

# Handling Uncertainty in Financial Decision Making: A Clustering Estimation of Distribution Algorithm With Simplified Simulation

Wen Shi , Wei-Neng Chen , Senior Member, IEEE, Tianlong Gu, Hu Jin , and Jun Zhang , Fellow, IEEE

**Abstract**—In financial decision making models, parameters are usually obtained based on historical data, which involve strong uncertainties. In some cases, the fluctuation caused by environmental uncertainty may even be more significant than that caused by utilizing different strategies. Such phenomenon makes the optimization and uncertainty handling in financial optimization a great challenge. In this article, a group insurance portfolio problem is considered as an instance of financial optimization with strong uncertainty. To handle uncertainty, we first analyze the feature of the problem and discover that in such kind of optimization problem with strong uncertainty, the solutions are strongly relative to the scenario. In view of the scenario-relevant feature, a simplified simulation approach is designed. Only one scenario is simulated for each generation in the evolution process to deal with the uncertainties. Combining this approach with a clustering estimation of distribution algorithm, a new algorithm (CEDA-SS) is proposed. Estimation of current profit is made by Monte Carlo (MC) simulation based on historical data. Solutions in each generation are evaluated in the same scenario. Two kinds of clustering mechanisms are applied to further improve the performance of the algorithm. Moreover, a comparison mechanism based on the Wilcoxon rank sum test is proposed to evaluate the performance of the algorithms. Experimental results show that the proposed CEDA-SS is suitable for the group insurance portfolio problem and it outperforms other uncertain evolutionary algorithms.

**Index Terms**—Financial decision making, optimization under uncertainty, insurance portfolio, estimation of distribution algorithm.

## I. INTRODUCTION

**I**N THE era of big data, the model construction and parameter acquisition of many problems are obtained through

Manuscript received April 29, 2020; revised June 29, 2020; accepted July 18, 2020. Date of publication August 12, 2020; date of current version January 21, 2021. This work was supported in part by the National Key Research and Development Project, Ministry of Science and Technology, China under Grant 2018AAA0101300, in part by the National Natural Science Foundation of China under Grants 61976093 and 61876111, in part by the Guangdong-Hong Kong Joint Innovative Platform of Big Data and Computational Intelligence 2018B050502006, and in part by the Guangdong Natural Science Foundation Research Team 2018B030312003. (Corresponding author: Wei-Neng Chen.)

Wen Shi and Wei-Neng Chen are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: 327736263@qq.com; cwnraul634@aliyun.com).

Tianlong Gu is with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: gu@guet.edu.cn).

Hu Jin and Jun Zhang are with the Division of Electrical Engineering, Hanyang University, Ansan 15588, Korea (e-mail: hjin@hanyang.ac.kr; junzhang@ieee.org).

Digital Object Identifier 10.1109/TETCI.2020.3013652

statistical analysis of the data, which is inherently uncertain [1], [2]. Therefore, more and more attentions have been paid to optimization under uncertainty in recent years. However, due to the uncertainties in the problem, the objective function cannot be calculated by deterministic expressions, which makes traditional optimization methods difficult to apply.

Evolutionary algorithm (EA) [3], [4] is a population-based stochastic optimization method which applies an uncertain mechanism to approximate the optimal solution of the problem. It has become an important tool for solving uncertain decision making problems. Generally, the application of EA in decision making optimization under uncertainty are divided into the following four categories [5].

- 1) Noisy fitness function [6], [7]: In noisy fitness function, the evaluation of the fitness value is affected by noise. Therefore, the evaluated fitness value in the optimization process deviates from the real fitness value. To deal with this kind of uncertainties, the average value of several evaluated values is applied to measure the real fitness value.
- 2) Search for Robust solutions [8], [9]: In some decision making problems, the design variables might change after the optimization process. In such problems, robust solutions that can work satisfactorily even though the design variables change should be found.
- 3) Approximated fitness function [10]–[13]: For decision making problems that fitness values are hard to evaluate, surrogate model is applied to approximate the fitness function. In this kind of problems, the error will not be changed once the surrogate model is established. Therefore, a balance should be found between the accuracy and the computing costs.
- 4) Dynamic environments [14]–[16]: For decision making problems that objective functions change over time, the optimal solutions change over time as well. The objective of this kind of problems are to apply algorithms that can continuously track the optimal solution so that the optimization process does not need to restart.

Monte Carlo (MC) simulation [17]–[19] and fuzzy system [20]–[23] are two widely applied approaches to deal with uncertainties in different problems. Fuzzy system is a generalization of deterministic system. The input, output and state variables in fuzzy system are defined on fuzzy sets. It can imitate human comprehensive inference to deal with fuzzy information

processing problems that are difficult to solve by conventional mathematical methods. MC simulation is a numerical calculation method based on the law of large number that the average value for a large number of repeated random trials is always close to a certain value [24]. MC simulation method estimates the probability of random events or get numerical characteristics of random variables by executing a large number of experiments. Compared with traditional derivation method, MC has strong problem adaptability. The complexity of the problem has little effect on it. The increase of the problem dimension will not affect its convergence speed.

Financial decision making problem is one of the most important issues in the real life. All households and companies face financial decision making problems to maintain their operation. Although the problem about optimization under uncertainty has been considered in many studies, the uncertainties appeared in financial decision making exhibit a quite different characteristic. In most existing studies about optimization under uncertainty, uncertainties usually appear as some noises added to the objective function of the problem, or disturbances associated with the decision variables. However, in financial decision making, the uncertainties appear in a quite different way. The uncertainties in financial decision making affect not only objective function but also decision variables. Moreover, external uncertainties have a greater impact on the reward than the investment strategy itself in some cases. This kind of uncertainties is not the one considered in most of the existing studies on uncertain optimization. To this end, this article analyzes the characteristics of uncertainties in financial decision making problems and proposes a clustering estimation of distribution algorithm with simplified simulation (CEDA-SS) to deal with them. MC simulation method is applied to deal with the uncertainties in the financial decision making problems in this article due to its superiority and convenience in handling statistical problems. On one hand, it can deal with statistical problems directly without the complicated mathematical derivation and calculation process. On the other hand, it can avoid the fat tail phenomenon in statistical distributions.

In most of the financial decision making problems, the uncertainties of the investment decisions are very strong. On one hand, the construction of the model is usually based on a large amount of irregular historical data. The uncertainties in financial decision making rarely obey certain statistical distributions. On the other hand, there are many extreme scenarios in financial markets, which lead to a large amount of outliers in the solutions. Therefore, a large amount of simulations are needed to obtain a stable expectation value for evaluation. How to reduce the number of simulations and improve the efficiency of evolutionary algorithm has become a big challenge in dealing with uncertainties in financial decision making problems.

In this article, a group insurance portfolio problem is taken as an instance to study how to deal with decision making problems under strong uncertainty. Insurance portfolio problem is a hot issue studying how to make a proper portfolio plan among several insurance products. In group insurance portfolio problem, the decision making model is built based on a huge number of historical data such as the historical profit, historical mortality, historical incidence rate and so on. Former study [25] takes

the approximated expectation value of the annual reward as the objective function of the optimization problem. The expectation reward is approximated by utilizing the statistical average value of parameters in the decision making model. It is convenient in calculation, but it cannot estimate the expectation reward accurately. MC simulation can make a better approximation of the expectation reward if a large number of simulations are executed, but it is computationally intensive. To improve efficiency of the algorithm, we analyze the feature of group insurance portfolio problem and propose CEDA-SS. Our work includes the following three aspects:

First, we analyze the feature of group insurance portfolio problem. From the analysis, we firstly show that it is still meaningful to perform optimization with the presence of strong uncertainty since the solutions in this problem can still be distinguished. Then we discover that a large amount of MC simulations are needed to stabilize the estimation of the fitness function value. Lastly, we find out that although the uncertainties of the problem are strong, the correlations of the solutions in different scenarios are high. That is to say, the impact of the scenario on the result is more significant than the impact of the solution. Therefore, the number of simulations can be reduced based on this scenario-relevant feature.

Second, in view of the scenario-relevant feature, a simplified simulation approach is introduced. Only one scenario is needed for each generation in the evolution process. The fitness value of the optimization problem is the profit of each simulated result. An evaluation mechanism that makes comparisons in every simulated scenario and evaluates the algorithm by considering the performance of each scenario is applied in this article. This new evaluation mechanism can make full use of the information on the solution in each simulation.

Third, we combine the simplified simulation approach with a clustering estimation of distribution algorithm (EDA) [26]. EDA is a probability based EA which has a great potential in dealing with uncertainties. It establishes a probability model according to the distribution of the promising individuals. In the group insurance portfolio problem, since far apart portfolio plans may have similar profit, there are peaks that far from each other in the problem. To further improve the effectiveness of the algorithm, two kinds of clustering method [27]: crowding clustering and speciation clustering are applied during model construction. The promising individuals in the generation are divided into several independent clusters based on their location. The probability model for each cluster is established according to the distribution of the individuals in that cluster. Then the new population is sampled by the probability model for each cluster. With the clustering method, more peaks of the problem can be captured, which helps the optimization process to achieve better performance.

This article proceeds as follows. Section II summarizes the application of EA for optimization under uncertainty. In Section III, a group insurance portfolio problem is illustrated as an instance of financial decision making problems with strong uncertainty. Feature analysis of the group insurance portfolio problem is given in Section IV. Section V elaborates the proposed CEDA-SS. Section VI presents the experimental results and analyses

of the proposed CEDA-SS to validate its effectiveness. Finally, the conclusions are drawn in Section VII.

## II. EA FOR OPTIMIZATION UNDER UNCERTAINTY

### A. Application of MC in Uncertain Decision Making

1) *Application of MC in Noisy Fitness Function*: In noisy fitness function, the evaluation of the fitness function is disturbed by noise. The most common method to reduce this kind of disturbance is to sample the fitness value of a solution by MC simulation for several times and calculating the average value of them [18].

Efforts have been made to improve the efficiency of the MC method by reducing the number of evaluation times. Sano and Kita [28] proposed a memory-based fitness evaluation GA applying history of search to reduce fitness evaluation times. Branke and Schmidt [29], [30] took the inherent noise in the optimization problem into account in the selection operator to reduce the number of evaluation times. Pietro *et al.* [31] proposed an algorithm that determines the number of evaluation times of each candidate according to the amount of noise at that point in the search space.

In addition to the above methods, there are literatures that improve the effectiveness of the optimization by adaptively selecting the number of evaluation times. Cantúpaz [32] proposed an adaptive sampling strategy that adjusts the number of evaluation times according to the uncertainty of deciding between two specific individuals. Aizawa and Wah [33] introduced an adaptive mechanism that increases the number of evaluation times with iterations times.

2) *Application of MC in Searching for Robust Solutions*: In searching for robust solutions, the designed variables are perturbed after the optimization process. The most common method to calculate the expected fitness value of the problem is to sample the fitness values with perturbations by MC simulation for several times and calculating the average value of them [34].

Efforts have been made to improve the performance of the algorithms for searching for robust solutions. Branke [35] proposed a novel method that estimates the expected fitness value by similar individuals in former generations to reduce the number of sampling times. Beyer and Sendhoff [36] introduced two evolution strategies that increase the population size when the residual error to the optimizer state has been reached. Kruisselbrink *et al.* [37] incorporated an archive maintenance scheme in the evaluation mechanism to generate locally well-spread distributions of archive points. Fei *et al.* [19] followed this idea and proposed an archive sample approximation strategy based on Wasserstein distance metric and improved the performance in three ways.

### B. Application of EA in Financial Decision Making

1) *Application of EA in Insurance Decision Making*: The most widely application field of EA in insurance decision making problem is the optimization of investment strategies. Chen and Liao [38] applies genetic algorithms (GA) to optimize the parameters in a new proposed piecewise linear goal-directed

constant proportion portfolio insurance strategy. After that, Chen and Lin [39] followed their work and proposed a relation-based GA named relational genetic algorithm to optimize the partitioned portfolio insurance strategy. Then, Yu *et al.* [40] developed a multi-phase evolution strategy to search for optimal asset allocations for the insurer.

Except for the optimization of investment strategy, EAs are also applied in other fields of insurance decision making problem. Duma *et al.* [41] utilized GA and a new proposed hybrid multi-layered artificial immune system to solve the problem of missing data in datasets with numerous variables. Salcedo-Sanz *et al.* [42] applied genetic programming to predict the insolvency in non-life insurance companies.

2) *Application of EA in Stock Decision Making*: The most widely application fields of EA in stock decision making problem are the prediction of stock price index and the optimization of asset allocation. Kim and Han [43] applied GA to improve the learning algorithm and reduce the complexity in feature space in artificial neural networks. They used the improved artificial neural networks to predict the stock price index. Kwon and Moon [44] applied GA to optimize neural network and proposed a hybrid neurogenetic system for stock forecasting. Rout *et al.* [45] proposed a forecasting model based on adaptive linear combiner and differential evolution (DE) for stock index prediction.

Hagströmer *et al.* [46] employed DE to solve full-scale optimization based asset selection for stock portfolio optimization. Chang and Shi [47] applied the investment satisfied capability index to evaluate individual stock performance and then utilize a particle swarm optimization (PSO) with moving interval windows to find the optimal investment allocation of the stocks. Yu *et al.* [48] proposed a stock selection model with both discrete and continuous decision variables and developed a sigmoid-based mixed discrete-continuous differential evolution algorithm to deal with this model.

## III. AN INSTANCE OF FINANCIAL DECISION MAKING PROBLEMS - GROUP INSURANCE PORTFOLIO PROBLEM

In this section, a specific sample of financial decision making problem is given. The concept of insurance portfolio was investigated quite intensively in recent years. Insurance portfolio problems study the decision making among several kinds of endowment insurance policies and hospitalization insurance policies. The construction of the decision making model of insurance portfolio is based on a huge number of accumulated historical data.

Insurance portfolio problem for single insured has been investigated in [21]. In this article, we extend the model to a group case and consider the simulation reward of the portfolio plan as the fitness value. The symbols used in the model are listed in Table I. The optimization problem is formulated as (1) and (2)

$$f(t) = \sum_{k=1}^n [V^k(t) + Z^k(t)] + C(t) \quad (1)$$

$$\max J(\mathbf{X}^k, \beta^k) = \max \sum_{t=1}^T f(t) \quad (2)$$



TABLE I  
SYMBOLS IN THE MODEL

$n_1$	The number of endowment policies
$n_2$	The number of hospitalization policies
$n$	The number of insured in the group
$f(t)$	The profit at time $t$
$\beta^k$	The kind of hospitalization insurance the $k$ -th insured in the group to invest ( $\beta^k = 0$ implies does not buy any hospitalization insurance)
$x_{ij}^k$	The premium of the $j$ -th payment period of the $i$ -th endowment insurance for the $k$ -th insured in the group
$V^k(t)$	The cash value of the $k$ -th insured at time $t$
$Z^k(t)$	The death compensation of the $k$ -th insured at time $t$
$C(t)$	The total disposable cash for the whole group at time $t$
$V_{ij}^k(t)$	The cash value of the $i$ -th endowment policy with the $j$ -th payment period for the $k$ -th insured at time $t$
$m_i$	The number of payment period for the $i$ -th endowment policy
$\tau^k$	The lifetime of the $k$ -th insured
$p^k(t)$	The mortality rate of the $k$ -th insured at age $t$
$p_s^k(t)$	The incidence rate of the $k$ -th insured for diseases of degree $s$ at age $t$
$\eta_{1ij}^k(t_1, t)$	The increase rate of the cash value of the $i$ -th endowment insurance with the $j$ -th payment period for the $k$ -th insured at age $t$ with the purchase at age $t_1$
$\eta_{2ij}^k(t_1, t)$	The death compensation rate of the $i$ -th endowment insurance with the $j$ -th payment period for the $k$ -th insured at age $t$ with the purchase at age $t_1$
$\delta_{\beta}^k(t)$	The premium of hospitalization policy $\beta$ for the $k$ -th insured at age $t$
$t_0^k$	The initial age of the $k$ -th insured in the group
$I_1$	The total disposable income for the whole group
$I_2^k(t)$	The disposable income of the $k$ -th insured at time $t$
$v_{ij}^k(t)$	An indicator of whether the payment period is terminated
$w_s$	An indicator of whether or not the $k$ -th insured suffers from the diseases in the $s$ -th group
$q_{ij}$	The latest time of purchase for the $j$ -th payment period of the $i$ -th endowment insurance
$l_{ijt}$	The minimum amount for purchase of the $j$ -th payment period of the $i$ -th endowment insurance at age $t$
$T$	The duration to be considered
$z_s$	The medical expense of diseases at degree $s$

Here,  $f(t)$  is the profit at time  $t$ .  $X^k$  is an irregular matrix indicating the endowment policies purchase amount of the  $k$ -th insured.  $\beta^k$  indicates the selection of hospitalized policy of the  $k$ -th insured.  $V^k(t)$  is the cash value of the  $k$ -th insured at time  $t$ .  $Z^k(t)$  denotes the death compensation of the  $k$ -th insured at time  $t$ .  $C(t)$  is the total disposable cash for the whole group at time  $t$ . The objective of the optimization problem is to maximize the total profit of the whole group for a given period  $T$ .

The cash value  $V^k(t)$  in (1) is the sum of the cash value of each endowment policy for the  $k$ -th member, i.e.,

$$V^k(t) = \sum_{i=1}^{n_1} \sum_{j=1}^{m_i} V_{ij}^k(t). \quad (3)$$

where  $n_1$  denotes the number of endowment policies and  $m_i$  denotes the number of payment period for the  $i$ -th endowment policy.  $V_{ij}^k(t)$ , which is determined by MC simulation given by (4), is the cash value of each endowment policy for the  $k$ -th insured at time  $t$

$$V_{ij}^k(t) = \begin{cases} x_{ij}^k \eta_{1ij}^k(t_0^k, t_0^k + t), & \text{if } t = 1 \\ V_{ij}^k(t-1) + x_{ij}^k [\eta_{1ij}^k(t_0^k, t_0^k + t) - \eta_{1ij}^k(t_0^k, t_0^k + t - 1)], & \text{if } 1 < t < \tau^k \\ 0, & \text{if } t \geq \tau^k \end{cases} \quad (4)$$

---

**Algorithm 1:** Fitness Evaluation Process.

---

**Input:** purchase amount of endowment policy  $X^k$ , choice of hospitalization policy  $\beta^k$

**Output:** total profit

---

```

1: for  $t = 1$  to  $T$ 
2:   for  $k = 1$  to  $n$ 
3:     if insured  $k$  is alive at  $t - 1$ 
4:       if insured  $k$  is alive at  $t$ 
5:         calculate cash value by (4)
6:         subtract the endowment premium from  $C(t)$ 
7:         subtract the medical expense from  $C(t)$ 
8:         subtract the hospitalization premium from  $V^k(t)$ 
9:         if  $C(t) < 0$  or  $V^k(t) < 0$ 
10:            $f(t) = 0$ 
11:           exit
12:         end if
13:       else
14:         calculate death compensation by (6)
15:       end if
16:     add the corresponding reward to  $f(t)$ 
17:   else
18:     exit
19:   end if
20: end for
21: add the corresponding income to  $C(t + 1)$ 
22: end for

```

---

Here,  $\tau^k$  is the lifetime of the  $k$ -th insured. The value of  $\tau^k$  is randomly generated by Roulette-wheel selection [30] according to the death rate  $p^k(t)$  of each insured. That is, randomly generate a number  $r$  in the range of (0,1) for each member,  $\tau^k$  is the minimum integer that the sum from  $p^k(0)$  to  $p^k(\tau^k)$  is larger than  $r$ . For example, if  $r = 0.001324$ , since  $p^k(0) = 0.000722$  and  $p^k(1) = 0.000603$ ,  $p^k(0) + p^k(1) = 0.001325 > r$ , it can be drawn that  $\tau^k = 1$ . The value of increase rate  $\eta_{1ij}^k(t_1, t)$  is randomly generated by normal distribution according to the mean value of the increase rate of each endowment policy from time  $t_1$  to  $t$  for the  $k$ -th insured  $\bar{\eta}_{1ij}^k(t_1, t)$  and its standard deviation  $\tilde{\eta}_{1ij}^k(t_1, t)$ :

$$\eta_{1ij}^k(t_1, t) = N(\bar{\eta}_{1ij}^k(t_1, t), \tilde{\eta}_{1ij}^k(t_1, t)). \quad (5)$$

The policyholder can get the reward of cash value before the death of the insured. The cash value cannot be taken out for consumption except for paying the premium of hospitalization policy. The cash values  $V_{ij}^k(t)$  are taken out in descending order to pay the premium  $\delta_{\beta}^k(t)$  for the  $k$ -th insured till  $\delta_{\beta}^k(t)$  is paid off. If  $\delta_{\beta}^k(t)$  cannot be fulfilled with all the cash values taken out (i.e.,  $\delta_{\beta}^k(t) > V^k(t)$ ) for any insured at any time  $t$ , the choice of hospitalization policy is improper. The portfolio strategy thus becomes infeasible and the fitness value is set to be 0. For the endowment policy whose cash values are taken to pay the premium of hospitalization, its cash values are reduced correspondingly. For example, if the premium  $\delta_{\beta}^k(t) = 5000$ ,  $V_{12}^k(t) = 4000$  and  $V_{11}^k(t) = 3000$  are the maximum two cash value among  $V_{ij}^k(t)$ . 4000 in  $V_{12}^k(t)$  and 1000 in  $V_{11}^k(t)$  are

taken out to pay the hospitalization premium. After that,  $V_{12}^k(t)$  becomes 0 and  $V_{11}^k(t)$  becomes 2000.

Similarly, the death compensation  $Z^k(t)$  can be gained when the insured dies. The amount of  $Z^k(t)$  at time  $t$  is calculated by summing up the death compensations of all the endowment policies for the  $k$ -th insured:

$$Z^k(t) = \begin{cases} \sum_{i=1}^{n_1} \sum_{j=1}^{m_i} x_{ij} \eta_{2ij}^k(t_0^k, t_0^k + t), & \text{if } t = \tau^k \\ 0, & \text{if } t \neq \tau^k. \end{cases} \quad (6)$$

The value of death compensation rate  $\eta_{2ij}^k(t_1, t)$  is randomly generated by normal distribution according to the mean value of death compensation rate of the  $i$ -th endowment policy with the  $j$ -th payment period for the  $k$ -th insured at time  $t$ ,  $\bar{\eta}_{2ij}^k(t_1, t)$  and its standard deviation  $\tilde{\eta}_{2ij}^k(t_1, t)$ :

$$\eta_{2ij}^k(t) = N(\bar{\eta}_{2ij}^k(t), \tilde{\eta}_{2ij}^k(t)). \quad (7)$$

Denoted the initial disposable amount for investment as  $I_1$  and the total disposable income for the whole group at time  $t$  as  $I(t)$ . The amount of disposable cash value  $C(t)$  at time  $t$  is calculated by (8) and (9)

$$C(t) = \begin{cases} I_1, & \text{if } t = 0 \\ C(t-1) + I(t) - \sum_{k=1}^n \left( \sum_{i=1}^{n_1} \sum_{j=1}^{m_i} v_{ij}^k(t) x_{ij}^k - \sum_{s=1}^{n_2-\beta^k} w_s z_s \right), & \text{otherwise} \end{cases} \quad (8)$$

$$I(t) = \sum_{k=1}^n I_2^k(t + t_0^k) \quad (9)$$

where  $I_2^k(t)$  is the disposable income of the  $k$ -th insured at time  $t$ .  $v_{ij}^k(t) \in \{0, 1\}$  is an indicator of whether the payment period is terminated.  $z_s$  is the medical expense for the diseases in the  $s$ -th group.  $w_s \in \{0, 1\}$  is an indicator of whether or not the  $k$ -th insured suffers from the diseases in the  $s$ -th group. The value of  $w_s$  is determined by the incidence rate for diseases in the  $s$ -th group at time  $t$ ,  $p_s^k(t)$ . Generate a random number  $r$  from  $(0, 1)$ . If  $r > p_s^k(t)$ ,  $w_s = 0$ ; If  $r \leq p_s^k(t)$ ,  $w_s = 1$ .

Here, the value of  $p^k(t)$ ,  $p_s^k(t)$ ,  $z_s$ ,  $\bar{\eta}_{1ij}^k(t)$ ,  $\bar{\eta}_{1ij}^k(t)$ ,  $\bar{\eta}_{2ij}^k(t)$  and  $\tilde{\eta}_{2ij}^k(t)$  are all derived from historical data.

The pseudocode of the fitness evaluation process is given in Algorithm 1.

#### IV. FEATURE ANALYSIS OF GROUP INSURANCE PORTFOLIO PROBLEM

As an example of financial optimization problem, since many of the parameters in the model are calculated based on historical statistics data, the insurance portfolio problem considered in this article involves strong uncertainty. In order to handle such uncertainty, in this section we make an insight analysis on the feature of uncertainty in this problem. In the analysis, we intend to answer three questions:

- 1) With the presence of strong uncertainty, is it still meaningful to perform optimization?
- 2) How many Monte Carlo simulations are needed to stabilize the estimation of the fitness function value?
- 3) Can we reduce the number of simulations?

##### A. Feature Analysis of Solutions

Despite the strong uncertainty in the considered problem, given two solutions, if we can still tell apart which solution is better, then there is still room for optimization in this problem. In other words, it is meaningful to perform optimization only if some solutions perform better than the others in general. For those uncertain problems that the performances of the solutions are difficult to be distinguished, the optimization process is difficult to carry on since different solutions may have advantages in different scenarios. Therefore, it is important to classify whether the problem is optimizable.

To verify whether the solutions in this problem can still be distinguished, the features of several different solutions in situation  $\{t_0 = 10, 35, 40, T = 30\}$  are plotted in Fig. 1. Firstly, five solutions with different fitness value at different stages of evolution are randomly selected as the object of study. Then, the fitness values of these five solutions are calculated in the same one thousand simulation scenarios. Finally, the box-plots are drawn according to these fitness values.

Fig. 1 shows that the differences in performance among different solutions are significant, which means that it is easy to distinguish good solutions with others in this problem. In other words, it makes sense to optimize the group insurance portfolio problem. Moreover, as a representative of financial decision making problem, the characteristic of group insurance portfolio problem is reflected in Fig. 1. First, it can be discovered in Fig. 1 that the distribution of the solutions is irregular which means that the uncertainties in group insurance portfolio problem do not obey certain statistical distributions. Second, the group insurance portfolio problem is under strong uncertainty. The spans of the solutions are very large and there are many outliers in Fig. 1.

##### B. Stability Analysis of Solutions

It has been discovered that the group insurance portfolio problem is under strong uncertainty. Therefore, a large amount of simulation number may be needed to approximate the expectation value of the profit. To further analyze how many simulation numbers are needed to stabilize the estimation of the fitness function value, the stability analysis of solutions in different situations are given in Fig. 2. The blue lines and the green lines represent two different solutions in each situation. The mean fitness values of the solutions from one simulation to 500 simulations are given. It can be discovered that more than 200 times of simulations are needed to make the mean fitness value of the solutions stable. Especially when  $T = 10$ , since the payment periods of some endowment policies are longer than 10 years, the policyholder cannot get a considerable reward in such a short duration if investing in those endowment policies with long payment periods. Once such a situation occurs, the reward will be lowered, resulting in an unstable curve. Therefore, if we want to apply the mean fitness value of simulation result to evaluate the solutions during the evolution process, more than 200 times of simulations are needed. This method wastes a lot

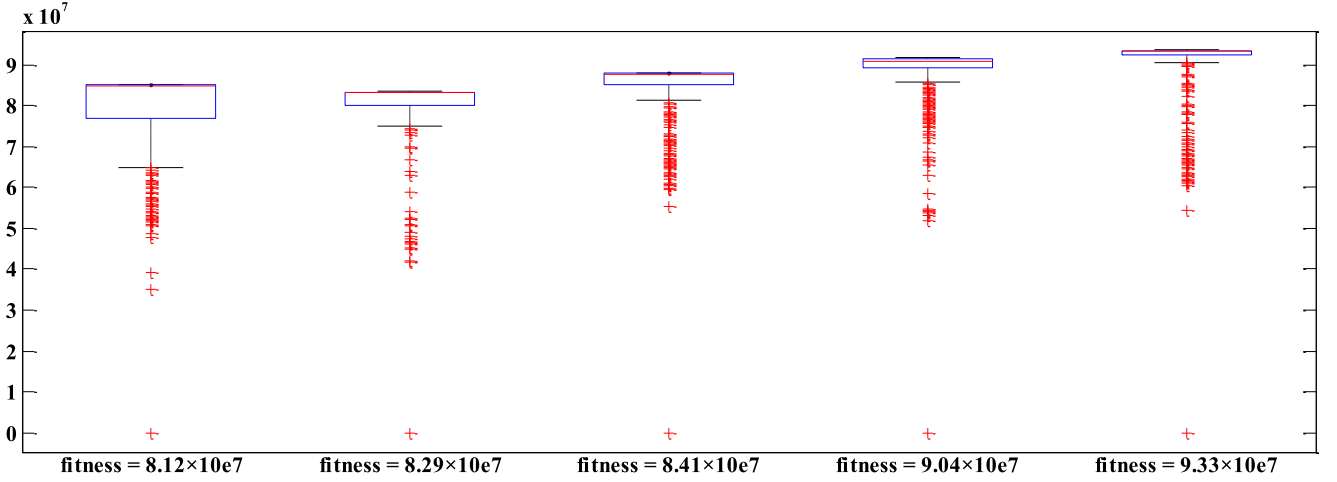
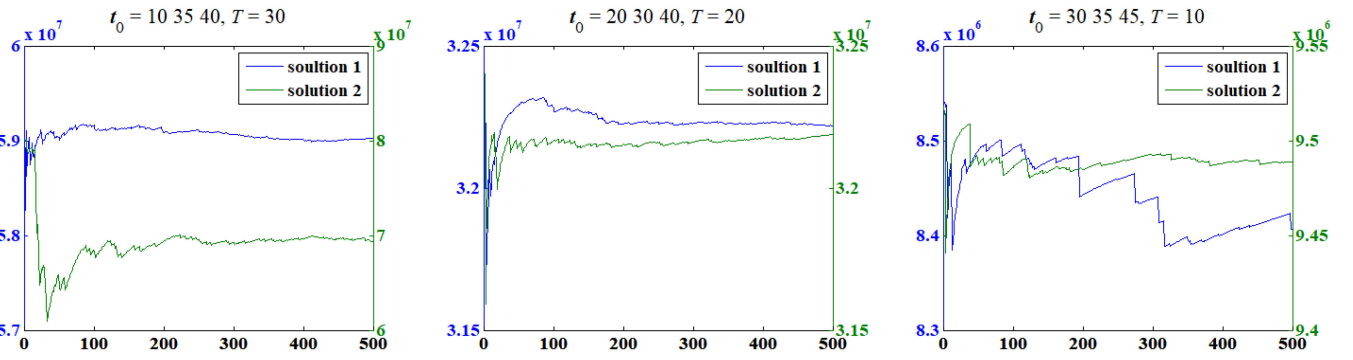
Fig. 1. Box-plot of different solutions in situation  $\{t_0 = 10, 35, 40, T = 30\}$ .

Fig. 2. Stability analysis of solutions in different situations.

of computational time and resources, which greatly reduces the efficiency of the algorithm.

### C. Correlation Analysis of Solutions

In order to search for an approach to improve the efficiency of the algorithm, we analyze the correlation of the solutions in different scenarios. The correlations of the solutions are determined by the correlation coefficient [49]. Correlation coefficient is an indicator with value in interval  $[-1, 1]$ . The calculation of the correlation coefficient is given in (10):

$$\rho_{XY} = \frac{\text{Cov}(X, Y)}{\sqrt{DX} \cdot \sqrt{DY}} = \frac{E(XY) - EX \cdot EY}{\sqrt{DX} \cdot \sqrt{DY}} \quad (10)$$

Here,  $\text{Cov}(X, Y)$  is the covariance of two variables while  $D(\cdot)$  stands for the variance of the variable and  $E(\cdot)$  stands for the expectation of the variable. Correlation coefficient reflects the correlation degree between two solutions. A negative correlation coefficient indicating that the solutions are negatively correlated while a positive correlation coefficient indicating that the solutions are positively correlated. The closer the correlation coefficient is to one, the greater the correlation between the solutions is. When the solutions are completely linear correlation, the

TABLE II  
AVERAGE CORRELATION COEFFICIENT OF SOLUTIONS IN DIFFERENT STAGE

	Initialization	Early stage	Mid-stage	Later stage
$t_0 = 10, 35, 40$ $T = 30$	0.9734	0.9784	0.9791	0.9508
$t_0 = 20, 30, 40$ $T = 20$	0.9407	0.9084	0.9334	0.9410
$t_0 = 30, 35, 45$ $T = 10$	0.7826	0.7308	0.7876	0.9822

value of correlation coefficient equals to one. When the solutions are completely independent, the value of correlation coefficient equals to zero.

Five solutions in thirty independent runs are randomly selected from four different stages during the evolution process: initialization stage, early stage of evolution, mid-stage of evolution and later stage of evolution. Then the fitness value of the solutions is simulated in twenty scenarios according to Algorithm 1. The correlation coefficients of the solutions are calculated according to the fitness value in the simulation scenarios. The average values of the correlation coefficients in thirty independent runs for the five solutions in four different stages of three situations are given in Table II.

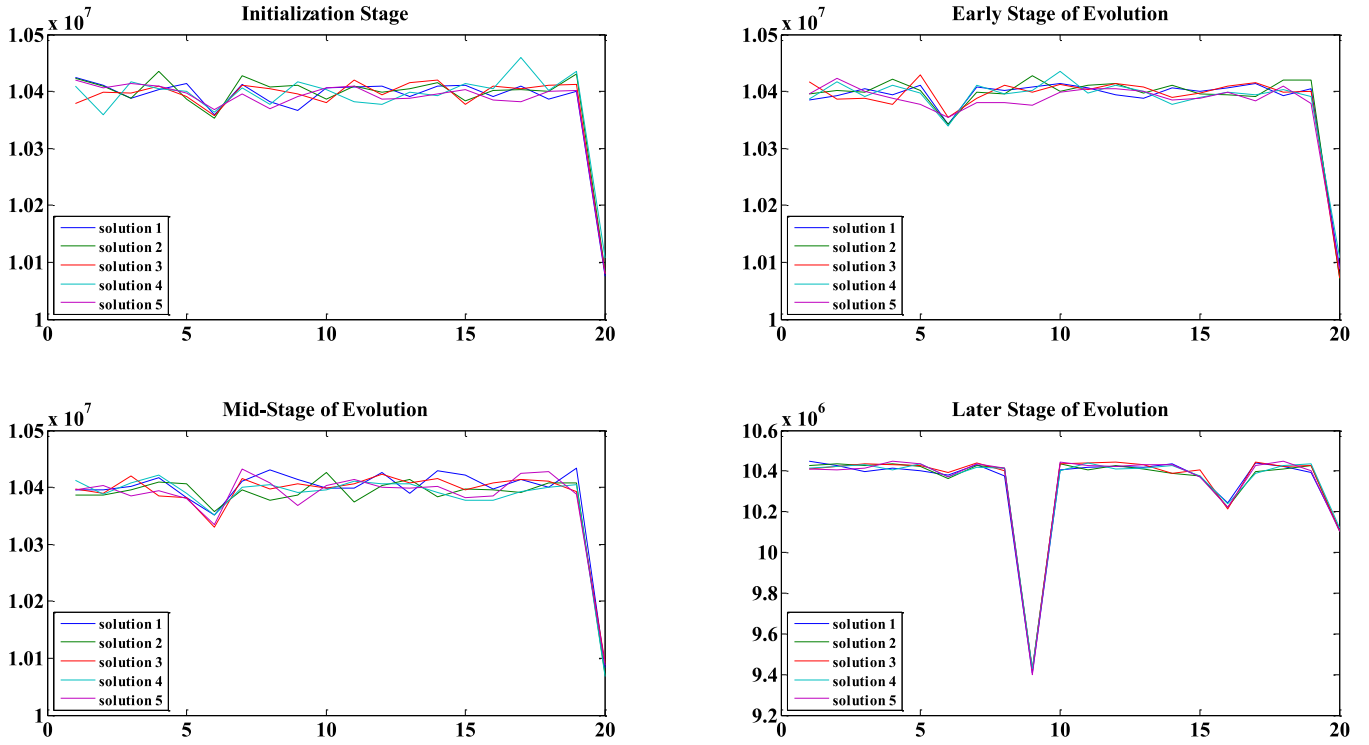


Fig. 3. Simulation result for solutions from different stages in situation  $\{t_0 = 30 \ 35 \ 45, T = 10\}$ .

In the first two situations, all of the average correlation coefficient values are very large in every stage. In the last situation, the correlation coefficient values are not so large in the beginning, but as the evolution process progresses, the values become larger. It can be discovered that the average correlation coefficient of the four stages in the three situations are all very close to 1, which indicates that the degree of correlation of the solutions are very high in all of the situations in the proposed group insurance portfolio problem.

To show the correlation of the solutions more clearly, simulation result for solutions from different stages in situation  $\{t_0 = 30 \ 35 \ 45, T = 10\}$  in one of the runs is given in Fig. 3. It clearly shows that the performances of the solutions are synchronous, especially at the later stage of evolution. In negative scenarios, the performances of the solutions are all poor while in positive scenarios, all of the solutions perform relatively well. Since the performances of the solutions are highly relevant to the scenarios, it can be concluded that the group insurance portfolio problem considered in this article is scenario-relevant. Based on this feature, we can design a corresponding method to reduce simulation number and improve the efficiency.

In conclusion, the uncertainties in the group insurance portfolio problem can be summarized as follows. Firstly, the solution is distinguishable despite the uncertainties in this problem are strong. Secondly, a large number of simulations are needed to stabilize the estimation of the fitness function value. Lastly, if the expectation value of one solution is larger than another solution, the performance of this solution is better than that solution in most of the scenarios.

## V. CLUSTERING ESTIMATION OF DISTRIBUTION ALGORITHM WITH SIMPLIFIED SIMULATION

An adaptive estimation of distribution algorithm (AEDA) [25] has been proven to be effective to deal with single-insured insurance portfolio problem. In this article, we extend the problem to the group case and employ a simplified simulation approach to calculate the fitness value of the problem. Moreover, a clustering method is utilized to further improve the performance of the algorithm.

### A. Simplified Simulation Approach

A simplified simulation approach is proposed in this article to deal with the uncertainties in the group insurance portfolio problem. As analyzed in Section IV.C, the group insurance portfolio problem is scenario-relevant. The correlation coefficients of solutions in different scenarios are high. That is to say, a good solution performs better than a bad solution in the same scenario for most of the situation. Therefore, a simplified simulation approach that applies only one scenario to define the quality of the solution in each generation during the optimization process is proposed. This approach can save a large amount of computational costs and improve the efficiency of the algorithm.

### B. Adaptive Estimation of Distribution Algorithm

At the beginning of AEDA, an initialization mechanism with specific constraint handling is employed to ensure that each solution is feasible for the group insurance portfolio problem. In



this mechanism,  $x_{ij}^k$  is initialized in sequence. The upper bound of  $x_{ij}^k$  depends on the disposable cash within the payment period, which changes dynamically during the initialization process. Once  $x_{ij}^k$  has been decided, the disposable cash in the payment period should be reduced accordingly. So that the upper bound for the investment amount to the rest of endowment policies is reduced. To ensure that the initialization process is carried out randomly, the order of determining the investment amount to each endowment policy is disrupted for each solution.

After that, the elite individuals with the best  $NP_{best}$  fitness value are selected to establish the probability model. For the investment amount  $x_{ij}^k$  of the  $j$ -th payment period of the  $i$ -th endowment policy for the  $k$ -th insured, the probability model is established using the mean value  $\mu_{ij}^k$  and the standard deviation  $\sigma_{ij}^k$  calculated as (11) and (12)

$$\mu_{ij}^k = \frac{\sum_{g=1}^{NP_{best}} x_{ij}^k}{NP_{best}} \quad (11)$$

$$\sigma_{ij}^k = \sqrt{\frac{\sum_{g=1}^{NP_{best}} (x_{ij}^k - \mu_{ij}^k)^2}{NP_{best}}}. \quad (12)$$

In terms of the choice of hospitalization policy, an adaptive selection mechanism is applied to choose the solution according to their performance in the former generation. Denoted the number of individuals with hospitalization policy  $i$  in the whole population as  $count_i$  and the number of individuals with hospitalization policy  $i$  in the elite population as  $count\_best_i$ . The probability for the  $i$ -th hospitalization policy to be chosen in the next generation is calculated as (13):

$$\beta_{i-p} = \left[ \sum_{j=0}^{n_2} \frac{count\_best_j}{count_j} \right]^{-1} \frac{count\_best_i}{count_i}. \quad (13)$$

Similarly, the decision between the Gaussian distribution and the Cauchy distribution in sampling new population is according to the performance of these two distributions in the former generation. The probability of choosing Gaussian distribution to generate the new population is given in (14):

$$g-p = \left[ \sum_{i=0}^1 \frac{top\_d_i}{count\_d_i} \right]^{-1} \frac{top\_d_0}{count\_d_0}. \quad (14)$$

Here,  $count\_d_i$  is the number of individuals sampling from the two distributions in the whole population.  $top\_d_i$  is the number of individuals sampling from the two distributions in the elite population. Gaussian distribution is denoted as  $i = 0$  and Cauchy distribution is denoted as  $i = 1$ .

In the sampling process, a constraint handling mechanism similar to initialization process is employed to ensure that the individual in the generated population is feasible for the given group insurance portfolio problem.

Finally, a local search strategy is employed to improve the effectiveness of the algorithm. New individuals are sampled in the neighbor of the best-so-far individual. The choice of hospitalization policy is made according to a probability calculated by (13) and the value of  $x_{ij}^k$  is generated with a Gaussian distribution

---

**Algorithm 2: Crowding Clustering [26].**


---

**Input:** population  $P$ , cluster number  $c_n$ , cluster size  $c_s$   
**Output:** a set of crowds

---

- 1: **Initialization**
  - 2: **for**  $i = 1$  **to**  $c_n$
  - 3: generate a reference point  $R$  randomly
  - 4: calculate the distance from other individual to  $R$
  - 5: choose the nearest individual to  $R$ , denoted as  $P_{near}$
  - 6: calculate the distance from other individual to  $P_{near}$
  - 7: sort the rest individuals by the distance to  $P_{near}$
  - 8: choose the  $c_s - 1$  nearest individuals to  $P_{near}$  and form a crowd together with  $P_{near}$
  - 9: eliminate these  $c_s$  individuals from the population
  - 10: **end for**
- 

---

**Algorithm 3: Speciation Clustering [26].**


---

**Input:** population  $P$ , cluster number  $c_n$ , cluster size  $c_s$   
**Output:** a set of species

---

- 1: **Initialization**
  - 2: sort the population by fitness value
  - 3: **for**  $i = 1$  **to**  $c_n$
  - 4: choose the best individual, denoted as  $P_{best}$
  - 5: calculate the distance from other individual to  $P_{best}$
  - 6: sort the rest individuals by the distance to  $P_{best}$
  - 7: choose the  $c_s - 1$  nearest individuals to  $P_{best}$  and form a crowd together with  $P_{best}$
  - 8: eliminate these  $c_s$  individuals from the population
  - 9: **end for**
- 

given in (15):

$$x_{ij}^k \sim N(x\_best_{ij}^k, \sigma_i) \quad (15)$$

where  $x\_best$  is the best-so-far individual which is updated in each generation.  $\sigma_i$  is the scale factor depends on the magnitude of the solution.

### C. Clustering Method

Since the correlation degrees of the solutions are very high in the proposed problem, only one simulation is needed in each generation during the optimization process to identify the quality of the solution. Therefore, a large population size can be set to find the optimal solution easier. With a large population size, a clustering method is applied to divide the population into several clusters. After that, the establishment of the probability model and the sampling are executed within each cluster.

In this article, two widely applied clustering methods: crowding clustering and speciation clustering [50] are considered. The number of the cluster is denoted as  $c_n$  and the size of each cluster is denoted as  $c_s$ .

1) *Crowding Clustering:* For each crowd, a reference point  $R$  is randomly generated firstly. Then, the nearest individual to  $R$ , which is denoted as  $P_{near}$ , is selected as the seed in the crowd. The rest individuals in the population are sorted according to



**Algorithm 4:** CEDA-SS.

**Input:** population size  $NP$ , cluster number  $c_n$ , cluster size  $c_s$  iteration time  $G$ , elite individual number  $NP_{best}$   
**Output:** an optimized portfolio plan

- 1: **Initialize**  $X[k]$
- 2: find the best solution  $x[k]_{best}$
- 3: **for**  $g = 1$  **to**  $G$
- 4:   calculate fitness values with the same scenario
- 5:   find the best  $NP_{best}$  solutions
- 6:   divide the solutions into  $c_n$  clusters
- 7:   **for**  $i = 1$  **to**  $c_n$
- 8:     establish probability model by (11) – (13)
- 9:     calculate  $g_p$  by (14)
- 10:    sample a new population
- 11:   **end for**
- 12:   local search
- 13: **end for**

their Euclidean distance to  $P_{near}$ . The closest  $c_s - 1$  ones are selected to form the whole crowd together with  $P_{near}$ . Finally, these  $c_s - 1$  closest individuals and  $P_{near}$  are eliminated from the population. The above processes are executed repeatedly until each individual in the population is assigned to a corresponding crowd.

The pseudocode of crowding clustering is given in Algorithm 2.

2) *Speciation Clustering*: For each species, the individual with the best fitness value in the population, which is denoted as  $P_{best}$ , is selected as the seed. The rest individuals in the population are sorted according to their Euclidean distance to  $P_{best}$ . The closest  $c_s - 1$  ones are selected to form the whole species together with  $P_{best}$ . Finally, these  $c_s - 1$  closest individuals and  $P_{best}$  are eliminated from the population. The above processes are executed repeatedly until each individual in the population is assigned to a corresponding crowd.

The pseudocode of speciation clustering is given in Algorithm 3.

Both of these two clustering methods divide individuals into clusters according to Euclidean distance [51]. The difference between them is the selection of the seed. In crowding clustering method, the individual closest to a random point in the search space is selected while in speciation clustering method, the individual with the best fitness value is selected.

The overall pseudocode of the proposed CEDA-SS is given in Algorithm 4.

## VI. EXPERIMENTAL RESULTS

### A. Experiment Settings

In order to validate the effectiveness of the proposed CEDA-SS, we do experiments on the following three situations with different initial age and duration:

- 1)  $t_0 = 10, 35, 40$ ;  $T = 30$ ;
- 2)  $t_0 = 20, 30, 40$ ;  $T = 20$ ;
- 3)  $t_0 = 30, 35, 45$ ;  $T = 10$ ;

TABLE III  
SETTINGS OF INCOME

Age	Yearly Income
< 20	12000
20 ~ 24	25000
25 ~ 29	40000
30 ~ 34	50000
35 ~ 39	53000
40 ~ 44	50000
45 ~ 49	48000
50 ~ 60	45000
> 60	20000

TABLE IV  
DETAILED SETTINGS OF THE ENDOWMENT POLICIES

	Payment Period ( $y_{ij}$ ) and Latest Purchasing Time ( $q_{ij}$ )					Minimum of the face amount ( $c_i$ )
A	5		10			20,000
	75		70			
B	1	5	10	18	25	10,000
	75	65	60	60	55	
C	5		10			10,000
	75		70			
D	6	10	15	20		5,000
	65	60	55	55		
E	1	5	10			1,500
	80	80	75			

TABLE V  
DETAILED SETTINGS OF THE HOSPITALIZATION POLICIES

	Coverage scope	Premium ( $\delta_k(t_0)$ )
A	critical illness	lowest
B	critical and serious illness	medium
C	critical, serious and general illness	highest

The initial disposable amount is set as 20,000. The disposable income at time  $t$  is given in Table III according to [25]. The detailed settings of the endowment policies and hospitalization policies are given in Table IV and Table V according to [25]. The data of insurance policies are collected from real world insurance company.

To validate the effectiveness of the proposed method, we design three sets of experiments for comparisational analysis. Firstly, to validate that considering simulation results as fitness value is more accurate than considering the approximated expectation reward as fitness value, comparisons are made between original AEDA and AEDA with simplified simulation (AEDA-SS). Secondly, to validate the rationality of the simplified simulation approach, we compare it with the algorithm that applies different scenarios in each generation and the algorithm that applies the same scenario in the whole optimization process. To further validate its effectiveness, we compare AEDA-SS with an adaptive setting mechanism given in [33]. Thirdly, to validate the necessity of the clustering methods, comparisons are made between CEDA-SS and AEDA-SS. To further validate the effectiveness of the proposed algorithm, comparisons are made between CEDA-SS and two state-of-the-art evolutionary algorithm for uncertain optimization: dual-environmental particle swarm optimizer (DEPSO) [7] and EDA with stochastic local search (EDASLS) [52].

To reduce statistical error, each algorithm runs 30 times independently for each situation. The fitness value is evaluated  $2 \times 10^5$  times. The simulation time to evaluate the solution during optimization process is set as 1 and the population size is set as 1000. The percentage of elite individual is set as 45% [25]. The cluster number is set as 10 in CEDA-SS [50]. The inertia weight in DEPSO changes from 0.9 to 0.4. Accelerate coefficient in DEPSO is set as 3.2 [7]. The probability of executing local search in EDASLS is set as 0.1 [52].

In order to evaluate the performance of the algorithms, a new evaluation mechanism based on Wilcoxon rank sum test [53] is proposed in this article. In this evaluation mechanism, the fitness value simulation process is executed in ten thousand scenarios. In each scenario, the Wilcoxon rank sum test is made to compare the performance between two algorithms. Since the uncertainty of the problem considered in this article is quite strong, the confidence level is set as 80% in the comparisons. The number of outperformed scenarios of each algorithm is applied to evaluate the performance of the algorithm. In this article, the number of scenarios that the latter algorithms perform better than the first algorithm listed in the table is denoted as b-number. The number of scenarios that the latter algorithms perform worse than the first algorithm listed in the table is denoted as w-number. The number of scenarios that the latter algorithms perform equivalent to the first algorithm listed in the table is denoted as e-number. The maximum fitness value of the algorithms among thirty runs in ten thousand scenarios is denoted as max. The average fitness value of the algorithms among thirty runs in ten thousand scenarios is denoted as mean. To further evaluate the performance of the algorithm, the efficiency and robustness of the algorithms are also analyzed.

### B. Effectiveness of MC Simulation

Existing work considered the statistical average value of each parameter in the model to approximate the expectation reward of the problem [25]. By considering this approximated expectation reward as the objective function, the fitness value of the algorithm is deterministic. However, it cannot estimate the expectation reward accurately. By considering the reward of the portfolio plan in the MC simulation of the investment, a more accurate value of the expectation reward is approximated. The modification of the objective function makes the algorithm more practical.

To validate the effectiveness of the modification of the objective function, we compare the performance of the original AEDA with AEDA-SS. The comparison results are shown in Table VI. It can be discovered that in the first and second situation, AEDA-SS outperforms AEDA in most of the scenarios. In the third situation, the two algorithms perform equivalently in most of the scenarios. However, in the rest scenarios, the scenarios that AEDA-SS performs better are far more than the scenarios that AEDA performs better. By considering simulation reward as the fitness value, a more accurate approximation of the real expectation reward can be gain. Better solutions that perform well in most of the scenarios in all situations can be optimized

TABLE VI  
COMPARISON OF AEDA-SS WITH AEDA

		AEDA-SS	AEDA
$t_0 = 10 \ 35 \ 40$ $T = 30$	max	8.4361E+07	8.5815E+07
	mean	8.0603E+07	7.9740E+07
	b-number	-	1.8460E+03
	w-number	-	<b>6.3790E+03</b>
	e-number	-	1.7750E+03
$t_0 = 20 \ 30 \ 40$ $T = 20$	max	3.6038E+07	3.5611E+07
	mean	3.5666E+07	3.4389E+07
	b-number	-	1.2100E+03
	w-number	-	<b>8.5220E+03</b>
	e-number	-	2.6800E+02
$t_0 = 30 \ 35 \ 45$ $T = 10$	max	1.0320E+07	1.0337E+07
	mean	1.0314E+07	1.0328E+07
	b-number	-	4.7800E+02
	w-number	-	1.9620E+03
	e-number	-	<b>7.5600E+03</b>

b-number: scenarios that AEDA performs better than AEDA-SS.  
w-number: scenarios that AEDA performs worse than AEDA-SS.  
e-number: scenarios that AEDA perform equivalent to AEDA-SS.

TABLE VII  
COMPARISON OF AEDA-SS WITH AEDA-D AND AEDA-G

		AEDA-SS	AEDA-D	AEDA-G
$t_0 = 10 \ 35 \ 40$ $T_2 = 30$	max	8.5445E+07	8.8125E+07	8.8086E+07
	mean	7.9372E+07	8.1330E+07	7.9714E+07
	b-number	-	1.7000E+02	1.7040E+03
	w-number	-	<b>6.4490E+03</b>	<b>6.2410E+03</b>
	e-number	-	3.3810E+03	2.0550E+03
$t_0 = 20 \ 30 \ 40$ $T_2 = 20$	max	3.5599E+07	3.5782E+07	3.5253E+07
	mean	3.4438E+07	3.4636E+07	3.4194E+07
	b-number	-	7.5000E+02	9.9800E+02
	w-number	-	3.5370E+03	<b>7.1440E+03</b>
	e-number	-	<b>5.7130E+03</b>	1.8580E+03
$t_0 = 30 \ 35 \ 45$ $T_2 = 10$	max	1.0339E+07	1.0339E+07	1.0340E+07
	mean	1.0331E+07	1.0329E+07	1.0339E+07
	b-number	-	9.2800E+02	8.0900E+02
	w-number	-	1.1050E+03	1.5900E+03
	e-number	-	<b>7.9670E+03</b>	<b>7.6010E+03</b>

b-number: scenarios that AEDA-D(G) perform better than AEDA-SS.  
w-number: scenarios that AEDA-D(G) perform worse than AEDA-SS.  
e-number: scenarios that AEDA-D(G) perform equivalent to AEDA-SS.

by AEDA-SS. Therefore, the rationality of the objective function has been validated.

### C. Effectiveness of Simplified Simulation Approach

It has been analyzed in Section IV.B that the group insurance portfolio problem is scenario-relevant. Therefore, a simplified

TABLE VIII  
COMPARISON OF AEDA-SS WITH ADAPTIVE SIMULATION TIME SETTING STRATEGY

		AEDA-SS	$NP = 50$	$NP = 100$	$NP = 200$	$NP = 500$	$NP = 1000$
$t_0 = 10 \ 35 \ 40$ $T = 30$	max	8.5787E+07	8.7508E+07	8.4787E+07	8.5468E+07	8.5255E+07	8.7864E+07
	mean	7.9588E+07	7.9415E+07	7.9273E+07	7.7420E+07	7.9979E+07	8.0923E+07
	b-number	-	1.5920E+03	1.3910E+03	1.0040E+03	5.7000E+02	1.1830E+03
	w-number	-	<b>7.4870E+03</b>	<b>7.1490E+03</b>	<b>7.2440E+03</b>	1.6180E+03	2.8180E+03
	e-number	-	9.2100E+02	1.4600E+03	1.7520E+03	<b>7.8120E+03</b>	<b>5.9990E+03</b>
$t_0 = 20 \ 30 \ 40$ $T = 20$	max	3.5624E+07	3.6552E+07	3.6837E+07	3.6800E+07	3.6573E+07	3.6380E+07
	mean	3.4244E+07	3.4705E+07	3.5122E+07	3.4873E+07	3.4757E+07	3.5110E+07
	b-number	-	8.7200E+02	9.0900E+02	9.0500E+02	8.3300E+02	1.5380E+03
	w-number	-	<b>8.2640E+03</b>	<b>7.9880E+03</b>	<b>8.2280E+03</b>	<b>7.7290E+03</b>	<b>7.3280E+03</b>
	e-number	-	8.6400E+02	1.1030E+03	8.6700E+02	1.4380E+03	1.1340E+03
$t_0 = 30 \ 35 \ 45$ $T = 10$	max	1.0343E+07	1.0359E+07	1.0357E+07	1.0355E+07	1.0354E+07	1.0343E+07
	mean	1.0334E+07	1.0321E+07	1.0331E+07	1.0191E+07	1.0318E+07	9.9967E+06
	b-number	-	3.3400E+02	7.4600E+02	2.0900E+02	1.0500E+02	2.6000E+01
	w-number	-	<b>7.8740E+03</b>	<b>8.4950E+03</b>	<b>8.8230E+03</b>	<b>8.7680E+03</b>	<b>8.7930E+03</b>
	e-number	-	1.7920E+03	7.5900E+02	9.6800E+02	1.1270E+03	1.1810E+03

b-number: scenarios that adaptive simulation time setting strategy perform better than AEDA-SS.

w-number: scenarios that adaptive simulation time setting strategy perform worse than AEDA-SS.

e-number: scenarios that adaptive simulation time setting strategy perform equivalent to AEDA-SS.

simulation approach that simulates only one scenario in each generation during the evolution process is employed. To validate the rationality of the simplified simulation approach, we compare AEDA-SS with other two algorithms. The first one is the algorithm that simulates different scenarios in each generation, which is denoted as AEDA-D. The other one applies the most general situations (each insured dies at 70 and never get sick, the profits of the endowment policies are the mean value of their historical profit) to optimize the portfolio plan in the whole optimization process, which is denoted as AEDA-G.

The comparison results are shown in Table VII. It can be discovered that AEDA-SS outperforms AEDA-G evidently in the first and second situation and outperforms AEDA-D evidently in the first situation. In the other situations, although the three algorithms perform equivalently in most of the scenarios, the scenarios that AEDA-SS performs better are far more than the scenarios that the other two algorithms perform better in the rest scenarios. The comparison results validate that employs the same scenario to evaluate the solutions in each generation is more effective than employ different scenarios. By employing the same scenario in each generation, deviations caused by scenario when evaluating the solutions can be eliminated. Therefore, better solutions that perform well in most of the scenarios can be optimized by AEDA-SS compared with AEDA-D. However, only one scenario is not enough for the whole optimization process. The solutions optimized by AEDA-G may lose their effectiveness in other scenarios. The solutions optimized by AEDA-SS are able to show superiority in more scenarios.

To further validate the effectiveness of the simplified simulation approach, an adaptive strategy [33] which fixes the population size and increases the evaluation times during the iteration is applied as a comparison. We make comparisons between

AEDA-SS and adaptive strategy with different population size from 50 to 1000. The comparison results are shown in Table VIII.

From Table VIII, it can be discovered that AEDA-SS outperforms the adaptive strategy with all population sizes in most of the scenarios in most of the three situations except that when the population size is 500 and 1000 in the first situation. In these two cases, although AEDA-SS performs equivalently to the adaptive strategy in most of the scenarios, in the rest scenarios, the scenarios that AEDA-SS performs better are far more than the scenarios that the adaptive strategy performs better. That is to say, it is more effective to apply computational resources to enlarge population size than to increase the number of Monte Carlo simulations. The effectiveness of the simplified simulation approach has been validated through the comparison results.

#### D. Effectiveness of Clustering Method

In this article, two kinds of clustering method are introduced to improve the ability to search for more peaks for the algorithm. To validate the necessity of the clustering mechanism, we firstly compare the proposed CEDA-SS with AEDA-SS without clustering method. Here, CEDA-SS with crowding clustering is denoted as CEDA-SS-C and CEDA-SS with speciation clustering is denoted as CEDA-SS-S.

The comparison results are shown in Table IX. It can be discovered that in the second situations, both CEDA-SS-C and CEDA-SS-S outperform AEDA-SS in most of the scenarios. In the first and third situations, the three algorithms perform equivalently in most of the scenarios. However, in the rest scenarios, the scenarios that CEDA-SS-C and CEDA-SS-S perform better than AEDA-SS are far more than the scenarios that they perform worse. The clustering method integrates nearby

TABLE IX  
COMPARISON OF CEDA-SS WITH AEDA-SS

		AEDA-SS	CEDA-SS-C	CEDA-SS-S
$t_0 = 10 \ 35 \ 40$ $T_2 = 30$	max	8.5945E+07	8.5512E+07	8.5568E+07
	mean	7.9959E+07	8.0171E+07	7.9862E+07
	b-number	-	3.5470E+03	1.6270E+03
	w-number	-	2.7900E+02	7.5000E+01
	e-number	-	<b>6.1740E+03</b>	<b>8.2980E+03</b>
$t_0 = 20 \ 30 \ 40$ $T_2 = 20$	max	3.5586E+07	3.6625E+07	3.6786E+07
	mean	3.4365E+07	3.5023E+07	3.4874E+07
	b-number	-	<b>7.4780E+03</b>	<b>7.7440E+03</b>
	w-number	-	9.1800E+02	8.1600E+02
	e-number	-	1.6040E+03	1.4400E+03
$t_0 = 30 \ 35 \ 45$ $T_2 = 10$	max	1.0342E+07	1.0350E+07	1.0347E+07
	mean	1.0332E+07	1.0335E+07	1.0329E+07
	b-number	-	1.6240E+03	2.3400E+03
	w-number	-	4.4200E+02	2.8400E+02
	e-number	-	<b>7.9340E+03</b>	<b>7.3760E+03</b>

b-number: scenarios that CEDA-SS perform better than AEDA-SS.  
w-number: scenarios that CEDA-SS perform worse than AEDA-SS.  
e-number: scenarios that CEDA-SS perform equivalent to AEDA-SS.

TABLE X  
COMPARISON OF CEDA-SS WITH DEPSO

		DEPSO	CEDA-SS-C	CEDA-SS-S
$t_0 = 10 \ 35 \ 40$ $T_2 = 30$	max	8.7529E+07	8.5777E+07	8.5290E+07
	mean	8.0580E+07	8.0148E+07	7.9828E+07
	b-number	-	<b>6.1420E+03</b>	<b>6.4620E+03</b>
	w-number	-	3.8100E+02	4.4800E+02
	e-number	-	3.4770E+03	3.0900E+03
$t_0 = 20 \ 30 \ 40$ $T_2 = 20$	max	3.6335E+07	3.6582E+07	3.6765E+07
	mean	3.4465E+07	3.5045E+07	3.4890E+07
	b-number	-	<b>8.7570E+03</b>	<b>8.4720E+03</b>
	w-number	-	5.4000E+02	6.1200E+02
	e-number	-	7.0300E+02	9.1600E+02
$t_0 = 30 \ 35 \ 45$ $T_2 = 10$	max	1.0338E+07	1.0351E+07	1.0345E+07
	mean	1.0334E+07	1.0334E+07	1.0329E+07
	b-number	-	<b>8.1390E+03</b>	<b>7.4160E+03</b>
	w-number	-	3.0200E+02	2.5500E+02
	e-number	-	1.5590E+03	2.3290E+03

b-number: scenarios that CEDA-SS perform better than DEPSO.  
w-number: scenarios that CEDA-SS perform worse than DEPSO.  
e-number: scenarios that CEDA-SS perform equivalent to DEPSO.

individuals when constructing probability model to improve the applicability of the probability model. It can be proven that the supplement of the clustering mechanism can improve the performance of the algorithm. Different clustering method has its own merits in different situations.

Then, we compare CEDA-SS-C and CEDA-SS-S with DEPSO [7] algorithm and EDASLS [52], which have been proven to be effective to deal with uncertain optimization problems. The comparison results are shown in Table X and Table XI.

TABLE XI  
COMPARISON OF CEDA-SS WITH EDASLS

		EDASLS	CEDA-SS-C	CEDA-SS-S
$t_0 = 10 \ 35 \ 40$ $T_2 = 30$	max	8.7436E+07	8.5459E+07	8.5441E+07
	mean	8.1722E+07	8.0138E+07	7.9774E+07
	b-number	-	<b>4.1040E+03</b>	1.5600E+03
	w-number	-	9.3500E+02	1.4000E+01
	e-number	-	4.9610E+03	<b>8.4260E+03</b>
$t_0 = 20 \ 30 \ 40$ $T_2 = 20$	max	3.6095E+07	3.6520E+07	3.6781E+07
	mean	3.5093E+07	3.5038E+07	3.4908E+07
	b-number	-	<b>6.9430E+03</b>	<b>6.5940E+03</b>
	w-number	-	6.4900E+02	1.1100E+02
	e-number	-	2.4080E+03	3.2950E+03
$t_0 = 30 \ 35 \ 45$ $T_2 = 10$	max	1.0346E+07	1.0350E+07	1.0347E+07
	mean	1.0321E+07	1.0334E+07	1.0329E+07
	b-number	-	<b>8.4550E+03</b>	<b>7.8620E+03</b>
	w-number	-	2.4300E+02	4.7100E+02
	e-number	-	1.3020E+03	1.6670E+03

b-number: scenarios that CEDA-SS perform better than EDASLS.  
w-number: scenarios that CEDA-SS perform worse than EDASLS.  
e-number: scenarios that CEDA-SS perform equivalent to EDASLS.

It can be discovered that in all of the three situations, both CEDA-SS-C and CEDA-SS-S outperform DEPSO in most of the scenarios. As for EDASLS, CEDA-SS-C outperforms it in most of the scenarios in all of the three situations. CEDA-SS-S outperforms it in most of the scenarios in the last two situations. In the first situation, the two algorithms perform equivalently in most of the scenarios. But in the rest scenarios, the scenarios that CEDA-SS-S performs better are far more than the scenarios that EDASLS performs better. That is to say, CEDA-SS is more suitable for the group insurance portfolio problem than these two state-of-the-art uncertain evolutionary algorithms. Therefore, the effectiveness of the proposed CEDA-SS has been validated.

### E. Time Complexity Analysis of the Algorithms

1) *Theoretical Analysis:* Firstly, we analyze the time complexity of the evolution process in each generation for the algorithms. The time complexity of AEDA, AEDA-SS, AEDA-D and AEDA-G is the same. It takes  $O(NP \log_2 NP)$  to find the elite solutions, takes  $O(NP \times D)$  to establish the probability model and takes  $O(NP \times D)$  to generate the new population. In CEDA-SS-C, it takes additional  $O(NP \log_2 NP + D)$  to decompose the population. In CEDA-SS-S, it takes  $O(NP \log_2 NP)$  to decompose the population. In DEPSO, it takes  $O(NP \log_2 NP)$  for elite selection, takes  $O(NP \times D)$  for weight calculation, takes  $O(NP \times D)$  for search center generation and takes  $O(NP \times D)$  for population update. In EDASLS, it takes  $O(NP \log_2 NP)$  to find the elite solutions, takes  $O(NP \times D)$  to establish the probability model, takes  $O(D)$  for stochastic local search and takes  $O(NP \times D)$  to generate the new population.

Secondly, we analyze the time complexity of the fitness evaluation process for each generation. In AEDA, it takes  $O(NP \times$



TABLE XII  
COMPUTATIONAL TIME OF THE ALGORITHMS

	CEDA-SS-C	CEDA-SS-S	AEDA-SS	AEDA-D	AEDA-G	AEDA	DEPSO	EDASLS
$t_0 = 10 \ 35 \ 40$ $T = 30$	2.7783E+02	2.6876E+02	2.5994E+02	2.7083E+02	1.6163E+01	1.8789E+01	3.1657E+02	2.7624E+02
$t_0 = 20 \ 30 \ 40$ $T = 20$	1.9709E+02	1.9775E+02	1.7935E+02	1.8695E+02	1.0137E+01	1.3755E+01	2.3425E+02	1.9335E+02
$t_0 = 30 \ 35 \ 45$ $T = 10$	1.1970E+02	1.1594E+02	1.1280E+02	1.1988E+02	6.0970E+00	9.6760E+00	1.2739E+02	1.1894E+02

TABLE XIII  
STANDARD DEVIATION OF THE ALGORITHMS

	CEDA-SS-C	CEDA-SS-S	AEDA-SS	AEDA-D	AEDA-G	AEDA	DEPSO	EDASLS
$t_0 = 10 \ 35 \ 40$ $T = 30$	2.6234E+06	2.7492E+06	3.4108E+06	3.3111E+06	2.8314E+06	1.9877E+06	3.5617E+06	2.4053E+06
$t_0 = 20 \ 30 \ 40$ $T = 20$	7.5010E+05	9.1261E+05	4.0007E+05	5.6446E+05	4.2531E+05	3.7870E+05	7.5727E+05	6.6244E+05
$t_0 = 30 \ 35 \ 45$ $T = 10$	8.8853E+03	1.2680E+04	8.8351E+03	9.3604E+03	1.1268E+03	1.0672E+04	6.7715E+03	1.7632E+04

D). While in all the rest algorithms, it takes  $O(NP \times D \times T_2)$  for fitness evaluation.

2) *Experimental Verification*: To further verify the efficiency of the algorithms, we test the computational time of the algorithms in the same running environment. Each algorithm runs 30 times to eliminate experimental errors. The experimental results are given in Table XII.

It can be discovered that the theoretical analysis has been verified in the experiment. The computational time of CEDA-SS-C and CEDA-SS-S are very close since they have a similar complexity time. However, CEDA-SS-S runs a bit faster than CEDA-SS-C since crowding clustering method spends more time generating the random reference point  $R$  with  $O(D)$ . Both of these two algorithms runs slower than AEDA-SS due to the addition cost of the decomposition operation. Although the time complexity of AEDA-D, AEDA-SS and AEDA-G are the same, AEDA-D runs slower than AEDA-SS since only one scenario is needed to simulate in each generation for AEDA-SS, which can save the running time. AEDA-G runs much faster than the other two algorithms since only one scenario which is already determined before the experiment is applied for the optimization. DEPSO runs slower than other algorithms since the evolution process in DEPSO is the most time-wasting. EDASLS runs slower than AEDA-SS due to the addition cost of stochastic local search. AEDA runs much faster than the other algorithms since the fitness evaluation process is much simpler in this algorithm.

However, although AEDA-G and AEDA is more efficient than CEDA-SS, they are not as effective as CEDA-SS. Since the group insurance portfolio problem can be solved usually in an off-line manner without instant decision-making, the effectiveness of the algorithm is more important than the efficiency of the algorithm.

#### F. Robustness Analysis of the Algorithms

To analyze the robustness of the algorithms, 30 solutions are optimized by each algorithm. The mean profit of each solution in ten thousand scenario is calculated. The standard deviation of

the mean profit of 30 solutions for each algorithm is calculated as the measurement of the robustness of the algorithm. The experimental results are given in Table XIII.

It can be discovered that all of the algorithms are robust since such standard deviations are relatively small for the magnitude of the solution. AEDA-G is more robust than the other algorithms since only one scenario is applied for the optimization. The solutions optimized by this algorithm are similar solutions which are suitable for this scenario. AEDA is more robust than the other algorithms since the approximated expectation reward is taken as the fitness value of the algorithm. Therefore, the problem considered in AEDA is a deterministic problem. As for the rest algorithms, some algorithms are more robust in one situation while the others are more robust in the other situation.

## VII. CONCLUSION

In this article, attentions are paid to financial decision making problems with strong uncertainty. Different from other uncertain decision making problems, the uncertainties in financial decision making problems have their own features, which brings a big challenge for research. A group insurance portfolio problem is given as an instance to study the features of financial decision making problems. Based on the correlation analysis of group insurance portfolio problem, a simplified simulation approach is proposed to deal with the strong uncertainties in financial decision making problems. This approach is combined with a clustering estimation of distribution algorithm to deal with the proposed group insurance portfolio problem. The clustering method impels the algorithm to find more peaks during the optimization process and improve the performance of the algorithm.

The application of the simplified simulation approach to evolutionary algorithm is a new attempt. this article explores the possibility and rationality of such an attempt and shows that the proposed CEDA-SS is effective for optimizing group insurance portfolio problem. The effectiveness of the algorithm can be further improved to deal with other financial decision making problems with strong uncertainty in the future. To

reduce computational cost, classification methods based on neural networks [54], [55] can be applied to construct a surrogate model to approximate the fitness value of uncertain financial decision making problems in the future.

## REFERENCES

- [1] C. Avery and P. Zemsky, "Multidimensional uncertainty and herd behavior in financial markets," *Amer. Econ. Rev.*, vol. 88, pp. 724–748, 1998.
- [2] L. R. A. C. Shannon, "Uncertainty and risk in financial markets," *Econometrica*, vol. 73, no. 1, pp. 203–243, 2005.
- [3] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 97–111, Apr. 2017.
- [4] Y. Chou, S. Kuo, and Y. Jiang, "A novel portfolio optimization model based on trend ratio and evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 4, pp. 337–350, Aug. 2019.
- [5] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Computation*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
- [6] A. N. Aizawa, "Dynamic control of genetic algorithms in a noisy environment," in *Proc. Int. Conf. Genetic Algorithms*, 1993, pp. 48–55.
- [7] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2011–2021, Jun. 2019.
- [8] H. Greiner, "Robust optical coating design with evolutionary strategies," *Appl. Opt.*, vol. 35, no. 28, pp. 5477–5483, 1996.
- [9] B. S. Yaochu Jin, "Trade-off between Performance and Robustness: An evolutionary multiobjective approach," *Lecture Notes Comput. Sci.*, vol. 2632, pp. 237–251, 2003.
- [10] J. J. Grefenstette and J. M. Fitzpatrick, "Genetic search with approximate function evaluation," in *Proc. Int. Conf. Genetic Algorithms*, 1985, pp. 112–120.
- [11] S. Y. Ho, J. H. Chen, D. E. Goldberg, and K. Sastry, "Fitness inheritance in multiobjective optimization," in *Proc. Genetic Evol. Comput. Conf.*, 2002, pp. 319–326.
- [12] M. O. Akinsolu, B. Liu, V. Grout, P. I. Lazaridis, M. E. Mognaschi, and P. D. Barba, "A parallel surrogate model assisted evolutionary algorithm for electromagnetic design optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 95–105, Apr. 2019.
- [13] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 106–116, Apr. 2019.
- [14] C. Li and S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments," *IEEE Trans. Evol. Computation*, vol. 16, no. 4, pp. 556–577, Aug. 2012.
- [15] C. Bierwirth and D. C. Mattfeld, "Production scheduling and rescheduling with genetic algorithms," *Evol. Computation*, vol. 7, no. 1, pp. 1–17, 2014.
- [16] W. Luo, R. Yi, Y. Bin, and P. Xu, "Surrogate-assisted evolutionary framework for data-driven dynamic optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 137–150, Apr. 2019.
- [17] W. C. Yeh, Y. Lin, Y. Y. Chung, and M. Chih, "A particle swarm optimization approach based on Monte Carlo simulation for solving the complex network reliability problem," *IEEE Trans. Rel.*, vol. 59, no. 1, pp. 212–221, Mar. 2010.
- [18] C. Qian, Y. Yang, K. Tang, Y. Jin, and Z. H. Zhou, "On the effectiveness of sampling for evolutionary optimization in noisy environments," *Evolut. Comput.*, vol. 26, no. 2, pp. 237–267, 2018.
- [19] X. Fei, J. Branke, and N. Gulpinar, "New sampling strategies when searching for robust solutions," *IEEE Trans. Evol. Computation*, vol. 23, no. 2, pp. 273–287, Apr. 2019.
- [20] F. Xiao, "EFMCDM: Evidential fuzzy multicriteria decision making based on belief entropy," *IEEE Trans. Fuzzy Sys.*, vol. 28, no. 7, pp. 1477–1491, Jul. 2019.
- [21] F. Xiao, "CED: A distance for complex mass functions," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020, to be published.
- [22] F. Xiao, "GIQ: A generalized intelligent quality-based approach for fusing multi-source information," *IEEE Trans. Fuzzy Syst.*, 2020, to be published.
- [23] F. Xiao, "A new divergence measure for belief functions in D-S evidence theory for multisensor data fusion," *Inf. Sci.*, vol. 514, pp. 462–483, 2020.
- [24] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [25] W. Shi, W. N. Chen, Y. Lin, T. Gu, S. Kwong, and J. Zhang, "An adaptive estimation of distribution algorithm for multi-policy insurance investment planning," *IEEE Trans. Evol. Computation*, vol. 23, no. 1, pp. 1–14, Feb. 2019.
- [26] E. Bengoetxea, P. Larrañaga, I. Bloch, and A. Perchant, "Estimation of distribution algorithms," *Genetic Algorithms Evol. Comput.*, vol. 64, no. 5, pp. 1140–1148, 2001.
- [27] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.
- [28] Y. Sano and H. Kita, "Optimization of noisy fitness functions by means of genetic algorithms using history of search," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2000, pp. 1001–1008.
- [29] J. Branke and C. Schmidt, "Selection in the presence of noise," *Lecture Notes Comput. Sci.*, vol. 28, no. 2, pp. 766–777, 2003.
- [30] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A Statist. Mech. Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012.
- [31] A. D. Pietro, L. While, and L. Barone, "Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions," in *Proc. Congr. Evol. Comput.*, 2004, pp. 1254–1261.
- [32] E. Cantúpaz, "Adaptive sampling for noisy problems," in *Proc. Genetic Evol. Comput. Gecco, Genetic Evol. Comput. Conf.*, Seattle, WA, USA, Jun. 2004, pp. 947–958.
- [33] A. N. Aizawa and B. W. Wah, "Scheduling of genetic algorithms in a noisy environment," *Evol. Comput.*, vol. 2, no. 2, pp. 97–122, 2014.
- [34] B. S. Yaochu Jin, "Trade-off between performance and robustness: An evolutionary multiobjective approach," *Lecture Notes Comput. Sci.*, vol. 2632, pp. 237–251, 2003.
- [35] J. Branke, "Creating robust solutions by means of evolutionary algorithms," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 1998, pp. 119–128.
- [36] H. G. Beyer and B. Sendhoff, "Evolution strategies for robust optimization," in *Proc. IEEE Congr. Evol. Computation*, 2006, pp. 1346–1353.
- [37] J. Kruisselbrink, M. Emmerich, and T. Bäck, "An archive maintenance scheme for finding robust solutions," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 2010, pp. 214–223.
- [38] J. S. Chen and B. P. Liao, "Piecewise nonlinear goal-directed CPPI strategy," *Expert Syst. Appl.*, vol. 33, no. 4, pp. 857–869, 2007.
- [39] J. S. Chen and Y. T. Lin, "A partitioned portfolio insurance strategy by a relational genetic algorithm," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2727–2734, 2009.
- [40] T. Y. Yu, C. Tsai, and H. T. Huang, "Applying simulation optimization to the asset allocation of a property-casualty insurer," *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 499–507, 2010.
- [41] M. Duma, T. Marwala, B. Twala, and F. Nelwamondo, "Partial imputation of unseen records to improve classification using a hybrid multi-layered artificial immune system and genetic algorithm," *Appl. Soft Comput.*, vol. 13, no. 12, pp. 4461–4480, 2013.
- [42] S. Salcedo-Sanz, J. L. Fernández-Villacañas, M. J. Segovia-Vargas, and C. Bousoño-Calzón, "Genetic programming for the prediction of insolvency in non-life insurance companies," *Comput. Oper. Res.*, vol. 32, no. 4, pp. 749–765, 2005.
- [43] K. J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Syst. Appl.*, vol. 19, no. 2, pp. 125–132, 2000.
- [44] Y. Kwon and B. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Trans. Neural Networks*, vol. 18, no. 3, pp. 851–864, May 2007.
- [45] M. Rout, B. Majhi, R. Majhi, and G. Panda, "Novel stock market prediction using a hybrid model of adaptive linear combiner and differential evolution," *Commun. Comput. Inf. Sci.*, vol. 142, pp. 187–191, 2011.
- [46] B. Hagströmer and J. M. Binner, "Stock portfolio selection with full-scale optimization and differential evolution," *Appl. Financial Econ.*, vol. 19, no. 19, pp. 1559–1571, 2009.
- [47] J. F. Chang and P. Shi, "Using investment satisfaction capability index based particle swarm optimization to construct a stock portfolio," *Inform. Sci.*, vol. 181, no. 14, pp. 2989–2999, 2010.
- [48] L. Yu, H. U. Luchao, and T. Ling, "Stochastic selection with a novel sigmoid-based mixed discrete-continuous differential evolution algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1891–1904, Jul. 2016.
- [49] R. Taylor, "Interpretation of the correlation coefficient: A basic review," *J. Diagnostic Med. Sonogr.*, vol. 6, no. 1, pp. 35–39, 1990.
- [50] Q. Yang, W. N. Chen, Y. Li, C. L. Chen, X. M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.
- [51] P. E. Danielsson, "Euclidean distance mapping," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 227–248, 1980.

- [52] J. Wang, K. Tang, J. A. Lozano, and X. Yao, "Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated Arc routing problems," *IEEE Trans. Evol. Computation*, vol. 20, no. 1, pp. 96–109, Feb. 2016.
- [53] W. Haynes, "Wilcoxon rank sum test," New York, NY, USA: Springer, 2013, pp. 2354–2355.
- [54] L. Zhang, Y. Zhu, and W. X. Zheng, "Synchronization and state estimation of a class of hierarchical hybrid neural networks with time-varying delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 459–470, Feb. 2016.
- [55] L. Zhang, Y. Zhu, and W. X. Zheng, "State estimation of discrete-time switched neural networks with multiple communication channels," *IEEE Trans. Cyber.*, vol. 47, no. 4, pp. 1028–1040, Apr. 2017.



**Wen Shi** (Student Member, IEEE) received the bachelor's degree from Sun Yat-sen University, China, in 2016. She is now a Ph.D. Student with the School of Computer Science and Engineering, South China University of Technology, China. Her current research interests include evolutionary computation algorithms and their applications on financial problems. So far, she specifically works on differential evolution, particle swarm optimization and estimation of distribution algorithms and their applications on financial problems, such as insurance investment optimization problem.



**Wei-Neng Chen** (Senior Member, IEEE) received the bachelor's and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2006 and 2012, respectively, both in computer science. Since 2016, he has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, where he is now also the Vice Dean of the school. He has co-authored more than 100 international journal and conference papers, including more than 40 papers published in the IEEE TRANSACTIONS JOURNALS. His current research in-

terests include computational intelligence, swarm intelligence, network science, and their applications.

Dr. Chen was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Dissertation Award in 2016, and the National Science Fund for Excellent Young Scholars in 2016. He is currently the Vice-Chair of the IEEE Guangzhou Section. He is also a Committee Member of the IEEE CIS Emerging Topics Task Force. He serves as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the *Complex & Intelligent Systems*.



data and knowledge engineering, software engineering, and information security protocol.

**Tianlong Gu** received the M.Eng. degree from Xidian University, China, in 1987, and the Ph.D. degree from Zhejiang University, China, in 1996. From 1998 to 2002, he was a Research Fellow with the School of Electrical and Computer Engineering, Curtin University of Technology, Australia, and a Postdoctoral Fellow with the School of Engineering, Murdoch University, Australia. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include formal methods,



**Hu Jin** (Senior Member, IEEE) received the B.E. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2004, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2006 and 2011, respectively.

From 2011 to 2013, he was a Postdoctoral Fellow with the University of British Columbia, Vancouver, BC, Canada. From 2013 to 2014, he was a Research Professor with Gyeongsang National University, Tongyeong, South Korea. Since 2014, he has been with the Division of Electrical Engineering, Hanyang University, Ansan, South Korea, where he is currently an Associate Professor. His research interests include wireless communications, Internet of Things, and machine learning.



**Jun Zhang** (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, in 2002. Currently, he is a Visiting Scholar with the University of Hanyang University, Korea. His research interests include computational intelligence, cloud computing, data mining, and power electronic circuits. He has published over 200 technical papers in his research area. Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the

First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON CYBERNETICS.