A Cooperative Coevolutionary Approach to the Inference of Genetic Networks

Shuhei Kimura, Mariko Hatakeyama and Akihiko Konagaya
RIKEN Genomic Sciences Center
1-7-22 Suehiro-cho, Tsurumi, Yokohama, Japan
email: {skimura, marikoh, konagaya}@gsc.riken.jp

***Abstract*— We propose a new method for inferring S-system models of large-scale genetic networks. The proposed method is based on a problem decomposition strategy and a cooperative coevolutionary algorithm. The problem decomposition strategy divides the genetic network inference problem into several subproblems that are then solved using the cooperative coevolutionary algorithm. The cooperative coevolutionary algorithm is an extension of the evolutionary algorithm. It consists of several subpopulations, each of which contains competing individuals for each subproblem. As the subpopulations interact with each other through the gene expression curves, the model inferred by the proposed method computationally simulates the genetic network. The availability of the inferred model for the computational simulation is important, since the computational simulation brings about a better understanding of genetic networks. The effectiveness of the proposed method is verified through numerical experiments.**

I. Introduction

Advances in DNA microarrays and other technologies allow us to measure gene expression patterns on a genomic scale [4]. Many researchers have become interested in the inference of underlying genetic networks using the observed time-series data of gene expression patterns, and the development of this methodology has become a major topic in the bioinformatics field [18]. Numerous models have been proposed to describe networks, and numerous algorithms have been proposed for inferring individual models of genetic networks [1], [2], [5], [12], [18].

The S-system model is an ideal means of inferring genetic networks. The model possesses a rich structure capable of capturing various dynamics, and methods are available for analyzing the model [7], [20]. The S-system model is a set of non-linear differential equations of the form

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}}, \quad (i = 1, \cdots, N), \qquad (1)$$

where X_i is the state variable and N is the number of components in the network. In a genetic network, X_i is the expression level of the i-th gene and N is the number of genes in the network. α_i and β_i are multiplicative parameters called rate constants, and $g_{i,j}$ and $h_{i,j}$ are exponential parameters called kinetic orders.

Several network inference algorithms based on the S-system model have been proposed [7], [14], [20], [21]. These algorithms estimate the S-system parameters (α_i, β_i, $g_{i,j}$ and $h_{i,j}$) using observed time-series data for gene expression patterns. Because the number of S-system parameters is proportional to the square of the number of network components, the algorithms must simultaneously estimate a large number of S-system parameters if they are to be used to infer large-scale network systems containing many network components. This is why inference algorithms based on the S-system model have been applied only to small-scale networks of less than five genes.

To resolve the high-dimensionality of the genetic network inference problem in the S-system model, a problem decomposition strategy, that divides the original problem into several subproblems, has been proposed [13], [9]. This approach enables us to infer S-system models of larger-scale genetic networks. However, when the given time-series data contain the measurement noise, the inferred model cannot be used to computationally simulate a genetic network. This is one of disadvantages of the problem decomposition approach, since the computational simulation is needed to analyze and understand the genetic network.

In this paper, we propose a new method that eliminates the disadvantage of the problem decomposition approach. The proposed method simultaneously solves the decomposed subproblems using a cooperative coevolutionary algorithm [16]. In the proposed coevolutionary algorithm, all of the subproblems interact with each other through the gene expression curves which are updated when more reasonable curves are obtained. Because of this interaction, the proposed method has the ability to infer an S-system model that is ready for the computational simulation. In order to verify its effectiveness, the proposed method was applied to a genetic network inference problem containing 30 genes.

II. Genetic Network Inference Problem

A. Canonical Problem Definition

The genetic network inference problem is defined as a function optimization problem to minimize the following sum of the squared relative error [20].

$$f = \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}} \right)^2, \qquad (2)$$

where $X_{i,exp,t}$ is an experimentally observed gene expression level at time t of the i-th gene, $X_{i,cal,t}$ is a numerically

computed gene expression level acquired by solving a system of differential equations (1), N is the number of components in the network, and T is the number of sampling points of observed data.

Since $2N(N+1)$ S-system parameters must be determined in order to solve the set of differential equations (1), this function optimization problem is $2N(N+1)$ dimensional. This problem is too high-dimensional for non-linear function optimizers in cases where we try to infer S-system models of large-scale genetic networks containing many network components [12].

B. Decomposition of the Problem

Because of the high-dimensionality, function optimizers have difficulty inferring S-system models of large-scale genetic networks. To resolve this high-dimensionality, the strategy of dividing the genetic network inference problem into several subproblems was proposed [13]. In this strategy, each subproblem corresponds to each gene. The objective function of the subproblem corresponding to the i-th gene is

$$f_i = \sum_{t=1}^{T} \left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}} \right)^2, \tag{3}$$

where $X_{i,cal,t}$ is a numerically computed gene expression level at time t of the i-th gene, as described in the previous subsection. In contrast to the previous subsection, however, $X_{i,cal,t}$ is obtained by solving the following differential equation.

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} Y_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} Y_j^{h_{i,j}}, \tag{4}$$

where

$$Y_j = \begin{cases} X_j, & \text{if } j = i, \\ \hat{X}_j, & \text{otherwise.} \end{cases} \tag{5}$$

$\hat{X}_j$ is an estimated gene expression level that is acquired not by solving a differential equation, but by making a direct estimation from the observed time-series data. In this study, we call $\hat{X}_j$ an estimated gene expression curve of the j-th gene, and the local linear regression [3] is used to obtain the estimated gene expression curves.

The equation (4) is solvable when $2(N+1)$ S-system parameters (i.e., $\alpha_i, \beta_i, g_{i,1}, \cdots, g_{i,N}, h_{i,1}, \cdots, h_{i,N}$) are given. Therefore, the problem decomposition strategy divides a $2N(N+1)$ dimensional network inference problem into N subproblems that are $2(N+1)$ dimensional.

C. Use of a Priori Knowledge

The genetic network inference problem based on the S-system model may have multiple optima because the degree-of-freedom of the model is high and the observed time-series data are usually polluted by the measurement error. To increase the probability of inferring a correct S-system model, we introduced a priori knowledge of the genetic network into the objective function [9].

Genetic networks are known to be sparsely connected [19]. When an interaction between two genes is clearly absent, the S-system parameter values corresponding to the interaction (i.e., $g_{i,j}$ and $h_{i,j}$) are zero. We incorporated this knowledge into the objective function (3) by using a penalty term, as shown below [9].

$$F_i = \sum_{t=1}^{T} \left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}} \right)^2 + c \sum_{j=1}^{N-I} \left(|G_{i,j}| + |H_{i,j}| \right), \tag{6}$$

where $G_{i,j}$ and $H_{i,j}$ are given by rearranging $g_{i,j}$ and $h_{i,j}$, respectively, in descending order of their absolute values (i.e., $|G_{i,1}| \leq |G_{i,2}| \leq \cdots \leq |G_{i,N}|$ and $|H_{i,1}| \leq |H_{i,2}| \leq \cdots \leq |H_{i,N}|$). c is a penalty coefficient and I is a maximum indegree. The maximum indegree determines the maximum number of genes that directly affect the i-th gene.

The penalty term is the second term on the right hand side of the equation (6). This term forces most of the kinetic orders ($g_{i,j}$ and $h_{i,j}$) down to zero. In other words, when the penalty term is applied, most of the genes are disconnected from each other. However, when the number of genes that directly affect the i-th gene is smaller than the maximum indegree I, the term has no penalty effect. Thus, the optimum solutions to the objective functions (3) and (6) are identical when the number of interactions that affect the focused (i-th) gene is lower than the maximum indegree. In this paper, we use the equation (6) as an objective function that should be minimized.

The same parameter values used in reference [9] were used in this paper; the maximum indegree $I = 5$, and the penalty coefficient $c = 1.0$.

III. Coevolutionary Algorithm for Inference of Genetic Networks

A. Concept

The problem decomposition strategy mentioned in the previous section enables us to infer large-scale genetic networks [13], [9]. However, when the given time-series data contain the measurement noise, the inferred model cannot be used for the computational simulation of genetic networks.

The differential equation (4) is not a suitable model for analyzing genetic networks, because the perturbation in the i-th gene does not affect the expression levels of the other genes in this equation. Therefore, the set of equations (1) must be used as the model for computational simulation, while the differential equation (4) is used to compute the time courses of the gene expressions in the decomposed subproblems. Solving the equation (4) requires the gene expression curves $\hat{X}_j$ that are directly estimated from the observed data. Because of the noise in the observed data, these directly estimated curves usually differ from those obtained from solving the differential equations. Therefore, when the estimated parameters are used to solve the set of differential equations (1), the solution does not coincide with those of the equations (4). This means that the estimated parameters do not always provide us with a model that is fitted into the observed data. For this reason, we cannot use the inferred model for the computational simulation.

To use the inferred model for the computational simulation, each subproblem must be solved using the gene

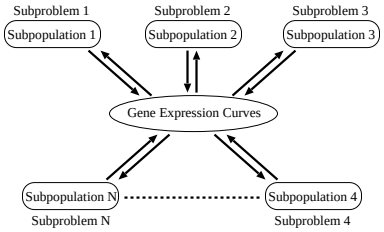

Fig. 1. The cooperative evolutionary model in this paper.

expression curves obtained from the candidate solutions of other subproblems. The cooperative coevolutionary approach [11], [15], [16] satisfies this requirement. The cooperative coevolutionary algorithm is an extension of the evolutionary algorithm [6]. It consists of several subpopulations, each of which contains competing individuals (candidate solutions) for each subproblem. The subpopulations are genetically isolated, i.e., individuals mate only with other members of their subpopulation. These subpopulations interact with each other only when the fitness values (objective values) are calculated. In this paper, the subpopulations interact with each other only through the gene expression curves that are obtained from the best individuals in the subpopulations (see Fig.1).

B. Algorithm

On the basis of the idea described above, we propose a new cooperative coevolutionary algorithm for inferring genetic networks. The following is an algorithm of the proposed method.

1) *Initialize*
   Generate N subpopulations, where N is the number of components in the genetic network. As an initial guess, estimate the gene expression curves from the observed time-series data. Set Generation $= 0$.
2) *Execution of Function Optimization*
   Execute one cycle of a function optimization algorithm on each subpopulation.
3) *Update of Estimated Gene Expression Curves*
   Update all of the estimated gene expression curves using the best individuals of the subpopulations.
4) Stop if halting criteria are satisfied. Otherwise, Generation $\leftarrow$ Generation $+ 1$ and go to step 2.

Each of these steps is described below in greater detail.

1) Step: Initialize: N subpopulations, each of which corresponds to each subproblem, are generated. Each subpopulation contains n_p individuals which are randomly created. At the same time, initial estimations of the gene expression curves $\hat{X}_j$ are made directly from the observed time-series data. In this paper, the local linear regression [3] is used to estimate the curves.

2) Step: Execution of Function Optimization: Any type of function optimizer can be applied to the decomposed subproblem. In this study, an evolutionary algorithm called GLSDC [8] is used, since it has been successfully applied to the genetic network inference problem [9], [10]. One cycle (generation) of GLSDC is performed on each subpopulation in this step.

When the algorithm calculates the fitness value of each individual on each subpopulation, the differential equation (4) is solved using the estimated gene expression curves. At this time, an initial gene expression level, as well as the S-system parameters, is required. In this study, the initial gene expression level of the i-th gene was obtained from its estimated gene expression curve, i.e., the value of $\hat{X}_i(0)$ was used for $X_{i,cal,0}$.

3) Step: Update of Estimated Gene Expression Curves: The gene expression curves of the best individuals of the subpopulations, each of which is given as a solution of the differential equation (4), are calculated. The old gene expression curves are then updated to these calculated curves.

The initial levels of the gene expression are required to calculate the gene expression curves. These values are obtained from the old curves, as described in the section III-B.2. However, since the noise in the actual time-series data corrupts the values of the initial gene expression levels, we should estimate these values in addition to the S-system parameters [10]. In this step, before the gene expression curves are updated, the initial levels of the gene expression are adjusted to fit the new calculated curves into the observed time-series data.

The adjustment of the initial gene expression level of the i-th gene is formulated as a single-dimensional function minimization problem [10]. This is because the initial gene expression level of the i-th gene is a unique variable and all of the S-system parameters are fixed to the values of the best individual. The objective function of this adjustment problem is

$$F_i^{adj} = \sum_{t=1}^{T} \gamma^{t-1} \left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}} \right)^2, \qquad (7)$$

where $X_{i,cal,t}$ is acquired by solving the differential equation (4), and γ ($0 \leq \gamma \leq 1$) is a discount parameter. Since the fixed model parameters obtained from the best individual are not always optimal, the calculated gene expression curve may differ greatly from the actual curve. When the estimated curve is incorrect, the algorithm should not fit the curve, especially the latter half of it, into the observed data. Therefore, in this study, we introduce a discount parameter γ.

A golden section search [17] is used to solve the one-dimensional function minimization problem mentioned above. The search region for this problem was set to ± 30 % of the observed initial gene expression level (i.e., $[0.7X_{i,exp,0}, 1.3X_{i,exp,0}]$). After the adjustment, the new calculated gene expression curves are substituted for the old ones. In this paper, a discount parameter $\gamma = 0.75$ was used. This value was determined through several preliminary experiments.

IV. Numerical Experiments

To show the effectiveness of the proposed method, we applied it to a genetic network inference problem consisting of 30 genes. Lacking actual biological data, we used an artificial genetic network model as a case study.

TABLE I

S-SYSTEM PARAMETERS OF THE TARGET MODEL.

α_i	1.0
β_i	1.0
$g_{i,j}$	$g_{1,14}$ =-0.1, $g_{5,1}$ =1.0, $g_{6,1}$ =1.0, $g_{7,2}$ =0.5, $g_{7,3}$ =0.4, $g_{8,4}$ =0.2, $g_{8,17}$ =-0.2, $g_{9,5}$ =1.0, $g_{9,6}$ =-0.1, $g_{10,7}$ =0.3, $g_{11,4}$ =0.4, $g_{11,7}$ =-0.2, $g_{11,22}$ =0.4, $g_{12,23}$ =0.1, $g_{13,8}$ =0.6, $g_{14,9}$ =1.0, $g_{15,10}$ =0.2, $g_{16,11}$ =0.5, $g_{16,12}$ =-0.2, $g_{17,13}$ =0.5, $g_{19,14}$ =0.1, $g_{20,15}$ =0.7, $g_{20,26}$ =0.3, $g_{21,16}$ =0.6, $g_{22,16}$ =0.5, $g_{23,17}$ =0.2, $g_{24,15}$ =-0.2, $g_{24,18}$ =-0.1, $g_{24,19}$ =0.3, $g_{25,20}$ =0.4, $g_{26,21}$ =-0.2, $g_{26,28}$ =0.1, $g_{27,24}$ =0.6, $g_{27,25}$ =0.3, $g_{27,30}$ =-0.2, $g_{28,25}$ =0.5, $g_{29,26}$ =0.4, $g_{30,27}$ =0.6, other $g_{i,j}$ = 0.0
$h_{i,j}$	1.0 if $i = j$, 0.0 otherwise.

A. Problem Setup

As a target genetic network, we used an S-system model with the parameters listed in Table I [12]. This model consists of 30 genes ($N = 30$).

15 sets of time-series data, each covering all 30 genes, were used as observed gene expression patterns. In practice, these sets of time-series data are obtained from biological experiments where some gene is disrupted or overexpressed. The sets of time-series data began from randomly generated initial values in $[0.0, 2.0]$ and were obtained by solving the set of differential equations (1) for the target model. We added 10% Gaussian noise to the time-series data, in order to simulate the measurement noise that often corrupts the observed data obtained from actual measurements of gene expression patterns. We assigned 11 sampling points for the time-series data on each gene in each set, assuming that it would be difficult to measure the gene expression patterns more times in actual biological experiments. Thus, the observed time-series data for each gene consisted of $15 \times 11 = 165$ sampling points.

As this network model contains 30 genes, we have to estimate $2 \times 30 \times (30 + 1) = 1860$ S-system parameters in order to infer the network. In addition, we have to estimate all of the initial levels of the gene expression, which totaled $30 \times 15 = 450$. Therefore, $1860 + 450 = 2310$ parameters must be estimated in this problem.

B. Experimental Setup

The following parameters in GLSDC [8] were used in this paper; the population size n_p is $3n$, where n is the dimension of the search space and n equals 62 in this section; the number of children generated by the crossover per selection n_c is 10; and the number of applied the converging operations N_0 is n_p.

Five runs were carried out. Each run was continued until the number of generations reached 75. The search regions of the parameters were $[0.0, 3.0]$ for α_i and β_i, $[-3.0, 3.0]$ for $g_{i,j}$ and $h_{i,j}$, and $[0.7X_{i,exp,0}, 1.3X_{i,exp,0}]$ for the initial levels of the gene expression described in the section III-B.3. The experiments were executed in parallel on a PC cluster.

In order to reduce the computational cost, we applied a structure skeletalizing technique [20]. This technique assigns a value of zero to the kinetic orders ($g_{i,j}$ and $h_{i,j}$) whose absolute values are less than the given threshold δ_s. Structure skeletalizing reduces the computational cost because the exponential calculation of the equation (4) can be omitted when the kinetic orders are zero. In this paper, the given threshold δ_s was 1.0×10^{-3}.

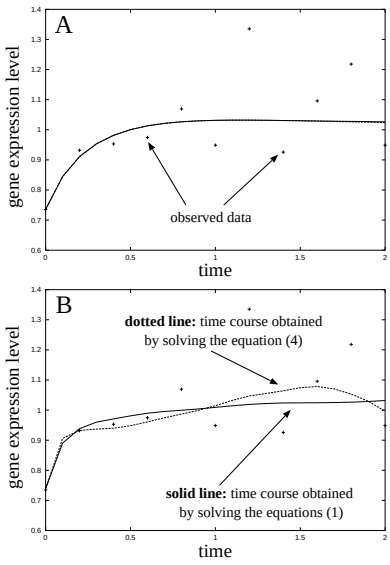

Fig. 2. Samples of calculated time courses obtained from A) the proposed coevolutionary approach, and B) the problem decomposition approach. Solid line: the solution of the set of differential equations (1) where the estimated values are used as the model parameters. Dotted line: time course obtained at the end of the search, i.e., the solution of the differential equation (4). +: noisy time-series data given as the observed data.

To confirm the effectiveness of the coevolutionary approach, we compared its results to those of a non-coevolutionary method that did not consider the interactions between decomposed subproblems. In this paper, this non-coevolutionary method is referred to as the problem decomposition approach [10].

C. Results

Fig. 2 shows the calculated gene expression curves obtained from the method with and without the coevolution. As shown in Fig. 2A, when the proposed coevolutionary approach was applied, the time course obtained by solving the set of equations (1) was almost identical to that obtained by solving the equation (4). Therefore, even when the values estimated by our method are used as the model's parameters, the system of the differential equations (1) produces a model that is fitted into the observed time-series data. On the other hand, the calculated time courses of the problem decomposition approach differed greatly (see Fig. 2B). In this case, we cannot use the set of differential equations (1) as the mathematical model because it may not be what we are trying to infer.

Typical results are shown in Tables II and III. The S-system parameters estimated with and without the coevolution are listed for the 11th, 20th and 24th subproblems. The tables show that both methods failed to infer some of the interactions present in the target model, and they inferred several erroneous interactions that had absolute parameter values too large to ignore. As it is difficult to estimate true gene expression curves under noisy environment, the failure to infer the correct

TABLE II

SAMPLES OF S-SYSTEM PARAMETERS ESTIMATED BY THE PROPOSED METHOD FOR THE 11TH, 20TH AND 24TH SUBPROBLEMS.

Estimated S-system parameters for the 11th subproblem		
$\alpha_{11} = 0.719$	$g_{11,4} = 0.401$, $g_{11,7} = -0.253$, $g_{11,10} = -0.340$, $g_{11,11} = -0.106$, $g_{11,22} = 0.579$,	other $g_{11,j} = 0.00$
$\beta_{11} = 0.728$	$h_{11,7} = 0.0408$, $h_{11,10} = -0.677$, $h_{11,11} = 1.25$, $h_{11,14} = 0.0550$, $h_{11,30} = 0.0880$,	other $h_{11,j} = 0.00$
Estimated S-system parameters for the 20th subproblem		
$\alpha_{20} = 0.850$	$g_{20,3} = -0.250$, $g_{20,8} = 0.233$, $g_{20,15} = 0.686$, $g_{20,20} = -0.202$, $g_{20,26} = 0.293$,	other $g_{20,j} = 0.00$
$\beta_{20} = 0.826$	$h_{20,17} = -0.201$, $h_{20,18} = -0.0672$, $h_{20,20} = 0.935$, $h_{20,27} = -0.104$, $h_{20,29} = -0.0848$,	other $h_{20,j} = 0.00$
Estimated S-system parameters for the 24th subproblem		
$\alpha_{24} = 0.474$	$g_{24,1} = -0.255$, $g_{24,15} = -0.442$, $g_{24,17} = 0.541$, $g_{24,23} = 0.820$, $g_{24,24} = -0.298$,	other $g_{24,j} = 0.00$
$\beta_{24} = 0.472$	$h_{24,16} = -0.115$, $h_{24,17} = 0.196$, $h_{24,23} = 0.548$, $h_{24,24} = 2.23$, $h_{24,28} = 0.185$,	other $h_{24,j} = 0.00$

TABLE III

SAMPLES OF S-SYSTEM PARAMETERS ESTIMATED BY THE PROBLEM DECOMPOSITION APPROACH.

Estimated S-system parameters for the 11th subproblem		
$\alpha_{11} = 0.609$	$g_{11,4} = 0.503$, $g_{11,10} = -0.352$, $g_{11,11} = -0.289$, $g_{11,16} = 0.0813$, $g_{11,22} = 0.701$,	other $g_{11,j} = 0.00$
$\beta_{11} = 0.624$	$h_{11,7} = 0.352$, $h_{11,10} = -0.730$, $h_{11,11} = 1.47$, $h_{11,22} = 0.214$, $h_{11,30} = 0.0792$,	other $h_{11,j} = 0.00$
Estimated S-system parameters for the 20th subproblem		
$\alpha_{20} = 0.729$	$g_{20,15} = 0.498$, $g_{20,20} = -0.200$, $g_{20,26} = 0.214$, $g_{20,27} = 0.169$, $g_{20,29} = 0.164$,	other $g_{20,j} = 0.00$
$\beta_{20} = 0.712$	$h_{20,3} = 0.182$, $h_{20,8} = -0.237$, $h_{20,15} = -0.246$, $h_{20,17} = -0.180$, $h_{20,20} = 1.00$,	other $h_{20,j} = 0.00$
Estimated S-system parameters for the 24th subproblem		
$\alpha_{24} = 0.412$	$g_{24,3} = 0.523$, $g_{24,5} = 0.261$, $g_{24,21} = 0.212$, $g_{24,22} = -0.413$, $g_{24,24} = -0.340$,	other $g_{24,j} = 0.00$
$\beta_{24} = 0.434$	$h_{24,1} = 0.813$, $h_{24,20} = -0.214$, $h_{24,24} = 2.34$, $h_{24,29} = 0.500$, $h_{24,30} = -0.158$,	other $h_{24,j} = 0.00$

TABLE IV

S-SYSTEM PARAMETERS OF THE SMALL-SCALE TARGET MODEL.

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.0	0.0	0.0	1.0	0.0	-1.0	10.0	2.0	0.0	0.0	0.0	0.0
2	10.0	2.0	0.0	0.0	0.0	0.0	10.0	0.0	2.0	0.0	0.0	0.0
3	10.0	0.0	-1.0	0.0	0.0	0.0	10.0	0.0	-1.0	2.0	0.0	0.0
4	8.0	0.0	0.0	2.0	0.0	-1.0	10.0	0.0	0.0	0.0	2.0	0.0
5	10.0	0.0	0.0	0.0	2.0	0.0	10.0	0.0	0.0	0.0	0.0	2.0

TABLE V

S-SYSTEM PARAMETERS ESTIMATED BY THE PROPOSED METHOD.

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.917	-0.009	-0.003	1.019	-0.017	-1.014	9.922	2.021	-0.009	0.002	-0.009	-0.009
2	10.030	1.995	0.002	-0.002	0.006	-0.001	10.026	0.002	1.995	-0.002	0.002	0.000
3	9.851	-0.005	-0.991	-0.004	-0.003	0.002	9.835	-0.004	-0.993	2.036	-0.010	0.002
4	8.020	-0.007	0.006	2.000	-0.002	-0.998	10.054	0.001	0.003	0.008	1.988	0.007
5	9.875	-0.002	0.003	0.018	2.015	-0.020	9.892	0.004	0.002	0.008	-0.010	2.017

interactions occurred even when the proposed method was applied. The failure to infer the correct interactions, however, does not seriously hinder our investigation, as the inferred model is intended mainly for use by biologists as a tool for generating hypotheses and for facilitating the design of experiments. The necessary interactions that were not correctly inferred should be added, and the wrong interactions should be removed in either of two ways, by using more sets of time-series data obtained from additional biological experiments, or by using further a priori knowledge about the genetic network. The availability of the model inferred by the proposed method for the computational simulation is also convenient for the model refinement.

The model inferred by the proposed method contained 58.4 ± 2.1 true-positive interactions and 241.6 ± 2.1 false-positive interactions on average. In addition, our method failed to infer an average of 9.6 ± 2.1 interactions that were present in the target model (i.e., the number of the false-negative interactions was 9.6 ± 2.1). On the other hand, in the experiment of the problem decomposition approach, the numbers of true-positive, false-positive and false-negative interactions averaged 57.6 ± 2.3, 242.4 ± 2.3 and 10.4 ± 2.3, respectively. The proposed method seems to slightly enhance the probability of finding the correct interactions. This may be because the proposed method updates the estimated gene expression curves $\hat{X}_j$. In this study, the algorithm uses the estimated gene expression curves to solve the decomposed subproblems. Therefore, in order to increase the probability of finding the correct interactions, these estimated gene expression curves must be precise. Because the proposed coevolutionary approach updates these curves, their precision may be improved through searches. Solving this inference problem required about 45.3

hours on the PC cluster (Pentium III 933MHz $\times$ 32 CPUs).

V. Discussion

In the experiments described above, the proposed method failed to estimate the correct S-system parameters because of the noise in the given data. When a sufficient amount of noise-free data is available, however, our method can correctly estimate the S-system parameters.

We used a small-scale model as the target genetic network. The target model consisted of 5 genes ($N = 5$). Table IV lists the model's parameters [7]. As a sufficient amount of observed data, we gave 15 sets of noise-free time-series data. The search regions of the parameters were $[0.0, 20.0]$ for α_i and β_i, $[-3.0, 3.0]$ for $g_{i,j}$ and $h_{i,j}$, and $[0.7X_{i,exp,0}, 1.3X_{i,exp,0}]$ for the initial levels of the gene expression. Other experimental conditions were identical to those described in the previous section. This problem has $2 \times 5 \times (5+1) + 5 \times 15 = 135$ parameters to be estimated.

The estimated S-system parameters are listed in Table V. As can be seen from the table, our method was unable to estimate the parameter values with perfect precision. However, they were precise enough to biologically interpret the network.

In this experiment, the effectiveness of the proposed method was confirmed by estimating both the initial gene expression levels and the S-system parameters. In practice, however, it is not necessary to estimate the initial levels of the gene expression when the observed data seem to contain no measurement error. When the initial gene expression levels do not need to be estimated, the estimated parameters are more precise since the problem contains fewer unknown parameters.

VI. Conclusion

In this paper, we proposed a new method for inferring the S-system models of large-scale genetic networks. The proposed method uses the problem decomposition strategy to divide the genetic network inference problem into several subproblems. The decomposed subproblems are then solved using the cooperative coevolutionary algorithm. Because the decomposed subproblems interact with each other through their calculated gene expression curves, the inferred model can be used in the computational simulation. This feature is important because the computational simulation provides us with a better understanding of genetic networks. Through numerical experiments, we showed that the proposed method slightly enhanced the probability of finding the correct interactions of a network. Updating the gene expression curves also seems to enhance the probability of inferring a correct network structure.

When attempting to analyze actual DNA microarray data, many hundreds or thousands of genes must be handled. This task lies far beyond the powers of the coevolutionary method proposed in this paper. As things stand at present, the high-dimensionality of the problems render this method incapable of inferring networks containing more than a few hundred network components. One possible strategy to improve its inference capability is to use the clustering technique to identify genes with similar expression patterns and group them together [5]. By treating groups of similar genes as single network components, the proposed coevolutionary method may be capable of analyzing systems containing many hundreds of genes. In the future, we will attempt to combine the the proposed method and the clustering technique.

References

[1] T. Akutsu, S. Miyano and S. Kuhara, "Inferring Qualitative Relations in Genetic Networks and Metabolic Pathways," *Bioinformatics*, Vol.16, No.8, pp. 727-734, 2000.

[2] T. Chen, H.L. He and G.M. Church, "Modeling Gene Expression with Differential Equations," *Proc. of Pacific Symposium on Biocomputing*, 4, pp. 29-40, 1999.

[3] W.S. Cleveland, "Robust Locally Weight Regression and Smoothing Scatterplots," *J. of American Statistical Association*, Vol.79, No.368, pp. 829-836, 1979.

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

[5] P. D'haeseleer, S. Liang and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, Vol.16, No.8, pp. 707-726, 2000.

[6] J.H. Holland, "Adaptation in Natural and Artificial Systems," The University of Michigan Press, Ann Arbor, 1975.

[7] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinformatics*, Vol.19, No.5, pp. 643-650, 2003.

[8] S. Kimura and A. Konagaya, "High Dimensional Function Optimization using a new Genetic Local Search suitable for Parallel Computers," *Proc. of 2003 Conference on Systems, Man & Cybernetics*, pp. 335-342, 2003.

[9] S. Kimura, M. Hatakeyama and A. Konagaya, "Inference of S-system Models of Genetic Networks using a Genetic Local Search," *Proc. of 2003 Congress on Evolutionary Computation*, pp. 631-638, 2003.

[10] S. Kimura, M. Hatakeyama and A. Konagaya, "Inference of S-system Models of Genetic Networks from Noisy Time-series Data," *Chem-Bio Informatics Journal*, Vol.4, No.1, pp. 1-14, 2004.

[11] Y. Liu, X. Yao, Q. Zhao and T. Higuchi, "Scaling Up Fast Evolutionary Programming with Cooperative Coevolution," *Proc. of 2001 CEC*, pp. 1101-1108, 2001.

[12] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe and Y. Eguchi, "Development of a System for the Inference of Large Scale Genetic Networks," *Proc. of PSB*, 6, pp. 446-458, 2001.

[13] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, Y. Inamura and Y. Eguchi, "Inference of Genetic Network Using the Expression Profile Time Course Data of Mouse P19 Cells," *Genome Informatics*, Vol.13, pp. 382-383, 2002.

[14] R. Morishita, H. Imade, I. Ono, N. Ono and M. Okamoto, "Finding Multiple Solutions Based on An Evolutionary Algorithm for Inference of Genetic Networks by S-system," *Proc. of 2003 CEC*, pp. 615-622, 2003.

[15] M.A. Potter and K.A. De Jong, "A Cooperative Coevolutionary Approach to Function Optimization," *Proc. of PPSN 3*, pp. 249-257, 1994.

[16] M.A. Potter and K.A. De Jong, "Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents," *Evolutionary Computation*, Vol.8, No.1, pp. 1-29, 2000.

[17] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, "Numerical Recipes in C second edition," Cambridge University Press, 1995.

[18] E. Sakamoto and H. Iba, "Inferring a System of Differential Equations for a Gene Regulatory Network by using Genetic Programming," *Proc. of 2001 CEC*, pp. 720-726, 2001.

[19] D. Thieffry, A.M. Huerta, E. Pérez-Rueda and J. Collado-Vides, "From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli," *BioEssays*, Vol.20, pp. 433-440, 1998.

[20] D. Tominaga, N. Koga and M. Okamoto, "Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem," *Proc. of GECCO 2000*, pp. 251-258, 2000.

[21] T. Ueda, I. Ono and M. Okamoto, "Development of System Identification Technique Based on Real-Coded Genetic Algorithm," *Genome Informatics*, Vol.13, pp. 386-387, 2002.