A Two Levels Evolutionary Modeling System For Financial Data

Zhou Kang
Computation Center,
Wuhan University,
Wuhan, 430072, China

Yan Li
Computation Center,
Wuhan University,
Wuhan, 430072, China

Hugo de Garis
Computer Science
Department, Utah State
University, Logan, Utah
84322, USA

Li-Shan Kang
State Key Laboratory of
Software Engineering,
Wuhan University,
Wuhan, 430072, China
kang@whu.edu.cn

Abstract

The discovery of evolutionary laws of financial market is always built on the basis of financial data. Any financial market must be controlled by some basic laws, including macroscopic level, submicroscopic level and microscopic level laws. How to discover its necessity-laws from financial data is the most important task of financial market analysis and prediction. Based on the evolutionary computation, this paper proposes a multi-level and multi-scale evolutionary modeling system which models the macro-behavior of the stock market by ordinary differential equations while models the micro-behavior of the stock market by natural fractals. This system can be used to model and predict the financial data(some time series), such as the stock market data of Dow-Jones index and IBM stock price, and always get good results.

1 INTRODUCTION

The financial data or data get from some real-world systems such as stock market usually are very complex, but any complex system is bound to be controlled by some basic laws, including macroscopic level laws, submicroscopic level laws and microscopic level laws. Consider the following financial data as the time series:

$$x(t_0), x(t_1), \cdots, x(t_m) \qquad (1)$$

where t_i= t_0 + i?t, ?t is the time stepsize. As for the time series, besides the traditional method of time series analysis[1], evolutionary algorithm is usually used to cope with these data[2][3][4].

Suppose that the financial data are controlled by macroscopic, sub-macroscopic and microscopic rules. We take the multi-level, multi-scale models for analyzing and predicting the financial data. In this paper, we use the ordinary differential equation (ODE) model to describe the macroscopic behavior of the stock market, while we use the natural fractal models (a kind of natural discrete wavelets) to describe its microscopic behavior. In this way, we build a multi-level and multi-scale evolutionary modeling system for financial data. This system provides a strong tool for the analysis and prediction of complex time series. The observed data of the Dow-Jones index and IBM stock price are used as the test data for this system.

The rest of the paper is arranged as follows: in section 2, we introduce the macroscopic ODE model; in section 3, we introduce the microscopic natural fractal model; numerical experiments are given in section 4; and in the end, section 5 is some conclusions.

2 MACROSCOPIC ODE MODEL

The complex time series are usually characteristics of multi-level and multi-scale. Assume that it has two levels: macro and micro. In order to describe it in macroscopic level, the researchers take many kinds of methods to pre-handle the time series.

2.1 DECOMPOSITION OF ORIGINAL DATA

In order to find out macroscopic laws from complex data, the first step is to decompose the original data $x(t_i)$, i=0,1,2,...,m as in (1) into two parts: the smooth part and the coarse part (non-smooth part). We assume that the evolutionary process of the smooth part is controlled by macroscopic factors, and the evolutionary process of the coarse part is controlled by microscopic factors. The smooth part will be modeled by ordinary differential equations (ODE), while the coarse part be modeled by natural fractals (a kind of multi-scale discrete wavelets).

For the time series (1), we decompose it into two parts:

$$x(t_i) = \overline{x}(t_i) + \tilde{x}(t_i) \qquad i = 0, 1, \cdots, m \qquad (2)$$

where the smooth part $\overline{x}(t_i)$ is defined as:

$$\overline{x}(t_i) = \begin{cases} \frac{1}{i+1}\sum_{j=0}^{i} x(t_j) & i < l \\ \frac{1}{l+1}\sum_{j=i-l}^{i} x(t_j) & l \le i \le m \end{cases} \qquad (3)$$

and the coarse part:

$$\tilde{x}(t_i) = x(t_i) - \overline{x}(t_i),\ i = 0,1,\cdots,m \qquad (4)$$

Notice: (3) and (4) is related to the value of the smooth parameter l which is often proportional to m, when l is bigger, the time series $\{\overline{x}(t_i)\}$ is more smooth.

2.2 MACROSCOPIC HIGHER-ORDER ODE MODEL

In this subsection, we will mainly introduce how to model and predict the smooth data $\{\overline{x}(t_i)\}_{i=0}^{m}$. Since the smooth data describe the macro behavior of dynamic system and determine the macroscopic tendency of the system, it is the essential part of observed data. Because it is the smooth part of observed data, we assume that $\overline{x}(t)$ is sufficiently smooth, that is, assume that $\overline{x}(t) \in C_n[t_0,T]$, $l=n=4$.

The modeling problem of the dynamic system $\overline{x}(t)$ is to find an initial value problem of the nth–order ordinary differential equation:

$$\begin{cases} x^{(n)}(t) = f(t, x(t), x'(t), x''(t), \cdots, x^{(n-1)}(t)) \\ x^{(i)}(t)\big|_{t=t_0} = x^{(i)}, \quad i = 0,1,\cdots,n-1 \end{cases} \qquad (5)$$

such that the mean square error between the values of its solution $x^*(t)$ at t_i, $i = 0,1,\ldots,m$ and the series $\{\overline{x}(t_i)\}$ as small as possible.

Denote

$$\left\| x^* - \overline{x} \right\| \equiv \frac{1}{m+1}\sqrt{\sum_{i=0}^{m}(x^*(t_i) - \overline{x}(t_i))^2} \qquad (6)$$

That is to say, to find f in function space F, such that

$$\min_{f\in F} \left\| x^* - \overline{x} \right\|. \qquad (7)$$

This problem is solved by evolutionary modeling algorithm described in [7]. The main idea of the algorithm is to embed a genetic algorithm in genetic programming used to discover and optimize the structure of a model, while GA is used to optimize its parameters.

The evolutionary modeling algorithm for higher-order ordinary differential equation can be simply described as follows:

```
PROCEDURE 1
begin
  Initialize population $P(0) = \{p_1(0), p_2(0), \ldots, p_N(0)\}$;
  (produce $N$ parse trees randomly)
  $t := 0$;
  Evaluate the fitness of $p_i(t)$, $i = 1, 2, \ldots, N$;
  while not terminate do
  begin
      $P_c(t)$ := crossover $\{P(t)\}$;
      $P_m(t)$ := mutation $\{P_c(t)\}$;
      Evaluate $P_m(t)$;
      $P(t+1)$ := selection $\{P_m(t), P(t)\}$;
      $t := t + 1$;
  end
  Output solution of $p_{best}$ :$\overline{x}^*(t_i), i = 0,1,\cdots,m+q$;
end
```

For the details of the process of modeling, please refer to [7].

3 MICROSCOPIC FRACTAL MODEL

For the coarse part $\{\tilde{x}(t_i)\}_{i=0}^{m}$ of the time series (1):

$$\tilde{x}(t_i) = x(t_i) - \overline{x}(t_i),\quad i = 0,1,\cdots,m,$$

we are going to build a multi-scale micro natural fractal model.

3.1 CONSTRUCTION OF NATURAL WAVELETS

Denote $$\overline{x} = \sum_{i=0}^{m} \tilde{x}(t_i) \Big/ (m+1) \qquad (8)$$

In order to search an l-scale basic natural wavelet of series (4), we divide the series $\{\tilde{x}(t_i)\}_0^m$ into l groups (from row to column to form the following matrix (see Table 1), each column as a group, including l groups), where xi denotes $\tilde{x}(t_i)$.

Table 1

	1	2	...	S+1	...	l
1	x_0	x_1	...	x_S	...	x_{l-1}
2	x_l	x_{l+1}	...	x_{l+S}	...	x_{2l-1}
⋮						
k	$x_{(k-1)l}$	$x_{(k-1)l+1}$	...	$x_{(k-1)l+S}$		
average	$\overline{x}_1^l$	$\overline{x}_2^l$	...	$\overline{x}_{S+1}^l$	...	$\overline{x}_l^l$

$$\overline{x}_i^l = \left(\sum_{j=1}^{k_i^*} x_{(j-1)l+i-1} \right) \Big/ k_i^* \qquad (9)$$

where

$$k_i^* = \begin{cases} k & i \geq S+1 \\ k-1 & i > S+1 \end{cases} \qquad (10)$$

When $S = l - 1$, $k_i^* = k$, then $(m+1)/ l = k$.

Through the points $(t_{i-1}, \bar{x}_i^l), i = 1, 2, \cdots, l$ in the $x - t$ plane, we can get a polygonal line as follows:

$$x^l(t) = \begin{cases} \dfrac{(t-t_i)\bar{x}_{i+1}^l - (t-t_{i+1})\bar{x}_i^l}{t_{i+1} - t_i} \\ \quad \text{if} \quad t_i \leq t \leq t_{i+1} \, and \quad 0 \leq i \leq l-2 \\ 0 \qquad otherwise \end{cases} \qquad (11)$$

Evidently, the function $x^l(t)$ has a local compact support $[t_0, t_{l-1}]$, we call it l–scale basic natural wavelet.

In order to test whether $x^l(t)$ is a basic natural wavelet of time series (4), we introduce the variance ratio:

$$E_l = \frac{\sum_{i=1}^{l} k_i^* (\bar{x}_i^l - \bar{x})^2 \Big/ (l-1)}{\sum_{i=1}^{l} \sum_{j=1}^{k_i^*} (x_{(j-1)l+i-1} - \bar{x}_i^l)^2 \Big/ (m-l+1)} \qquad (12)$$

Assume that E_l has an F-distribution with (l-1, m-l+1) degrees of freedom. For different confidence level a and (l-1, m-l+1) degrees of freedom, we can get $F_a(l-1, m-l+1)$ from a F-distribution table.

If $E_l = F_a(l-1, m-l+1)$, then the time series (4) exists l–scale basic natural wavelet $x^l(t)$ in confidence level a. If El < $F_a(l-1, m-l+1)$, then the series (4) does not exist l–scale basic natural wavelet in confidence level a.

3.2 MICROSCOPIC NATURAL FRACTAL MODEL

In order to build the mathematical model for the coarse part $\tilde{x}(t)$ of the time series (4), we construct a multi-scale natural fractal model with scale l=2, 3, ..., L. The process can be described as follows:

PROCEDURE 2

```
begin
  initialize $\bar{x} := \tilde{x}$ ; where $\tilde{x} = \{\tilde{x}(t_i)\}_{i=0}^{m}$
  $x^* := 0$ ; where $x^* = \{x^*(t_i)\}_{i=0}^{m+q}$
  for  $l$ =2, L,  do
  using $\{\bar{x}(t_i)\}_{i=0}^{m}$ calculate $\bar{x}^l = \{\bar{x}_1^l, \bar{x}_2^l, \cdots, \bar{x}_l^l\}$;
        if  $E_l = F_a(l-1, m-l+1)$  then
$x^* := x^* + x^l$ ; where $x^l(i) = \bar{x}_{j+1}^l, \ j \equiv i (\mathrm{mod}\ l)$
   end for
   $e := \sum_{i=o}^{m} \frac{\tilde{x}(i) - x^*(i)}{m+1}$ ;
      for  $i = 0, m+q$  do
         $x^*(i) := x^*(i) + e$ ;
      end for
end
```

Remark 1: The output $\{\tilde{x}^*(t_i)\}_{i=0}^{m}$ is the fitting part of $\{\tilde{x}(t_i)\}_{i=0}^{m}$ and $\{\tilde{x}^*(t_i)\}_{i=m+1}^{q}$ is the prediction part of $\tilde{x}(t)$. e is the average fitting error, and it has been eliminated as the correction (random error).

Remark 2: The first part of the procedure is to test successively whether the time series exists basic natural wavelet with scale l which is less than L, usually L=(m+1)/3, for some special problems, where m is relatively small, L can be magnified to L=(m+1)/2. if it exists, then prolong it periodically to whole interval $[t_0, t_{m+q}]$ and add to the time series $\{x^*(i)\}_0^{m+q}$.

Remark 3: The second part of the procedure is to evaluate the random error of $\{\tilde{x}(t_i)\}$, and then correct it when fitting and prediction.

3.3 THE MULTI-LEVEL AND MULTI-SCALE EVOLUTIONARY MODELING SYSTEM

Using the macroscopic modeling PROCEDURE 1 of nth-order ordinary differential equation (5), and the microscopic modeling PROCEDURE 2 of natural fractal, we can build a multi-level and multi-scale evolutionary modeling system for fitting and prediction of complicated time series. Firstly, call PROCEDURE 1 to build the ODE (5), and use Runge-Kutta method to solve it to get the fitting and prediction values of the smooth part $\bar{x}^*(t_i), i = 0, 1, \cdots, m+q$, then call PROCEDURE 2 to get the fitting and prediction values of the coarse part: $\tilde{x}^*(t), i = 0, 1, \cdots, m+q$. Adding up these data, we can get the needed fitting and prediction values:

$$\bar{x}^*(t_i) + \tilde{x}^*(t_i) = x^*(t_i), i = 0, 1, \cdots, m+q$$

This procedure can be described as follows:

PROCEDURE 3

```
begin
Decompose data x [0, m] into $\bar{x}[0, m]$ and $\tilde{x}[0, m]$;
Call PROCEDURE 1 to get $\bar{x}^*[0, m+q]$;
Call PROCEDURE 2 to get $\tilde{x}^*[0, m+q]$;
    for  $i$ =0, $m$  do
        $x^*(t_i) := \bar{x}^*(i) + \tilde{x}^*(i)$;
        $e(i) := x(i) - x^*(t_i)$;
    endfor
    for  $i = m+1, m+q$  do
        $x^*(t_i) := \bar{x}^*(i) + \tilde{x}^*(i)$;
    endfor
    output  $x^*(t_i), i = 0, 1, ..., m+q$;
```

```
        output   $e(i), i = 0,1,..., m;$
end
```

Remark 1: The first step of the procedure is to decompose the original time series into two parts: the smooth part and the coarse(non-smooth) part.

Remark 2: The second step of the procedure is to call PROCEDURE 1 to deal with the smooth data $\{\overline{x}(t)\}$ and get an ODE model of and the values of its solution: $\overline{x}^*(t_0), \overline{x}^*(t_1), \cdots, \overline{x}^*(t_{m+q})$, where the first m+1 values are the fitting values of $\overline{x}(t)$, and the later q values are the prediction values of $\overline{x}(t)$.

Remark 3: The third step of the procedure is to call PROCEDURE 2 to deal with the coarse data $\tilde{x}(t)$ and get a multi-scale natural fractal model and its solution: $\tilde{x}^*(t_0), \tilde{x}^*(t_1), \cdots, \tilde{x}^*(t_{m+q})$, where the first m+1 values are the fitting values of $\tilde{x}(t)$, and the later q values are the prediction values of $\tilde{x}(t)$ at $t_{m+1}, t_{m+2}, ..., t_{m+q}$.

Remark 4: The fourth step of the procedure is to combine the fitting values of the smooth part with those of the coarse part of $x(t)$ to get the time series $\{x^*(t_i)\}_0^m$ and the fitting error $\{e(i)\}_0^m$. The fifth step of the procedure is to combine the prediction values of the smooth part with those of the coarse part of $x(t)$ to get the prediction values of $x(t)$ at the time $t_{m+1}, t_{m+2}, ..., t_{m+q}$, where the prediction length q can be decided by the users.

4 NUMERICAL EXPERIMENTS

In this section, we mainly study the applications of multi-level and multi-scale evolutionary modeling system to the financial data.

Firstly we use the smooth data of BUMP problem as the test data of the smooth model.

$$\text{Maximize } f_n(x) \equiv \frac{\left|\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)\right|}{\sqrt{\sum_{i=1}^{n} i x_i^2}}$$

$$\text{subject to} 0 < x_i < 10, i = 1,2,\cdots,n \;, \quad \prod_{i=1}^{n} x_i >= 0.75 \text{ and } \sum_{i=1}^{n} x_i <= 7.5n$$

4.1 MODELING OF SMOOTH SCIENTIFIC DATA

In 1994, Keane [8] proposed the BUMP problem in optimum structural design as follows:

The solutions of the BUMP problem are unknown. According to this problem, Liu proposed a challenge problem in his doctoral dissertation [9] as follows:

$$\lim_{n\to\infty} Max\ f_n(X) \quad \text{s.t.} \quad 0 \le x_i \le 10,\ 1 \le i \le n,$$

$$\text{where} \quad \prod_{i=1}^{n} x_i >= 0.75 \quad \text{and} \quad \sum_{i=1}^{n} x_i <= 7.5n$$

Table2. Solution table of BUMP problem

n	f_n	**n**	f_n	**n**	f_n
1		18	0.79717388	35	0.82743885
2	0.36497975	19	0.79800887	36	0.82783593
3	0.51578550	20	0.80361910	37	0.82915387
4	0.62228103	21	0.80464587	38	0.82896840
5	0.63444869	22	0.80833226	39	0.83047389
6	0.69386488	23	0.81003656	40	0.82983459
7	0.70495107	24	0.81182640	41	0.83148885
8	0.72762616	25	0.81399253	42	0.83226201
9	0.74126604	26	0.81446495	43	0.83226624
10	0.7473103	27	0.81694692	44	0.83323002
11	0.76105561	28	0.81648731	45	0.83285734
12	0.76256413	29	0.81918437	46	0.83397823
13	0.77333853	30	0.82188436	47	0.83443462
14	0.77726156	31	0.82210164	48	0.83455114
15	0.78244496	32	0.82442369	49	0.8318462
16	0.78787044	33	0.82390233	50	0.83526201
17	0.79150564	34	0.82635733		

Liu got the best solutions of the BUMP problem for n = 2,3,...,50 as showed in Table 2, where f_n = Max f_n (x). The best solutions are depicted in Fig. 1.

We want to discover higher-order ODEs to model the time series f_2, f_3 , f_4 ,..., f_{50}. Denote $f_i = f(t_i)$, where $t_i = t_0 + i?t$, t_0 =2, and ?t = 0.01.

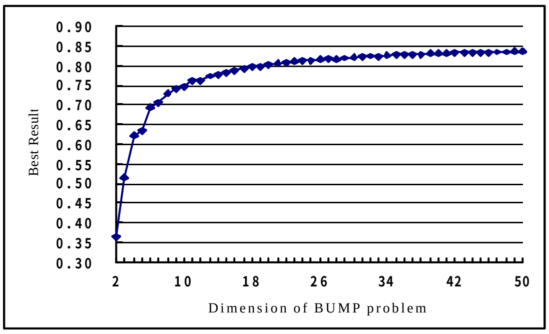

Fig. 1: Best results of f_2 to f_{50}

Using the method described in section 2, we discoverd the following model by computer automatically:

$$d^2 f(t)/dt^2 = \text{-15.658156} \quad df(t)/dt\,(t + df(t)/dt\ )$$

$$f(2) = 0.36497978$$

$$df(t)/dt\,\Big|_{t=2} = 15.08058$$

Where the modeling error is 0.00095677, this means the model fits the solutions of BUMP problems very well. We use it to predict the solution of the challenge problem by using Runge-Kutta method with ? t =0.01 in 1000000 steps, the results f(100), f(200),..., f(1000 000) are shown in Table 3. The results of f100, f200, ..., f1000000 of the BUMP problem[9] got by Liu on a massively parallel computer are compared in Table 3.

Table 3: the Comparison of f_n and $f(n)$

n	f_n	$f(n)$
100	0.8448539	0.8445141
200	0.8468442	0.84503153
300	0.8486441	0.84503450
400	0.8511074	0.84503451
500	0.8504975	0.84503451
1500	0.8449622	0.84503451
10000	0.8456407	0.84503451
20000	0.8455883	0.84503451
100000	0.8448940	0.84503451
1000000	0.8445861	0.84503451

These results show that the smooth model got by the new modeling system gives a good long-range prediction.

4.2 MODELING OF THE DATA OF DOW-JONES INDEX

The observed data shown in Fig.2 are taken from [10] giving the daily Dow-Jones index over 132 days in 2000. We take the observed data of the first 126 days as historical data (training data) to build models which are used to predict the Dow-Jones index of the last 6 days.

Parameter settings of the modeling experiments are m=126, l=4 for smoothing, m=126, q=6, t_0=0,? t = 0.01 (one day), N=100, n=2 (the second-order ODE) for macroscopic ODE model, and m=126, q=6, L=53,a = 0.1 for microscopic natural fractal model. We get a second-order ODE model as follows:

$$\frac{d^2x}{dt^2} = -17228.009766 + \frac{3672.875732 / \sin(\frac{dx}{dt})}{\sin(\cos t * 1115.356812)}$$

The results are shown in Fig.2.

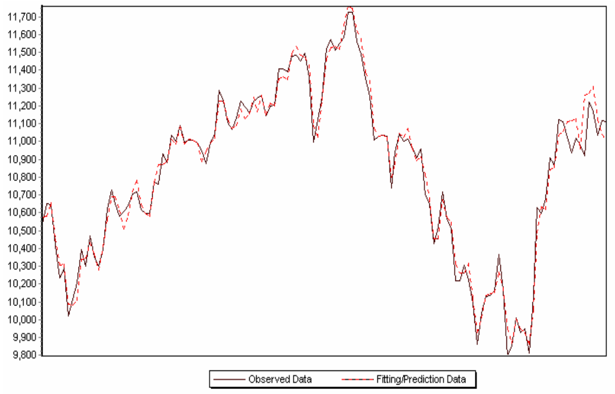

Fig. 2: the fitting and prediction curves for Dow-Jones index

4. 3 MODELING OF IBM STOCK PRICE DATA

The observed data shown in Fig.3 are taken from [10] giving the daily stock price of IBM Company from May 17,1961 to November 2,1962. We take the observed data of the first 359 days as the training data to build models which are used to predict the stock price of the last 10 days.

Parameter settings of the modeling experiments are m=54, l=10 for smoothing, m=359, q=10, t_0=0, ? t=0.01(one day), N=100, n=2 (the second-order ODE) for macroscopic ODE model, and m=359, q=10, L=120,a =0.1 for microscopic natural fractal model. We get a second-order ODE model as follows:

$$\frac{d^2x}{dt^2} = \frac{-1507.55.537}{\cos x} - \cos^2 x$$

The results are shown in Fig.3.

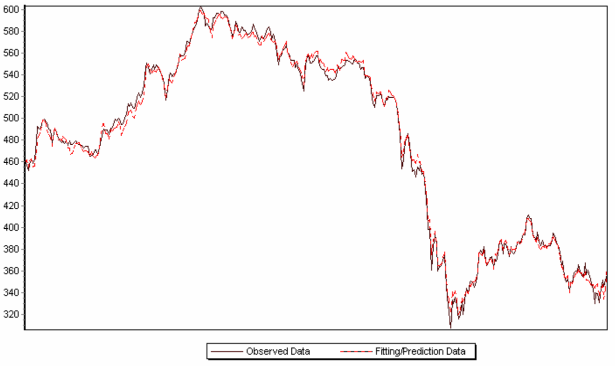

Fig.3: the fitting and prediction curves for IBM stock price

5 CONCLUSION

Compared with most available modeling methods, the multi-level and multi-scale evolutionary modeling system has the following advantages:

Firstly, the entire process is automatic and requires little information in the way of the real-world system or expertise.

Secondly, it allows one to model the macro-behavior of the system by ordinary differential equations and to model the micro-behavior of the system by multi-scale natural fractals simultaneously.

Finally, the models discovered by computers from the complicated financial data can fit the original data quite well, and the structures of the ODE models are unimaginably to humans.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Nos. 70071042, 60073043, 60133010).

References

[1] C.Chatfield, The Analysis of Time Series: An Introduction, Fifth Edition, Chapman & Hall, 1996.

[2] K.R.Vazquez, Genetic programming in the time series modeling: An application to meteorological data, in Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, Korea, 261-266, May 27-30, 2001.

[3] Y.Liu and X.Yao, Evolving neural network for Hang Seng stock index forecast, in Proceedings of the 2001 IEEE congress on Evolutionary Computation, Seoul, Korea, 256-260, May 27-30,2001.

[4] T-C.Fu, F-L.Chung, V.Ng and R.Luk, Evolutionary segmentation of financial time series into subsequences, in Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, Korea, 426-430, May 27-30, 2001.

[5] C.Hafner and J.Frohlich, Generalized function analysis using hybrid evolutionary algorithms, Proceedings of the Congress on Evolutionary Computation, Washington D.C, USA, Vol.1, 287-294, July 6-9, 1999.

[6] L.Kang, Y.Li and Y. Chen, A tentative research on complexity of automatic programming, Wuhan University Journal of Natural Sciences, Vol.6, No.1-2, 59-62, 2001.

[7] H.Cao, L.Kang and Y. Chen, Evolutionary modeling of systems of ordinary differential equations with genetic programming, Genetic Programming and Evolvable Machines, Vol.1, No.4, 309-337, 2000.

[8] Keane,A.J., “Experiences with optimizers in structural design”, in Proc. of the Conf. on Adaptive Computing in Engineering Design and Control 94, ed. Parmee, I.C., Plymouth, 1994, pp. 14 -27.

[9] Liu, P., Evolutionary Algorithms and Their Parallelization, Doctoral Dissertation, Wuhan University, 2000.

[10] R.C. Gan, The Statistical Analysis of Dynamic Data, Beijing, Beijing University of Science and Technology Press, China, 1991.