



# The kinetic evolutionary modeling of complex systems of chemical reactions

Hongqing Cao<sup>a,\*</sup>, Jingxian Yu<sup>b</sup>, Lishan Kang<sup>a</sup>, Yuping Chen<sup>a</sup>,  
Yongyan Chen<sup>b</sup>

<sup>a</sup>State Key Laboratory of Software Engineering, Wuhan University, Wuhan, 430072, People's Republic of China

<sup>b</sup>Institute of Electrochemistry, Department of Chemistry, Wuhan University, Wuhan, 430072, People's Republic of China

Received 10 August 1998; accepted 19 January 1999

---

## Abstract

To overcome the drawbacks of most available methods for kinetic analysis, this paper proposes a hybrid evolutionary modeling algorithm called HEMA to build kinetic models of systems of ordinary differential equations (ODEs) automatically for complex systems of chemical reactions. The main idea of the algorithm is to embed a genetic algorithm (GA) into genetic programming (GP) where GP is employed to optimize the structure of a model, while a GA is employed to optimize its parameters. The experimental results of two chemical reaction systems show that by running the HEMA, the computer can discover the kinetic models automatically which are appropriate for describing the kinetic characteristics of the reacting systems. Those models can not only fit the kinetic data very well, but also give good predictions. © 1999 Elsevier Science Ltd. All rights reserved.

*Keywords:* Kinetic analysis; Complex systems of chemical reactions; Evolutionary modeling; Genetic programming; Genetic algorithms

---

## 1. Introduction

During a chemical engineering production process, in order to optimize the chemical reaction process, to restrain secondary reactions, to maximize the yield, and to improve the safety of operation, chemical engineers often need to carry out kinetic analysis for complex systems of chemical reactions. This is usually considered an arduous task. On one hand, chemical

reactions involve complex stoichiometry and thermodynamics. On the other hand, heterogeneous reactions are concerned with mass diffusion, and the reaction rate is affected by the speed of agitation, by interfacial area, by diffusion coefficients and by many other factors. The traditional method for kinetic analysis is to build exact kinetic models only for simple chemical reactions, based on molecular kinetics. However due to the complexity of chemical reaction processes and the short market window of chemical products, it is usually not worthwhile for producers and developers to spend a lot of time studying the detailed mechanism of complex reactions. Hence, it is necessary to find approximate kinetic models for these complex reaction systems by using fast and effective methods, so as to

---

\* Corresponding author. Tel: +86-27-87682438; fax: +86-27-87654569.

*E-mail addresses:* chq@rjgc.whu.edu.cn (H. Cao), kang@whu.edu.cn (L. Kang)

provide a basis for the optimization of chemical reaction processes in further steps. At present, the most utilized approaches for kinetic analysis are the tendency modeling and approximation methods. The tendency modeling, i.e. the so-called gray model method (Filippi et al., 1986; Deng, 1985), which is based on a phenomenological approach, reduces the complex chemical reactions to simple kinetic equations by considering general laws in chemical reactions such as mass and energy balances, and then estimates a small number of kinetic parameters by using regression, integral methods (Himmelblau et al., 1967) and differential methods (Kennard and Melsen, 1985). Approximation methods, i.e. the so-called “black box” methods (Galvan et al., 1996; Cao et al., 1998), do not take into account the physical characteristics of practical reacting systems in principle but employ approximating functions including polynomials, trigonometric series, splines and so on, to fit experimental data by adjusting the parameters. Nevertheless, people often feel troubled when applying these two approaches to practical problems. Firstly, it is difficult to choose an appropriate kinetic model to describe the reacting system due to the large amount of complex kinetic data. Secondly, the estimation of parameters requires the modeler to have a rich mathematical knowledge and chemical professional skill. The computation process is also usually rather complicated.

Recently, due to the merits of self-adaptation, self-organization, self-learning, intrinsic parallelism and generality, evolutionary algorithms (EAs) have been successfully applied in a wide range of economic, engineering and scientific computations (Goldberg, 1989; Mitchell, 1996). The applications of EAs in chemistry are also very wide (Hibbert, 1993; Lucasius and Kateman, 1994). EAs are adaptive methods for solving computational problems in many fields, which mimic the process of biological evolution and the mechanisms of natural selection and genetic variation. They use suitable codings to represent possible solutions to a problem, and guide the search by using some genetic operators and the principle of “survival of the fittest”. EAs originally consist of three branches, namely genetic algorithms (GAs), evolutionary programming (EP) and evolution strategies (ES). In the 1990s, a new branch called genetic programming (GP), was added to the group which was introduced by John Koza (Koza, 1992, 1994). GP is an extension of John Holland’s GA (Holland, 1975) in which the genetic population consists of computer programs of varying sizes and shapes. In standard GP, computer programs can be represented as parse trees, where a branch node represents an element from a function set, which usually contains some arithmetic operations and elementary functions of at least one argument, and a leaf node represents an element from a terminal set,

which usually contains variables, constants and functions of no arguments. These symbolic programs are subsequently evaluated by running them on a set of “fitness cases”. Fitter programs are selected for recombination to create the next generation by using crossover and mutation. This step is iterated for some number of generations until the termination criterion of the run has been satisfied.

As the kinetic behaviors of most chemical reactions need to be described by a system of ordinary differential equations (ODEs), our research aims to build the kinetic ODEs model automatically based on the kinetic data by using GP. However, when we apply the standard GP to the modeling problem of ODEs, one major problem arises. Since the fitness value of a model depends largely upon the values of its parameters, a model with a favorable structure will have a great probability of being eliminated from the population during the evolution, if the randomly generated parameters are inappropriate. Consequently it is unlikely that we will obtain a highly accurate model for the system. Moreover the evolutionary process can be slow due to the large number of generations needed, as well as suffering from the “premature convergence” phenomenon. Besides, as an ODEs model is composed of multiple differential equations, the representation of single tree in standard GP is no longer appropriate to our problem. In order to overcome these drawbacks, we propose a hybrid evolutionary modeling algorithm called HEMA to build kinetic ODEs models automatically for complex systems of chemical reactions. The main idea of the algorithm is to embed a GA into GP, where GP is employed to optimize the structure of a model, while a GA is employed to optimize its parameters.

This paper is organized as follows. In Section 2, we present the structure of the HEMA and give its detailed descriptions. In Section 3, two examples of chemical reacting systems are used to test the effectiveness of the HEMA, and their experimental results and some discussions are also given here. Finally in Section 4, we give some conclusions.

## 2. HEMA

The HEMA mainly consists of two processes: one is the evolutionary modeling process used to optimize the structure of models based on GP, and the other is the parameter optimization process used to optimize the parameters of a model based on a GA. These two processes, accompanied by the simplification and normalization of the models, and the system prediction, constitute the framework of the HEMA. The structure

of the HEMA can be described in pseudo code as follows:

```

Procedure HEMA;
begin
input the kinetic data;
s:=0;
initialize the ODEs model population P(s) : {evolutionary modeling process begin}
evaluate P(s);
repeat
  simplify and normalize P(s);
  optimized:=Φ;
  for i:=1 to popsize do {popsize is the population size of the model population}
  begin
    if (structure (pi)≠optimized)
    begin {parameter optimization process begins}
      check all the constants contained in pi;
      t:=0;
      initialize the parameter population Q(t);
      evaluate Q(t);
      repeat
        select Q(t+1) from Q(t) according to fitness and selection strategy;
        recombine Q(t+1) by using genetic operators (crossover, mutation and
        reproduction);
        evaluate Q(t+1);
        t:=t+1;
      until termination criterion II;
      replace all the parameters in pi with the best individual in Q(t);
      optimized:=optimized∪{structure (pi)};
    end {parameter optimization process end}
  end
  select P(s+1) from P(s) according to fitness and selection strategy;
  recombine P(s+1) by using genetic operators (crossover, mutation and reproduc-
  tion);
  evaluate P(s+1);
  s:=s+1;
until termination criterion I; {evolutionary modeling process end}
make system prediction based on the best individual in P(s);
end

```

We will give the detailed descriptions of the algorithm in following subsections.

## 2.1. Evolutionary modeling process

### 2.1.1. Representation of ODE models

We extend the representation of single tree in standard GP to a vector of binary trees denoted as  $(T_1, T_2, \dots, T_n)$  where  $n$  is the number of equations in the system of ODEs. For example, an ODEs model with the form of

$$dx_1/dt = x_1 + x_2^2 - 2x_3$$

$$dx_2/dt = x_1 + x_2 + x_3$$

$$dx_3/dt = x_1 + t^3 + 3 \quad (1)$$

can be represented as a vector of binary trees  $(T_1, T_2, T_3)$  illustrated in Fig. 1. The maximum depth per tree is restricted by a constant  $D$ .

### 2.1.2. Fitness evaluation of the individuals in $P(s)$

We first define the norm of a matrix  $A$  where  $A \in R^{m \times n}$  as

$$\|A\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} \quad (2)$$

Suppose a chemical reaction system contains  $n$  correlated variables  $x_1, x_2, \dots, x_n$  which change with time and that a series of observed data collected at the time  $t_i = t_0 + i^* \Delta t$  ( $i = 0, 1, 2, \dots, m$ ) can be written in the following form

$$X = \begin{pmatrix} x_1(t_0) & x_2(t_0) & \dots & x_n(t_0) \\ x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ \vdots & \vdots & & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_n(t_m) \end{pmatrix} \quad (3)$$

where  $t_0$  denotes the starting time, and  $\Delta t$  denotes the interval between two observations. For one individual in the model population,  $p_i$ , written in the form of

$$dx_1/dt = f_1(t, x_1, \dots, x_n)$$

$$dx_2/dt = f_2(t, x_1, \dots, x_n)$$

⋮

$$dx_n/dt = f_n(t, x_1, \dots, x_n) \quad (4)$$

where  $f_i(t, x_1, x_2, \dots, x_n)$  ( $i = 1, 2, \dots, n$ ) is composed of some elementary functions, its fitness value,  $fitness(p_i)$ , can be calculated as follows:

```

Procedure cal_fitness;
begin
  let X* and ΔX be two m×n matrices,
  assign the first row of X to that of X*;
  for i:=2 to m do
  begin
    integrate the system (4) for a step
    with some numerical method by taking
    the (i-1)st row of X as the initial
    conditions;
    assign the solution to the ith row of
    X*;
  end
  ΔX:=X-X*;
  fitness(p_i):=||ΔX||;
end

```

Obviously, the lower the fitness is, the better is the individual. During the evaluation of fitness, we use the modified Euler method with fixed stepsize 0.01 to do the integration.

### 2.1.3. Simplification and normalization of models

The simplification of models simplifies the tree structures of each individual in the model population by replacing subtrees which consist of arithmetic operations between constants. This operation is performed on all individuals in every generation, which affects the number of parameters to be optimized but does not change the fitness of individuals.

The normalization of models adjusts the structure of subtrees in the model whose roots are “+” (plus) or “\*” (multiplication) and whose left branches or right branches are a constant to ensure that the constant always lies on the right of the “+” or the “\*” in the S-expression of the model. This operation is useful to distinguish the model structures correctly so that “a+x” and “x+a” or “a\*x” and “x\*a” will not be regarded as different structures, doing the optimization process redundantly.

### 2.1.4. Selection strategy and genetic operators

We use tournament selection with sample size of 4 in the HEMA. An elitism strategy is also adopted which keeps the best individual of the population in the next generation.

Since each individual is represented as a vector of trees, there are two levels of crossover, namely the vector-level crossover and the tree-level crossover. Consider parent  $a$  denoted as  $(T_1^{(a)}, T_2^{(a)}, \dots, T_n^{(a)})$  and parent  $b$  denoted as  $(T_1^{(b)}, T_2^{(b)}, \dots, T_n^{(b)})$ . The vector-level crossover is performed by selecting one tree from  $T_k^{(a)}$  and  $T_k^{(b)}$  randomly as the tree  $T_k^{(c)}$  of the offspring  $c$  denoted as  $(T_1^{(c)}, T_2^{(c)}, \dots, T_n^{(c)})$  for each  $k$  ( $k = 1, 2, \dots, n$ ). The tree-level crossover performs the following operations on each pair of  $T_k^{(a)}$  and  $T_k^{(b)}$  ( $k = 1, 2, \dots, n$ ): randomly select a node within each tree as a crossover point, swap the subtrees rooted at the crossover points and produce two new trees, then use either of them as the tree  $T_k^{(c)}$  of offspring  $c$  denoted as  $(T_1^{(c)}, T_2^{(c)}, \dots, T_n^{(c)})$  on condition that its maximum depth does not exceed  $D$ .

We use the deterministic dynamic adaptation method (Robert et al., 1997) to alter the mutation rate of individuals according to fitness so that the fitter individuals change only in a small range but the less fit individuals in a wide range. Given parent  $i$  denoted as  $(T_1^{(i)}, T_2^{(i)}, \dots, T_n^{(i)})$ , its mutation rate  $p_m(i)$  is defined as

$$p_m(i) = 0.3 * (1 - f_{\min} / f_i) \quad (5)$$

where  $f_{\min}$  is the fitness value of the best individual in the current generation and  $f_i$  is the fitness value of  $i$ . The mutation of  $i$  begins by randomly selecting a  $k$  ( $1 \leq k \leq n$ ), and performs the following operations on the tree  $T_k^{(i)}$ : randomly select a node within the tree as the mutation point, replace the subtree rooted at the mu-

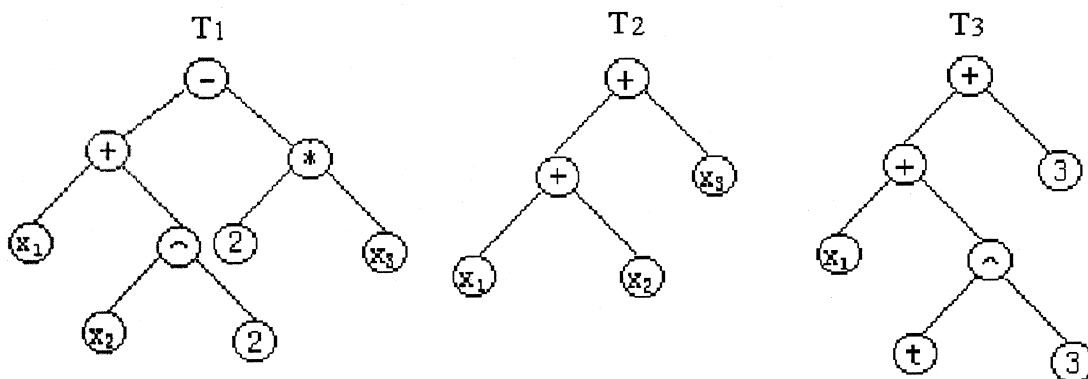


Fig. 1. An example of the representation of an ODEs model.

tation point with a tree randomly generated, thus producing a new tree  $T_k^{*(i)}$ . Then the offspring can be written in the form of  $(T_1^{(i)}, T_2^{(i)}, \dots, T_k^{*(i)}, \dots, T_n^{(i)})$ .

2.2. Parameter optimization process

2.2.1. Representation of the parameters

At the beginning of this process, we examine whether the parameters of the model structure have been optimized in the current generation. If so, do nothing with it. Otherwise, we check all the constants contained in each  $T_i (1 \leq i \leq n)$  first, namely count their number  $l_i$  and record their positions in the tree. Then each individual in the parameter population can be represented as a vector of  $n$  row vectors with varying dimensions of  $l_i$ :

$$(C_1, C_2, \dots, C_n)^T \quad \text{where } C_i = (c_{i1}, c_{i2}, \dots, c_{il_i}) \quad (6)$$

and each element  $c_{ij}$  of vector  $C_i$  is encoded as a floating point number.

2.2.2. Fitness evaluation of the individuals in  $Q(t)$

Consider one individual in the parameter population denoted as  $(C_1, C_2, \dots, C_n)^T$ . Before the evaluation of its fitness, we shall return to the original model denoted as  $(T_1, T_2, \dots, T_n)$  and replace all the constants of each tree  $T_i (1 \leq i \leq n)$  with the corresponding elements of vector  $C_i$ . Then follow the same procedure as in Section 2.1.2. to calculate its fitness value.

2.2.3. Selection strategy and genetic operators

We adopt the same selection strategy as Section 2.1.4. Consider parent  $a$  denoted as  $(C_1^{(a)}, C_2^{(a)}, \dots, C_n^{(a)})^T$  where  $C_i^{(a)} = (c_{i1}^{(a)}, c_{i2}^{(a)}, \dots, c_{il_i}^{(a)})$  and parent  $b$  denoted as  $(C_1^{(b)}, C_2^{(b)}, \dots, C_n^{(b)})^T$  where  $C_i^{(b)} = (c_{i1}^{(b)}, c_{i2}^{(b)}, \dots, c_{il_i}^{(b)})$ . The crossover performs the following

operations on each pair of  $C_i^{(a)}$  and  $C_i^{(b)}$  ( $i=1, 2, \dots, n$ ) based on the integrated arithmetical crossover: randomly select a crossover point  $k (1 \leq k \leq l_i)$  first, then produce two new elements  $c_{ik}^{(c)}$  and  $c_{ik}^{(d)}$  from  $c_{ik}^{(a)}$  and  $c_{ik}^{(b)}$  as follows:

$$c_{ik}^{(c)} = \alpha c_{ik}^{(a)} + (1 - \alpha) c_{ik}^{(b)}$$

$$c_{ik}^{(d)} = \alpha c_{ik}^{(b)} + (1 - \alpha) c_{ik}^{(a)} \quad (7)$$

where  $\alpha$  is a random number ranging from 0 to 1. Thus offspring  $c$  and offspring  $d$  can be written in the form of

$$c: (C_1^{(c)}, C_2^{(c)}, \dots, C_n^{(c)})^T \quad \text{where}$$

$$C_i^{(c)} = (c_{i1}^{(a)}, c_{i2}^{(a)}, \dots, c_{ik}^{(c)}, \dots, c_{il_i}^{(a)})$$

$$d: (C_1^{(d)}, C_2^{(d)}, \dots, C_n^{(d)})^T \quad \text{where}$$

$$C_i^{(d)} = (c_{i1}^{(b)}, c_{i2}^{(b)}, \dots, c_{ik}^{(d)}, \dots, c_{il_i}^{(b)})$$

Consider a parent denoted as  $(C_1, C_2, \dots, C_n)^T$  where  $C_i = (c_{i1}, c_{i2}, \dots, c_{il_i}) (1 \leq i \leq n)$ . The mutation performs the following operations on each  $C_i$  based on multi-level mutation (Mühlenbein et al., 1991; Mühlenbein and Schlierkamp-rose, 1993): randomly select a mutation point  $k (1 \leq k \leq l_i)$  and produce a new element  $c_{ik}^*$  from  $c_{ik}$  as follows:

$$c_{ik}^* = c_{ik} \pm 0.1 \times 2^{-j} \times (\max - \min) \quad (8)$$

where  $j$  is a random integer in the range of 0–15, [min, max] is the field of definitions of  $c_{ik}$ , “+” or “-” is chosen randomly with equal probability. Then the off-

spring can be written as the form of  $(C_1^*, C_2^*, \dots, C_n^*)^T$  where  $C_i^* = (i \ c_{i1}, c_{i2}, \dots, c_{ik}^*, \dots, c_{in})$ .

### 2.3. System prediction

Once the best evolved model is obtained in one run, we then take the last line of  $X$  as the initial conditions, and integrate the system of ODEs with the form of Eq. (4) for several steps with the modified Euler method to get the predicted series of the system.

## 3. Computational experiments

### 3.1. Parameter settings and measures

To examine the effectiveness of the HEMA, we apply it to two chemical reaction systems with different numbers of variables and build the kinetic models of systems of ODEs for them. Twenty runs are conducted independently for each example and the best model is shown and discussed. All the experiments are performed on a Pentium II (266 MHz) using Visual C++ compilers. The parameter settings are as follows:

- For the evolutionary modeling process we use the function set  $F = \{+, -, *, /, ^, \sin, \cos, \exp, \ln\}$  where  $x^k$  symbolizes  $x^k (0 < k < 5)$ , and the terminal set  $T = \{x_1, \dots, x_n, t, c\}$  where  $n$  is the number of equations in the system of ODEs and  $c$  is a set of random constants, a population size of 50, a maximum tree depth of 3 and a maximum of 50 generations per run.
- For the parameter optimization process we use a population size of 20, a 60% crossover rate, a 30% mutation rate and a 10% reproduction rate, and the termination criterion whereby the fitness value of the best individual remains unchanged for three generations.

In addition, to measure the modeling results, we define the average fitting error (AFE) and the average prediction error (APE) of variable  $X$  contained in a chemical reacting system as

$$AFE = \frac{1}{m} \sqrt{\sum_{i=1}^m (\hat{x}_i - x_i)^2}$$

$$APE = \frac{1}{n} \sqrt{\sum_{i=m+1}^{m+n} (\hat{x}_i - x_i)^2} \quad (9)$$

where  $x_i$  denotes the observed value or simulated value of  $X$ ,  $\hat{x}_i$  denotes the fitting value and the predicted value of the ODEs model for AFE and APE, respect-

ively, and  $m$  and  $n$  are the number of data to be fitted and to be predicted, respectively.

### 3.2. Example I

The thermolysis reaction of chloro-cyclohexane at 641 K is as follows:



This is a first-order reaction. Swinbourne (1958) studied the reaction by measuring the total intensity of pressure (TIOP) of the reaction system and recording the measured values (every 3 min) within 48 min. We now take the 16 points as samples to build kinetic models. Meanwhile, we calculate the TIOP of the next four time steps according to the reaction rate constant  $k = 5.31 \times 10^{-4} \text{ s}^{-1}$  (Moore and Pearson, 1981) which we take as test samples to compare with the predicted values of models.

The best kinetic model in 20 runs we obtained by running the HEMA is

$$dP_t/dt = \exp(8.340384 - t) - (P_t * 9.688192) \quad (11)$$

which can be simplified as

$$dP_t/dt = \exp(8.340384 - t) - 9.688192 * P_t \quad (12)$$

Its modeling results are shown in Table 1 and its fitting and prediction curves are illustrated in Fig. 2.

As can be seen from Fig. 2, almost all the fitting points lie on the measured curves exactly with a small APE of 0.029654 torr. From Table 1, we see that the evolutionary model can also make reasonable predictions for the next four unseen TIOP with a rather small APE of 0.161525 torr. It shows that the kinetic model built by running the HEMA is capable of reflecting the kinetic characteristics of the thermolysis reaction of chloro-cyclohexane at 641 K. In addition, it should be noted that the evolutionary model is a complex linear differential equation which contains an exponential function in the expression. Such a complex

Table 1  
Modeling results of the best evolutionary model in 20 runs for example I

Time (min)	Calculated value (torr)	Predicted value (torr)
51	389.2	389.075500
54	393.2	392.866058
57	396.7	396.307190
60	399.8	399.431061
AFE (torr)		0.029654
APE (torr)		0.161525

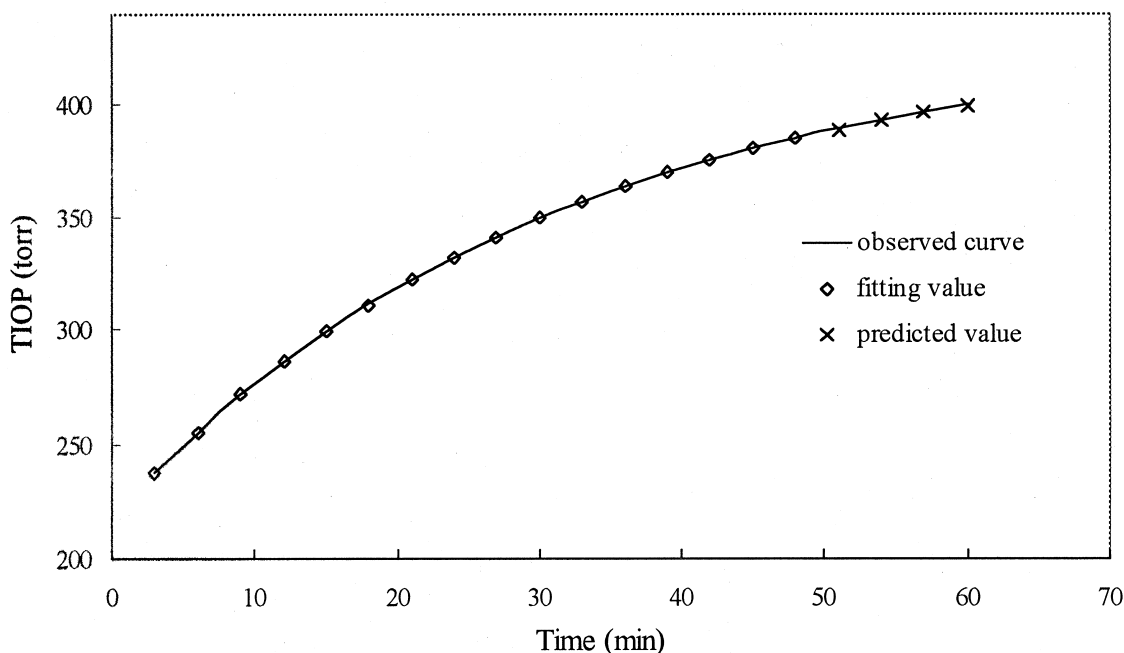
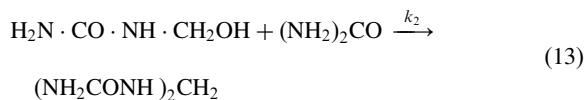
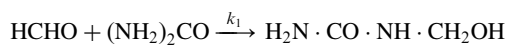


Fig. 2. The fitting and prediction curves of the best evolutionary model for example I.

model structure is usually unimaginable to human minds but can be found easily by the computer by running the HEMA.

### 3.3. Example II

The reaction between formaldehyde ( $X_1$ ) and carbamide in the aqueous solution gives methylol urea ( $X_2$ ) which continues to react with carbamide and form methylene urea ( $X_3$ ). The reaction equation (Moore and Pearson, 1981) is



The reactions occur at 308.15 K and under the excessive carbamide with the concentration ( $c_{2(0)}$ ) of  $2 \text{ mol dm}^{-3}$ . The concentration of chlorohydric acid as a catalyst is  $0.0008 \text{ mol dm}^{-3}$  and the initial concentration of formaldehyde ( $X_{1(0)}$ ) is  $0.1 \text{ mol dm}^{-3}$ . As a kind of typical consecutive reaction, the concentrations of the three components in the system satisfy the following system of ODEs:

$$\frac{dX_1}{dt} = -k'_1 X_1$$

$$\frac{dX_2}{dt} = -k'_1 X_1 - k'_2 X_2$$

$$\frac{dX_3}{dt} = -k'_2 X_2 \quad (14)$$

where  $k'_1 = k_1 c_{2(0)}$ ,  $k'_2 = k_2 c_{2(0)}$  with  $k_1 = 0.007 \text{ dm}^3 \text{ mol}^{-1} \text{ min}^{-1}$ ,  $k_2 = 0.021 \text{ dm}^3 \text{ mol}^{-1} \text{ min}^{-1}$ . According to the exact solution of the consecutive reaction

$$X_1 = X_{1(0)} e^{-k'_1 t}$$

$$X_2 = \frac{k'_1 X_{1(0)}}{k'_2 - k'_1} (e^{-k'_1 t} - e^{-k'_2 t})$$

$$X_3 = X_{1(0)} - X_1 - X_2 \quad (15)$$

we calculate the concentrations of  $X_1$ ,  $X_2$ ,  $X_3$  every other minute within 110 min after the reactions occur and take them as simulated data of our experiment. From them, the first 100 points are used as modeling samples and the next 10 points are used as test samples to evaluate the prediction results of the model.

The best kinetic model we have obtained in 20 runs is

$$\frac{dX_1}{dt} = (X_1 - (X_1 + 1.40035)) * X_1$$

Table 2

Modeling results of the best evolutionary model in 20 runs for example II

Time (min)	$X_1$ (mol dm <sup>-3</sup> )		$X_2$ (mol dm <sup>-3</sup> )		$X_3$ (mol dm <sup>-3</sup> )	
	Simulated value	Predicted value	Simulated value	Predicted value	Simulated value	Predicted value
100	0.024660	0.024659	0.011580	0.011605	0.063760	0.063766
101	0.024317	0.024317	0.011439	0.011489	0.064244	0.064256
102	0.023979	0.023978	0.011300	0.011373	0.064721	0.064742
103	0.023645	0.023645	0.011162	0.011258	0.065193	0.065223
104	0.023317	0.022316	0.011025	0.011143	0.065659	0.065699
105	0.022993	0.022992	0.010889	0.011029	0.066119	0.066171
106	0.022673	0.022673	0.010754	0.010916	0.066573	0.066638
107	0.022358	0.022357	0.010620	0.010804	0.067022	0.067100
108	0.022047	0.022047	0.010488	0.010693	0.067466	0.067558
109	0.021740	0.021740	0.010356	0.010582	0.067903	0.068011
<i>AFE</i>	0.00000004		0.0000018		0.00000055	
<i>APE</i>	0.0000002		0.0000452		0.0000191	

$$dX_2/dt = (X_2 * 3.307096 - (X_1 + t)) * (-1.355543)$$

$$dX_3/dt = 4.069420X_2 + t - 0.002812 \quad (17)$$

$$dX_3/dt = ((X_2 * 4.069420) + t) + (-0.002812) \quad (16)$$

which can be simplified as

$$dX_1/dt = -1.400035X_1$$

$$dX_2/dt = 1.355543(X_1 + t) - 4.482911X_2$$

Its modeling results are listed in Table 2 and its fitting and prediction curves are shown in Fig. 3.

As shown in Fig. 3, the fitting curves of the best evolutionary model and the simulated curves with respect to the concentrations of three components are nearly coincident and more surprisingly, the predicted values also coincide very well with the simulated values. This can be observed from Table 2 clearly. The

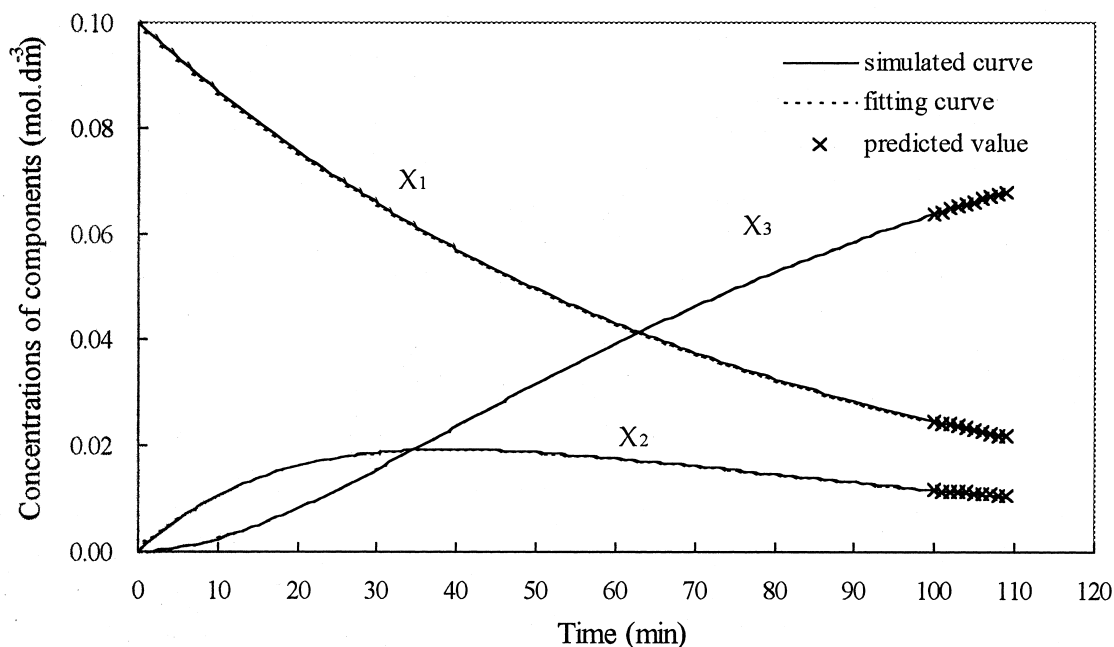


Fig. 3. The fitting and prediction curves of the best evolutionary model for example II.

difference between the simulated value and the predicted value for  $X_1$ ,  $X_2$  or  $X_3$  appears only in the fourth digit after the decimal point. This demonstrates the effectiveness of the HEMA in modeling the complex systems of chemical reactions by system of ODEs. Additionally, by comparing the exact model, Eq. (14), after the replacement of  $k_1'$ ,  $k_2'$  with corresponding values and the evolutionary model, Eq. (17), one notices that their structures are rather similar. As for the equation  $dX_1/dt$ , they have the same structure in essence despite the difference of 100 times between their coefficients due to the fact that we use the modified Euler method with the stepsize 0.01 to do the numerical integration whose time unit is 1% of the exact model. As for the equations  $dX_2/dt$  and  $dX_3/dt$ , they contain the same variables  $X_1$ ,  $X_2$  and the corresponding coefficients are almost identical (likewise, there is a difference of 100 times) except that the evolutionary model has some additional  $t$  item and constant item.

#### 4. Conclusions

To overcome the drawbacks of most available methods for kinetic analysis, this paper proposes a hybrid evolutionary modeling algorithm called HEMA to build kinetic models of systems of ordinary differential equations (ODEs) automatically for complex systems of chemical reactions. The main idea of the algorithm is to embed a genetic algorithm (GA) into genetic programming (GP) where GP is employed to optimize the structure of a model, while a GA is employed to optimize its parameters.

Compared with most available methods for kinetic analysis, this algorithm has the following advantages: (1) the optimization of the model structure and the optimization of the parameters of the model can be performed simultaneously by using the two main processes in the HEMA; (2) some complex kinetic models can be built automatically, whose structures are usually unimaginable to human minds; (3) the algorithm depends very little on domain details and chemical professional expertise. The whole modeling process occurs automatically.

The effectiveness of the algorithm was tested on two chemical reaction systems. The results show that the kinetic model built by using the HEMA can generate satisfactory fitting and prediction values. We expect that the HEMA will prove to be a new and powerful tool for kinetic analysis of complex systems of chemical reactions.

#### Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (no. 69635030) and National 863 High Technology Project of China. The authors would like to thank the anonymous referees for their helpful comments on the paper and to thank Dr Hugo de Garis for his efforts in improving the paper's English and style.

#### References

- Cao, H-Q., Yu, J-X., Kang, L-S., 1998. *Comput. Appl. Chem* 15 (1), 19–22 (in Chinese).
- Deng, J-L., 1985. *Grey Control Systems*. Huazhong University of Science and Technology Press, Wuhan (in Chinese).
- Filippi, C., Greffe, J.L., Border, J., Villermaux, J., Barnay, J.L., Bonte, P., Georgakis, C., 1986. *Chem. Engng Sci* 41, 913–920.
- Galvan, I.M., Zaldivar, J.M., Hernandez, H., Molga, E., 1996. *Comput. Chem. Engng* 20 (12), 1451–1465.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, MA.
- Hibbert, D.B., 1993. *Chemometric Intelligent Lab. Syst* 19, 277.
- Himmelblau, D.M., Jones, C.R., Bischoff, K.R., 1967. *Ind. Engng Chem., Process Des. Dev* 6, 539–543.
- Holland, J.H., 1975. *Adaptation in Natural and Artificial System*. University of Michigan Press, Ann Arbor, MI.
- Kennard, M.L., Melsen, A., 1985. *Ind. Engng Chem., Fund* 24, 129–140.
- Koza, J.R., 1992. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.
- Koza, J.R., 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA.
- Lucasius, C.B., Kateman, G., 1994. *Chemometric Intelligent Lab. Syst* 25, 99.
- Mitchell, M., 1996. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.
- Moore, J.W., Pearson, R.G., 1981. *Kinetic and Mechanism: A Study of Homogeneous Chemical Reactions*, Third. Wiley, New York.
- Mühlenbein, H., Schomisch, M., Born, J., 1991. *Parallel Comput* 17, 619–632.
- Mühlenbein, H., Schlierkamp-rose, D., 1993. *Evolutionary Comput* 1 (1), 25–49.
- Robert, H., Michalewicz, Z., Agoston, E., 1997. *Proceedings of 1997 International Conference on Evolutionary Computation*, pp. 65–69.
- Swinbourne, E.S., 1958. *Aust. J. Chem* 11, 314.