Expert Systems With Applications 145 (2020) 113104

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid approach for portfolio selection with higher-order moments: Empirical evidence from Shanghai Stock Exchange

Bilian Chen[a,b], Jingdong Zhong[c], Yuanyuan Chen[d,*]

[a] Department of Automation, Xiamen University, Xiamen 361005, China
[b] Amoy Key Lab. of Big Data Intelligent Analysis and Decision, Xiamen 361005, China
[c] School of Economics, Peking University, Beijing 100871, China
[d] Department of Finance and Insurance, Nanjing University, Jiangsu Province 210093, China

ARTICLE INFO

Article history:
Received 27 August 2019
Revised 21 November 2019
Accepted 27 November 2019
Available online 2 December 2019

Keywords:
Portfolio optimization
Higher-order moments
Genetic algorithm
Machine learning algorithm

ABSTRACT

Skewness and kurtosis, the third and fourth order moments, are statistics to summarize the shape of a distribution function. Recent studies show that investors would take these higher-order moments into consideration to make a profitable investment decision. Unfortunately, due to the difficulties in solving the multi-objective problem with higher-order moments, the literature on portfolio selection problem with higher-order moments is few. This paper proposes a new hybrid approach to solve the portfolio selection problem with skewness and kurtosis, which includes not only the multi-objective optimization but also the data-driven asset selection and return prediction, where the techniques of two-stage clustering, radial basis function neural network and genetic algorithm are employed. With the historical data from Shanghai stock exchange, we find that the out-of-sample performance of our model with higher-order moments is significantly better than that of traditional mean-variance model and verify the robustness of our hybrid algorithm.

1. Introduction

Portfolio selection problem has been one of the core issues of the modern investment theory. It originates from the mean-variance model by Markowitz (1952), which measured the expected return and risk of a portfolio by mean and variance, and thus first transformed the portfolio selection problem into a mathematical model. Based on this, researchers further extended the classical model by applying other risk measurements, such as semi-variance, absolute deviation, and semi-absolute deviation. See Steinbach (2001) for a review on these significant ones which considered only the first and second order moments of return distribution. However, these moments are inadequate in practice. Meanwhile, the performance of a portfolio selection model highly depends on the parameter estimation and the initial asset selection. We thus intend to build a hybrid algorithm, which includes not only data-driven asset selection and prediction, but also solving the portfolio selection problem with higher-order moments.

In the literature, the importance of higher order moments has attracted increasing attention. Liu (2004) and Maringer and Parpas (2009) proved that only when the investor's utility function is quadratic or the asset's yield obeys the normal distribution, the influence of higher-order moments can be ignored. Unfortunately, both of these two premises are hardly satisfied in the real world. Empirical studies found that the risk assets' yields have fat tails and do not obey the normal distribution (Chunhachinda, Dandapani, Hamid, & Prakash, 1997; Konno & Suzuki, 1992). There are also works (see e.g. Paravisini, Rappoport, & Ravina (2016)) observing the fact that an investor's risk aversion degree and sensitivity will change with the wealth accumulation, whereas the quadratic utility function is unable to characterize this feature and thus is not practical in reality. Hence, it is necessary to consider higher-order moments when constructing the portfolio optimization model.

In the past decades, an increasing number of studies have involved skewness into the portfolio optimization model, and observed that the mean-variance-skewness (MVS) model performs better than the classical mean-variance one (see e.g. Adcock (2014); Briec, Kerstens, and Jokung (2007); Joro and Na (2006); Liu, Han, and Han (2016); Zhai, Bai, and Wu (2018)). Unfortunately, there are few studies on kurtosis, due to the difficulties in obtaining the optimal solution to the the higher-order-moment optimization problem, which is a multi-objective problem in essence. Maringer and Parpas (2009) applied stochastic

* Corresponding author.
E-mail addresses: blchen@xmu.edu.cn (B. Chen), jdzhong@pku.edu.cn (J. Zhong), yychen@nju.edu.cn (Y. Chen).

https://doi.org/10.1016/j.eswa.2019.113104
0957-4174/

optimization algorithms to extended mean-variance portfolio selection problems with either skewness or kurtosis being considered at a time. Saranya and Prasanna (2014) and Aksaraylı and Pala (2018) studied the mean-variance-skewness-kurtosis framework by constructing a polynomial goal programming model for the higher moments. Chen and Zhou (2018) applied multiobjective particle swarm optimization to deal with higher moments, but focused on the parameter uncertainty.

Inspired by the existing works on the mean-variance portfolio selection problem (see e.g., Li and Ng (2000) and Cui, Li, Wang, and Zhu (2012)), we, instead of construct a polynomial goal programming model, incorporate the trade-off parameters over the mean, variance, skewness and kurtosis to formulate a single target model. However, this constrained non-linear optimization problem is still difficult to solve. Traditional optimization algorithms for constrained non-linear optimization problem have quite low convergence rates, and may even not converge to a solution in extreme cases. Hence, we turn to machine learning algorithms to boast a high generalization and a fast convergence. In particular, we apply genetic algorithm, which can not only boast a fast searching speed, but also reduce the risk of falling into a local minimum trap, to solve our single target problem.

Moreover, we focus on how to improve the asset selection and return prediction with machine learning algorithm and historical data, instead of considering robust model with parameter uncertainty. In fact, a real portfolio selection problem starts with the selection of assets. Most present studies rely on investors' experience to select risk assets as samples to build the prior portfolio, which appears subjective and lacks the support of scientific selection criteria. In addition, the performance of a portfolio selection model highly depends on the parameter estimations, such as the return vector and the risk measurements. The parameters estimated by the historical data may not be able to predict the future.

Our paper is thus also related to the works applying machine learning algorithms in various financial fields. Patel, Shah, Thakkar, and Kotecha (2015) and Chong, Han, and Park (2017) applied machine learning techniques for market analysis and prediction. See (Henrique, Sobreiro, & Kimura, 2019) for a review on the machine learning techniques applied to financial market prediction. Paiva, Cardoso, Hanaoka, and Duarte (2019) proposed a unique decision-marking model for day trading investments on stock market based on machine learning methodology with the classical mean-variance model. Varied clustering techniques are applied on financial data to identify the similarity inside the time series data (See e.g., Iorio, Frasso, D'Ambrosio, and Siciliano (2016); Zhang, Liu, Du, and Lv (2011) and the references therein). Iorio, Frasso, D'Ambrosio, and Siciliano (2018) furthermore applied the P-spline based clustering approach on financial data to build a portfolio. We also include the two-stage clustering method for asset selection in our hybrid algorithm. In fact, we apply machine learning algorithms for not only the market prediction and asset selection before the optimization problem, but also solving the multi-objective portfolio selection problem with higher order moments to identify the optimal portfolio.

To conclude, in this paper, we investigate the portfolio selection problem with skewness and kurtosis by applying a hybrid approach to select pre-diversified assets, predict the returns and optimize the portfolio.

First of all, to deal with this multi-objective problem, we introduce the risk preference to transfer it to a non-linear optimization problem and apply the genetic algorithm, which is a probability-based directional searching tool and is able to reduce the risk of falling into a local minimum trap, to numerically solve the considered portfolio optimization.

Second, instead of solving the portfolio selection problem with parameters directly estimated from the historical data, we propose a hybrid approach which includes the asset selection, return prediction and portfolio optimization. In particular, besides the genetic algorithm applied to solve the optimization problem, our hybrid approach also includes two-stage clustering and radial basis function neural network. The two-stage clustering, also known as Chameleon algorithm, is a statistical data analysis technique proposed by Karypis, Han, and Kumar (1999). The basic idea of this method is to classify a set of objects according to the interconnectivity and closeness such that objects in a cluster are more similar to each other than those in the other cluster. We thus select assets with the two-stage clustering for a diversified portfolio. We then apply the radial basis function neural network on these selected assets to predict the future returns.

Finally, we apply the historical data from Shanghai Exchange to compare the performance of our model with those of mean-variance and mean-variance-skewness models and testify the robustness and efficiency of our hybrid algorithm.

The rest of this paper is organized as follows. Section 2 introduces the higher-order-moment portfolio optimization model and the genetic algorithm for solving it. Section 3 illustrates our hybrid approach, which includes not only the higher-order-moment portfolio optimization, but also the data-driven asset selection and return prediction for a pre-diversified portfolio selection. Section 4 performs the numerical experiments to test the model's efficiency and reports the experimental results. Section 5 investigates the robustness of our algorithm. The last section concludes this paper.

2. Higher-order-moment portfolio optimization

In this section, we formulate our portfolio selection model with skewness and kurtosis, and introduce the genetic algorithm to solve this higher-order-moment portfolio optimization problem.

2.1. Higher-order-moment portfolio selection model

In this subsection, we first consider the problem of constructing a portfolio from n financial products, which will be selected by our two-clustering method introduced in Section 3. A portfolio $X = (x_1, x_2, \ldots, x_n)^T$ is a vector of the proportions in each of these n assets with $x_1 + x_2 + \cdots + x_n = 1$. The mean return of the i-th asset is $\overline{R}_i$, $i = 1, \ldots, n$. $\overline{R}$ is the vector of the expected return, while V is the co-variance matrix. Denote σ_i^2 as the variance of asset i's return, and σ_{ij} as the covariance between the returns of assets i and j. The expected return and its variance, skewness and kurtosis are thus computed as follows:

$$\mathbb{E}(X) = X^T\overline{R} = \sum_{i=1}^{n} x_i\overline{R}_i, \tag{1}$$

$$Var(X) = X^TVX = \sum_{i=1}^{n} x_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} x_ix_j\sigma_{ij} \ (i \neq j), \tag{2}$$

$$Skew(X) = \mathbb{E}(X^T(R - \overline{R})^3), \tag{3}$$

$$Kurt(X) = \mathbb{E}(X^T(R - \overline{R})^4). \tag{4}$$

The classical mean-variance portfolio selection problem is thus expressed as the following multi-objective optimization problem.

$$\begin{aligned} &\max\ \mathbb{E}(X) \\ &\min\ Var(X) \\ &s.t.\ X^T\mathbf{1} = 1, X \geq 0. \end{aligned}$$

where $\mathbf{1}$ is a all-one vector. The constraint that $X \geq 0$ is the no-shorting constraint. In other words, short selling is not allowed in our model. Because a portfolio with a larger skewness and

a smaller kurtosis is more appealing to the investors, we improve the classical mean-variance model's performance by including skewness and kurtosis. The higher-order-moment model is thus described as the following multi-objective optimization problem (P_1).

$$(P_1)\begin{cases}\max \mathbb{E}(X)\\ \min Var(X)\\ \max Skew(X)\\ \min Kurt(X)\\ s.t.\ X^T\mathbf{1}=1, X\geq 0.\end{cases}$$

In order to solve problem (P_1), we can transform it into the following nonlinear programming:

$$(P_2)\begin{cases}\max\ \lambda_1(X^T\bar{R})-\lambda_2(X^TVX)+\lambda_3[\mathbb{E}(X^T(R-\bar{R})^3)]\\ \qquad\qquad -\lambda_4[\mathbb{E}(X^T(R-\bar{R})^4)]\\ s.t.\ X^T\mathbf{1}=1, X\geq 0,\end{cases}$$

where λ_1, λ_2, λ_3 and λ_4 are non-negative investment preference factors, corresponding to the objectives: mean, variance, skewness and kurtosis, respectively. In fact, the auxiliary parameters, λ_i, $i=1,2,3,4$, capture the risk preference of the investor. For example, an investor who doesn't care about the kurtosis can set the parameter λ_4, which is corresponding to the kurtosis, to be zero. We focus on the solution scheme for model (P_2) with different sets of given auxiliary parameters $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ in this paper.

2.2. Genetic algorithm for higher-order-moment portfolio optimization

In this subsection, we apply the genetic algorithm to solve the higher-order-moment portfolio optimization problem. Note that model (P_2) is a constrained non-linear optimization problem. Solving this problem will inevitably consume a significant amount of computing resources, and even may not converge to the global optimal solution. As a heuristic computational intelligent tool, *genetic algorithm* not only boasts a fast searching speed, but also reduces the risk of falling into a local minimum trap. We will thus apply the genetic algorithm to solve problem (P_2).

In particular, a feasible solution to the optimization problem (P_2) is called a chromosome, which is evaluated by the fitness function designed according to the objective function of the optimization problem. These chromosomes are selectively evolved by individual reproduction, crossover and mutation. We summarize the work flow of the genetic algorithm in Fig. 1. The detailed steps are described as follows.

Step 1: Individual coding. We translate asset weight $X=(x_1,x_2,\ldots,x_n)^T$ into the binary string, whose length depends on permissible accuracy. The accuracy in this paper is p_a, and the variable x_i ranges from 0 to 1 for $i=1,2,\ldots,n$. According to the calculation on the string length, we can obtain the only integer m satisfying $2^{m-1}<(1-0)/p_a<2^m-1$, and thus the length of each chromosome X as nm bits.[1] Each m-bit denotes the weight of one of the n assets.

Step 2: Population initialization. Assuming that the initial population has L chromosomes,

$$X^1, X^2, \ldots, X^L,$$

we randomly generate L binary strings with nm bits which are feasible to model (P_2). In particular, a chromosome $X^l=(x_1^l,x_2^l,\ldots,x_n^l)$ $(l=1,2,\ldots,L)$ is decoded at m bits per unit to get the real value of each variable x_i^l, $i=1,\ldots,n$. For example, if the binary string corresponding to the variable x_i^l is

$$b_m b_{m-1} b_{m-2}\cdots b_3 b_2 b_1,$$

its decimal value is computed as

$$x_i^l=\left(\sum_{j=1}^{m} b_j 2^{j-1}\right)\cdot\frac{1-0}{2^m-1},\quad (i=1,2,\ldots,n). \tag{5}$$

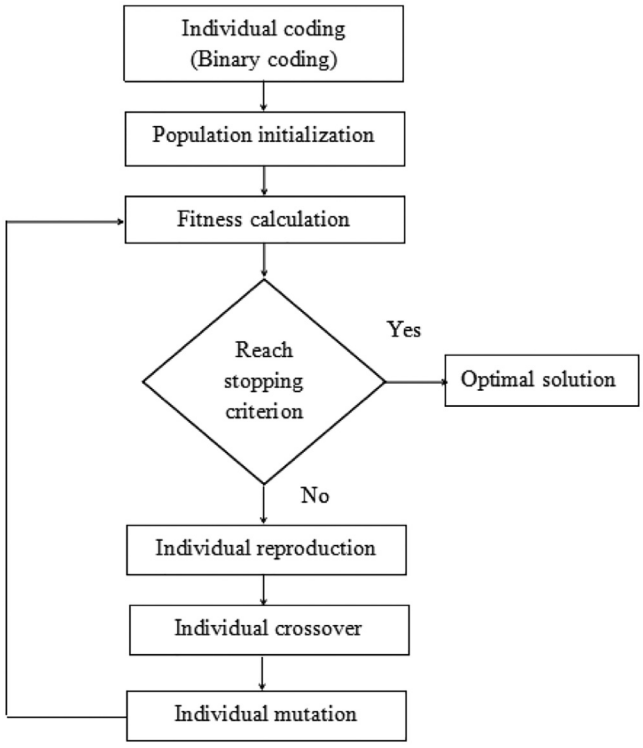


Fig. 1. The flowchart of Genetic Algorithm.

Step 3: Fitness calculation. According to model (P_2), we define the fitness function $g(\cdot)$ to evaluate the solutions as

$$g(X)=\lambda_1(X^T\bar{R})-\lambda_2(X^TVX)+\lambda_3[E(X^T(R-\bar{R})^3)] -\lambda_4[E(X^T(R-\bar{R})^4)]-P\left|\sum_{i=1}^{n}x_i-1\right|, \tag{6}$$

where P is the penalty factor to eliminate individuals that dissatisfy the constraints. The target is to minimize the fitness function. The fitness of the l-th chromosome is obtained as $g(X^l)$, and the total fitness of the population is calculated by

$$G=\sum_{l=1}^{L} g(X^l). \tag{7}$$

Step 4: Individual reproduction. A certain number of chromosomes are selected to join in the reproduction. To select the chromosomes to be reproduced, we calculate the accumulated probability of each chromosome to be reproduced by

$$Q_l=\sum_{k=1}^{l} g(X^k)/G. \tag{8}$$

The roulette selection algorithm is simulated in N_R times by the computer. At each time, a number s between 0 and 1 is randomly generated. When $Q_{l-1}\leq s<Q_l$ the individual l is reproduced as the member of new generation. This algorithm indicates that the higher-fitness individual owns a larger probability to be selected for the reproduction, which assures that the feasible solutions with

[1] For different accuracy p_a, one would need varied length of the binary strings to guarantee varied digits of accuracy. For example, a 14-bit code can represent numbers that are accurate to four decimal places.

larger objective values have higher probabilities to reproduce the new "generations".

Step 5: Individual crossover. The number of chromosomes in each crossover equals to the population size times the crossover probability, which is set to be p_c. There are L individuals in the population, so p_cL chromosomes will participate in the crossover. In order to determine the specific ones, we use the computer to generate L random numbers to denote the chromosomes in the population. They are ranked in the decreasing order of the value. The top p_cL chromosomes are selected into the crossover each time.

Step 6: Individual mutation. We assume that the mutation probability is p_m, which is to say, there is 0-1 reversal in one percent of genes. We can illustrate the process of crossover and mutation with a simple example. Assume that there are two three-bit parent individuals, one is 110, the other is 011. If the crossover position is at the third bit, then the two child individuals are 111 and 010. Similarly, for the parent 110, if the third bit is the mutation point, then the new child becomes 111.

The genetic algorithm repeats the process above continuously, until the evolution generation reaches the given bound. The result is the optimal solution of the higher-order-moment portfolio optimization selected by the genetic algorithm.

3. A hybrid approach for higher-order-moment portfolio optimization

In this section, we present our hybrid approach for the portfolio optimization problem, which includes not only the multi-objective optimization but also the data-driven asset selection and return prediction. We adopt the genetic algorithm mentioned in Section 2, and the techniques of two-stage clustering and radial basis function (RBF) neural network in the hybrid approach. We summarize the framework of the hybrid approach for solving higher-order-moment portfolio optimization problem in Fig. 2. The detailed flowchart of our method is illustrated as follows.

Step 1: Calculate the return. When the day t is the ex-dividend date, the daily return rate with cash bonus reinvestment at day t is calculated as

$$r_t = \frac{P_t(1 + B_t + S_t)C_t + D_t}{P_{t-1} + C_tS_tK_t} - 1, \tag{9}$$

where P_t denotes the settlement price, and the symbols B_t, S_t, C_t, D_t and K_t represent the corresponding bonus share, rights issue, split, cash dividend, and rights issue price per share, respectively. When the day t is not the ex-dividend date, the daily return rate is computed as

$$r_t = \frac{P_t}{P_{t-1}} - 1. \tag{10}$$

Step 2: Select assets. We apply the two-stage clustering to analyze the return rates of N risky assets to select $n(< N)$ risky assets to construct the portfolio in model (P_2). The process is described in Algorithm 1. Denote the historical return vectors of the N risky assets are as $a_1, a_2, \ldots, a_N$. At the first stage, the k-nearest-neighbor method is applied to generate $N'(< N)$ sub-clusters. One point a_i is selected as the initial sub-cluster. Then, the following points will be connected into the nearest sub-cluster according to the similarity between them and the existing points. Here we use Euclidean distance to measure the similarity between each two points. For any two arbitrary points, there exists an edge between them if and only if either one of them is one of the nearest k points of the other one. We call this stage as the pre-clustering.

At the second stage, these sub-clusters are merged into final clusters according to their relative inter-connectivity RI and relative closeness RC. The sub-clusters i and j which maximize $RC(SC_i, SC_j) \times RI(SC_i, SC_j)$ will be selected to combine. Finally, we can obtain n clusters. The definitions of RC and RI are as follows:

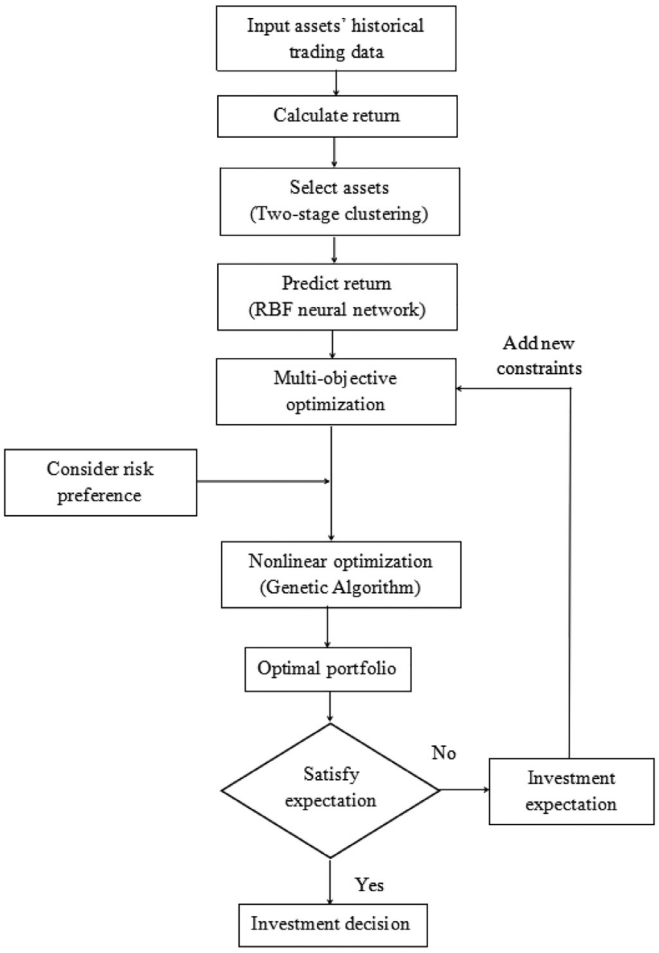


Fig. 2. The framework of the hybrid approach for solving higher-order-moment portfolio optimization problem.

$$RC(SC_i, SC_j) = \frac{|SEW(SC_i, SC_j)|}{(|SEW(SC_i)| + |SEW(SC_j)|)/2}, \tag{11}$$

$$RI(SC_i, SC_j) = \frac{AEW(SC_i, SC_j)}{\frac{|SC_i|}{|SC_i|+|SC_j|}AEW(SC_i) + \frac{|SC_j|}{|SC_i|+|SC_j|}AEW(SC_j)}, \tag{12}$$

where $SEW(SC_i)$ and $SEW(SC_i, SC_j)$ denote the summation of the weights of the edges in the sub-cluster i and that between sub-clusters i and j, respectively. The weight of edge is the reciprocal of distance, and thus represents the similarity. $|SC_i|$ represents the number of data points in the sub-cluster i. $AEW(SC_i)$ is the average of edges' weights in sub-cluster i. $AEW(SC_i, SC_j)$ is the average of connecting edges' weights between sub-clusters i and j.

With the two-stage clustering, we pick out cluster-center points which have the longest distance between each other both in closeness and interconnection, in order to realize the pre-diversification of portfolio risk.

Step 3: Predict the return. Because the investors focus on the future distribution of the return instead of the history distribution, we predict the assets' return rates by RBF neural network trained with the assets' historical return rates.

As shown in Fig. 3, the RBF neural network has three layers: the input layer, the hidden layer with a non-linear RBF function and a linear output layer. The l neurons in the first layer are corresponding to the input vector $y = [y_1, y_2, \cdots, y_l]^T$. Here the input

Algorithm 1 Select assets by two-stage clustering.

Require: $a_1, a_2, \ldots, a_N$
Ensure: n clusters
Stage I: Pre-clustering
Select one point randomly as the initial point, e.g., point a_1;
Initialize $p = 2$
while $p \leq N$ **do**
  Measure the distance between the point a_p and the existing point(s), i.e., $a_1, a_2, \ldots, a_i$ $(i < p)$ by Euclidean distance successively;
  if either one of a_p and a_i is one of the nearest k points of the other one **then**
    Connect the point a_p to the sub-cluster, to which the point a_i belongs, and generate the corresponding edge;
  end if
  $p \leftarrow p + 1$;
end while
Let the number of sub-clusters be N' and sub-cluster be SC.
Stage II: Re-clustering
Initialize $i = 1$
while $i < N'$ **do**
  $j \leftarrow i + 1$;
  while $j \leq N'$ **do**
    Compute: $SEW(SC_i)$, $SEW(SC_i, SC_j)$, $AEW(SC_i)$, $AEW(SC_i, SC_j)$;
    Compute: $RC(SC_i, SC_j)$, $RI(SC_i, SC_j)$;
    if $RC \times RI$ is maximal **then**
      Merge SC_i and SC_j into a new cluster;
      Set the new cluster at the end of queue;
      $N' \leftarrow N' - 1$;
    end if
    $j \leftarrow j + 1$;
  end while
  $i \leftarrow i + 1$;
end while
return n final clusters;

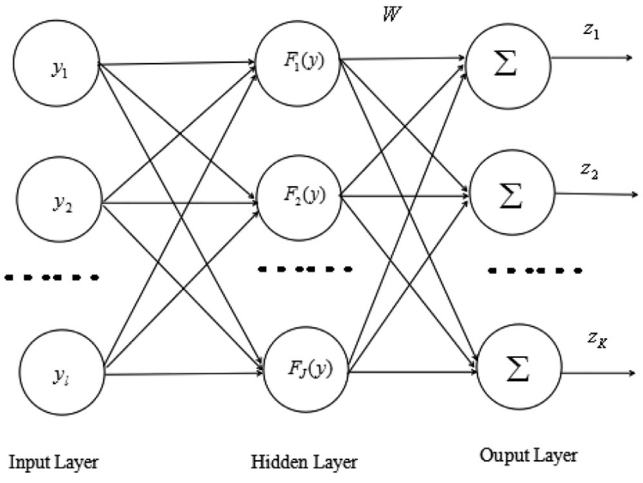


Fig. 3. The structure of RBF neural network.

vector y is each asset's history return and l is the length of history data. We choose the optimal value of l which minimizes the mean square error of prediction. The second layer is the hidden layer with J neurons. We set the kernel function as the Gaussian function. It implies that the output of neuron j, $F_j(y)$ can be expressed as

$$F_j(y) = exp\left(-\left\|\frac{y - c_j}{\sigma_j}\right\|^2\right), \quad j = 1, 2, \cdots, J. \tag{13}$$

The parameter c_j is the basis function's central parameter of neuron j, and σ_j is the corresponding width parameter. $\|\cdot\|$ is the Euclidean norm. The last layer is the output layer with K neurons. We calculate the weighted summation of the information from the hidden layer to get the network's final output $z = [z_1, z_2, \cdots, z_K]^T$ as

$$z_k = \sum_{j=1}^{J} w_{kj} F_j(y), \quad k = 1, 2, \cdots, K, \tag{14}$$

where w_{kj} is the connecting weight between output neuron k and hidden neuron j. In this paper, the output vector y is each asset's future return predicted by RBF neural network.

The key of RBF neural network is the update of the three sets of parameters, c_j, σ_j and w_{kj}, which are determined by the training process presented in Algorithm 2. The network whose prediction accuracy meets certain requirements is used to forecast return distribution of the portfolio in the future. After the prediction, we can calculate the four target functions, the mean, variance, skewness and kurtosis of the portfolio's future return, for each portfolio.

Algorithm 2 RBF neural network's training progress.

Given the maximal iteration steps T, learning rate η, adjusting rate β and threshold ϵ;
Initialize $c_j(0), \sigma_j(0)$, and $w_{kj}(0)$;
Initialize $t = 1$;
Define: z_{mk} is the real output of output neuron k on input sample m, o_{mk} is the corresponding desired output;
while $t \leq T$ **do**
  Compute: evaluation function $E \leftarrow \frac{1}{2}\sum_{m=1}^{l}\sum_{k=1}^{K}(z_{mk} - o_{mk})^2$;
  if $t = 1$ **then**
    $c_j(t) \leftarrow c_j(t-1) - \eta\frac{\partial E}{\partial c_j(t-1)}$;
    $\sigma_j(t) \leftarrow \sigma_j(t-1) - \eta\frac{\partial E}{\partial \sigma_j(t-1)}$;
    $w_{kj}(t) \leftarrow w_{kj}(t-1) - \eta\frac{\partial E}{\partial w_{kj}(t-1)}$;
  end if
  if $\frac{\partial E}{\partial c_j(t-1)} < \epsilon$ **then**
    $c_j(t) \leftarrow c_j(t-1)$;
  end if
  if $\frac{\partial E}{\partial \sigma_j(t-1)} < \epsilon$ **then**
    $\sigma_j(t) \leftarrow \sigma_j(t-1)$;
  end if
  if $\frac{\partial E}{\partial w_{kj}(t-1)} < \epsilon$ **then**
    $w_{kj}(t) \leftarrow w_{kj}(t-1)$;
  end if
  $c_j(t) \leftarrow c_j(t-1) - \eta\frac{\partial E}{\partial c_j(t-1)} + \beta[c_j(t-1) - c_j(t-2)]$;
  $\sigma_j(t) \leftarrow \sigma_j(t-1) - \eta\frac{\partial E}{\partial \sigma_j(t-1)} + \beta[\sigma_j(t-1) - \sigma_j(t-2)]$;
  $w_{kj}(t) \leftarrow w_{kj}(t-1) - \eta\frac{\partial E}{\partial w_{kj}(t-1)} + \beta[w_{kj}(t-1) - w_{kj}(t-2)]$;
  $t \leftarrow t + 1$;
end while
return $c_j(t), \sigma_j(t)$, and $w_{kj}(t)$;

Step 4: Set the risk preferences. We consider the risk preferences of investors and set the preference factors in the model (P_2) to search optimal solution with the genetic algorithm in Section 2.1.

Step 5: Search optimal portfolio. The optimal portfolio return is calculated. If it satisfies the investors' expectation, the optimal

Table 1
Two-stage clustering result

Cluster ID	Stock Number	Stock ID	Cluster Center
1	7	601601, 601628, 601318, 601169, 600016, 601088, 601390	600016
2	6	600028, 601857, 600050, 600104, 600837, 601766	600050
3	4	600100, 600519, 600887, 601211	600887
4	9	601198, 601800, 600999, 601985, 600036, 601186, 601901, 601336, 601377	600036
5	7	600958, 601288, 601688, 601818, 601668, 601989, 600030	600030
6	9	600048, 600109, 600111, 600485, 600518, 600547, 600637, 600893, 601998	600518
7	8	601006, 601166, 601328, 601398, 601988, 600000, 600029, 601788	601166

portfolio is the investment decision. Otherwise, the expectations will be added into the multi-objective optimization problem as new constraints to search the optimal solution again.

4. Empirical analysis

In this section, we apply our hybrid algorithm, as stated in Section 3, to real transaction data from Shanghai Stock Exchange in China. We first briefly introduce the dataset we investigate, and then present our results of each step for our sample data in this section, according to the flowchart stated in Section 3.

4.1. Data description

The data set is the daily trading data of SSE 50 index's latest constitutes from Jan. 4th, 2010 to Feb. 20th, 2017, provided by the CSMAR China Stock Market Trading Database.

The SSE 50 index ranks the stocks on Shanghai Stock Exchange based on the total market value and the turnover to choose the top 50 as the index's constituents. Therefore, the index's constituents are the most representative stocks in terms of the transaction size and liquidity in the market.

In particular, to verify the effectiveness and robustness of our proposed method for solving higher-order-moment portfolio optimization model, we apply the data from Jan. 4th, 2010 to Dec. 30th, 2016 as the in-sample to train and test the model, and the data from Jan. 1st, 2017 to Feb. 20th, 2017 (30 trading days) as the out-of-sample to evaluate the performance of the optimization model and the solution found by our method. In fact, we also vary the out-of-sample length from 30 trading days to 100, 250 and 500 trading days, and find that all the performance results are similar. Therefore, we only present the case of 30 trading days in our numerical tests.

4.2. Two-stage clustering result

As described in Section 3, we first calculate the daily return rates with cash bonus reinvestment of each constituent during 2010-2016 according to the historical trading data. Then, we apply the two-stage clustering method to analyze them and document the result in Table 1.

Table 1 presents that the two-stage clustering divides 50 stocks into seven clusters. Unlike the usual classification, which simply clustering the bank shares, security shares and infrastructure shares, the two-stage clustering takes the relative distance of inner characteristics and the hidden pattern in approximation and interconnection as a criterion. Moreover, using these seven stocks instead of all stocks to construct the portfolio will save considerable computing resources for our algorithm to find the optimal solutions.

After clustering, the seven cluster centers are selected to construct the portfolio. Their stock IDs are are 600016, 600050, 600887, 600036, 600030, 600518, and 601166. The statistical description of selected stocks' return vectors are summarized in Table 2.

4.3. Forecasting of RBF neural network

We divide the in-sample data into training set and testing set, i.e., 80% of the data is set to train the RBF neural network, and the other 20 percentage is to test whether the prediction precision of the network satisfies the accuracy requirement. Here, the metric *mean squared error* (MSE) is used to evaluate the forecasting accuracy of the network. In order to avoid over-fitting, we set the target MSE as 0.0001. If its forecasting accuracy meets the target criterion, the trained network is put into use to predict the assets' daily return rates in the future 30 trading dates, i.e., from Jan. 1st, 2017 to Feb. 20th, 2017. Otherwise, the learning process on the training set repeats with adjusted parameters, such as the learning speed, until the accuracy reaches the target criterion. The testing result of forecasting accuracy of RBF neural network is presented in Table 3.

Table 3 shows that the number of in-sample data of different stocks varies during the same observation period. The reason is that different stocks were suspended for various reasons during the period, and that the intervals between suspension and resumption differ in length, which causes different actual trading dates. However, the proportion between the training set and the testing set remains 8:2 for all the stocks. The results in Table 3 show that the MSE of the network prediction on the testing set meets the requirement, i.e., we can apply it to forecast the assets' future return. Table 4 reports the statistical description of the network's prediction on the assets' daily return rate in the future 30 trading days.

4.4. Optimal portfolios obtained by genetic algorithm

Based on the prediction results of the RBF neural network, genetic algorithm (GA) is applied to search the optimal portfolios under different risk preferences. We adopt the same parameter values in the experiment as Davis (1991) and Kacprzyk and Pedrycz (2015): $p_a = 0.0001, L = 20, P = 10^4, N_R = 20, p_c = 0.5$ and $p_m = 0.01$. Fig. 4 presents its fitness variation in the process of solving the higher-order-moment model.

As shown in Fig. 4, the blue and red lines represent the best and mean fitness, respectively. The blue line implies that the convergence rate of GA is fast, and it can converge to a point near the minimum within 100 iterations. After its convergence, the genetic algorithm keeps fine search in a small area, indicated by the fluctuation of its mean fitness in the red line, and then stops evolution at around 520 generations. In general, the performance of GA in solving the model is satisfactory. It is mainly due to the fact that the span of variables is narrow, and the number of chromosomes is sufficient. A smaller search range and a larger population ensure a greater probability to approach the optimal solution.

In Table 5, different values of the investment preference factor (λ_1, λ_2, λ_3, λ_4) represent different risk preferences of an investor, where (1,1,1,1) denotes that the weights on the mean, variance, skewness, and kurtosis of the portfolio are the same, i.e., the investor attaches equal importance to the average, volatility, symmetry, and steep of the return. That is also the foundation of our model, the Mean-Variance-Skewness-Kurtosis model, referred to as

Table 2
Statistical description of selected stocks

ID	Stock name	Number	Mean	Std. Dev.	Skewness	Kurtosis	Min	Max
600016	Minsheng Bank	1691	0.0006	0.0194	0.4799	5.7746	−0.0999	0.1001
600050	China Unicom	1696	0.0003	0.0230	0.4303	5.2698	−0.1005	0.1010
600887	Yili	1665	0.0012	0.0239	0.0494	2.1477	−0.1001	0.1001
600036	China Merchants Bank	1679	0.0004	0.0183	0.5650	4.8578	−0.0991	0.1003
600030	CITIC Securities	1677	0.0003	0.0268	0.2979	3.3499	−0.1001	0.1004
600518	Kangmei	1672	0.0012	0.0258	0.1121	3.2570	−0.1001	0.1002
601166	Industrial Bank of China	1680	0.0004	0.0212	0.3661	4.3090	−0.1002	0.1005

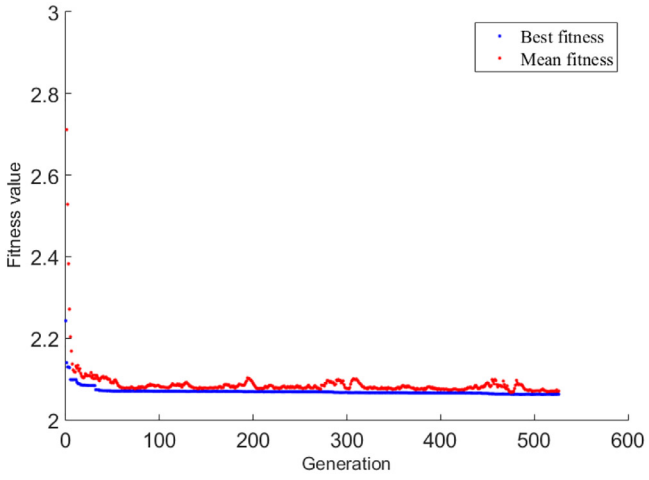


Fig. 4. The fitness variation of GA in solving higher-order-moment model.

Table 3
Testing result of RBF neural network's forecasting accuracy.

ID	In-sample data	Training set	Testing set	MSE
600016	1691	1353	338	0.00001
600050	1696	1357	339	0.00005
600887	1665	1332	333	0.00001
600036	1679	1343	336	0.00002
600030	1677	1342	335	0.00005
600518	1672	1338	334	0.00003
601166	1680	1344	336	0.00001

Table 4
Statistical description of the network prediction in the future 30 trading dates.

ID	Number	Mean	Std. Dev.	Min	Max
600016	30	−0.0002	0.0051	−0.0093	0.0136
600050	30	−0.0036	0.0278	−0.0858	0.0562
600887	30	0.0008	0.0912	−0.0176	0.0201
600036	30	0.0029	0.0081	−0.0148	0.0224
600030	30	0.0015	0.0065	−0.0078	0.0142
600518	30	−0.0008	0.0087	−0.0189	0.0175
601166	30	0.0012	0.0079	−0.0186	0.0178

the M-V-S-K model hereinafter. Similarly, the preference (1,1,1,0) represents that the investor attaches equal importance to the average, volatility, symmetry of the return but ignores its steep. This is the focus of the studies on the Mean-Variance-Skewness model, referred to as the M-V-S model hereinafter. The preference (1,1,0,0) is the Markowitz M-V model in essence, which is taken as the benchmark to evaluate the performance of our considered model and proposed method.

As mentioned above, the value of the risk preference factor implies the investor's emphasis. For example, (2,1,1,1) implies that the investor prefers to pursue a higher return under certain risk, while (1,2,1,1) indicates that the investor emphasizes the volatility of the portfolio more than other factors.

4.5. Comparisons on model performance

In this section, in order to evaluate the performance of M-V-S-K model, the out-of-sample data is used to compare it with the M-V-S model and M-V model in terms of the investment yield, risk management (Sharpe Ratio) and error tracking (Information Ratio). In particular, we consider the M-V model with the risk preference

Table 5
Optimal portfolios under different risk preferences.

Risk Preferences	600016	600050	600887	600036	600030	600518	601166
(1,1,0,0)	0.0283	0.3200	0.2492	0.0540	0.2134	0.0426	0.0924
(1,1,1,0)	0.0429	0.1693	0.0798	0.1748	0.0651	0.4540	0.0141
(1,1,1,1)	0.1323	0.0874	0.0697	0.1018	0.4101	0.1317	0.0669
(2,1,1,1)	0.0192	0.1248	0.0796	0.2651	0.2731	0.1719	0.0663
(1,2,1,1)	0.0435	0.1037	0.0711	0.2804	0.0336	0.2699	0.1977
(1,1,2,1)	0.0452	0.0633	0.0211	0.3073	0.0001	0.4872	0.0757
(1,1,1,2)	0.0279	0.1183	0.0417	0.2659	0.1823	0.1626	0.2014
(1,2,1,2)	0.1428	0.1213	0.0711	0.1742	0.0464	0.1409	0.3033
(2,1,2,1)	0.0277	0.0284	0.0500	0.3108	0.1487	0.3403	0.0940

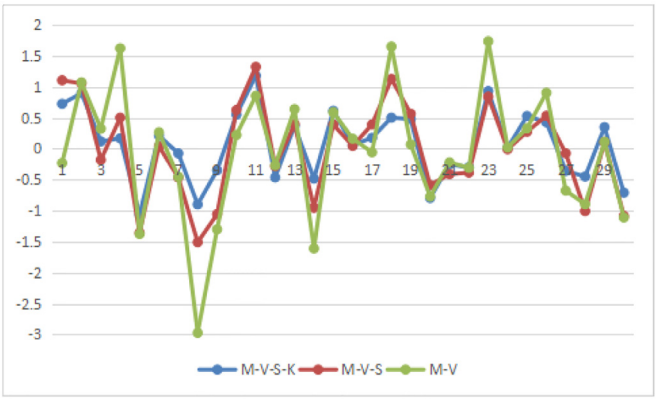


Fig. 5. The daily return rates (%) of three portfolio optimization models from Jan. 3rd to Feb. 20th, 2017.

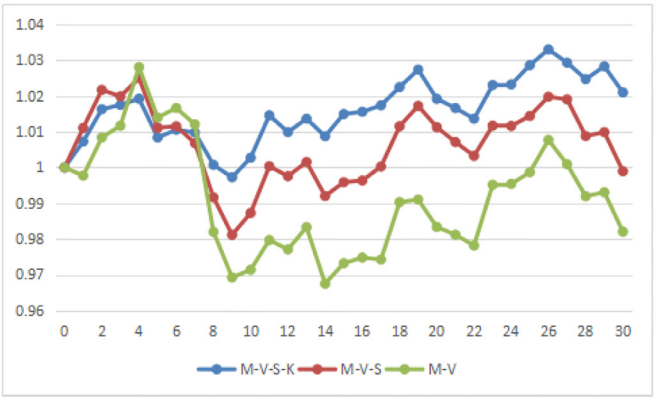


Fig. 6. Accumulated gains of the three models from Jan. 3rd to Feb. 20th, 2017.

Table 6
Comparison on yields of the portfolios selected by the three models (%).

Model	Out-of-sample	Daily	Annualized	250-trading-date
M-V-S-K	**2.1011**	**0.0693**	**28.3414**	**18.9200**
M-V-S	−0.1051	−0.0035	−1.2542	−0.8726
M-V	−1.7967	−0.0604	−19.5522	−14.0225

$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (1, 1, 0, 0)$, the M-V-S model with the risk preference (1,1,1,0) and the M-V-S-K model with the risk preference (1,1,1,1).[2]

Fig. 5 shows the realized daily return rate of the three models from Jan. 3rd to Feb. 20th in 2017. The fluctuation amplitude of the daily return rate of M-V portfolio[3] is the largest, whose maximum is over 1.5%, and minimum is close to −3%. By comparison, the daily return rate of portfolio selected by M-V-S model fluctuation is smaller, while that of M-V-S-K model is the smallest with less than 1.3% at high points and around −1% at the bottom.

Intuitively, the M-V portfolio performs worst on the risk management among the three portfolios. Although it might achieve a high yield when the stock market rises, its loss will also be quite heavy once the market plunges. On the contrary, the performance of M-V-S-K portfolio is relatively stable. As it considers the symmetry and steep degrees of the return on the basis of average and volatility, the M-V-S-K model is able to make a faster response to the share price adjustment.

Table 6 further reports the quantitative comparison on the return rate of the three models. The M-V-S-K portfolio performs best on the investment return among all the three portfolios, and its out-of-sample cumulative yield reaches 2.1011%, while the yield of M-V portfolio is lowest as −1.7967%. The M-V-S portfolio has a slight loss in the out-sample with −0.1051% cumulatively. According to the compound interest calculation formula, the out-of-sample cumulative yield can be converted into the daily, annualized, 250-trading-date yield. It is worth pointing out that, because there are only 250 trading dates in China's stock market, the 250-trading-date yield may own more practical reference value than the annualized one. After the conversion, the 250-trading-date yield of M-V-S-K portfolio reaches 18.9200%, which is impressive, while the loss of M-V model in 250 trading dates reaches 14.0225%.

Fig. 6 describes accumulated gains of the out-of-sample of the optimal portfolios selected by the three models from Jan. 3rd to Feb. 20th, 2017. Here, the initial principal is set to be one unit. According to Fig. 6, the accumulated gain of M-V portfolio is far below those of M-V-S and M-V-S-K portfolios during the trading period except the first several days. In the end, the M-V-S-K portfolio earns a profit, and the M-V-S portfolio stays near break-even. The M-V portfolio makes a heavy loss. This is consistent with the analysis above. It proves that, when the skewness and kurtosis are considered in portfolio selection, the model can respond to the market fluctuation more quickly according to the change of return distribution's shape.

In addition, we next introduce several common quantitative indicators to summarize the overall out-of-sample performances of the three optimal portfolios. As we know, the *sharpe ratio* is the commonest indicator to measure the portfolio risk-adjusted return, i.e., divide the average return by its standard deviation to estimate the portfolio return to the system risk. However, the average return of M-V model is negative in value, and the negative sharpe ratio is meaningless in reality, so we will not compare the risk-adjusted return of the models here. As an alternative, we choose the *information ratio* to evaluate the error-tracking ability of a portfolio. In fact, in the last decade, the indicator *information ratio* has been adopted by a number of studies or reports to investigate new portfolios' performances (See, e.g., Israelsen (2005) and the references therein).

The *information ratio*, namely *IR*, takes the M-V model as benchmark to measure the excess return per unit of the active risk. We thus calculate *IR* according to the following equation,

$$IR = \frac{E(R_p - R_b)}{\sqrt{var(R_p - R_b)}}, \tag{15}$$

where R_b is the return of M-V portfolio, and R_p is the return of the evaluated portfolio. The numerator is the excess return, while the denominator represents the active risk. $IR > 0$ indicates that the evaluated portfolio performs better than the M-V portfolio, while $IR < 0$ implies that its performance is worse than that of M-V portfolio. Therefore, a higher value of IR illustrates that the portfolio performance on error-tracking is better.

Table 7 presents that the *IR* values of M-V-S-K and M-V-S portfolios are both higher than zero, which indicates that they perform better than M-V on the return adjusted by the active risk. Furthermore, it is worth noting that the active risk of M-V-S-K portfolio is slightly larger than that of M-V-S portfolio, but the excess return of M-V-S-K portfolio is much higher than that of M-V-S portfolio, so the *IR* of M-V-S-K portfolio is also higher. It implies that the error-tracking performance of M-V-S-K portfolio is better than that of M-V-S portfolio.

[2] The setting is in line with Yu, Wang, and Lai (2008). We also apply the other sets of risk preferences and find similar results.

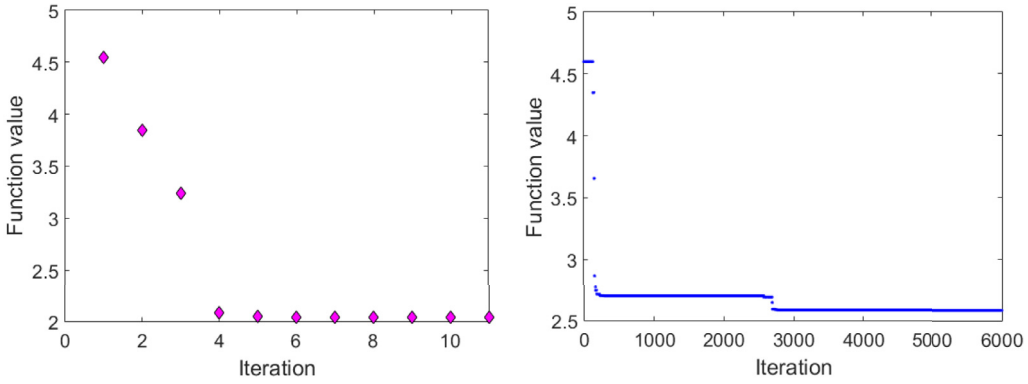
[3] To simplify the expression, we will denote the portfolios selected by M-V, M-V-S and M-V-S-K models by M-V, M-V-S and M-V-S-K portfolios, respectively.

Table 7
Overall performances of out-of-sample of the three optimal portfolios.

Model	Average Return	System risk	Excess return	Active risk	*IR*	Min	Max	Final value
M-V-S-K	**0.0709**	**0.0057**	**0.1262**	0.0066	**0.1911**	**0.9972**	**1.0330**	**1.0210**
M-V-S	−0.0006	0.0077	0.0546	**0.0054**	0.1010	0.9812	1.0250	0.9989
M-V	−0.0553	0.0103	/	/	/	0.9675	1.0281	0.9820

Table 8
Solutions to the M-V-S-K model (P_2) provided by the three algorithms. The numbers are the optimal strategies, i.e., the weights of these stocks in the portfolio, calculated by the three algorithms.

Method	600016	600050	600887	600036	600030	600518	601166
GA	0.1323	0.0874	0.0697	0.1018	0.4101	0.1317	0.0669
MPFM	0.0621	0.1162	0.0940	0.2399	0.1238	0.2370	0.1271
SA	0.1267	0.1252	0.1225	0.1230	0.0884	0.1352	0.2789



(a) MPFM (b) SA

Fig. 7. The variation of objective functions of MPFM and SA methods in solving the M-V-S-K model.

Table 9
Comparison on the yields of the three portfolios (%).

Method	Out-of-sample	Daily	Annualized	250-trading-date
GA	**2.1011**	**0.0693**	**28.3414**	**18.9200**
MPFM	1.9425	0.0642	25.9695	17.3894
SA	1.3608	0.0451	17.6095	11.9228

5. Robustness of our hybrid method

This section further compares the solutions of the M-V-S-K model obtained by different algorithms, in order to test the robustness of our hybrid method. We also present some computational time analysis of our hybrid method and other alternatives.

For the purpose of a comparison, we choose another two algorithms, *the mixed penalty function method* (MPFM) and *the simulated annealing algorithm* (SA). The former one combines the advantages of exterior and interior penalty and is able to solve nonlinear programming containing both equality and inequality constraints. The latter one does not depend on the initial value strongly, and hardly falls into the local optimum trap. In fact, the *simulated annealing algorithm* has been applied to solve portfolio selection problem and got an excellent performance (Crama & Schyns, 2003), while the *mixed penalty function method* is an efficient mathematical tool for nonlinear programming (Bertsekas, 2016).

The presentation of the rest of this section is consistent with Sections 4.4 and 4.5. We use the prediction results of the RBF neural network, but apply GA, MPFM, and SA to solve the M-V-S-K model separately to compare these three algorithms. Table 8 presents numerical results of the three algorithms on the M-V-S-K model.

Fig. 7 shows the variation of objective function values of MPFM and SA methods in solving the M-V-S-K model respectively. It is observed that the convergence rate of MPFM is quite fast: it takes only eleven generations to stop iterating. By contrast, the convergence rate of SA is much lower. It takes over 2000 iterations to converge to a point that near the optimum. Then, it conducts the small-area fine search for a long time, and stops computation at about 6000 generations. In Section 4.4, GA iterates less than 600 generations in total to find the optimal solution, and also performs the long-time fine search (see Fig. 4). It indicates that MPFM may perform better than those of GA and SA in terms of the convergence rate, but may miss a potential better solution. We thus furthermore compare the performances of the portfolios selected by these three algorithms (the GA, SA and MPFM portfolios for short) in the following part.

Fig. 8 presents the daily return rates of the three algorithms form Jan. 3rd to Feb. 20th, 2017. Their variations are quite similar. The fluctuation of SA method is slightly greater than those of GA and MPFM, but the nuance between those of GA and MPFM is hardly distinguished. Hence, further calculations on relevant quantitative indicators are necessary to compare the performances of the portfolios optimized by the three algorithms.

Table 9 reports the comparison on the yields of the three portfolios. According to Table 9, the out-of-sample cumulative yield of SA portfolio is lower than those of GA and MPFM portfolios. GA performs best on the investment return, whose 250-trading-date yield is about 1.5% higher than that of MPFM portfolio, and around 7% higher than that of SA portfolio.

Fig. 9 presents the accumulated gains of the three algorithms from Jan. 3rd to Feb. 20th, 2017. The initial principal is one unit. The accumulated gain of SA portfolio is almost the same as MPFM

Table 10
Overall performances of out-of-sample of the three portfolios.

Method	Average return	System risk	Sharpe ratio	Excess return	Active risk	*IR*	Min	Max	Final value
GA	**0.0709**	**0.0057**	**12.341**	**0.1262**	0.0066	0.1911	**0.9972**	1.0330	**1.0210**
MPFM	0.0663	0.0067	9.8534	0.1216	**0.0061**	**0.2003**	0.9936	**1.0350**	1.0194
SA	0.0473	0.0068	6.9509	0.0853	0.0076	0.1122	0.9910	1.0310	1.0136

Table 11
Comparisons of time complexity.

Different parts of algorithm	Two-stage Clustering	RBF Neural Network	GA
Running time (in s)	13.6244	12.3718	16.2050
Different searching methods	GA	MPFM	SA
Running time (in s)	16.2050	11.7691	32.1475

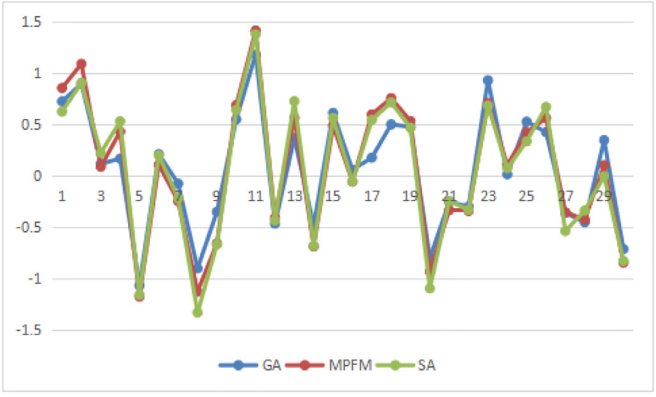


Fig. 8. The daily return rates of the three portfolios from Jan. 3rd to Feb. 20th, 2017.

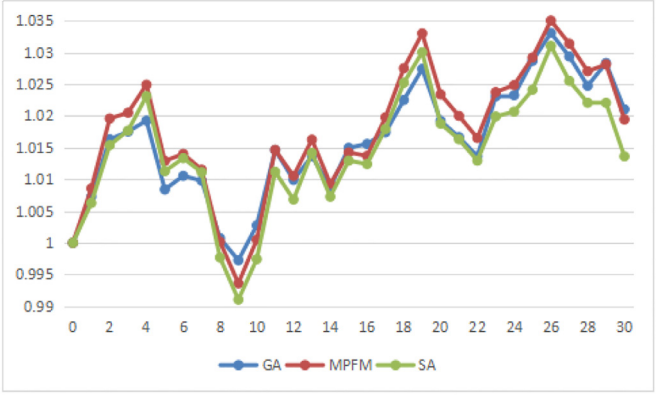


Fig. 9. The accumulated gains of the three portfolios from Jan. 3rd to Feb. 20th, 2017.

portfolio, and higher than GA portfolio during the first eight trading days. However, in the latter trading dates, the performances of GA and MPFM portfolios are more robust on quick stop-loss and the short-time profits in batch. The final value of SA portfolio is much lower than that of MPFM portfolio, whereas the final value of MPFM portfolio is slightly lower than that of GA portfolio.

Table 10 describes overall out-of-sample performances of the three algorithms via quantitative indicators. First, we compare the return adjusted by the system risk. As it performs best on both the mean and the standard deviation of return, GA portfolio undoubtedly has the highest sharpe ratio. Hence, from the perspective of the overall system risk, GA portfolio is better than MPFM portfolio, which is better than SA portfolio.

Secondly, from the viewpoint of active risk, the *IR* of MPFM portfolio is little larger than that of GA portfolio, which is mainly due to the fact that the active risk of MPFM portfolio is a bit smaller. Combining the results shown in Table 7, the optimal portfolio suggested by GA portfolio does not have the lowest active risk, partially due to the fact that we set the penalty factor P as a large fixed value 10^4, which increases the active risk of the model inevitably. Nevertheless, the penalty factor of MPFM portfolio is not determined artificially in advance, but iterates and updates in the process of searching the optimal solution. Hence, the performance of GA is a bit inferior to that of MPFM on the error tracking. Combining GA with the mixed penalty algorithm may help GA to reduce the active risk, and further improve its error-tracking ability. It is worth pointing out that the *IR* of portfolio provided by GA is still higher than that by SA significantly.

We furthermore compare the computational time of our method with the identified alternatives discussed above. The results are shown in Table 11. We present the average running times of these three searching algorithms and the main three parts of our hybrid algorithm in 50 trials for solving M-V-S-K model. We see that the total running time of our hybrid algorithm is 42.2 s on average, which consists of 13.6 s for the asset selection by the two-stage clustering method, 12.4 s for the return prediction by RBF neural network, and 16.2 s for the model optimization by GA. As for the different searching methods, we find that the average running time of GA is between those of the other two alternatives, i.e., slightly longer than that of MPFM (11.8 s), but much shorter than that of SA (32.1 s). Taken together the overall performances in investment yield and error tracking, the consuming time of our method is reasonable.

To summarize, our proposed hybrid method provides the optimal portfolio with robust overall performances and acceptable computational times.

6. Conclusion and future work

This paper studies the portfolio selection problem with higher-order moments. Aiming at extending the classical mean-variance model, this paper introduces skewness and kurtosis of the portfolio to construct the mean-variance-skewness-kurtosis (M-V-S-K) model. To solve our model, we transform the multi-objective optimization problem into a non-linear programming model with risk preferences. Furthermore, we propose a hybrid approach including three machine learning algorithms to select the pre-diversified assets, predict the future returns of these assets and numerically construct the optimal portfolio to invest.

The empirical analysis with historical data from Shanghai Stock Exchange in China shows that the out-of-sample performance of the optimal portfolio selected by our M-V-S-K model is better

than those by the classical mean-variance model and the mean-variance-skewness model. We also compare the genetic algorithm with the other widely used algorithm to find that the overall performance of genetic algorithm (GA) we applied is appealing. Specifically, the performance of the optimal portfolio provided by GA on yield and risk management is superior to those by SA and MPFM, the other two widely used approaches for non-linear programming. Our hybrid algorithm serves the purpose as a good portfolio selection algorithm for investors and hedge funds.

For higher-order-moment portfolio selection, there are still many interesting problems remaining open for future research. For example, one may notice that we only discuss several sets of widely-used risk preference parameters in our paper. Optimizing the higher-order-moment portfolio selection problem, with the consideration of all possible risk preference parameters, to identify the efficient frontier is worth investigating. In addition, we consider static portfolio selection problem in this paper. A dynamic higher-order-moment portfolio selection problem could be non-separable and remains open for future research.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Bilian Chen: Conceptualization, Methodology, Formal analysis, Validation, Supervision, Investigation, Writing - review & editing, Resources, Funding acquisition. **Jingdong Zhong:** Conceptualization, Methodology, Formal analysis, Software, Investigation, Data curation, Writing - original draft, Visualization. **Yuanyuan Chen:** Conceptualization, Methodology, Resources, Formal analysis, Investigation, Data curation, Visualization, Writing - review & editing.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grants nos. 61772442, 11671335 and 61836005).

References

Adcock, C. J. (2014). Mean-variance-skewness efficient surfaces, Stein's lemma and the multivariate extended skew-student distribution. *European Journal of Operational Research, 234*(2), 392–401. doi:10.1016/j.ejor.2013.07.011.

Aksaraylı, M., & Pala, O. (2018). A polynomial goal programming model for portfolio optimization based on entropy and higher moments. *Expert Systems with Applications, 94*, 185–192.

Bertsekas, D. P. (2016). *Nonlinear programming: 3rd edition*. Nashua.: Athena Scientific.

Briec, W., Kerstens, K., & Jokung, O. (2007). Mean-variance-skewness portfolio performance gauging: A general shortage function and dual approach. *Management Science, 53*(1), 135–149. doi:10.1287/mnsc.1060.0596.

Chen, C., & Zhou, Y.-s. (2018). Robust multiobjective portfolio with higher moments. *Expert Systems with Applications, 100*, 165–181.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187–205. doi:10.1016/j.eswa.2017.04.030.

Chunhachinda, P., Dandapani, K., Hamid, S., & Prakash, A. J. (1997). Portfolio selection and skewness: Evidence from international stock markets. *Journal of Banking & Finance, 21*(2), 143–167. doi:10.1016/S0378-4266(96)00032-5.

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational research, 150*(3), 546–571. doi:10.1016/S0377-2217(02)00784-1.

Cui, X., Li, D., Wang, S., & Zhu, S. (2012). Better than dynamic mean-variance: Time inconsistency and free cash flow stream. *Mathematical Finance, 22*(2), 346–378.

Davis, L. (1991). *Handbook of genetic algorithms*. New York.: Van Nostrand Reinhold.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications, 124*, 226–251. doi:10.1016/j.eswa.2019.01.012.

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2016). Parsimonious time series clustering using p-splines. *Expert Systems with Applications, 52*, 26–38. doi:10.1016/j.eswa.2016.01.004.

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2018). A p-spline based clustering approach for portfolio selection. *Expert Systems with Applications, 95*, 88–103. doi:10.1016/j.eswa.2017.11.031.

Israelsen, C. (2005). A refinement to the sharpe ratio and information ratio. *Journal of Asset Management, 5*(6), 423–427. doi:10.1057/palgrave.jam.2240158.

Joro, T., & Na, P. (2006). Portfolio performance evaluation in a mean–variance–skewness framework. *European Journal of Operational Research, 175*(1), 446–461. doi:10.1016/j.ejor.2005.05.006.

Kacprzyk, J., & Pedrycz, W. (2015). *Springer handbook of computational intelligence*. Springer.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer, 32*(8), 68–75. doi:10.1109/2.781637.

Konno, H., & Suzuki, K. (1992). A fast algorithm for solving large scale mean-variance models by compact factorization of covariance matrices. *Journal of the Operations Research Society of Japan, 35*(1), 93–104. doi:10.15807/jorsj.35.93.

Li, D., & Ng, W.-L. (2000). Optimal dynamic portfolio selection: Multiperiod mean–variance formulation. *Mathematical Finance, 10*(3), 387–406.

Liu, D., Han, L., & Han, X. (2016). High spatial resolution remote sensing image classification based on deep learning. *Acta Optica Sinica, 36*(4), 306–314.

Liu, L. (2004). A new foundation for the mean–variance analysis. *European Journal of Operational Research, 158*(1), 229–242. doi:10.1016/S0377-2217(03)00301-1.

Maringer, D., & Parpas, P. (2009). Global optimization of higher order moments in portfolio selection. *Journal of Global Optimization, 43*(2), 219–230. doi:10.1007/s10898-007-9224-3.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91. doi:10.1111/j.1540-6261.1952.tb01525.x.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications, 115*, 635–655. doi:10.1016/j.eswa.2018.08.003.

Paravisini, D., Rappoport, V., & Ravina, E. (2016). Risk aversion and wealth: Evidence from person-to-person lending portfolios. *Management Science, 63*(2), 279–297.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259–268. doi:10.1016/j.eswa.2014.07.040.

Saranya, K., & Prasanna, P. K. (2014). Portfolio selection and optimization with higher moments: Evidence from the indian stock market. *Asia-Pacific Financial Markets, 21*(2), 133–149.

Steinbach, M. C. (2001). Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM Review, 43*(1), 31–85.

Yu, L., Wang, S., & Lai, K. K. (2008). Neural network-based mean–variance–skewness model for portfolio selection. *Computers & Operations Research, 35*(1), 34–46. doi:10.1016/j.cor.2006.02.012.

Zhai, J., Bai, M., & Wu, H. (2018). Mean-risk-skewness models for portfolio optimization based on uncertain measure. *Optimization, 67*(5), 701–714.

Zhang, X., Liu, J., Du, Y., & Lv, T. (2011). A novel clustering method on time series data. *Expert Systems with Applications, 38*(9), 11891–11900. doi:10.1016/j.eswa.2011.03.081.