Computing and Informatics, Vol. 41, 2022, 1078–1113, doi: 10.31577/cai_2022_4_1078

RESEARCH ON UBI AUTO INSURANCE PRICING MODEL BASED ON PARAMETER ADAPTIVE SAPSO OPTIMAL FUZZY CONTROLLER

Xindong Wang, Wei Liu*

College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao Shandong 266590, China
e-mail: liuwei_doctor@yeah.net

Zhaochuang Ou

Qingdao Haishi Baking Electric Corporation, Qingdao
Shandong 266111, China

Chun Yan

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao
Shandong 266590, China

Man Qi

School of Engineering, Technology and Design
Canterbury Christ Church University
CT1 1QU, UK

Wenli Jia

Shandong University of Science and Technology, Qingdao
Shandong 266590, China

---

* Corresponding author

Abstract. Aiming at the problem of "dynamic" accurate determination of rates in UBI auto insurance pricing, this paper proposes a UBI auto insurance pricing model based on fuzzy controller and optimizes it with a parameter adaptive SASPO. On the basis of the SASPO algorithm, the movement direction of the particles can be mutated and the direction can be dynamically controlled, the inertia weight value is given by the distance between the particle and the global optimal particle, and the learning factor is calculated according to the change of the fitness value, which realizes the parameter in the running process. Effective self-adjustment. A five-dimensional fuzzy controller is constructed by selecting the monthly driving mileage, the number of violations, and the driving time at night in the UBI auto insurance data. The weights are used to form fuzzy rules, and a variety of algorithms are used to optimize the membership function and fuzzy rules and compare them. The research results show that, compared with other algorithms, the parameter adaptive SAPAO algorithm can calculate more reasonable, accurate and high-quality fuzzy rules and membership functions when processing UBI auto insurance data. The accuracy and robustness of UBI auto insurance rate determination can realize dynamic and accurate determination of UBI auto insurance rates.

Keywords: Parameter adaptive SAPSO, fuzzy controller, fuzzy rule, membership function

1 INTRODUCTION

In recent years, China has become the world's second largest insurance market. According to the China 2019 statistical yearbook, the total revenue of property insurance in China was 1 077 billion yuan in 2018. The premium of auto insurance is 783.4 billion yuan, accounting for 72.7 % of the total premium of property insurance, which is still the largest type of property insurance in China. However, according to the 2019 annual reports of insurance companies, only 14 of 58 auto insurance companies are profitable, and more than 90 % of property and casualty insurance companies underwrite losses. China's auto insurance market as a whole showed a comprehensive profit loss. In addition to the sharp increase in claims, the reasonable pricing of auto insurance products and the unsatisfied consumer demand are also important reasons of the phenomenon.

Faced with the above problems, the key to solve is to find a new pricing model to accurately price according to the customer's risk level. In recent years, the independent pricing model of usage-based insurance UBI in foreign countries has gained huge benefits for foreign insurance companies. It has been put into the market for many years and has been widely used in European and American countries. For the definition of UBI, there are usually two interpretations. One is Usage Based Insurance, which is insurance based on usage and represents insurance based on driving mileage. The other is User Behavior Insurance, which is an insurance designed according to driver behavior. In contrast, the current interpretation of Usage-Based

Insurance is more widely used. Although the two explanations are completely different, they are essentially the same. They both upload vehicle-related information to a remote server, and then the insurance company redesigns the product based on a large amount of data. Obviously, in the two definitions of UBI, the model of designing insurance based on driving behavior is superior. The UBI pricing model in this article is constructed following the concept of charging premiums based on driving behavior.

The first part of this paper introduces the background and significance of auto insurance pricing model. The second part summarizes some literatures about UBI auto insurance and particle swarm optimization, and leads to the SAPSO optimization fuzzy controller algorithm. The third part describes the principle of fuzzy controller, parameter adaptive PSO and parameter adaptive SASPO algorithm, and explains how particle swarm optimization algorithm optimizes fuzzy rules and membership functions. In the fourth part, the optimized fuzzy controller based on parameter adaptive SASPO algorithm is empirically studied and the results are analyzed. The last part is the conclusion.

2 LITERATURE REVIEW

2.1 UBI Auto Insurance Domestic and Foreign Literature

Research abroad on UBI has been relatively mature. Arbabzadeh and Jafari constructed a real-time driver risk prediction algorithm through multiple logistic regression model, proving that this algorithm can be applied to driver assistance systems and help reduce the risk of accidents [1]. Ma et al. made risk measurements based on the same situation based on GPS data, compared the driving behaviors of drivers on the same road segment, and found indicators such as hard braking that are highly correlated with accident rate [2]. Ayuso et al. proposed to use remote technology to record drivers' behaviors and driving mileage. Based on the frequency-based regression model with counting data to calculate the premium or compensation, the calculation of auto insurance rate is improved [3]. Yan et al. proposed to build a fuzzy controller to calculate the autonomous underwriting coefficient based on the driver's driving mileage and the number of violations, so as to realize the "dynamic" pricing of premiums [4]. After decades of development, foreign UBI theories have formed a relatively complete system. It has abundant research achievements in dynamic determination of auto insurance premium, feedback of bad driving behavior and prediction of real-time risk.

Foreign UBI studies have pointed out the direction for the development of auto insurance in China. Although the foreign UBI model is not suitable for "copying" to China, Chinese scholars can absorb the theoretical research results of foreign UBI auto insurance and conduct research on UBI in combination with China's national conditions. In the aspect of rate determination, scholars have tried many methods and models to quantify risk factors. Zhu proposed a driving behavior rating model based on UBI. The index system was established and the weight was determined by entropy weight-analytic hierarchy process, and it quantifies driving behavior to assess risk [5]. On the basis of UBI, Gao proposed to use classification technology of data mining to classify each customer's driving behavior and score each classification result, and the result was more ideal than the driving behavior scoring model [6]. Li and Meng proposed to construct a reward and punishment system for auto insurance, in which a generalized linear model is used to determine the prior fee rate based on the characteristic information of the policy. Then the prior fee rate was adjusted based on historical claim information, which improved the accuracy and rationality of the rate determination result [7]. Zhang and Wang established the zero-expansion negative binomial model and the gamma model for the claim frequency and the claim in tensity respectively under the condition of large claims, which showed that the auto insurance rate is relatively accurate [8]. Some scholars proposed to combine data collection equipment. On the basis of a large amount of data support, the risk factors are extracted, the risks are quantified objectively, and differentiated pricing is provided. Peng et al. pointed out the establishment of an intelligent UBI system based on 3C (pipe-cloud-end) architecture. For driving behavior data collection, transmission, processing, analysis and mining, the differentiation of premium is realized through system [9]. Wang extracted major risk factors from a large amount of original data in their study of UBI pricing model. The Poisson regression model was used to construct the vehicle accident frequency prediction [10]. Pang analyzed the driving behavior of users according to the data records, and then comprehensively scored the driving behavior. Through multiple data record scores, the driving behavior was quantified, combined with big data, and a differentiated pricing strategy was implemented to conduct research [11]. Wang and Kuang proposed that with the growth of car ownership in China, the demand of the auto insurance industry has become more and more diversified, and UBI auto insurance as an innovative auto insurance has received attention from the industry [12]. Ji proposed that traditional auto insurance generally follows a "car-based" auto insurance system when determining insurance rates, that is, it mainly considers some factors of the car itself, such as the model, the nature of use of the vehicle and the price, etc. The driver factors are only used as an auxiliary reference, such as the driver's driving behavior, driving experience and violation records. The UBI auto insurance is based on the driver's driving behavior as the main reference, and follows the "humanistic" auto insurance system [13]. The research on UBI in China is not mature enough. At present, the development of Internet insurance in China has entered the substantial transformation stage, and some large insurance companies have also developed UBI products. However, compared with foreign countries, there is still a big gap. Scholars need to learn from the mature theories of foreign countries and continuously practice various pricing models to explore "localized" UBI products.

UBI auto insurance is calculated by quantifying driving risk. Therefore, there are many and complicated factors influencing the independent pricing of auto insurance premiums. Including driving time, driving environment, driving mode and many other aspects, it is difficult to express with a specific mathematical model, dynamic characteristics is difficult to grasp. However, fuzzy controller does not need specific mathematical model to express. The presupposition results under the combined action of all factors can be obtained by reasoning calculation of the influencing factors through fuzzy controller. Therefore, a fuzzy controller is proposed to describe the auto insurance problem. Combined with the current premium calculation formula in China, the risk factor is input into the fuzzy controller for reasoning calculation to calculate the dynamic autonomous underwriting coefficient. However, the fuzzy rules and membership functions of the fuzzy controller can be set according to the expert experience. Expert experience has cognitive limitations, so the optimization algorithm needs to adjust membership functions and fuzzy rules more precisely and objectively through data.

2.2 Related Algorithm Literature

Due to too many parameters of fuzzy rules, a powerful swarm optimization algorithm is needed. PSO algorithm is a swarm intelligence optimization algorithm and optimization performance has been successfully applied in many fields. PSO algorithm is a mature optimization algorithm with strong optimization performance, wide applicability and simple application. Garg proposed a PSO-GA hybrid algorithm that uses genetic algorithms to cross and mutate decision variables in PSO, further improving the balance between exploration and development capabilities [14]. Olivas et al. proposed a parameter adaptive particle swarm algorithm, which uses interval-type-2 fuzzy logic modules to improve the convergence and diversity of particle swarms [15]. Durand and Abrão proposed a multi-beam satellite (MBS) communication power distribution method based on heuristic particle swarm optimization, which considered the distribution of satellite communication power under different sky conditions and improved the overall energy of the satellite system Efficiency [16]. Valdez et al. proposed a fuzzy inference system to dynamically control inertia weights and learning factors, which enhances the performance of the algorithm [17]. Pan et al. proposed the PSO based on coevolution. In the antenna synthesis problem, decision variables were assigned to subgroups. Then, the probability based learning strategy is optimized by PSO to maintain the diversity of the population and accelerate the convergence speed [18]. Lagunes et al. adopt a variety of meta-heuristic algorithms to test algorithms through competitive methods until they find the best way to solve the problem. In the model check, the benchmark mathematical function and membership function of the fuzzy controller of the autonomous mobile robot are optimized [19]. There are many improvements to the particle swarm algorithm [20, 21, 22], and the pursuit of particle swarm parameter adaptation is one of the main directions of improvement. In addition, fuzzy controller and PSO can be well combined to get better results. Meng et al. proposed an adaptive fuzzy controller based on improved PSO, which realized the operation navigation of adaptive control of agricultural vehicles. Compared with conventional fuzzy control, the navigation accuracy of the improved fuzzy control algorithm is significantly improved [23]. Wang et al. adopted a speed control algorithm based on improved particle swarm optimization to optimize the fuzzy controller, and we solved the problems such as system overharmony and poor stability of the PID controller. In the experiment, the table showed good robustness and control accuracy [24]. Zhang et al. proposed a new adaptive simulated annealing particle swarm algorithm, and then used the algorithm to optimize the fuzzy rule weights and quantized scale factors of the fuzzy controller. The simulation results show that the controller enables the brushless DC motor speed control system to have good rapidity, stability and robustness [25]. Tian and Liang proposed a cuckoo search algorithm based on mini-batch gradient descent, and obtained good algorithm performance [26]. Wang et al. proposed a feature fusion sequence annotation model based on attention mechanism [27]. Xu et al. used residual attention-based network to identify potato leaf diseases [28]. In order to quickly and accurately identify various geometric errors, Yu and Sun proposed a hybrid SAPSO-GA algorithm. The compensation effect is analyzed by comparing the motion trajectory before and after the compensation with the ballbar. The results show that the proposed method improves the identification accuracy, and greatly improves the machining accuracy of the crankshaft follower grinder through compensation [29]. The simulation results of Zhang and Huang show that the SAPSO-RBF algorithm has good self-organizing ability, and compared with other self-organizing RBF neural network optimization algorithms, the network structure compactness and accuracy have been greatly improved [30]. PSO combined with fuzzy controller has good complementarity. Fuzzy controller can describe complex problems through language rules, calculated according to the designers intend to reasoning. But the adaptive ability is poor, there are man-made factors, the result accuracy is not high. And powerful optimizing ability of PSO can reduce subjective conceptions of fuzzy controller, with global optimization ability, and found the result of the high precision in the numerous possible. PSO algorithm combined with fuzzy controller can be complementary, supplement each other, get ideal results.

To sum up, the domestic scholars have adopted many methods to study the reform of the car insurance rates and introducing UBI car insurance, through three fee changes to make the car insurance industry pricing power gradually expanding, promoted some car insurance companies insurance rates. But domestic UBI research is still in the attempt stage, large scale implementation needs a lot of theoretical research and practice. However, there are few studies on the application of fuzzy controller in the field of UBI auto insurance, so we can try to describe the problem of rate coefficient determination in the auto insurance pricing model through fuzzy controller. In addition, the combination of PSO algorithm and fuzzy controller has achieved good results in engineering control and other fields. After optimizing the membership function and fuzzy rules of the fuzzy controller, the rationality of the fuzzy controller can be further improved. Therefore, on the basis of SAPSO algorithm, this paper further improves and proposes a parametric adaptive SAPSO algorithm. The inertial weights and learning factors are determined in real time according to particle optimization in the iteration to improve the convergence speed and optimization performance of the algorithm. The direction of motion is adjusted in real time to avoid falling into the local optimal solution. The UBI auto insurance pricing model based on parametric adaptive SAPSO optimized fuzzy controller was proposed. The algorithm model was put into practice on the basis of data verification. In comparison with analytic hierarchy process (AHP), entropy weight (EW) method, genetic algorithm (GA) and parametric adaptive PSO, it was found that the algorithm had higher accuracy and stability. The numerical results show that the algorithm is an effective algorithm to optimize fuzzy rules and membership functions, and applications in UBI pricing model is also a kind of new exploration and practice in the study.

3 ARITHMETIC STATEMENT

3.1 Fuzzy Controller

The calculation of self-underwriting coefficient is the key to accurate pricing. The self-underwriting coefficient is closely related to the driving behavior of customers, and the value of the coefficient represents the quantification of the risk level of customers' driving behavior. There are many factors influencing the pricing, and there are correlations among the factors, so it is difficult to calculate the comprehensive influence of various factors with specific mathematical models. While the fuzzy controller can calculate the comprehensive influence according to the input factor without describing the specific mathematical model, and has strong control over the uncertain object and strong robustness to the changing parameters. It is a new attempt to select the autonomous underwriting coefficient of fuzzy controller to calculate the rate coefficient.

Fuzzy control is based on the production experience of operators and developed by modern control theory. According to the practical experience of operators, a qualitative and imprecise control rule is obtained by using language for reasonable summary and description. Then it is quantified by the method of fuzzy mathematics, which is transformed into fuzzy control algorithm, and finally the fuzzy control theory is formed. Fuzzy control has many advantages: high reliability, continuous output and can give full play to the effect of automation. But at the same time, there is also the disadvantage of subjectivity, which becomes more complicated and less accurate when there are more parameters.

3.1.1 Composition and Structure of Fuzzy Controller

The composition of the fuzzy controller mainly includes three parts: the fuzzification of the input variables, the fuzzy reasoning of the rule base, and the defuzzification of the output variables. The composition block diagram and calculation process are shown in Figure 1. First, the exact value of the input variable is blurred, the membership function and the rule base are set, and then fuzzy reasoning is performed according to the rule base. Finally, defuzzification is performed to obtain the exact value, that is, the output variable.

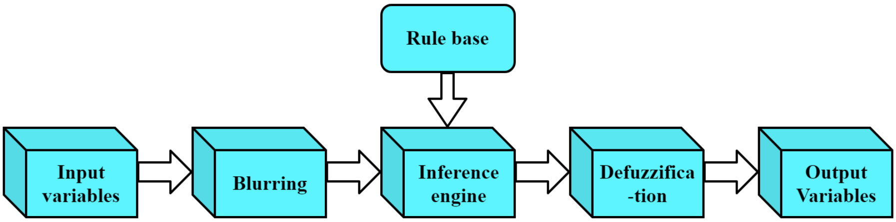

Figure 1. The block diagram of fuzzy controller

In the fuzzy toolbox of MATLAB, there are two inference algorithms, Sugeno and Mamdani. Although the Sugeno algorithm is more compact and comparatively more efficient than the Mamdani algorithm, it fits better with optimization and adaptive techniques. But based on the limitation of the algorithm in this paper, Sugeno inference algorithm cannot be fused with the algorithm. Therefore, Mamdani inference algorithm is selected to construct the fuzzy controller. In addition, Mamdani algorithm has been widely accepted by the world, and the reasoning process is more intuitive, which is very suitable for human input and has wider adaptability.

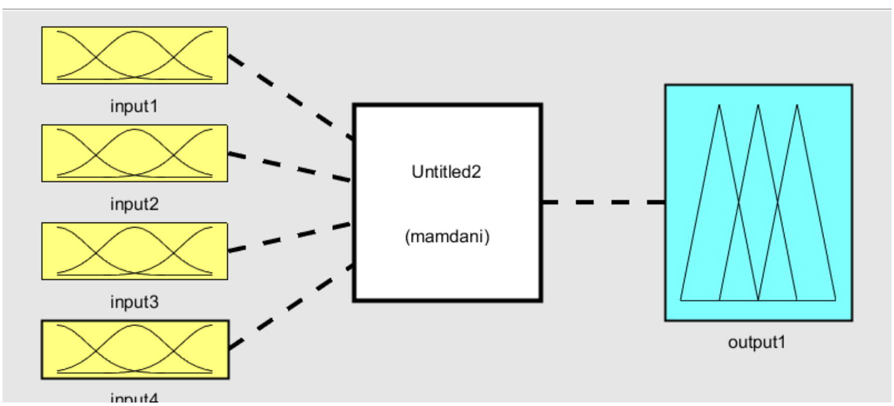

Figure 2. Structure diagram of Mamdani algorithm

According to the number of input and output variables, the structure of the fuzzy controller can be divided into three types. They are SISO (single input single output), MISO (multiple input single output), and MIMO (multiple input multiple output). The structure diagram of the Mamdani algorithm in Figure 2 is an example of a MISO structure. In Figure 2, input variables are on the left, inference algorithm is in the middle, and output variables are on the right. Obviously, the fuzzy controller in the figure is a MISO structure with four inputs and one output. The structure of the fuzzy controller is determined according to the input and output variables of the actual problem. In addition, as the input variables in the fuzzy controller increase, the subjective arbitrariness becomes larger. Setting fuzzy rules is also more difficult than ever, and the algorithm is cumbersome and not precise enough, so the input variables of the fuzzy controller should be appropriately selected.

3.1.2 Fuzzification of Input Variables

Input fuzzification is the first step to realize fuzzy control. Fuzzification is the process of converting the exact value of input into the corresponding value of fuzzy language variable, which is a fuzzy set. For example, speed can be divided into "fast", "normal", "slow" and other fuzzy sets according to the size of the speed. Therefore, the fuzzification method is the transformation method from precise quantity to fuzzy set. The common fuzzy methods include the classification fuzzy set method, the membership degree of the input point is 1, the single-point fuzzy set method and the membership degree value method. Because the classification fuzzy set method is simple and practical, it can be used to divide the fuzzy level directly according to the numerical value. Therefore, this paper adopts the classification fuzzy set method, according to the meaning of each variable, according to the value of several fuzzy sets. Each fuzzy set represents a certain degree of the variable, which can be represented by letters VS (very short), S (short), L (long), etc.

In 1965, the American professor L. A. Zadeh first published a paper entitled Fuzzy Sets [31]. It pointed out that if there is a number $A(x) \in [0, 1]$ corresponding to any element x in the field U, then A is called the fuzzy set on U, and $A(x)$ is called the membership of x to A. When x varies in U, $A(x)$ is a function, called a membership function of A. The closer the membership of $A(x)$ is to 1, the more x belongs to A, and the closer $A(x)$ is to 0, the less x belongs to A. The degree of membership is expressed by the membership function of the interval $[0, 1]$. Therefore, the selection of membership function is an important step in the process of fuzzification. Each fuzzy set needs to be expressed by membership function and converted into membership value. The common methods to determine membership functions include intuitive methods, fuzzy statistics, fuzzy distribution, binary comparison ordering and direct grading by experts and scholars. Figure 3 shows 5 fuzzy sets divided by hierarchical fuzzy set method, namely [VS, S, M, L, VL]. The Y-axis in the figure is the membership degree, and the range is $[0, 1]$. The X-axis is the argument field of variables $[0, 8]$. According to the above principle of maximum membership, it can be seen from the figure that element 2 is within the scope of the three fuzzy sets of VS, S and M, but the membership degree in S is the highest, so element 2 belongs to the fuzzy set of S.

Membership functions can be continuous or discrete, continuous membership functions are more accurate, discrete membership functions are more intuitive. Figure 3 shows four membership functions, among which, the membership functions of VS fuzzy sets are ZMF, S and L fuzzy sets are triangular, M fuzzy sets are Gaussian, and VL fuzzy sets are SMF. SMF and ZMF only need to determine the starting and

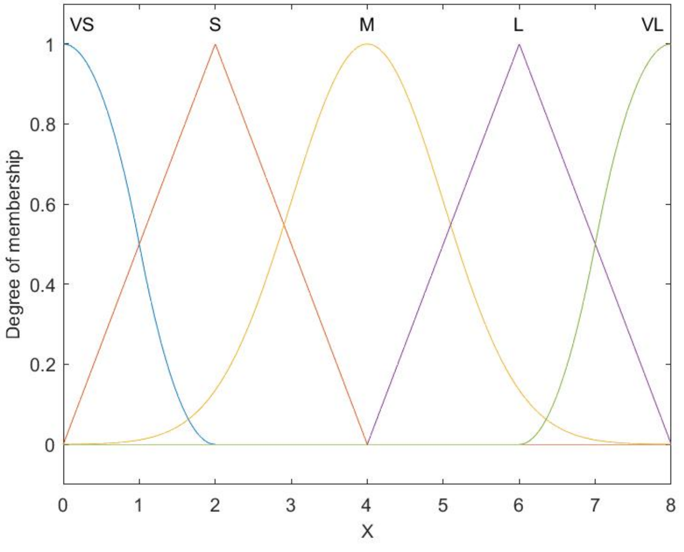

Figure 3. Example of membership function diagram

ending points of the fuzzy set interval. The two common membership functions are Triangular and Gaussian. The membership function of the triangle can be determined by the three parameters a, b and c, and the shape is determined by its slope. Its distribution is shown in Equation (1).

$$f(x) = \begin{cases} x - a/b - a, & \text{if } a < x < b, \\ x - c/b - c, & \text{if } b < x < c. \end{cases} \tag{1}$$

The Gaussian membership function can be determined by c and σ parameters. Since the Gaussian function is continuous and differentiable, its distribution is shown in Equation (2).

$$f(x) = \exp\left[-(x-c)^2/\sigma^2\right]. \tag{2}$$

In the input fuzzification, for example, $[a, c]$ in the triangular membership function is the range of each fuzzy set, while a, b and c determine the distribution of membership function. Therefore, the selection of membership function parameters is the most important. However, when setting parameters, they are generally selected subjectively based on experience, which is not reasonably enough.

3.1.3 Fuzzy Rules

According to the different arrangement and combination of fuzzy language variables of fuzzy controller, a large number of fuzzy conditional statements can be constituted, which is what we call fuzzy control rules. Fuzzy rules are generally established based on the experience of experts. The control rules contained in the fuzzy rule base are the embodiment of the transition from the actual control experience to the fuzzy controller, and also reflect the thinking mode of manual control. Fuzzy rules are designed by people subjectively, and the design needs to be tested and compared repeatedly to reduce the subjective factors as much as possible. For example, a two-dimensional fuzzy controller has the following rules.

$$R: \text{ if } X_1 \text{ is } A, \text{ and } X_2 \text{ is } B, \text{ then } Y \text{ is } C$$

where X_1 and X_2 are input variables, Y is output variable, A, B and C are fuzzy sets, and R stands for a rule, which means that when the value of X_1 belongs to set A and the value of X_2 belongs to set B, the value of Y belongs to set C. Only by setting the appropriate control rules the control strategy in the operator's mind can be better reflected, and the satisfactory control effect can be achieved. But the fuzzy rule is artificial, there is an error. Therefore, PSO is used to optimize fuzzy rules in the following paper, and the fuzzy rule combinations with the highest correlation with the historical number of accidents are screened out from the randomly generated rule library. This does not only minimize the influence of human factors, but it also makes the fuzzy rules of screening more reasonable.

3.1.4 Defuzzification

Fuzzy reasoning results are generally fuzzy values. The process of converting a fuzzy value to an exact value is called defuzzification. The most common methods are maximum membership, center of gravity, median and weighted average. The center-of-mass method is more sensitive than other methods, and small changes in the input variables can be reflected in the output variables. Therefore, the method of center of gravity is selected to solve fuzziness. The barycenter method is to take the barycenter of the area enclosed by the curve of membership function and abscissa as the final output value of fuzzy reasoning. Its formula is shown in Equation (3).

$$v_0 = \frac{\int_V v u_v \, \mathrm{d}v}{\int_V u_v \, \mathrm{d}v} \tag{3}$$

where v is the abscissa, $u_v(v)$ stands for membership function, and $\int_V u_v \, \mathrm{d}v$ represents the area bounded by the membership function and the abscissa. The process of using the fuzzy controller to calculate the independent underwriting factor is as described above. After inputting the pricing factor, after fuzzification, fuzzy reasoning, and defuzzification, the result obtained is the independent underwriting factor to calculate the premium.

3.2 Description of Particle Swarm Algorithm

3.2.1 Parameter Adaptive Particle Swarm Algorithm

Particle Swarm Optimization is an intelligent algorithm that simulates the predation of birds. It was proposed by Kennedy and Eberhart in 1995 [32]. Today, after more than a decade of development, PSO theory has become one of the important topics discussed at the international conference on evolutionary computing and has been widely applied in many fields.

The mathematical description of the standard PSO algorithm is as follows: set the search space as D dimension, the population size as N, the position and velocity of particle i are the vectors $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ and $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})$, respectively. The best position *gbest* experienced by the i^{th} particles $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$, that is, the individual optimal value. The best position experienced by the whole particle swarm is $G = (G_1, G_2, \dots, G_D)$, that is, the global optimal value.

The formula for updating the position and velocity of particle i at $t + 1$ time is shown in Equations (4) and (5).

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1}, \tag{4}$$

$$V_{id}^{t+1} = \omega V_{id}^{t} + c_1(P_{id}^{t} - X_{id}^{t}) + c_2(G_{d}^{t} - X_{id}^{t}). \tag{5}$$

In the formula, c_1 and c_2 are the learning factors of the particles, and ω is the inertia weight. For inertia weight coefficient, the appropriate value can effectively balance the global and local optimization ability. The linear decline formula of inertial weight can make the algorithm search a wider range at the initial stage, and then quickly gather in the optimal solution region at the later stage, which can improve the optimization accuracy. After a series of experiments on test functions, Eberhart and Shi [33] came to the conclusion that when the strategy of linear decline of inertial weight from 0.9 to 0.4 was adopted, a good optimization effect could always be obtained. The linear decline formula of inertial weight is shown in Equation (6) [22].

$$\omega = \omega_s - \frac{(\omega_s - \omega_e)}{t_{max}} \tag{6}$$

where ω_s and ω_e are the initial and final inertial weights of ω, t is the number of iterations, and t_{max} is the maximum number of iterations. For the learning factor c_1 and c_2, namely the acceleration coefficient, its value also has a great influence on the performance of the algorithm. Ratnaweera et al. [34] studied the acceleration coefficient and proposed a time-varying acceleration coefficient, as shown in Equations (7) and (8). This strategy enables the algorithm to focus on learning its own *gbest*. This strategy enables the algorithm to focus on learning its own [23].

$$c_1(t) = c_{1f} + (c_{1s} - c_{1f})t/t_{max}, \tag{7}$$

$$c_2(t) = c_{2f} + (c_{2s} - c_{2f})t/t_{max} \tag{8}$$

where, c_{1f}, c_{1s}, c_{2f} and c_{2s} are fixed values, and the acceleration coefficient changes with the increase of iteration times. Experimental results show that the algorithm achieves good performance when c_1 decreases from 2.5 to 0.5 and c_2 increases from 0.5 to 2.5.

The basic flow of PSO is described below.

Step 1. Particle swarm parameter initialization, particle swarm particle position and velocity initialization.

Step 2. The fitness values of all the initialized particles in the population are calculated according to the fitness formula. The positions and fitness values of the currently initialized particles are stored in *gbest* (individual optimum). Stores the position and fitness value of the particle with the best fitness value among all initialized particles in *zbest* (global optimum).

Step 3. Update the parameters, update the inertia weight according to formula (6), update the velocity and position of each particle according to formulas (4) and (5); update learning factors by formulas (7) and (8).

Step 4. Calculate the adaptive value of each particle in the new population.

Step 5. Compare the fitness of each particle in the population with *gbest* fitness. If the current particle fitness value is better, update *zbest* with the particle's current position and fitness value.

Step 6. If the termination condition is met and the maximum number of iterations is reached, output *zbest* and terminate the algorithm; otherwise, jump back to Step 3.

3.2.2 Parameter Adaptive Simulated Annealing Particle Swarm Optimization Algorithm

The basic idea of simulated annealing algorithm (SA) was first proposed by N. Metropolis in 1953, which inspired the principle of solid annealing. The basic idea is that the objective function value is the internal energy E, and the solution of each solution space has a corresponding internal energy value E. A random perturbation to the current solution in each iteration is applied. Metropolis acceptance criterion is used to decide whether $e^{-\Delta E/T}$ accepts the new solution by calculating whether it is larger than ε (ε is $[0,1]$ random number). In the particle position update process, simulated annealing and Metropolis criteria were introduced to form a new adaptive simulated annealing particle swarm optimization (SAPSO) algorithm. The algorithm uses the Metropolis criterion to determine whether it accepts the new solution during the location update process, so that each particle adds a judgment process after each move. The algorithm keeps the particle population a certain diversity, enhances the ability to jump out of the local optimal solution, and improves the global search ability of the algorithm.

The Metropolis criterion is the core of the simulated annealing algorithm. In the iterative process, it is necessary to judge whether to accept the transition from the current solution to x the new solution x' according to the probability. The calculation formula of P_T is shown in formula (9):

$$P_T = \exp\left(\frac{(f(x') - f(x))}{T}\right) \tag{9}$$

where, x and x' represent the old solution and the new solution when the control parameter is T, respectively; $f(x)$ and $f(x')$ represent the non-negative objective function of x and x', respectively. The control parameter T will decrease with the number of iterations, so that the probability that the algorithm accepts the deteriorated solution is continuously reduced and finally converges to the global optimal solution.

For the above-mentioned parameter adaptive standard particle swarm optimization algorithm, its inertia weight and learning factor are closely related to the number of iterations, and there is a regular linear change as the number of iterations changes. However, the number of iterations is only a judgment condition, and cannot provide feedback on the specific situation of the particle swarm optimization process. For this reason, on the basis of the SAPSO algorithm, the parameters are adjusted purposefully, effectively and dynamically according to the distance between particles and the changes in fitness values, so as to achieve the optimal solution faster.

1) **Variation in the direction of movement.** The direction of motion refers to the direction of the particle velocity. As shown in the updated formula (4), the new particle position is the old particle position plus the moving speed. Movement direction variation means that the new particle position can be added or subtracted from the old particle position, and the movement direction of the particle swarm can be controllably changed according to the diversity of the particle swarm. The formula is shown in Equation (10).

$$X_{id}^{i+1} = X_{id}^{t} + (-1)^g * V_{id}^{t+1}. \tag{10}$$

$(-1)^g$ in formula (10) is a directional coefficient of particle velocity, which is used to change the direction of particle movement. The g is an integer, and the odd and even numbers are changed according to the population diversity of the particles and the distance of the particles. Whenever the particle swarm reaches a local optimal solution, the direction coefficient will change the parity. The particle swarm will move in the opposite direction and find another local optimal solution globally, which enhances the search capability of the algorithm while ensuring the quality of the optimized solution.

2) **Adaptive inertial weight.** Using the distance between each particle and the optimal particle to adjust the inertia weight can effectively quickly gather and maintain the diversity of the population, and improve the optimization

ability of the algorithm. The distance formula of particles is shown in Equation (14). The positions of particle i and particle j and D dimensional vectors $X_i = (X_{i1}, X_{i2}, \ldots, X_{id})$ and $X_j = (X_{j1}, X_{j2}, \ldots, X_{jd})$, respectively, $f(X_i)$ and $f(X_j)$ are the fitness values of the two particles, and $D(i,j)$ is the distance between the particle and the particle j.

$$D(i,j) = \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2 + (f(X_i) - f(X_j))^2}. \tag{11}$$

The maximum distance between particles can be defined as:

$$D_{max} = \max D(i,j). \tag{12}$$

If N is the number of particles, the average distance between particles is the quotient of the sum of the spacing of all pairs of particles and the number of particles, which can be defined as:

$$D_{average} = \sum_{j=1}^{N}\sum_{j=1}^{N}\frac{D(i,j)}{N^2}. \tag{13}$$

The average distance between a particle and the optimal particle in the population can be defined as:

$$D_{aver} = \sum_{j=1}^{N} D(i, p_g). \tag{14}$$

When the particle is far away from the optimal particle, the particle needs to have a greater inertia weight. This helps to search for the global optimum and converge faster. If the particle is closer to the optimal particle, it should be given a smaller inertia weight. The weight update formula is as Equation (15). This formula can be used to calculate the inertia weight of each particle in the algorithm, and after substituting the updated formula, it can fine-tune each particle generated in the next iteration. Greatly improve the quality of the next particle population and improve the optimization speed.

$$w = \begin{cases} w_{max}, & D(X,p_g) > D_{average}, \\ w_{min} + \left(w_{max} - w_{min} \ln\left(1 + (e-1)\frac{X,p_g}{D_{average}}\right)\right), & \text{else.} \end{cases} \tag{15}$$

3) **Adaptive learning factor.** The strategy of automatically adjusting the learning factor according to the fitness of each particle. The main idea is to determine the value of the learning factor based on the relative difference between the fitness of each particle and the optimal fitness of the individual and the optimal

fitness of the group. The learning factor adjustment formula under this strategy is shown in Equation (16).

$$\begin{cases} c_1 = k_1\left|\frac{f(x_i^t)-f(p_i)}{f(p_i)}\right|, \\ c_2 = k_2\left|\frac{f(x_i^t)-f(p_g)}{f(p_g)}\right|. \end{cases} \tag{16}$$

In formula (16), c_1 and c_2 are the learning factors of particles. $f(x_i^t)$ is the fitness value of the position of particle i at time t, $f(p_i)$ is the fitness value of the optimal solution of individual particles i, and $f(p_g)$ is the fitness value of the optimal solution of all particles. k_1, k_2 are the learning factor weight coefficients. It can be seen from the two formulas that the greater the difference between the fitness value of the particle and the optimal particle fitness value, the greater the learning factor; the smaller the difference between the fitness value of the particle and the optimal particle fitness value, the smaller the learning factor. According to the fitness value, the learning parameters can be adjusted in real time, and the fuzzy controller can be optimized more efficiently.

The basic process of parameter adaptive simulated annealing particle swarm optimization algorithm is as follows:

Step 1. Initialize the particle swarm parameters, initialize the position and velocity of the particle swarm;

Step 2. Calculate the fitness value of all initialized particles in the population according to the fitness formula, store the position and fitness value of each particle currently initialized in the of each particle, and store the position of the particle with the best fitness value among all initialized particles. And the fitness value is stored in *zbest*;

Step 3. Parameter update, according to formula (15) to update the inertia weight, formula (17) to update the control parameter temperature, and formula (16) to update the learning factor;

$$T = |f(p_g)| \times (q)^t. \tag{17}$$

In the formula, q is the temperature control parameter, and the value is generally $[1.01, 1.3]$.

Step 4. Update the speed and position of each particle according to formulas (10) and (5);

Step 5. Calculate the fitness value of each particle in the new population;

Step 6. Compare the fitness value of each particle in the population with the fitness value of *gbest*. If the current particle fitness value is better, update *gbest* with the particle's current position and fitness value;

Step 7. Determine whether to accept new solutions according to Metropolis criteria;

Step 8. If the termination condition is met, output *zbest* and terminate the algorithm, otherwise skip back to Step 3.

3.3 Particle Swarm Optimization Fuzzy Controller

In the above description, the parameter selection of membership function is subjective and arbitrary, so the higher accuracy of fuzzy controller results cannot be guaranteed. PSO is used to search intelligently and find the most suitable membership function in the simulation process, which can improve the accuracy.

The membership function of PSO firstly divides the input quantity into the universe of discourse. It is not necessary to be accurate to the range of each fuzzy set, but only to determine the number of fuzzy sets of each variable. As shown in Figure 4, the variable is divided into 5 fuzzy sets such as HS, but the fuzzy number x_i is unknown.

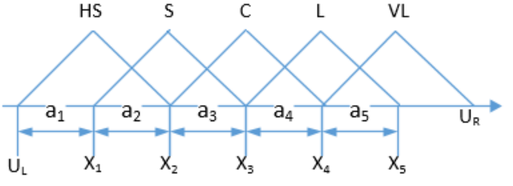

Figure 4. Fuzzy number display diagram of membership function

Secondly, when the position of the particle is expressed as the membership parameter, a certain transformation is required. The particle swarm first generates m random numbers $a_0 = [a_{01}, a_{02}, \ldots, a_{0m}]$ from 0 to 1 randomly, where m represents the number of membership parameters. In order to ensure the complete representation of membership function, the parameters of the same variable must satisfy $x_i < x_j$ $(i < j)$, that is, the fuzzy number increases with i. Multiply each random number by the maximum interval a_{max}, which is the maximum value of the difference between adjacent parameters of the membership function $(x_{i+1} - x_i)$, and set its size according to the scope of the universe of discourse. The result is the randomly generated parameter interval $a = [a_1, a_2, \ldots, a_m]$, as shown in Figure 4. The former fuzzy number plus the parameter interval equals the latter fuzzy number, and the formulas are shown in Equations (18) and (19).

$$a_i = (a_{0i})_{max}, \tag{18}$$

$$\begin{aligned} &i = 1, 2, \ldots, m, \\ &x_1 = a_1; \\ &x_2 = x_1 + a_2; \\ &x_3 = x_2 + a_3; \\ &\ldots \\ &x_m = x_{m-1} + a_m. \end{aligned} \tag{19}$$

Finally, the generated parameters are put into the position of the particle, and the program is run to generate the membership function. Add to the fuzzy controller to generate a new fuzzy controller. The accuracy of fuzzy controller is verified in simulation and the optimal membership function is found by PSO.

3.3.1 Fuzzy Rules of Particle Swarm Optimization

The number of fuzzy rules is determined by the number of input variables and the number of fuzzy sets of variables. The product n of the fuzzy sets of input variables is the number of fuzzy rule bars, the number of fuzzy sets of output variables is, then the combination of fuzzy rules has n^m species. PSO is not only inefficient and computationally intensive, but also easy to get into local optimal solution. In order to improve the efficiency, fuzzy rules are synthesized by using fuzzy rule weights, and then PSO weights are used to select the optimal combination of fuzzy rules.

First, the digitization of fuzzy rules. Each input variable has a different fuzzy set, representing different degrees of driving. For example, the monthly mileage is divided into five fuzzy sets according to the length of the mileage, which are digitized in terms of 1 to 5, respectively. Where the number 1 means the mileage is very short, and the number 5 means the mileage is very long. After all of them are digitized, they are arranged in the form of fuzzy rules in the fuzzy controller to form a matrix combination of $n \times 5$. It covers all combinations of fuzzy sets of five variables, and each fuzzy rule represents a different combination of fuzzy sets. The autonomic underwriting coefficient is calculated through five driving behavior reasoning, which is essentially seven fuzzy sets of the comprehensive influence of input variables distributed to output variables.

Then, the weight is given and the case of the output variable is calculated. Give each input variable a weight. The weight represents the influence on the result, and it also represents the risk level of the driving behavior in the driving process. For each variable, the weight is multiplied to represent the number of the fuzzy set, and the sum of the fuzzy set combinations of each row is calculated, as shown in formula (20). The total value of each combination represents the comprehensive risk size of five driving behaviors, and then the total value is equally distributed to seven fuzzy sets, resulting in a complete set of fuzzy rules, namely the matrix combination of $n \times 8$.

$$s_i = \sum_{j=1}^{5} h_{ij} \times w_j, \quad i = 1, 2, \ldots, m. \tag{20}$$

In formula (20), s_i represents the total value calculated by each fuzzy set combination, w_j represents the weight, and h_{ij} represents the fuzzy set combination.

Finally, PSO optimizes the fuzzy rule weights of input variables. The weight of each input variable closely affects the fuzzy rule, and every small change will affect the entire fuzzy rule. By constantly adjusting the weights through the PSO algorithm, each weight combination will generate different fuzzy rules. The fuzzy rules run in the fuzzy controller to obtain the results, and the calculated adaptive values feedback the advantages of the weight combination. The specific process is shown in Figure 5.

4 EMPIRICAL ANALYSIS

4.1 Design of Fuzzy Controller

4.1.1 Select the Pricing Factor

In the actual driving process, driving safety is often affected by a human, car, road and traffic environment. When the input parameters of fuzzy controller are too many, the calculation precision will decrease. In order to facilitate calculation and operation, it is necessary to select appropriate representative pricing factors. Among the multifarious impact factors, the selection of representative pricing factors can well reflect the driving risk of customers and shape the accurate rate determination coefficient. It is the basic work to build the pricing model of auto insurance, and it is also an important guarantee to calculate the premium accurately. By consulting a great deal of literature, this paper selected five representative pricing factors, including monthly driving mileage, violation times, night driving time, over speed driving time proportion and sharp turn times, and then it is explaining and analyzing the pricing factors.

1) **Monthly mileage.** Monthly mileage refers to the total monthly mileage of the customer in a given month. The driving mileage of a driver is a basic factor affecting driving safety. Through the study, it is found that the traffic accident rate will increase with the increase of drivers' mileage and time. So the longer the mileage, the greater the risk, the more representative it is as one of the pricing factors.
2) **Number of violations.** The number of violations is the number of violations of the customer's driving in a year. According to statistics, illegal driving is one of the main factors inducing road traffic accidents. Traffic accidents caused by illegal driving account for more than 50 % of highway traffic accidents and more than 70 % of human casualties [6]. Therefore, illegal driving is taken as one of the pricing factors.
3) **Driving time at night.** Night driving time refers to the total time spent driving at night. Insufficient illumination on the road at night, blinding caused by vehicle high beam lights, and decreased night vision of human eyes under dynamic conditions will lead to decreased visibility and slow response of drivers at night. In addition, nighttime is a period of frequent accidents, which makes driving at night risky. Therefore, nighttime driving time is taken as one of the pricing factors to measure the risk.
4) **Proportion of speeding time.** The proportion of speeding time refers to the proportion of driving time to total driving time at a speed of more than

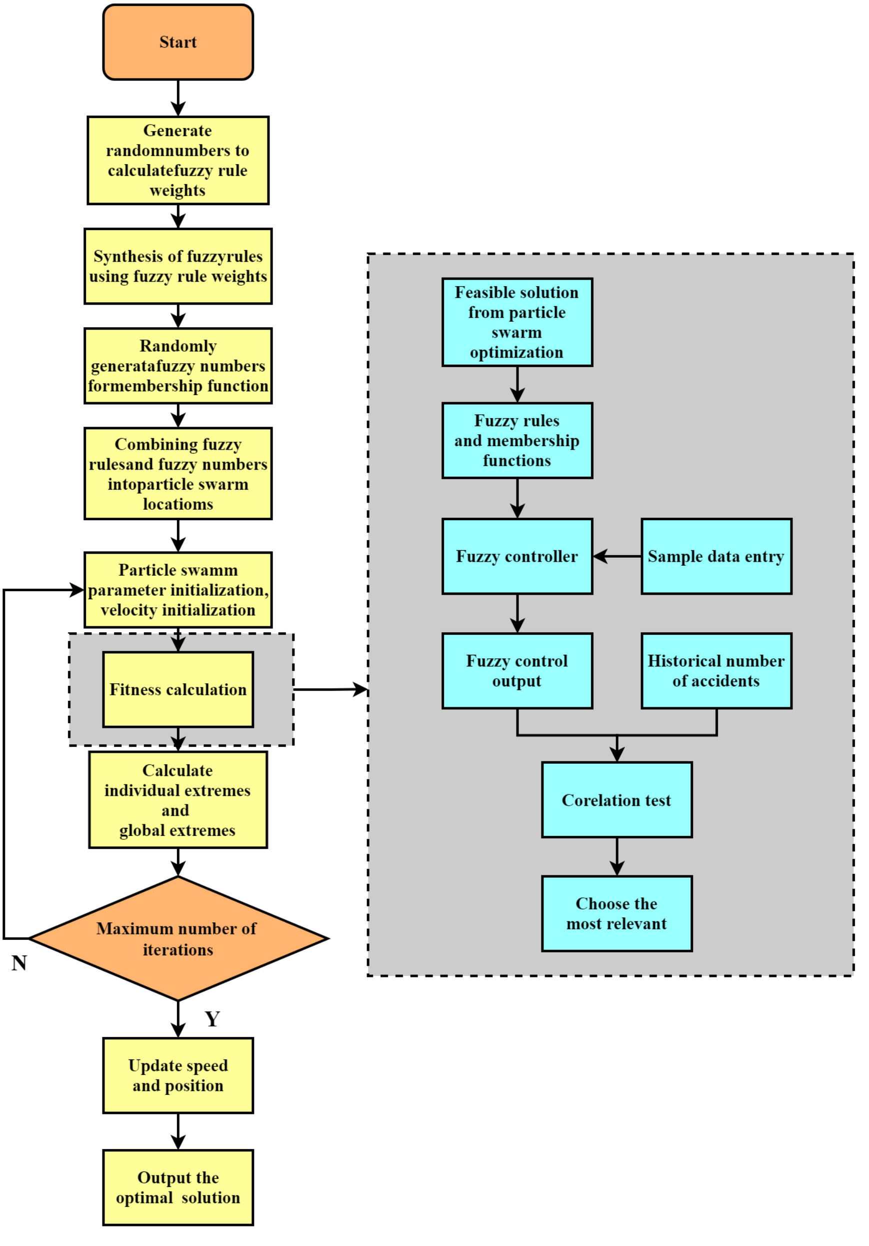

Figure 5. Flow chart of PSO optimized fuzzy controller

120 km/h. Speeding is one of the main causes of traffic accidents, the accident rate and the degree of danger will multiply with the increase of driving speed. And when driving at high speed, the driver's field of vision narrows and his ability to react decreases. To sum up, one of the pricing factors is the proportion of speeding time.

5) **Number of sharp turns.** The number of sharp turns is the number of sharp turns in the course of the vehicle, it is one of the "three urgent" behavior factors. Due to the limited number of pricing factors, sharp turn is taken as the top representative of the "three urgent" behavior factors. Sharp turning behavior will not only lead to vehicle wear and tear, vehicle easy reveals a hidden trouble, and it is easy to roll and cause a collision. So the number of sharp turns is taken as one of the pricing factors.

This paper comprehensively considers the driving habits of customers and selects five indicators from the four aspects of driving distance, driving time, driving environment and driving behavior. As the risk factor influencing the self-underwriting coefficient, the index is input into the fuzzy controller to measure the driving risk level of the customer.

4.1.2 The Setting of Membership Function

Based on the reasoning calculation of the pricing factor, the fuzzy controller obtains the autonomous underwriting coefficient. Therefore, the pricing factor monthly driving distance, the number of violations, night driving time, the proportion of speeding time and the number of sharp turns are taken as the input parameters X, which are denoted as $X = [X_1, X_2, X_3, X_4, X_5]$ in turn. The self-underwriting coefficient is taken as the output parameter Y. The value range is set as the discourse domain, and the binned fuzzy set method is used. Refer to the scoring model table in the article "Research on the UBI based auto insurance rate determination model and method in the context of Internet of Vehicles" by Zhu [5]. Blur X and Y into fuzzy subsets, as shown in Table 1.

Variables	Discourse Domain	Fuzzy Subset	Membership Function Type
X_1	$[0, 3\,000]$	{VS, S, M, L, VL}	zmf + triangle + smf
X_2	$[0, 4]$	{S, M, JL, L}	zmf + triangle + smf
X_3	$[0, 20]$	{S, M, JL, L}	zmf + triangle + smf
X_4	$[0, 7]$	{S, C, L}	zmf + triangle + smf
X_5	$[0, 65]$	{S, C, L}	zmf + triangle + smf
Y	$[0.85, 1.15]$	{HS, S, JS, C, JL, L, VL}	triangle

Table 1. Fuzzy subsets of variables

In the selection of membership function, triangular membership function is mainly used. Because the triangle function is simple and easy to use, with a strong sensitivity and wide applicability, it can better reflect the differences between the grades. The fuzzy toolbox in MATLAB software is used to set the initial membership function. The pricing factor's domain is all positive, so ZMF type membership function and SMF type membership function are used to process the boundary of the domain, representing that the membership degree beyond the scope of the domain is regarded as 1. Using the fuzzy toolbox in MATLAB, the membership function curve of the set pricing factor and independent underwriting coefficient is shown in Figure 6.

4.1.3 Fuzzy Rules

This paper establishes a five-dimensional fuzzy controller with hundreds of fuzzy rules. PSO is to select reasonable fuzzy rules from the fuzzy rule base. In order to improve the efficiency of fuzzy rules of PSO and to construct more reasonable fuzzy rules, fuzzy rule weights are selected to construct fuzzy rules. Firstly, fuzzy rule digitization. In the MATLAB software environment, the fuzzy rules of letter form cannot be recognized by the program, so it needs to be converted into numbers. For example, in Table 1, the fuzzy subset of monthly mileage is set as $\{VS = 1, S = 2, M = 3, L = 4, VL = 5\}$, and the other five factors are also digitized.

Secondly, the weights of fuzzy rules are determined. For this reason, this paper refers to the weight calculated by AHP-EW in [5] and takes it as the initial solution, with the weight $w_0 = [0.1183, 0.1934, 0.1615, 0.4024, 0.1244]$ [5].

Finally, fuzzy rules are constructed. Formula (20) is used to calculate the sum of each fuzzy rule. Each corresponds to a fuzzy subset of Y, whose fuzzy subset is digitized to 1 to 7. The determinant of 960×8 is formed. Columns 1 through 5 represent the five pricing factors, respectively. The sixth column represents the independent underwriting coefficient. Each row represents a combination of fuzzy subsets. MATLAB software was used to write the program, the fuzzy controller was established, and the file abcf.fis was generated after saving.

4.2 Optimization of Fuzzy Controller Based on Particle Swarm

4.2.1 Initialization

Using environment MATLAB software, the parameters of PSO are set as follows: population is 100, dimension is 960, number of iterations is 100. For the learning factor, c_{1f} in c_1 is 2.5 and c_{1s} is 0.5; c_{2f} in c_2 is 0.5, c_{2s} is 2.5, the inertia weights ω_s and ω_e are 0.9 and 0.4, and the particle position and velocity ranges are $[0.1, 7]$, read abcf.fis.

Parameter adaptive SAPSO algorithm parameter settings are as follows: population is 100, dimension is 960, number of iterations is 100. The initial values of the learning factors c_1 and c_2 are both 2, the learning factor weight coefficients k_1 and k_2 are set to 0.6, the inertia weight ω is set to 0.8, the particle position and velocity

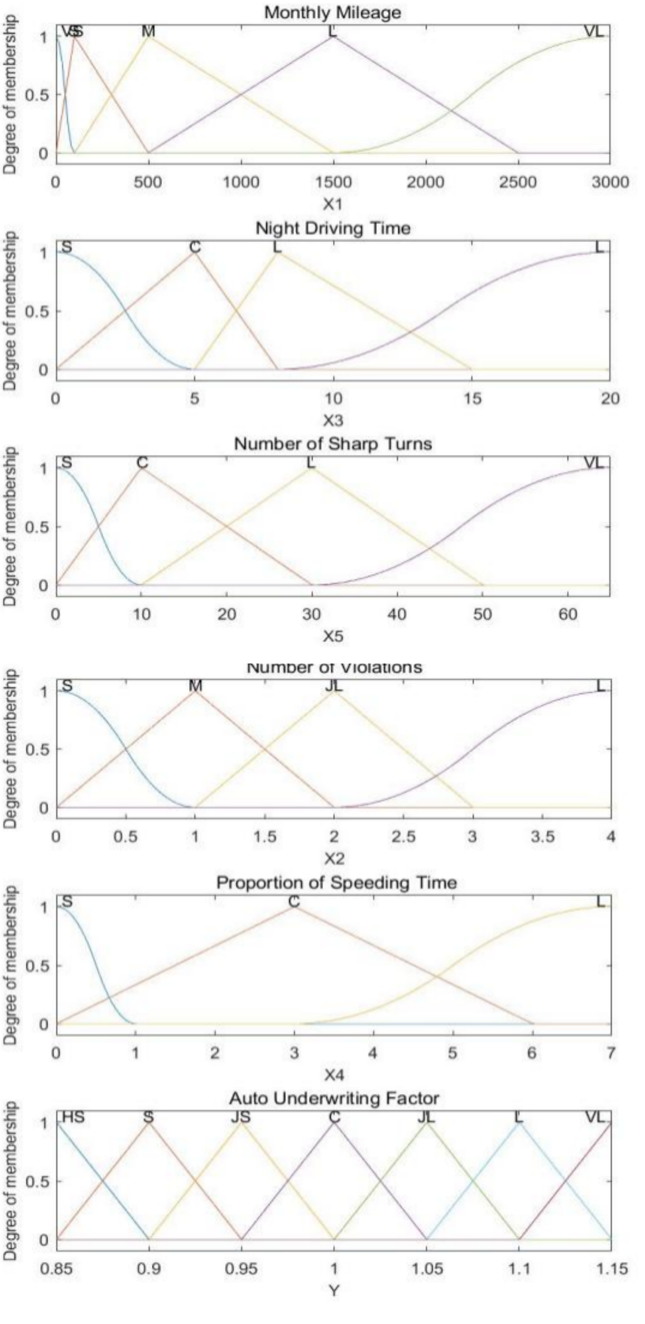

Figure 6. The membership curve of each variable

range are both $[0.1, 7]$, and is set to 0. The temperature control parameter value is 1.3, read into abcf.fis.

In this paper, the membership function and fuzzy rules are optimized by PSO algorithm, and the parameters to be optimized are 973. Where the fuzzy number to be optimized in the membership function is 13 and the fuzzy rule is 960. The details are shown in Table 2.

Price Factors to Be Optimized	Number of Fuzzy Subsets	Number of Parameters to Be Optimized
Monthly mileage	5	4
Night-driving time	4	3
Proportion of speeding time	3	3
Number of sharp turns	4	3

Table 2. The membership function of each pricing factor

Since the scope of the number of violations of the factor is small, the membership function can be optimized in less space after it is divided into four fuzzy subsets. In order to reduce the task of optimization, membership function can be set directly. The membership functions of the other four pricing factors have different fuzzy numbers, among which the left and right endpoints of the domain range are also fuzzy numbers of each factor, which are determined in the membership function. By comparing the figure in Figure 4, it can be seen that the remaining fuzzy numbers after removing the endpoints need to be optimized, among which the membership function of the ratio of proportion of speeding time is special, and the fuzzy number to be optimized is set as 3.

4.2.2 Fitness Calculation and Data Processing

The sample data came from Zhu's work [5] on the UBI based auto insurance rate determination model and method under the Internet of vehicles environment, with 100 customers' sample data. The sample included data on monthly mileage, number of violations, night-time driving time, percentage of speeding times, number of sharp turns and number of historical accidents. The data of 60 customers were selected as the training set and the remaining 40 as the test set. Some users' driving behavior data examples are shown in Table 3.

There is a dimensional difference between the indicators of the index system, so normalization needs to be performed. The specific normalization operation is implemented by the mapminmax function, and the implemented formula is shown in Equation (21):

$$y = \frac{(y_{\max} - y_{\min}) * (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min}. \tag{21}$$

Among them, x in the formula is the feature vector that needs to be normalized, $x_{\max}$ and $x_{\min}$ respectively represent the largest and smallest vectors in D, $y_{\max}$ and $y_{\min}$ correspond to the largest and smallest values in the interval range after the

User	Monthly Mileage (km)	Proportion of Time Higher Than 120 km/h (%)	Rapid Acceleration	Night-Driving Time (h)
1	84	0	0	0
2	362	0	1	0.9
3	1 137	0.66	1	0
4	870	0.48	4	0
5	754	1.02	2	1.1
6	1 180	0.63	3	4.3
7	1 240	1.24	13	6
8	574	3.04	6	1.6
9	1 710	0.95	34	7.3
10	1 185	4.86	4	6.1

Table 3. Some examples of user driving behavior data

data is normalized. At this time, all the values in the original stock characteristic data are converted into the interval $[0, 1]$. Such processing can avoid the dimension between different characteristic attributes without affecting the internal relationship of the same attribute value.

When the new fuzzy rule and membership function constitute the fuzzy controller, the training set data is imported into the fuzzy controller to obtain the autonomous underwriting coefficient. Works [5] and [35] use the same data to score driving behaviors using AHP-EW and gray correlation-level analysis respectively. This article draws on the driving behavior score data in the two articles as the basis for judgment. Since the autonomous underwriting coefficient is inversely proportional to the driving behavior score, the fitness value is set as the absolute value of the correlation between the autonomous underwriting coefficient and the driving behavior score. The fuzzy rule with the largest fitness value is selected as the optimal solution. Then, the correlation analysis between the number of historical risks and the independent underwriting coefficient is made, and the correlation degree is calculated using the function in MATLAB software.

4.3 Result Analysis

Based on MATLAB software, this paper runs particle swarm algorithm to optimize the membership function and fuzzy rules. The driving score in [35] is included in the fitness function as the criterion. The driving scores of work [5] were compared as a control group. The following is a comparison of fuzzy rule construction methods, a comparative analysis of weighting algorithms, and correlation testing. For ease of listing, in the following table, parameter adaptive PSO is abbreviated as APSO, and parameter adaptive SAPSO is abbreviated as ASAPSO.

4.3.1 Comparative Analysis of Fuzzy Rule Construction Methods

Fuzzy rules are generally set artificially in fuzzy controllers based on the expert experience or design concepts. In the algorithm, a random array that conforms to fuzzy rules is generated by a random group function. Due to the high dimensionality of the pricing model, there are as many as 973 parameters to be optimized for the particle swarm. In order to simplify the calculation and optimization process, this paper proposes a method of constructing fuzzy rules indirectly using fuzzy rule weights, calculate the weight of each input variable, that is, the weight of the fuzzy rule. The weighted average algorithm is used to construct fuzzy rules based on the weights of fuzzy rules. Compared with direct generation, the fuzzy rules constructed by this method are not only more regular and reasonable, but also reduce a lot of randomness, and the optimized optimal solution is also more representative. The two algorithms of adaptive PSO and parameter adaptive SAPAO are optimized by using two construction algorithms respectively. The training set is used for optimization in the algorithm, and the test set is used for comparative analysis of the optimal solution. The specific results are shown in Table 4.

Construction Method	APSO	ASAPSO
Fitness 1	**0.7589**	0.7791
Test value 1	0.6572	0.6047
Relative error 1	**13.40 %**	**17.44 %**
Fitness 2	**0.9293**	**0.9422**
Test value 2	0.8962	0.9413
Relative error 2	**3.56 %**	**0.09 %**
Fitness difference	**0.1704**	0.1739

Table 4. Comparison of the results of two methods of constructing fuzzy rules (1 stands for direct generation, 2 stands for use weight)

Table 4 lists the comparison of the results of the two optimization algorithms. From a longitudinal comparison, the fuzzy rules directly generated by the parametric PSO algorithm optimization have a fitness 1 of 0.7580; the fuzzy rules composed of optimized weights have a fitness 2 of 0.9293, and the fitness difference between the two is 0.1704. Similarly, in the parameter adaptive SAPSO algorithm, the fitness level 2 is higher than the fitness level 1. Through the relative error comparison, the relative error 1 of the parameter adaptive SAPSO algorithm is 17.44 %, the relative error 2 is 0.09 %, the error is greatly reduced, and the same is true for the parameter adaptive PSO. Therefore, it is considered that the optimal fuzzy rules generated by the weight construction fuzzy rules are more representative. Obviously, in the optimization algorithm, the method of using weights to construct fuzzy rules is better than directly generated fuzzy rules. Not only is the accuracy of the optimized fuzzy rules higher, but also the stability under different data is better and more representative.

In a horizontal comparison, the performance of parameter adaptive SAPSO algorithm is higher than parameter adaptive PSO under the same fuzzy rule construction method. Regardless of the fuzzy rule generation method, the accuracy of the parameter adaptive SAPSO algorithm is higher than that of the parameter adaptive PSO. Especially combined with the fuzzy rule method of weight formation, the stability of parameter adaptive SAPSO is greatly improved. Oblique comparison found that the adaptability of parameter adaptive PSO 2 is 0.9293, and the adaptability of parameter adaptive SAPSO 1 is 0.7791. It can be seen that the method of constructing fuzzy rules has a great influence on the results of the optimization algorithm. Using weights to construct fuzzy rules can greatly improve the accuracy and stability of the optimization algorithm.

4.3.2 Comparison of Algorithms for Determining the Weights of Fuzzy Rules

There are many ways to determine the weight. This paper chooses five algorithms: AHP, EW, GA, parameter adaptive PSO and parameter adaptive SAPSO algorithm to determine the weights, and then construct fuzzy rules to obtain the optimal solution. The specific results are shown in Table 5.

Algorithm	Training Set	Test Set	Relative Error	Experimental Data Set	Control Data Set	Difference
AHP	0.7984	0.7438	0.0684	0.7841	0.8078	−0.0237
EW	0.8859	0.8562	0.0335	0.8783	0.8917	−0.0133
GA	0.7673	0.6014	0.2162	0.7162	0.7016	0.0146
APSO	0.9293	0.8962	0.0356	0.9193	0.9183	0.0011
ASAPSO	**0.9422**	**0.9413**	**0.0009**	**0.9428**	**0.9439**	**−0.0011**

Table 5. Comparison of the results of determining the weighting algorithm

In Table 5, five methods are used to determine the weight. Among them, AHP determines the weight subjectively based on expert experience, and EW objectively extracts the index weight based on driving data. The other three are all weighed in the simulation of optimizing fuzzy rules. Through comparative analysis, it can be seen that the weights determined by the APSO and ASAPSO algorithms are more accurate, and the difference with the control data set is also the smallest, only −0.0011. Among them, ASAPSO has the highest accuracy. It can be seen that the algorithm has stronger optimization performance when optimizing fuzzy rules.

Figure 7 is the individual optimal solution diagram of the ASAPSO algorithm during 100 iterations. It can be seen from Figure 7 that the ASAPSO algorithm regularly seeks the optimization up and down with the adaptation degree of 0.9 as the axis during the entire iterative process. During the whole iteration process, the fitness value of most optimal individuals is above 0.86, and the quality of the particle population is very good. It is the adaptive adjustment of parameters in the optimization process that guarantees the population quality of each iteration. In

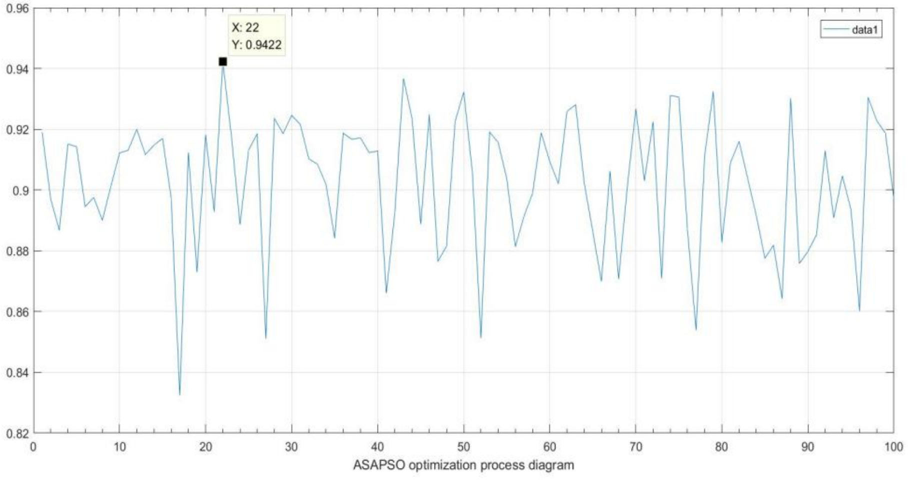

Figure 7. Iterative diagram of parameter adaptive SAPSO optimization

the 22$^{\text{nd}}$ iteration, the global optimal solution was found to be 0.9422, which shows that the optimization speed is relatively fast and the optimization performance is strong. The motion direction variation of the algorithm makes it move backward immediately when it finds a local optimal solution so as not to fall into the local optimal solution, and obtain a larger exploration range and optimization space.

When optimizing fuzzy rules, the membership function is optimized simultaneously. The membership functions of four input variables are optimized by the fuzzy numbers of 13 membership functions, which are monthly mileage, night driving time, proportion of speeding time, and number of sharp turns. The details are shown in Figure 8.

4.3.3 Correlation Test

In this paper, the driving score is used as the relevant data of the fitness function to test its correlation. The driving score in [35] is used as experimental data, and the correlation with the number of historical accidents is -0.5497. The driving score in [5] is used as the control group data, and the correlation with the number of historical accidents is -0.5593. The optimal fuzzy rules and membership functions are screened out and substituted into the fuzzy controller to obtain the autonomous underwriting coefficient. Using SPSS software to test the correlation between the independent underwriting coefficient and the number of historical risks, the Pearson correlation between the two is 0.5836, as shown in Figure 9. Since the historical number of risks are all integers, and the range of change is 0–4, individual differences are too small, so the correlation with the independent underwriting coefficient is still relatively high. This shows that the results obtained in this paper are reasonable and the algorithm performance is relatively superior.

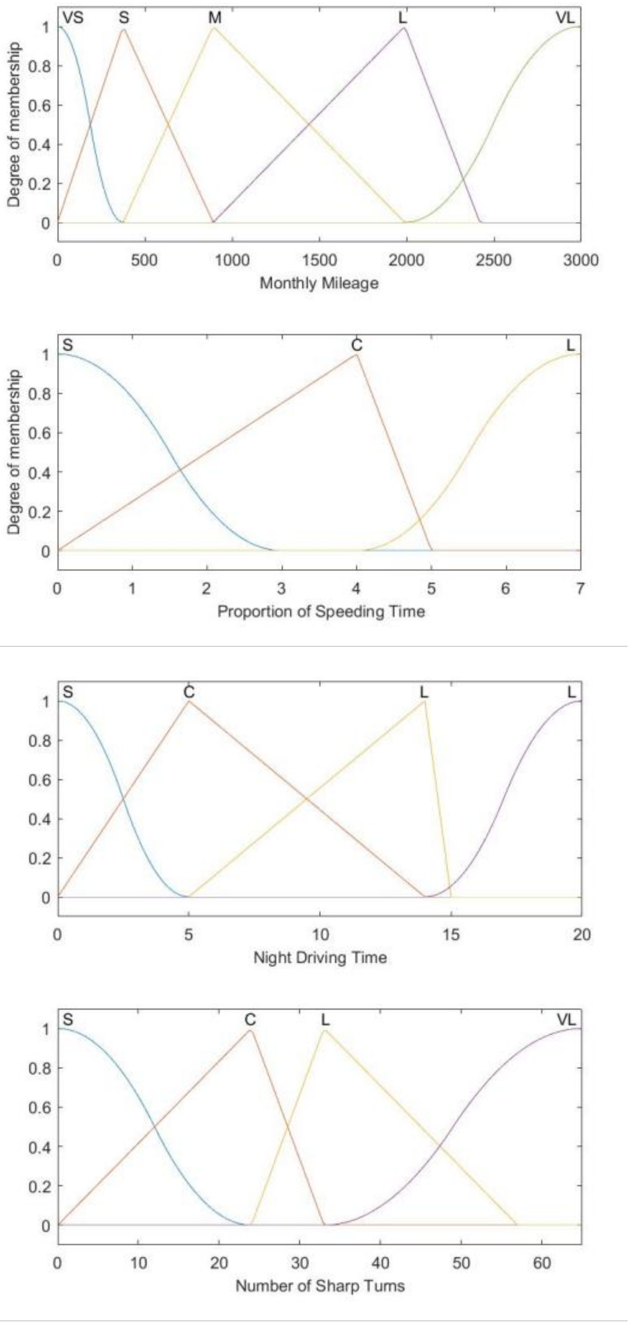

Figure 8. Optimized membership function

Correlation

		NAH	IUC
NAH	Pearson correlation	1	.584**
	Significance		.000
	N	100	100
IUC	Pearson correlation	.584**	1
	Significance	.000	
	N	100	100

NAH=Number of accidents in history; IUC=Independent underwriting coefficient.

Figure 9. Optimized membership function diagram

After optimization by parameter adaptive particle swarm optimization, the optimal fuzzy rules and membership functions are added to the fuzzy controller. Part of the obtained independent underwriting coefficients for 100 customers is shown in Table 6.

Order Number	Independent Underwriting Factors	Order Number	Independent Underwriting Factors
1	0.8869	11	0.9490
2	0.9505	12	0.8995
3	1.0261	13	0.9760
4	0.8995	14	0.8515
5	1.1046	15	0.8995
6	0.8828	16	0.9250
7	0.8853	17	1.0239
8	0.9060	18	0.8995
9	1.0000	19	0.8605
10	0.9505	20	1.0510

Table 6. Independent underwriting coefficient for 20 customers

It can be seen from Table 6 that through the fuzzy controller, the driving information of 20 customers is input to obtain the independent underwriting coefficient of each person. In the customer's UBI auto insurance premium calculation, different independent underwriting coefficients are substituted into the premium calculation formula to meet the differentiation and autonomy of auto insurance pricing.

In this chapter, stability is included in the screening conditions, and the improved algorithm is used to process UBI customer data to obtain the fitness function. On the premise of ensuring stability, high-precision fuzzy rules are screened, which avoids the over-optimization of fuzzy rules and the dependence on data. By comparing the results of various algorithms, it is found that the SAPSO algorithm has high accuracy and good stability in processing UBI auto insurance data, so on this basis, the optimal fuzzy rules are selected to construct a fuzzy controller. Finally, the optimal fuzzy controller is used to calculate the autonomous underwriting coefficient for UBI auto insurance pricing.

5 CONCLUSION

Aiming at the precise pricing and premium differentiation of UBI auto insurance, a pricing model using fuzzy controllers to calculate autonomous underwriting coefficients is proposed. On the basis of this model, the fuzzy controller is optimized, and a parameter adaptive SAPSO algorithm is proposed. This algorithm optimizes the membership function and fuzzy rules of the fuzzy controller. Among them, the weight composition method is designed to achieve the purpose of optimizing fuzzy rules, and good results have been achieved. In the optimization process, the correlation between the autonomous underwriting coefficient and the driving score is used as the fitness value to judge the pros and cons of each set of fuzzy rules, and train the optimal combination in the driving score data. In comparison with the results of other algorithms, it can be seen that this algorithm has a higher accuracy. In order to verify the stability of the algorithm, the comparison of the results of test samples, training samples and control samples is studied. The correlation test was carried out to prove the rationality of the model results. The experimental results show that the parameter adaptive SAPSO algorithm can self-adjust the parameters during the optimization process, so that the particle swarms produced are of higher quality, with higher accuracy and stability. The UBI pricing model of parameter adaptive SAPSO optimized fuzzy controller can realize accurate differentiated pricing. In the future, we will add more pricing factors to improve the optimization algorithm and optimize on the basis of big data to establish a more accurate and scientific auto insurance pricing model.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

This work was supported in part by the General Projects of Shandong Provincial Natural Science Foundation (Nos. ZR2022MG059, ZR2020MF033), National Bureau of Statistics, National Statistical Science Research Project Key Project, "Research on the Co-movement and Spatial Spillover Effect Between the Development of Digital Economy and Real Economy", (No. 2022LZ31).

REFERENCES

[1] Arbabzadeh, N.—Jafari, M.: A Data-Driven Approach for Driving Safety Risk Prediction Using Driver Behavior and Roadway Information Data. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, 2018, No. 2, pp. 446–460, doi: 10.1109/TITS.2017.2700869.

[2] Ma, Y. L.—Zhu, X.—Hu, X.—Chiu, Y. C.: The Use of Context-Sensitive Insurance Telematics Data in Auto Insurance Rate Making. Transportation Research Part A: Policy and Practice, Vol. 113, 2018, pp. 243–258, doi: 10.1016/j.tra.2018.04.013.

[3] Ayuso, M.—Guillén, M.—Nielsen, J. P.: Improving Automobile Insurance Ratemaking Using Telematics: Incorporating Mileage and Driver Behaviour Data. SSRN, Working Paper XREAP2016-08, 2017, doi: 10.2139/SSRN.2885214.

[4] Yan, C.—Ou, Z.—Liu, W.—Xu, Q.: Research on UBI Auto Insurance Pricing Model Based on Adaptive SAPSO to Optimize the Fuzzy Controller. International Journal of Fuzzy Systems, Vol. 22, 2020, No. 2, pp. 491–503, doi: 10.1007/s40815-019-00789-6.

[5] Zhu, S.: Research on Model and Method of Auto Insurance Rate Determination Based on UBI in Vehicle Network Environment. Ph.D. Thesis. Beijing Jiaotong University, 2015 (in Chinese).

[6] Gao, Y.: UBI Rate Determination Model Based on Driving Behavior Classification. Ph.D. Thesis. Beijing Jiaotong University, 2017 (in Chinese).

[7] Li, Z. X.—Meng, S. W.: Construction and Evaluation of BMS for Automobile Insurance in China. Systems Engineering – Theory and Practice, Vol. 38, 2018, No. 4, pp. 938–949, doi: 10.12011/1000-6788(2018)04-0938-12 (in Chinese).

[8] Zhang, L. Z.—Wang, D.: Auto Insurance Rate-Making with Large Claims. Statistics and Information Forum, Vol. 34, 2019, No. 1, pp. 58–63 (in Chinese).

[9] Peng, J. Q.—Liu, N. J.—Zhao, H. T.—Yu, M. L.: Research on Smart Usage-Based Insurance System. Computer Technology and Development, Vol. 26, 2016, No. 1, pp. 142–146 (in Chinese).

[10] Wang, X.: UBI Pricing Model Based on Natural Driving Data. Ph.D. Thesis. Beijing Jiaotong University, 2018 (in Chinese).

[11] Pang, L. F.: Research on the Application of UBI Auto Insurance in Shared Cars. Internal Combustion Engine and Accessories, Vol. 4, 2021, pp. 185–186, doi: 10.19475/j.cnki.issn1674-957x.2021.04.085 (in Chinese).

[12] Wang, R.—Kuang, Y. Q.: Challenges and Countermeasures of UBI Auto Insurance in My Country Under the Background of the Internet of Vehicles. China Insurance, Vol. 3, 2021, pp. 34–36 (in Chinese).

[13] Ji, X. T.: The Application of UBI in China's Auto Insurance Market. China Insurance, Vol. 3, 2021, pp. 22–25 (in Chinese).

[14] Garg, H.: A Hybrid PSO-GA Algorithm for Constrained Optimization Problems. Applied Mathematics and Computation, Vol. 274, 2016, pp. 292–305, doi: 10.1016/j.amc.2015.11.001.

[15] Olivas, F.—Valdez, F.—Castillo, O.—Melin, P.: Dynamic Parameter Adaptation in Particle Swarm Optimization Using Interval Type-2 Fuzzy Logic. Soft Computing, Vol. 20, 2016, No. 3, pp. 1057–1070, doi: 10.1007/s00500-014-1567-3.

[16] Durand, F. R.—Abrão, T.: Power Allocation in Multibeam Satellites Based on Particle Swarm Optimization. AEU – International Journal of Electronics and Communications, Vol. 78, 2017, pp. 124–133, doi: 10.1016/j.aeue.2017.05.012.

[17] Valdez, F.—Vazquez, J. C.—Melin, P.—Castillo, O.: Comparative Study of the Use of Fuzzy Logic in Improving Particle Swarm Optimization Variants for Mathematical Functions Using Co-Evolution. Applied Soft Computing, Vol. 52, 2017, pp. 1070–1083, doi: 10.1016/j.asoc.2016.09.024.

[18] Pan, A. Q.—Wang, L.—Guo, W. A.—Wu, Q. D.: A Probability-Based Coevolving Multi-Objective Algorithm for Antenna Array Synthesis. Applied Soft Computing, Vol. 73, 2018, pp. 178–191, doi: 10.1016/j.asoc.2018.08.011.

[19] Lagunes, M. L.—Castillo, O.—Valdez, F.—Soria, J.: Multi-Metaheuristic Competitive Model for Optimization of Fuzzy Controllers. Algorithms, Vol. 12, 2019, No. 5, Art. No. 90, doi: 10.3390/a12050090.

[20] He, S. N.—Chen, W. X.—Mu, X. P.—Cui, W.: Constrained Optimization Model of the Volume of Initial Rainwater Storage Tank Based on ANN and PSO. Environmental Science and Pollution Research, Vol. 27, 2020, pp. 21057–21070, doi: 10.1007/s11356-020-08630-6.

[21] Patel, S.—Eldho, T. I.—Rastogi, A. K.: Hybrid-Metaheuristics Based Inverse Groundwater Modelling to Estimate Hydraulic Conductivity in a Nonlinear Real-Field Large Aquifer System. Water Resources Management, Vol. 34, 2020, No. 6, pp. 2011–2028, doi: 10.1007/s11269-020-02540-5.

[22] Wu, H.—Liu, J.—Dong, Z.—Liu, Y.: A Hybrid Mobile Node Localization Algorithm Based on Adaptive MCB-PSO Approach in Wireless Sensor Networks. Wireless Communications and Mobile Computing, Vol. 2020, 2020, Art. No. 3845407, doi: 10.1155/2020/3845407.

[23] Meng, Q.—Qiu, R.—Zhang, M.—Liu, G.—Zhang, Z.—Xiang, M.: Navigation System of Agricultural Vehicle Based on Fuzzy Logic Controller with Improved Particle Swarm Optimization Algorithm. Transactions of the Chinese Society of Agricultural Machinery, Vol. 46, 2015, No. 3, pp. 29–36 (in Chinese).

[24] Wang, G. Y.—Huang, Z. G.—Dai, M.: Fuzzy Control Research for Brushless DC Motor Based on Improved Particle Swarm Algorithm. Journal of Guangxi Normal University (Natural Science Edition), Vol. 34, 2016, No. 2, pp. 21–27, doi: 10.16088/j.issn.1001-6600.2016.02.004 (in Chinese).

[25] Zhang, Y.—Liu, Q.—Liu, H. F.—Wang, Z.: Improved Particle Swarm Optimization-Based Fuzzy Control for Brushless DC Motor. Modular Machine Tool and Automatic Manufacturing Technique, Vol. 10, 2017, pp. 101–104, doi: 10.13462/j.cnki.mmtamt.2017.10.024 (in Chinese).

[26] Tian, Y.—Liang, Y. Q.: Cuckoo Search Algorithm Based on Mini-Batch Gradient Descent. Journal of Shandong University of Science and Technology (Natural Science), Vol. 39, 2020, No. 5, pp. 56–67, doi: 10.16452/j.cnki.sdkjzk.2020.05.007 (in Chinese).

[27] Wang, X. Q.—Yue, S. M.—Zhang, Y. H.—Liu, J.—Wang, Y.—Yang, Q.:

Attention Based Sequence Labeling Model with Feature Fusion. Journal of Shandong University of Science and Technology (Natural Science), Vol. 39, 2020, No. 5, pp. 79–88, doi: 10.16452/j.cnki.sdkjzk.2020.05.009 (in Chinese).

[28] Xu, Y.—Li, X. Z.—Wu, Z. H.—Gao, Z.—Liu, L.: Potato Leaf Disease Recognition via Residual Attention Network. Journal of Shandong University of Science and Technology (Natural Science), Vol. 40, 2021, No. 2, pp. 76–83, doi: 10.16452/j.cnki.sdkjzk.2021.02.009 (in Chinese).

[29] Yu, Y. W.—Sun, Z. J.: Geometric Error Identification Method of High-Speed Crankshaft Servo CNC Grinder Based on SAPSO-GA Algorithm. Machine Tool and Hydraulics, Vol. 49, 2021, No. 19, pp. 1–4 (in Chinese).

[30] Zhang, W.—Huang, W. M.: Design of RBF Neural Network Based on SAPSO Algorithm. Control and Decision, Vol. 36, 2021, No. 9, pp. 2305–2312, doi: 10.13195/j.kzyjc.2020.0176 (in Chinese).

[31] Zadeh, L. A.: Fuzzy Sets. Information and Control, Vol. 8, 1965, No. 3, pp. 338–353, doi: 10.1016/S0019-9958(65)90241-X.

[32] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of ICNN '95 – International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[33] Eberhart, R. C.—Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. Proceedings of the 2000 Congress on Evolutionary Computation (CEC '00), Vol. 1, 2000, pp. 84–88, doi: 10.1109/CEC.2000.870279.

[34] Ratnaweera, A.—Halgamuge, S. K.—Watson, H. C.: Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients. IEEE Transactions on Evolutionary Computation, Vol. 8, 2004, No. 3, pp. 240–255, doi: 10.1109/TEVC.2004.826071.

[35] Jiang, J. F.: Research on Car Insurance Product Design and Pricing Based on Driving Behavior. Ph.D. Thesis. Huaqiao University, 2020, doi: 10.27155/d.cnki.ghqiu.2020.000380 (in Chinese).

Xindong Wang received his B.Sc. degree in financial mathematics and the M.Sc. degree in statistics from the Shandong University of Science and Technology, Qingdao, China, in 2018 and in 2020, respectively. He is Ph.D. candidate of the Shandong University of Science and Technology. His research interests are in the areas of big data, blockchain, actuarial and computer algorithm improvement, and financial risk control.

Wei Liu is engaged in research work in workflow, service computing, Petri net theory and its application, software formal analysis and verification, and big data intelligent analysis and processing.

Zhaochuang Ou received his B.Sc. and M.Sc. degree in statistics from the Shandong University of Science and Technology, Qingdao, China, in 2018 and in 2020, respectively. His research interests are in the areas of big data, blockchain, actuarial and computer algorithm improvement, and financial risk control.

Chun Yan received her B.Sc., M.Sc. and Ph.D. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2000, 2003 and 2011, respectively. She is presently Associate Professor of the Shandong University of Science and Technology, Qingdao, China. Her research interests include applied mathematics, statistics and economic management. She has about 20 technical papers published in journals and conference proceedings in the research areas.

Man Qi is Senior Lecturer in computing at the Canterbury Christ Church University, UK. Her main research interests are in the areas of intelligent systems and applications, data analytics, cyber security and HCI. Her research has been founded by EPSRC, EU and QR etc. She is Fellow of the British Computer Society (FBCS) and Fellow of the Higher Education Academy (FHEA). She has published over 70 research papers and is on the editorial board of five international journals. She has been an external Ph.D. examiner for a number of universities in the UK and Australia. She has served as Chair and Program Committee member for over 50 international conferences and has been a long-term reviewer for many international journals.

Wenli Jia received her Ph.D. degree in history from the Capital Normal University, Beijing, China, in 2011. She works at Archives in Shandong University of Science and Technology, Qingdao, China. Her research interest is in the area of information resource management.