Applied Mathematics, 2019, 10, 627-645
http://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Inference of General Mass Action-Based State Equations for Oscillatory Biochemical Reaction Systems Using *k*-Step Genetic Programming

Tatsuya Sekiguchi[1,2], Hiroyuki Hamada[2,3], Masahiro Okamoto[2,3]

[1]Department of Life Sciences and Informatics, Faculty of Engineering, Maebashi Institute of Technology, Maebashi, Japan
[2]Synthetic Systems Biology Research Center, Kyushu University, Fukuoka, Japan
[3]Graduate School of Systems Life Sciences, Kyushu University, Fukuoka, Japan
Email: okahon@brs.kyushu-u.ac.jp

How to cite this paper: Sekiguchi, T., Hamada, H. and Okamoto, M. (2019) Inference of General Mass Action-Based State Equations for Oscillatory Biochemical Reaction Systems Using *k*-Step Genetic Programming. *Applied Mathematics*, **10**, 627-645.
https://doi.org/10.4236/am.2019.108045

Received: May 14, 2019
Accepted: July 29, 2019
Published: August 1, 2019

Abstract

Systems biology requires the development of algorithms that use omics data to infer interaction networks among biomolecules working within an organism. One major type of evolutionary algorithm, genetic programming (GP), is useful for its high heuristic ability as a search method for obtaining suitable solutions expressed as tree structures. However, because GP determines the values of parameters such as coefficients by random values, it is difficult to apply in the inference of state equations that describe oscillatory biochemical reaction systems with high nonlinearity. Accordingly, in this study, we propose a new GP procedure called "*k*-step GP" intended for inferring the state equations of oscillatory biochemical reaction systems. The *k*-step GP procedure consists of two algorithms: 1) Parameter optimization using the modified Powell method—after genetic operations such as crossover and mutation, the values of parameters such as coefficients are optimized by applying the modified Powell method with secondary convergence. 2) GP using divided learning data—to improve the inference efficiency, imposes perturbations through the addition of learning data at various intervals and adaptations to these changes result in state equations with higher fitness. We are confident that *k*-step GP is an algorithm that is particularly well suited to inferring state equations for oscillatory biochemical reaction systems and contributes to solving inverse problems in systems biology.

Keywords

Systems Biology, Genetic Programming, Inverse Problems, Oscillatory Biochemical Reaction Systems, GMA-Based State Equations

1. Introduction

In recent years, the development of experimental technologies has enabled researchers to obtain various types of omics data. Consequently, functional analyses of complicated, large-scale metabolic pathways have been and will continue to be conducted more and more routinely. This trend is based on the conclusion that even when all of the individual biochemical reactions are analyzed, it is not always possible to predict the function of all the interaction networks among the biomolecules working as a complete system. We refer to this system as a "biochemical reaction system". Henceforth, systems biology [1] [2], which aims to understand biochemical reaction systems, will increasingly occupy an important position.

One major aim of systems biology is to comprehensively analyze omics data and to elucidate biochemical reaction systems. Biochemical reaction systems can be expressed as state equations, most of which are represented by simultaneous differential equations of interrelated biomolecules. In conventional research, the structure of state equations have been fixed and only inferring of their parameter values has been performed [3]-[14]. But in the case where the structure of state equations is unknown because of uncertain information on the interaction networks among the biomolecules, it is necessary to infer the structure from only experimentally observed time-series data. In other words, state equations (*i.e.*, both structures and parameter values) of biochemical reaction systems must be inferred from empirical results, a process which can be referred to as an inverse problem. To overcome this inverse problem encountered in systems biology, researchers predict and scrutinize unknown biochemical reaction systems from experimentally observed time-series data under various conditions. Following this, they formulate the state equations that are expressed by using simultaneous ordinary differential equations based on the general mass action (GMA) law. This method requires a great deal of trial and error, empirical guesswork, and labor. Thus, it would be invaluable to establish a method to infer state equations for biochemical reaction systems using computational science. When inferring state equations capable of reproducing experimentally observed time-series data, multiple state equations may reproduce similar dynamic behaviors. Thus, these potential state equations should be narrowed down by biologically verifying their validity. To infer undiscovered biochemical reaction systems, it is therefore essential to develop an algorithm with high heuristic ability to obtain multiple likely state equations in a short time. Moreover, obtaining various state equations can also lead to the discovery of new biochemical reaction systems that have not yet been considered.

Various solution methods of inverse problems in systems biology have been reported [3]-[22]. Notably, genetic programming (GP) [23] is an evolutionary algorithm that has high heuristic ability as a search method for obtaining suitable solutions expressed as tree structures. GP is widely used in systems biology; for example, Miyahara and Kuboyama [24] reported the determination of glycan motifs by using GP. GP can also be adapted to the inference of state equations expressed as tree structures. Importantly, it is unnecessary to fix the structure of the state equations in advance. In other words, GP requires information about neither the biomolecules that constitute various system components nor their reaction mechanisms. Consequently, GP can obtain various state equations that are able to reproduce experimentally observed time-series data [25]. In GP, since the values of parameters such as coefficients are set to random values, they do not necessarily take optimal values. On the other hand, biochemical reaction systems are highly nonlinear, and the values of these parameters exert strong influences on the dynamic behavior of these systems (*i.e.*, the predicted time-series data). Thus, it is essential to develop an algorithm that can optimize the values of these parameters after inferring the structure of state equations using GP. Ando *et al.* [26], Sugimoto *et al.* [27] and Iba [28] reported a method for inferring the structures of state equations by using GP and optimizing the values of the parameters with the least squares method. However, the authors showed only examples in which monotonically increasing or monotonically decreasing biochemical reaction systems were inferred. Since highly stable biochemical reaction systems often show oscillatory dynamic behaviors, it is essential the development of an algorithm that can infer the state equations of oscillatory biochemical reaction systems.

Accordingly, in this study, we propose a new GP methodology, called "*k*-step GP", for inferring state equations of oscillatory biochemical reaction systems. The *k*-step GP procedure consists of two algorithms: 1) Parameter optimization using the modified Powell method—after genetic operations such as crossover and mutation, the values of parameters such as coefficients are optimized by applying the modified Powell method with secondary convergence. 2) GP using divided learning data—to improve the inference efficiency, imposes perturbations through the addition of learning data at various intervals and adaptations to these changes result in state equations with higher fitness.

2. Proposed *k*-Step Genetic Programming Method

2.1. Data Structure for Genetic Programming

State equations describing biochemical reaction systems often use simultaneous ordinary differential equations based on GMA. State equations based on GMA can be written as

$$\frac{\mathrm{d}X_i}{\mathrm{d}t} = \sum_{k=1}^{p} \alpha_{ik} \prod_{j=1}^{n} X_j^{g_{ijk}} - \sum_{k=1}^{q} \beta_{ik} \prod_{j=1}^{n} X_j^{h_{ijk}}, \tag{1}$$

where n is the number of the system components (*i.e.*, biomolecules), X_i is the state variable for the system components, α_{ik} and β_{ik} are reaction rate constants, p and q are the numbers of generation and decomposition terms, respectively, and g_{ijk} and h_{ijk} are the interaction coefficients. The state equations based on GMA can describe the biochemical reaction systems in detail, and these state equations can also be obtained by intuitive understanding.

Based on Equation (1), it is possible to infer state equations that can reproduce experimentally observed time-series data. Each individual in the GP algorithm is a set of simultaneous ordinary differential equations that is equal in number to the number of state variables. The genotype of each individual is the set of simultaneous ordinary differential equations represented by the tree structure, and the phenotype of each individual is the time-series data calculated using these equations. We have defined the terminal and nonterminal symbols, respectively, in the tree structure data for expressing Equation (1) as follows:

$$\begin{aligned} &T = \{aX_1, aX_2, \cdots, aX_n\} \\ &F = \{+, \times\}. \end{aligned} \tag{2}$$

Thus, all terminal symbols contain the coefficient *a* for expressing tree structure data based on Equation (1). Furthermore, nonterminal symbols need not contain a minus symbol because the coefficient *a* is contained in all terminal symbols and the optimum value of each coefficient *a* (including a negative value) is calculated using the modified Powell method [29].

After generating new individuals (*i.e.*, offspring) by genetic operations such as crossover and mutation, the tree structure data are optimized by the following rules.

- If a term aX_n is added to the same term aX_n (where both terms have the same *n*), then merge the two terms into one term (*i.e.*, $aX_n + aX_n \to aX_n$).
- If a term aX_n is multiplied by a term aX_m, then the value of coefficient *a* of the multiplied term is replaced with 1 (*i.e.*, $aX_n \times aX_m \to aX_n \times X_m$).

For example, the tree structure data of the state equations shown in Equation (3) below are rendered in **Figure 1**.

$$\begin{cases} \dfrac{\mathrm{d}X_1}{\mathrm{d}t} = aX_1X_2 + aX_1 \\ \dfrac{\mathrm{d}X_2}{\mathrm{d}t} = aX_2^2 + aX_1 + aX_2 \end{cases} \tag{3}$$

2.2. Modified Powell Method

After optimizing the tree structure data described in the previous section, the modified Powell method is applied to optimize (decide) the value of the coefficient of the term aX_n. The modified Powell method is well known to have an ultimate fast convergence among various direct search methods without the calculation of the derivative of the objective function.

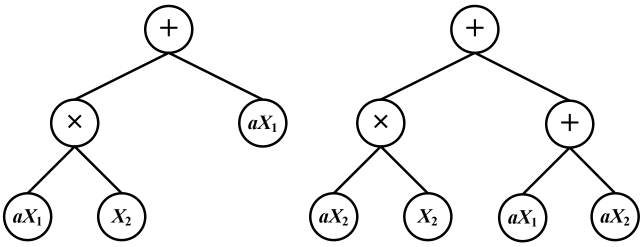


Figure 1. Tree structure data for Equation (3).

2.3. Fitness

For the purpose of evaluating each individual, we have calculated the average value of the relative error between the time-series data obtained by calculating state equations expressed for each individual and the experimentally observed time-series data using the equation

$$S=\sum_{i=1}^{L}\sum_{j=1}^{M}\sum_{t=1}^{N}\left(\frac{X_{cal\,i,j,t}-X_{\exp i,j,t}}{X_{exp\,i,j,t}}\right)^2\Bigg/(LMN), \tag{4}$$

where L is the number of experimentally observed time-series data sets, M is the number of state variables describing system components, and N is the number of observation points for each experimentally observed time-series dataset.

The fitness f of each individual is calculated using the equation

$$f=\frac{1}{1+S+\delta m} \tag{5}$$

where δ is the penalty coefficient and m is the number of terms contained in each individual. The introduction of the penalty coefficient is based on the minimum description length (MDL) principle [30] [31], which is commonly used in GP. The MDL principle can also be used to evaluate the length of the description of the model itself. By including the penalty coefficient in the calculation of the fitness, the simpler state equations have higher fitness. The value of δ needs to be determined according the number of state variables. In this study, we have empirically determined the value of δ.

2.4. Genetic Programming Using Divided Learning Data

GP is a type of algorithm that mimics the evolutionary process of biological systems. Environmental changes (*i.e.*, perturbations) exert strong influences on the evolution of such biological systems. When a stable biological species A is perturbed, it adapts to the perturbation and becomes biological species A', which has newly acquired functions. In this way, biological systems evolve through repeated adaptations to perturbations. In inferring state equations by GP, the learning data can be considered the environment to which adaptation occurs. In standard GP, all learning data are only given at the beginning, and subsequent changes in the data environment are not taken into consideration. In *k*-step GP, the learning data are divided and gradually change in accordance with the progress of the inference process. Accordingly, these changing data become the perturbations to GP individuals, which then collectively adapt to the perturbations and evolve, thus leading to the inference of state equations that can reproduce the learning data. In *k*-step GP, individuals evolve into better individuals by repeating the changing learning data as perturbations and by adapting to them. Consequently, *k*-step GP is expected to achieve improved inference efficiency.

2.5. The *k*-Step GP Procedure

The *k*-step GP procedure is shown below, and a flowchart of this procedure is illustrated in Figure 2.

1) Before processing, the learning data are divided. Length of learning dataset

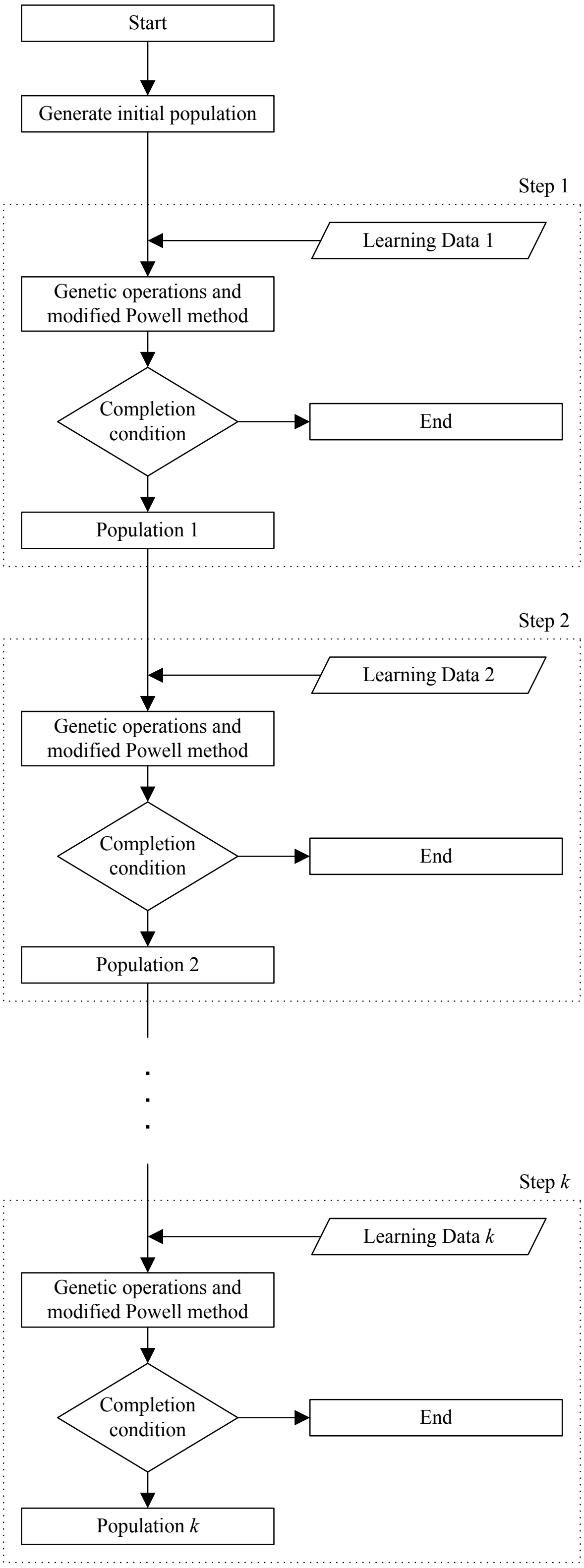


Figure 2. Flowchart of k-step GP.

1 starts from the initial values to the first dividing point; length of learning dataset 2 is starts from the initial value to the second dividing point; that is, length of learning dataset k starts from the initial value to the *k*-th dividing point. Learning dataset *k* ultimately comprises all sections of the learning data.

2) The state equations that reproduce learning dataset 1 are inferred using the genetic operations and modified Powell method, yielding a population of these state equations as population 1 (Step 1). The completion conditions for Step 1 are as follows.

- An individual with sufficiently high fitness (over 95%) is obtained (Success).
- The average fitness of the whole population (all individuals) becomes 80% or more (Success).
- The maximum number of generations is reached (Failure).

3) Based on population 1, state equations are inferred that reproduce learning dataset 2 by using the genetic operations and modified Powell method, yielding a population of these state equations as population 2 (Step 2). The completion conditions for Step 2 are the same as those above in 2.

4) In the same way, based on population *k*-1, state equations are inferred that reproduce learning dataset *k* by using the genetic operations and modified Powell method, yielding populations of these state equations as population *k* (Step *k*). The individual with the highest fitness in population *k* represents the inferred state equations.

3. Experiments and Results

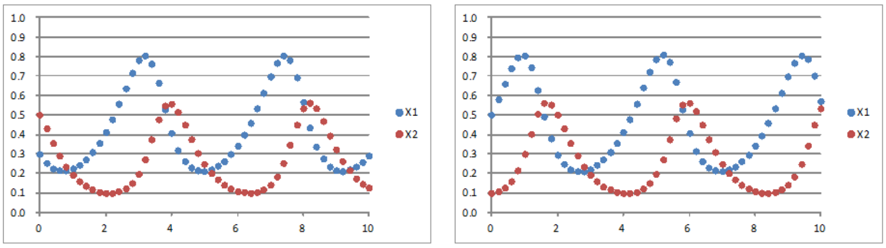
We have examined the inference efficiency of *k*-step GP compared with standard GP. The comparison was conducted based on the inference of state equations expressing two- and three-variable oscillatory biochemical reaction systems. When there only one time-series dataset used as the learning data, there may be cases where a monotonically increase or a monotonically decrease has a high fitness against the oscillation with small amplitude. By initializing the GP with multiple time-series datasets with different initial values, such circumstances can be avoided, thus improving the inference efficiency. In contrast, when the number of time-series datasets is increased, the constraint conditions increase, ultimately making it impossible to obtain a variety of state equations. Thus, we adopt the use of two time-series datasets in these experiments. Moreover, to investigate the influence of the number of partitions in *k*-step GP on inference efficiency, we examined *k* values of 1, 2, and 5 divisions. For $k = 1$, the learning data is undivided (*i.e.*, parameter values are optimized by the modified Powell method only). We refer to this case as "GP + MP". We conducted 100 trials with the standard GP and *k*-step GP procedures and scrutinized the state equations of the obtained oscillatory biochemical reaction systems and their associated time-series data.

3.1. Two-Variable Oscillatory Biochemical Reaction System

We used time-series data in which state variables X_1 and X_2 were obtained by using Equation (6) to generate pseudo-experimentally observed time-series data for a two-variable oscillatory biochemical reaction system. This equation is known as the Lotka-Volterra model [32].

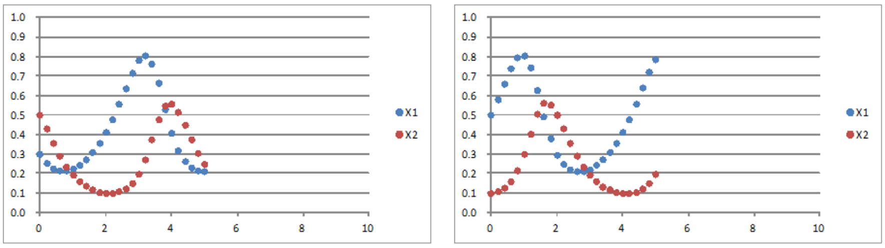
$$\begin{cases} \dfrac{\mathrm{d}X_1}{\mathrm{d}t} = 1.2X_1 - 3.5X_1X_2 \\ \dfrac{\mathrm{d}X_2}{\mathrm{d}t} = 3.5X_1X_2 - 1.8X_2 \end{cases} \tag{6}$$

Figure 3 shows the time-series data generated using Equation (6). In Figure 3, the initial values of the state variables X_1 and X_2 are given as follows: 1) $X_1(0) = 0.3$, $X_2(0) = 0.5$; 2) $X_1(0) = 0.5$, $X_2(0) = 0.1$. We inferred the state equations that can reproduce Figure 3 by using standard GP and *k*-step GP. The parameters related to GP are shown in Table 1. The learning data for standard GP and GP + MP are shown in Figure 3. For $k = 2$, the learning dataset is divided into two sections at time $t = 5$, and from the initial value to time $t = 5$ is learning dataset 1, while that from the initial value to time $t = 10$ is learning dataset 2 (*i.e.*, all learning data). Learning dataset 1 is shown in Figure 4, while learning dataset 2 is the same as the data shown in Figure 3. For $k = 5$, the learning dataset is divided into five sections according to times $t = 2, 4, 6, 8, 10$; from the initial value to time $t = 2$ is learning dataset 1, from the initial value to time $t = 4$ is learning dataset 2, and so on, from the initial value to time $t = 10$ is learning dataset 5 (*i.e.*, all learning data). Learning datasets 1, 2, 3, and 4 are shown in Figures 5-8, respectively, while learning dataset 5 is the same as the data shown in Figure 3.



(a) (b)

Figure 3. Time-series data for a two-variable oscillatory biochemical reaction system as calculated by Equation (6).

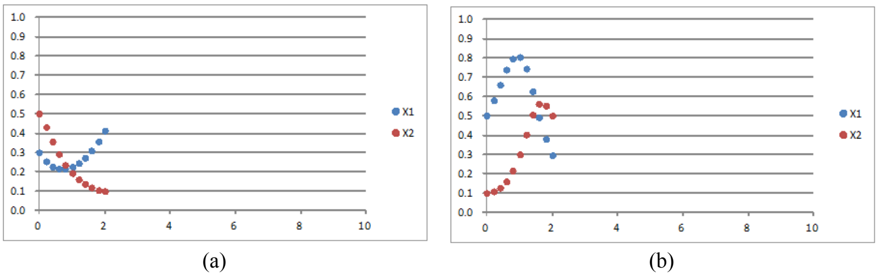


(a) (b)

Figure 4. Learning dataset 1 for $k = 2$ in *k*-step GP for a two-variable oscillatory biochemical reaction system.

Table 1. Genetic operation parameters.

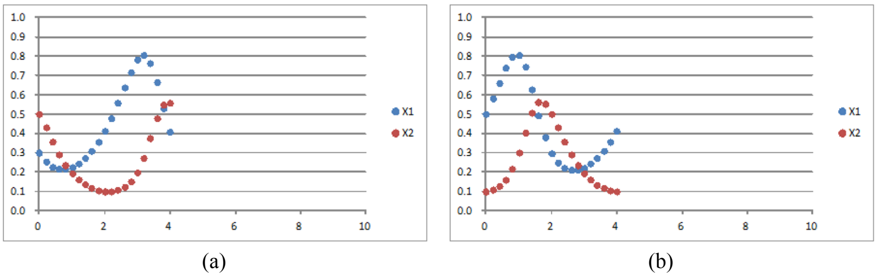
Number of individuals	20
Maximum generation	50
Crossover rate	0.6
Mutation rate	0.3
δ	0.001



(a)

(b)

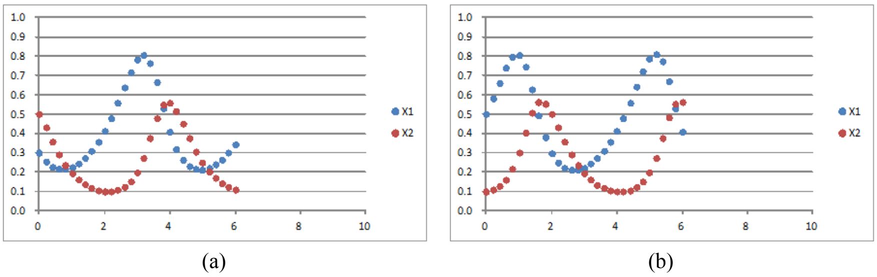
Figure 5. Learning dataset 1 for $k = 5$ in *k*-step GP for a two-variable oscillatory biochemical reaction system.



(a)

(b)

Figure 6. Learning dataset 2 for $k = 5$ in *k*-step GP for a two-variable oscillatory biochemical reaction system.

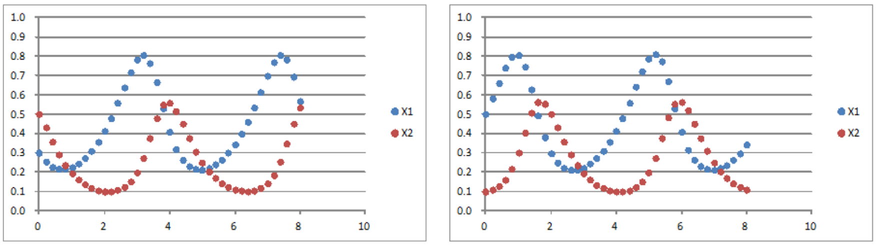


(a)

(b)

Figure 7. Learning dataset 3 for $k = 5$ in *k*-step GP for a two-variable oscillatory biochemical reaction system.

We have been able to infer many oscillatory biochemical reaction systems by using *k*-step GP. The numbers of successfully obtained equations (out of 100 trials) for the proposed oscillatory biochemical reaction systems are shown in Table 2. The examples of state equations obtained as two-variable oscillatory biochemical reaction systems are shown in Table 3; these equations differ from Equation (6).



(a)

(b)

Figure 8. Learning dataset 4 for $k=5$ in *k*-step GP for a two-variable oscillatory biochemical reaction system.

Table 2. Numbers of successfully obtained state equations for two-variable oscillatory biochemical reaction systems (out of 100 trials).

Standard GP	0
GP + MP ($k=1$)	73
k-step GP ($k=2$)	69
k-step GP ($k=5$)	69

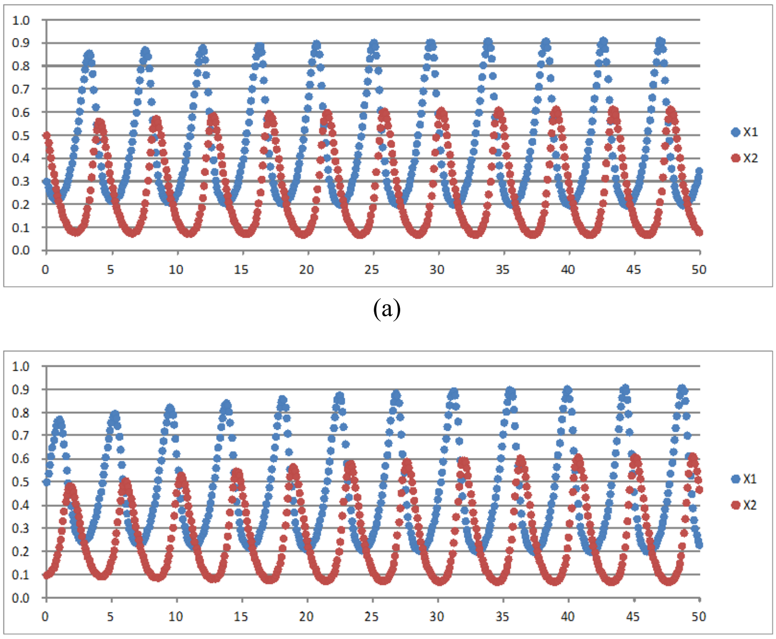
Table 3. Examples of state equations for obtained two-variable oscillatory biochemical reaction systems.

State equations		Fitness
$\begin{cases} \frac{\mathrm{d}X_1}{\mathrm{d}t} = 2.41X_1^2 - 3.23X_1X_2 \\ \frac{\mathrm{d}X_2}{\mathrm{d}t} = 1.49X_1^2 + 5.76X_1X_2 - 3.00X_2 \end{cases}$	(7)	95.3%
$\begin{cases} \frac{\mathrm{d}X_1}{\mathrm{d}t} = 2.06X_1^2 - 8.22X_1X_2 + 1.91X_2 \\ \frac{\mathrm{d}X_2}{\mathrm{d}t} = 9.80X_1^2X_2 - 5.32X_1X_2 \end{cases}$	(8)	95.0%

Figure 9 and Figure 10 show the time-series data for Equation (7) and Equation (8), respectively. As shown in Figure 9 and Figure 10, it was possible to obtain biochemical reaction systems that express steady-state oscillation. These results demonstrate that this approach may lead to the discovery of new biochemical reaction systems that have not yet been considered.

As shown in Table 2, while the standard GP had no success, the GP + MP approach had 73 successful results out of 100 trials. This demonstrates that the optimization of parameter values plays an important role in the inference of biochemical reaction systems involving high nonlinearity.

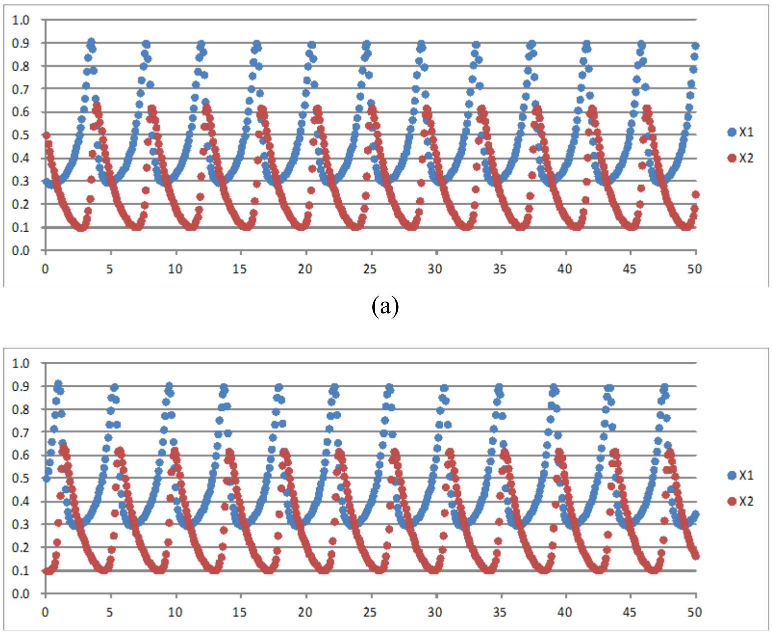
The average values of the elapsed time for the inference (*i.e.*, computation time using a Xeon E5-1620V4 3.5 GHz CPU with 16 GB of memory) are shown in Table 4. The standard GP utilized less computation time, but failed to obtain solutions. To obtain solutions with standard GP, it is expected that many individuals must be prepared, which will require substantial computation time. In *k*-step GP, because the values of parameters such as coefficients are optimized by



(a)

(b)

Figure 9. Time-series data for Equation (7).



(a)

(b)

Figure 10. Time-series data for Equation (8).

Table 4. Average values of elapsed computation time for inference (in sec per 100 trials using a Xeon E5-1620V4 3.5 GHz CPU with 16 GB of memory).

Standard GP	131
GP + MP ($k=1$)	1764
k-step GP ($k=2$)	1823
k-step GP ($k=5$)	1833

the modified Powell method, there is no need to prepare a great number of individuals, which contributes to a reduction in the computation time. Although there was no obvious influence of increasing the number of learning datasets on the inference success rate, this is likely because the target biochemical reaction systems were considered on a relatively small scale. In the following section, we examine cases in which the scale of biochemical reaction systems to be inferred was increased.

3.2. Three-Variable Oscillatory Biochemical Reaction System

We used time-series data in which state variables X_1, X_2, and X_3 were obtained by using Equation (9) to generate pseudo-experimentally observed time-series data for a three-variable oscillatory biochemical reaction system.

$$\begin{cases} \dfrac{dX_1}{dt} = 2.5X_1 - 5.0X_1X_2 - 1.0X_1X_3 \\ \dfrac{dX_2}{dt} = 5.0X_1X_2 - 1.0X_2 - 0.2X_2X_3 \\ \dfrac{dX_3}{dt} = 1.0X_1X_3 + 0.2X_2X_3 - 1.1X_3 \end{cases} \tag{9}$$

Figure 11 shows the time-series data based on calculations with Equation (9); the initial values of state variables X_1, X_2, and X_3 are as follows: 1) $X_1(0) = 0.1$, $X_2(0) = 0.2$, $X_3(0) = 0.3$; 2) $X_1(0) = 0.5$, $X_2(0) = 0.3$, $X_3(0) = 0.2$. We have inferred state equations that can reproduce Figure 11 using standard GP and *k*-step GP. The parameters related to GP are shown in Table 5. The learning data for standard GP and GP + MP are shown in Figure 11. For $k = 2$, the learning dataset is divided in the same way as the two-variable oscillatory biochemical reaction system. Figure 12 shows learning dataset 1, while Figure 11 shows all the data, which is equivalent to learning dataset 2. For $k = 5$, the learning dataset is divided in the same way as the dataset for the two-variable oscillatory biochemical reaction system. Figures 13-16 show learning datasets 1, 2, 3, and 4, respectively, while Figure 11 shows all the data, which is the same as learning dataset 5.

We have been able to obtain many oscillatory biochemical reaction systems using *k*-step GP. The numbers of successfully obtained equations (out of 100

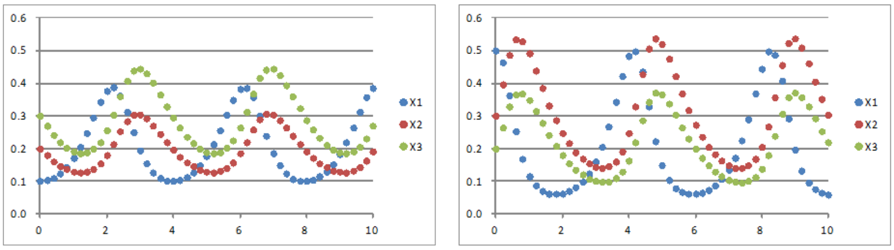
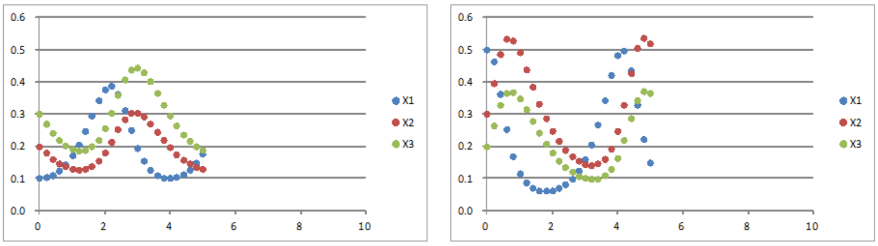


Figure 11. Time-series data based on the three-variable oscillatory biochemical reaction system calculated by Equation (9).

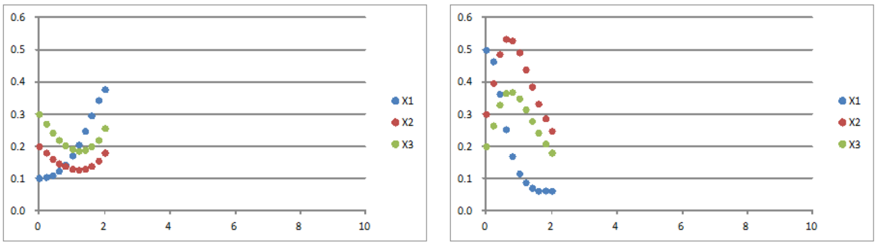
Table 5. Genetic operations parameters.

Number of individuals	30
Maximum generation	100
Crossover rate	0.6
Mutation rate	0.3
δ	0.0005



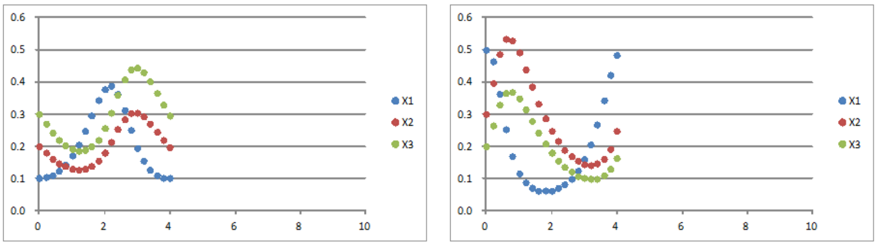
(a) (b)

Figure 12. Learning dataset 1 for $k = 2$ in *k*-step GP for a three-variable oscillatory biochemical reaction system.



(a) (b)

Figure 13. Learning dataset 1 for $k = 5$ in *k*-step GP for a three-variable oscillatory biochemical reaction system.

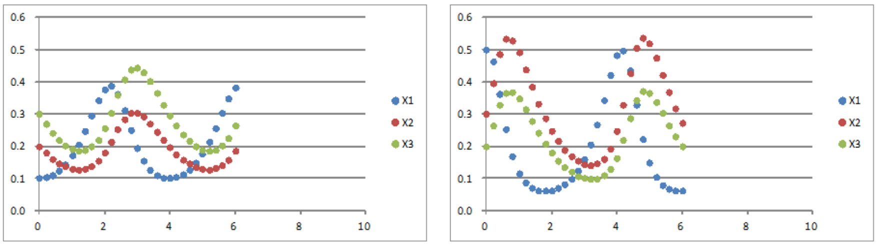


(a) (b)

Figure 14. Learning dataset 2 for $k = 5$ in *k*-step GP for a three-variable oscillatory biochemical reaction system.

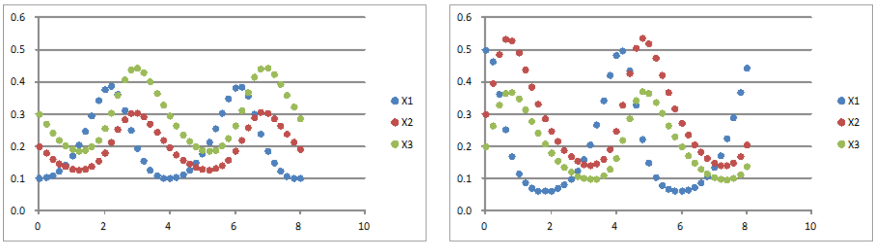
trials) for the proposed oscillatory biochemical reaction systems are shown in Table 6. The examples of state equations obtained as three-variable oscillatory biochemical reaction systems are shown in Table 7; these equations differ from Equation (9).

Figure 17 and Figure 18 show the time-series data for Equation (10) and Equation (11), respectively. As shown in Figure 17 and Figure 18, it was possible



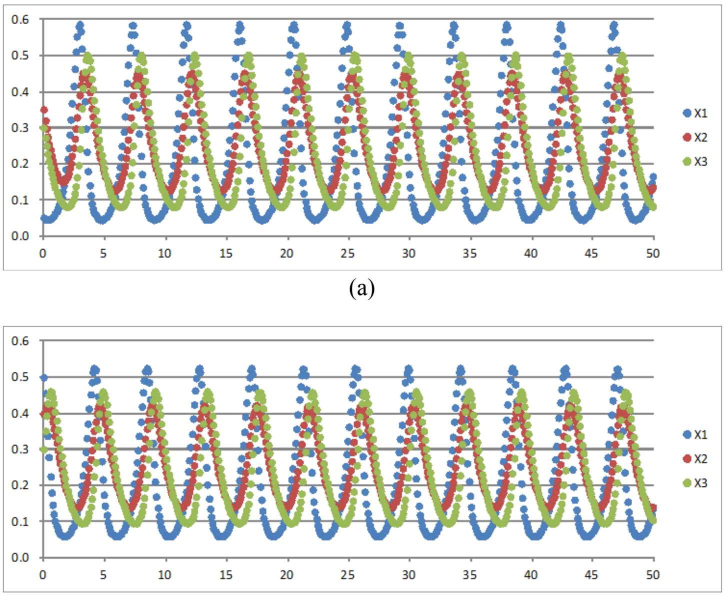
(a) (b)

Figure 15. Learning dataset 3 for $k = 5$ in *k*-step GP for a three-variable oscillatory biochemical reaction system.



(a) (b)

Figure 16. Learning dataset 4 for $k = 5$ in *k*-step GP for a three-variable oscillatory biochemical reaction system.



(a)

(b)

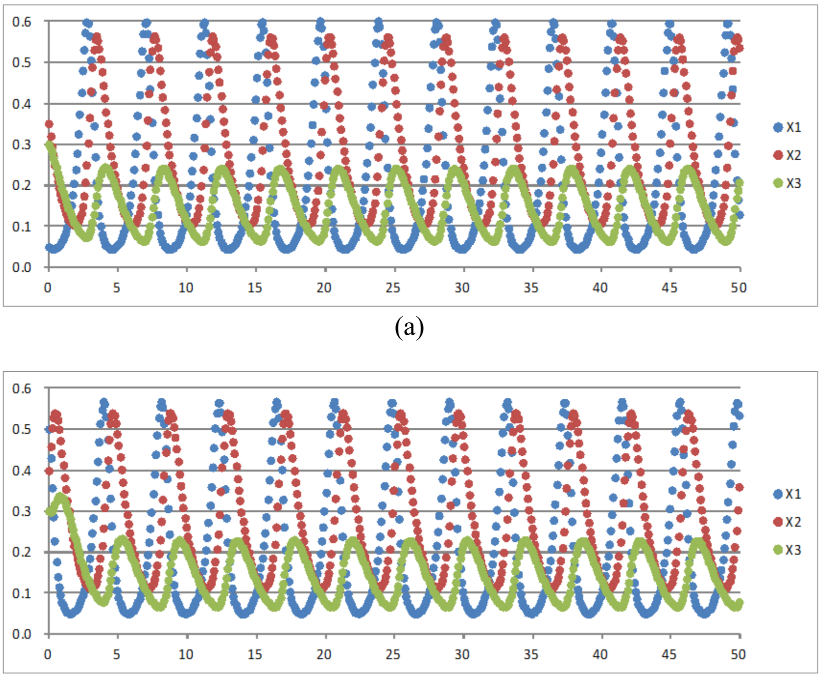
Figure 17. Time-series data based on Equation (10).

Table 6. Numbers of successfully obtained state equations for three-variable oscillatory biochemical reaction systems (out of 100 trials).

Standard GP	0
GP + MP ($k = 1$)	37
k-step GP ($k = 2$)	54
k-step GP ($k = 5$)	64

Table 7. Example state equations for obtained three-variable oscillatory biochemical reaction systems.

State equations		Fitness
$\begin{cases} \dfrac{\mathrm{d}X_1}{\mathrm{d}t} = 2.16X_1 - 9.36X_1X_3 \\ \dfrac{\mathrm{d}X_2}{\mathrm{d}t} = 1.13X_1 - 0.948X_2 \\ \dfrac{\mathrm{d}X_3}{\mathrm{d}t} = 5.41X_1X_3 - 1.15X_3 \end{cases}$	(10)	95.3%
$\begin{cases} \dfrac{\mathrm{d}X_1}{\mathrm{d}t} = 2.45X_1 - 8.99X_1X_2 \\ \dfrac{\mathrm{d}X_2}{\mathrm{d}t} = -1.09X_2 + 5.06X_1X_2 \\ \dfrac{\mathrm{d}X_3}{\mathrm{d}t} = -0.747X_3 + 1.11X_2^2 \end{cases}$	(11)	95.0%



(a)

(b)

Figure 18. Time-series data based on Equation (11).

to obtain biochemical reaction systems that express steady-state oscillation. These results demonstrate that this approach may lead to the discovery of new biochemical reaction systems that have not yet been considered. Table 6 shows that increasing the number of divisions of the learning data improves the inference success rate. The numbers of successfully obtained state equations in GP + MP is reduced compared with Table 2 (the two-variable oscillatory biochemical reaction system). This is considered the number of parameters to be optimized has increased. In *k*-step GP, individuals that have high fitness are gathered during the early stage (Step 1), where the quantity of the learning data is relatively small, and that the individuals evolve while inheriting the characteristics. The

Table 8. Average values of elapsed computation time for inference (in sec per 100 trials using a Xeon E5-1620V4 3.5 GHz CPU with 16 GB of memory).

Standard GP	385
GP + MP ($k=1$)	6407
k-step GP ($k=2$)	6886
k-step GP ($k=5$)	6720

larger scale of the biochemical reaction system to be inferred, the division of learning data will have considered the more effective.

The average values of elapsed time for inference are shown in Table 8 (*i.e.*, computation time using a Xeon E5-1620V4 3.5 GHz CPU with 16 GB of memory). The two- and three-variable oscillatory biochemical reaction systems exhibit similar patterns in computation time.

4. Conclusions

In progress of systems biology, it is essential to develop an algorithm with high heuristic ability for efficiently inferring multiple likely state equations of oscillatory biochemical reaction systems. In conventional research, fixed the structure of state equations, only the inference of included parameter values has been performed. In this study, we obtained various structures of the state equations for two- and three-variable oscillatory biochemical reaction systems with high nonlinearity from only the experimentally observed time-series data by using k-step GP. In particular, we showed that parameter optimization is indispensable for inferring the state equations of oscillatory biochemical reaction systems with high nonlinearity. Moreover, in k-step GP, the learning data are divided and are gradually changed in accordance with the progress of the inference process. This change essentially becomes a series of perturbations to the individuals in the GP procedure. As evolution occurs, the tree structure data of individuals reproduce the learning data by adapting to the perturbations. Through repetition of these perturbations and adaptations, individuals are expected to yield offspring with progressively higher fitness, thus improving inference efficiency.

In the inverse problem, since the correct solution is unknown, it is important to propose a variety of solutions, verify and scrutinize from there and narrow down the solution. Consequently, it is thought that this approach will ultimately lead to the discovery of new biochemical reaction systems that may not yet have been considered. The dynamic behavior of stable biochemical reaction systems can be described as monotonically increasing, monotonically decreasing, steady-state oscillation, or damped oscillation. We have shown that k-step GP can infer state equations for oscillatory biochemical reaction systems. Thus, k-step GP can be applied and contributed to solving the inverse problem of inferring state equations (*i.e.*, both structures and parameter values) of biochemical reaction systems from systems biology data. We are confident that k-step GP is an algorithm that is particularly well suited to inferring state equations for oscillatory biochemical reaction systems.

5. Future Works

One problem that remains to be overcome in *k*-step GP methodology is its high computation time requirements. In particular, the inference of the three-variable oscillatory biochemical reaction system required an average value of computation time of 6720 seconds when the division number was 5 (using a Xeon E5-1620V4 3.5 GHz CPU with 16 GB of memory). We are planning to parallelize *k*-step GP using GPGPU (General Purpose computing on Graphics Processing Units) which has been attracting attention in recent years and to improve its computation time.

Acknowledgements

This study was supposed by Gant-in-Aid for Scientific Research on Innovative Areas (Research in a proposed area), "Synthetic Biology for the Comprehension of Biomolecular Networks" from the Ministry of Education, Culture, Sports, Science and Technology, Japan (No. JP23119001 (M. Okamoto)).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kitano, H. (2001) Systems Biology: A Brief Overview. *Science*, **295**, 1662-1664. https://doi.org/10.1126/science.1069492

[2] Kitano, H. (2001) Foundations of Systems Biology. MIT Press, Cambridge. https://doi.org/10.7551/mitpress/3087.001.0001

[3] Tominaga, D., Koga, N. and Okamoto, M. (2000) Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem. *Proceedings of the Genetic and Evolutionary Computation Conference*, Las Vegas, 8-12 July 2000, 251-258.

[4] Ando, S. and Iba, H. (2001) Quantitative Modeling of Gene Regulatory Network: Identifying the Network by Means of Genetic Algorithms. *Proceedings of the 11th Workshop on Genome Informatics*, Tokyo, 18-19 December 2000, 278-280.

[5] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K. and Tomita, M. (2003) Dynamic Modeling of Genetic Networks Using Genetic Algorithm and S-System. *Bioinformatics*, **19**, 643-650. https://doi.org/10.1093/bioinformatics/btg027

[6] Tsai, K.Y. and Wang, F.S. (2005) Evolutionary Optimization with Data Collocation for Reverse Engineering of Biological Networks. *Bioinformatics*, **21**, 1180-1188. https://doi.org/10.1093/bioinformatics/bti099

[7] Nasimul, N. and Iba, H. (2007) Inferring Gene Regulatory Networks Using Differential Evolution with Local Search Heuristics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**, 634-647. https://doi.org/10.1109/TCBB.2007.1058

[8] Liu, P.K. and Wang, F.S. (2008) Inference of Biochemical Network Models in S-System

Using Multi-Objective Optimization Approach. *Bioinformatics*, **28**, 1085-1092. https://doi.org/10.1093/bioinformatics/btn075

[9] Mitra, K., Nasimul, N. and Iba, H. (2010) Reverse Engineering Gene Regulatory Network from Microarray Data Using Linear Time-Variant Model. *BMC Bioinformatics*, **11**, S56. https://doi.org/10.1186/1471-2105-11-S1-S56

[10] Nakayama, T., Seno, S., Takenaka, Y. and Matsuda, H. (2011) Inference of S-System Models of Gene Regulatory Networks Using Immune Algorithm. *Journal of Bioinformatics and Computational Biology*, **9**, 75-86. https://doi.org/10.1142/S0219720011005768

[11] Komori, A., Maki, Y., Nakatsui, M., Ono, I. and Okamoto, M. (2012) Efficient Numerical Optimization Algorithm Based on New Real-Coded Genetic Algorithm, AREX + JGG, and Application to the Inverse Problem in Systems Biology. *Applied Mathematics*, **3**, 1463-1470. https://doi.org/10.4236/am.2012.330205

[12] Komori, A., Maki, Y., Ono, I. and Okamoto, M. (2013) How to Infer the Interactive Large Scale Regulatory Network in "Omic" Studies. *Procedia Computer Science*, **23**, 44-52. https://doi.org/10.1016/j.procs.2013.10.007

[13] Komori, A., Maki, Y., Ono, I. and Okamoto, M. (2015) Investigating Noise Tolerance in an Efficient Engine for Inferring Biological Regulatory Networks. *Journal of Bioinformatics and Computational Biology*, **13**, Article ID: 1541006. https://doi.org/10.1142/S0219720015410061

[14] Sun, J., Jonathan, M.G. and Hodgman, C. (2012) Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **9**, 185-202. https://doi.org/10.1109/TCBB.2011.63

[15] Heinz, W.E., Christoph, F., Philipp, K., James, L., Stefan, M. and Peter, S. (2009) Inverse Problems in Systems Biology. *Inverse Problems*, **25**, 123014-123051. https://doi.org/10.1088/0266-5611/25/12/123014

[16] Iba, H. and Nasimul, N. (2016) Evolutionary Computation in Gene Regulatory Network Research (Wiley Series in Bioinformatics). Wiley, Hoboken. https://doi.org/10.1002/9781119079453

[17] Chi, Y. and Liu, J. (2016) Reconstructing Gene Regulatory Networks with a Memetic-Neural Hybrid Based on Fuzzy Cognitive Maps. *Natural Computing*, **18**, 301-312. https://doi.org/10.1007/s11047-016-9547-4

[18] Chen, L., *et al.* (2017) Uncertain Programming Model for Uncertain Minimum Weight Vertex Covering Problem. *Journal of Intelligent Manufacturing*, **28**, 625-632. https://doi.org/10.1007/s10845-014-1009-1

[19] Chen, L., *et al.* (2017) Pricing and Effort Decisions for a Supply Chain with Uncertain Information. *International Journal of Production Research*, **55**, 264-284. https://doi.org/10.1080/00207543.2016.1204475

[20] Chen, L., *et al.* (2017) Uncertain Goal Programming Models for Bicriteria Solid Transportation Problem. *Applied Soft Computing*, **51**, 49-59. https://doi.org/10.1016/j.asoc.2016.11.027

[21] Yang, B., Xu, Y., Maxwell, A., Koh, W., Gong, P. and Zhang, C. (2018) MICRAT: A Novel Algorithm for Inferring Gene Regulatory Networks Using Time Series Gene Expression Data. *BMC Systems Biology*, **12**, 115. https://doi.org/10.1186/s12918-018-0635-1

[22] Rodolfo, G., Teresa, C. and Paola, P. (2018) Inverse Problems in Systems Biology: A Critical Review. In: Mariano, B., Ed., *Systems Biology*, Springer, Berlin, 69-94.

https://doi.org/10.1007/978-1-4939-7456-6_6

[23] Koza, J. (1992) Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press, Cambridge.

[24] Miyahara, T. and Kuboyama, T. (2014) Learning of Glycan Motifs Using Genetic Programming and Various Fitness Functions. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **18**, 401-408. https://doi.org/10.20965/jaciii.2014.p0401

[25] Cao, H., Kang, L., Chen, Y. and Yu, J. (2000) Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming. *Genetic Programming and Evolvable Machines*, **1**, 309-337. https://doi.org/10.1023/A:1010013106294

[26] Ando, S., Sakamoto, E. and Iba, H. (2002) Evolutionary Modeling and Inference of Gene Network. *Information Sciences*, **145**, 237-259. https://doi.org/10.1016/S0020-0255(02)00235-9

[27] Sugimoto, N., Sakamoto, E. and Iba H. (2004) Inference of Differential Equations by Using Genetic Programming. *Journal of Japanese Society of Artificial Intelligence*, **19**, 450-459. https://doi.org/10.1527/tjsai.19.450

[28] Iba, H. (2008) Inference of Differential Equation Models by Genetic Programming. *Information Sciences*, **178**, 4453-4468. https://doi.org/10.1016/j.ins.2008.07.029

[29] Powell, M.J.D. (1964) An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *The Computer Journal*, **7**, 155-162. https://doi.org/10.1093/comjnl/7.2.155

[30] Iba, H., de Hugo, G. and Sato, T. (2003) Genetic Programming Using a Minimum Description Principle. In: Kenneth, E.K., Ed., *Advances in Genetic Programming* (*Complex Adaptive Systems*), MIT Press, Cambridge, 265-285.

[31] Byoung, T.Z. and Heinz, M. (1995) Balancing Accuracy and Parsimony in Genetic Programming. *Evolutionary Computation*, **3**, 17-38. https://doi.org/10.1162/evco.1995.3.1.17

[32] Lotka, A.J. (1925) Elements of Physical Biology. Williams & Wilkins Co. Ltd., Baltimore.