumulative probability \( q_i \) for each chromosome \( V_i \),

\[
q_0 = 0, \quad q_i = \sum_{j=1}^i \eval(V_j) / \sum_{i=1}^{pop \_size} \eval(V_j), \quad i = 1, 2, \ldots, pop \_size
\]

Where \( \eval(V) \) is evaluation function.

Thirdly, generate a random number \( r \) in \( (0, q_{pop \_size}) \), and select the chromosome \( V_i \) if \( r \) satisfies \( q_{i-1} < r < q_i \).

Fourthly, repeat the third step \( pop \_size \) times and obtain \( pop \_size \) copies of chromosome.

Crossover operation: A crossover parameter \( p_c \) is defined first[24]. Repeating the following process from \( i = 1 \) to \( pos \_size \) : generating a random number \( r \) from the interval \( [0, 1] \), the chromosome \( V_i \) is selected as a parent if \( r < p_c \). We denote the selected parents by \( V'_1, V'_2, V'_3, \ldots \), and divided them into the following pairs: \( (V'_1, V'_2), (V'_3, V'_4), (V'_5, V'_6), \ldots \). The crossover operation on each pair is illustrated by \( (V'_1, V'_2) \). At first, we generate a random number \( c \) from the open interval \( (0, 1) \), then the operator on \( V'_1 \) and \( V'_2 \) will produce two chile \( X \) and \( Y \) as follows:

\[
X = cV'_1 + (1 - c)V'_2, \quad Y = (1 - c)V'_1 + cV'_2
\]

If \( X = (x_1, x_2, \ldots, x_n) \) and \( Y = (y_1, y_2, \ldots, y_n) \), we mutate it in \( \omega_n \) from the interval \( [0, 1] \), the chromosome \( y_i \) is selected as a parent for mutation if \( 2 \leq i \leq n \), which ensures that \( \sum_i x_i = 1 \) and \( \sum_i y_i = 1 \) always hold.

Checking whether \( E[\sum_i x_i \xi] \geq \mu \) and \( E[\sum_i y_i \xi] \geq \mu \) through NN, if both children are feasible, then we replace the parents with them. If not, we keep the feasible one if it exists, and then redo the crossover operator by regenerating a random number \( c \) until two feasible children are obtained or a given number of cycles is finished.

Mutation operation[24]: We define a parameter \( p_m \) as the probability of mutation. This probability gives us the expected number of \( p_m \cdot pos \_size \) of chromosomes undergoing the mutation operations. Repeating the following steps from \( i = 1 \) to \( pos \_size \) : generating a random number \( r \) from the interval \( [0, 1] \), the chromosome \( V_i \) is selected as a parent for mutation if \( r < p_m \). For each selected parents \( V_i \), we mutate it in the following way. Let \( M \) be an appropriate large positive number. We choose a mutation direction \( d \) in \( R^n \) randomly. Let \( X = V + M \cdot d \). If \( X = (x_1, x_2, \ldots, x_n) \), we check the feasibility through NN, If \( X \) is not feasible, we set \( M \) as a random number between \( 0 \) and \( M \) until it is feasible. If the above process cannot find a feasible solution in a predetermined number of iterations, then we set \( M = 0 \).

D Hybrid intelligent algorithm

The hybrid intelligent algorithm that is integrated fuzzy random simulation, genetic algorithm and NN is summarized as follows[24]:

Step 1 Generate training data set for the following uncertain functions by fuzzy random simulation.
\[ U_1 : x \to Ch^a \left( \sum_{i=1}^{n} x_i \xi_i \leq R_0 \right) \]
\[ U_2 : x \to E \left[ \sum_{i=1}^{n} x_i \xi_i \right] \]

Step 2 Train NN to approximate the objective value of mean chance \( Ch^a \left( \sum_{i=1}^{n} x_i \xi_i \leq R_0 \right) \) and the expected return rates \( E \left[ \sum_{i=1}^{n} x_i \xi_i \right] \).

Step 3 Determine the population size \( \text{pop}_\text{size} \), crossover probability \( P_c \), mutation \( P_m \) in genetic algorithm.

Step 4 Initialize feasible \( \text{pop}_\text{size} \) chromosomes. Use the trained NN to check the feasibility of chromosomes.

Step 5 Update the chromosomes by crossover and mutation operations in which the feasibility of offspring may be checked by the trained neural network.

Step 6 Calculate the objective values for all chromosomes by the trained neural network.

Step 7 Compute the fitness of each chromosome according to the objective values.

Step 8 Select the chromosomes by spinning the roulette wheel.

Step 9 Repeat the fifth to eighth steps for a given number of cycles.

Step 10 Report the best chromosome as the optimal solution.

The method to solve model(2) is similar.

V NUMBER EXAMPLE

To illustrate the optimization idea and to test the effectiveness of the proposed algorithm, two numerical example is presented here. Supposing there are five kinds of loan in model(1) and model (2), each return rate is fuzzy random variable, described as follows.

\[ \xi_1 = (\rho_1 - 0.012, \rho_1 + 0.045, \rho_1 + 0.075, \rho_1 + 0.075) \]
\[ \rho_1 \sim N(0.01, 0.01^2) \]
\[ \xi_2 = (\rho_2 - 0.015, \rho_2 + 0.06, \rho_2 + 0.06) \]
\[ \rho_2 \sim N(0.02, 0.03^2) \]
\[ \xi_3 = (\rho_3 - 0.02, \rho_3 + 0.04, \rho_3 + 0.085, \rho_3 + 0.085) \]
\[ \rho_3 \sim N(0.01, 0.02^2) \]
\[ \xi_4 = (\rho_4 - 0.02, \rho_4 + 0.05, \rho_4 + 0.09, \rho_4 + 0.09) \]
\[ \rho_4 \sim N(0.03, 0.03^2) \]
\[ \xi_5 = (\rho_5 - 0.016, \rho_5 + 0.08, \rho_5 + 0.08) \]
\[ \rho_5 \sim N(0.02, 0.04^2) \]

Let \( R_0 = -0.02 \), \( \mu = 0.05 \), the model(1) is formulated as follows:

\[ \min Ch^a \left( \sum_{i=1}^{n} x_i \xi_i \leq -0.02 \right) \]
\[ s.t. \]
\[ E \left[ \sum_{i=1}^{n} x_i \xi_i \right] \geq 0.05 \]
\[ \sum_{i=1}^{n} x_i = 1 \]
\[ x_i \geq 0, \ i = 1, 2, \cdots, n \]

Step 1: The model(3) is solved through running hybrid intelligent algorithm, the parameters in the algorithm are set as follows: 500 cycles in simulation, 2000 data in NN (NN has 5 input neurons, 15 hidden neurons, 2 output neuron), 400 generations in GA, the population size \( \text{pop}_\text{size} = 30 \), the crossover probability \( P_c = 0.3 \), the mutation probability \( P_m = 0.2 \). The run of the hybrid intelligent algorithm shows the best allocation proportion is \( X^* = (0.4348, 0.0994, 0.2103, 0.2124, 0.0431) \).

The minimal mean chance of the return rates less than the preset bad outcome -0.02 is 0.073272. The genetic process of algorithm is shown as Fig.2:

![Fig.2 Genetic process of algorithm for model(3)](image)

In order to further test the effectiveness of the designed algorithm, we use more numerical experiments with different values of parameters in the GA. The results are shown in Table 1.

<table>
<thead>
<tr>
<th>Number of generations</th>
<th>( \text{pop}_\text{size} )</th>
<th>( P_c )</th>
<th>( P_m )</th>
<th>Objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>30</td>
<td>0.3</td>
<td>0.2</td>
<td>0.073272</td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td>0.5</td>
<td>0.2</td>
<td>0.073232</td>
</tr>
<tr>
<td>400</td>
<td>80</td>
<td>0.3</td>
<td>0.5</td>
<td>0.073221</td>
</tr>
<tr>
<td>500</td>
<td>60</td>
<td>0.1</td>
<td>0.4</td>
<td>0.073222</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>0.6</td>
<td>0.3</td>
<td>0.073225</td>
</tr>
<tr>
<td>800</td>
<td>30</td>
<td>0.3</td>
<td>0.3</td>
<td>0.073226</td>
</tr>
<tr>
<td>800</td>
<td>90</td>
<td>0.3</td>
<td>0.2</td>
<td>0.073228</td>
</tr>
<tr>
<td>1000</td>
<td>70</td>
<td>0.2</td>
<td>0.3</td>
<td>0.073222</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>0.3</td>
<td>0.2</td>
<td>0.073261</td>
</tr>
</tbody>
</table>
From table 1, we can see when different values of parameter in GA are set, the objective value changes very tiny, so the designed algorithm is robust to set parameters and effective to solve the model (3).

Let $R_1 = 0.07$, $\mu = 0.05$, the model (2) is formulated as follows:

$$\begin{align*}
\max Ch^{\alpha}(\sum_{i=1}^{n} x_i \xi_i \geq 0.07) \\
\text{s.t.} E[\sum_{i=1}^{n} x_i \xi_i] \geq 0.05 & \quad (4) \\
\sum_{i=1}^{n} x_i = 1 \\
x_i \geq 0, \ i = 1,2,\cdots, n
\end{align*}$$

Through running hybrid intelligent algorithm we solve model (4), the parameters setting are as same as above. The run of the hybrid intelligent algorithm shows the maximal mean chance of the return rates more than the prospective return rate 0.07 is 0.556277, the best allocation proportion is $X^* = (0.0192, 0.0059, 0.0363, 0.9251, 0.0136)$, the genetic process of algorithm is shown as Fig.3:

![Fig.3 Genetic process of algorithm for model(4)](image)

Similarly, we test the effectiveness of the designed algorithm for model (4) through setting different values of parameters in the GA. The results are shown in Table 2.

From table 2, we can see that the designed algorithm is robust to set parameters and effective to solve the model (4).

<table>
<thead>
<tr>
<th>Number of generations</th>
<th>pos. size</th>
<th>$P_c$</th>
<th>$P_m$</th>
<th>Objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>30</td>
<td>0.3</td>
<td>0.2</td>
<td>0.556277</td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td>0.5</td>
<td>0.2</td>
<td>0.567125</td>
</tr>
<tr>
<td>400</td>
<td>80</td>
<td>0.3</td>
<td>0.5</td>
<td>0.573905</td>
</tr>
<tr>
<td>500</td>
<td>60</td>
<td>0.1</td>
<td>0.4</td>
<td>0.574286</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>0.6</td>
<td>0.3</td>
<td>0.548707</td>
</tr>
<tr>
<td>800</td>
<td>30</td>
<td>0.3</td>
<td>0.3</td>
<td>0.557875</td>
</tr>
<tr>
<td>800</td>
<td>90</td>
<td>0.3</td>
<td>0.2</td>
<td>0.579590</td>
</tr>
<tr>
<td>1000</td>
<td>70</td>
<td>0.2</td>
<td>0.3</td>
<td>0.570203</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>0.3</td>
<td>0.2</td>
<td>0.568431</td>
</tr>
</tbody>
</table>

VI CONCLUSION

In the paper, we discuss the optimization of loan portfolio under fuzzy random environment, give two new dependent-chance programming models of loan portfolio based on mean chance and design hybrid intelligent algorithms integrating genetic algorithm, fuzzy random simulation and neural network to solve the models. At the end, two numerical examples are presented to illustrate the modelling idea and the effectiveness of the proposed algorithm.

REFERENCES


Dongjing Pan, was born in Dezhou Shandong, China in 1970. She received her bachelor’s degree from Shandong Teachers’ University in 1991, the specialty is computer science and technology, and she received her master's degree from Shandong Teachers’ University in 2002, the specialty is management science and engineering. Shandong Teachers’ University is in Jinan Shandong, China. She is a teacher in the department of computer science and technology, Dezhou University. Her job title is Associate Professor. Currently, her research interests include operating system, uncertain programming, hybrid intelligent algorithm, loan portfolio, risk investment, etc.

ACKNOWLEDGMENT
This paper is supported by
[1] Shandong Provincial Social Science Programming Research Project (No. 09CJGJ07)
[3] Scientific Research Development Plan Project of Shandong Provincial Education Department (No. J08LJ54)