

Research Article

An Intelligence Optimized Rolling Grey Forecasting Model Fitting to Small Economic Dataset

Li Liu,^{1,2} Qianru Wang,^{2,3} Ming Liu,⁴ and Lian Li⁵

¹ School of Computing, National University of Singapore, Singapore 117417

² School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China

³ Department No. 68028, Lanzhou Military Region, Lanzhou 730058, China

⁴ Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China

⁵ Department of Computer Science and Technology, HeFei University of Technology, Hefei 230009, China

Correspondence should be addressed to Li Liu; dcslili@gmail.com

Received 10 October 2013; Accepted 9 April 2014; Published 28 April 2014

Academic Editor: Jaeyoung Chung

Copyright © 2014 Li Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grey system theory has been widely used to forecast the economic data that are often highly nonlinear, irregular, and nonstationary. The size of these economic datasets is often very small. Many models based on grey system theory could be adapted to various economic time series data. However, some of these models did not consider the impact of recent data or the effective model parameters that can improve forecast accuracy. In this paper, we proposed the PRGM(1,1) model, a rolling mechanism based grey model optimized by the particle swarm optimization, in order to improve the forecast accuracy. The experiment shows that PRGM(1,1) gets much better forecast accuracy among other widely used grey models on three actual economic datasets.

1. Introduction

Forecasting can be an important issue to many fields of economy; especially its accuracy was ensured to do a reasonable prediction that could change the economic policy of large companies and governments and ensure a more reasonable behavior by the financial actors. The ideal state is that the prediction error tends to be more and more smaller, but in fact, we can only do our best to research and develop the prediction algorithm as much as possible to improve the prediction accuracy.

Many forecasting models have been proposed; in general, these models can be divided into two categories: causal models and time-series models [1]. Causal models assume that historical relationship between dependent and independent variables will remain valid in future. Causal models include multiple linear regression analysis and econometric models which assume that independent variables could explain the variations in dependent variable. However, the limitation of causal models is the availability and reliability of independent variables.

Time-series models assume that history will repeat itself and its prediction refers to the process by which the future values of a system are forecasted based on the information obtained from the past and current data points. In the literature, two main techniques for time series prediction are statistical and artificial intelligence (soft computing) based approaches. The well-known statistical models proposed include AR (autoregressive), MA (moving average), ARMA (autoregressive moving average), ARIMA (autoregressive integrated moving average), and Box-Jenkins models. The statistical models are too weak to solve the nonlinear problems and too complex to be used in predicting future values of a time series.

The widely used artificial intelligence approaches include neural network (NN) [2–4], support vector machines (SVM) [5–8], fuzzy systems [9], linear regression, Kalman filtering [10], and hidden Markov models (HMM) [3]. All of these approaches are used for updating the model parameters. In the recent years, several hybrid models [11–14] were proposed to improve the forecast accuracy. However, these artificial intelligence based approaches demand a great deal

of training data and relatively long training period for robust generalization [11]. For those economic predictions, which are very difficult to construct a model by using neither the conventional linear statistical methods nor the artificial neural networks because the economic time series are highly nonlinear, highly irregular, and highly nonstationary [15].

Grey system theory was introduced and developed by Deng back in 1989 to be used for mathematical analysis on the phenomenon of uncertainty and roughness. It requires a small set of training data, which are discrete or incomplete, to construct a model for future forecast. The uncertainty and roughness training data are “grey” data [16]. Similarly, “white” data means that the information is completely clear, while “black” indicates that the information is completely unclear.

Grey system theory has been widely and successfully used to forecast all kinds of data in the many areas such as economic, financial, agricultural, and industrial areas and energy. In the past few years, grey system theory has been employed for solving the forecasting economic problems. The model GM(1,1) built from grey system theory has shown that this approach is very efficient to forecast the irregular and nonlinear economic time series data. A combination of residual modification and residual artificial neural network (ANN) sign estimation is proposed to improve the accuracy of the original GM(1,1) model [17–19]. However, this approach needs long training period.

Rolling mechanism is one of the most effective methods to improve the performance of grey system model and handle noisy data [7, 20–22]. The authors in [22] used the rolling mechanism to improve the forecast accuracy of grey model for education expenditure. Zhao et al. [23] proposed rolling mechanism to forecast the per capita annual net income of rural households in China and showed that it outperformed other traditional grey prediction models and a differential evolution algorithm proposed to optimize rolling grey prediction model. The authors in [24–26] proposed an improved rolling grey model, which can update the model parameters on the coal production forecast and semiconductor industry production forecast, respectively.

However, although these improved rolling mechanism based grey models could adapt to various economic time series data because they considered the recent data that can improve forecast accuracy in future prediction, they did not consider the impact of their model parameters which are fixed through the whole prediction period or only considered a simple change of the model parameters for the prediction which could perform well on noiseless sequence, but it could not adapt to the noisy data.

In this paper, we proposed an improved rolling mechanism based grey model optimized by the particle swarm optimization (PSO for short) to improve the forecast accuracy, especially for the highly irregular and noiseless data. PSO, which belongs to swarm intelligence methods, is considered as a tool for modeling behavior and for optimization of difficult numerical solutions, since it was developed by [27] as an evolutionary computing technology. PSO algorithm had been enormously successful on about 700 applications [28]. We choose PSO to optimize our model parameters for two significant reasons: its routinely delivering of good

optimization results like NN methods and its simplicity to get better results in a faster and cheaper way that NN methods cannot achieve.

This paper examines a rolling mechanism based grey model with PSO optimization on economic data. Section 2 outlines the original grey model GM(1,1) and the improved GM(1,1) model with rolling mechanism. Section 3 presents the rolling mechanism based grey model with PSO optimization. We also propose a PSO based algorithm that searches the best value for the model parameter. Furthermore, we illustrate that our model gets much better performance on three economic dataset: financial intermediation in Beijing, real estate in Beijing, and semiconductor production in Taiwan, compared with other grey system theory based models. Section 5 concludes this paper.

2. Grey Model Background

The grey system theory mainly focuses on extracting realistic governing laws of the system from the available data of the system generally with white noise data. A grey model in grey system theory is denoted by GM(n, m), where n indicates the order of the difference equation and m indicates the number of variables.

GM(1,1) is the original grey model, which has been widely applied to carry on the short-term prediction because of its computational efficiency. It uses a first order differential equation to predict an unknown system. A GM(1,1) algorithm is described below.

Step 1. The original time sequence is initiated by

$$x^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)), \quad (1)$$

where $x^{(0)}(i)$ the time series data at time i and n is the length of sequence which must be equal to or larger than 4.

On the basis of the initial sequence $x^{(0)}$, a new sequence

$$x^{(1)} = (x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)) \quad (2)$$

is set up through the accumulated generating operator (AGO), which is monotonically increasing to weaken the variation tendency defined as

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i). \quad (3)$$

Grey system theory is applied to accumulate generation of $x^{(0)}$ to obtain a new sequence $x^{(1)}$, which has a clear growing tendency.

Step 2. Establishing the first-order differential equation of grey model GM(1,1) as

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \quad (4)$$

and its difference equation is

$$x^{(0)}(k) + az^{(1)}(k) = b, \quad 2 \leq k \leq n, \quad (5)$$

where a is the development coefficient, b is the driving coefficient, and $z^{(1)} = (z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(n))$ is the generated sequence of $z^{(1)}(k) = \alpha x^{(1)}(k) + (1 - \alpha)x^{(1)}(k - 1)$.

In the original GM(1,1), α is set to the mean value of adjacent data $z^{(1)}(k) = 0.5 \cdot x^{(1)}(k) + 0.5 \cdot x^{(1)}(k - 1)$. In this paper, we proposed a method by using the PSO algorithm to find a more efficient value of α .

Step 3. From (5), we can obtain the following equation:

$$\begin{aligned} x^{(0)}(2) + az^{(1)}(2) &= b, \\ x^{(0)}(3) + az^{(1)}(3) &= b, \\ &\vdots \\ x^{(0)}(n) + az^{(1)}(n) &= b. \end{aligned} \quad (6)$$

In the above, $P = [a, b]^T$ is a sequence of coefficient parameters that can be computed by employing the least squares method:

$$P = (B^T B)^{-1} B^T Y_N, \quad (7)$$

where y_N is the constant vector

$$Y_N = [x^{(0)}(2), x^{(0)}(3), \dots, x^{(0)}(n)]^T \quad (8)$$

and B is the accumulated matrix

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix}. \quad (9)$$

Step 4. Substituting P in (6) with (7), the solution of the prediction value of $x^{(1)}$ at time k is

$$\hat{x}^{(1)}(k) = \left[x^{(1)}(1) - \frac{b}{a} \right] e^{-a(k-1)} + \frac{b}{a}. \quad (10)$$

After performing an inverse accumulated generating operation on (10), the predicted value of $x^{(0)}(k)$ at time k is $\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k - 1)$, where $2 \leq k \leq n$.

GM(1,1) uses the whole data set for prediction. However, the recent data can improve forecast accuracy in future prediction [21]. Rolling mechanism, which is a metabolism technique that updates the input data by discarding old data for each loop in grey prediction, can be applied to perform the perfect prediction. The purpose of RM is that, in each rolling step, the data utilized for next forecast is the most recent data. The RM-GM is an efficient technique to increase the forecast accuracy in the case of having noisy data. The data may exhibit different trends or characteristics at different times, so to address these differences, it is preferable to study such noisy data with the RM-GM, and the RM provides a means to guarantee input data are always the most recent values.

3. PSO Optimized RM-GM Model

Because α directly influences the calculation of a and b in GM(1,1) model and is one of the most important factors that may decide the performance of the models; we present an algorithm based on RM-GM(1,1) combined with PSO which optimizes the parameter α in each rolling period to improve the forecast accuracy.

In basic GM(1,1) model, the value of α is customarily set to the mean value 0.5 for each $z^{(1)}(k) = \alpha x^{(1)}(k) + (1 - \alpha)x^{(1)}(k - 1)$ in the generated sequence $z^{(1)} = (z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(n))$. It means that each data has the equal impact on every future predicted data. However, the authors in [29] found that GM(1,1) model often performs very poor and makes delay errors for quick growth sequences because of the mean value on the generated sequence $z^{(1)}$. Tan proposed a method that set α to $(p - 1)/2p$, where $p = (\sum_{k=2}^n (x^{(1)}(k)/x^{(1)}(k - 1)))$, in order to widen the adaptability of GM(1,1) model to various kinds of time sequences. The authors in [26] found that the RM-GM with variable α value generates better forecasts than with a fixed α value. They determined the α value by the timely percent change. From this study, we can find that for the trend prediction of nonmonotonous functions, the forecast outcomes are much better if the value of α is set appropriately on the grey predicted results. However, Tan's method used the whole data set to calculate a fixed value of α . It did not consider the influence of recent data which would improve accuracy.

In an improved RM-GM(1,1) algorithm, the strategy of finding a value of α could be proposed in a variety of ways. The basic RM-GM(1,1) sets the value of α to 0.5, which does not consider any influence of sequence data. Although Tan's strategy could adapt to various sequences, it did not consider the impact of the recent data from the sequence. Chang's strategy only considered the timely percent change for the prediction. It could perform well on regular and noiseless sequence, but it could not acclimatize itself to the noisy data sequence. In this paper, we select PSO as our strategy to find the value of α in each loop in α -RM-GM(1,1). We named our PSO-based algorithm as PRGM(1,1).

3.1. Characteristics of PSO. Two significant reasons that make using PSO to calculate the parameter α are its routinely delivering good optimization results and its simplicity. Compared with another commonly used swarm intelligence method, ant colony optimization (ACO), which is not easy to be used to define variables for the given problems, PSO is not only a metaheuristic that makes few or no assumptions about the problem being optimized, but can also search very large spaces of candidate solutions. It does not require that the optimization problem be differentiable. Since the problem of predicting economic data is partially irregular, noisy, and, changing over time, PSO is a better choice to be employed to optimize parameter α . Another one of the most significant advantages of PSO algorithm is its relatively simple coding and low computational cost. Compared with other optimization algorithms, like ACO, which requires massive computation, PSO can get better results in a faster and

cheaper way [30]. Hence, PSO algorithm can even perform well in the applications that need power-aware computing on smart or personal devices that have limited computational, storage, and energy resources in the case of guarantying the prediction accuracy.

3.2. Calculating α by PSO. The PSO is a population-based optimization technique in which the optimal solution can be found by iteration and the solution quality is evaluated by the fitness. In the PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) that it has achieved so far. First, a dimensional space D with m particles is initialized. The particles' position and velocity are randomly initialized. The position of the i th particle is represented as $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$ and its velocity is represented as $v_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$, where $1 \leq i \leq m$ and $1 \leq d \leq D$. Then the objective function values (forecast errors) of all particles can be computed. Then, the particles are updated iteratively until the termination condition is satisfied. It includes the particles' own speed and location according to the following two formulas for all particles:

$$\begin{aligned} v_{id}^{k+1} &= w \times v_{id}^k + c_1 \times \text{rand} \times (p\text{Best}_{id} - x_{id}^k) \\ &\quad + c_2 \times \text{rand} \times (g\text{Best}_d - x_{id}^k), \\ x_{id}^{k+1} &= x_{id}^k + v_{id}^{k+1}, \quad i = 1, 2, \dots, m, \quad d = 1, 2, \dots, D, \end{aligned} \quad (11)$$

where, $p\text{Best}_{id}$ and $g\text{Best}_d$ are determined as the objective function values fitness which should be set according to the actual problem solving. For the prediction, it can be set to the smallest prediction error. $p\text{Best}_{id}$ and $g\text{Best}_d$, respectively, represent the individual extreme value of the i th particle found by the particle itself at d th dimension and the global optimal value which records the best particle among all the particles in the group; k is the pointer of iterations; c_1 and c_2 are two positive acceleration constants; $\text{rand}()$ is the uniform random value in the range $[0, 1]$; v_i^k is the velocity of a particle i at iteration k ; $v_{d\min} \leq v_{id}^k \leq v_{d\max}$ is the current position of the i th particle at iteration k ; w ($0 \leq w \leq 1$) is the inertia weight determining how much of the particle's previous velocity is preserved. If the current value is better (with smaller forecast accuracy index value), then update the best position and its objective function value of the particle with the current position and corresponding objective function value. Finally, determine the best particle of the whole population based on their best objective function values. If their objectives function value is smaller than the current global optimal objective function value $\text{Fitness}_{\text{Best}_t}$, then update the best position and objective function value for the entire swarm with the current best particle's position and objective function value.

3.3. Parameter Selection. In α -PSO algorithm, the values for the cognitive weight (c_1), social weight (c_2), and the inertia

weight (w) having to be selected would have an impact on the convergence speed and the ability of the algorithm to find the optimum. However, different values may be better for different problems. Many works have been done to select a combination of values that works well in a wide range of problems. Both theoretical and empirical studies are available to help in selection of proper values [31–34].

Generally, the individual and sociality weights c_1 and c_2 are both set to 2. A proper value of inertia weight provides a balance between global and local explorations. A large inertia weight favors global search, while a small inertia weight favors local search [31, 35]. In practice, w often reduces linearly from about 0.4 (w_{\min}) to 0.9 (w_{\max}). The authors in [31] suggested that utilizing LDW (linear decreasing weight) policy which improved a lot compared with optimization of the benchmark equation algorithm, but not the most common and suitable for the reason that demanding the searching process is linear. It is suggested that for each iteration setting the inertia weight according to the following equation may be a better choice:

$$\begin{aligned} c_1(k) &= c_1^+ - (c_1^+ - c_1^-) \frac{k}{k_{\max}}, \\ c_2(k) &= c_2^- + (c_2^+ - c_2^-) \frac{k}{k_{\max}}. \end{aligned} \quad (12)$$

A proper value of the inertia weight provides a balance between global and local explorations. A large inertia weight favors global search, while a small inertia weight favors local search. In general, settings near 1 facilitate global search, and settings ranging from $[0.2, 0.5]$ facilitate rapid local search. The linear decreasing weight (16) is introduced to dynamically adapt the inertia weight (13). w^+ and w^- are usually set to 0.9 and 0.4:

$$w(k) = w^+ - (w^+ - w^-) \frac{k}{k_{\max}}. \quad (13)$$

The nonlinearly decreasing inertia weight (14) incorporates the hyperbolic tangent function (15) to update w_i of each particle i :

$$w_i(k) = \frac{1}{1 + \tanh(\text{NI}_i(k))}, \quad (14)$$

$$\tanh(z) = \frac{e^{-z} - e^z}{e^{-z} + e^z}, \quad (15)$$

where NI_i^k is the neighborhood index of the particle i , which is calculated at each iteration as

$$\text{NI}_i^k = \frac{\text{Fitness}_i^k - g\text{Worst}}{g\text{Best} - g\text{Worst}}, \quad (16)$$

where $g\text{Worst}^k$ is the global worst fitness value at the current iteration. A small NI_i^k indicates the current position is bad and needs global exploration with a large inertia weight. On the contrary, a large NI_i^k indicates the requirement of local exploitation with a small inertia weight.

TABLE 1: Criterion of MAPE.

MAPE (%)	Forecasting power
<10	Excellent
10–20	Good
20–50	Reasonable
>50	Incorrect

The constriction factor χ was used to control the magnitude of the velocities, instead of w . The velocity update scheme is replaced with the following:

$$v_i^{k+1} = \chi \times [v_i^k + c_1 \times \text{rand} \times (p\text{Best}_i - \alpha_i^k) + c_2 \times \text{rand} \times (g\text{Best} - \alpha_i^k)], \quad (17)$$

$$\chi = \frac{2k}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \quad (18)$$

where $\phi = c_1 + c_2$ and generally $k = 1$.

4. Experiments and Evaluations

4.1. Datasets. The prediction of the development of tertiary industry is a very important topic in economic and financial areas. However, time series prediction in economic area is generally very difficult because it is nonstationary, nonlinear, and highly noisy.

In order to illustrate that our PRGM(1,1) algorithm gets better performance on both smoothing and noisy data forecasting model by using small set of training data, we used three datasets: *financial intermediation* in Beijing during 1994 to 2010 which has relatively smoothing trends, *real estate* in Beijing during 1994 to 2010 which seems much nonlinear, and *semiconductor industry production* in Taiwan from 1994 to 2002 which seems regular from 1994 to 2000 but irregular since 2000. All datasets are collected from the China Statistical Yearbook, National Bureau of Statistics of China.

4.2. Evaluation Metrics. Prediction accuracy is an important criterion for evaluating a forecasting technique [36]. In this paper, three metrics, namely, mean absolute percentage error (MAPE), mean absolute deviation (MAD), and mean squared error (MSE), which are often adopted for the performance of each model [6, 22], are used to evaluate the prediction accuracy. MAPE is a general accepted metric in percent of prediction accuracy. The criterion of MAPE [37] is listed in Table 1:

$$\text{MAPE}(\%) = \frac{1}{n} \sum_{i=1}^n \left| \frac{x^{(0)}(i) - \hat{x}^{(0)}(i)}{x^{(0)}(i)} \right|. \quad (19)$$

MAD and MSE are two metrics of the average magnitude of the forecast errors, but the latter imposes a greater penalty on a large error than several small errors. The smaller the

values, the closer the predicted values to the actual values [38]:

$$\begin{aligned} \text{MAD} &= \frac{1}{n} \sum_{i=1}^n |x^{(0)}(i) - \hat{x}^{(0)}(i)|, \\ \text{MSE} &= \frac{1}{n} \sum_{i=1}^n (x^{(0)}(i) - \hat{x}^{(0)}(i))^2. \end{aligned} \quad (20)$$

Besides, the coefficient of determination, denoted as r^2 , is also applied to evaluate models in our experiments:

$$r^2 = 1 - \frac{\text{SSE}}{\text{SST}}, \quad (21)$$

where

$$\text{SSE} = \sum (x^{(0)}(k) - \hat{x}^{(0)}(k))^2, \quad k = 1, 2, \dots, n,$$

$$\text{SST} = (\sum x^{(0)}(k)^2 - (\sum x^{(0)}(k))^2/n), \quad k = 1, 2, \dots, n.$$

The higher the value of r^2 is, the more successful the model is at predicting statistical data [23]. The maximum value of the coefficient of determination r^2 is 1.

4.3. Experimental Setup. The experiments are divided into two parts, *Experiment I* and *Experiment II*. *Experiment I* used the datasets of *financial intermediation* and *real estate* in Beijing. The data from 1994 to 2005 were used as sample data, while the data from 2006 to 2010 were used for prediction and test. *Experiment I* compared three prediction models on these data, GM(1,1), RM-GM(1,1), and PRGM(1,1). *Experiment II* compared various PRGM(1,1) with different parameter settings.

The values of the parameters for PRGM(1,1) are selected in both experiments. We set the number of candidates of α in particle searching space *partinum* to 1,000 and the maximum number of iterations iter_{\max} to 100. For the basic PRGM(1,1), we set the two weights, $c_1 = 2$ and $c_2 = 2$.

4.4. Experiment I. Table 2 shows the parameters calculated by the three prediction models, GM(1,1), RM-GM(1,1), and PRGM(1,1). In GM(1,1) which is constructed by all of the data 1994–2005 with the fixed α value 0.5, the parameter a is equal to a fixed value -0.148 and b is also equal to a fixed value 165.061 for all the predicted years in financial intermediation. Similarly, $a = -0.254$ and $b = 42.152$ for all the predicted years in real estate.

In RM-GM(1,1), we set the sample sequence with $l = 12$ and $m = 1$ starting from 1994 to forecast the data from 2006 to 2010. Hence, the rolling number k equals 5. The α value is also fixed to 0.5 in RM-GM(1,1). However, the parameters a and b change for every predicted year because of the rolling mechanism.

In PRGM(1,1), similar with RM-GM(1,1), the sample sequence with $l = 12$ and $m = 1$ that starts from 1994 to 2005 was used for predicting the 5 years' data since 2006. However, the value of α is a variable of year that is different among the predictions of 2006–2010. Hence, the parameters a and

TABLE 2: The parameter values calculated by GM(1, 1), RM-GM(1, 1), and PRGM(1, 1), respectively.

Year	GM(1, 1)			RM-GM(1, 1)			PRGM(1, 1)		
	α	a	b	α	a	b	α	a	b
Financial intermediation									
2006					-0.148	165.061	0.467	-0.147	164.250
2007					-0.146	190.743	0.850	-0.152	202.890
2008	0.500	-0.148	165.061	0.500	-0.144	221.805	0.692	-0.162	204.620
2009					-0.143	251.047	0.384	-0.162	204.700
2010					-0.142	291.635	0.411	-0.156	262.010
Real estate									
2006					-0.254	42.152	0.253	-0.239	39.421
2007					-0.281	35.488	0.317	-0.234	50.558
2008	0.500	-0.254	42.152	0.500	-0.285	41.182	0.131	-0.217	61.900
2009					-0.278	58.408	0.274	-0.197	100.510
2010					-0.273	78.809	0.009	-0.177	123.310

TABLE 3: The evaluation metrics to compare GM(1, 1), RM-GM(1, 1), and PRGM(1, 1).

	GM(1, 1)	RM-GM(1, 1)	PRGM(1, 1)
Financial intermediation			
MAPE (%)	6.3452	8.3619	0.0514
MAD	93.1390	126.4600	0.7191
MSE	12666.000	20607.0000	0.5667
r^2	0.8560	0.7657	1.0000
Real estate			
MAPE (%)	63.8925	61.7673	0.9890
MAD	599.65	579.9800	9.5706
MSE	520120.00	494280.0000	287.4100
r^2	-24.2190	-22.9660	0.9861

b change for every predicted year because of both the rolling mechanism and the variety of α .

Table 3 shows the evaluation metrics among GM(1,1), RM-GM(1,1), and PRGM(1,1). For the dataset of financial intermediation, PRGM(1,1) with the MAPE value 0.0514%, compared with the MAPE value of GM(1,1) and RM-GM(1,1), 6.3452% and 8.3619%, respectively, shows much better prediction performance than the other two models. The MAD and the MSE also indicate the excellent results produced by PRGM(1,1). The coefficient of determination r^2 produced by PRGM(1,1) is nearly to the maximum value 1. For the dataset of real estate, the prediction by PRGM(1,1) model still shows excellent results with the MAPE 0.9890% comparing results 63.8925% and 61.7673% produced by GM(1,1) and RM-GM(1,1), respectively. PRGM(1,1) shows nearly 60 times better performance than either GM(1,1) or RM-GM(1,1) in both the MAPE and the MAD metrics and 2000 times better in the MSE metric. PRGM(1,1) could predict the future data much more successfully with r^2 which equals 0.9861.

Table 4 shows the forecasting results of the semiconductor industry production from 1998 to 2002 predicted by P values RM-GM(1,1) and PRGM(1,1) using the sample data of 1994–2002. We compared our results of PRGM(1,1) with the

results produced by P value RM-GM(1,1) from the literature [26]. The MAPE value of PRGM(1,1), that is, 8.3787%, is better than the value of 10.52% from P-RM-GM(1,1). The error of predication, which is defined as

$$\sum \left| \frac{\text{actual} - \text{predictive}}{\text{actual}} \right| \quad (22)$$

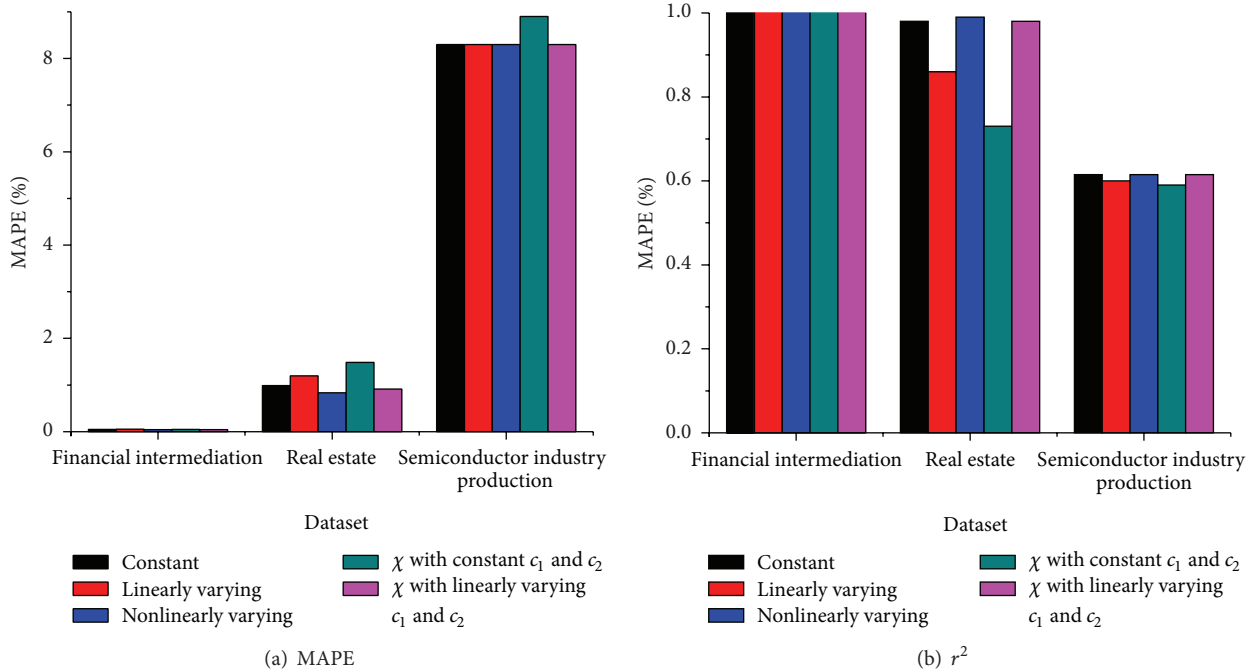
that indicates the deviation degree of the predictive data from the actual data for each year among 1998–2000 from PRGM(1,1), is much lower than from P-RM-GM(1,1). The actual value suddenly fell by more than 10%. PRGM catches the trends well, which means that PRGM has remarkable ability to predict the irregular sequence, especially to sense the unexpected changes. However, P value RM-GM(1,1) model gets the predictive value of 2002 with a very small percentage error 0.48%, but the error value of PRGM(1,1) model is 11.839%. The reason is that PRGM(1,1) model can get better results of matching the trends that the production data rebounded from the slump of 2001. PRGM(1,1) is much better than P value RM-GM(1,1) to forecast the trends of time series sequences, which is significant for the economic prediction.

4.5. Experiment II. In this experiment, we estimated PSO variants of different parameter configurations. We evaluated the constant setting and linearly varying settings of c_1 and c_2 on prediction accuracy. In constant settings, the configuration of $c_1 = c_2 = 1.5$ is the best. It is in accordance with most of the previous conclusions. In linearly varying setting (see (12)), there is not much improvement on the metrics compared with the constant setting. We also evaluated the forecasting performances with diverse combinations of the start values c_1^+ and c_2^- , and the end values c_1^- and c_2^+ ranging from $[0.5, 4]$ with a step of 0.5 and found that there still is not much difference among them for all of the three datasets.

We evaluated three kinds of w settings, constant, linearly decreasing, and nonlinearly decreasing. In constant setting, the optimal setting is $w = 0.5$ for all of the datasets. We also observed that the performance is exactly the same when

TABLE 4: The forecasting data and evaluation metrics produced by PRGM(1, 1) and P value RM-GM(1, 1).

Year	Actual value	<i>P</i> value RM-GM(1, 1)		PRGM(1, 1)	
		Predictive value	Error	Predictive value	Error
Semiconductor industry production					
1998	2834	3020	6.56	2830	0.1519
1999	4235	3749	11.48	4096	3.2841
2000	7144	6546	8.37	6792	4.9309
2001	5269	6624	25.72	6412	21.6878
2002	6529	6498	0.48	7302	11.839
MAPE (%)			10.52		8.3787

FIGURE 1: The influence of PSOs with various parameter settings on MAPE and r^2 .

the population size is 10 with different values of w ranging from $[0, 1]$. In linearly decreasing setting (13), we varied the combinations of w^- and w^+ ranging from $[0, 1]$ with a step of 0.1, respectively. The results showed that there is nearly no difference on the metrics among different combinations. It indicates that the historical setting does not have much impact on the forecasting performance by linearly updating w .

We also used the nonlinearly varying (14) and the constriction factor χ (18) to update particles' velocities (17). Figure 1 shows that the nonlinearly varying setting and the constriction factor setting with linearly varying c_1 and c_2 in the meantime can improve the prediction performance. The nonlinearly varying method does not require an initial setting of w^- or w^+ . It calculates the w dynamically according to the current situation. A large w is set if current position is far away from the global best position, or a small w is set if current position is near to the global best position. The constriction factor can slow down the velocities but needs to combine with

linearly varying method to control the effects of c_1 and c_2 in order to search much more spaces.

Figure 2 shows an illustration of the evolution of the fitness at the first predicted year in all of the datasets. According to our empirical study, the maximum iteration k_{\max} can be set to 60–80 in the single particle PSO. Figure 2 shows the comparison of the convergence speed among variant PSOs. There is no general rule on these PSOs for all of the datasets, but all PSOs converge after 60–80 iterations at most. The time complexity of the PSO is $O(k_{\max} \cdot m \cdot O(\text{Fitness}))$. The runtime is dependent on both population size and iteration number.

5. Conclusions

In this paper, we proposed a rolling mechanism based grey model, and its parameter α is optimized by the PSO algorithm, which has the significant impact of the forecast accuracy. The experiments show that the prediction made by

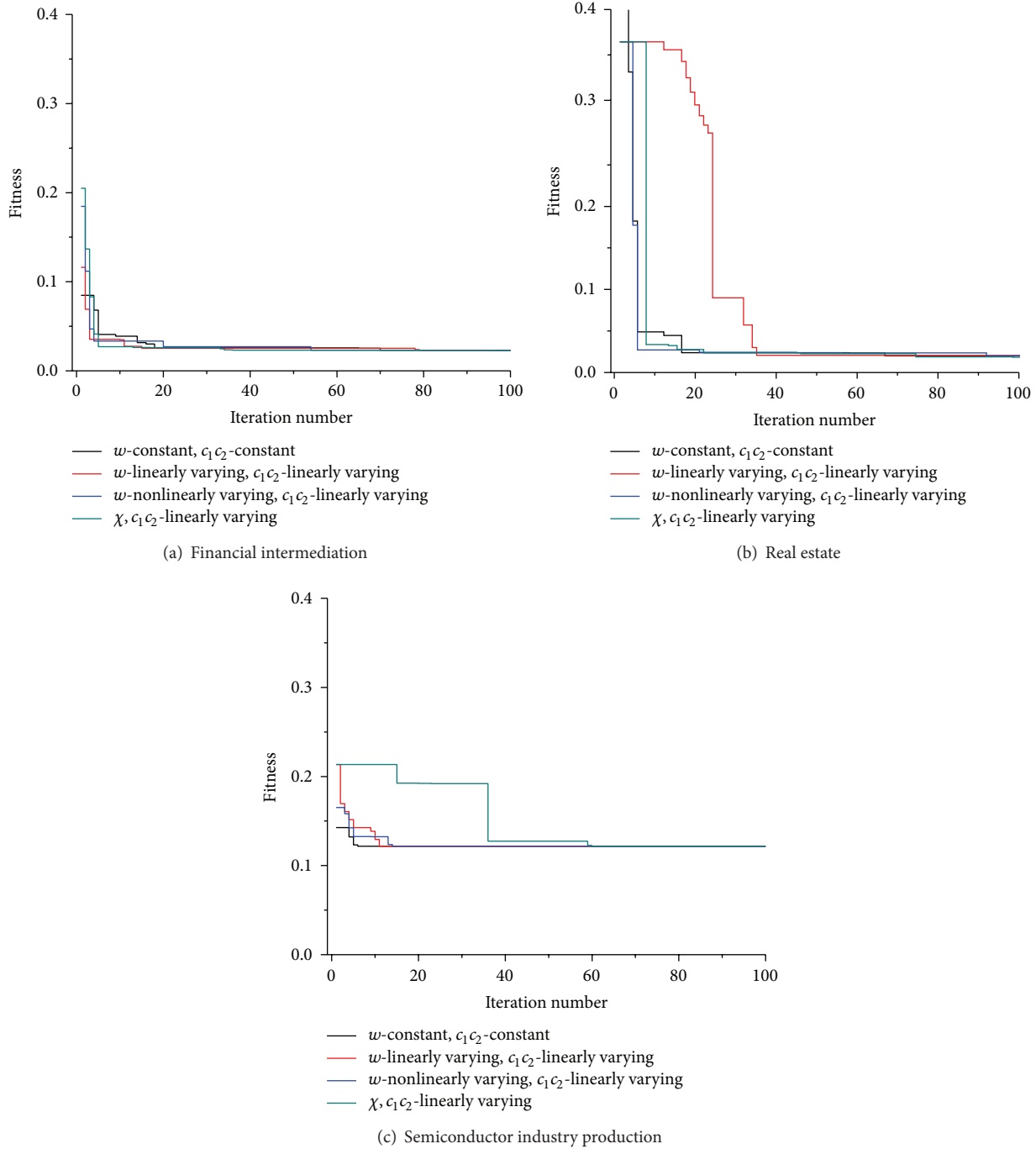


FIGURE 2: The convergence speed comparison of the fitness values among PSOs with various parameter setting methods.

PRGM(1,1) model is almost perfect among three economic datasets, which are either regular or noisy. PRGM(1,1) gets much better forecast accuracy compared with three widely used grey models: GM(1,1) that has a fixed α and ignores the impact of recent data, RM-GM(1,1) that considers the impact of recent data but has a fixed α through the whole prediction period, and P value RM-GM(1,1) that not only considers the recent data but also adjusts α in each rolling step.

We evaluated other variant PSOs with different parameter settings. Almost all of metaheuristics are required to set a number of parameters, which might lead to different outcomes, for example, multiple locally optimal solutions in the parameter space in terms of solution quality. An extension of this work includes analyzing the principles of balancing exploitation and exploration of metaheuristics on forecasting. We will focus on the work of the details of comparing the effectiveness of the exploitation or the exploration among

them and analyzing the different concepts or philosophy within them.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by Cuiying Grant of China Telecom, Gansu Branch (Grant no. lzudxcy-2013-3), Science and Technology Planning Project of Chengguan District, Lanzhou (Grant no. 2013-3-1), and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (Grant no. 44th). This work is also partially supported by Fundamental Research Funds for the Central Universities (Grant no. XDJK2014C141 and SWU114005).

References

- [1] X. Q. Liu, B. W. Ang, and T. N. Goh, "Forecasting of electricity consumption: a comparison between an econometric model and a neural network model," in *Proceedings of the 1991 IEEE International Joint Conference on Neural Networks (IJCNN '91)*, pp. 1254–1259, November 1991.
- [2] T. S. Quah and B. Srinivasan, "Improving returns on stock investment through neural network selection," *Expert Systems with Applications*, vol. 17, no. 4, pp. 295–301, 1999.
- [3] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [4] J. Roman and A. Jamee, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," in *Proceedings of the 29th Hawaii International Conference on System Sciences*, vol. 2, pp. 454–460, 1996.
- [5] G. Tkacz, "Neural network forecasting of Canadian GDP growth," *International Journal of Forecasting*, vol. 17, no. 1, pp. 57–69, 2001.
- [6] J. G. de Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [7] W. He, Z. Wang, and H. Jiang, "Model optimizing and feature selecting for support vector regression in time series forecasting," *Neurocomputing*, vol. 72, no. 1–3, pp. 600–611, 2008.
- [8] J. Shen, C. Zhang, C. Lian, H. Hu, and M. Mammadov, "Investment decision model via an improved BP neural network," in *Proceedings of the 2010 IEEE International Conference on Information and Automation (ICIA '10)*, pp. 2092–2096, June 2010.
- [9] A. Kandel, *Fuzzy Expert Systems*, CRC Press, 1992.
- [10] J. Ma and J. F. Teng, "Predict chaotic time-series using unscented kalman filter," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 687–690, August 2004.
- [11] T. C. Jo, "The effect of virtual term generation on the neural based approaches to time series prediction," in *Proceedings of the 4th International Conference on Control and Automation (ICCA '03)*, pp. 516–520, June 2003.
- [12] L. W. Lee, L. H. Wang, and S. M. Chen, "Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques," *Expert Systems with Applications*, vol. 34, no. 1, pp. 328–336, 2008.
- [13] S. M. Chen, N. Y. Wang, and J. S. Pan, "Forecasting enrollments using automatic clustering techniques and fuzzy logical relationships," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11070–11076, 2009.
- [14] H. L. Wong, Y. H. Tu, and C. C. Wang, "Application of fuzzy time series models for forecasting the amount of Taiwan export," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1465–1470, 2010.
- [15] E. Kayacan, B. Ulutas, and O. Kaynak, "Grey system theory-based models in time series prediction," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1784–1789, 2010.
- [16] J. Deng, *Grey Prediction and Decisionmaking*, Huazhong University of Science and Technology Press, Wuhan, China, 1989.
- [17] C. C. Hsu and C. Y. Chen, "Applications of improved grey prediction model for power demand forecasting," *Energy Conversion and Management*, vol. 44, no. 14, pp. 2241–2249, 2003.
- [18] L. C. Hsu, "A genetic algorithm based nonlinear grey Bernoulli model for output forecasting in integrated circuit industry," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4318–4323, 2010.
- [19] L. C. Hsu, "Using improved grey forecasting models to forecast the output of opto-electronics industry," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13879–13885, 2011.
- [20] D. Ju-Long, "Control problem of grey systems," *Systems & Control Letters*, vol. 1, no. 5, pp. 288–294, 1982.
- [21] D. Akay and M. Atak, "Grey prediction with rolling mechanism for electricity demand forecasting of Turkey," *Energy*, vol. 32, no. 9, pp. 1670–1675, 2007.
- [22] H. W. V. Tang and M. S. Yin, "Forecasting performance of grey prediction for education expenditure and school enrollment," *Economics of Education Review*, vol. 31, no. 4, pp. 452–462, 2012.
- [23] Z. Zhao, J. Wang, J. Zhao, and Z. Su, "Using a Grey model optimized by Differential Evolution algorithm to forecast the per capita annual net income of rural households in China," *Omega*, vol. 40, no. 5, pp. 525–532, 2012.
- [24] J. Wang, Y. Dong, J. Wu, R. Mu, and H. Jiang, "Coal production forecast and low carbon policies in China," *Energy Policy*, vol. 39, no. 10, pp. 5970–5979, 2011.
- [25] Z. X. Wang, K. W. Hipel, Q. Wang, and S. W. He, "An optimized NGBM(1,1) model for forecasting the qualified discharge rate of industrial wastewater in China," *Applied Mathematical Modelling*, vol. 35, no. 12, pp. 5524–5532, 2011.
- [26] S. C. Chang, H. C. Lai, and H. C. Yu, "A variable P value rolling Grey forecasting model for Taiwan semiconductor industry production," *Technological Forecasting and Social Change*, vol. 72, no. 5, pp. 623–640, 2005.
- [27] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromechatronics and Human Science*, pp. 39–43, 1995.
- [28] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 685175, 10 pages, 2008.
- [29] G. Tan, "The structure method and application of background value in grey system gm(1,1) model (i)," *Systems Engineering: Theory and Practice*, vol. 4, no. 4, pp. 98–103, 2000.
- [30] X. Hu, Y. Shi, and R. Eberhart, "Recent advances in particle swarm," in *Proceedings of the 2004 Congress on Evolutionary Computation (CEC '04)*, pp. 90–97, June 2004.

- [31] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC '99)*, vol. 3, May 1999.
- [32] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 2001.
- [33] F. van den Bergh, *An analysis of particle swarm optimizers [Ph.D. thesis]*, University of Pretoria, 2002.
- [34] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [35] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, 1998.
- [36] J. T. Yokuma and J. S. Armstrong, "Beyond accuracy: comparison of criteria used to select forecasting methods," *International Journal of Forecasting*, vol. 11, no. 4, pp. 591–597, 1995.
- [37] L. C. Hsu and C. H. Wang, "Forecasting the output of integrated circuit industry using a grey model improved by the Bayesian analysis," *Technological Forecasting and Social Change*, vol. 74, no. 6, pp. 843–853, 2007.
- [38] C. F. Chen, M. C. Lai, and C. C. Yeh, "Forecasting tourism demand based on empirical mode decomposition and neural network," *Knowledge-Based Systems*, vol. 26, pp. 281–287, 2012.