Multi-stage Evolutionary Algorithms for Efficient Identification of Gene Regulatory Networks

Kee-Young Kim, Dong-Yeon Cho, and Byoung-Tak Zhang

Biointelligence Lab, School of Computer Science and Engineering,
Seoul National University, Seoul 151-742, Korea
{kykim, dycho, btzhang}@bi.snu.ac.kr

Abstract. With the availability of the time series data from the high-throughput technologies, diverse approaches have been proposed to model gene regulatory networks. Compared with others, S-system has the advantage for these tasks in the sense that it can provide both quantitative (structural) and qualitative (dynamical) modeling in one framework. However, it is not easy to identify the structure of the true network since the number of parameters to be estimated is much larger than that of the available data. Moreover, conventional parameter estimation requires the time-consuming numerical integration to reproduce dynamic profiles for the S-system. In this paper, we propose multi-stage evolutionary algorithms to identify gene regulatory networks efficiently. With the symbolic regression by genetic programming (GP), we can evade the numerical integration steps. This is because the estimation of slopes for each time-course data can be obtained from the results of GP. We also develop hybrid evolutionary algorithms and modified fitness evaluation function to identify the structure of gene regulatory networks and to estimate the corresponding parameters at the same time. By applying the proposed method to the identification of an artificial genetic network, we verify its capability of finding the true S-system.

1 Introduction

Although mathematical modeling for the biochemical networks can be achieved at different level of detail (see [1] and [2] for the reviews of metabolic and genetic regulatory networks modeling), we can cluster them into three dominant approaches [3]. One extreme case is mainly intended to describe the pattern of interactions between the components. Graph-based representation gives us the insight for large architectural features within a cell and allows us to discovery principles of cellular organization [4]. However, it is difficult to handle the dynamics of the whole system since these models are very abstract. The other extreme primarily focuses on describing the dynamics of the systems by some kinds of equations which can explain the biochemical interactions with stochastic kinetics [5, 6]. While these approaches lead to realistic, quantitative modeling on cellular dynamics, the application is limited to the small systems due to their computational complexities.

One of the appropriate approaches for the pathway structure and dynamics identification is S-system [7]. It is represented as a system of ordinary differential equations which have a particular form, where each component process is characterized by

F. Rothlauf et al. (Eds.): EvoWorkshops 2006, LNCS 3907, pp. 45 – 56, 2006.

power-law functions. S-system is not only general enough to represent any nonlinear relationship among the components but also able to be mapped onto the network structure directly. In spite of these advantages, there is a serious drawback which prevents the S-system from wildly spreading over the systems biology communities. Its large number of parameters should be estimated for the small number of observed dynamic trends. Provided that n components such as genes and proteins are involved in a certain living system, we must optimize at least $2n(n+1)$ parameters for the S-system.

Evolutionary computation has been used from its inception for automatic identification of a given system or process [8]. For the S-system models, some evolutionary search techniques have been proposed [9-12]. However, they require the time-consuming numerical integrations to reproduce dynamic profiles for the fitness evaluations. To avoid this problem, Almeida and Voit have employed an artificial neural network (ANN) to smooth the measured data for obtaining slopes of gene expression level curves [13]. By comparing the slope of each S-system in the population with the estimated one from ANN, we can evaluate the fitness values of the individuals without the computationally expensive numerical integrations of differential equations. This method also provides the opportunity for a parallel implementation of the identification task since a tightly coupled system of non-linear differential equations can be separated. Hence, they are able to reduce the time complexity drastically. While collocation method [14] can save the computational cost by approximating dynamic profiles, their estimated systems tend to be invalid since the number of measured data is usually insufficient. This lack of data problem can be resolved by sampling new points from the fitted curves. For the well-estimated profiles, however, we should determine the optimal topology of the artificial neural network such as the number of hidden units and layers.

In this paper, we propose multi-stage evolutionary algorithms to identify gene regulatory networks efficiently. With the symbolic regression by genetic programming (GP), we could evade the numerical integration steps. Here, we have no need to predetermine the topology of the model for the expression profiles since genetic programming can optimized the topology automatically. We also develop hybrid evolutionary algorithms to identify the structure of gene regulatory networks and to estimate the corresponding parameters at the same time. Most previous evolutionary approaches for the S-system identification have used the structural simplification procedure in which some parameters whose values are less than a given threshold are reset to zero. Although this method is able to make the network structure sparse, the true connections which represent somewhat small effect can be deleted during the procedures. That is, it is not easy to set the suitable value for the threshold. In our scheme, Binary matrices for a network structure and real vectors and matrices for parameter values of S-system are combined into a chromosome and co-evolved to find the best descriptive model for the given data. Hence we can identify the S-system without specifying the threshold values for the structural simplification. By applying the proposed method to the artificial gene expression profiles, we successfully identified the true structure and estimated the reasonable parameter values with the smaller number of data than the previous study.

2 Materials and Methods

2.1 S-System

The S-system [7, 15] is a set of nonlinear differential equations described as follows:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}}, \quad i = 1,2,...,n, \tag{1}$$

where n is the number of dependent variables whose concentrations X_i change dynamically according to above equations and m is the number of independent variables whose concentrations remain constant during the processes. The non-negative parameters α_i and β_i are called rate constants. The real-value exponents g_{ij} and h_{ij} are kinetic orders to represent the interactive effect of X_j to X_i. These differential equations can be divided into two components. The first term represents influences that increase X_i and the second term represents influences that decrease X_i. Thus, X_j induces the synthesis of X_i in case $g_{ij} > 0$, whereas X_j inhibits the increase of X_i if $g_{ij} < 0$. Similarly, a positive (negative) value of h_{ij} indicates that X_j expedites (restrains) the decline of X_i.

We should estimate at least $2n(n+1)$ parameters even if the values related to the independent variables are assumed to be known. That is, α_i, β_i, g_{ij}, and h_{ij} are parameters that must be estimated by evolutionary algorithms (EAs). The generally adopted fitness function for EAs is the sum of relative squared errors:

$$E = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(\frac{X_i(t) - \hat{X}_i(t)}{X_i(t)} \right)^2, \tag{2}$$

where T is the number of sampling points for fitness evaluation, X is the measured data points from the biological experiments and $\hat{X}$ is the values obtained by the numerical integration step of the S-system in the population.

2.2 Symbolic Regression by Genetic Programming

As we mentioned in the introduction, numerical integration steps for the fitness evaluations are very time-consuming. To circumvent these processes, we propose 2-stage evolutionary algorithms for finding the structures and parameters of S-systems. As a preprocessing step, genetic programming (GP) [16] performs symbolic regression for the given time-course data. Through this step, we can predict the dynamic profiles of the given S-system and obtain more data points as well as the derivations of the point for the second stage of our algorithm. This allows us to get rid of the numerical integrations in the fitness evaluations. Compared with the study which employed the artificial neural networks [13], genetic programming has the following advantages for the curve fitting tasks. First, there is no necessity for adjusting the number of hidden layers and units. Second, the results of GPs are more understandable than those of neural networks since GPs return some mathematical functions

instead of the illegible graph and a set of weights. Hence, we can easily obtain derivations of each time point if the result functions of GPs are differentiable.

2.3 Hybrid Evolutionary Algorithms

At the second stage of our evolutionary algorithm, we use a hybrid evolutionary algorithm for searching the structures and parameters of S-system. The whole procedure of our algorithm is summarized in Fig. 1.

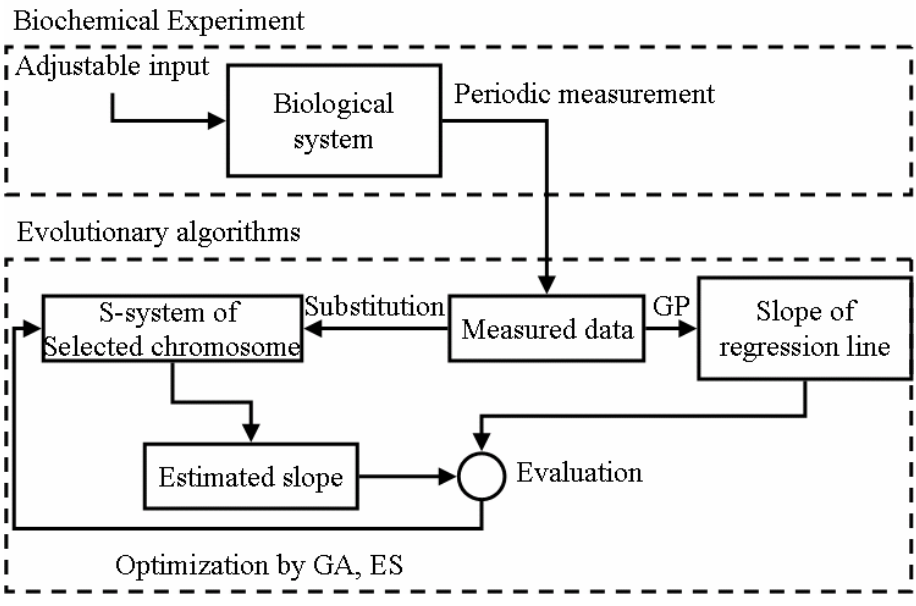
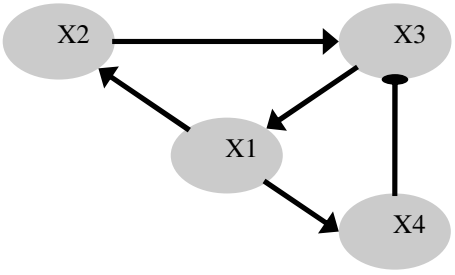


Fig. 1. The whole proposed multi-stage evolutionary algorithm for the identification of S-system

In biochemical experiment step, the dynamic profiles of the involved genes can be obtained by the periodic measurement. We are also able to predict slopes at each time point from the regression line of the measured data by genetic programming. Then, we can evaluate and optimize the chromosomes of evolutionary algorithms by comparing the estimated slopes which came from the data substitution into the S-system with the predicted slopes of the GP regression lines. Hence, the fitness values of each S-system can be evaluated without the time-consuming numerical integrations as follows:

$$E = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(\frac{\dot{X}_i(t) - X_i'(t)}{\dot{X}_i(t)} \right)^2, \tag{3}$$



(a) graph representation

$$\frac{dX_1}{dt} = 12X_3^{0.8} - 10X_1^{0.5} \qquad \frac{dX_2}{dt} = 8X_1^{0.5} - 3X_2^{0.75}$$
$$\frac{dX_3}{dt} = 3X_2^{0.75} - 5X_3^{0.5}X_4^{0.2} \qquad \frac{dX_4}{dt} = 2X_1^{0.5} - 6X_4^{0.8}$$

(b) S-system

$$g = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \qquad h = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

binary matrices for the structure

$$\alpha = \begin{pmatrix} 12.0 \\ 8.0 \\ 3.0 \\ 2.0 \end{pmatrix} \quad g = \begin{pmatrix} 0.0 & 0.0 & -0.8 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.75 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.0 \end{pmatrix} \quad \beta = \begin{pmatrix} 10.0 \\ 3.0 \\ 5.0 \\ 6.0 \end{pmatrix} \quad h = \begin{pmatrix} 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.75 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.8 \end{pmatrix}$$

real vectors and matrices for the parameters

(c) the structure of the chromosome

Fig. 2. (a) The true structure of the artificial gene regulatory network (b) S-system representation (c) The structure of the chromosome in our hybrid evolutionary algorithms

where T is the number of sampling points, $\dot{X}$ is the gradient of the regression line obtained by the genetic programming and X' is the calculated value of each equation in the S-system.

We develop hybrid evolutionary algorithms to identify the structure of gene regulatory networks and to estimate the corresponding parameters at the same time. In this scheme, Binary matrices for a network structure and real vectors and matrices for parameter values of S-system are combined into a chromosome (Fig. 2) and co-evolved to find the best descriptive model for the given data. While crossover operator is applied to binary matrices for searching the structure of the system, their corresponding parameter values also exchanges. This kind of crossover can inherit the good structures as well as the parameter values in the parents to the offspring. That is, we use a row exchange crossover which simply selects the row of the matrix g or h (or both) on the parents, and swaps each other with the parameter values in the real vectors and matrices. For example, Fig. 3(a) shows the case in which we select the

second row of g on parents. Mutation operator randomly selects an equation of a parent, and then inserts or deletes a component at the selected parent as shown in Fig. 3(b) and (c). These operators perform the local searches in the structure space. In this case, the parameter values are randomly generated for the insertion and reset to zero for the deletion.

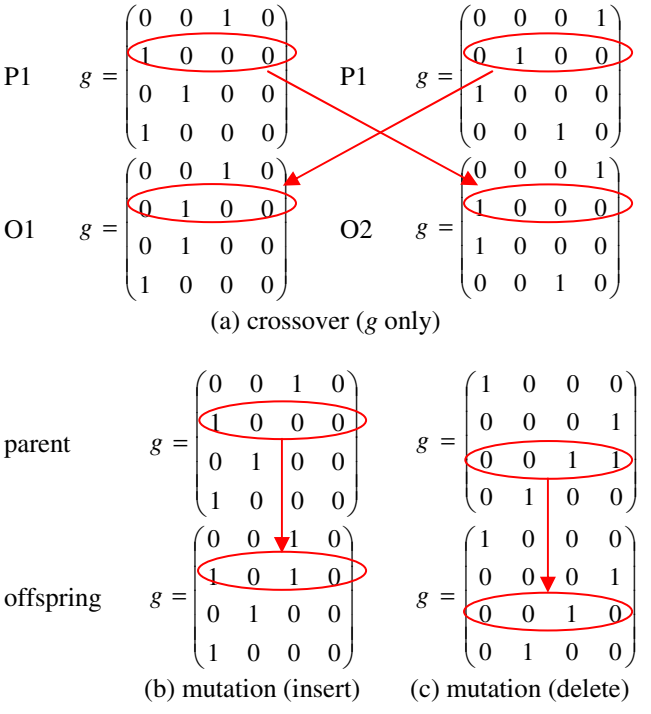


Fig. 3. Crossover and mutation operators for the binary matrices

We also give some variety to the fitness function in equation (3). In conventional scheme, all points of data have the same weights on the fitness values. However, it is difficult to fit the data points which have large second-order differentiation. Moreover, this makes the parameter values of the chromosomes different from the true one even if they have good fitness values. Thus we multiply the second-order differentiation to each term of evaluation function. The modified fitness function in our algorithm described as follows:

$$E = \sum_{i=1}^{n} \sum_{t=1}^{T} \ddot{X}_i(t) \left(\frac{\dot{X}_i(t) - X'_i(t)}{\dot{X}_i(t)} \right)^2, \tag{4}$$

where $\dot{X}$ is the gradient of the GP regression line, $\ddot{X}$ is second-order differentiation and X' is the calculated value of the each equation in the S-system. By introducing $\ddot{X}$

to fitness function, we can obtain better fitness value of true structure than those of other structures. After the fitness values of the offspring created by the crossover and mutation according to their probabilities are evaluated by equation (4), parameters in the real vectors and matrices are adjusted through the (1+1) evolutionary strategy [17].

We employ the restricted tournament selection (RTS) proposed originally in [18] to prevent the premature convergence on a local-optimum structure and to find multiple topology candidates. In RTS, a subset of the current population is selected for each newly created offspring. The size of these subsets is fixed to some constant called the window size. Then, the new offspring competes with the most similar member of the subset. Since the window size is set to the population size in our implementation, each offspring is compared with all S-system in the current population. If the new one is better, it replaces the corresponding individual; otherwise, the new one is discarded. For the similarity measure, we calculate the structural hamming distances between the new offspring and all individuals in the population by using the binary matrices.

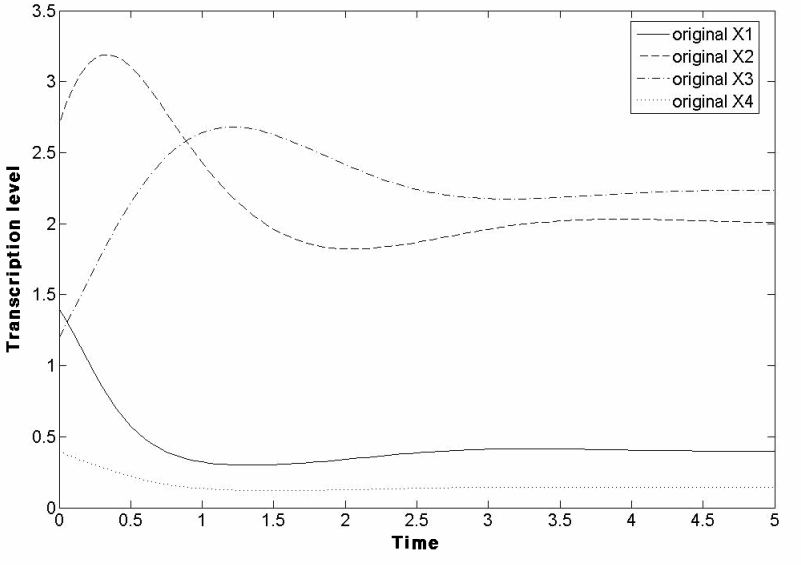
3 Experimental Results

3.1 Data

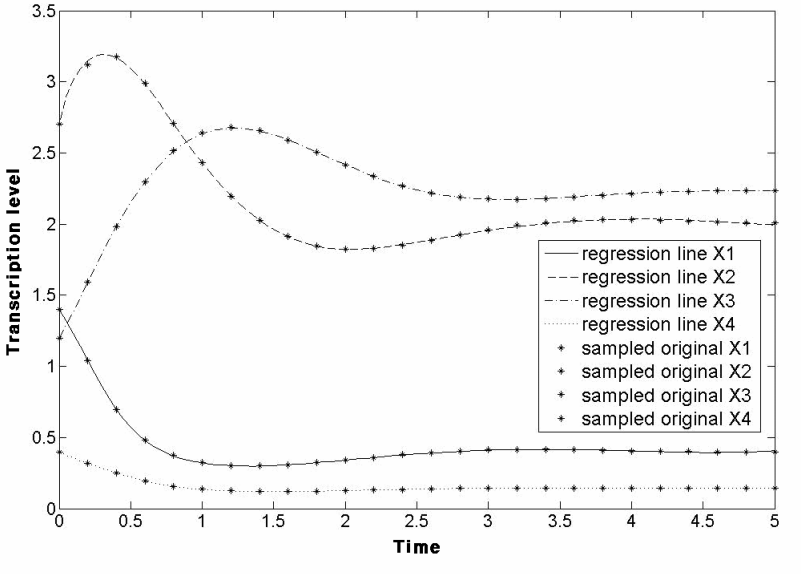
To evaluate the performance of the proposed algorithms, we consider the artificial gene regulatory network which came from [13]. This gene regulatory network modeled 4 reactants influenced with one another and all reactants are auto-regulated. The S-system model for the used gene regulatory network is represented in Fig. 2(a) and (b) and the time-course graph of the given gene regulatory network is represented in Fig. 4(a). To create artificial time-course profiles, we solve this true S-system with the initial values, $X_1(0)=1.4$, $X_2(0)=2.7$, $X_3(0)=1.2$, and $X_4(0)=0.4$ by using the 4th-order Runge-Kutta method. The size of sampled data point is 25 and their time intervals are 0.2 seconds as shown in Fig. 4(b). We set 4 times smaller number of time points than that of the previous study [13] since it is very difficult to obtain a lot of time-course data from the real biological measurements.

3.2 Results of the Genetic Programming

We use Frayn's GPLib library [18] for the symbolic regression with GP. To evolve the mathematical models for the given data, we use the function set $F = \{+, -, \times, /, \hat{}, \sin, \exp, \log, \mathrm{sqrt}\}$ and terminal set $T = \{t, \Re, \pi\}$, where $\Re$ is real constant and t is the time point. The population size is 3,000 and the maximum number of generations is 2,000. Tournament selection is used and its size is 4. Crossover rate is 0.35 and mutation rate is 0.5. We set the length penalty of the genetic program as zero for the accurate curve fitting. We generate 100 points and derivations from the obtained models for the input of the next stage, that is, the hybrid evolutionary algorithms.



(a) original graph



(b) sample data points and regression results by GP

Fig. 4. The true (a) simulated (b) time-series data for the artificial genetic network

Results of the genetic programming step are as follows:

$f_1(t)$=(sqrt((((sqrt(((sqrt(2.957153))-((sin(sqrt(t)))+((sin((1.854912)-(((sqrt((3)*(t)))
(sqrt(t)))+(4.435898))))+(-2.355442))))(t)))+((40.675830)/(exp((t)*((sqrt(t))-
((sin(((sin((((sqrt((3.756391)*(t)))*(sqrt(t)))+(log(86)))+(7)))+(3))+(sin(sin((1.654737
)*(t))))))+(-2.355442)))))))/(sqrt(54.598150)))/(sqrt(7.931547)))),

$f_2(t)$=(sqrt(((3.777992)-((((((4.190957)-((t)-((sin(((((t)*((t)^(t)))-(((((t)*((2.883554)
/((4)-(log(t)))))+(2.791190))-((exp((t)-(2.226704)))^(sqrt(9.642561)))))/(((t)^(t))))
*(2.678347))))/(3.462360))+(3.792098))-(4.796861))/(4)))+(((((3.792098)-((exp(t))
/(3.462360)))-(((t)*((t)^(t)))) /((t)^(t)))/(((t)^(t))))),

$f_3(t)$=((log(log(exp(log(log(exp(exp(((log(exp(π)))-((sin((9)+(((t)+(8.000000))
^(sqrt(1.420245)))))/(((exp(t))*(379.000000))/(84.000000))))-((sin((8)+(((t)
+(((log(109))-(1.258803))/(6.620476)))^(sqrt(log(4.059461))))))
/(((exp(((8.337047)*((log(log(sqrt(3.021610))))+(2.000000)))-(5.912041)))*(exp(t)))
/(85.000000)))))))))^(5.933873)),

$f_4(t)$=((((log(6.249382))^((sqrt(6))*(((sqrt(10.000000))^((1.172486)-(t)))
/(6.981835))))^((1.161464)-((1.161464)/(((((sqrt(6.249382))*((log(7.008566))
*((((((exp((6.980522)/((sqrt(6.288201))^(1.344547))))^((1.735082)-(t)))
/(0.290257))*(sqrt(6.000000)))^(((9.704760)^((-0.050358)-(t)))-(t)))/(0))))
^((1.634223)+((7.277223)^((0.290257)-(t)))))^(0.161464))/(t)))))/(6.980522)).

Fig. 4 shows the true profiles and estimated curves and sampled data points from the genetic programming and we confirm that the GP recover the true dynamics of the S-system

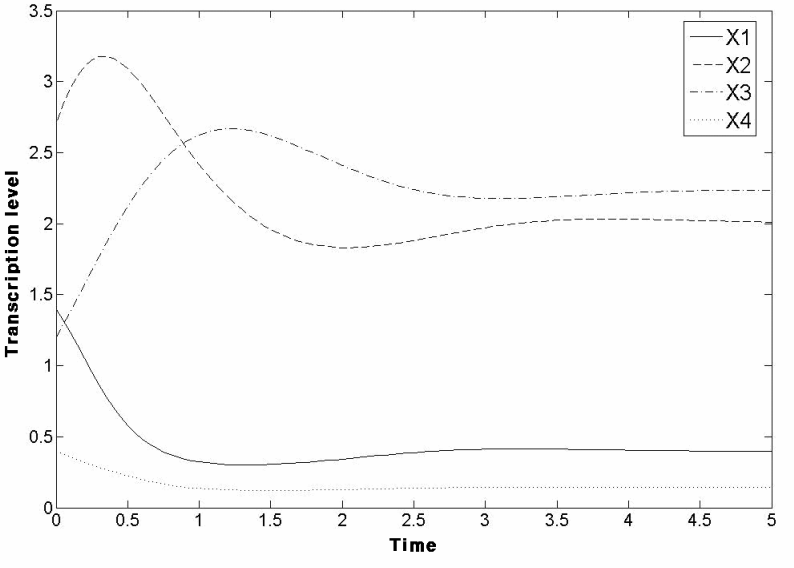
3.3 Results of the Hybrid Evolutionary Algorithms

Using 100 points obtained from the GP step, we search the S-system by the hybrid evolutionary algorithms. This step is composed with steady-state evolutionary algorithms with local optimization procedure - (1+1) evolutionary strategy. For the sparse network structures, we adopt a structural constraint which the evolved networks should satisfy. That is, each gene is assumed to be related to one or two other genes in the system. Hence, the randomly generated initial individuals and offspring by crossover and mutation operators should have one or two non-zeros elements in g and h. Crossover rate is 0.8 and mutation rate is 0.3. As a local optimization for the parameter values, (1+1) evolutionary strategy is performed for 80 fitness evaluations. The search ranges of the parameters are [0.0, 15.0] for α_i and β_i, and [-1.0, 1.0] for g_{ij} and h_{ij}. With the population size of 10^4, the proposed algorithm successfully identified the true structure after 10^6 generation. As shown in Fig. 5, we can also recover the dynamic profiles with the estimated parameters.

$$\alpha = \begin{pmatrix} 14.614 \\ 8.123 \\ 3.088 \\ 2.866 \end{pmatrix} \qquad g = \begin{pmatrix} 0.0 & 0.0 & -0.623 & 0.0 \\ 0.486 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.761 & 0.0 & 0.0 \\ 0.254 & 0.0 & 0.0 & 0.0 \end{pmatrix}$$

$$\beta = \begin{pmatrix} 12.670 \\ 3.076 \\ 5.583 \\ 5.391 \end{pmatrix} \qquad h = \begin{pmatrix} 0.398 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.753 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.486 & 0.231 \\ 0.0 & 0.0 & 0.0 & 0.443 \end{pmatrix}$$

(a) the S-system obtained by our hybrid EAs



(b) time course profiles of the identified system

Fig. 5. The results of the proposed hybrid evolutionary algorithms

4 Conclusion

We propose multi-stage evolutionary algorithms to identify gene regulatory networks efficiently with the S-system representation. We adopt the pre-processing symbolic regression step by genetic programming for avoiding the time-consuming numerical integration. We also develop hybrid genetic algorithms and modify the fitness function to identify the structure of gene regulatory networks and to estimate the corresponding parameters simultaneously without the threshold values for the sparse network structures. By applying the proposed method to the identification of an artificial genetic network, we verify its capability of finding the true S-system.

One important future work is to demonstrate the usefulness of the proposed algorithm for real experimental biological data such as the gene expression profiles from the microarrays and NMR measurements of some metabolites. As the by-product of the population diversity maintenance of our evolutionary algorithms, we will be able to attain the different plausible topologies for the network very efficiently. These can

be reliable candidates to the biologists who want to discover unknown interactions among some components in the genetic networks.

Acknowledgement

This work was supported by the Korea Ministry of Science and Technology through National Research Lab (NRL) project and the Ministry of Education and Human Resources Development under the BK21-IT Program. The ICT at the Seoul National University provided research facilities for this study.

References

1. Covert, M.W., Schilling, C.H., Famili, I. Edwards, J.S., Goryanin, I.I. Selkov, E., Palsson, B.O.: Metabolic modeling of microbial strains in silico. *Trends in Biochemical Science* 26 (2001) 179-186
2. De Jong, H.: Modeling and simulation of genetic regulatory system: a literature review. *Journal of Computational Biology* 9 (2002) 67-103
3. Stelling, J.: Mathematical models in microbial systems biology. *Current Opinion in Microbiology* 7 (2004) 513-518
4. Barabasi, A.-L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics* 5 (2004) 101-113
5. De Jong, H., Gouze, J.-L., Hernandez, C., Page, M., Sari, T., Geiselmann, J.: Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology* 66 (2004) 301-340
6. Rao, C.V., Wolf, D.M., Arkin, A.P.: Control, exploitation and tolerance of intracellular noise. *Nature* 402 (2002) 231-237
7. Voit, E.O.: *Computational Analysis of Biochemical Systems.* Cambridge University Press (2000)
8. Fogel, D.B.: *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling.* Ginn Press (1991)
9. Tominaga, D., Koga, N., Okamoto, M.: Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In: Whitley, D. et al. (Eds.), *Proceedings of the 2000 Genetic and Evolutionary Computation Conference* (2000) 251-258
10. Ando, S., Sakamoto, E., Iba, H.: Evolutionary modeling and inference of gene network. *Information Science* 145 (2002) 237-259
11. Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., Tomita, M.: Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics* 19 (2003) 643-650
12. Spieth, C., Streichert, F., Speer, N., Zell, A.: Optimizing topology and parameters of gene regulatory networt models from time-series experiments. In: Deb, K. et al. (eds.): *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*. Lecture Notes in Computer Science, Vol. 3102. Springer-Verlag (2004) 461-470
13. Almeida J.S., Voit E.O.: Neural network-based parameter estimation in S-system models of biological networks, *Genome Informatics* 14 (2003) 114-123
14. Tsai, K.-Y., Wang, F.-S.: Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics* 21 (2005) 1180-1188
15. Savageau, M.A.: *Biochmical System Analysis: A Study of Function and Design in Molecular Biology,* Addison-Wesley (1976)

16. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Natural Selection*. MIT Press (1992)
17. Bäck, T.: *Evolutionary Algorithm in Theory and Practice,* Oxford University Press (1996)
18. http://www.cs.bham.ac.uk/~cmf/GPLib/GPLib.html
19. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Eshelman, L.J. (ed.): *Proceedings of the Sixth International Conference on Genetic Algorithms* (1995) 24-31