Inferring a System of Differential Equations for a Gene Regulatory Network by using Genetic Programming

Erina Sakamoto
School of Engineering,
Dept. of Inf. Comm.Engineering, Univ. of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
erina@miv.t.u-tokyo.ac.jp

Hitoshi Iba
Graduate School of Frontier Sciences,
Dept. of Frontier Informatics, Univ. of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
iba@miv.t.u-tokyo.ac.jp

Abstract- This paper describes an evolutionary method for identifying the gene regulatory network from the observed time series data of the gene's expression. We use a system of ordinary differential equations as a model of the network and infer their right-hand sides by using Genetic Programming (GP). To explore the search space more effectively in the course of evolution, the least mean square (LMS) method is used along with the ordinary GP. We apply our method to three target networks and empirically show how successfully GP infers the systems of differential equations.

1 Introduction

Technologies such as DNA microarrays have seen rapid development in recent years [DeRisi97], allowing large quantities of gene's expression data to become more available. As a result, many researchers are interested in inferring the gene regulatory network from the observed time series and it has become one of the major topics in the field of bioinformatics.

Many models have been proposed to describe the network, among them is the Boolean network [Akutsu99]. In this model, the expression level is either 1 (on) or 0 (off) and the difference in expression levels is not taken into consideration. The relationship among genes is represented in the form of Boolean functions. This modeling is relatively so simple though it suffers from being too coarse. The weighted matrix model which considers the continuous level of the expression was also proposed by Weaver et.al. [Weaver99]. The coefficients and the topology of this model were successfully acquired by using Genetic Algorithms (GA) [Ando00].

Another candidate is the system of differential equations, which is a very powerful and flexible model to describe complex relations among components. Some researchers studied the learning of the gene regulatory network by using the system of differential equations as the model [Chen99] [Tominaga00] [Sakamoto00]. But it is not necessarily easy to determine the suitable form of equations which represent the network. Thus, the form of the differential equation had been fixed during the learning phase in previous studies. As a result, their goal was to simply optimize parameters, i.e., coefficients in the fixed equations. For example, the fixed form of the system of differential equations named S-system[1] was proposed as the model and parameters were optimized by using GA [Tominaga00]. Furthermore, they integrated the Boolean network model and S-system to treat a large-scale genetic network [Maki01].

In this paper, we deal with an arbitrary form in the right-hand side of the system of differential equations to allow flexibility of the model. More precisely, we consider the following general form:

$$\frac{dX_i}{dt} = f_i(X_1, X_2, \ldots, X_n) \quad (i = 1, 2, \ldots, n) \tag{1}$$

where X_i is the state variable and n is the number of the components in the network. In terms of the gene regulatory network, X_i is the expression level of the ith gene and n is the number of genes in the network.

For the sake of identifying the system of differential equations, we use Genetic Programming (GP) to evolve the right-hand side of the equation from the observed time series of the gene's expression.

Although GP is effective in finding the suitable structure, it is sometimes difficult to optimize the parameters, such as constants or coefficients of the polynomials. This is because the ordinary GP searches for them simply by combining randomly generated constants. To avoid this difficulty, we introduce the least mean square (LMS) method.

This paper describes how successfully GP is applied to the inference of the systems of differential equations. More precisely, we empirically show the following points by several experiments:

- The success in the acquisition of the system of equations which is very close to the observed time series.
- The inference of the exact equation form, i.e., the exact causal relationship between the genes.
- The robustness of the acquired system.

[1] S-system is a type of power-law formalism. The concrete form of S-system is given as follows:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} \;.\; \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} (i = 1, 2, \ldots, n)$$

where X_i is a state variable. The first term represents all influences that increase X_i, whereas the second term represents all the influences that decrease X_i [Savageau76].

0-7803-6657-3/01/$10.00 ©2001 IEEE

- The effectiveness of the LMS method.

The rest of this paper is organized as follows. In Section 2, we describe the details of our method, i.e., how GP and LMS methods work in the course of evolution. Three examples of the target networks are used to examine the effectiveness of our method. Their experimental results are shown in Section 3. Then, we discuss the results in Section 4 and give conclusions in Section 5.

2 Method

2.1 Optimization of models using GP

We use GP to identify the gene regulatory network in the form of the system of differential equations. For this purpose, we encode right-hand sides of the equations into a GP individual. Each individual contains a set of n trees, i.e., an n-tuple of trees$(f_1, \ldots, f_n)$. For example, consider the following system of differential equations:

$$\left\{ \begin{array}{rl} \dot{X_1} = & 0.3X_1X_2 + X_2 \\ \dot{X_2} = & 0.5X_1X_2 \end{array} \right. \qquad (2)$$

This is represented as two trees in Fig.1.

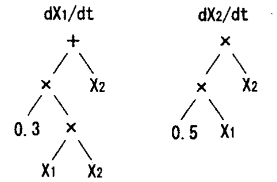

Figure 1: Example of GP individual

Each equation uses a distinct program. A GP individual maintains multiple branches, each of which serves as the right-hand side of a differential equation. Crossover operations are restricted to correspondent branch pairs and mutation is applied to only one randomly selected tree of the individual.

The fitness of each individual is defined as the sum of the squared error and the penalty for the degree of the equations:

$$fitness = \sum_{i=1}^{n}\sum_{k=0}^{T-1}(x'_i(t_0+k\Delta t)-x_i(t_0+k\Delta t))^2+\sum_{j=0} a_j b_j \qquad (3)$$

$$\left(\begin{array}{lcl} t_0 & : & \text{the starting time} \\ \Delta t & : & \text{the stepsize} \\ n & : & \text{the number of the components in the network} \\ T & : & \text{the number of the data points} \end{array} \right)$$

where $x_i(t_0 + k\Delta t)$ is the given target time series ($k = 0, 1, \cdots, T-1$). $x'_i(t_0 + k\Delta t)$ is the time series acquired by calculating the system of differential equations represented by a GP individual. All these time series are calculated by using the forth-order Runge-Kutta method. a_j is the penalty coefficient for jth degree, and b_j is the sum of the absolute value of coefficients of jth degree. a_i must be smaller than a_j when $i < j$ so that the penalty for the higher degree is larger than that for the lower degree. In other words, the individual which is of lower degree and closer to the target time series has the higher possibility to be selected and inherited to the next generation. This fitness derivation is based on the MDL (Minimum Description Length) criterion, which was often used in GP (see [Iba94] and [Zhang95] for examples). When calculating the time series, some individuals may go overflow. In this case, the individual's fitness value gets so large that it will be weeded out from the population.

We use several sets of time series as the training data for GP. This is to acquire the equations as close to the target as possible. Each data set was generated from the same target by using different initial values.

2.2 Optimization of models using LMS method

GP is capable of finding a desirable structure effectively. However, when it comes to optimizing the constants or coefficients, the ordinary GP can not always be effective in the sense that it relies mainly on the combination of randomly generated constants. Thus, we use the least mean square (LMS) method to explore the search space more effectively. More precisely, some individuals are created by the LMS method at some intervals of generations as follows:

1. Choose some data points from the target time series randomly.

2. Derive the coefficients of the approximate expressions of the right-hand sides of the system of differential equations by using the LMS method (The detail of the LMS method is mentioned in 2.3). The approximate expression forms for $n = 3$ are either linear or quadratic as described below:

$$\dot{X_i} = a_iX_1 + b_iX_2 + c_iX_3 + d_i \qquad (4)$$

$$\begin{aligned} \dot{X_i} = & \; a_iX_1^2 + b_iX_2^2 + c_iX_3^2 + d_iX_1X_2 + e_iX_2X_3 \\ & + f_iX_1X_3 + g_iX_1 + h_iX_2 + j_iX_3 + k_i \end{aligned} \qquad (5)$$

3. Replace the worst individuals in the population with the new ones generated above.

2.3 Detail of LMS

In this subsection, we explain the details of the LMS method. Assume that we want to acquire the approximate expression in the following form.

$$y(x_1, \ldots, x_L) = \sum_{k=1}^{M} a_kF_k(x_1, \ldots, x_L) \qquad (6)$$

where $F_k(x_1, \ldots, x_L)$ is the basis function, $x_1, \ldots, x_L$ are the independent variables, $y(x_1, \ldots, x_L)$ is the dependent variable, and M is the number of the basis functions. Let $\mathbf{a}$ be the vector of coefficients, i.e., $(a_1, \ldots, a_M)$. Then, our purpose is to minimize χ^2 described in (7) to acquire $\mathbf{a}$.

$$\chi^2 = \sum_{i=1}^{N} \Big(y(i) - \sum_{k=1}^{M} a_k F_k(x_1(i), \ldots, x_L(i)) \Big)^2 \quad (7)$$

where $x_1(i), \ldots, x_L(i)$ and $y(i)$ are data given for the LMS method and N is the number of data points. Let $\mathbf{b}$ be the vector of $(y(1), \ldots, y(N))$ and $\mathbf{A}$ be the $N \times M$ matrix described below:

$$\begin{pmatrix} F_1(x_1(1), \ldots, x_L(1)) & \ldots & F_M(x_1(1), \ldots, x_L(1)) \\ F_1(x_1(2), \ldots, x_L(2)) & \ldots & F_M(x_1(2), \ldots, x_L(2)) \\ \vdots & \ddots & \vdots \\ F_1(x_1(N), \ldots, x_L(N)) & \ldots & F_M(x_1(N), \ldots, x_L(N)) \end{pmatrix}$$

Then, (8) should be satisfied to minimize χ^2.

$$(\mathbf{A}^T \cdot \mathbf{A}) \cdot \mathbf{a} = \mathbf{A}^T \cdot \mathbf{b} \quad (8)$$

Thus, $\mathbf{a}$ can be acquired by solving this equation.

When applying to the time-series problem, $y(i)$ for the jth equation of the system of differential equations is calculated according to the following discrete difference of the time-series $x_j(t)$:

$$y(i) = \dot{X}_j|_{t=t_i} = \frac{x_j(t_i + \Delta t) - x_j(t_i)}{\Delta t} \quad (9)$$

where t_i is the time of the ith selected data point. For example, consider the system of differential equations (4), in which the number of the network components is three ($L = n = 3$). In this case, we are using four basis functions, i.e., $M = 4$ and (F_1, F_2, F_3, F_4)=$(X_1, X_2, X_3, 1)$. Then, the ith row of the matrix $\mathbf{A}$ is determined as $(x_1(t_i), x_2(t_i), x_3(t_i)$,1).

The coefficients in the approximate expressions of the right-hand sides of the equations can be derived by using $\mathbf{A}$ and $\mathbf{b}(y(1), \ldots, y(N))$ acquired above.

2.4 Overall algorithms

Overall algorithms can be summarized as shown below:

1. Initialization, i.e., creation of the population.

2. Fitness derivation according to (3) by using the Runge-Kutta method.

3. Replacement with offspring newly generated by means of the LMS method at some interval of generations.

4. Selection and genetic recombination.

5. Go to 1.

Population	1000
Generation	300
Crossover rate	0.70
Mutation rate	0.20
LMS method	every 30 generations 2%
Stepsize	0.05
Datapoint	100

Table 1: Experimental parameters for Eg.1

3 Experimental results

We have prepared three networks as the target tasks to test the effectiveness of our method. For each target, three sets of time series with a different initial value were used for the training of GP. Function set F and Terminal set T are as follows:

$$F = \{+, -, *\}$$
$$T = \{X_1, \ldots, X_n, c\}$$

where c is a random constant.

3.1 Example 1: Simple case

In the first experiment, we used a simple system of differential equations shown in (10) as the target model.

$$\begin{cases} \dot{X}_1 = & -0.9X_2 + X_3 \\ \dot{X}_2 = & 0.2X_1X_2 \\ \dot{X}_3 = & X_3 - 0.5X_1 \end{cases} \quad (10)$$

Experimental parameters are shown in Table 1 and initial values of the target data sets are (X_1,X_2,X_3) = (0.2,0.1,0.3), (0.1,0.1,0.1), (0.2,0.2,0.2). The LMS method is applied every 30 generations and 2% of the worst individuals are replaced by the newly generated ones.

The model we have obtained in a typical run is shown in (11) and the mean squared error (MSE) of this model was $6.088 * 10^{5}$. Its time series is shown in Fig.2 (pred X_i) along with that of the target (X_i).

$$\begin{cases} \dot{X}_1 = & 0.0400X_2^2 + 0.0184X_1X_2 - 0.0200X_2X_3 \\ & -0.9020X_2 + 1.0000X_3 \\ \dot{X}_2 = & 0.0369X_2^2 + 0.1920X_1X_2 \\ \dot{X}_3 = & -0.0012X_1^3 + 0.0024X_1^2X_3 + 0.0004X_1X_3 \\ & -0.5200X_1 + 1.0208X_3 - 0.0006 \end{cases} \quad (11)$$

As we compare the acquired system of equations (11) with the target (10), we can confirm that the two systems are almost coincident. For instance, the first equation in (11) is almost equal to

$$\dot{X}_1 = -0.9020X_2 + 1.0000X_3, \quad (12)$$

considering that the very small coefficients, e.g. 0.0184, are regarded as zeros.

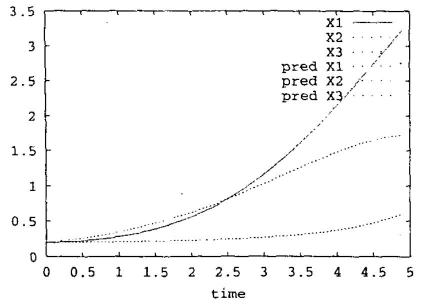

Figure 2: Time series of the acquired model for Eg.1

3.2 Example 2 : Fertility equation

Next, consider the fertility equation in the field of biology [Hofbauer88]. This is the model of the natural selection which takes the following facts into consideration:

1. Mating is not random.
2. Fertilities of mating pairs are different.
3. Viabilities of genotypes are different.

For two alleles A_1 and A_2, there are three genotypes, i.e., A_1A_1, A_1A_2, and A_2A_2. Thus, there are nine different mating types and each of them has an average fertility. Let X_1, X_2, X_3 be the frequencies of A_1A_1, A_1A_2 and A_2A_2, respectively. Then, the typical example model is described as shown below:

$$\begin{cases} \dot{X_1} = & 2.0X_1^2 + 2.5X_1X_2 + 0.375X_2^2 \\ & -X_1P(X_1, X_2, X_3) \\ \dot{X_2} = & 0.75X_2^2 + 2.5X_1X_2 + 2.5X_2X_3 + 3.0X_1X_3 \\ & -X_2P(X_1, X_2, X_3) \\ \dot{X_3} = & 1.5X_3^2 + 2.5X_2X_3 + 0.375X_2^2 \\ & -X_3P(X_1, X_2, X_3) \end{cases} \tag{13}$$

where

$$\begin{aligned} P(X_1, X_2, X_3) = \; & 2.0X_1^2 + 5.0X_1X_2 + 1.5X_2^2 \\ & +3.0X_1X_3 + 5.0X_2X_3 + 1.5X_3^2. \end{aligned} \tag{14}$$

Each coefficient in (13) and (14) is derived from the average fertilities of nine mating types.

We used (13) as the target model. Experimental parameters are shown in Table 2 and the initial values of the target data sets are (X_1,X_2,X_3) = (0.5,0.5,0.0), (0.5,0.0,0.5), (0.0,0.5,0.5).

Population	3000
Generation	900
Crossover rate	0.70
Mutation rate	0.20
LMS method	every 30 generations 2%
Stepsize	0.25
Datapoint	40

Table 2: Experimental parameters for Eg.2 and Eg.3

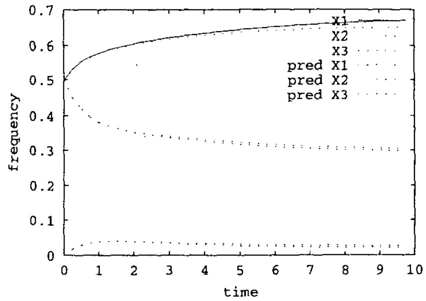

Figure 3: Time series of the best evolved model for Eg.2

The average MSE of 10 runs is $6.225 * 10^{\ 5}$. The best model we have obtained is seen in (15) and the MSE of this model was $2.942 * 10^{\ 5}$. Its time series is shown in Fig.3 along with that of the target.

$$\begin{cases} \dot{X_1} = & 1.0000X_3^3 - 0.0160X_1^2 \\ & +0.2000X_1X_2 - 0.0936X_2X_3 - 0.7000X_1X_3 \\ & -0.2400X_1 + 0.3964X_2 - 0.4800X_3 + 0.0200 \\ \dot{X_2} = & -0.5624X_2^2 - 3.0520X_3^2 \\ & -0.9620X_1X_2 - 2.3680X_2X_3 \\ & +0.2812X_1 - 0.2288X_2 + 2.7380X_3 + 0.1036 \\ \dot{X_3} = & 0.1052X_1X_2X_3 \\ & -0.6400X_1^2 + 0.7400X_2^2 - 0.0057X_3^2 \\ & -0.4200X_1X_2 + 0.7391X_2X_3 - 2.2597X_1X_3 \\ & +0.6800X_1 - 0.2800X_2 - 0.2017X_3 - 0.0400 \end{cases} \tag{15}$$

As can be seen from the Fig.3, the two curves, i.e., the one from target (13) and the other from the acquired system (15), show almost coincident behaviors. Thus, we can confirm the success in the acquisition of the system to be very close to the target.

3.3 Example 3 : Weighted gene regulatory network

A weighted network was proposed to represent the gene regulatory network [Weaver99]. The example of this model is shown in Fig.4. In this network, each node represents the gene. The arrow and value indicate the regulatory relation and its level. Positive values represent the promotion while negative ones the repression.

In this model, the gene expression level($x_j(t)$) is calcu-

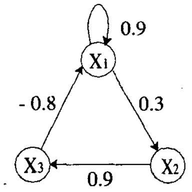

Figure 4: Weighted Gene Regulatory Network

	0% noise	5% noise	10% noise
Linear LMS	0.0100	0.0123	0.0153
Quadratic LMS	0.0107	0.0226	0.0267

Table 3: The average MSE of Eg.3

lated by the following equations:

$$s_i(t) = \sum_{j=0}^{n} \omega_{ji} x_j(t) \tag{16}$$

$$x_i(t+1) = \frac{m_i}{1 + e^{\;s_i(t)}} \tag{17}$$

where ω_{ij} is the weight on gene i from gene j, $s_i(t)$ is the regulation state of gene i at step t, and m_i is the maximum expression level of gene i. For instance, ω_{31}=-0.8 and ω_{11}=0.9 for the network shown in Fig.4. If ω_{ij}=0, then the correspondent arrow is omitted for the simplicity.

We used the model shown in Fig.4 as the target. Experimental parameters are shown in Table 2 and initial values of the target data sets are (X_1,X_2,X_3) = (0.5,0.0,0.0), (0.0,0.5,0.0), (0.0,0.0,0.5).

The average MSE of 10 runs is 0.0100. The best model we have obtained is (18) and MSE of this model was 0.0039. Its time series is shown in Fig.5 along with that of the target.

$$\begin{cases} \dot{X}_1 = & -0.1159X_1X_3^2 - 0.0023X_3^2 + 0.1974X_1X_3 \\ & -0.6010X_3 + 3.0344 \\ \dot{X}_2 = & -0.0200X_1^3 - 0.0017X_1^2X_2 \\ & -0.0173X_1^2 - 0.2109X_2^2 + 0.1750X_1X_2 \\ & +0.4138X_1 - 0.5144X_2 + 2.6125 \\ \dot{X}_3 = & -0.0151X_2X_3^2 - 0.0135X_1X_2X_3 \\ & +0.0352X_3^2 + 0.0593X_2X_3 \\ & -0.0135X_1 + 0.9008X_2 - 1.2917X_3 + 3.0039 \end{cases} \tag{18}$$

It can be said from Fig.5 that the time series of the acquired system (18) fits that of the target system quite well.

To test the robustness of our method to the real noisy world, we have conducted the experiment with noise-added data sets. 5% and 10% random noises were added to the target time series. The averaged MSE values of 10 runs are shown in Table 3.

We have also experimented without the LMS method to confirm its effectiveness. The average MSE of 10 runs is 0.0262 while that of the experiment with the LMS method is 0.0100. From this result, we can see that the LMS method worked effectively to acquire the better individual. The typical case of the evolution is shown in Fig.6.

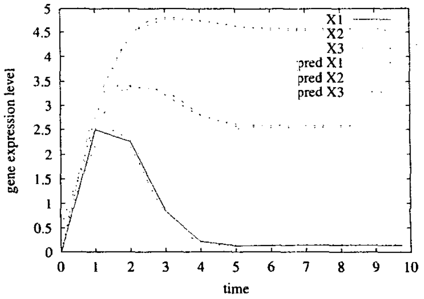

Figure 5: Time series of the best evolved model for Eg.3

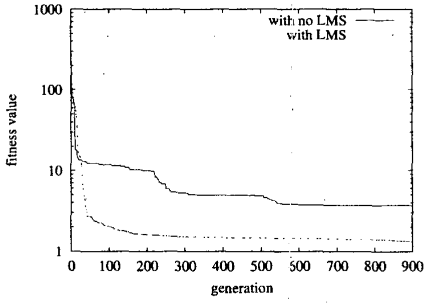

Figure 6: Effectiveness of LMS method

4 Discussion

Previous experimental results have shown that the acquired time series fit the target ones quite well in all the target tasks. Thus, our method is supposed to derive the causal relationship among the components for the gene networks.

In the above experiments, the number of the genes we chose was around three and seems far less than the total number in the real network. To tackle the identification of a large-sized network, some other method should be used for the purpose of pre-processing, i.e., reducing the network into a set of small-sized sub-networks. For instance, the network was devided into moderate-sized subnetworks by using the static Boolean network model [Maki01]. We will work on the integration of some methods.

The LMS method has played a significant role in searching for better individuals effectively in the course of the evolution. As can be seen from Fig.6, the fitness values with the LMS method decrease more quickly in the early phase than that without the LMS method. The LMS method seems to provide a better seed for the GP search, in the sense that the better offspring are created from those of better seeds. This explains why the effect of the LMS method is not only restricted to the generation when the LMS method is applied. In other words, the fitness values keep decreasing, i.e., improving, even after the generation of the LMS method.

As a related work, Cao et. al. also proposed to acquire the system of differential equations in a arbitrary form from the target time series using GP [Cao00]. In this method, GA is used for the optimization of parameters. On the other hand, the salient features of our approach is (1) the LMS method for the parameter tuning, and (2) the MDL criterion to derive the fitness value along with the error of the acquired time series. The comparative experiments between the two approaches will be conducted in our future research.

We will work on the following extensions of the current method in our future works:

1. The choice of the appropriate degree for the LMS method.
2. The setting of the penalty function.
3. The presentation of multiple data sets for the GP training.

First, the proper degree of the approximate expression may depend on the target time series. However, we don't know the appropriate degree for the LMS method beforehand. Moreover, there is another factor to be considered, i.e., the robustness. Though we cannot draw any concluding remarks only from the above-mentioned experimental results, it seems that using the linear expression results in the acquisition of a more robust system than using the quadratic expression, especially with the noise-added data (see Table 3). The overfitting seems to have occurred with the quadratic expression. We will examine the robustness of our method on more networks and will try to achieve more robust systems by extending the LMS algorithms. One possible solution will be to change the degree automatically according to the evolution speed.

Second, the penalty coefficients, i.e., a_i in (3), should be carefully determined. In general, a_i must be smaller than a_j when $i < j$ so that the penalty for the higher degree is larger than that for the lower degree. However, their absolute values are heuristically given for a specific task. The more algorithmic determination of these values is our current research concerns. For instance, the assignment algorithm using the radius of curvature is proposed by Nikolaev et.al. [Nikolaev00].

Third, we used several sets of time series as the training data for GP. Each data set was generated from the same network with different initial values. We provided all of them as the training data throughout the whole generation. This is to enhance the robustness of the acquired system. However, there can be other ways to the provision. For example, the whole generation is divided into a set of segments. In the first segment, one time series is used for the training of GP. In the second segment, another time series as well as the first one is used as the training data for GP. In the same way, the third set of time series is used in addition to the above two sets as the training data for GP in the third segment. Clearly the computational burden is reduced with the latter provision method and the learning is expected to proceed gradually and reasonably. We have conducted the experiments using this latter method, and found that in some cases the average of MSE values were not necessarily better. We are working on this extension as our current research topic.

5 Conclusion

We have proposed the method of inferring the right-hand sides of the system of differential equations from the observed time series by using GP along with the LMS method. We showed how successfully our method can infer the network by several experiments.

As with many other proposed models, the solution which fits the given time series quite well is not necessarily determined uniquely. In other words, there may exist more than one solution which behave consistently with the target. Therefore, even if one system of differential equations is acquired as a solution, we cannot disregard other candidates. Our aim is to obtain the candidates scattered in the huge search space and to propose to users the possible causal relationship among the network components. Therefore, as future works, we will concentrate on the construction of the interactive system, which proposes the possible solutions and tells users what kinds of data are needed to determine the relationship among the components. By using this interactive system, users will be able to pick up the biologically correct equations or discard the biologically meaningless equations from the suggested ones. We will also try to solve some of the real biological problems.

Acknowledgments

The authors thank Professor Masahiro Okamoto for his comments on our early work.

Bibliography

[Akutsu99] Akutsu, T., Miyano, S., Kuhara, S., Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model, Proc. of Pacific Symposium on Biocomputing, pp.17-28, 1999.

[Ando00] Ando, S., Iba, H., Quantitative Modeling of Gene Regulatory Network -Identifying the Network by Means of Genetic Algorithms-, Poster Session of Genome Informatics Workshop 2000.

[Cao00] Cao, H., Kang, L., Chen, Y., Yu, J., Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming, Genetic

Programming and Evolvable Machines, 1, pp.309-337, 2000.

[Chen99] Chen, T., He, H.L., Church, G.M., Modeling Gene Expression with Differential Equations, Proc. of Pacific Symposium on Biocomputing, pp.29-40, 1999.

[DeRisi97] DeRisi, J.L., Lyer, V.R., Brown, P.O., Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale, Science vol.278, pp.680-686, 1997.

[Hofbauer88] Hofbauer, J. and Sigmund, K., The Theory of Evolution and Dynamical Systems, Cambridge University Press, 1988.

[Iba94] Iba, H., deGaris, H., Sato, T., Genetic Programming using a Minimum Description Principle, in Advances in Genetic Programming, MIT Press, pp.265-284, 1994.

[Maki01] Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., Eguchi, Y., Development of a System for the Inference of Large Scale Genetic Networks, Proc. of Pacific Symposium on Biocomputing, 2001.

[Nikolaev00] Nikolaev, N.I. and Iba, H., Inductive Genetic Programming of Polynomial Learning Networks , In: X. Yao (Ed.), Proc. of the IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks, ECNN-2000, IEEE Press, pp.158-167, 2000.

[Sakamoto00] Sakamoto, E., Iba, H., Identifying Gene Regulatory Network as Differential Equation by Genetic Programming, Poster Session of Genome Informatics Workshop 2000.

[Savageau76] Savageau, M.A., Biochemical Systems analysis: a study of function and design in molecular biology, Addison-Wesley, Reading, 1976.

[Tominaga00] Tominaga, D., Koga, N., Okamoto, M., Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem, Proc. of Genetic and Evolutionary Computation Conference, pp.251-258, 2000.

[Weaver99] Weaver, D.C., Workman, C.T., Storm, G.D., Modeling Regulatory Networks with Weight Matrices, Proc. of Pacific Symposium on Bioinformatics 4, pp.112-123, 1999.

[Zhang95] Zhang, B.-T. and Muehlenbein, H., Balancing accuracy and parsimony in genetic programming, Evolutionary Computation, 3(1):17-38, 1995.