Retraction

Retracted: Application of Multiobjective Particle Swarm Optimization in Rural Credit System

Journal of Sensors

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] W. Xue, "Application of Multiobjective Particle Swarm Optimization in Rural Credit System," *Journal of Sensors*, vol. 2021, Article ID 3468479, 9 pages, 2021.

Research Article

Application of Multiobjective Particle Swarm Optimization in Rural Credit System

Wuxia Xue (ID)

Department of Economics, Wenzhou Vocational College of Science and Technology, Zhejiang 325006, China

Correspondence should be addressed to Wuxia Xue; xuewuxia@wzvcst.edu.cn

In China's rural credit system, the problem of credit constraints is prominent. Due to the imperfect credit market, a large number of rural residents have credit constraints. Rural credit constraint is a serious problem restricting China's rural economic development. Aimed at solving the rural credit constraints, this paper makes an optimization analysis on the rural credit system and loan decision-making. To more reasonably evaluate customers' borrowing ability, the credit risk based on farmers' data on the big data platform is evaluated in this paper. The stacked denoising autoencoder network is improved by adopting the deep learning framework to improve the accuracy of credit evaluation. For improving the loan decision-making ability of rural credit system, a loan optimization strategy based on multiobjective particle swarm optimization algorithm is proposed. The simulation results show that the optimization ability, speed, and stability of the proposed algorithm have achieved good results in dealing with the loan portfolio decision-making problem.

1. Introduction

In the period of market economy, farmers' consumption has been monetized and socialized to a great extent. On the one hand, the products and services needed by farmers are no longer self-sufficient. Like urban residents, it uses money as the medium to exchange needed products. On the other hand, the income level of farmers is low, and the income and expenditure are highly inaccurate. There is often an imbalance between farmers' income and consumption expenditure, so the need of the forming productive and consumer financing is exiting. A serious problem facing China's rural economic development is rural credit constraints [1]. Due to the imperfect credit market, a large number of rural residents have credit constraints. The probability of farmers getting loans is very low, which has also become one of the important reasons to curb farmers' consumption [2].

China's rural residents have strong credit constraints [3]. In 2007, the People's Bank of China and the National Bureau of statistics conducted a special survey on 20000 farmers in 10 provinces. The results show that 46 percent of farmers have loan needs, but only about 26 percent of farmers with

loan needs can obtain loans through formal financial institutions. According to the survey, the network coverage of financial service institutions in rural areas is insufficient. By the end of June 2009, there were 2945 townships in China without any financial institutions. There are more than 8000 townships and towns with only one bank network. Among all townships and towns in China, the serious shortage of financial service accounts for 1/3.

The rural credit constraints are caused with four reasons as follows: (1) There are defects in the rural financial system. The Agricultural Development Bank has a single business, which is limited to the issuance of grain and cotton loans. At the same time, the business scope of most banks has not yet involved rural areas. (2) The construction of rural social credit system lags behind, and there is information asymmetry. The reasons why formal financial institutions are unwilling to lend to farmers also include that financial institutions are unable to understand the accurate information of rural borrowers. For example, farmers' credit level and repayment ability, what is the actual purpose of their loans, what risks will there be, and so on. This makes financial institutions reluctant to extend loans to rural residents.

(3) Rural financial institutions generally require mortgage guarantee. However, the main properties owned by rural residents are land with only use right and real estate without mobility. These properties can not be used as collateral for bank loans in the existing financial regulations. Other properties owned by farmers are mainly livestock and means of production. These can not be used as collateral for loans. (4) The credit products issued by financial institutions are not suitable for the consumption needs of rural residents. The important characteristics of Chinese rural residents' loan behavior are the huge number of farmers, high dispersion, and small loan amount. These require financial institutions to have very high operation efficiency in order to meet the needs of rural loan business [4]. The analysis of this paper focuses on the second and fourth points here, which study to improve the accuracy of credit assessment and optimize the loan program.

The business purpose of the bank is to maximize profits and minimize risks. It follows the principle of efficiency, safety, and liquidity. Risk loan portfolio rationing decision-making is a process of selecting combining appropriate loan objects from many loan objects on the premise of comprehensively considering loan income and risk. Literature [5] established a loan portfolio optimization decision-making model based on the principle of maximum unit risk return. The above combinatorial optimization problem belongs to NP (nondeterministic polynomial) hard problem [6]. The solving process of the problem is simple and easy when the scale is small, but with the increase of the scale of the problem, the amount of calculation increases exponentially. Therefore, it is necessary to design a better algorithm that takes into account the quality of solution and running time.

Particle swarm optimization (PSO) is inspired by the research results of bird swarm foraging behavior [7, 8]. The particle swarm optimization algorithm has the advantages of fast convergence, simple operation, and easy implementation. The algorithm has attracted extensive attention of scholars in the fields of evolutionary computing [9], computer science [10], and management science [11] and has achieved a lot of research results [12]. Most of these research results are obtained from the research of various multidimensional continuous space optimization problems, but there are few application research results in discrete optimization problems [13].

The innovations and contributions of this paper are listed below:

(1) In order to evaluate farmers' borrowing ability more reasonably, the deep learning framework is used to evaluate farmers' personal credit. Due to the existence of high-dimensional data in the bank's big data platform, the stacked denoising autoencoder network is firstly used for feature compression and extraction

(2) Different from the previous bank credit evaluation, this paper takes bank big data as the data source of risk evaluation. It enriches the characteristics of credit evaluation and improves the accuracy of evaluation

(3) In order to improve the loan decision-making ability of rural credit system, an adaptive decomposition multiobjective particle swarm optimization algorithm is designed in this paper. Then, by calculating the optimal solution and the membership information of the subspace, the appropriate solution in the external file is selected, and an external file update strategy based on decomposition method is designed. Finally, the convergence of the algorithm is improved by balancing the exploration and development ability of the algorithm

The structure of this paper is as follows. The related theories are described in Section 2. Section 3 focuses on the proposed algorithm model in this paper. Section 4 is experiment and analysis. Section 5 is the conclusion.

2. Related Theories

2.1. Selection of Risk Assessment Features. The data used by banks to assess personal credit risk mainly comes from the collection of customer data, such as the central bank's credit investigation data, basic personal information, financial status, credit records, and debts [14]. With the expansion of various businesses of the bank, especially the cross-marketing of internal customers, customer information is constantly enriched. Bank customers no longer belong to a single business category, and the footprints left by customers in multiple business areas of the bank will become the data source of personal credit risk assessment. These data have also become a necessary supplement to personal credit portrait.

Based on the traditional evaluation features, namely, credit card lending and loan repayment data, this paper relies on the big data platform of a commercial bank. Customer cross-business information is added to the evaluation system to integrate the two parts of data, so as to build a complete bank personal credit portrait. Figure 1 shows the logical representation of data splicing and integration.

2.2. Multiobjective Optimization Problem. For a multiobjective optimization problem (MOP) [15], its objective function can be described as

$$\text{Min } (H(i)) = [h_1(i), h_2(i), \cdots, h_W(i)], \tag{1}$$

where i is an n-dimensional decision variable. $h_W(i)$ is the wth objective function. W is the number of objective functions.

For a set of decision variables i and j, it is assumed that the problem to be optimized is the minimization target value. If $h(i)$ is less than or equal to $h(j)$ for any target value, and at least one target of $h(i)$ is less than $h(j)$, then i dominates j.

$$\begin{aligned} \forall x : h_x(i) \leq h_x(j), \\ \exists y : h_y(i) < h_y(j), \end{aligned} \tag{2}$$

where $x = 1, 2, \cdots, W, y = 1, 2, \cdots, W$. If i and j are dominant, then i and j can be compared. Otherwise, i
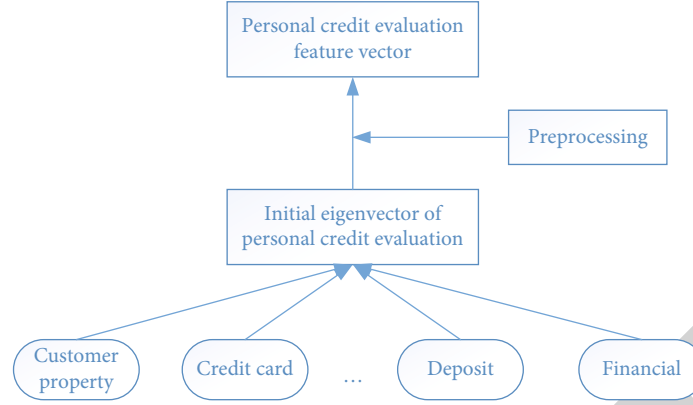
Figure 1: Construction of personal credit evaluation characteristics.

and j cannot be compared. When a solution is not dominated by any solution, it is called a nondominated solution. All the nondominated solutions finally obtained by the algorithm are the optimal solution set obtained by the algorithm.

Multiobjective particle swarm optimization algorithm uses a group of randomly distributed populations to guide the population to search the optimization space by selecting the appropriate solution. A multiobjective optimization algorithm is used to obtain the optimal solution set of MOP. In each iteration, the update formulas of particle velocity and position are

$$i_x(n+1) = i_x(n) + q_x(n+1), \tag{3}$$

$$q_x(n+1) = \omega_x q_x(n) + e_{x,1}R_{x,1}(u_x(n) - i_x(n)) + e_{x,2}R_{x,2}(a(n) - i_x(n)), \tag{4}$$

where ω_x is the inertia weight. $e_{x,1}$ and $e_{x,2}$ are learning factors. $R_{x,1}$ and $R_{x,2}$ are random values. $u_x(n)$ is the individual optimal position of the particle in the nth iteration. $a(n)$ is the global optimal position of the tth iteration. $i_x(n)$ is the position of the ith particle in iteration n. $q_x(n+1)$ is the velocity of the ith particle at iteration n.

3. Application of Multiobjective Particle Swarm Optimization Algorithm

3.1. Credit Evaluation Based on Deep Learning. The stacked autoencoder network applied in this paper is a special structure of deep learning. It is mainly applied to dimensionality reduction of high-dimensional data, namely, feature compression [16]. Firstly, a shallow neural network is constructed for the input item, hidden layer, and output item, and the hidden layer is solved when the output item is closest to the input item. Then, the hidden layer is used as the input of the next shallow network. Repeat the above process to solve the new hidden layer. Finally, the hidden layers are stacked layer by layer in the form of stack to form a deep neural network.

Big data samples have noise caused by data deviation, error, and missing. In order to enhance the antinoise ability

of the autoencoder, the denoising autoencoder (DAE) adds noise to the original data to generate a shallow neural network structure and then compresses and extracts its features [17]. The algorithm has good performance in multidimensional and high noise data scenarios. Firstly, the training sample $i(i \in R^t)$ is subjected to a series of random transformations $v_f(i' \mid i)$, and the variable i' is obtained after corrosion. Then, the reconstructed input i'' is obtained by learning and training the autoencoder of i'.

In the neural network, the gradient descent method is used for iterative solution, so that i'' is closest to the original data i. At this time, the corresponding hidden layer j (or compressed feature layer) is

$$j = \mathrm{argmin}\left[\sum_{x=1}^{t} \frac{1}{t} L\left(i_x, i_x'\right) + Y_m\right], \tag{5}$$

where $i_x \in i, i_x' \in i''(x = 1, 2, \cdots, t)$ are the corresponding elements belonging to the original feature and reconstruction feature, respectively. L is the loss function of mean square error or cross entropy. Y_m is a norm constraint added to prevent overfitting in the process of model learning.

The random transformation generated by the original denoising autoencoder network only adds a small change to the characteristics of the original data. It does not take into account the noise correlation between input features, which is obviously insufficient to improve the robustness of the model. On the premise of considering the correlation of sample noise, in order to improve the quality of noise generation, its truncation form is used to improve the computational efficiency.

Firstly, by analyzing the correlation between input features, the positive definite correlation matrix is determined Σ_{II}. For any $t \times 1$ vector g_z and diagonal matrix $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \cdots, \lambda_t]$, equation (6) can be established.

$$\Sigma_{II} g_z = \Lambda g_z, \tag{6}$$

where λ_x is a nonzero characteristic term of the correlation matrix Σ_{II}. Expansion can be defined as the decomposition of correlation matrix.

$$\Sigma_{II} = \sum_{x=1}^{t} \sqrt{\lambda_x} q_x q_x^N, \tag{7}$$

where q_x is the normalized eigenvalue of λ_x. Therefore, the random variable can be expressed as

$$I = \mu_I + \sum_{x=1}^{t} \sqrt{\lambda_x} q_x I_x, \tag{8}$$

where μ_I represents the mean of the random variable I. Since normalization will be carried out first in the calculation process, it can be assumed that $\mu_I = 0$. I is a random number. It follows the standard normal distribution, $I \sim T(0, 1)$. In practice, in order to reduce the amount of calculation, the truncated approximation can be obtained only by obtaining the W term ($w < t$).

$$\widehat{I} = \sum_{x=1}^{W} \sqrt{\lambda_x} q_x I_x. \tag{9}$$

The item W needs to ensure that the truncation error is less than 0.05. The truncation error is defined as

$$\text{err}_{r-\text{Var}} = \frac{\text{Var}[I - \widehat{I}]}{\text{Var}[I]} = \frac{\sum_{x=W+1}^{t} \lambda_x q_x^2}{\sum_{x=1}^{t} \lambda_x q_x^2}. \tag{10}$$

The stacked denoising autoencoder (SDAE) network is formed by cascading a single autoencoder layer by layer. Specifically, the middle hidden layer obtained from the training of the previous autoencoder participates in the training as the input of the next autoencoder. This stack in turn form a deep neural network structure, as shown in Figure 2.

The big data platform of commercial banks takes personal credit card data as the main data to construct the personal credit portrait. And the personal credit portrait will also concatenate personal business attributes and transaction data. Because personal information is scattered in various data sources, it needs to be integrated and spliced in order to facilitate the use and unified management of upper business logic. With individual customers as the granularity and credit dimension as the starting point, splice and integrate other business data of individual customers in the big data platform. Finally, it can form a complete view of personal credit based on bank big data. The following is a formal expression of the overall algorithm of personal credit risk assessment.

(1) Data preprocessing: first, extraction, integration, cleaning, and conversion of the original data are preprocessed. The feature vector $[i_1, i_2, \cdots, i_z]$ is formed from the traditional credit evaluation features. The intrabank business data of individual customers are preprocessed to form feature vectors $[i_{z+1}, i_{z+2}, \cdots, i_t]$. Two parts of features are fused to form the input features of the model; that is, $I = [i_1, i_2, \cdots, i_t]$ is used as the input data for model training
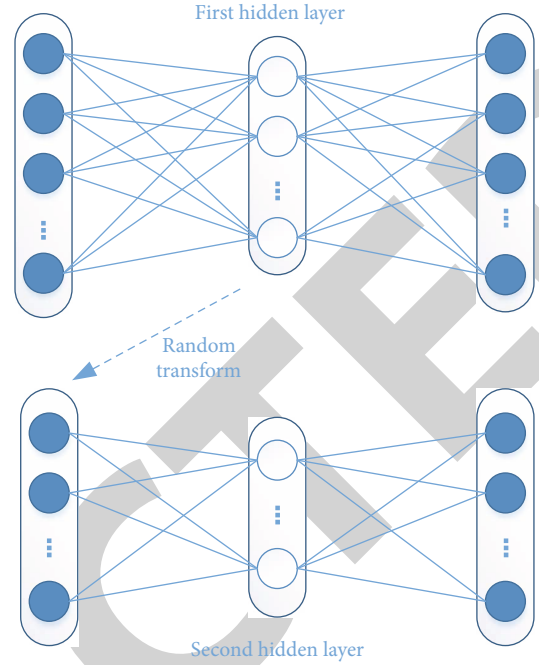

FIGURE 2: Stacked denoising autoencoder network.

(2) Model training: the input feature set is constructed based on big data, and an improved SDAE network is constructed

(i) Construct layer 1 network

We obtain the correlation matrix between features Σ_{II}.

The random disturbance transformation $v_f(i' \mid i)$ is carried out through decomposition to obtain the input term i' after corrosion. According to the denoising autoencoder calculation flow, the first hidden layer h_1 is obtained and changed into the first input layer.

(ii) Construct layer 2 to layer N networks

Obtain the correlation matrix between implicit features Σ_{hh}.

The hidden layer of the previous neural network is used as the input of the next neural network.

All hidden layers $h_i (i = 1, 2, \cdots, N)$ after the second layer are retained and constructed as a deep neural network structure.

(iii) Reverse tuning

SVM is used for final recognition and classification. BP is used to inversely optimize the network parameter weight w_i and bias b_i; that is, the gradient descent method is used to adjust $[w_i, b_i](1 \leq i \leq N)$.

3.2. Credit Decision Based on Multiobjective Particle Swarm Optimization.

Text multiobjective particle swarm optimization algorithm is mainly improved around the external file update strategy and speed parameter update strategy. The external file update strategy improves the spatial search ability of the algorithm by optimizing the spatial distribution

information of the solution. The speed parameter updating strategy mainly uses the evolution direction information of particles to realize the adaptive adjustment of parameters and balance the global exploration ability and local development ability of the algorithm.

3.2.1. Spatial Distribution Information of Optimization Solution.

Given a set of uniformly distributed direction vectors, $Q = \{q_1, q_2, \cdots, q_z\}$. The target domain O is uniformly decomposed into z subspaces $(\Omega_1, \Omega_2, \cdots, \Omega_z)$. The spatial distribution information of the optimal solution $s_x(n)$ is determined by the angle between its position vector and direction vector$<\boldsymbol{\lambda}(s_x(n)), q>$ decision, where $\boldsymbol{\lambda}(s_x(n)) = [\lambda_1(s_x(n)), \lambda_2(s_x(n)), \cdots, \lambda_W(s_x(n))]$ is the position vector of the optimal solution $s_x(n)$.

$$\lambda_w(s_x(n)) = \arccos\left(\frac{|h_w(s_x(n)) - r_w(n)|}{\sqrt{\sum_{y=1}^{W}\left(h_y(s_x(n) - r_y(n))^2\right)}}\right), \quad (11)$$

where $w = 1, 2, \cdots, W$. W is the number of objective functions. $R = [r_1, r_2, \cdots, r_w]$ is the reference point on the target space, $r_w(n) = \min(h_w(S(n)))$. The position vector satisfies the following conditions.

$$\sum_{w=1}^{W} \cos^2(\lambda_w(s_x(n))) = 1. \quad (12)$$

If the direction vector q_y satisfied $<\lambda(s_x(n)), q_z > \leqslant <\lambda(s_x(n)), q_y >$, then the xth optimization solution $s_x(n)$ belongs to the subspace Ω_Z determined by the direction vector q_z in the target space.

3.2.2. External File Update Strategy.

In order to ensure the search ability of the algorithm in the optimization space, an external file update strategy based on the optimal solution spatial distribution information in the external file is designed. The update strategy includes two parts: subspace solution set allocation process and subspace solution set selection process.

The angle between the newly generated optimization solution $s_x(n)$ and all given direction vectors is calculated. The subspace of the direction vector with the smallest included angle is selected as the attribution space. The home space of all optimal solutions is allocated. For any subspace α, the optimal solution set $e_\alpha(n)$ is as follows:

$$e_\alpha(n) = \{s_x(n) \mid \langle\lambda(s_x(n)), q_\alpha\rangle \\ \leq \left\langle\lambda(s_x(n)), q_y\right\rangle, s_x(n) \in S(n), x = 1, 2, \cdots, |S(n)|\}, \quad (13)$$

where q_y is any other direction vector, and $y = 1, 2, \cdots, Z$. In order to avoid the problem of poor search ability in algorithm space, this algorithm filters the optimal solution set in subspace. The upper limit of the number of

optimization solutions for each subspace is $\eta(n)$, where $\eta(n)$ is defined as below:

$$\eta(n) = \left\lfloor\frac{R}{B(n)}\right\rfloor, \quad (14)$$

where R is the default value of the given external file. $B(n)$ is the number of subspaces with optimal solutions in the current iterative process. When the algorithm does not fully explore the target space, $B(n)$ is small. At this time, more optimization solutions need to be reserved in the subspace to ensure that the algorithm can be fully explored in the target space. With the optimization process, $B(n)$ gradually increases. At this time, the subspace needs to retain the optimal solution with good convergence to ensure the convergence of the algorithm.

3.2.3. Adaptive Flight Parameter Adjustment Mechanism.

To enhance particle exploration, if particle $i_x(n)$ is dominated by $u_x(n)$, the value of $e_{x,1}$ needs to be lowered. At the same time, the values of ω_i and $e_{x,2}$ need to be increased. In order to enhance particle development capability, if particle $i_x(n)$ dominates $u_x(n)$, the value of $e_{x,1}$ needs to be increased. At the same time, lower the values of ω_i and $e_{x,2}$. Therefore, in order to adapt to the characteristics of particle search, a flight parameter adjustment mechanism based on the information of particle evolution direction is designed in this paper. The adaptive flight parameter $\delta_x = [\omega_x, e_{x,1}, e_{x,2}]$ adjustment mechanism of particle $i_x(n)$ is as follows:

$$\delta_x(n+1) = \begin{cases} \delta_x(n)\lambda_{x,1}(n), & u_x(n) > i_x(n) \\ \delta_x(n)\lambda_{x,2}(n), & u_x(n) < i_x(n) \\ \delta_x(n), & \text{otherwise} \end{cases}, \quad (15)$$

where $\lambda_{x,1}(n)$ and $\lambda_{x,2}(n)$ is the exploration parameter matrix and development parameter matrix of particle $i_x(n)$, respectively.

$$\lambda_{x,1}(n) = \begin{bmatrix} U_x(n) + 1 & 0 & 0 \\ 0 & U_x(n) & 0 \\ 0 & 0 & U_x(n) + 1 \end{bmatrix}, \quad (16)$$

$$\lambda_{x,2}(n) = \begin{bmatrix} U_x(n) & 0 & 0 \\ 0 & U_x(n) + 1 & 0 \\ 0 & 0 & U_x(n) \end{bmatrix}, \quad (17)$$

where $U_x(n)$ is the fine tuning parameter.

$$U_x(n) = \left(1 + g^{\left(\bar{F}(n) - F_x(n)\right)}\right)^{-1}, \quad (18)$$

where $F_x(n)$ is the Euclidean distance between particle $i_x(n)$ and global guidance point $a_x(n)$. $\bar{F}(n)$ is the average value of the distance between all particles and the guide point $a_x(n)$. When the particle $i_x(n)$ is close to $a_x(n)$, the

$U_x(n)$ value is low. At this time, the particles can be deeply developed in the $a_x(n)$ neighborhood. When the particle $i_x(n)$ is far away from $a_x(n)$, the $U_x(n)$ value is higher. At this time, particles can fully explore the optimization space of $a_x(n)$ neighborhood. The exploration parameter matrix and development parameter matrix will be updated with the change of particle harmonic parameters. The particle velocity is effectively adjusted according to the flight characteristics of particles.

3.2.4. Algorithm Steps. The calculation flow of adaptive decomposition multiobjective particle swarm optimization algorithm is as follows.

Step1. Randomly initialize the velocity $q_x(0)$, position $i_x(0)$, and inertia weight of each particle in the particle swarm $\omega_x(0)$ and learning factors $e_{x,1}(0)$ and $e_{x,2}(0)$. Set the initial position of each particle as the current historical optimal position $U_x(n)$.

Step2. Calculate the fitness value of each particle. The direction vector of the optimization solution is calculated from equation (11). The angle value between the optimization solution and the given direction vector is calculated from equation (13), and the attribution region of the optimization solution is determined.

Step3. Update the policy according to the external file. Select the appropriate optimization solution for the subspace solution set to update the external file.

Step4. Arbitrarily select the optimization solution from the external file as the gbest of the population, and update the pbest point of each particle.

Step5. Influences the evolution direction of each particle guidance point. The evolutionary harmonic coefficients are calculated from equations (16)–(18). The flight parameters of particles are adaptively adjusted by equation (15).

Step6. Update the position and velocity of particles according to equations (3) and (4).

Step7. Determines whether the algorithm meets the termination conditions. If so, it exits the loop and outputs the final optimization solution set. Otherwise, return to step 2.

In the implementation process, the external file update strategy based on the spatial distribution information of the optimized solution is used to ensure the distribution of the solution set of the algorithm. At the same time, combined with the evolution direction information of particles to adaptively update the flight parameters, the algorithm can obtain the optimal solution set with good distribution and convergence.

4. Experiment and Analysis

4.1. Credit Risk Assessment. The data of this paper comes from the big data platform of a commercial bank in China. In order to get a more comprehensive user portrait, the data required for modeling is centered on personal credit card data. Then, splice the customer's business data in the bank, including deposit, loan, financial management, guarantee, tripartite deposit, fund, debit card, and e-banking transaction. A total of 540000 customers were collected as modeling data, including 521361 nondefaulting customers.

Table 1: Effectiveness verification of DAE layers.

Model	Recall (%)	Precision (%)	MCC
SDAE3	83.5	83.9	0.736
SDAE4	84.3	85.7	0.759
SDAE5	84.8	86.4	0.773
SDAE6	84.4	85.3	0.766
SDAE7	83	83.8	0.745

The evaluation indexes commonly used in machine learning classification tasks are used to measure and evaluate the methods in this paper, such as recall, precision, accuracy, Matthews correlation coefficient (MCC) [18], F-score, and AUC-ROC [19]. Since the number of DAE layers of stacked denoising autoencoder network has an impact on the learning results of the model, this paper evaluates and verifies the models with different DAE layers, and the results are shown in Table 1.

Table 1 lists the different comparative experimental results when the number of DAE layers increases from 3 to 7. It can be seen from Table 1 that with the increase of DAE layers, the recall rate, the precision, and the MCC gradually increase. When the number of DAE layers increases to 5, the index values have reached the highest. This shows that the deep neural network is not the deeper the better but needs to be dynamically adjusted and selected according to the application scenarios and data of specific business. Therefore, in the subsequent experiments, the model with 5 DAE layers is used for related experiments.

In order to illustrate the advantages of the improved algorithm in this case study, the comparison results between the improved algorithm and other common algorithms in personal credit risk assessment are described in detail below. This algorithm is compared with traditional feature selection algorithm and advanced deep learning algorithm, where the traditional feature selection algorithms include paper [20] and paper [21]. Advanced deep learning algorithms include paper [16] and paper [17]. The experimental results of comparison are shown in Figure 3.

It can be seen from the results in Figure 3 that the results of the credit evaluation algorithm proposed in this paper are better than other traditional methods. At the same time, it is better than the original stacked denoising autoencoder network, and the accuracy is improved by about 3%. In particular, compared with the original feature set (Raw Feat without any feature selection), AUC-ROC, ACC, and F-score increased by 15%, 18%, and 21%, respectively. The above results illustrate this. In the big data scenario, this algorithm based on the deep learning framework can fully extract the potential essential characteristics of personal credit risk. It effectively compresses and embeds high-dimensional sparse features, so that the relationship between credit features can be expressed in low-dimensional space, so as to improve the final credit evaluation ability. The autoencoder network based on denoising stack improves the antinoise ability of the model when the data quality in big data environment is not high, so as to obtain better credit evaluation results.
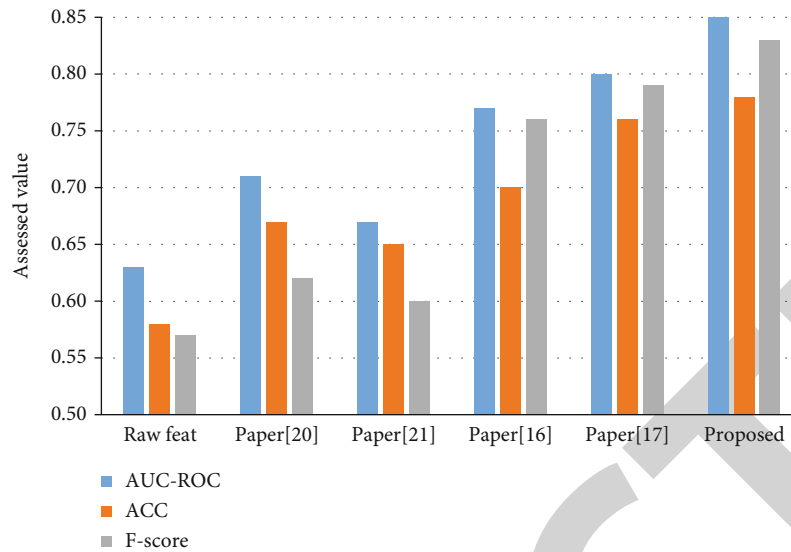
Figure 3: Comparison of evaluation results of different feature selection methods.

Table 2: Basic information table of loan portfolio options.

Project	Applied loan limit	NPV (good credit)	NPV (medium credit)	NPV (bad credit)	Average NPV
1	17.5	38.59	23.59	8.59	23.59
2	14	32.61	22.61	12.61	22.61
3	19.95	47.7	2.7	-12.3	12.7
4	15.75	48.095	38.095	23.095	36.43
5	28	51.225	6.225	-23.775	11.225
6	13.125	38.385	3.385	-10.615	10.385
7	31.5	73.38	63.38	8.38	48.38
8	10.5	39.835	19.835	4.835	21.505
9	12.25	41.045	15.045	9.045	21.71
10	5.6	19.88	13.88	-0.12	11.21

4.2. Loan Portfolio Decision. In order to better illustrate the operation effect of this algorithm in loan portfolio decision-making, the following two simulation examples are tested.

4.2.1. Simulation Example 1. The total loan amount of a bank's rural project is 1.5 million yuan. The minimum loan task required by the superior bank is 1.35 million yuan. At present, 10 farmer enterprises apply for loans, and the net present value (NPV) of the enterprises applying for loans under good, medium, and bad credit is shown in Table 2.

The multiobjective particle swarm optimization algorithm proposed in this paper is used to solve the simulation example 1. The population size of the algorithm and the maximum capacity of external files are set to 100. The initial inertia weight is 0.5, and the learning factor is set to 1. The maximum number of iterations for all test functions is set to 200. The algorithm runs independently for 30 times, and the results are taken as the average value. The operation result of one time and the convergence curve of objective function are shown in Figure 4. The results of 30 runs have reached the optimal value, and the average time is less than 1 s.

4.2.2. Simulation Example 2. In order to better illustrate that the algorithm proposed in this paper can be used to solve the optimization decision-making problem of large-scale loan portfolio, a simulation example 2 is designed in this paper. Simulation example 2 repeats the data of simulation example 1 10 times, so as to construct an example in which 100 farmer enterprises apply for loans. This not only does not affect the test of the algorithm in this paper but is more conducive to the test of the algorithm. Because the solution of the above problem requires exponential order time of the problem scale, for the simulation example 2 constructed according to this method, this can really test the effect of this algorithm in large-scale loan portfolio optimization decision-making problem. To sum up, this paper uses the above method to construct simulation example 2, and the specific data are as follows.

The total loan amount of a bank's rural project is 15 million yuan. The minimum loan task required by the superior bank is 13.5 million yuan. At present, 100 farmer enterprises apply for loans, and the net present value of the enterprises applying for loans under good, medium and poor operation
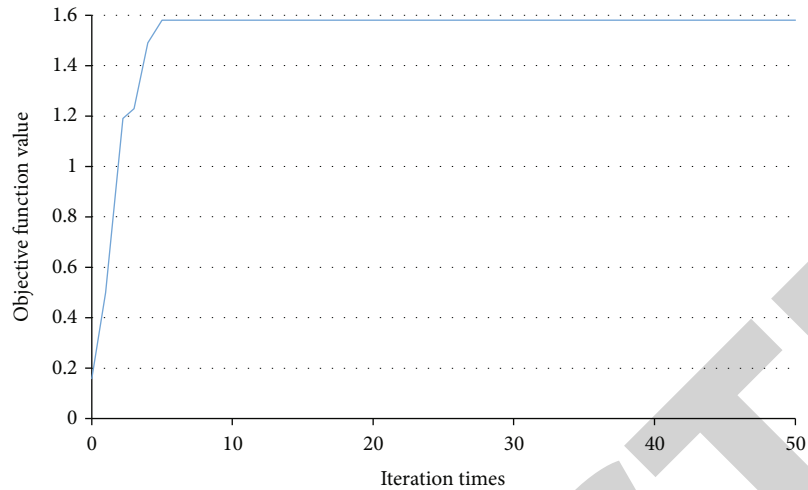
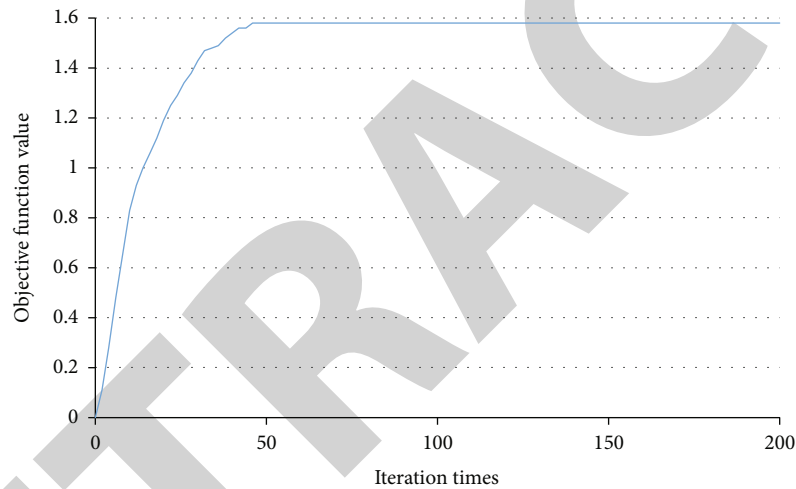Figure 4: Convergence curve of simulation example 1.

Figure 5: Convergence curve of simulation example 2.

is known. Here, the net present value, that is, repeat the data in Table 1 10 times. The population size of the algorithm and the maximum capacity of external files are set to 1000. The initial inertia weight is 0.5 and the learning factor is set to 1. The maximum number of iterations for all test functions is set to 200. The algorithm runs independently for 30 times, and the results are taken as the average value. The operation result of one time and the convergence curve of objective function are shown in Figure 5. The results of 30 runs have reached the optimal value, and the average time is less than 10 s. This algorithm finds the optimal solution in an average time of less than 10 s, which shows that this algorithm has achieved good results in solving speed and optimization ability. In 30 independent random experiments, the proposed algorithm has reached the optimal solution, which shows that the proposed algorithm has strong stability.

To sum up, the multiobjective particle swarm optimization algorithm proposed in this paper has achieved good results in optimization ability, solution speed, and stability in solving large-scale loan portfolio optimization decision-making problems.

5. Conclusion

Rural credit constraint is a serious problem facing China's rural economic development. China's rural credit market is imperfect, and the probability of a large number of farmers getting loans is very low. For solving these problems, this paper optimizes the rural credit system and loan decision-making. To solve the problem of data quality in big data applications, this paper improves the stacked denoising autoencoder depth network and improves the accuracy of credit evaluation. In order to improve the loan portfolio strategy, an adaptive decomposition multiobjective particle swarm optimization algorithm is proposed. Its main advantages include the following two aspects. (1) The external file update strategy based on spatial optimization can effectively balance the convergence and diversity of external files, so as to improve the spatial search ability of the algorithm. (2) The adaptive flight parameter adjustment mechanism can balance the exploration and development ability of the algorithm, so as to improve the convergence of the algorithm. Experimental results show that the proposed algorithm is

superior to other algorithms in speed and stability. Future work will add national policies, local characteristics, seasons, and other factors into the influence factors of the algorithm to make rural credit constraints more general.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

References

[1] L. Lin, W. Wang, C. Gan, and Q. T. T. Nguyen, "Credit constraints on farm household welfare in rural China: evidence from Fujian Province," *Sustainability*, vol. 11, no. 11, p. 3221, 2019.

[2] Z. Chen, K. Peng, and W. Yuan, "Usury, market power and poverty traps: a study of rural credit in 1930s' China," *Frontiers of Economics in China*, vol. 13, no. 3, pp. 369–396, 2018.

[3] L. Li and X. Wu, "Number of siblings, credit constraints, and entrepreneurship in China," *The Journal of Development Studies*, vol. 54, no. 7, pp. 1253–1273, 2018.

[4] Y. Nan, Y. Gao, and Q. Zhou, "Rural credit cooperatives' contribution to agricultural growth: evidence from China," *Agricultural Finance Review*, vol. 79, no. 1, pp. 119–135, 2019.

[5] G. Q. Chi, X. Z. Qin, and Z. Y. Zhu, "Decision-making model of Loan's portfolio optimization based on principle of maximum earnings per risk," *Control and decision*, vol. 15, no. 4, p. 4, 2000.

[6] N. Metawa, M. K. Hassan, and M. Elhoseny, "Genetic algorithm based model for optimizing bank lending decisions," *Expert Systems with Applications*, vol. 80, pp. 75–82, 2017.

[7] W. Zhiheng and L. Jianhua, "Flamingo search algorithm: a new swarm intelligence optimization algorithm," *IEEE Access*, vol. 9, pp. 88564–88582, 2021.

[8] M. A. M. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 621–630, 2021.

[9] R. F. Mansour, "Evolutionary computing enriched ridge regression model for craniofacial reconstruction," *Multimedia Tools and Applications*, vol. 79, no. 31-32, pp. 22065–22082, 2020.

[10] J. Zhang, X. Hu, Z. Ning et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2633–2645, 2018.

[11] H. Xiping, J. Cheng, M. Zhou et al., "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.

[12] N. K. Jain, U. Nangia, and J. Jain, "A review of particle swarm optimization," *Journal of The Institution of Engineers (India): Series B*, vol. 99, no. 4, pp. 407–411, 2018.

[13] D. Zouache, A. Moussaoui, and F. B. Abdelaziz, "A cooperative swarm intelligence algorithm for multi-objective discrete optimization with application to the knapsack problem," *European Journal of Operational Research*, vol. 264, no. 1, pp. 74–88, 2018.

[14] Y. Zhang, D. Wang, Y. Chen, H. Shang, and Q. Tian, *Credit risk assessment based on long short-term memory model [C]// International Conference on Intelligent Computing*, Springer, Cham, 2017.

[15] G. G. Wang, X. Cai, Z. Cui, G. Min, and J. Chen, "High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 20–30, 2017.

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[17] S. Nurmaini, A. Darmawahyuni, A. N. Sakti Mukti, M. N. Rachmatullah, F. Firdaus, and B. Tutuko, "Deep learning-based stacked denoising and autoencoder for ECG heartbeat classification," *Electronics*, vol. 9, no. 1, p. 135, 2020.

[18] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[19] A. Davis, S. Gill, R. Wong, and S. Tayeb, *Feature selection for deep neural networks in cyber security applications [C]//2020 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS)*, IEEE, 2020.

[20] A. Likas, N. Vlassis, and J. J. Verbeek, "The global _k_ -means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[21] Q. Xu, C. Ding, J. Liu, and B. Luo, "PCA-guided search for _K_ -means," *Pattern Recognition Letters*, vol. 54, pp. 50–55, 2015.