A multi-factor and high-order stock forecast model based on Type-2 FTS using cuckoo search and self-adaptive harmony search

Article *in* Neurocomputing · February 2017

DOI: 10.1016/j.neucom.2017.02.054

Shixiong Zhang
City University of Hong Kong

Shuai Zhang
Zhejiang University of Finance and Economics

Dejian Yu
Zhejiang University of Finance and Economics

Neurocomputing 000 (2017) 1–12

Contents lists available at ScienceDirect

Neurocomputing

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

A multi-factor and high-order stock forecast model based on Type-2 FTS using cuckoo search and self-adaptive harmony search

Wenyu Zhang*, Shixiong Zhang, Shuai Zhang, Dejian Yu, NingNing Huang

School of Information, Zhejiang University of Finance and Economics, 18 Xueyuan Street, Xiasha, Hangzhou 310018, China

ARTICLE INFO

Article history:
Received 26 March 2016
Revised 11 February 2017
Accepted 16 February 2017
Available online xxx

Communicated by Dr. J. Zhang

Keywords:
Type-2 fuzzy time series
Cuckoo search
Harmony search
High orders and multi-factors
Stock forecast

ABSTRACT

With the developing economy, an increasing number of factors and high-order data have considerably influenced fluctuations in stock. The fuzzy time series (Type-1) model has been recently studied by many scholars. In this study, an enhanced model on the study of stock forecasting is proposed. This model is a multi-factor and high-order time series forecast model that the Type-2 fuzzy time series model by integrating several other factors. We first employ the cuckoo search algorithm instead of the conventional average method to partition the universe of discourse, and then propose a novel self-adaptive harmony search algorithm to optimize the high-order weight. Furthermore, the Shanghai Stock Exchange Composite Index and Taiwan Stock Exchange Capitalization Weighted Stock Index are used to verify the better performance of the proposed method. Experimental results show that the proposed method outperforms other baseline methods.

1. Introduction

With the rapid development of modern technology and the complication and diversification of the economy and market conditions, financial investment that includes stocks, futures, and funds has superseded the conventional models of investment. Quantitative financial analysis, considered to be a novel and high-tech investment method, has been extensively researched in recent years. Because of its volatility and randomness, the stock market, produces huge returns for investors but can also generate considerable loss. Therefore, many time series forecasting models and methods are proposed to mitigate loss and risk. Form the conventional statistical point of view, some financial time series models have been proposed, such as the autoregressive moving average (ARMA) [1], autoregressive conditional heteroscedastic (ARCH) [2], generalized autoregressive conditional heteroscedastic (GARCH) [3], and Copula models [4]. These models are all restricted by certain hypotheses for meta-data such as normality or autocorrelation. However, these hypotheses are not very suitable for complex real-world data. With the development of machine learning and deep learning, some unrestricted models, from the non-statistical perspective, have been applied to forecasting stock indexes. Chen and Kao [5] and Cao [6] used a support vector machine to forecast a stock index. Yu and Xu [7] and Xiao et al. [8] proposed forecasting models using an artificial neural network. Chen and Hwang [9] proposed a hybrid forecasting model by using a deep learning algorithm to fit the time series. The aforementioned models and methods all used real-valued data in the process of forecasting. However, other researchers indicated that the time series forecasting model could employ a linguistic value because the native stock market includes both nonlinear and vague data [10–12].

Fuzzy time series (FTS), first proposed by Song and Chissom [10–12], is a method that uses linguistic-valued data to construct a forecasting model. This method, which is based on fuzzy sets [13], has been applied to many fields such as enrollment forecasting [10–12], electric load forecasting [7,8], and temperature forecasting [9]. Stock index forecasting in particular has attracted much attention from researchers (e.g., Shanghai Stock Exchange Composite Index (SSECI) forecasting [14], Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) forecasting [5], and many others). In these application fields, FTS has shown strong forecasting ability. However, all of these methods adopt the Type-1 fuzzy set concept. As an extension of the Type-1 fuzzy set, Zadeh [15] proposed the theory of the Type-2 fuzzy set, and Huarng and Yu [16] applied it to financial time series forecasting. Because the Type-2 fuzzy set has higher performance in nonlinear, uncertain, and complicated systems, it is suitable for use in modeling FTS forecasting [17,18]. Type-2 FTS integrates many factors into the model, such as “open,” “high,” “low,” and “closed” indexes. Thus, Type-2 FTS is also a multi-factor time series. Within Type-1 FTS, some studies have focused on multi-factors [19,20], high orders [21,22], and

* Corresponding author.
E-mail addresses: wyzhang@e.ntu.edu.sg (W. Zhang), zhangsx@zufe.edu.cn (S. Zhang), zs760914@sina.com (S. Zhang), yudejian62@126.com (D. Yu), hnn0702@126.com (N. Huang).

http://dx.doi.org/10.1016/j.neucom.2017.02.054
0925-2312/

multi-factors and high orders [23,24] FTS and have proposed models that utilize fully historical data. For Type-2 FTS, few works have addressed the high-order problem.

The basic steps of Type-2 FTS include partitioning the universe of discourse, fuzzification, finding the fuzzy logical relationships, and defuzzification. Partitioning the universe of discourse is the most important step, which is critical to improve the prediction accuracy. Many researchers have undertaken significant efforts to address this issue. For example, Cai et al. [25] used ant colony optimization, Egrioglu [26] applied particle swarm optimization, and Chen et al. [20] utilized fuzzy C-means. In this study, we propose a novel Type-2 FTS forecasting model to address this problem. This model applies the cuckoo search algorithm, which is a population-based global optimization evolutionary algorithm, to partition the universe of discourse. We also present a novel self-adaptive harmony search (SAHS) algorithm based on Type-2 FTS for the high-order Type-2 FTS forecasting model. Finally, we perform empirical and comparative analyses to verify the effectiveness and practicability of our proposed model by using SSECI and TAIEX.

The rest of paper is organized as follows. In Section 2, we briefly review the related works covered in this study. In Section 3, we present the proposed novel Type-2 FTS forecasting model. Section 4 provides an illustrative example that contains an empirical and comparative analysis. Section 5 concludes the paper.

2. Related works

In this section, we review the basic concepts of Type-1 and Type-2 FTS, cuckoo search, and harmony search. For each part, we will point out the shortcomings and insufficiency as well as our improvements.

2.1. Type-1 FTS

Song and Chissom [10–12] introduced the theory of fuzzy sets to time series forecasting, and this concept can be called Type-1 FTS. Below are the basic concepts of Type-1 FTS explained through some definitions.

Definition 1 [10]**.** U is the universe of discourse, and it is partitioned into n intervals. It can be expressed as $U=\{u_1, u_2,..., u_n\}$. Thus, A is a fuzzy set and can be represented as

$$A = \frac{f_A(u_1)}{u_1} + \frac{f_A(u_2)}{u_2} + \ldots + \frac{f_A(u_n)}{u_n} \tag{1}$$

where $f_A(u_i)$, within [0,1], represents the membership of A that belongs to interval u_i.

Definition 2 [11]**.** If A_j is the fuzzification data $F(t)$ at time t and A_i is the fuzzification data $F(t-1)$ at time t-1, we can consider that there is a fuzzy logical relationship (FLR) between A_i and A_j. It can be represented as

$$A_i \to A_j \tag{2}$$

In other words, this FLR can be regarded as first-order Type-1 FTS, which means that A_j, referring to right-hand side (RHS), is determined only by A_i, referring to left-hand side (LHS).

Definition 3 [27]**.** If there are many RHSs determined by only one LHS, then the many FLRs form a FLR group (FLRG). For example:

$$\begin{aligned} &A_i \to A_{j1} \\ &A_i \to A_{j2} \\ &\ \ \ldots \\ &A_i \to A_{jk} \end{aligned} \tag{3}$$

In the definition of Chen's [27] proposed model, these FLRs form a FLRG, say:

$$A_i \to A_{j1}, A_{j2}, \ldots, A_{jk} \tag{4}$$

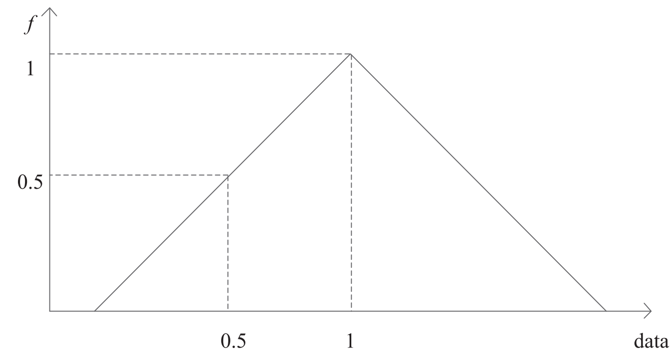


Fig. 1. Type-1 fuzzy set.

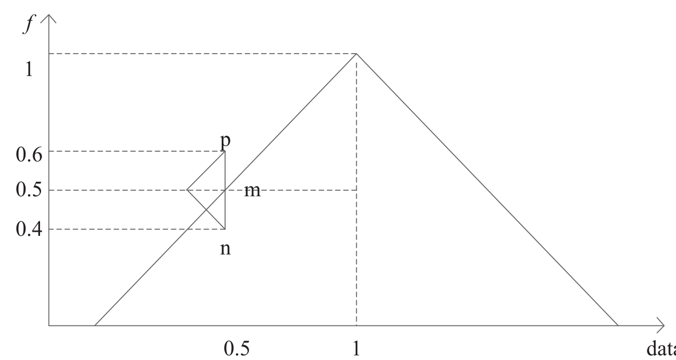


Fig. 2. Type-2 fuzzy set.

2.2. Type-2 FTS

The aforementioned Type-1 FTS model considers only one variable to forecast in the process of establishing a logical relationship. To use more information from other observations, Huarng and Yu [16] introduced the theory of the Type-2 fuzzy set into FTS forecasting. The Type-2 FTS model is also called a multivariable model. Below are the concepts of the basic Type-2 FTS model.

Definition 4 [16]**.** A Type-2 FTS model is extended from the Type-1 FTS model. In essence, through some certain operations that integrate other variables' observations, the membership degree of Type-2 FTS is extended from one to three dimensions. For example, in Fig. 1, based on Type-1 fuzzy sets, for the data value = 1, the corresponding value of the crisp membership degree is 1; for the data value = 0.5, the corresponding value of the crisp membership degree is 0.5. In Fig. 2, based on the Type-2 fuzzy set, there is a triangular fuzzy set (0.4, 0.5, 0.6) that is the corresponding value of the membership degree for the data value = 0.5.

Definition 5. As mentioned in Definition 4, to integrate other variables of observations, Huarng and Yu [16] defined two operators: union ($\vee$) and intersection ($\wedge$). These two operators include and screen out the FLRs from Type-1 and Type-2 FLRGs. The basic structure of union ($\vee$) and intersection ($\wedge$) is

$$\vee(LHS_d, LHS_e) = RHS_d \cup RHS_e \tag{5}$$

$$\wedge(LHS_d, LHS_e) = RHS_d \cap RHS_e \tag{6}$$

where $\cup$ represents the union in Eq. (5) and $\cap$ represents the intersection in Eq. (6) operator for set theory. LHS_d (LHS_e) and RHS_d (RHS_e) represent the LHS and RHS of an FLRG, respectively.

Please cite this article as: W. Zhang et al., A multi-factor and high-order stock forecast model based on Type-2 FTS using cuckoo search and self-adaptive harmony search, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.054

Definition 6 [16]**.** For the operation of multiple FLRGs, the union and intersection operators are defined as follows:

$$\begin{aligned}&\vee_m(LHS_c, LHS_d, LHS_e, \ldots)\\&= \vee\ldots(\vee(\vee(LHS_c, LHS_d), LHS_e), \ldots)\\&= (RHS_c \cup RHS_d \cup RHS_e \cup \ldots)\end{aligned} \tag{7}$$

$$\begin{aligned}&\wedge_m(LHS_c, LHS_d, LHS_e, \ldots)\\&= \wedge\ldots(\wedge(\wedge(LHS_c, LHS_d), LHS_e), \ldots)\\&= (RHS_c \cap RHS_d \cap RHS_e \cap \ldots)\end{aligned} \tag{8}$$

where LHS_c, LHS_d, LHS_e,... and RHS_c, RHS_d, RHS_e,... represent LHS_s and RHS_s of FLRGs, respectively.

Definition 7 [16]**.** In the process of union and intersection operators, there are some special cases:

(a) If $\vee_m(LHS_c, LHS_d, LHS_e, \ldots) = \emptyset$, then let $\vee_m(LHS_c, LHS_d, LHS_e, \ldots) = LHS_x$, where LHS_x is obtained from the FLRG established by Type-1 FTS observations.
(b) If $\wedge_m(LHS_c, LHS_d, LHS_e, \ldots) = \emptyset$, then let $\wedge_m(LHS_c, LHS_d, LHS_e, \ldots) = LHS_x$, where LHS_x is obtained from the FLRG established by Type-1 FTS observations.

2.3. Cuckoo search

Cuckoo search, proposed in 2009 by Cambridge scholars Yang and Deb [28], is a population-based global optimization evolutionary algorithm, which has been widely applied in academic research and industrial applications for optimization problems. Many optimization evolutionary algorithms have been proposed in recent years, including genetic algorithm, simulated annealing, ant colony optimization, and particle swarm optimization. However, research has shown that cuckoo search outperforms the others in global search. Cuckoo search is inspired by the nature of brood parasitism among cuckoos. Cuckoo search has three hypotheses defined by Yang and Deb [28]: 1) each cuckoo lays only one egg one time and randomly selects a parasitic nest to hatch it; 2) in a group of randomly selected parasitic nests, the best parasitic nest will be retained by the next generation; 3) the number of available parasitic nests is fixed, and a host of parasitic nests can find an exotic bird's eggs within a probability P_α.

Considering that the Type-2 FTS must first divide the meta-data into intervals, in this paper, we use the global search ability of the cuckoo search algorithm to partition the universe of discourse (time series meta-data).

2.4. Harmony search

In 2001, a new meta-heuristic optimization algorithm, harmony search, was proposed by Geem et al. [29]. Harmony search is inspired by vocal and instrumental music performance. In the process of playing music, the musician, with his/her memory, repeatedly adjusts the band of tones in various types of musical instruments. Through many adjustments, a harmony state will be reached [29]. Since this algorithm was put forward, it has been widely applied to solve combinatorial optimization problems and has achieved excellent results. Poursalehi et al. [30] applied harmony search to reactor core fuel management optimization. Fesanghary et al. [31] used harmony search to optimize the design of shell and tube heat exchangers. In the field of financial time series prediction, Dash et al. [32] proposed a self-adaptive differential harmony search based on an optimized extreme learning machine. Within these applications, most have made modifications or improvements, e.g., Poursalehi et al. [30] and Dash et al. [32] both extended the harmony search with self-adaptive ability. These improvements revolve around the two parameters of harmony search: *PAR* (pitch adjustment rate) and *bw* (bandwidth).

Thus, in this paper, to address the issue of high-order Type-2 FTS forecasting, we propose a novel SAHS algorithm. The two parameters (*PAR* and *bw*) will be adjusted by the improvement rate of the new solution. Detailed descriptions will be shown in the next section.

3. A high-order and multi-variable Type-2 FTS forecasting combined with cuckoo search and SAHS

Based on Huarng and Yu [16]'s Type-2 FTS model, our proposed high-order Type-2 FTS model has the architecture shown in Fig. 3.

The steps of the extended Type-2 FTS model are illustrated below:

***Step* 1**. Pick a variable and high-order Type-1 observations.

In this paper, we take the daily closing price of SSECI as the forecasting variable and main factor, which is the Type-1 observation. Because Type-2 FTS is a multi-variable model, we select three SSECIs: daily open, low, and high price as the Type-2 observations. Meanwhile, we consider n-orders ($n \geq 2$) observations. We then convert the closing price to percentage change, which is calculated as follows:

$$\text{percentage\_change}_t = \frac{\text{closing\_price}_t - \text{closing\_price}_{t-1}}{\text{closing\_price}_{t-1}} \tag{9}$$

Next, the maximum and minimum values of the percentage change are calculated to construct the scope of all observations, including Type-1 and Type-2 observations. The scope is represented by $U = [D_{min}, D_{max}]$, where D_{min} and D_{max} refer to the maximum and minimum values, respectively.

***Step* 2**. Use cuckoo search to partition the universe of discourse.

In this step, we explain how the cuckoo search algorithm (we choose the standard cuckoo search algorithm proposed in [28]) partitions the universe of discourse. As in Step 1, we define the scope $U = [D_{min}, D_{max}]$, which will be partitioned into n suitable intervals. Here is the specific expression:

$$U = [D_{\min}, p_1) \cup [p_1, p_2) \cup \ldots \cup [p_{n-2}, p_{n-1}) \cup [p_{n-1}, D_{\max}] \tag{10}$$

To apply cuckoo search and for the convenience of representation, we use the right interval-boundary points to represent the intervals, as described below:

$$M = (p_1, p_2, \ldots, p_{n-1}, p_n) \tag{11}$$

where $p_1, p_2, \ldots, p_{n-1}, p_n$ are the right interval-boundary points in Eq. (10), and we use it as the n intervals.

Because the scope contains n intervals, we just need n-1 split points: $M' = (p_1, p_2, \ldots, p_{n-1})$.

In the following, we introduce the basic steps of the cuckoo search algorithm through the pseudo-code depicted in Fig. 4.

Some points must be clarified:

(1) *Initialize the populations*. Each host nest $x_i = (p_{i1}, p_{i2}, \ldots, p_{in-1})$ is a solution, which randomly generates ($D_{\min} < p < D_{\max}$);
(2) *Fitness function*. We use the measure function of the k-means algorithm to construct our fitness function, as follows:

$$CS\text{fitness} = \frac{1}{1 + \sum_{i=1}^{n} (\text{dataset}_i - \text{averagevalue}_i)^2} \tag{12}$$

where $i(1 < i < n)$ represents the ith interval, dataset_i represents the data that belong to the ith interval, and averagevalue_i represents the average value of data that belong to the ith interval.

Our objective is to obtain the highest fitness value *CS*fitness through iteration.

(3) *Levy flight* [28]. The new solution is generated using Eqs. (13) and (14), which describe the Levy flight, referring to a ran-

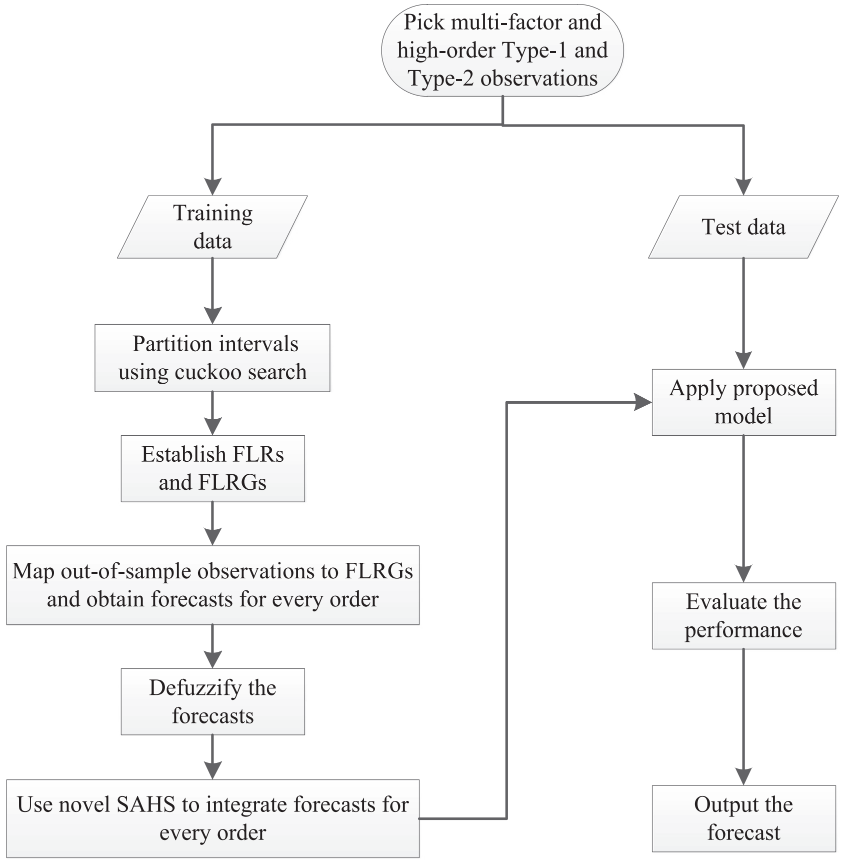


Fig. 3. The proposed novel Type-2 FTS model that extends the Type-2 FTS model.

Cuckoo search algorithm

Begin
Initialize the populations: n host nests(solutions) $x_i(i=1,2,\cdots,n)$
Calculate the fitness value $F_i(i=1,2,\cdots,n)$
While *(Stop criterion)*
Generate a new solution (Cuckoo) x_k via Levy flight and calculate the corresponding fitness value F_k
Select a net x_j randomly and the corresponding fitness value F_j
If *$(F_k>F_j)$*
replace x_j by x_k
End if
Abandon a solution by p_α $(0<p_\alpha<1)$ and rebuild a new solution randomly
Keep the best solution
End while
End

Fig. 4. The pseudo code of cuckoo search algorithm.

dom walk process.

$$\begin{cases} x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \\ Levy(\lambda) = 0.01 \times (\frac{u}{|v|})^{1/\lambda} \times (x_i^t + x_{best}^t) \end{cases} \quad (13)$$

$$\begin{cases} u \sim N\left(0, \sigma_u^2\right), v \sim N(0,1) \\ \sigma_u = \left[\dfrac{\Gamma(\lambda)\sin(0.5\pi(\lambda-1))}{2^{(\lambda-2)/2}\Gamma(0.5\lambda)(\lambda-1)}\right]^{1/(\lambda-1)} \end{cases} \quad (14)$$

In Eq. (13), ${x^{t+1}}_i$ and ${x^t}_i$ refer to solution i at iterations $t+1$ and t, respectively. x_{best}^t is the one that has the best fitness in every iteration. α denotes the step size and, in general, is always taken as 1. The operational symbol $\oplus$ represents entry-wise multiplication. $\Gamma(\lambda)$ is the Gamma function.

In Eq. (14), u and v represent the normal distribution with mean 0 and variance σ_u^2 as 1, respectively; the standard deviation is calculated from the second equation of Eq. (14).

(4) *Stop criterion*. In this study, we set two rules as the stop criterion: Rule 1, which says the best fitness value is not updated; Rule 2, which says the iteration process ends when it reaches the maximum number of iteration. Through repeated experiments, we observe the best fitness value change and set 500 as the maximum number of iterations in this study.

Step 3. Fuzzify observations and establish FLRs and obtain FLRGs.

After step 2, we have n suitable intervals. Take the March 2005 SSECI data as an example, the whole scope is U=[−5.50%, 9.52%]. The partitioned suitable interval set ($n=10$) is

$$\begin{aligned} U = {} & [-5.50\%, -2.15\%) \cup [-2.15\%, -1.29\%) \cup \\ & [-1.29\%, -0.61\%) \cup [-0.61\%, -0.10\%) \cup \\ & [-0.10\%, 0.42\%) \cup [0.42\%, 1.30\%) \cup \\ & [1.30\%, 2.51\%) \cup [2.51\%, 4.16\%) \cup \\ & [4.16\%, 6.26\%) \cup [6.26\%, 9.52\%] \end{aligned}$$

Table 1
The fuzzified Type-1 and Type-2 observations between 3/1/2005 and 3/31/2005.

Date	Closing	Percent (%)	Fuzzy set	Open	Percent (%)	Fuzzy set	High	Percent (%)	Fuzzy set	Low	Percent (%)	Fuzzy set
3/1/2005	1303.412	−0.1984	A_4	1305.249	−0.4720	A_4	1308.758	−0.4850	A_4	1295.642	−0.2530	A_4
3/2/2005	1287.45	−1.2246	A_3	1303.31	−0.1486	A_4	1316.723	0.6086	A_6	1285.916	−0.7507	A_3
3/3/2005	1294.339	0.5351	A_6	1285.197	−1.3898	A_2	1295.253	−1.6306	A_2	1276.88	−0.7027	A_3
3/4/2005	1287.714	−0.5118	A_4	1294.156	0.6971	A_6	1299.596	0.3353	A_5	1286.481	0.7519	A_6
3/7/2005	1293.739	0.4679	A_6	1288.408	−0.4442	A_4	1296.88	−0.2090	A_4	1287.74	0.0979	A_5
3/8/2005	1318.271	1.8962	A_7	1298.25	0.7639	A_6	1318.895	1.6975	A_7	1296.908	0.7119	A_6
3/9/2005	1316.791	−0.1123	A_4	1321.208	1.7684	A_7	1326.076	0.5445	A_6	1310.235	1.0276	A_6
3/10/2005	1286.233	−2.3206	A_1	1316.9	−0.3261	A_4	1316.9	−0.6920	A_3	1282.528	−2.1147	A_2
3/11/2005	1289.941	0.2883	A_5	1284.513	−2.4593	A_1	1292.529	−1.8506	A_2	1278.153	−0.3411	A_4
3/14/2005	1293.5	0.2759	A_5	1288.985	0.3481	A_5	1304.112	0.8962	A_6	1271.511	−0.5197	A_4
3/15/2005	1269.144	−1.8830	A_2	1293.132	0.3217	A_5	1293.259	−0.8322	A_3	1266.637	−0.3833	A_4
3/16/2005	1255.589	−1.0680	A_3	1267.052	−2.0168	A_2	1268.187	−1.9387	A_2	1247.893	−1.4798	A_2
3/17/2005	1243.475	−0.9648	A_3	1255.985	−0.8734	A_3	1257.794	−0.8195	A_3	1242.513	−0.4311	A_4
3/18/2005	1227.403	−1.2925	A_2	1242.453	−1.0774	A_3	1248.034	−0.7760	A_3	1223.615	−1.5209	A_2
3/21/2005	1231.046	0.2968	A_5	1226.143	−1.3127	A_2	1232.817	−1.2193	A_3	1220.824	−0.2281	A_4
3/22/2005	1206.922	−1.9596	A_2	1230.682	0.3702	A_5	1231.676	−0.0926	A_5	1202.427	−1.5069	A_2
3/23/2005	1201.649	−0.4369	A_4	1205.755	−2.0255	A_2	1218.011	−1.1095	A_3	1194.322	−0.6741	A_3
3/24/2005	1208.192	0.5445	A_6	1200.257	−0.4560	A_4	1208.608	−0.7720	A_3	1187.581	−0.5644	A_4
3/25/2005	1205.34	−0.2117	A_4	1207.203	0.5787	A_6	1211.16	0.2112	A_5	1199.87	1.0348	A_6
3/28/2005	1200.113	−0.4579	A_4	1204.85	−0.1949	A_4	1204.85	−0.5210	A_4	1185.456	−1.2013	A_3
3/29/2005	1195.018	−0.4245	A_4	1199.691	−0.4282	A_4	1209.229	0.3634	A_5	1194.589	0.7704	A_6
3/30/2005	1172.574	−1.8781	A_2	1192.447	−0.6038	A_4	1192.447	−1.3878	A_2	1171.811	−1.9068	A_2
3/31/2005	1181.236	0.7387	A_6	1168.828	−1.9807	A_2	1181.531	−0.9154	A_3	1162.031	−0.8346	A_3

The intervals' average values are then (−2.52%, −1.66%, −0.94%, −0.33%, 0.19%, 0.79%, 1.77%, 2.93%, 5.09%, 7.68%).

Next, we define the fuzzy sets for the observations. Suppose $n=10$; we then have10 fuzzy sets:

$$A_1 = 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 + \cdots + 0/u_{10}$$
$$A_2 = 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 + \cdots + 0/u_{10}$$
$$\ldots$$
$$A_{10} = 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + \cdots + 1/u_{10}$$

According to the Definition 1 in Sub-Section 2.1, we define the fuzzy sets based on the membership set (0.5, 1, 0.5), that is, a data has three memberships to three intervals and it belongs to the interval whose membership is 1. For example, the percentage change of 3/3/2005 is 0.5251%. Thus, according to the partitioned intervals set U, it has a membership 1 to [0.42%, 1.30%) and membership 0.5 to [−0.10%, 0.42%), and [1.30%, 2.51%) separately. Therefore, the data 0.5251% belongs to A_6.

Step 3–1. Fuzzify observations.

For every SSECI dataset, there is a corresponding fuzzy set. We take the fuzzy sets contained in Step 3 as the benchmark. For example, the SSECI's closing, open, high, and low prices for 3/3/2003 are 1294.339, 1285.197, 1295.253, and 1276.88, respectively, and the corresponding percentage changes are 0.5351%,−1.3898%,−1.6306% and−0.7027%, respectively. On 3/3/2005, the fuzzy set corresponding to the closing percentage change is A_6, open is A_2, high is A_2, and low is A_3. In Table 1, we list the fuzzified Type-1 and Type-2 observations (parts) for 3/1/2005 to 3/31/2005.

Step 3–2. Establish FLRs and obtain FLRGs.

According to Table 1, we make an example to depict the process on how to establish the FLRs and obtain the FLRGs. For the closing percentage change on 3/1/2005, the fuzzy set is A_4, and for the closing percentage change on 3/2/2005, the corresponding fuzzy set is A_3. For two consecutive data, the FLR is then $A_4 \rightarrow A_3$. Considering this rule, we list all FLRGs of Table 1's Type-1 and Type-2 FLR observations in Table 2.

As seen in Table 2, for a given LHS, there are many corresponding RHSs. For instance, $A_7 \rightarrow A_4$, $A_7 \rightarrow A_5$, $A_7 \rightarrow A_6$; these all have similar LHSs and different RHSs; we then establish a FLRG for A_7,that is, $A_7 \rightarrow A_4, A_5, A_6$. Based on this rule, we can establish all FLRGs between 3/1/2005 and 3/31/2005, depicted in Table 3.

Step 4. Map out-of-sample observations to FLRGs and obtain forecasts for every order.

In this step, we map out-of-observations to FLRGs and obtain forecasts for every order and every variable. In this study, we propose a multi-order Type-2 FTS forecasting model, which leverages n-orders information to forecast the stock index. Suppose that we need to forecast $F(t)$; we then map $F(t-1) \rightarrow F(t)_1, F(t-2) \rightarrow F(t)_2, \ldots, F(t-n) \rightarrow F(t)_n$ to FLRGs. For instance, if $n=2$, then we map$F(t-1) \rightarrow F(t)_1, F(t-2) \rightarrow F(t)_2$. For example, if $F(t-1) = A_7(t = 3/8/2005, \textit{SSECI of closing})$, then $F(t-2) = A_6$ and the forecasted $F(t)_1 = A_4, F(t)_2 = A_4, A_7$.

Step 5. Apply union and intersection operators to the FLRGs for all variables.

In Step 4, we obtained the forecasts from each variable and order; we now apply union and intersection operators to integrate all variables for every order of Type-2 forecasting. In Table 4, for example, if $t=3/17/2005$, we integrate *t-1, t-2, t-3*, and *t-4* (four orders) with the intersection operator, respectively.

For $t-4 = 3/11/2005$,

$$\wedge_m(A_5, A_1, A_2, A_4) = \{A_2, A_5\} \cap \{A_5\} \cap \{A_3, A_5, A_6\} \cap \{A_2, A_3, A_4, A_6\} = \{\varnothing\}.$$

For $t-3 = 3/14/2005$,

$$\wedge_m(A_5, A_5, A_6, A_4) = \{A_2, A_5\} \cap \{A_2, A_5\} \cap \{A_2, A_3\} \cap \{A_2, A_3, A_4, A_6\} = \{A_2\}.$$

Of particular note is that for 3/11/2005, the integrated result is {Ø}. According to Definition 7, the LHS of the Type-1 observation (closing percentage change) isA_5, so $\wedge_m(A_5, A_5, A_6, A_4) = A_5$.

Similarly, by applying union operator to *t*-1, *t*-2, *t*-3, and *t*-4, we obtain the following results:

For $t-4 = 3/11/2005$,

$$\vee_m(A_5, A_7, A_6, A_6) = \{A_2, A_5\} \cup \{A_5\} \cup \{A_3, A_5, A_6\} \cup \{A_2, A_3, A_4, A_6\} = \{A_2, A_3, A_4, A_5, A_6\}.$$

Step 6. Defuzzify the forecasts.

We defuzzify the forecasts by Eq. (15). Because every forecast is a set that contains many fuzzy sets, each fuzzy set has a corresponding interval constructed in Step 2. We set the middle value of the interval to be the corresponding real value. Suppose there

Table 2
The FLRs between 3/1/2005 and 3/31/2005.

Type-1 (Closing)	$A_1 \to A_5, A_2 \to A_3, A_2 \to A_4, A_2 \to A_5, A_2 \to A_6, A_3 \to A_2, A_3 \to A_3, A_3 \to A_6, A_4 \to A_1,$ $A_4 \to A_2, A_4 \to A_3, A_4 \to A_4, A_4 \to A_6, A_5 \to A_2, A_5 \to A_5, A_6 \to A_4, A_6 \to A_7, A_7 \to A_4$
Type-2 (Open)	$A_1 \to A_5, A_2 \to A_3, A_2 \to A_4, A_2 \to A_5, A_2 \to A_6, A_3 \to A_2, A_3 \to A_3, A_4 \to A_1,$ $A_4 \to A_2, A_4 \to A_4, A_4 \to A_6, A_5 \to A_2, A_5 \to A_5, A_6 \to A_4, A_6 \to A_7, A_7 \to A_4$
Type-2 (High)	$A_2 \to A_3, A_2 \to A_5, A_2 \to A_6, A_3 \to A_2, A_3 \to A_3, A_3 \to A_5, A_4 \to A_5, A_4 \to A_6,$ $A_4 \to A_7, A_5 \to A_2, A_5 \to A_3, A_5 \to A_4, A_6 \to A_2, A_6 \to A_3, A_7 \to A_6$
Type-2 (Low)	$A_2 \to A_3, A_2 \to A_4, A_3 \to A_3, A_3 \to A_4, A_3 \to A_6, A_4 \to A_2, A_4 \to A_3,$ $A_4 \to A_4, A_4 \to A_6, A_5 \to A_6, A_6 \to A_2, A_6 \to A_3, A_6 \to A_5, A_6 \to A_6$

Table 3
The FLRGs between 3/1/2005 and 3/31/2005.

Type-1(Closing)	Type-2(Open)	Type-2(High)	Type-2(low)
$A_1 \to A_5$	$A_1 \to A_5$	$A_2 \to A_3, A_5, A_6$	$A_2 \to A_3, A_4$
$A_2 \to A_3, A_4, A_5, A_6$	$A_2 \to A_3, A_4, A_5, A_6$	$A_3 \to A_2, A_3, A_5$	$A_3 \to A_3, A_4, A_6$
$A_3 \to A_2, A_3, A_6$	$A_3 \to A_2, A_3$	$A_4 \to A_5, A_6, A_7$	$A_4 \to A_2, A_3, A_4, A_6$
$A_4 \to A_1, A_2, A_3, A_4, A_6$	$A_4 \to A_1, A_2, A_4, A_6$	$A_5 \to A_2, A_3, A_4$	$A_5 \to A_6$
$A_5 \to A_2, A_5$	$A_5 \to A_2, A_5$	$A_6 \to A_2, A_3$	$A_6 \to A_2, A_3, A_5, A_6$
$A_6 \to A_4, A_7$	$A_6 \to A_4, A_7$	$A_7 \to A_6$	
$A_7 \to A_4$	$A_7 \to A_4$		

Table 4
Forecast after Λ_m and V_m.

Date	Forecast	FLRGs	Forecast after Λ_m	Forecast after V_m
11/3 (t-4)	Closing	$A_5 \to A_2, A_5$	A_5	A_2, A_3, A_4, A_5, A_6
	Open	$A_1 \to A_5$		
	High	$A_2 \to A_3, A_5, A_6$		
	Low	$A_4 \to A_2, A_3, A_4, A_6$		
14/3 (t-3)	Closing	$A_5 \to A_2, A_5$	A_2	A_2, A_3, A_4, A_5, A_6
	Open	$A_5 \to A_2, A_5$		
	High	$A_6 \to A_2, A_3$		
	Low	$A_4 \to A_2, A_3, A_4, A_6$		
15/3 (t-2)	Closing	$A_2 \to A_3, A_4, A_5, A_6$	A_2	A_2, A_3, A_4, A_5, A_6
	Open	$A_5 \to A_2, A_5$		
	High	$A_3 \to A_2, A_3, A_5$		
	Low	$A_4 \to A_2, A_3, A_4, A_6$		
16/3 (t-1)	Closing	$A_3 \to A_2, A_3$	A_3	A_2, A_3, A_4, A_5, A_6
	Open	$A_2 \to A_3, A_4, A_5, A_6$		
	High	$A_2 \to A_3, A_5, A_6$		
	Low	$A_2 \to A_3, A_4$		

is a forecast set $F(t) = \{A_1, A_2, \ldots, A_m\}$, and the corresponding real interval average values are $\{h_1, h_2, \ldots, h_m\}$; the defuzzified value is then

$$\text{defuzzification} = \frac{\sum_{i=1}^{m} h_i}{m} \tag{15}$$

where h_i represents the interval average value of fuzzy set in forecast set $F(t) = \{A_1, A_2, \ldots, A_m\}$. The defuzzification is the real forecast value of the forecast set $F(t) = \{A_1, A_2, \ldots, A_m\}$.

For instance, $\text{defuzzification}_{union}$*(3/11/2005)* = 0.68%, $\text{defuzzification}_{intersection}$*(3/11/2005)* = 0.19%, defuzzification *(3/11/2005)*$_1$ = (−0.68% + 0.19%) / 2 = −0.25%;

Step 7. Use novel SAHS to integrate the forecasts for every order.

After Step 6, we obtain the defuzzified forecasts for every order. Because other Type-2 FTS models rarely consider the high-order relations that have an effect on the forecast, we propose a novel SAHS algorithm to construct the coefficients among different orders. Suppose that the defuzzified forecasts $y_1, y_2, \ldots, y_m$ representdefuzzification$_1$, defuzzification$_2$, ..., defuzzification$_n$. Hence, the final predicted value for t is

$$final\_predict\_value = \omega_0 + \omega_1 y_1 + \omega_2 y_2 + \ldots + \omega_m y_m \tag{16}$$

where $\omega_i (1 < i < m)$ represents the coefficient of y_i, and ω_0 is a constant term.

Next, we introduce the proposed novel SAHS:

Step 7–1. Parameter setting.

In this study, our optimized problem can be described as follows:

Minimize the fitness function $f(\omega)$, depicted in Eq. (17), where *final_predict_value* is calculated by Eq. (13), *real_value* is the actual value of the forecast data, and n denotes the number of training data. If the fitness value is greater, the weight value is more suitable.

$$f(\omega) = e^{-\frac{1}{1+\sum_{i=1}^{n}(final\_predict\_value_i - real\_value_i)^2}} \tag{17}$$

where $\omega = (\omega_0, \omega_1, \omega_2, \ldots, \omega_m)$ are the decision variables, and each vector component is between $\omega_{\min}$ and $\omega_{\max}$ in this study. The SAHS algorithm also has other parameters, which are specified as follows: the harmony memory (*HM*) denotes the solution sets; the harmony memory size (*HMS*) is the number of solutions in the *HM*; the harmony memory consideration rate (*HMCR*); pitch adjustment rate (*PAR*); bandwidth (*bw*) and stopping criterion.

Step 7–2. HM Initialization.

In this step, the *HM* is stored with *HMS* harmonies (solutions) generated randomly by Eq. (18).

$$\omega_i^j = \omega_{\min} + r \times (\omega_{\max} - \omega_{\min}) \tag{18}$$

where r is selected randomly from *U(0,1)*, which is a uniform distribution. The initial *HM* is depicted in Eq. (19), which is a matrix

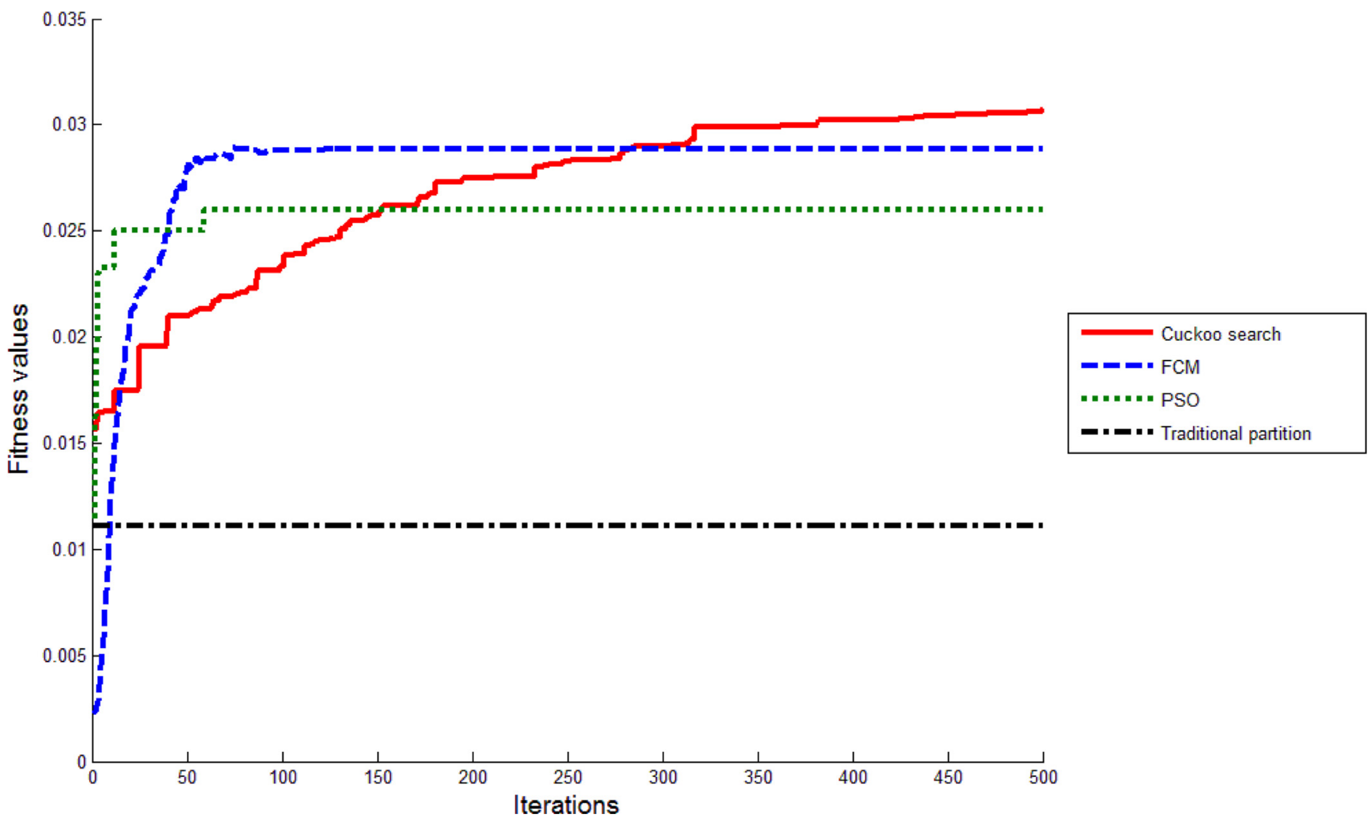


Fig. 5. The performance and comparisons of cuckoo search and other methods.

with size $HMS \times (m+1)$.

$$HM = \begin{bmatrix} \omega_0^1 & \omega_1^1 & \dots & \omega_m^1 \\ \omega_0^2 & \omega_1^2 & \dots & \omega_m^2 \\ \vdots & \vdots & \vdots & \vdots \\ \omega_0^{HMS} & \omega_1^{HMS} & \dots & \omega_m^{HMS} \end{bmatrix} \tag{19}$$

Every row represents a harmony (solution).

Step 7–3. A new harmony improvisation.

A new harmony $\omega' = (\omega_1', \omega_2', \dots, \omega_m')$ is generated (improvised) by two rules:

Rule 1 (memory consideration): At first, generate a random number $rand_1$ between 0 and 1, and then compare it with *HMCR*. If $rand_1 \leq HMCR$, then ω_i' is chosen randomly from column *i* in *HM*; otherwise, it is generated by Eq. (18).

Rule 2 (pitch adjustment): Similarly, generate two random numbers $rand_2$ and $rand_3$ between 0 and 1, and then compare them with *PAR*. If $rand_2 \leq PAR$, then ω_i' is replaced by Eq. (20); otherwise, it is generated by Eq. (18).

$$\omega_i' = \omega_i' \pm rand_3 \times bw, \quad (0 \leq i \leq m) \tag{20}$$

Step 7–4. *HM* update and *PAR* and *bw* self-adaptive update.

After the improvisation based on two rules, we obtain a new harmony ω' and calculate the fitness value $f(\omega')$. If $f(\omega')$ is greater than the worst harmony in HM, then the new harmony ω' will replace it. Galletly [33] presented a 1/5 rule, which claimed that the improvement rate of every iteration evaluation (new solution) should be held at 20%, and then adjusted the parameters to dynamically adapt the algorithm. Because the parameters *PAR* and *bw* have an impact on the diversity of the solution space and search capabilities, we, according to Galletly's 1/5 rule, proposed a self-adaptive parameter adjustment strategy, as shown in Eqs. (21) and (22).

$$PAR = \begin{cases} PAR/2, & IR < 0.1 \\ PAR, & 0.1 \leq < IR \leq 0.2 \\ PAR \times 2, & IR > 0.2 \end{cases} \tag{21}$$

$$bw = \begin{cases} bw/1.05, & IR < 0.1 \\ bw, & 0.1 \leq < IR \leq 0.2 \\ bw \times 1.05, & IR > 0.2 \end{cases} \tag{22}$$

where *IR* denotes the improvement rate. Because *IR* = 20% is single point, the *IR* hardly attach this point, we set *IR* = 10%˜ 20% as the optimal *IR*. When *IR* < 10%, we should reduce the *bw* and *PAR* to enhance the local search capability. When *IR* > 20%, we should increase *bw* and *PAR* to enlarge the search space and stabilize the diversity of the solution.

Step 7–5. Stop criterion check.

Similar to cuckoo search, we still use two rules (Rule 1, the best fitness value is not updated; Rule 2, the iteration process ends when it reaches the maximum number of iteration.) as the stop criterion. Through repeated experiments, the authors observe the best fitness value change and set 500 as the maximum number of iterations in this paper.

After updating *HM*, the stop criterion should be checked as shown in Eq. (23).

$$\begin{cases} \text{Stop} & \text{iteration} \geq \text{maxiteration} \\ \text{Back to Step7} - 3 & \text{iteration} < \text{maxiteration} \end{cases} \tag{23}$$

Step 8. Evaluate the performance.

In this study, we employ Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) to evaluate the

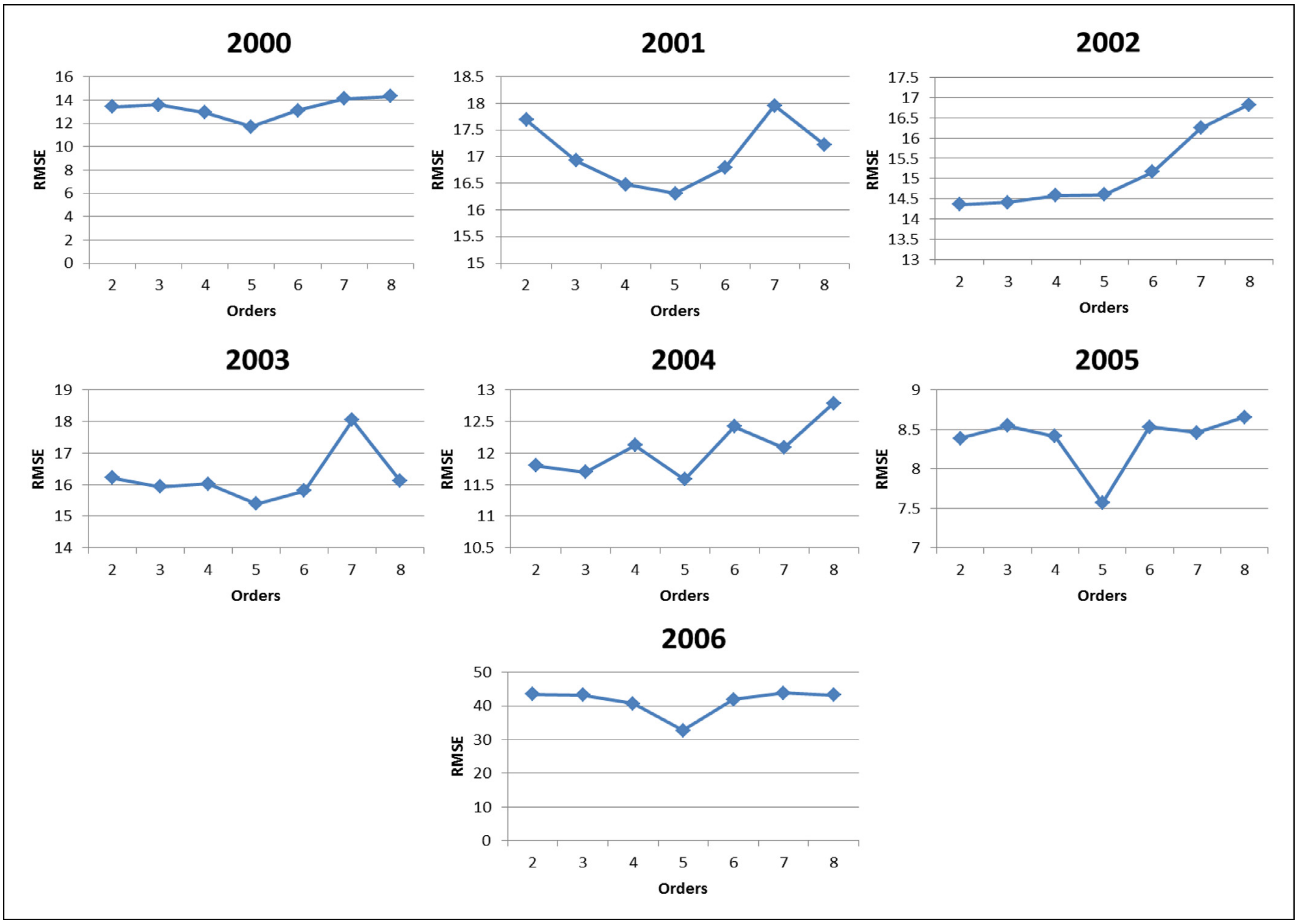


Fig. 6. The RMSE of orders 2–8 from 2000 to 2006.

performance of the proposed model:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{forecast}(i) - \text{actual}(i)}{\text{actual}(i)}\right| * 100 \tag{24}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}[\text{forecasted}(i) - \text{actual}(i)]^2}{n}} \tag{25}$$

where forecast(i) and actual(i) represent the ith forecast value by using the proposed method and the actual SSECI value, separately.

4. Experiment and evaluation

To illustrate the forecast performance of the proposed method, we run several experiments to validate the forecast accuracy by using the SSECI and TAIEX data set on the period from 2000 to 2006. To evaluate the proposed method, we use eight other methods or models (Chen [21], Lee et al. [24], Cheng et al. [20], Huarng and Yu [16], Egrioglu et al. [35], Wang et al. [36], Bas *et al*[37]. and Yolcu et al. [38]) as benchmark approaches for comparison.

The data set is divided into two parts: training and test parts. The first ten months of each year (1–10) are used as a training set, and the last two months (11–12) are used as the test set.

In this study, we select closing price as the main factor (Type-1 observations) and the open, low, and high prices as the secondary factors (Type-2 observations). Furthermore, to determine the optimized number of orders, we run some experiments to calculate it. We choose all data sets from 2000 to 2006. For every year, the data are partitioned into 11 intervals via the cuckoo search. The

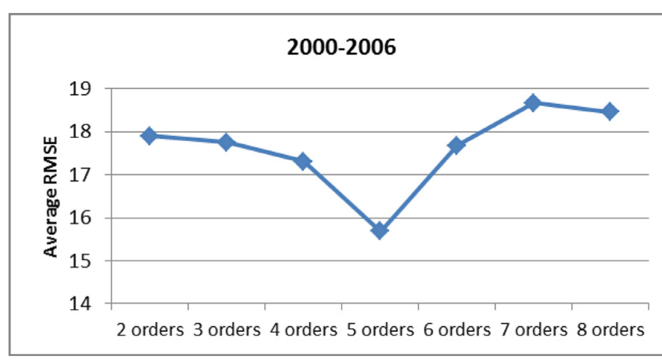


Fig. 7. The average RMSE of orders 2–8 from 2000 to 2006.

performance of the cuckoo search is depicted in Fig. 5. To evaluate the performance, we compare it with fuzzy C-means (FCM) [20], particle swarm optimization (PSO) [25], and the conventional average partition method. In Fig. 5, we can see that the fitness value is higher than other 3 benchmark methods, even though the increased trend is slower than FCM and PSO. FCM is better than PSO for partitioning the universe of discourse. The result demonstrates that the cuckoo search outperforms the other partition methods and achieves the best fitness value.

Figs. 6 and 7 show the RMSE and average RMSE of the SSECI forecast from 2000 to 2006, respectively. As seen, most of the lower RMSEs are present in order 4 or 5. In Fig. 7, the lower RMSEs

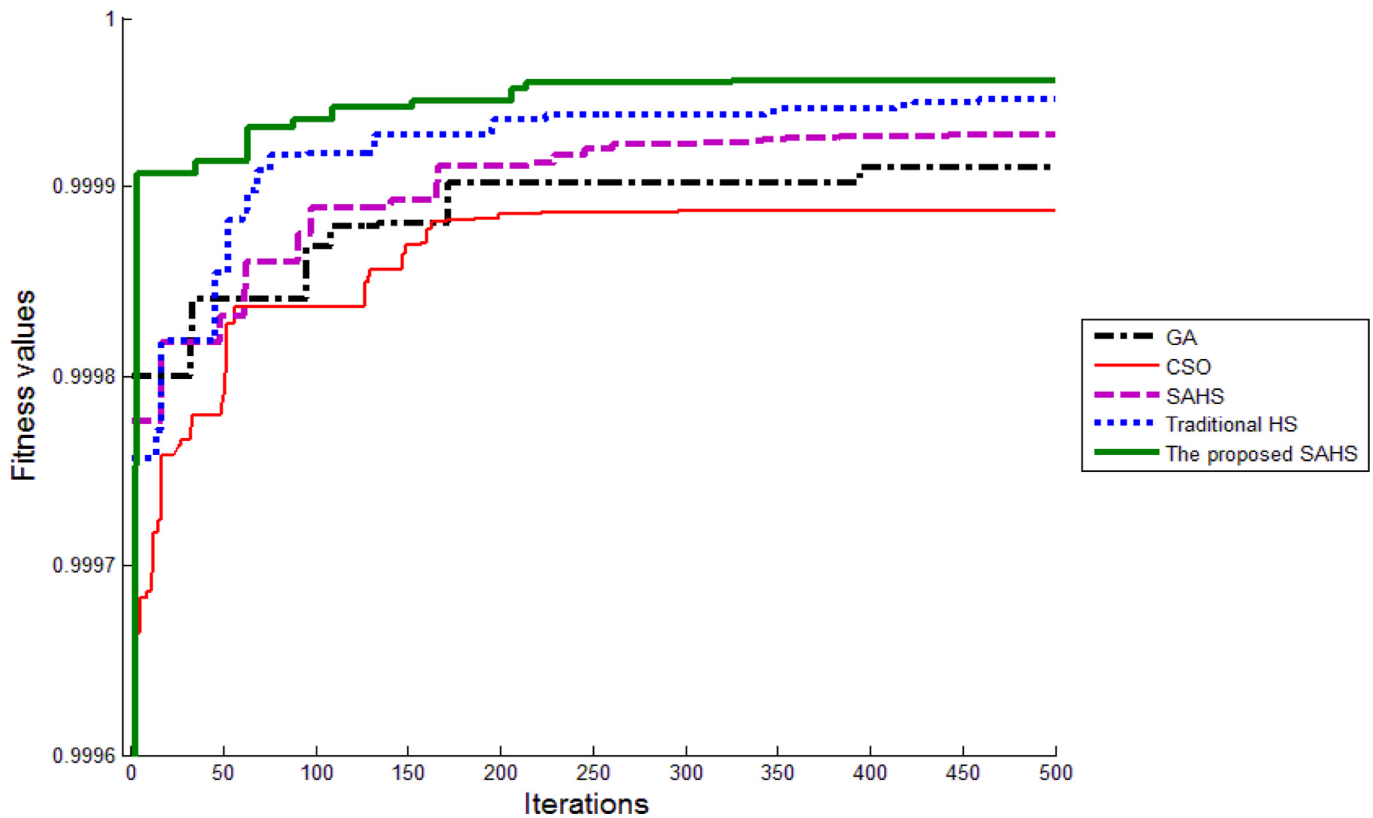


Fig. 8. The performance and comparison of the proposed SAHS and other methods.

Table 5
The parameter setting of the proposed SAHS.

Parameter	Value
HMS	15
$\omega_{\max}$	1
$\omega_{\min}$	−1
HMCR	0.8
PAR (initial value)	0.1
bw (initial value)	1
PAR	[0.1,0.9]
bw	[0.001,1]
Maxiteration	500

are present in order 5. After order 5, the value of RMSE increases. This is because the phenomenon of over fitting appears with increasing order number. To avoid the phenomenon of over fitting, we choose the order 5 as the best order in this study.

In this paper, we proposed a novel SAHS algorithm to optimize the weight of orders, and we compare it with benchmark methods. Table 5 presents the parameter settings of the proposed SAHS. Fig. 8 depicts the performance of the proposed SAHS. To verify the effectiveness and efficiency of the proposed SAHS, we compare it with the genetic algorithm (GA), traditional harmony search (HS), SAHS, and cat swarm optimization (CSO) [34]. In Fig. 8, we can see that the proposed SAHS is better than other four benchmark methods. The convergence speed is also faster than other methods. For every iteration, the fitness value of the proposed SAHS is higher than others. And the traditional HS is only worse than the proposed SAHS, and better than SAHS, GA and CSO. We find that the proposed SAHS outperforms the other methods. This suggests that parameter changing according to the improvement rate is effective for the algorithm within this study.

We implement the proposed method by using the SSECI data set and TAIEX data set from 2000 to 2006 and compare it with eight benchmark approaches, separately. We also evaluate the results through RMSE and MAPE. Chen's [21] method is a high-order FTS model that overcomes the forecasting problem; Lee et al. [24] proposed a two-factor, high-order FTS model, and Cheng et al. [20] proposed a multi-factor FTS model. In particular, the proposed model extends Type-2 FTS, which was proposed by Huarng and Yu [16], and integrates only three factors. Egrioglu et al. [35] proposed a one-order fuzzy time series model. Wang et al. [36] proposed a one-factor method using information granules to forecast. Bas et al. [37] proposed a three-order fuzzy time series model. Yolcu et al. [38] also proposed a three-order fuzzy time series model.

The forecast results are shown below in Tables 6 and 7. It should be noted that the proposed model results are the average of 50 times runs.

Table 6 shows the evaluation and comparison results for SSECI. In Table 6, for each year, the smallest RMSE and MAPE calculated by different methods are shown in bold. Overall, from Table 6, most of the smallest RMSE and MAPE values appear to result from the proposed method. The RMSE of the proposed method is higher than Egrioglu et al. [35] in 2004 and 2005. The MAPE of the proposed method is only higher than Egrioglu et al. [35] in 2005. The average RMSE and MAPE of the proposed method are smaller than those of the other methods. The results demonstrate that our proposed method outperforms the other models for SSECI.

Table 7 shows the evaluation and comparison results for TAIEX. In Table 7, for each year, the smallest RMSE and MAPE calculated by different methods are shown in bold. Overall, from Table 7, most of the smallest RMSE and MAPE values appear to result from the proposed method. The RMSE of the proposed method is higher than that of Egrioglu et al. [35] in 2001. The MAPE of the proposed method are only higher than those of Egrioglu et al. [35] in 2001

Please cite this article as: W. Zhang et al., A multi-factor and high-order stock forecast model based on Type-2 FTS using cuckoo search and self-adaptive harmony search, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.02.054

Table 6
Performance comparisons for SSECI forecast.

Training set (Jan.–Oct.) Test set (Nov.–Dec.)	2000	2001	2002	2003	2004	2005	2006	Average
RMSE								
Huarng and Yu [16]	23.9147	31.9274	31.9575	21.9938	21.7138	14.6053	75.0643	31.5967
Cheng et al. [20]	29.4617	33.5855	33.4515	21.6367	32.0092	12.3227	64.0943	32.3659
Chen [21]	40.768	43.009	57.6315	32.2600	28.4259	16.4664	62.6612	40.1746
Lee et al. [24]	30.5366	48.4292	45.2494	24.1420	22.3151	12.0581	82.0055	37.8194
Egrioglu et al. [35]	17.9911	24.0736	26.3361	18.1261	**12.5963**	**5.9938**	114.9601	31.4396
Wang et al. [36]	43.0975	34.0014	26.4196	17.8860	20.1084	11.8674	379.5415	75.9888
Bas et al. [37]	35.1766	55.1909	55.0887	66.6560	37.5188	27.9020	221.1243	71.3955
Yolcu et al. [38]	34.0485	51.7665	56.8118	65.4207	33.7176	24.0424	226.9612	70.3955
The proposed method	**16.2662**	**20.3227**	**18.0470**	**17.7821**	13.7292	9.0226	**36.5687**	**18.8198**
MAPE								
Huarng and Yu [16]	0.8978	1.5267	1.8158	1.2160	1.3603	1.0471	3.2408	1.5864
Cheng et al. [20]	1.1399	1.7446	1.8727	1.2224	1.9585	0.8838	2.2764	1.5855
Chen [21]	1.7487	2.2552	3.6917	1.8868	1.8290	1.2693	2.4705	2.1645
Lee et al. [24]	1.2189	2.4889	2.4522	1.3871	1.3772	0.8678	3.0686	1.8372
Egrioglu et al. [35]	0.7120	1.1655	1.5943	1.0127	0.8326	**0.4470**	4.7604	1.5035
Wang et al. [36]	1.8593	1.5061	1.5349	0.9059	1.2437	0.8683	13.9241	3.1203
Bas et al. [37]	1.4220	2.8859	3.3800	4.0143	2.4406	1.9489	8.7491	3.5487
Yolcu et al. [38]	1.3502	2.6985	3.5448	4.0378	2.2255	1.8144	9.1566	3.5468
The proposed method	**0.5678**	**0.9102**	**0.9593**	**0.9415**	**0.7358**	0.6169	**1.2471**	**0.8541**

Table 7
Performance comparisons for TAIEX forecast.

Training set (Jan.–Oct.) Test set (Nov.–Dec.)	2000	2001	2002	2003	2004	2005	2006	Average
RMSE								
Huarng and Yu [16]	321.7837	149.9766	89.1342	53.3832	67.1137	67.3667	82.8194	118.7968
Cheng et al. [20]	316.9862	160.8542	114.6180	81.7129	109.4653	83.5819	99.7014	138.1314
Chen [21]	698.1162	326.9913	94.8811	127.5321	87.4035	133.5814	92.4910	222.9995
Lee et al. [24]	277.8274	171.0212	112.4518	128.4544	77.5898	75.3659	84.2257	132.4195
Egrioglu et al. [35]	156.5856	**83.2095**	85.9581	65.6791	61.1491	87.7782	78.7521	88.4445
Wang et al. [36]	225.8288	128.0151	92.0711	132.8846	58.4024	76.2607	222.9535	133.7737
Bas et al. [37]	396.1299	461.3980	123.3420	114.3276	89.6206	209.1707	214.8089	229.8282
Yolcu et al. [38]	383.4513	485.9794	100.4504	89.8964	87.8249	201.4647	201.0883	221.4508
The proposed method	**150.3925**	113.4419	**66.4063**	**53.1497**	**54.9532**	**53.2321**	**52.9927**	**77.7954**
MAPE								
Huarng and Yu [16]	5.3303	2.5661	1.5105	**0.6532**	0.8720	0.8429	0.8708	1.8065
Cheng et al. [20]	5.1934	2.5554	1.9158	0.9612	1.5407	1.0960	1.0944	2.0510
Chen [21]	11.6238	5.0574	1.5830	1.8803	1.2457	1.2588	1.0256	3.3821
Lee et al. [24]	4.3036	3.1452	1.8984	1.6835	1.1137	1.0325	0.9602	2.0196
Egrioglu et al. [35]	2.4339	**1.4507**	1.5602	0.8150	0.8473	0.8730	0.7796	1.2514
Wang et al. [36]	3.3432	2.1666	1.6452	1.9782	0.7225	0.9480	2.4270	1.8901
Bas et al. [37]	6.0479	8.3770	2.0875	1.6396	1.2234	2.6639	2.5143	3.5076
Yolcu et al. [38]	5.6281	8.6650	1.8197	1.3021	1.1818	2.6687	2.3646	3.3757
The proposed method	**2.0702**	1.8828	**1.1299**	0.6908	**0.6642**	**0.6753**	**0.5717**	**1.0978**

and Huarng and Yu [16] in 2003. The average RMSE and MAPE of the proposed method are smaller than those of the other methods. The results demonstrate that our proposed method outperforms the other models for TAIEX.

To fully verify the proposed model, a further comparative experiment was implemented between Ye et al. [39] and the proposed model. Ye et al. analyzed the RMSE of two TAIEX data sections (1999–2004 and 1990–1999). Therefore, in this comparative experiment, we also used the same two TAIEX data sections (1999–2006 and 1990–1999) covering the former. We executed the proposed model 100 times and used the average values as the results like Ye et al. did. The comparative results are shown in Tables 8 and 9.

Table 8 shows the comparative results for TAIEX from 1999 to 2006. In each year, the smallest RMSE are shown in bold. We can see that the number of outperformed RMSE of the proposed model is equal to that by Ye et al. However, the proposed model's average RMSE is higher than that by Ye et al. [39].

Table 9 shows the comparative results for TAIEX from 1990 to 1999. In each year, the smallest RMSE are also shown in bold. We can see that Ye et al.'s model outperforms the proposed model in most years. However, the proposed model's average RMSE is still lower than that by Ye et al.

Table 8
Further performance comparisons for TAIEX forecast from 1999 to 2006.

Training set (Jan.–Oct.) Test set (Nov.–Dec.)	Ye et al. [39]	The proposed method
1999	**101.29**	102.77
2000	**125.42**	150.39
2001	113.22	**112.33**
2002	**63.99**	66.41
2003	52.99	**52.66**
2004	**52.40**	54.95
2005	56.33	**53.23**
2006	60.03	**52.99**
Average	**78.23**	80.71

Table 9
Further performance comparisons for TAIEX forecast from 1990 to 1999.

Training set (Jan.–Oct.) Test set (Nov.–Dec.)	Ye et al. [39]	The proposed method
1990	189.30	**176.25**
1991	**41.74**	43.70
1992	**38.46**	40.35
1993	103.72	**102.94**
1994	**61.90**	69.38
1995	**48.85**	52.99
1996	**50.72**	51.16
1997	115.77	**113.54**
1998	**114.21**	114.93
1999	110.09	**102.77**
Average	87.47	**86.70**

5. Conclusion

In this study, we proposed an enhanced Type-2 FTS model to forecast SSECI. This model addresses the stock forecast problem of multi-factors and high orders. According to the results of our experiments, the proposed method outperforms the other baseline methods.

The main contributions of this study are as follows:

- We proposed an enhanced multi-factor and high-order Type-2 FTS stock forecasting model, in which we introduce multi factors into the forecasting model, and presented an effective algorithm for the selection of the order;
- We applied the cuckoo search algorithm instead of the conventional average method to partition the universe of discourse in the field of stock forecasting;
- We proposed a novel SAHS algorithm to optimize the high-order forecast and entrench order relations.

However, in future studies, additional issues should be addressed such as selecting influence factors and improving the process of partitioning the universe of discourse.

Acknowledgment

This work has been supported by China National Nature Science Foundation (nos 51375429, 51475410, 51175462), Zhejiang Nature Science Foundation of China (no. LY17E050010).

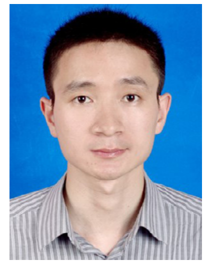
References

[1] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, et al., Time Series Analysis: Forecasting and Control, John Wiley & Sons, 2015.
[2] R.F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, Econom.: J. Econom. Soc. (1982) 987–1007.
[3] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, J. Econom. 31 (3) (1986) 307–327.
[4] M.S. Smith, Copula modeling of dependence in multivariate time series, Int. J. Forecast. 31 (3) (2015) 815–833.
[5] S.M. Chen, P.Y. Kao, TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines, Inform. Sci. 247 (2013) 62–71.
[6] L. Cao, Support vector machines experts for time series forecasting, Neurocomputing 51 (2003) 321–339.
[7] F Yu, X. Xu, A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network, Appl. Energy 134 (2014) 102–113.
[8] Z. Xiao, S.J. Ye, B. Zhong, et al., BP neural network with rough set for short term load forecasting, Expert Syst. Appl. 36 (1) (2009) 273–279.
[9] S.M. Chen, J.R. Hwang, Temperature prediction using fuzzy time series, Syst., Man Cybern., Part B: Cybern., IEEE Trans. 30 (2) (2000) 263–275.
[10] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part I., Fuzzy Sets. Syst. 54 (1) (1993) 1–9.
[11] Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets Syst. 54 (3) (1993) 269–277.
[12] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part II., Fuzzy Sets Syst. 62 (1) (1994) 1–8.
[13] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (3) (1965) 338–353.
[14] W. Qiu, X. Liu, H. Li, A generalized method for forecasting based on fuzzy time series, Expert Syst. Appl. 38 (8) (2011) 10446–10453.
[15] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-1, Inform. Sci. 8 (3) (1975) 199–249.
[16] K. Huarng, H.K. Yu, A type 2 fuzzy time series model for stock index forecasting, Physica A: Stat. Mech. Appl. 353 (2005) 445–462.
[17] M.H.F. Zarandi, B. Rezaee, I.B. Turksen, et al., A type-2 fuzzy rule-based expert system model for stock price analysis, Expert Syst. Appl. 36 (1) (2009) 139–154.
[18] C.F. Liu, C.Y. Yeh, S.J. Lee, Application of type-2 neuro-fuzzy modeling in stock price prediction, Appl. Soft Comput. 12 (4) (2012) 1348–1358.
[19] H.H. Chu, T.L. Chen, C.H. Cheng, et al., Fuzzy dual-factor time-series for stock index forecasting, Expert Syst. Appl. 36 (1) (2009) 165–171.
[20] C.H. Cheng, G.W. Cheng, J.W. Wang, Multi-attribute fuzzy time series method based on fuzzy clustering, Expert Syst. Appl. 34 (2) (2008) 1235–1242.
[21] S.M. Chen, Forecasting enrollments based on high-order fuzzy time series, Cybern. Syst. 33 (1) (2002) 1–16.
[22] H.J. Teoh, T.L. Chen, C.H. Cheng, et al., A hybrid multi-order fuzzy time series for forecasting stock markets, Expert Syst. Appl. 36 (4) (2009) 7888–7897.
[23] S. Askari, N. Montazerin, A high-order multi-variable fuzzy time series forecasting algorithm based on fuzzy clustering, Expert Syst. Appl. 42 (4) (2015) 2121–2135.
[24] L.W. Lee, L.H. Wang, S.M. Chen, et al., Handling forecasting problems based on two-factors high-order fuzzy time series, IEEE Trans. Fuzzy Syst. 14 (3) (2006) 468–477.
[25] Q. Cai, D. Zhang, W. Zheng, et al., A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression, Knowl.-Based Syst. 74 (2015) 61–68.
[26] E. Egrioglu, PSO-based high order time invariant fuzzy time series method: application to stock exchange data, Econ. Model. 38 (2014) 633–639.
[27] S.M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets Syst. 81 (3) (1996) 311–319.
[28] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing, 2009, pp. 210–214.
[29] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[30] N. Poursalehi, A. Zolfaghari, A. Minuchehr, et al., Self-adaptive global best harmony search algorithm applied to reactor core fuel management optimization, Ann. Nucl. Energy 62 (2013) 86–102.
[31] M. Fesanghary, E. Damangir, I. Soleimani, Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm, Appl. Therm. Eng. 29 (5) (2009) 1026–1031.
[32] R. Dash, P.K. Dash, R. Bisoi, A self-adaptive differential harmony search based optimized extreme learning machine for financial time series prediction, Swarm Evol. Comput. 19 (2014) 25–42.
[33] J. Galletly, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Kybernetes 27 (8) (1998) 979–980.
[34] S.K. Saha, S.P. Ghoshal, R. Kar, et al., Cat swarm optimization algorithm for optimal linear phase FIR filter design, ISA Trans. 52 (6) (2013) 781–794.
[35] E. Egrioglu, C.H. Aladag, U. Yolcu, et al., Fuzzy time series forecasting method based on Gustafson–Kessel fuzzy clustering, Expert Syst. Appl. 38 (8) (2011) 10355–10357.
[36] L. Wang, X. Liu, W. Pedrycz, Effective intervals determined by information granules to improve forecasting in fuzzy time series, Expert Syst. Appl. 40 (14) (2013) 5673–5679.
[37] E. Bas, U. Yolcu, E. Egrioglu, et al., A fuzzy time series forecasting method based on operation of union and feed forward artificial neural network, Am. J. Intell. Syst. 5 (3) (2015) 81–91.
[38] O.C. Yolcu, U. Yolcu, E. Egrioglu, et al., High order fuzzy time series forecasting method based on an intersection operation, Appl. Math. Model. 19 (40) (2016) 8750–8765.
[39] F. Ye, L. Zhang, D. Zhang, et al., A novel forecasting method based on multi--order fuzzy time series and technical analysis, Inform. Sci. 367 (2016) 41–57.

Wenyu Zhang received his B.S. degree from Zhejiang University, China in 1989, and his Ph.D. degree from Nanyang Technological University, Singapore in 2002. He is a full professor at the School of Information, Zhejiang University of Finance and Economics, China. He has published over 100 papers in international journals and international conference proceedings in the recent ten years, covering a wide range of business intelligence and manufacturing automation, especially data mining, supply chain management, concurrent engineering, multi-agent technology, and semantic Web.

Shixiong Zhang received his B.S. degree from Zhejiang University of Finance & Economics, China in 2014. He is currently a postgraduate at the school of information, Zhejiang University of Finance and Economics, China. His research interests include fuzzy time series, stock forecast, and machine learning.

Shuai Zhang received his Ph.D. degree in mechanical engineering from Zhejiang University, China in March 2005. He is currently a full-time professor at the school of information, Zhejiang University of Finance and Economics, China. He has published more than 30 papers in international journals in the recent ten years, covering knowledge management, business intelligence, data mining, E-government and manufacturing informatization.



Dejian Yu received the B.S. degree in Information Management and Information System from Heilongjiang University of Science and Technology, Harbin, China in 2006, and the Ph.D. degree in Management Science and Engineering from Southeast University, China, in 2012. He is currently an associated professor with the School of Information, Zhejiang University of Finance and Economics, Hangzhou, China. He has authored more than 30 scientific articles. His current research interests include aggregation operators, information fusion, and multi-criteria decision making.

Ningning Huang is currently an undergraduate student at the school of information, Zhejiang University of Finance and Economics, China. His research interests include fuzzy time series, stock forecast, and machine learning.