Knowledge-Based Systems 118 (2017) 204–216

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors

Shyi-Ming Chen*, Bui Dang Ha Phuong

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 2 June 2016
Revised 29 September 2016
Accepted 26 November 2016
Available online 29 November 2016

Keywords:
Fuzzy time series forecasting
Two-factors second-order fuzzy logical relationships
Two-factors second-order fuzzy-trend logical relationship groups
Particle swarm optimization

ABSTRACT

In this paper, we propose a new fuzzy time series (FTS) forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of two-factors second-order fuzzy-trend logical relationship groups (TSFTLRGs). The proposed method uses particle swarm optimization (PSO) techniques to obtain the optimal partitions of intervals and the optimal weighting vectors simultaneously. The proposed FTS forecasting method outperforms the existing methods for forecasting the TAIEX and the NTD/USD exchange rates in terms of forecasting accuracy rates. It provides us with a useful way to deal with forecasting problems to get higher forecasting accuracy rates.

1. Introduction

The drawbacks of the traditional time series forecasting methods are that (1) they cannot deal with forecasting problems in which historical data are linguistic values and (2) their forecasting accuracy rates are not good enough. In order to overcome the drawbacks of the traditional time series forecasting methods, Song and Chissom [29–31] proposed the concepts of fuzzy time series (FTS) based on the theory fuzzy sets [41]. In recent years, some FTS forecasting methods have been presented to deal with forecasting problems [1,3–10,12,14–17,19–24,28,29,31,32,34–39].

In [8], Chen and Kao presented a method for forecasting the TAIEX based on FTS, particle swarm optimization (PSO) techniques and support vector machines. They used the PSO techniques to obtain optimal intervals in the universe of discourse and used the support vector machine to classify the training data set for forecasting the TAIEX. In [9], Chen et al. presented a method to forecast the TAIEX and the NTD/USD exchange rates based on TSFTLRGs and PSO techniques. They constructed TSFLRs based on fuzzified the historical training data of the main factor and the secondary factor, respectively, constructed TSFTLRGs based on the obtained TSFLRs, and obtained the optimal weighting vector for each TSFTLRG using PSO techniques for performing the forecasting. The motivation of this paper is to combine the advantage of using PSO techniques to obtain optimal intervals in the universe of discourse presented in [8] and the advantage of using PSO techniques to obtain optimal weighting vectors for TSFTLRGs presented in [9] to propose a new FTS forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of two-factors second-order fuzzy-trend logical relationship groups (TSFTLRGs) simultaneously for forecasting the TAIEX and the NTD/USD exchange rates to get higher forecasting accuracy rates.

In this paper, we combine the advantages of the methods presented in [8] and [9] to propose a new FTS forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of TSFTLRGs obtained by PSO techniques simultaneously for forecasting the TAIEX and the NTD/USD exchange rates to improve the forecasting accuracy rates. Firstly, the proposed method uses PSO techniques to find the optimal partitions of intervals in the universe of discourse and the optimal weighting vectors of TSFTLRGs simultaneously based on the historical training data. Then, based on the obtained optimal partitions of intervals in the universe of discourse, it fuzzifies the historical testing datum of the main factor and the secondary factor on each trading day into fuzzy sets, respectively, where each secondary factor is used to assist the forecasting of the main factor. Finally, based on the fuzzified historical testing data, it chooses the corresponding TSFTLRG to perform the forecasting using the obtained optimal weighting vector of the chosen TSFTLRG. The experimental results show that the proposed FTS forecasting method gets higher forecasting accuracy rates than the existing methods [3–9,12,16,17,35,37–39] for forecasting the TAIEX and gets higher forecasting accuracy rates than the existing methods [7,9,23,35] for

* Corresponding author. Fax: +886 227301081.
E-mail address: smchen@mail.ntust.edu.tw (S.-M. Chen).

http://dx.doi.org/10.1016/j.knosys.2016.11.019
0950-7051/

forecasting the NTD/USD exchange rates. The main contribution of this paper is that we propose a new FTS forecasting method for forecasting the TAIEX and the NTD/USD exchange rates based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of TSFTLRGs obtained by PSO techniques simultaneously. The proposed FTS forecasting method outperforms the existing methods for forecasting the TAIEX and the NTD/USD exchange rates in terms of forecasting accuracy rates.

The rest of this paper is organized as follows. In Section 2, we briefly reviewed basic concepts of fuzzy sets [41], fuzzy time series (FTS) [29–31], one-factor nth-order FTS forecasting model [4], two-factors nth-order FTS forecasting model [20], two-factors second-order fuzzy logical relationships (TSFLRs) [9] and two-factors second-order fuzzy-trend logical relationship groups (TSFTLRGs) [9]. In Section 3, we briefly review the concepts of standard PSO techniques proposed by Kennedy and Eberhart [18]. In Section 4, we propose a new FTS forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of TSFTLRGs obtained by PSO techniques simultaneously. In Section 5, we make a comparison of the experimental results of the proposed FTS forecasting method with the ones of the existing methods. The conclusions are discussed in Section 6.

2. Preliminaries

In [41], Zadeh proposed the theory of fuzzy sets. A fuzzy set A in the universe of discourse $U = \{u_1, u_2, \ldots, u_n\}$ can be represented as follows:

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + \ldots + f_A(u_n)/u_n,$$

where f_A denotes the membership function of the fuzzy set A, f_A: $U \to [0, 1]$, $f_A(u_i)$ denotes the membership degree of element u_i in the fuzzy set A, and $1 \le i \le n$.

In [29–31], Song and Chissom presented the theory of fuzzy time series (FTS), where the values of a FTS are represented by fuzzy sets. In [9], Chen et al. presented the concepts of two-factors second-order fuzzy logical relationships (TSFLRs), described as follows. Assume that $F_1(t-2)=A_i$, $F_2(t-2)=B_j$, $F_1(t-1)=A_k$, $F_2(t-1)=B_l$ and $F(t)=A_s$, where A_i, B_j, A_k, B_l and A_s are fuzzy sets, then we can construct the TSFLR "$(A_i, B_j), (A_k, B_l) \to A_s$", where "$(A_i, B_j), (A_k, B_l)$" and "$A_s$" are called the current state and the next state of the TSFLR, respectively. There are nine kinds of fuzzy-trend relationships (FTRs) in the current state of a TSFLR [9].

If the constructed TSFLRs have the same trend on the adjacent fuzzy sets in their current state, then they can be grouped into the same two-factors second-order fuzzy-trend logical relationship group (TSFLRG) [9]. For example, assume that there are the following TSFLRs:

$$\begin{aligned}
&(A_a, B_b),\ (A_c, B_d) \to A_x,\\
&(A_i, B_j),\ (A_k, B_l) \to A_y,\\
&\qquad\vdots\\
&(A_m, B_n),\ (A_s, B_p) \to A_z,
\end{aligned}$$

where $a < c$ and $b < d$; $i < k$ and $j < l$; …; $m < s$ and $n < p$. Because all of the above TSFLRs have the same FTR (i.e., the "up-and-up FTR") in their current states, we can group these TSFLRs into the "up-and-up TSFLRG".

3. Particle swarm optimization techniques

In [18], Kennedy and Eberhart proposed traditional particle swarm optimization (PSO) techniques for dealing with optimization problems, where a set of potential solutions is represented by a swarm of particles and each particle is move through the search space for search the optimal solution. When particles moving, the position of the best particle among all particles found so far is recorded and each particle keeps its personal best position which has passed previously. Each element $v_{id,j}$ in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ and each element $x_{id,j}$ in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle id are calculated as follows:

$$v_{id,j} = v_{id,j} + 2 \times rand() \times (p_{id,j} - x_{id,j}) + 2 \times Rand() \times (x_{gBest,j} - x_{id,j}), \quad (1)$$

$$x_{id,j} = x_{id,j} + v_{id,j}, \quad (2)$$

where $v_{id,j}$ is the jth element in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ of particle id, $x_{id,j}$ is the jth element in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle id, $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1], $p_{id,j}$ is the jth element in the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n}]$ of particle id, $x_{gBest,j}$ is the jth element in the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n}]$ of the best particle $gBest$ among all particles and $1 \le j \le n$.

In [27], Shi and Eberhart pointed out that the drawback of the traditional PSO [18] is that at each generation, each particle can only fly in a limited number of directions which are expected to be good areas to fly toward based on the group's experience. Therefore, Shi and Eberhart proposed PSO techniques [26,27] with a parameter called the "inertia weight" for balancing the global and local search during the optimization process by using the following equation to replace Eq. (1):

$$v_{id,j} = \omega \times v_{id,j} + c_1 \times rand() \times (p_{id,j} - x_{id,j}) + c_2 \times Rand() \times (x_{gBest,j} - x_{id,j}), \quad (3)$$

where ω is the inertia weight, $v_{id,j}$ is the jth element in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ of particle id, $p_{id,j}$ is the jth element in the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n}]$ of particle id, $x_{id,j}$ is the jth element in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle id, $x_{gBest,j}$ is the jth element in the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n}]$ of the best particle $gBest$ among all particles, c_1 and c_2 are two positive constants, where $c_1 = c_2 = 2$, $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1] and $1 \le j \le n$. In [27], Shi and Eberhart pointed out that a large inertia weight ω facilitates a global search while a small inertia weight ω tends to facilitate local space, where a linearly decreasing inertia weight ω is used which starts at 0.9 and ends at 0.4.

In [13], Eberhart and Shi presented PSO techniques to improve the performance of the PSO techniques presented in [27] by using the following equation to replace Eq. (1):

$$v_{id,j} = k \times [v_{id,j} + c_1 \times rand() \times (p_{id,j} - x_{id,j}) + c_2 \times Rand() \times (x_{gBest,j} - x_{id,j})], \quad (4)$$

where the constriction factor [13] $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$, $\varphi = c_1 + c_2$, c_1 and c_2 are two positive constants to let $\varphi > 4$, $v_{id,j}$ is the jth element in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ of particle id, $p_{id,j}$ is the jth element in the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n}]$ of particle id, $x_{id,j}$ is the jth element in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle id, $x_{gBest,j}$ is the jth element in the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n}]$ of the best particle $gBest$ among all particles, n is the dimension of a particle, $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1] and $1 \le j \le n$.

In [33], Umapathy et al. presented PSO techniques with the time-varying inertia weight ω, where

$$\omega = (\omega_{\max} - \omega_{\min}) \times \left(\frac{iter_{\max} - iter}{iter_{\max}} \right) + \ \omega_{\min}, \tag{5}$$

$\omega_{\max}$ and $\omega_{\min}$ are the maximum value and the minimum value of the time-varying inertia weight, respectively, *iter* is the current iteration number and $iter_{\max}$ is the maximum number of iterations, by using the following equation to replace Eq. (1):

$$v_{id,j} = \omega \times v_{id,j} + c_1 \times rand_1 \times \left(p_{id,j} - x_{id,j}\right) + c_2 \times rand_2 \times \left(x_{gBest,j} - x_{id,j}\right), \tag{6}$$

where the time-varying inertia weight $\omega = (\omega_{\max} - \omega_{\min}) \times \frac{(iter_{\max} - iter)}{iter_{\max}} + \omega_{\min}$, $\omega_{\max}$ and $\omega_{\min}$ are the maximum value and the minimum value of the time-varying inertia weight, respectively, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, *iter* is the current iteration number, $iter_{max}$ is the maximum number of iterations, $v_{id,j}$ is the *j*th element in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ of particle *id*, $p_{id,j}$ is the *j*th element in the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n}]$ of the particle *id*, $x_{id,j}$ is the *j*th element in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle *id*, $x_{gBest,j}$ is the *j*th element in the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n}]$ of the best particle *gBest* among all particles, c_1 and c_2 are the acceleration factors which are normally set as 2.0, respectively, $rand_1$ are $rand_2$ are random numbers uniformly distributed in the range of [0, 1] and $1 \le j \le n$.

In this paper, we propose new PSO techniques by using the following equation to replace Eq. (1):

$$v_{id,j} = k \ \times [\omega \times v_{id,j} + c_1 \times rand() \times (p_{id,j} - x_{id,j}) + c_2 \times Rand() \times (x_{gBest,j} - x_{id,j})], \tag{7}$$

where the time-varying inertia weight $\omega = (\omega_{\max} - \omega_{\min}) \times \frac{(iter_{max} - iter)}{iter_{max}} + \omega_{min}$, $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, which are the same as the ones presented in [33]; *iter* is the current iteration number; $iter_{\max}$ is the total number of iterations; the constriction factor $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$ [13], c_1 and c_2 are the self-confidence coefficient and the social confidence coefficient, respectively, $c_1 = c_2 = 2.05$ which are the same as the ones presented in [9], such that $\phi = c_1 + c_2 = 2.05 + 2.05 = 4.1$ and $k = 0.7298$; $v_{id,j}$ is the *j*th element in the velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n}]$ of particle *id*, $p_{id,j}$ is the *j*th element in the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n}]$ of particle *id*, $x_{id,j}$ is the *j*th element in the position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n}]$ of particle *id*, $x_{gBest,j}$ is the *j*th element in the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n}]$ of the best particle *gBest* among all particles and $1 \le j \le n$; *rand*() and *Rand*() random numbers uniformly distributed in the range of [0, 1].

4. A new FTS forecasting method based on optimal partitions of intervals and optimal weighting vectors

In this section, we propose a new FTS forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of FTLRGs obtained by PSO techniques simultaneously. Firstly, the proposed method uses PSO techniques to find the optimal partitions of intervals in the universe of discourse and the optimal weighting vectors of TSFTLRGs simultaneously based on the historical training data. Then, based on the obtained optimal partitions of intervals in the universe of discourse, it fuzzifies the historical testing datum of the main factor and the secondary factor on each trading day into fuzzy sets, respectively. Finally, based on the fuzzified historical testing data, it chooses the corresponding TSFTLRG to perform the forecasting using the obtained optimal weighting vector of the chosen TSFTLRG. The proposed FTS forecasting method is now presented as follows:

$x_{id,1}$	$x_{id,2}$	…	$x_{id,n-1}$	$x_{id,n}$	$y_{id,1}$	$y_{id,2}$	…	$y_{id,n-1}$	$y_{id,n}$
Elements of the Main Factor					Elements of the Secondary Factor				

Fig. 1. Position vector X_{id} of particle *id*.

$v_{id,1}$	$v_{id,2}$	…	$v_{id,n-1}$	$v_{id,n}$	$g_{id,1}$	$g_{id,2}$	…	$g_{id,n-1}$	$g_{id,n}$
Elements of the Main Factor					Elements of the Secondary Factor				

Fig. 2. Velocity vector V_{id} of particle *id*.

Step 1: Let the universe of discourse *U* of the main factor be $[D_{\min} - D_1, D_{max} + D_2]$, where D_{max} and $D_{\min}$ are the maximum value and the minimum value of the historical training data of the main factor, respectively; D_1 and D_2 are two proper positive real values to let the universe of discourse *U* cover the noise of the testing data of the main factor. It should be noted that the values of D_1 and D_2 will be different for different data sets and the values of the testing data are belonging to the interval $[D_{\min} - D_1, D_{\max} + D_2]$. Let the universe of discourse *S* of the secondary factor be $[E_{\min} - E_1, E_{\max} + E_2]$, where $E_{\max}$ and $E_{\min}$ are the maximum value and the minimum value of the historical training data of the secondary factor, respectively; E_1 and E_2 are two proper positive real values to let the universe of discourse *S* cover the noise of the testing data of the secondary factor. It should be noted that the values of E_1 and E_2 will be different for different data sets and the values of the testing data are belonging to the interval $[E_{\min} - E_1, E_{\max} + E_2]$.

Step 2: Let the system randomly generate *m* particles. Let *id* be a particle with the position represented by a position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}, x_{id,n}, y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}, y_{id,n}]$ consisting of 2*n* elements, where there are *n* elements $x_{id,1}$, $x_{id,2}, \ldots, x_{id,n-1}$ and $x_{id,n}$ for the main factor and *n* elements $y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}$, and $y_{id,n}$ for the secondary factor, $x_{id,j} \in [x_{\min}, x_{\max}]$, $x_{\min}$ and $x_{\max}$ denote the round off values of $D_{\min}$ and $D_{\max}$, respectively, and $(D_{\min} - D_1) \le x_{id,1} \le x_{id,2} \le \ldots \le x_{id,n-1} \le x_{id,n} \le (D_{\max} + D_2)$, $y_{id,j} \in [y_{\min}, y_{\max}]$, $y_{\min}$ and $y_{\max}$ denote the round off values of $E_{\min}$ and $E_{\max}$, respectively, $(E_{\min} - E_1) \le y_{id,1} \le y_{id,2} \le \ldots \le y_{id,n-1} \le y_{id,n} \le (E_{\max} + E_2)$ and $1 \le j \le n$, as shown in Fig. 1.

The velocity of each particle *id* is represented by a velocity vector $V_{id} = [v_{id,1}, v_{id,2}, \ldots, v_{id,n-1}, v_{id,n}, g_{id,1}, g_{id,2}, \ldots, g_{id,n-1}, g_{id,n}]$ consisting of 2*n* elements, where there are *n* elements $v_{id,1}, v_{id,2}, \ldots, v_{id,n-1}$ and $v_{id,n}$ for the main factor and *n* elements $g_{id,1}, g_{id,2}, \ldots, g_{id,n-1}$ and $g_{id,n}$ for the secondary factor, $v_{id,j} \in [\frac{-0.2 \times (x_{max} - x_{min})}{2}, \frac{0.2 \times (x_{max} - x_{min})}{2}]$, $g_{id,j} \in [\frac{-0.2 \times (y_{max} - y_{min})}{2}, \frac{0.2 \times (y_{max} - y_{min})}{2}]$, and $1 \le j \le n$, as shown in Fig. 2. (Note: In [2], each element in the velocity vector is limited to the range $[-v_{\max}, v_{\max}]$, where $v_{max} = \frac{r \times (x_{max} - x_{min})}{2}$ and $0.1 \le r \le 1.0$. Therefore, in this paper, we choose $r = 0.2$ based on [2], such that $v_{id,j} \in [\frac{-0.2 \times (x_{max} - x_{min})}{2}, \frac{0.2 \times (x_{max} - x_{min})}{2}]$ and $g_{id,j} \in [\frac{-0.2 \times (y_{max} - y_{min})}{2}, \frac{0.2 \times (y_{max} - y_{min})}{2}]$, where $1 \le j \le n$.).

The initial position vector X_{id} and the initial velocity vector V_{id} of each particle *id* are generated randomly by the system. The personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots,$

$p_{id,n-1}$, $p_{id,n}$, $q_{id,1}$, $q_{id,2}$,..., $q_{id,n-1}$, $q_{id,n}$] of each particle id consists of $2n$ elements, where there are n elements $p_{id,1}$, $p_{id,2}$,..., $p_{id,n-1}$ and $p_{id,n}$ for the main factor and n elements $q_{id,1}$, $q_{id,2}$,..., $q_{id,n-1}$, and $q_{id,n}$ for the secondary factor denoting the best position (i.e., the position that gives the minimum objective value found so far). Initially, let the personal best position vector $P_{id}=[p_{id,1}$, $p_{id,2}$,..., $p_{id,n-1}$, $p_{id,n}$, $q_{id,1}$, $q_{id,2}$,..., $q_{id,n-1}$, $q_{id,n}]$ of each particle id be the same as its initial position vector $X_{id}=[x_{id,1}$, $x_{id,2}$, ..., $x_{id,n-1}$, $x_{id,n}$, $y_{id,1}$, $y_{id,2}$, ..., $y_{id,n-1}$, $y_{id,n}]$. Set the number of iterations $iter$ to one.

Step 3: Calculate the objective value of each particle id. For each particle id, do the following sub-steps:

Step 3.1: Based on the position vector X_{id} = [$x_{id,1}$, $x_{id,2}$, ..., $x_{id,n-1}$, $x_{id,n}$, $y_{id,1}$, $y_{id,2}$, ..., $y_{id,n-1}$, $y_{id,n}$] of particle id, divide the universe of discourse U of the main factor into n+1 intervals u_1, u_2,..., u_n and u_{n+1}, where $u_1=[D_{\min}- D_1, x_{id,1})$, $u_2=[x_{id,1}, x_{id,2})$,..., $u_n=[x_{id,n-1}, x_{id,n})$, and $u_{n+1}=[x_{id,n}, D_{\max}+D_2]$. Based on the obtained intervals u_1, u_2,..., u_n and u_{n+1}, define the fuzzy sets A_1, A_2,..., A_n and A_{n+1} based on [29], shown as follows:

$$
\begin{aligned}
A_1 &= 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 \\
&\quad + \cdots + 0/u_n + 0/u_{n+1}, \\
A_2 &= 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 \\
&\quad + \cdots + 0/u_n + 0/u_{n+1}, \\
&\ \vdots \\
A_n &= 0/u_1 + 0/u_2 + 0/u_3 + \cdots + 0.5/u_{n-1} \\
&\quad + 1/u_n + 0.5/u_{n+1}, \\
A_{n+1} &= 0/u_1 + 0/u_2 + 0/u_3 \\
&\quad + 0/u_4 + \cdots + 0.5/u_n + 1/u_{n+1}.
\end{aligned}
$$

Based on the position vector $X_{id}=[x_{id,1}$, $x_{id,2}$, ..., $x_{id,n-1}$, $x_{id,n}$, $y_{id,1}$, $y_{id,2}$, ..., $y_{id,n-1}$, $y_{id,n}]$ of particle id, divide the universe of discourse S of the secondary factor into n+1 intervals b_1, b_2,..., b_n and b_{n+1}, where $b_1=[E_{\min}-E_1, y_{id,1})$, $b_2=[y_{id,1}, y_{id,2})$,..., $b_n=[y_{id,n-1}, y_{id,n})$, and $b_{n+1}=[y_{id,n}, E_{\max}+E_2]$. Based on the obtained intervals b_1, b_2,..., b_n and b_{n+1}, define the fuzzy sets B_1, B_2,..., B_n and B_{n+1} based on [29], shown as follows:

$$
\begin{aligned}
B_1 &= 1/b_1 + 0.5/b_2 + 0/b_3 + 0/b_4 \\
&\quad + \cdots + 0/b_n + 0/b_{n+1}, \\
B_2 &= 0.5/b_1 + 1/b_2 + 0.5/b_3 + 0/b_4 \\
&\quad + \cdots + 0/b_n + 0/b_{n+1}, \\
&\ \vdots \\
B_n &= 0/b_1 + 0/b_2 + 0/b_3 \\
&\quad + \cdots + 0.5/b_{n-1} + 1/b_n + 0.5/b_{n+1}, \\
B_{n+1} &= 0/b_1 + 0/b_2 + 0/b_3 + 0/b_4 \\
&\quad + \cdots + 0.5/b_n + 1/b_{n+1}.
\end{aligned}
$$

Step 3.2: Based on [4], fuzzify the historical training data of the main factor and the secondary factor into fuzzy sets defined in **Step 3.1**, respectively. For example, if a historical training datum of the main factor on a trading day belongs to interval u_l, then it is fuzzified into fuzzy set A_l, where $1 \le l \le n + 1$. If the historical training datum of the secondary factor on a trading day belongs to interval b_l, then it is fuzzified into fuzzy set B_l, where $1 \le l \le n + 1$.

Table 1
Nine TSFTLRGs for the TSFLR "(A_{x2}, B_{y2}), (A_{x1}, B_{y1})→A_z" [9].

Group number	FTRs	
	between x2 and x1	between y2 and y1
Group 1	up	up
Group 2	up	equal
Group 3	up	down
Group 4	equal	up
Group 5	equal	equal
Group 6	equal	down
Group 7	down	up
Group 8	down	equal
Group 9	down	down

Step 3.3: Based on [20], construct the TSFLRs based on the fuzzified historical training data of the main factor and the fuzzified historical training data of the secondary factor obtained in **Step 3.2.** For example, if the fuzzified historical training data of the main factor on trading days $t-$ 2, $t-$ 1 and t are fuzzy sets A_{x2}, A_{x1} and A_z, respectively, and if the fuzzified historical training data of the secondary factor on trading days $t-$ 2 and $t-$ 1 are fuzzy sets B_{y2} and B_{y1}, respectively, then construct the TSFLR "(A_{x2}, B_{y2}), (A_{x1}, B_{y1})→A_z".

Step 3.4: **Step 3.4:** Based on [9], construct the TSFTLRGs based on the constructed TSFLRs obtained in **Step 3.3**. For example, for the TSFLR "(A_{x2}, B_{y2}), (A_{x1}, B_{y1})→A_z", we can get nine TSFTLRGs, as shown in Table 1 [9].

Step 3.5: Call the proposed **"PSO-Based Optimal-Weights Learning Algorithm"** described later to get the optimal weighting vector $W_{id,i}=[w_{i1}, w_{i2}]$ for each TSFTLRG "*Group i*" based on particle id, where $w_{ij} \in [0, 1]$, $w_{i1}+w_{i2} = 1$, $1 \le i \le 9$ and $1 \le j \le 2$.

Step 3.6: Calculate the forecasted value $F_{id,t}$ of each historical training datum on trading day t based on the obtained optimal weighting vectors of the TSFTLRGs obtained in **Step 3.5**, described as follows. Assume that we want to forecast the value $F_{id,t}$ of the historical training datum on trading day t and assume that the fuzzified historical training data of the main factor TAIEX on trading days $t-2$ and $t-1$ are fuzzy sets A_{x2} and A_{x1}, respectively, and assume that the fuzzified historical training data of the secondary factor on trading days $t-2$ and $t-1$ are fuzzy sets B_{y2} and B_{y1}, respectively. Based on the trend between the subscripts of the fuzzy sets A_{x2} and A_{x1} and the trend between the subscripts of the fuzzy sets B_{y2} and B_{y1}, choose the corresponding TSFTLRG ***"Group i"*** to perform the forecasting, where $1 \le i \le 9$. For example, if $x2 < x1$ and $y2 < y1$, then it belongs to the "up-and-up FTR" and from Table 1, we can see that its corresponding TSFTLRG is ***Group* 1** (i.e., the up-and-up TSFTLRG). Therefore, the system chooses the TSFTLRG ***Group*** to perform the forecasting. Assume that R_{t-2} and R_{t-1} are the actual values of the historical training data on the trading days $t-2$ and $t-1$, respectively. Assume that the system chooses TSFTLRG ***"Group i"*** to perform the forecasting, where the optimal weighting vector $W_{id,i}$ of the TSFTLRG "*Group i*" based on particle id is $W_{id,i}=[w_{i1}, w_{i2}]$, i.e., the weights of the historical training data on the trading days $t-2$ and $t-1$ are w_{i1} and w_{i2}, respectively, where $w_{ij} \in [0, 1]$, $w_{i1}+w_{i2} = 1$, $1 \le i \le 9$ and $1 \le j \le 2$. Calculate the forecasted value $F_{id,t}$ of the historical training datum on trading day t based on particle id, shown as follows [9]:

$$F_{id,t} = R_{t-2} \times w_{i1} + R_{t-1} \times w_{i2}, \tag{8}$$

where $w_{ij} \in [0, 1]$, $w_{i1} + w_{i2} = 1$, $1 \le i \le 9$ and $1 \le j \le 2$.

Step 3.7: Calculate the objective value (i.e., the root mean square error) RMSE_{id} of particle *id*, shown as follows:

$$\text{RMSE}_{id} = \sqrt{\frac{\sum_{t=3}^{t_{max}} \left(F_{id,t} - R_t\right)^2}{t_{max} - 2}}, \tag{9}$$

where $t_{\max}$ denotes the number of trading days of the historical training data, $F_{id,t}$ denotes the forecasted value of the historical training datum on trading day t based on particle *id* obtained in **Step 3.6**, and R_t denotes the actual value of the historical training datum on trading day t, where $3 \le t \le t_{\max}$.

Step 4: Update the personal best position vector $P_{id} = [p_{id,1}, p_{id,2}, \ldots, p_{id,n-1}, p_{id,n}, q_{id,1}, q_{id,2}, \ldots, q_{id,n-1}, q_{id,n}]$ of each particle *id* with its position vector $X_{id} = [x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}, x_{id,n}, y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}, y_{id,n}]$ if its objective value at the current iteration *iter* is smaller than its objective value at the previous iteration $iter - 1$, by assigning the values of each element in the position vector X_{id} at the current iteration to the elements in the personal best position vector P_{id}, where $p_{id,1} = x_{id,1}$, $p_{id,2} = x_{id,2}, \ldots, p_{id,n-1} = x_{id,n-1}$, $p_{id,n} = x_{id,n}$, $q_{id,1} = y_{id,1}$, $q_{id,2} = y_{id,2}, \ldots, q_{id,n-1} = y_{id,n-1}$ and $q_{id,n} = y_{id,n}$.

Step 5: Find the best particle *gBest* which has the minimum objective value among m particles. Let $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n-1}, x_{gBest,n}, y_{gBest,1}, y_{gBest,2}, \ldots, y_{gBest,n-1}, y_{gBest,n}]$ denote the position vector of the best particle *gBest*. Let $L_{gBest,i} = [w_{gBest,i1}, w_{gBest,i2}]$ be the optimal weighting vector of "*Group i*" of the TSFTLRGs of the best particle *gBest*, where $1 \le i \le 9$.

Step 6: Perform the following sub-steps to update the elements $v_{id,1}, v_{id,2}, \ldots, v_{id,n-1}$ and $v_{id,n}$ in the velocity vector V_{id} and update the elements $x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}$ and $x_{id,n}$ in the position vector X_{id} of each particle *id*, respectively, and to update the elements $g_{id,1}, g_{id,2}, \ldots, g_{id,n-1}$ and $g_{id,n}$ in the velocity vector V_{id} and update the elements $y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}$ and $y_{id,n}$ in the position vector X_{id} of each particle *id*, respectively.

Step 6.1: Based on Eqs. (2) and (7), update the elements $v_{id,1}, v_{id,2}, \ldots, v_{id,n-1}$ and $v_{id,n}$ in the velocity vector V_{id} and update the elements $x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}$ and $x_{id,n}$ in the position vector X_{id} of each particle *id*, respectively, shown as follows:

$$v_{id,j} = k \times [\omega \times v_{id,j} + c_1 \times rand() \times (p_{id,j} - x_{id,j}) + c_2 \times Rand() \times (x_{gBest,j} - x_{id,j})],$$
$$x_{id,j} = x_{id,j} + v_{id,j},$$

where $p_{id,j}$ is an element in the personal best position vector P_{id} of particle *id*, $x_{id,j}$ is an element in the current position vector X_{id} of particle *id*, $x_{gBest,j}$ is an element in the position vector P_{gBest} of the best particle *gBest* found in this iteration and $1 \le j \le n$; $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$, $\varphi = c_1 + c_2$, c_1 and c_2 are the self-confidence coefficient and the social confident coefficient, respectively, and $c_1 = c_2 = 2.05$ which are the same as the ones presented in [9], such that $\phi = 4.1$ and $k = 0.7298$; $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1]; the time-varying inertia weight $\omega = (\omega_{max} - \omega_{min}) \times \frac{(iter_{max} - iter)}{iter_{max}} + \omega_{min}$, $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, which are the same as the ones presented in [33]; *iter* is the current iteration number; $iter_{\max}$ is the total number of iterations. If $v_{id,j} > \frac{0.2 \times (x_{max} - x_{min})}{2}$, then let $v_{id,j} = \frac{0.2 \times (x_{max} - x_{min})}{2}$; if $v_{id,j} < \frac{-0.2 \times (x_{max} - x_{min})}{2}$, then let $v_{id,j} = \frac{-0.2 \times (x_{max} - x_{min})}{2}$, where $1 \le j \le n$. If $x_{id,j} > x_{\max}$ or $x_{id,j} < x_{\min}$, then $x_{id,j}$ is modified as follows [11]:

$$x_{id,j} = \begin{cases} x_{max} - \left(\frac{1}{2} \times rand() \times (x_{max} - x_{min})\right), & if\ x_{id,j} > x_{max} \\ x_{min} + \left(\frac{1}{2} \times rand() \times (x_{max} - x_{min})\right), & if\ x_{id,j} < x_{min} \end{cases} \tag{10}$$

where *rand*() is a random number uniformly distributed in the range of [0, 1] and $1 \le j \le n$. Perform the round off operations to the elements $x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}$ and $x_{id,n}$ in the position vector X_{id}, respectively. Sort the values of the elements $x_{id,1}, x_{id,2}, \ldots, x_{id,n-1}$ and $x_{id,n}$ in the position vector X_{id} of particle *id* in an ascending sequence, such that $x_{id,1} < x_{id,2} < \ldots < x_{id,n-1} < x_{id,n}$.

Step 6.2: Based on Eqs. (2) and (7), update the elements $g_{id,1}, g_{id,2}, \ldots, g_{id,n-1}$ and $g_{id,n}$ in the velocity vector V_{id} and update the elements $y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}$ and $y_{id,n}$ in the position vector X_{id} of each particle *id*, respectively, shown as follows:

$$g_{id,j} = k \times [\omega \times g_{id,j} + c_1 \times rand() \times (g_{id,j} - y_{id,j}) + c_2 \times Rand() \times (y_{gBest,j} - y_{id,j})],$$
$$y_{id,j} = y_{id,j} + g_{id,j},$$

where $g_{id,j}$ is an element in the personal best position vector P_{id} of particle *id*, $y_{id,j}$ is an element in the current position vector X_{id} of particle *id*, $y_{gBest,j}$ is an element in the position vector P_{gBest} of the best particle *gBest* found in this iteration and $1 \le j \le n$; $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$, $\phi = c_1 + c_2$ and $c_1 = c_2 = 2.05$, such that $\phi = 4.1$ and $k = 0.7298$; $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1]; the time-varying inertia weight $\omega = (\omega_{max} - \omega_{min}) \times \frac{(iter_{max} - iter)}{iter_{max}} + \omega_{min}$, $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, which are the same as the ones presented in [33]. If $g_{id,j} > \frac{0.2 \times (y_{max} - y_{min})}{2}$, then let $g_{id,j} = \frac{0.2 \times (y_{max} - y_{min})}{2}$; if $g_{id,j} < \frac{-0.2 \times (y_{max} - y_{min})}{2}$, then let $g_{id,j} = \frac{-0.2 \times (y_{max} - y_{min})}{2}$, where $1 \le j \le n$. If $y_{id,j} > y_{\max}$ or $y_{id,j} < y_{\min}$, then $y_{id,j}$ is modified as follows [11]:

$$y_{id,j} = \begin{cases} y_{max} - \left(\frac{1}{2} \times rand() \times (y_{max} - y_{min})\right), & if\ y_{id,j} > y_{max} \\ y_{min} + \left(\frac{1}{2} \times rand() \times (y_{max} - y_{min})\right), & if\ y_{id,j} < y_{min} \end{cases} \tag{11}$$

where *rand*() is a random number uniformly distributed in the range of [0, 1] and $1 \le j \le n$. Perform the round off operations to the elements $y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}$ and $y_{id,n}$ in the position vector X_{id}, respectively. Sort the values of the elements $y_{id,1}, y_{id,2}, \ldots, y_{id,n-1}$ and $y_{id,n}$ in the position vector X_{id} of particle *id* in an ascending sequence, such that $y_{id,1} < y_{id,2} < \ldots < y_{id,n-1} < y_{id,n}$.

Step 7: If the number of iterations *iter* is smaller than a predefined number of iterations $iter_{\max}$, then let $iter = iter + 1$ and go to **Step 3**. Otherwise, let the position vector $P_{gBest} = [x_{gBest,1}, x_{gBest,2}, \ldots, x_{gBest,n-1}, x_{gBest,n}, y_{gBest,1}, y_{gBest,2}, \ldots, y_{gBest,n-1}, y_{gBest,n}]$ of the best particle *gBest* obtained in **Step 5** be the optimal partition vector $D=[x_1, x_2, \ldots, x_{n-1}, x_n, y_1, y_2, \ldots, y_{n-1}, y_n]$, where $x_1=x_{gBest,1}$, $x_2=x_{gBest,2}, \ldots, x_{n-1}=x_{gBest,n-1}$ and $x_n=x_{gBest,n}$, $y_1=y_{gBest,1}$, $y_2=y_{gBest,2}, \ldots, y_{n-1}=y_{gBest,n-1}$ and $y_n=y_{gBest,n}$. In this case, the optimal partition of the intervals in the universe of discourse U of the main factor are $u_1=[D_{\min}-D_1, x_1)$, $u_2=[x_1, x_2), \ldots, u_n=[x_{n-1}, x_n)$, and $u_{n+1}=[x_n, D_{\max}+D_2]$; the optimal partition of the intervals in the universe of discourse S of the secondary factor are $b_1=[E_{\min}-E_1, y_1)$, $b_2=[y_1, y_2), \ldots, b_n=[y_{n-1}, y_n)$, and $b_{n+1}=[y_n, E_{\max}+E_2]$. Let the optimal weighting vector $L_{gBest,i}=[w_{gBest,i1}, w_{gBest,i2}]$ of "*Group i*" of the best particle *gBest* obtained in **Step 5** be the optimal weighting vector $W_i=[w_{i1}, w_{i2}]$ of "*Group i*" of the TSFTLRGs, where $w_{i1}=w_{gBest,i1}$ and $w_{i2}=w_{gBest,i2}$, $w_{ij}\in[0, 1]$, $w_{i1}+w_{i2}=1$, $1\le i\le 9$ and $1\le j\le 2$.

Step 8: Based on the optimal partition vector $D=[x_1, x_2, \ldots, x_{n-1}, x_n, y_1, y_2, \ldots, y_{n-1}, y_n]$ obtained in **Step 7**, divide the universe of discourse U of the main factor into $n+1$ intervals $u_1, u_2, \ldots, u_n$ and u_{n+1}, where $u_1=[D_{\min}-D_1, x_1)$, $u_2=[x_1, x_2), \ldots, u_n=[x_{n-1}, x_n)$, and $u_{n+1}=[x_n, D_{\max}+D_2]$; divide the universe of discourse S of the secondary factor into $n+1$ intervals $b_1, b_2, \ldots, b_n$ and b_{n+1}, where $b_1=[E_{\min}-E_1, y_1)$, $b_2=[y_1, y_2), \ldots, b_n=[y_{n-1}, y_n)$, and $b_{n+1}=[y_n, E_{\max}+E_2]$. Based on the obtained intervals $u_1, u_2, \ldots, u_n$ and u_{n+1}, define the fuzzy sets $A_1, A_2, \ldots, A_n$ and A_{n+1} based on [29], shown as follows [29]:

$$
\begin{aligned}
A_1 &= 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 + \cdots + 0/u_n + 0/u_{n+1},\\
A_2 &= 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 + \cdots + 0/u_n + 0/u_{n+1},\\
&\vdots\\
A_n &= 0/u_1 + 0/u_2 + 0/u_3 + \cdots + 0.5/u_{n-1} + 1/u_n + 0.5/u_{n+1},\\
A_{n+1} &= 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + \cdots + 0.5/u_n + 1/u_{n+1}.
\end{aligned}
$$

Based on the obtained intervals $b_1, b_2, \ldots, b_n$ and b_{n+1}, define the fuzzy sets $B_1, B_2, \ldots, B_n$ and B_{n+1} based on [29], shown as follows:

$$
\begin{aligned}
B_1 &= 1/b_1 + 0.5/b_2 + 0/b_3 + 0/b_4 + \cdots + 0/b_n + 0/b_{n+1},\\
B_2 &= 0.5/b_1 + 1/b_2 + 0.5/b_3 + 0/b_4 + \cdots + 0/b_n + 0/b_{n+1},\\
&\vdots\\
B_n &= 0/b_1 + 0/b_2 + 0/b_3 + \cdots + 0.5/b_{n-1} + 1/b_n + 0.5/b_{n+1},\\
B_{n+1} &= 0/b_1 + 0/b_2 + 0/b_3 + 0/b_4 + \cdots + 0.5/b_n + 1/b_{n+1}.
\end{aligned}
$$

Step 9: Based on [4], fuzzify the historical testing data of the main factor and the secondary factor into fuzzy sets defined in **Step 8**, respectively. For example, if a historical testing datum of the main factor on a trading day belongs to interval u_l, then it is fuzzified into fuzzy set A_l, where $1\le l\le n+1$. If the historical testing datum of the secondary factor on a trading day belongs to interval b_l, then it is fuzzified into fuzzy set B_l, where $1\le l\le n+1$.

Step 10: Calculate the forecasted value F_t of the historical testing datum on trading day t based on the obtained optimal weighting vectors of the TSFTLRGs obtained in **Step 7**, described as follows. Assume that we want to forecast the value F_t of the historical testing datum on trading day t and assume that the fuzzified historical testing data of the main factor TAIEX on trading days $t-2$ and $t-1$ are fuzzy sets A_{x2} and A_{x1}, respectively, and assume that the fuzzified historical testing data of the secondary factor on trading days $t-2$ and $t-1$ are fuzzy sets B_{y2} and B_{y1}, respectively. Based on the trend between the subscripts of the fuzzy sets A_{x2} and A_{x1} and the trend between the subscripts of the fuzzy sets B_{y2} and B_{y1}, choose the corresponding TSFTLRG ***"Group i"*** to perform the forecasting, where $1\le i\le 9$. For example, if $x2 < x1$ and $y2 < y1$, then it belongs to the "up-and-up FTR" and from Table 1, we can see that its corresponding TSFTLRG is ***Group* 1**. Therefore, the system chooses the TSFTLRG ***Group* 1** (i.e., the up-and-up TSFTLRG) to perform the forecasting. Assume that R_{t-2} and R_{t-1} are the actual values of fuzzy sets A_{x2} and A_{x1} of the historical testing data on the trading days $t-2$ and $t-1$, respectively. Assume that the system chooses TSFTLRG ***"Group i"*** to perform the forecasting, where the optimal weighting vector W_i of the TSFTLRG "*Group i*" is $W_i=[w_{i1}, w_{i2}]$, i.e., the weights of the historical testing data on the trading days $t-2$ and $t-1$ are w_{i1} and w_{i2}, respectively, where $w_{ij}\in[0, 1]$, $w_{i1}+w_{i2}=1$, $1\le i\le 9$ and $1\le j\le 2$. Calculate the forecasted value F_t of the historical testing datum on trading day t, shown as follows [9]:

$$F_t = R_{t-2} \times w_{i1} + R_{t-1} \times w_{i2}, \tag{12}$$

where $w_{ij}\in[0, 1]$, $w_{i1}+w_{i2}=1$, $1\le i\le 9$ and $1\le j\le 2$.

In the following, we present the **"PSO-Based Optimal-Weights Learning Algorithm"** used in **Step 3.5** of the proposed FTS forecasting method to get the optimal weighting vector $W_i=[w_{i1}, w_{i2}]$ for each TSFTLRG "*Group i*", where $w_{ij}\in[0, 1]$, $w_{i1}+w_{i2}=1$, $1\le i\le 9$ and $1\le j\le 2$. The "**PSO-Based Optimal-Weights Learning Algorithm**" is a modification of the method presented in [9]. The proposed "**PSO-Based Optimal-Weights Learning Algorithm**" is now presented as follows:

PSO-Based Optimal-Weights Learning Algorithm:

Step 1: Based on [9], construct the current-state actual data matrix C_i and the next-state actual data vector N_i from the TSFLRs appearing in the ith TSFTLRG "*Group i*", where $1\le i\le 9$, described as follows. Assume that there are the following TSFLRs shown in the ith TSFTLRG "*Group i*", where $1\le i\le 9$:

$$
\begin{aligned}
&(A_{X11}, B_{Y11}),\ (A_{X12}, B_{Y12}) \rightarrow A_{Q1},\\
&(A_{X21}, B_{Y21}),\ (A_{X22}, B_{Y22}) \rightarrow A_{Q2},\\
&\vdots\\
&(A_{Xf1}, B_{Yf1}),\ (A_{Xf2}, B_{Yf2}) \rightarrow A_{Qf}.
\end{aligned}
$$

Then, the constructed current-state actual data matrix C_i and the constructed next-state actual data vector N_i of the ith TSFTLRG "*Group i*" are shown as follows:

$$C_i = \begin{bmatrix} R_{t_1-2} & R_{t_1-1} \\ R_{t_2-2} & R_{t_2-1} \\ \vdots & \vdots \\ R_{t_f-2} & R_{t_f-1} \end{bmatrix} \text{ and } N_i = \begin{bmatrix} R_{t_1} \\ R_{t_2} \\ \vdots \\ R_{t_f} \end{bmatrix},$$

where R_{t_1-2}, R_{t_1-1} and R_{t_1} are the actual values of A_{X11}, A_{X12} and A_{Q1} of the historical training data on the trading days t_1-2, t_1-1 and t_1, respectively; R_{t_2-2}, R_{t_2-1} and

$z_{id^*,1}$	$z_{id^*,2}$

Fig. 3. Position vector Z_{id^*} of particle id^*.

$h_{id^*,1}$	$h_{id^*,2}$

Fig. 4. Velocity vector H_{id^*} of particle id^*.

R_{t_2} are the actual values of A_{X21}, A_{X22} and A_{Q2} of the historical training data on the trading days t_2-2, t_2-1 and t_2, respectively; ...; R_{t_f-2}, R_{t_f-1} and R_{t_f} are the actual values of A_{Xf1}, A_{Xf2} and A_{Qf} of the historical training data on the trading days t_f-2, t_f-1 and t_f, respectively.

Step 2: Generate p particles with dimension 2. The position of each particle id^* is denoted by a position vector $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$ consisting of two elements $z_{id^*,1}$ and $z_{id^*,2}$, where $z_{id^*,j} \in [0, 1]$, $1 \le j \le 2$ and $z_{id^*,1} + z_{id^*,2} = 1$, as shown in Fig. 3. The velocity of each particle id^* is denoted by a velocity vector $H_{id^*} = [h_{id^*,1}, h_{id^*,2}]$ consisting of two elements $h_{id^*,1}$ and $h_{id^*,2}$, where $h_{id^*,j} \in [-0.1, 0.1]$ and $1 \le j \le 2$, as shown in Fig. 4. The system generates randomly the initial position vector $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$ and the initial velocity vector $H_{id^*} = [h_{id^*,1}, h_{id^*,2}]$ of each particle id^*. The personal best position vector $L_{id^*} = [l_{id^*,1}, l_{id^*,2}]$ of each particle id^* consists of two elements $l_{id^*,1}$ and $l_{id^*,2}$ denoting the personal best position (i.e., the position that gives the minimum objective value found so far). Initially, let the personal best position vector $L_{id^*} = [l_{id^*,1}, l_{id^*,2}]$ of each particle id^* be the same as its initial position vector $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$. Set the number of iterations $iter^*$ to one.

Step 3: For each particle id^*, do the following sub-steps:

Step 3.1: Based on [9], construct the error vector $Err_{id^*} = [err_{id^*,1}, err_{id^*,2}, \ldots, err_{id^*,f}]$ of particle id^*, shown as follows:

$$
\begin{aligned}
Err_{id^*} &= |C_i \times Z_{id^*} - N_i| \\
&= \left| \begin{bmatrix} R_{t_1-2} & R_{t_1-1} \\ R_{t_2-2} & R_{t_2-1} \\ \vdots & \vdots \\ R_{t_f-2} & R_{t_f-1} \end{bmatrix} \times \begin{bmatrix} z_{id^*,1} \\ z_{id^*,2} \end{bmatrix} - \begin{bmatrix} R_{t_1} \\ R_{t_2} \\ \vdots \\ R_{t_f} \end{bmatrix} \right| \\
&= \begin{bmatrix} err_{id^*,1} \\ err_{id^*,2} \\ \vdots \\ err_{id^*,f} \end{bmatrix},
\end{aligned}
\tag{13}
$$

where C_i is the current-state actual data matrix, N_i is the next-state actual data vector and $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$ is the position vector of particle id^*.

Step 3.2: Calculate the objective value $avg\_error_{id^*}$ of particle id^*, shown as follows:

$$avg\_error_{id^*} = \frac{err_{id^*,1} + err_{id^*,2} + \ldots + err_{id^*,f}}{f}, \tag{14}$$

where $err_{id^*,1}$, $err_{id^*,2}$,..., and $err_{id^*,f}$ are the elements of the error vector Err_{id^*}.

Step 4: For each particle id^*, update the personal best position vector $L_{id^*} = [l_{id^*,1}, l_{id^*,2}]$ of particle id^* with its position vector $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$ if its objective value at the current iteration $iter^*$ is smaller than its objective value at the previous iteration $iter^* - 1$, by assigning the values of each element in the position vector Z_{id^*} at the current iteration to the elements in the personal best position vector L_{id^*}, where $l_{id^*,1} = z_{id^*,1}$ and $l_{id^*,2} = z_{id^*,2}$.

Step 5: Find the best particle $gBest^*$ which has the minimum objective value among p particles. Let $L_{gBest^*} = [z_{gBest^*,1}, z_{gBest^*,2}]$ be the position vector of the best particle $gBest^*$.

Step 6: Based Eqs. (2) and (7), update each element $h_{id^*,j}$ in the velocity vector $H_{id^*} = [h_{id^*,1}, h_{id^*,2}]$ and update each element $z_{id^*,j}$ in the position vector $Z_{id^*} = [z_{id^*,1}, z_{id^*,2}]$ of each particle id^*, respectively, shown as follows:

$$
\begin{aligned}
h_{id^*,j} &= k \times [\omega \times h_{id^*,j} + c_1 \times rand() \times (l_{id^*,j} - z_{id^*,j}) \\
&\quad + c_2 \times Rand() \times (z_{gBest^*,j} - z_{id^*,j})], \\
z_{id^*,j} &= z_{id^*,j} + h_{id^*,j},
\end{aligned}
$$

where $l_{id^*,j}$ is the jth element in the personal best position vector L_{id^*} of particle id^*, $z_{id^*,j}$ is the jth element in the current position vector Z_{id^*} of particle id^*, $z_{gBest^*,j}$ is the jth element in the position vector L_{gBest^*} of the best particle $gBest^*$ found in this iteration, $1 \le j \le 2$; $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$, $\phi = c_1 + c_2$ and $c_1 = c_2 = 2.05$, such that $\phi = 4.1$ and $k = 0.7298$; $rand()$ and $Rand()$ are random numbers uniformly distributed in the range of [0, 1]; the time-varying inertia weight $\omega = (\omega_{max} - \omega_{min}) \times \frac{(iter^*_{max} - iter^*)}{iter^*_{max}} + \omega_{min}$, $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$ [33]; $iter^*$ is the current iteration number; $iter^*_{max}$ is the total number of iterations. If $h_{id^*,j} > 0.1$, then let $h_{id^*,j} = 0.1$; if $h_{id^*,j} < -0.1$, then let $h_{id^*,j} = -0.1$. If $z_{id^*,j} > z_{\max}$ or $z_{id^*,j} < z_{\min}$, where $z_{\min} = 0$ and $z_{\max} = 1$, then $z_{id^*,j}$ is modified as follows [11]:

$$
z_{id^*,j} = \begin{cases} z_{max} - \left(\frac{1}{2} \times rand(\,) \times (z_{max} - z_{min})\right), & if\ z_{id^*,j} > z_{max} \\ z_{min} + \left(\frac{1}{2} \times rand() \times (z_{max} - z_{min})\right), & if\ z_{id^*,j} < z_{min} \end{cases}
\tag{15}
$$

where $rand()$ is a random number uniformly distributed in the range of [0, 1] and $1 \le j \le 2$. Let

$$z_{id^*,1} = \frac{z_{id^*,1}}{z_{id^*,1} + z_{id^*,2}}, \tag{16}$$

$$z_{id^*,2} = \frac{z_{id^*,2}}{z_{id^*,1} + z_{id^*,2}}, \tag{17}$$

where $z_{id^*,j} \in [0, 1]$, $1 \le j \le 2$ and $z_{id^*,1} + z_{id^*,2} = 1$.

Step 7: If the number of iterations $iter^*$ is smaller than a predefined number of iterations $iter^*_{max}$, then let $iter^* = iter^* + 1$ and go to **Step 3**. Otherwise, let the position vector $L_{gBest^*} = [z_{gBest^*,1}, z_{gBest^*,2}]$ of the best particle $gBest^*$ obtained in **Step 5** be the optimal weighting vector $W_i = [w_{i1}, w_{i2}]$ of the ith TSFTLRG "*Group i*", where $w_{i1} = z_{gBest^*,1}$, $w_{i2} = z_{gBest^*,2}$ and $1 \le i \le 9$.

In the following, we describe the main differences between the proposed "**PSO-Based Optimal-Weights Learning Algorithm**" and the method presented in [9], shown as follows:

1) The method presented in [9] divides the universe of discourse of the main factor U and the universe of discourse of the secondary factor V into equal-length of intervals, whereas the proposed "**PSO-Based Optimal-Weights Learning Algorithm**" divides the universe of discourse of the main factor U and the universe of discourse of the secondary factor V into different-length of intervals.
2) The method presented in [9] only uses one PSO to find the optimal weighting vector $W_g = [w_{g,1}, w_{g,2}]$ for each TSFTLRG "Group g", where $1 \le g \le 9$, whereas the proposed "**PSO-Based**

Optimal-Weights Learning Algorithm" uses two PSOs to obtain the optimal partitions of intervals and the optimal weighting vectors, where it uses the outer PSO to find the optimal partitions of intervals of the main factor and the optimal partitions of intervals the secondary factor and uses the inner PSO to get the optimal weighting vector $W_{id,i} = [w_{i1}, w_{i2}]$ for each TSFTLRG "*Group i*" based on particle *id*, where $w_{ij} \in [0, 1]$, $w_{i1} + w_{i2} = 1$, $1 \le i \le 9$ and $1 \le j \le 2$.

In summary, the flowchart of the proposed FTS forecasting method and the flowchart of the proposed "**PSO-Based Optimal-Weights Learning Algorithm**" are shown in Figs. 5 and 6, respectively.

5. Experimental results

We have implemented the proposed FTS forecasting method using MATLAB Version R2012a on an Intel core i7 PC to forecast the TAIEX [42] from 1990 to 2004 and to forecast the NTD/USD exchange rates [43] from March 1, 2006, to March 1, 2007, respectively. In our experiments, the proposed FTS forecasting method is executed 100 runs and the best result of all runs is taken to be the final result. The parameters of the PSO techniques used in the proposed FTS forecasting method for obtaining optimal forecasting results are shown as follows: The number of iterations is 100, the number of particles is 10, the time-varying inertia weight ω is decreased from 0.9 to 0.4, the self-confidence coefficient c_1 and the social confidence coefficient c_2 are 2.05 and 2.05, respectively, i.e., $c_1 = c_2 = 2.05$, such that the constriction factor $k = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|} = 0.7298$, where $\varphi = c_1 + c_2 = 2.05 + 2.05 = 4.1$.

5.1. Forecasting the TAIEX

In order to compare the experimental results of the proposed method with the ones of the existing methods for forecasting the TAIEX, we also let the TAIEX be the main factor and let the Dow Jones [45], the NASDAQ [46] and the M1b [44] be the secondary factors, respectively, where the secondary factors are used to assist the forecasting of the main factor. After a number of experiments, the best value found of the number of intervals in the universes of discourse of the main factor and each secondary factor is "13". In order to compare the experimental results of the proposed method with the ones of the methods presented in [3–7,9,12,16,17,35,37–39], we also adopt the ten-month/two-month split for training/testing, i.e., the historical data of the main factor and each secondary factor of each year are divided into two parts, where the data from January to October are used as the training data set and the data from November to December are used as the testing data set. In order to compare the experimental results of the proposed method with the ones of the existing methods [3–9,12,16,17,35,37–39] for forecasting the TAIEX, we also evaluate the performance of the proposed method using the root mean square error (RMSE), shown as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=3}^{t_{max}} (F_t - R_t)^2}{t_{max} - 2}}, \tag{18}$$

where $t_{\max}$ denotes the number of trading days of the historical testing data, F_t denotes the forecasted value of the historical testing datum of the TAIEX on trading day t, and R_t denotes the actual value of the historical testing datum of the TAIEX on trading day t, where $3 \le t \le t_{\max}$. Table 2 shows a comparison of RMSEs and the average RMSEs for different methods for forecasting the TAIEX from 1999 to 2004. Table 3 shows a comparison of RMSEs and the average RMSEs for different methods for forecasting the TAIEX from 1990 to 1999. From Tables 2 and 3, we can see that the proposed method gets smaller RMSEs than the existing methods [3–9,12,16,17,35,37–39] for forecasting the TAIEX. In other words, the proposed method gets higher forecasting accuracy rates than the methods presented in [3–9,12,16,17,35,37,38] and [39] for forecasting the TAIEX.

5.2. Forecasting the NTD/USD exchange rates

Leu et al. [23] proposed the two-factors second-order m-day forecasting model for forecasting the NTD/USD exchange rates [43], where $m \ge 1$. A fuzzy logical relationship for a two-factors second-order i-day forecasting on trading day t is represented as follows:

$$(X_{t-2i}, Y_{t-2i}), (X_{t-i}, Y_{t-i}) \rightarrow X_t,$$

where X_{t-2i}, X_{t-i} and X_t are fuzzified historical training data of the main factor represented by fuzzy sets on trading days $t-2i$, $t-i$ and t, respectively; Y_{t-2i} and Y_{t-i} are fuzzified historical training data of the secondary factor represented by fuzzy sets on trading days $t-2i$ and $t-i$, respectively. In our experiments for forecasting the NTD/USD exchange rates, we let the NTD/USD exchange rates be the main factor and we let the CNY/USD exchange rates, the JPY/USD exchange rates, the KRW/USD exchange rates and the TAIEX be the secondary factors, respectively. After a number of experiments, the best value found of the number of intervals in the universes of discourse of the main factor and each secondary factor is "13". For comparing the experimental results of the proposed method with the ones of the methods presented in [7,9,23] and [35], we also divide the historical data of the main factor and each secondary factor from March 1, 2006, to March 1, 2007, into two parts, i.e., we let the data from March 1, 2006, to October 26, 2006, and the data from October 27, 2006, to March 1, 2007, be the training data set and the testing data set, respectively. Moreover, in order to compare the experimental results of the proposed method with the ones of the existing methods [7,9,23,35] for forecasting the NTD/USD exchange rates, we also evaluate the performance of the proposed method using the mean square error (MSE), shown as follows:

$$\mathrm{MSE} = \frac{\sum_{t=3}^{t_{max}} (F_t - R_t)^2}{t_{max} - 2}, \tag{19}$$

where $t_{\max}$ denotes the number of trading days of the historical testing data, F_t denotes the forecasted value of the historical testing datum on trading day t, and R_t denotes the actual datum of the historical testing datum on trading day t, where $3 \le t \le t_{\max}$. Table 4 shows a comparison of MSEs for different methods for forecasting the NTD/USD exchange rates from March 1, 2006, to March 1, 2007. From Table 4, we can see that the proposed method gets smaller MSEs than the ones of the existing methods [7,9,23,35]. In other words, the proposed method gets higher forecasting accuracy rates than the methods presented in [7,9,23] and [35] for forecasting the NTD/USD exchange rates.

6. Conclusions

In this paper, we have proposed a new FTS forecasting method based on optimal partitions of intervals in the universe of discourse and optimal weighting vectors of FTLRGs obtained by PSO techniques simultaneously. From the ranking results shown in Tables 2–4, we can see that the proposed FTS forecasting method gets higher forecasting accuracy rates than the existing methods [3–5]-[10], [12,16,17,35,37]-[39] for forecasting the TAIEX and gets higher forecasting accuracy rates than the existing methods [7,9,23,35] for forecasting the NTD/USD exchange rates. The proposed FTS forecasting method outperforms the existing methods to deal with forecasting problems in terms of forecasting accuracy rates. In [25], Maciel et al. presented a generalized interval evolving possibilistic fuzzy modeling algorithm to process the interval

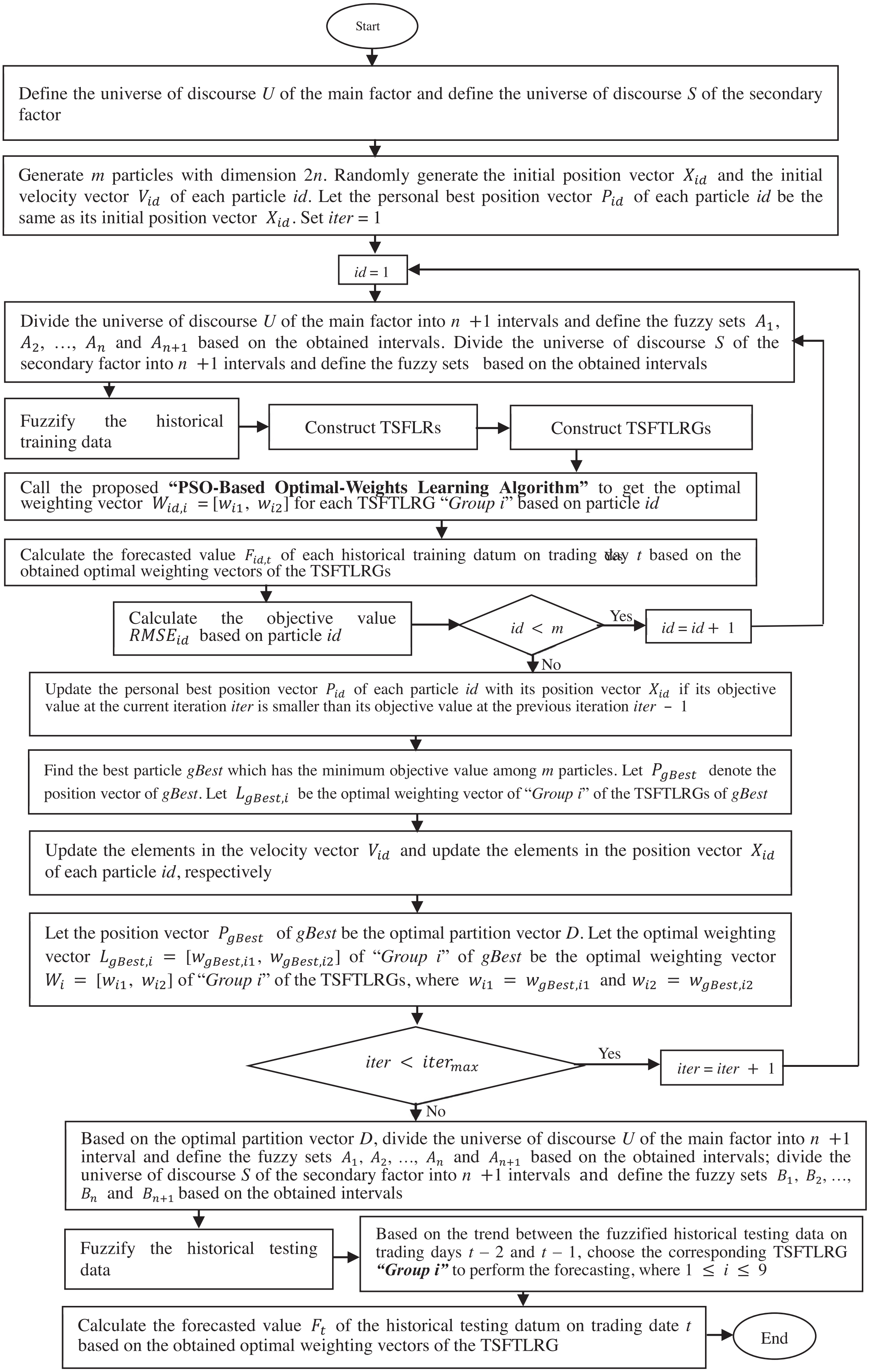

Fig. 5. Flowchart of the proposed FTS forecasting method.

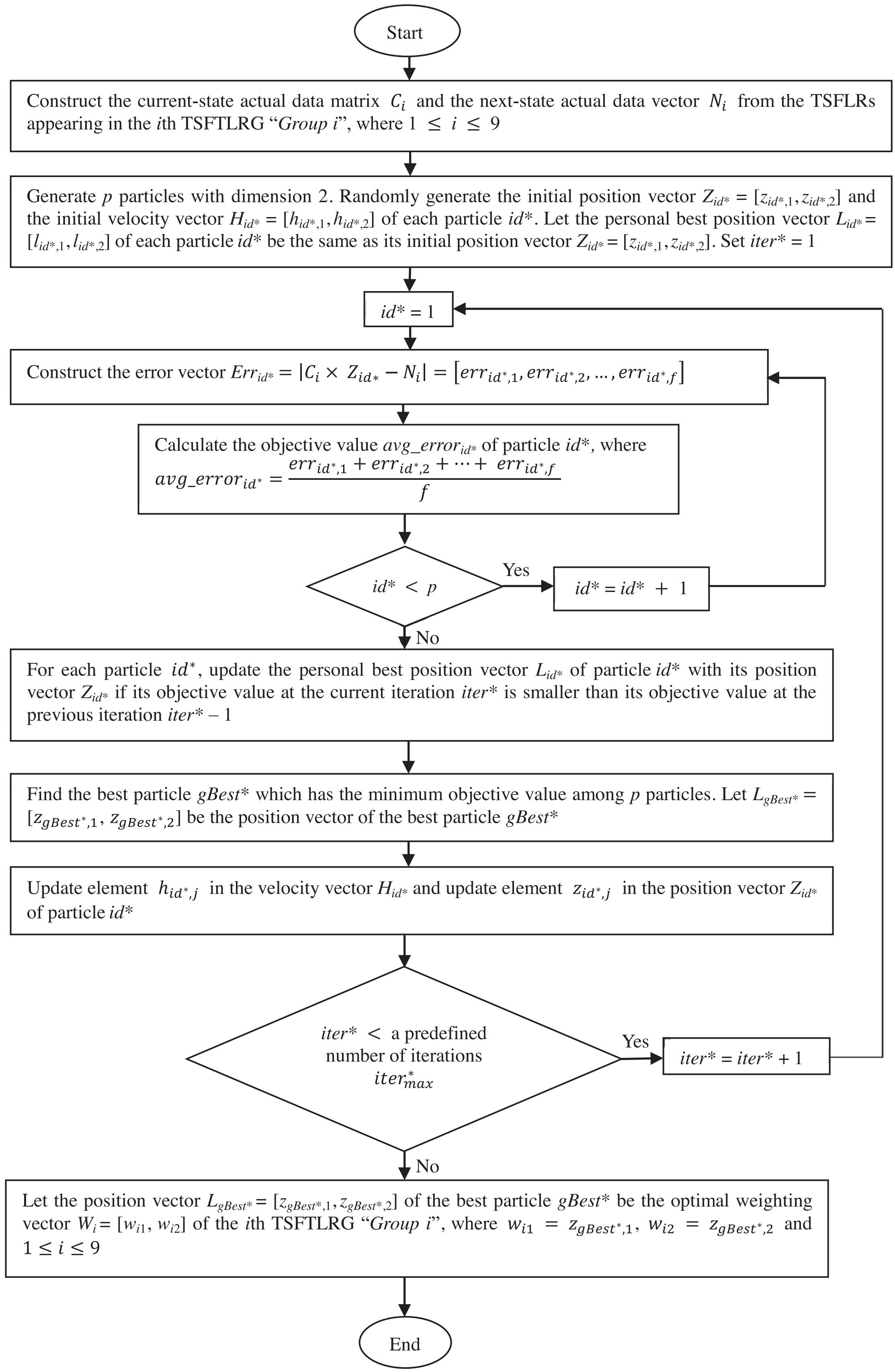

Fig. 6. Flowchart of the proposed "PSO-Based Optimal-Weights Learning Algorithm".

Table 2
A comparison of RMSEs and the average RMSEs for different methods for forecasting the TAIEX from 1999 to 2004.

Methods		1999	2000	2001	2002	2003	2004	Average RMSEs	Ranking
Cai et al.'s method [3]		102.22	131.53	112.59	60.33	51.54	50.33	84.75	5
Chen's FTS model (U_FTS model) [4,38]		120	176	148	101	74	84	117.4	37
Chen and Chang's method [5]	Use NASDAQ	123.64	131.10	115.08	73.06	66.36	60.48	94.95	22
	Use Dow Jones	101.97	148.85	113.70	79.81	64.08	82.32	98.46	27
	Use M1b	156.92	142.70	132.76	96.06	90.27	100.10	119.80	39
	Use Dow Jones & NASDAQ	106.34	130.13	113.33	72.33	60.29	68.07	91.75	16
	Use NASDAQ & M1b	116.22	134.63	116.59	76.48	53.51	69.29	94.45	21
	Use NASDAQ & Dow Jones & M1b	111.7	129.42	113.67	66.82	56.1	64.76	90.41	15
Chen and Chen's method [6]	Use Dow Jones	115.47	127.51	121.98	74.65	66.02	58.89	94.09	20
	Use NASDAQ	119.32	129.87	123.12	71.01	65.14	61.94	95.07	23
	Use M1b	120.01	129.87	117.61	85.85	63.10	67.29	97.29	26
	Use Dow Jones & NASDAQ	116.64	123.62	123.85	71.98	58.06	57.73	91.98	17
	Use Dow Jones & M1b	116.59	127.71	115.33	77.96	60.32	65.86	93.96	18
	Use NASDAQ & M1b	114.87	128.37	123.15	74.05	67.83	65.09	95.56	24
	Use NASDAQ & Dow Jones & M1b	112.47	131.04	117.86	77.38	60.65	65.09	94.08	19
Chen and Chen's method [7]	Use Dow Jones	103.9	127.32	115.37	64.71	52.84	53.36	86.25	9
	Use NASDAQ	104.99	124.52	114.66	64.79	53.63	52.96	85.93	7
	Use M1b	105.61	127.37	115.46	66.07	53.67	53.3	86.91	12
Chen and Kao's method [8]		87.63	125.34	114.57	76.86	54.29	58.17	86.14	8
Chen et al.'s method [9]	Use Dow Jones	102.34	131.25	113.62	65.77	52.23	56.16	86.89	11
	Use NASDAQ	102.11	131.30	113.83	66.45	52.83	54.17	86.78	10
	Use M1b	103.52	131.36	112.55	66.23	53.20	55.36	87.04	13
Cheng et al.'s method [12]		100.74	125.62	113.04	62.94	51.46	54.25	84.68	4
Univariate neural network-based FTS model (U_NN_FTS model) [16,38]		109	255	130	84	56	116	125	40
Univariate neural network-based FTS model use substitutes (U_NN_FTS_S model) [16,38]		109	152	130	84	56	116	107.8	33
Huarng et al.'s method [17]	Use NASDAQ	N/A	158.7	136.49	95.15	65.51	73.57	105.88	32
	Use Dow Jones	N/A	165.8	138.25	93.73	72.95	73.49	108.84	34
	Use M1b	N/A	169.19	133.26	97.1	75.23	82.01	111.36	35
	Use NASDAQ & Dow Jones	N/A	157.64	131.98	93.48	65.51	73.49	104.42	30
	Use NASDAQ & M1b	N/A	155.51	128.44	97.15	70.76	73.48	105.07	31
	Use NASDAQ & Dow Jones & M1b	N/A	154.42	124.02	95.73	70.76	72.35	103.46	29
Ye et al.'s method [35]		101.29	125.42	113.22	63.99	52.99	52.40	84.88	6
Univariate conventional regression model (U_R model) [38]		164	420	1070	116	329	146	374.2	42
Univariate neural network model (U_NN model) [38]		107	309	259	78	57	60	145	41
Bivariate conventional regression model (B_R model) [38]		103	154	120	77	54	85	98.8	28
Bivariate neural network mode (B_NN model) [38]		112	274	131	69	52	61	116.4	36
Bivariate neural network-based FTS model (B_NN_FTS model) [38]		108	259	133	85	58	67	118.3	38
*Bivariate neural network-based FTS model use substitutes (B_NN_FTS_S model) [38,40]		112	131	130	80	58	67	96.4	25
Yu and Huarng's method [39]		N/A	149.59	98.91	78.71	58.78	55.91	88.38	14
The proposed method	Use Dow Jones	99.97	126.59	110.17	61.62	53.01	53.28	84.11	**1**
	Use NASDAQ	99.43	127.05	112.56	60.73	53.07	52.67	84.25	3
	Use M1b	97.33	125.46	113.26	63.08	53.61	52.05	84.13	2

**Note:* In [40], Yu and Huarng have corrected the typing errors appearing in Table VI of [38].

Table 3
A comparison of RMSEs and the average RMSEs for different methods for forecasting the TAIEX from 1990 to 1999.

Methods		1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	Average RMSEs	Ranking
Cai et al.'s method [3]		187.10	39.58	39.37	101.80	76.32	56.05	49.45	123.98	118.41	102.34	89.44	5
Chen and Chen's method [6]	Use Dow Jones	172.89	72.87	43.44	103.21	78.63	66.66	59.75	139.68	124.44	115.47	97.70	11
	Use NASDAQ	169.93	66.12	49.61	104.75	75.66	67.01	60.90	140.86	144.13	119.32	99.83	13
	Use Dow Jones & NASDAQ	172.99	74.85	43.78	101.38	78.13	68.14	61.26	139.29	132.94	116.64	98.94	12
Chen and Chen's method [7]	Use Dow Jones	180.36	43.80	43.06	104.89	75.35	55.06	50.06	133.82	112.11	103.90	90.24	7
	Use NASDAQ	174.15	45.04	42.10	104.94	76.40	54.96	50.17	133.45	113.37	104.99	89.96	6
Chen and Kao's method [8]		156.47	56.50	36.45	126.45	105.52	62.57	51.50	125.33	104.12	87.63	91.25	8
Chen et al.'s method [9]	Use Dow Jones	174.35	43.78	43.12	108.02	88.32	53.69	51.02	139.86	113.58	102.34	91.81	10
	Use NASDAQ	176.17	43.16	43.34	106.66	87.95	53.30	51.10	138.41	113.88	102.11	91.61	9
Cheng et al.'s method [12]		168.77	46.63	44.53	105.30	77.17	50.34	53.19	131.45	115.89	100.74	89.40	4
Ye et al.'s method [35]		189.30	41.74	38.46	103.72	61.90	48.85	50.72	115.77	114.21	110.09	87.47	2
Conventional models [37]	Average-Based Lengths	220	80	60	110	112	79	54	148	167	149	117.9	15
	Distribution-based Lengths	270	79	60	105	132	79	52	149	159	159	124.4	17
Yu's method [37]	Average-Based Lengths	227	61	67	105	135	70	54	133	151	142	114.5	14
	Distribution-based Lengths	266	67	56	105	114	70	52	152	154	145	118.1	16
The proposed method	Use Dow Jones	173.08	40.13	41.33	104.14	70.20	52.31	49.33	127.00	109.03	99.97	86.65	**1**
	Use NASDAQ	174.63	42.66	41.03	103.87	73.22	52.51	49.26	129.00	109.98	99.43	87.56	3

Table 4
A comparison of MSEs for different methods for forecasting the NTD/USD exchange rates from March 1, 2006, to March 1, 2007.

Methods			MSEs	Average MSEs	Ranking
Chen and Chen's method [7]	Use JPY/USD	One-day forecasting	0.004710	0.004609	5
		Three-day forecasting	0.004495		
		Five-day forecasting	0.004625		
		Seven-day forecasting	0.004604		
	Use KRW/USD	One-day forecasting	0.004775	0.004645	7
		Three-day forecasting	0.004526		
		Five-day forecasting	0.004708		
		Seven-day forecasting	0.004571		
	Use CNY/USD	One-day forecasting	0.004893	0.004689	9
		Three-day forecasting	0.004514		
		Five-day forecasting	0.004689		
		Seven-day forecasting	0.004658		
	Use TAIEX	One-day forecasting	0.004819	0.004644	6
		Three-day forecasting	0.004505		
		Five-day forecasting	0.004599		
		Seven-day forecasting	0.004654		
Chen *et al*'s. method [9]	Use JPY/USD	One-day forecasting	0.004954	0.005321	13
		Three-day forecasting	0.005668		
		Five-day forecasting	0.004986		
		Seven-day forecasting	0.005675		
	Use KRW/USD	One-day forecasting	0.005100	0.005197	11
		Three-day forecasting	0.004963		
		Five-day forecasting	0.004967		
		Seven-day forecasting	0.005758		
	Use CNY/USD	One-day forecasting	0.004961	0.005145	10
		Three-day forecasting	0.004727		
		Five-day forecasting	0.004954		
		Seven-day forecasting	0.005936		
	Use TAIEX	One-day forecasting	0.004869	0.005275	12
		Three-day forecasting	0.004759		
		Five-day forecasting	0.005493		
		Seven-day forecasting	0.005979		
Random walk [23]		One-day forecasting	0.011100	0.051975	15
		Three-day forecasting	0.037600		
		Five-day forecasting	0.058200		
		Seven-day forecasting	0.101000		
Radial basis function neural network [23]		One-day forecasting	0.035900	0.066275	16
		Three-day forecasting	0.077200		
		Five-day forecasting	0.087000		
		Seven-day forecasting	0.065000		
Leu *et al*'s method [23]		One-day forecasting	0.006500	0.034525	14
		Three-day forecasting	0.028300		
		Five-day forecasting	0.050100		
		Seven-day forecasting	0.053200		
Ye et al.'s method [35]		One-day forecasting	0.004670	0.004666	8
		Three-day forecasting	0.004664		
		Five-day forecasting	0.004666		
		Seven-day forecasting	0.004665		
The proposed method	Use JPY/USD	One-day forecasting	0.004652	0.004467	**1**
		Three-day forecasting	0.004313		
		Five-day forecasting	0.004453		
		Seven-day forecasting	0.004451		
	Use KRW/USD	One-day forecasting	0.004640	0.004512	4
		Three-day forecasting	0.004429		
		Five-day forecasting	0.004532		
		Seven-day forecasting	0.004445		
	Use CNY/USD	One-day forecasting	0.004708	0.004507	3
		Three-day forecasting	0.004383		
		Five-day forecasting	0.004420		
		Seven-day forecasting	0.004515		
	Use TAIEX	One-day forecasting	0.004725	0.004499	2
		Three-day forecasting	0.004319		
		Five-day forecasting	0.004410		
		Seven-day forecasting	0.004540		

data stream and to produce interval forecasts. Their proposed algorithm uses interval arithmetic in its processing steps and calculates the (dis)similarity between interval data using the Hausdorff distance. Their computational experiments include the forecasting of an interval time series data produced by a synthetic time-varying model with parameter drift. In the future, we will propose a method for interval time series forecasting based on evolving granular analytics [25].

Acknowledgements

The authors would like to thank Professor Shou-Hsiung Cheng for his help during this work. This work is supported by the Ministry of Science and Technology, Republic of China, under grant MOST 104-2221-E-011-084-MY3.

References

[1] E. Bai, W.K. Wong, W.C. Chu, M. Xia, F. Pan, A heuristic time-invariant model for fuzzy time series forecasting, Expert Syst. Appl. 38 (3) (2011) 2701–2707.
[2] J. Blondin, Particle swarm optimization: a tutorial, [Online]. Available: www.cs.armstrong.edu/saad/csci8100/pso_tutorial.pdf, 2009.
[3] Q. Cai, D.F. Zhang, W. Zheng, S.C.H. Leung, A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression, Knowl.-Based Syst. 74 (2015) 61–68.
[4] S.M. Chen, Forecasting enrollments based on fuzzy time series, Fuzzy Sets Syst. 81 (3) (1996) 311–319.
[5] S.M. Chen, Y.C. Chang, Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy interpolation techniques, Inf. Sci. 180 (24) (2010) 4772–4783.
[6] S.M. Chen, C.D. Chen, TAIEX forecasting based on fuzzy time series and fuzzy variation groups, IEEE Trans. Fuzzy Syst. 19 (1) (2011) 1–12.
[7] S.M. Chen, S.W. Chen, Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and the probabilities of trends of fuzzy logical relationships, IEEE Trans. Cybern. 45 (3) (2015) 405–417.
[8] S.M. Chen, P.Y. Kao, TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines, Inf. Sci. 247 (2013) 62–71.
[9] S.M. Chen, G.M.T. Manalu, J.S. Pan, H.C. Liu, Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques, IEEE Trans. Cybern. 43 (3) (2013) 1102–1117.
[10] S.M. Chen, B.D.H. Phuong, S.H. Cheng, A new method for forecasting the TAIEX based on optimal partitions of intervals, optimal weights, and PSO techniques, in: Proceedings of the 2016 International Conference on Machine Learning and Cybernetics, Jeju, South Korea, 2016, pp. 633–637.
[11] S. Cheng, Y. Shi, Q. Qin, Experimental study on boundary constraints handling in particle swarm optimization: from population diversity perspective, Int. J. Swarm Intell. Res. 2 (3) (2011) 43–69.
[12] S.H. Cheng, S.M. Chen, W.S. Jian, Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures, Inf. Sci. 327 (2016) 272–287.
[13] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of the 2000 IEEE Congress on Evolutionary Computation, vol. 1, La Jolla, California, U. S. A., 2000, pp. 84–88.
[14] K. Huarng, Effective lengths of intervals to improve forecasting in fuzzy time series, Fuzzy Sets Syst. 123 (3) (2001) 387–394.
[15] K. Huarng, T.H.K. Yu, Ratio-based lengths of intervals to improve fuzzy time series forecasting, IEEE Trans. Syst. Man Cybern. Part B 36 (2) (2006) 328–340.
[16] K. Huarng, T.H.K. Yu, The application of neural networks to forecast fuzzy time series, Physica A 363 (2) (2006) 481–491.
[17] K. Huarng, T.H.K. Yu, Y.W. Hsu, A multivariate heuristic model for fuzzy time-series forecasting, IEEE Trans. Syst. Man Cybern.-Part B 37 (4) (2007) 836–846.
[18] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, 4, Perth, WA, Australia, 1995, pp. 1942–1948.
[19] I.H. Kuo, S.J. Horng, T.W. Kao, T.L. Lin, C.L. Lee, Y. Pan, An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization, Expert Syst. Appl. 36 (3) (2009) 6108–6117.
[20] L.W. Lee, L.H. Wang, S.M. Chen, Y.H. Leu, Handling forecasting problems based on two-factors high-order fuzzy time series, IEEE Trans. Fuzzy Syst. 14 (3) (2006) 468–477.
[21] L.W. Lee, L.H. Wang, S.M. Chen, Temperature prediction and TAIFEX forecasting based on fuzzy logical relationships and genetic algorithms, Expert Syst. Appl. 33 (3) (2007) 539–550.
[22] L.W. Lee, L.H. Wang, S.M. Chen, Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques, Expert Syst. Appl. 34 (1) (2008) 328–336.
[23] Y. Leu, C.P. Lee, Y.Z. Jou, A distance-based fuzzy time series model for exchange rates forecasting, Expert Syst. Appl. 36 (4) (2009) 8107–8114.
[24] J.W. Liu, T.L. Chen, C.H. Cheng, Y.H. Chen, Adaptive-expectation based multi--attribute FTS model for forecasting TAIEX, Comput. Math. Appl. 59 (2) (2010) 795–802.
[25] L. Maciel, R. Ballini, F. Gomide, Evolving granular analytics for interval time series forecasting, Granular Comput. 1 (4) (2016).
[26] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the 1998 IEEE International Conference Evolutionary Computation, Anchorage, AK, U. S. A., 1998, pp. 69–73.
[27] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, 1999, pp. 1945–1950.
[28] P. Singh, B. Borah, Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization, Int. J. Approximate Reasoning 55 (3) (2014) 812–833.
[29] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part I, Fuzzy Sets Syst. 54 (1) (1993) 1–9.
[30] Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets Syst. 54 (3) (1993) 269–277.
[31] Q. Song, B.S. Chissom, Forecasting enrollments with fuzzy time series-part II, Fuzzy Sets Syst. 62 (1) (1994) 1–8.
[32] B. Sun, H. Guo, H.R. Karimi, Y. Ge, S. Xiong, Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series, Neurocomputing 151 (3) (2015) 1528–1536.
[33] P. Umapathy, C. Venkataseshaiah, M.S. Arumugam, Particle swarm optimization with inertia weight variants for optimal power flow solution, Discrete Dyn. Nat. Soc. 2010 (2010) Article ID 462145, 15 pages (http://dx.doi.org/10.1155/2010/462145).
[34] W. Wang, X. Liu, Fuzzy forecasting based on automatic clustering and axiomatic fuzzy set classification, Inf. Sci. 294 (2015) 78–94.
[35] F. Ye, L. Zhang, D. Zhang, H. Fujita, Z. Gong, A novel forecasting method based on multi-order fuzzy time series and technical analysis, Inf. Sci. 367-368 (2016) 41–57.
[36] U. Yolcu, E. Egrioglu, V.R. Uslu, M.A. Basaran, C.H. Aladag, A new approach for determining the length of intervals for fuzzy time series, Appl. Soft Comput. 9 (2) (2009) 647–651.
[37] T.H.K. Yu, Weighted fuzzy time-series model for TAIEX forecasting, Physica A 349 (3-4) (2004) 609–624.
[38] T.H.K. Yu, K.H. Huarng, A bivariate fuzzy time series model to forecast the TAIEX, Expert Syst. Appl. 34 (4) (2008) 2945–2952.
[39] T.H.K. Yu, K.H. Huarng, A neural network-based fuzzy time series model to improve forecasting, Expert Syst. Appl. 37 (4) (2010) 3366–3372.
[40] T.H.K. Yu, K.H. Huarng, Corrigendum to “a bivariate fuzzy time series model to forecast the TAIEX, Expert Syst. Appl. 37 (7) (2010) 5529.
[41] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.
[42] TAIEX. [Online]. Available: http://www.twse.com.tw/en/products/indices/tsec/taiex.php.
[43] NTD/USD exchange rates. [Online]. Available: http://www.cbc.gov.tw/content.asp?mp=1&CuItem=36599.
[44] Monetary Aggregate M1b. [Online]. Available: http://www.cbc.gov.tw/mp2.html.
[45] Dow Jones Industrial Average Index. [Online]. Available: http://www.djindexes.com/mdsidx/?event=historicalValuesDJI.
[46] NASDAQ. [Online]. Available: http://www.nasdaq.com/symbol/nasdaq/historical