Inference of Gene Regulatory Networks Using S-system and Differential Evolution

Nasimul Noman
Department of Frontier Informatics
The University of Tokyo
Chiba 277-8561, Japan
noman@iba.k.u-tokyo.ac.jp

Hitoshi Iba
Department of Frontier Informatics
The University of Tokyo
Chiba 277-8561, Japan
iba@iba.k.u-tokyo.ac.jp

ABSTRACT

In this work we present an improved evolutionary method for inferring S-system model of genetic networks from the time series data of gene expression. We employed Differential Evolution (DE) for optimizing the network parameters to capture the dynamics in gene expression data. In a preliminary investigation we ascertain the suitability of DE for a multimodal and strongly non-linear problem like gene network estimation. An extension of the fitness function for attaining the sparse structure of biological networks has been proposed. For estimating the parameter values more accurately an enhancement of the optimization procedure has been also suggested. The effectiveness of the proposed method was justified performing experiments on a genetic network using different numbers of artificially created time series data.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; I.2.1 [**Applications and Expert Systems**]: Medicine and science; G.1.6 [**Optimization**]: Global Optimization

General Terms

Experimentation, Algorithms, Performance

Keywords

S-system, Gene regulatory network, Differential Evolution, Microarray data, Reverse Engineering

1. INTRODUCTION

Gene regulatory networks are complex biological systems which are dynamic and highly nonlinear in nature and comprise of many interacting components. Because of poor understanding of these biological components, their dependencies, interaction and nature of regulation grounded on molecular level, it is difficult to model these complex mechanisms mathematically [13]. However the tremendous advancement in molecular biology along with the help of cutting edge technologies such as DNA microarrays enables us cell-wide monitoring of gene and protein expression. And these massive amounts of biological data have grown interest among many researchers to use the model-based identification methods for inferring the possible regulatory architectures in genetic networks.

A genetic network model aims to capture the interrelated regulatory mechanisms among genes. Several genetic network models have been proposed, which integrate biochemical pathway information and expression data to trace genetic regulatory interactions [1, 9, 12, 3]. The modeling spectrum ranges from abstract Boolean descriptions to detailed Differential Equation based models, where every representation has its advantages and limitations. Given a dynamic model of gene interactions, the problem of gene network inference is equivalent to learning the structural and functional parameters from the time series representing the gene expression kinetics, i.e. the network architecture is reverse engineered from its activity profiles.

Among the familiar models for describing biochemical networks, a well studied one is S-system which is rich enough to reasonably capture the nonlinearity of genetic regulation [14]. S-system model is based on a set of non-linear ordinary differential equation in which the component processes are characterized by power-law functions of the form

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

where N is the number of network components or reactants $(X_i), i, j(1 \leq i, j \leq N)$ are suffixes of components. The terms g_{ij} and h_{ij} represent interactive affectivity of X_j to X_i. The first term represents all influences that increase X_i, whereas the second term represents all influences that decrease X_i. From the biological point of view, the two terms in right-hand side of (1) represent the productive and inhibitory regulation respectively, influencing the variable at the left hand side of the equation. The parameters that define the S-system are: $\{\alpha, \beta, g, h\}$. In a biochemical engineering context, the non-negative parameters α_i , β_i are called *rate constants*, and real-valued exponents g_{ij} and h_{ij} are referred to as *kinetic orders*.

Since the details of the molecular mechanisms that govern interactions among system components are not substantially known or well understood, the description of these processes requires a representation that is general enough to capture

GECCO'05, June 25–29, 2005, Washington, DC, USA

the essence of the experimentally observed response. The strength of S-system model is its structure which is rich enough to satisfy these requirements and to capture all relevant dynamics; an observed response (dynamic response) may be monotone or oscillatory, it may contain limit cycles or exhibit deterministic chaos [19]. Furthermore, the simple homogeneous structure of S-system has a great advantage in terms of system analysis and control design, because the structure allows analytical and computational methods to be customized specifically for this structure [6].

Tominaga et al. [19] formulated the S-system based gene network estimation as an optimization problem and they used *Genetic Algorithm* (GA) to estimate model parameters. Since methods for finding analytic solution for this problem is almost impracticable, use of *Evolutionary Computation* (EC) has become more feasible and popular approach among researchers [2, 7, 11, 15] . But because of high complexity of the problem these works still could not estimate the network topology and parameter values with high accuracy.

In this paper we propose an improved algorithm that finds the parameter values for S-system model based networks with higher accuracy using *Differential Evolution* (DE). An extension of the function for evaluating the estimated parameter set is also suggested. The effectivity of DE for a complex problem like S-system based genetic network inference was probed in a preliminary study. Then we used the modified fitness function and an enhanced algorithm for estimating the correct network topology and parameter values. Numerical experiments show that the proposed enhancements attain higher accuracy and efficiency compared to conventional methods. The paper is organized as follows. In the next section we present a brief overview of DE and the preliminary experiment to check the suitability of DE for gene network estimation. In Section 3 our proposed algorithm for parameter estimation of S-system model based gene networks is presented. Section 4 reports the experiments to verify the effectiveness of the proposed method. The experimental results are presented in Section 5. Finally a brief discussion is presented in Section 6 followed by the conclusion in Section 7.

2. OPTIMIZING S-SYSTEM MODEL USING DE

2.1 Basic Problem Definition

In the form of optimization problem each set of parameters estimated for the S-system model of a genetic network is evaluated as follows. Suppose that $X_{i,cal,t}$ is gene expression level of gene X_i at time t calculated numerically by solving the system of differential equation of (1) for the estimated parameter set, and $X_{i,exp,t}$ represents the experimentally observed gene expression level of X_i at time t. Sum of the relative squared error between $X_{i,cal,t}$ and $X_{i,exp,t}$ is taken as the relative standard error f for fitness estimation [19]

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

where N is the number of state variables, T is the number of sampling points of the experimental data. The problem is to find a set of parameters that minimizes f.

The problem has the difficulty of high-dimensionality, since $2N(N+1)$ S-system parameters must be determined in order to solve the set of differential equations (1). And estimation of parameters for a $2N(N+1)$ dimensional function optimization problem often causes bottlenecks and fitting the model to experimentally observed responses (time course of relative state variables or reactants) is never straightforward and is almost always difficult.

2.2 Differential Evolution

Differential Evolution (DE) is an effective, efficient and robust optimization method [17] capable of handling nondifferentiable, nonlinear and multimodal objective functions. The beauty of this algorithm is its simple and compact structure, which uses a stochastic direct search approach and utilizes common concepts of EAs. Furthermore DE uses few, easily chosen, parameters and surprisingly works very reliably with excellent overall results for a wide set of benchmark functions and real-world problems. Experimental results have shown that DE has good convergence properties and outperforms other well known EAs [17][16]. Because of these admirable properties we have chosen DE as optimizer for gene network inference problem.

In DE new individuals are generated by the combination of randomly chosen individuals from the population. Specifically, for each individual x^i_G, $i = 1, \cdots, P$, where G denotes the current generation, a new individual $y^i_{G+1}$is generated according to the following equation

$$y^i_{G+1} = x^j_G + F(x^k_G - x^l_G) \qquad (3)$$

where j, k and l are random integers such that j, k and $l \epsilon \{1, \cdots, P\}$ and $i \neq j \neq k \neq l$ and F is called *scaling factor* or *amplification factor*. This operation is similar to what is commonly known as *mutation* to EC community. In order to achieve higher diversity the mutated individual y^i_{G+1} is mated with the current population member x^i_G using a *crossover* operation to generate the *offspring* or *trial individual* x^i_{G+1}. The genes of x^i_{G+1} are randomly inherited from x^i_G or y^i_{G+1} determined by a parameter called *crossover factor* CF, i.e. if $r \leq CF$ (where r is a uniform random number in $[0, 1]$) then it is inherited from x^i_G otherwise from y^i_{G+1}. Finally the offspring is evaluated and replaces its parent x^i_G in next generation if and only if its fitness is better than that of its parent. This is the *selection* process.

Recently Fan and Lampinen have proposed a Trigonometric Mutation Operation (TMO) for DE to accelerate its convergence rate and robustness [4] which is defined as

$$y^i_{G+1} = (x^j_G + x^k_G + x^l_G)/3 + (p_k - p_j)(x^j_G - x^k_G) + (p_l - p_k)(x^k_G - x^l_G) + (p_j - p_l)(x^l_G - x^j_G) \qquad (4)$$

where

$$p_j = |f(x^j_G)|/p' \qquad p_k = |f(x^k_G)|/p' \qquad p_l = |f(x^l_G)|/p'$$
$$\text{and} \qquad p' = |f(x^j_G)| + |f(x^k_G)| + |f(x^l_G)|$$

This TMO is applied with a probability of M_t along with the regular mutation operation given by (3) and this modified DE algorithm is called Trigonometric mutation DE (TDE). Since the trigonometric mutation operation is a rather greedy search operator, this modification of the DE algorithm makes it possible to straightforwardly adjust the balance between the convergence rate and the robustness through the newly introduced parameter, M_t. The greediness of the algorithm can be tuned conveniently by increasing or decreasing M_t.

2.3 Optimization Performance of DE

To evaluate the performance of DE for a deceptive and highly multimodal search space, we perform preliminary experiments with two artificial gene network with $N = 5$. For each network we created artificial time series by integrating the S-system from $t_0 = 0$ to t_{max} using fourth order Runge-Kutta algorithm and taking equidistant sample points. These artificial microarray data sets were reengineered by DE and TDE algorithm. The time series data used for optimization of the first network is shown in Figure 1. Due to the fact that gene networks in nature are sparse systems, we created this network randomly with a maximum cardinality of $\kappa \leq 3$. The dynamics for the second network (shown in Figure 2) was created simulating the network model in Table 1 using the initial gene expression levels of the first data set shown in Table 2. From each time-course for the first network 25 sample points were used for optimization and for second network 50 samples were used from each time-course.

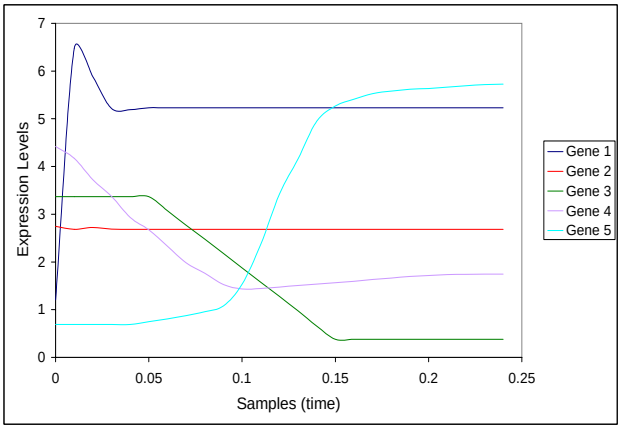

Figure 1: Target time dynamics of first gene network

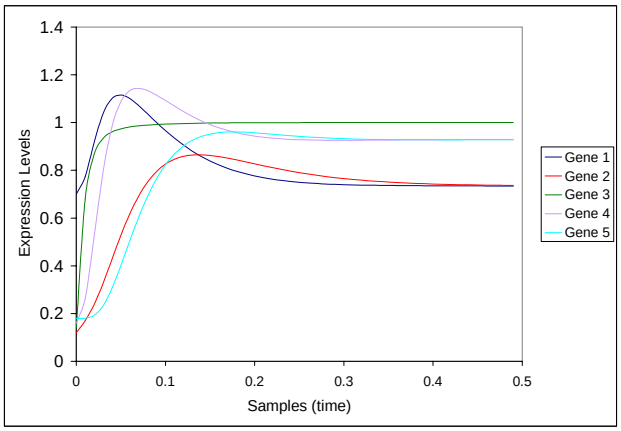

Figure 2: Target time dynamics of second gene network

To compare the results with established inference methods we also used a standard *Evolutionary Strategy* (ES) and *Genetic Algorithm* (GA) to optimize the networks. Inference by a standard ES was performed using a (μ, λ)-ES with $\mu = 10$ parents and $\lambda = 100$ offsprings together with a Covariance Matrix Adaptation (CMA-1) mutation operator without recombination [5]. On the other hand GA employed minimal generation gap (MGG) model with Simplex Crossover (SPX) and static Gaussian mutation operation [20]. For preserving the best solutions we extended the original model of GA in [20] with elitist strategy where the percentage of elite individuals was itself mutated with generation. The implementation details about the used GA and ES models could be found in [20] and [5] respectively.

In order to make the performance comparison fairer we used the same sets of initial random populations for evaluating different algorithms. Each experiment was repeated 20 times in this fashion. Maximum number of evaluations allowed for each algorithm was 1,000,000. The parameter setting for DE and/or TDE was as follows: $F = 0.5$, $CF = 0.8$, and $M_t = 0.05$. Population size for DE, TDE and GA was chosen 600 and ES was initialized by 10 random solutions among these 600 individuals. For other parameters of GA and ES values were chosen as suggested in [20] and [5] respectively.

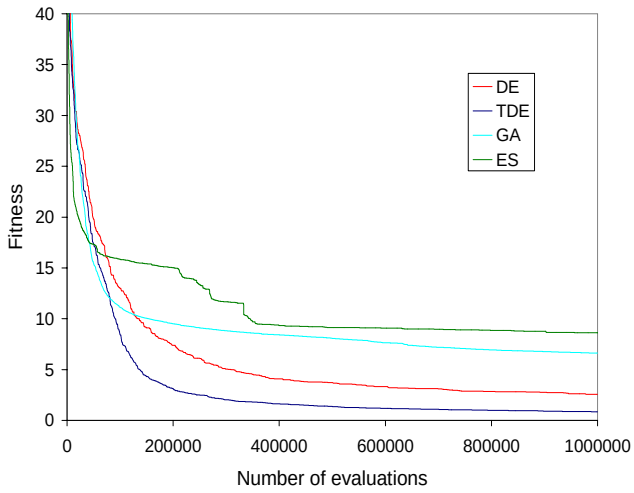

Figure 3: Convergence course for first gene network

The fitness transitions for different algorithms in these two experiments are shown in Figure 3 and Figure 4 respectively. Though not shown in the graph of Figure 3, ES started with worse fitness value due to the smaller population size, whereas the other three schemes, initialized with same and equal number of individuals, started with same fitness value. ES started with a steeper convergence curve in the beginning, but started to become almost stagnate after approximately 400,000 evaluations on average. In case of GA it was progressing slowly compared to ES in the beginning but continue to improve fitness value (though very slowly) until almost 750,000 evaluations on average and was able to reach a better fitness value compared to ES. In contrast to this, both DE strategies were successful to reach a much better fitness value compared to that of ES and GA. The basic DE strategy showed slowest convergence rate in the beginning but was steady in reaching a very good fitness value. On the other hand TDE algorithm started with a convergence rate almost like GA but continued to improve the fitness value up to a point where all other strategies convergence curve became almost horizontal. More or less similar relative performance was observed in the second experiment

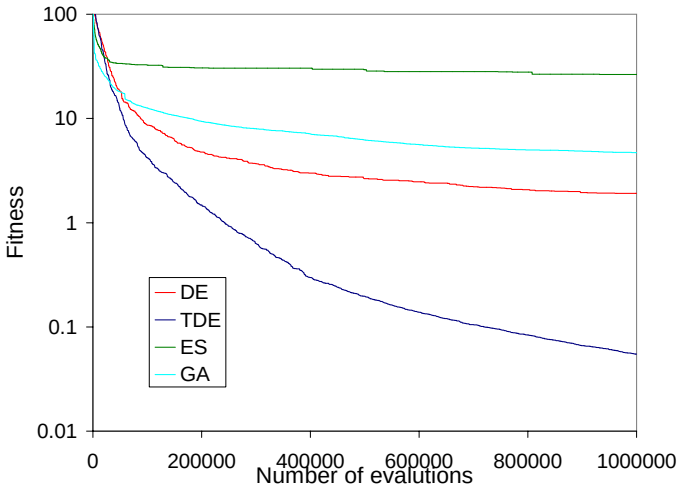

Figure 4: Convergence course for second gene network

as shown in Figure 4. Here one important observation is, use of more samples from each time course has improve the performance of TDE much compared to other three algorithms. Figure 4, plotted in a logarithmic scale, suggests that TDE strategy continues to improve the fitness value and seems not to be converged at the end of the optimization, which suggests even better results, with a higher number of fitness evaluations, is possible. These results suggest that the TDE algorithm is able to find a network structure as well as parameter values with higher accuracy that is similar to the correct one. After being ascertained by these results we employed Trigonometric mutation Differential Evolution (TDE) as optimizer in our algorithm for estimating parameters of S-System model based genetic networks (See Section 3).

3. PROPOSED METHOD

3.1 Concept

Continued development in the community has led to the discovery of two major pitfalls for S-system based gene regulatory network estimation. Firstly, use of a single time series for a gene is not sufficient to identify a unique solution for a complex system like gene network. This is because, it is only one path in a phase diagram and from such a single path no general conclusions about the overall behavior of the dynamic system can be drawn [18]. The other one is identifying the sparse structure of the biological networks. Because of the deceptive nature of the problem, solutions often converge to different local minima each of which reproduces almost same time-course. So any method attempting to capture the time dynamics only, fail to obtain the skeletal structure. Use of multiple time courses is being considered as a remedy for the first ambiguity. Use of an additional term called *pruning term* or *penalty term* for augmentation of the fitness equation was very successful to deal with the second difficulty [7][8]. Use of these two techniques were key points in our method as well as we have applied a iterative procedure to identify the skeletal structure progressively which Kikuchi et al. have called gradual optimization strategy [7].

3.2 Fitness Function

For identifying the sparse network structure which is more usual for real biological systems we extended the basic fitness function of (2) by adding a term based on Laplacian regularization term. The augmented fitness function takes the form of

$$f = \sum_{i=1}^{N}\sum_{t=1}^{T}\left\{\left(\frac{X_{i,cal,t} - X_{i,exp,t}}{X_{i,exp,t}}\right)^2\right\} + \frac{1}{N}\left\{\sum_{ij}|g_{ij}| + \sum_{ij}|h_{ij}|\right\} \quad (5)$$

In our algorithm this new fitness function (5) is used for obtaining rough skeletal structure of the network and the basic fitness (2) function is used to find the more accurate parameter values for the network, based on the structure estimated by (5). In our search we look for a set of parameters which will minimize the fitness value f expressed by (5). So the presence of the second term will force all the kinetic orders (g_{ij} and h_{ij}) towards zero. Therefore, while searching, the first term (the original fitness function) will try to find a set of parameters which will reproduce the time course, on the other hand the second term will try to find a set of parameters which will minimize it. And because of their join activity search will be directed to the sets of parameters which will have many zero values for g_{ij} and h_{ij}, representing skeletal structures. And by applying progressive refinement for identifying the complete sparse structure, the target network will be attained. In this fitness function our originality is the use of the reciprocal of network dimension as coefficient in penalty term. The reason for using this coefficient is to reduce the effect of the penalty term in total fitness as the network dimension grows. As the dimension of the genetic network increases the penalty term as well as the fitness value will also increase. Since we search for the minimum value of the fitness function, the search may be misguided because of the presence of large value of penalty term. Therefore to keep the effect of the penalty term indifferent with the increase of network components, we proposed the fitness function in (5).

3.3 Algorithm

As mentioned earlier, the S-system model of a sparse biological network will have many zero-valued parameters which have no effect in generating the system dynamics. Therefore if it is possible to identify these parameters then it will be easier to estimate the other parameters of S-system model more accurately. Since it is difficult to optimize all the parameters of S-system model simultaneously (because of the deceptive nature of the problem and presence of many local minima) we applied an iterative procedure to gradually detect those parameter values which become almost zero. Once we can identify a parameter as zero in some iteration then in the next iteration starting from the beginning we have to optimize fewer parameters and hope to identify other zero-valued parameters, if there is any. And this procedure is repeated until no more zero valued parameter could be detected.

For identifying the correct network structure and estimating accurate parameter values, in each iteration of our algorithm we optimized the S-system model parameters using three steps described below:

Table 1: S-system parameters for target network

i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
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

(S1) The first step, which we call **Structure Identification Step**, optimizes the parameters of the whole genetic network using the fitness function (5) for identifying the skeletal structure. In the first iteration of optimization, this step starts with a population where each parameter for each of its individuals is randomly initialized. In the subsequent iterations, optimization begins with random population where each individual will have zero values for all those parameters which were identified as zero in previous iteration.

(S2) In the second step of optimization we use gene-wise optimization for parameters using fitness function (2). In other words, in this step, starting with the resulting population of previous step as initial population, we try to tune the parameters for each gene separately, keeping the parameters of other genes fixed. The purpose of this step is to adjust the parameter values more accurately based on the identified structure in previous step, so we call it **Fine Tuning Step**.

(S3) The final step is the **Synchronization Step** where we try to compensate for any over tuning due to gene-wise adjustment in Step S2. This is done by optimizing all the parameters of the whole gene network using fitness function (2) and the final population of second step as the initial population. In each of these steps we employed TDE as optimizer, for finding a suitable parameter set for the target network.

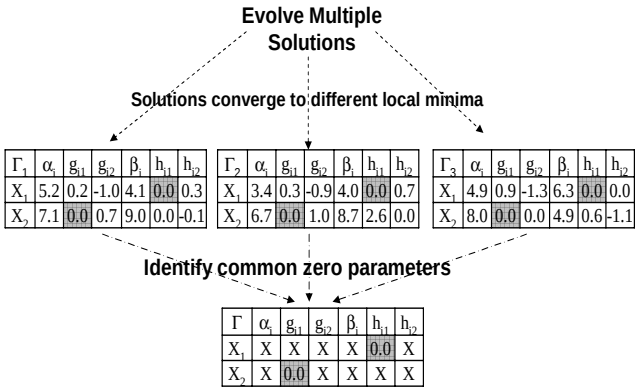

Figure 5: Method for escaping local minima

Escaping Local Minima

At the end of each iteration of our optimization algorithm we will find a solution for the target S-system model which possibly converged to a local optimum and failed to attain the actual parameter set. And because of local convergence it may lose some essential regulatory interaction among the genes. In other words we can say, due to convergence to local minima some parameter value could go down to zero, which is not actually zero in the target parameter set. In order to get rid of these incorrect identifications of parameter values as zero, due to convergence in local minima, we evolve multiple solutions $(\Gamma_1, \Gamma_2, \cdots, \Gamma_\rho)$ in each iteration. Initialized with different random populations each of these solutions will possibly resolve to different local optima but will have same essential parameters. So at the end of each iteration we nullify only those regulatory interactions as zero which are identified as zero in every solution $\Gamma_i (1 <= i <= \rho)$. The concept is illustrated in Figure 5 for $N = 2$ and $\rho = 3$. Adopting this technique ensures not to remove any essential regulatory interaction among the genes, as well as helps to avoid sticking in local minima. As the search progresses, at the end of each iteration, the proposed method will identify which parameters are more important and which are zero-valued, escaping local convergence. Finally, when no more parameters could be identified as zero at the end of some iteration, we assume the remaining regulatory interactions are indispensable for the network, and this completes the optimization process.

4. EXPERIMENT

To confirm the effectiveness of the proposed algorithm we experimented with an artificial genetic network inference problem. As the target network we used a small-scale S-system model with the parameter set listed in Table 1 [7]. This network was first studied by Tominaga in [19] and later many others have experimented with it [7, 8, 18]. Hence this network, which represents a typical gene interaction system consisting of 5 genes, has become like a benchmark network for evaluating the performance of optimization algorithms for S-system models.

Table 2: Sets of initial gene expression levels used

Trial	X_1	X_2	X_3	X_4	X_5
1	0.70	0.12	0.14	0.16	0.18
2	0.10	0.70	0.14	0.16	0.18
3	0.10	0.12	0.70	0.16	0.18
4	0.10	0.12	0.14	0.16	0.70
5	0.10	0.12	0.14	0.70	0.70
6	0.10	0.12	0.14	0.70	0.18
7	0.70	0.70	0.14	0.16	0.18
8	0.10	0.70	0.70	0.16	0.18
9	0.10	0.12	0.70	0.70	0.18
10	0.70	0.12	0.14	0.16	0.70

As specified in Section 3.1, if an insufficient amount of time series data is used for estimating the parameters for S-system model many candidate solutions will evolve due to the high-degree of freedom of the model. Therefore Kikuchi et al had used 50 time series data for solving this 5 gene network [7]. We have also used the same sets of time series

Table 3: Parameters estimated using 5 sets of time series

i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	4.95	0.0	0.0	0.97	0.0	-0.99	10.06	1.99	0.05	0.0	0.0	0.0
2	10.01	1.97	0.0	0.0	0.0	0.0	9.89	0.0	1.94	0.0	0.0	0.0
3	9.65	0.0	-0.98	0.0	0.0	0.0	9.60	0.0	-0.99	2.26	0.0	0.0
4	7.79	0.05	0.05	1.86	0.0	-0.97	9.82	0.0	0.0	0.0	2.06	0.0
5	10.12	0.0	0.0	0.0	1.99	0.0	10.13	0.0	0.0	0.0	0.0	1.98

Table 4: Parameters estimated using 10 sets of time series

i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	4.99	0.0	0.0	1.00	0.0	-1.00	9.99	2.00	0.00	0.0	0.0	0.0
2	9.99	1.99	0.0	0.0	0.0	0.0	9.99	0.0	1.99	0.0	0.0	0.0
3	10.00	0.0	-0.99	0.0	0.0	0.0	10.00	0.0	-1.00	1.99	0.0	0.0
4	8.00	0.00	0.00	2.00	0.0	-0.99	10.00	0.0	0.0	0.0	1.99	0.0
5	10.00	0.0	0.0	0.0	1.99	0.0	10.00	0.0	0.0	0.0	0.0	1.99

data in our experiments. The initial values of these sets are listed in Table 2. The sets of time-series were obtained by solving the set of differential equations (1) on the model in Table 1. A typical set of time course, obtained from trial 1 of Table 2, is shown in Figure 2. To show the effectiveness of proposed method we perform two separate experiments using the time dynamics. In our first experiment we used 25 time series obtained from the first five trial sets of Table 2 for optimizing the network parameters, and the second experiment was performed using all of the 50 series. From each time series data 25 sampling points were used for optimization.

The conditions of our experiments were as follows. The search regions of the parameters were $[0.0, 15.0]$ for α_i and β_i, and $[-3.0, 3.0]$ for g_{ij} and h_{ij}. The parameter values for TDE algorithm were $F = 0.5$, $CF = 0.8$ and $M_t = 0.05$, population size was 600 and the maximum number of generation was 1600 for each of the three steps S1, S2 and S3. In each iteration we evolved 5 ($\rho = 5$) independent solution from which we identified the parameters whose values reached zero. Our algorithm was implemented in Java language and the time required for a single evolution in one loop was approximately 10 hours for first experiment and 18 hours for second experiment using a PC with Athlon 2200 MHz processor and 512MB RAM.

In order to reduce the computational burden a structure skeletalizing was applied in a similar fashion used by Tominaga et al in [19]. If the absolute value of a parameter is less than a threshold value δ then structure skeletalizing reset it to zero. This process reduces the computational cost as well as helps to identify the zero valued parameters. In our experiment $\delta = 0.05$ was used.

5. RESULT

Table 3 shows the parameters estimated by our algorithm in the first experiment (5 set of trial data used). As shown in the table our model was able to attain the over all network structure and parameter values were also very close to targeted values. The sum of the squared relative error, between the time-courses produced by the inferred model and the given time series data, i. e. the final value of fitness function (2), is 4.2×10^{-3}. Our algorithm iterated 5 times to settle on this parameter sets. At the end of first iteration 19 parameters were identified to have zero values. In the subsequent iterations the numbers of parameters inferred to have value zero were 30, 33, 34 and 34 successively. At the last iteration the algorithm could not identify any more zero valued parameters and hence the optimization ends leaving 3 zero valued parameters unidentified. These three parameters (h_{12}, g_{41} and g_{42}) were still very close to zero, which proves that the search was directed in the right way. We also experimented with other 5 sets of trail data (chosen from Table 2). The results (not shown here) were more or less similar to that shown in Table 3.

The parameters estimated for the target gene regulatory network, using 10 sets of time series data, is shown in Table 4. This time our algorithm was able to identify the correct network structure i.e. all the 37 zero valued parameters were correctly identified. To estimate this set of parameter values our algorithm took 6 iterations identifying 18, 28, 30, 35, 37 and 37 zero valued parameters in respective iterations. If we look at the other estimated parameter values then it would be found that they are pretty close to the actual parameter values. And the sum of the squared relative error, between the original and estimated parameter set produced time-dynamics, is 4.5×10^{-7}.

6. DISCUSSIONS

Recently many researchers have used evolutionary computation for model based inference of gene regulatory network, one of the most important problems in bioinformatics. In our work we investigate the suitability of Differential Evolution (DE) for genetic network estimation problem. From our preliminary study, using two different small scale networks, we found that DEs yielded better fitness values compared to a standard GA or ES. Between DE and TDE we found the performance of TDE better compared to DE in terms of both fitness value and convergence rate. These experiments ascertain the superiority of TDE over other traditional evolutionary algorithms in searching network topology and parameter values.

Attaining lower fitness value or reproducing the time-course is not the ultimate goal in genetic network inference problem, rather estimating the actual network structure and parameter values is more important. To achieve this goal we proposed an extension of the basic fitness function (2) for identifying the sparse structure of the network which is more common in biological systems. And to identify the cor-

rect parameter values we proposed a three step optimization model which works with in the general framework of gradual optimization strategy [7].

The function for evaluating fitness is extended in a similar way done by Kikuchi and Kimura i.e. augmented by the sum of absolute values of kinetic orders [7, 8], but here we have used the reciprocal of network dimension, as penalty constant, to multiply the summation. This will minimize the effect of the penalty term on structure estimation (using equation 5) with the increase of network dimension and thus will help to find a more sparse structure as expected for larger biological networks.

For estimating parameter values we have used repeated optimization for gradual identification of zero valued parameters, which was originally proposed in [7]. But here we have performed the optimization in 3 phases for identifying the parameter values more precisely. Experimental results showed using the method we were able to identify the parameter values not only with higher accuracy but also with less number of iterations. And a well known problem for optimizing the S-system parameters is convergence to local minima. To deal with it we acquire multiple solutions in each iteration and take parameters, nullified by all solutions, as zero. This approach may slowdown the overall optimization process as we take common zero valued parameters but at the same time this will guarantee not to set a parameter value as zero incorrectly and lose the essential regulation. This method is also effective for escaping local minima, a real obstacle to find the global optimal value for a highly deceptive problem.

It is well known that if insufficient amount of time course data is used for inferring S-system parameters then, because of the flexibility of the model, it will converge to many local minima. The number of time series data required for each gene to uniquely identify the correct regulatory structure depends on the number of network components, nature of their interaction and regulation, the optimization algorithm applied, even the properties of the time series used. Unfortunately no concrete research has been done so far to illuminate this issue. However it is expected that using higher number of time courses the network parameters could be estimated with higher precision. But with the increase of time courses, the time for finding a solution also increases exponentially due to the complexity involved in solving equation (1). In our experiments we also found higher accuracy in estimated parameter set using more time course data for each gene. None of the experiments using 5 sets of data was successful to find the complete target network topology. But all of them were still very close to actual network. On the other hand, using 10 sets of data we could identify the correct structure. Comparing with the results obtained by PEACE1 (proposed by Kikuchi [7]), we found our method was more successful in estimating the skeletal structure as well as the parameter values. Our method captured all the 37 zero valued parameters whereas PEACE1 could identify 36 using same number of time series. Moreover our estimated parameter values were also closer to actual parameter values on an average when compared to those estimated by PEACE1. Furthermore the number of iteration required in PEACE1 was 7 where our algorithm converged after 6 iterations. Even when the result of PEACE1 is compared with that obtained by our method using 5 sets of time series data, the performance of our algorithm seems to be pretty good. PEACE1 running on PC Cluster (Pentium III 933MHz×1040CPUs) reportedly took 10 hours for one loop, where our algorithm took 18 hours for a single solution in one loop using a Athlon 2200 MHz processor. A significant time performance improvement could be achieved using C/C++ implementation and some sophisticated method other than fourth order Runge-Kutta method for solving S-system equations, and our method could be easily parallelized on a PC cluster.

7. CONCLUSION

In this paper we proposed an improved method for estimating genetic networks using S-system formalism. We used gradual optimization strategy implemented by Differential Evolution (DE) for capturing the skeletal structure of biological networks. A new fitness function together with a more effective optimization technique was proposed. A method to deal with the problem of local minima was also suggested. The performance of the method was verified using a small scale artificial network and the experiments showed that the proposed method is capable to identify the correct topology and to estimate parameter values close to actual. Modeling with differential equation, such as S-system, requires a lot of data points for estimating the parameter values. Our proposed method was successful to estimate parameter values more accurately compared to others method using same number of time course data.

The proposed method works well for small networks, but large scale networks are still out of the scope of the method in the current form because of the high dimensionality of the model. Maki *et al.* has proposed a method for subdividing the $2N(N+1)$ dimensional network inference problem (based on S-system) into N subproblems each of which are $2(N+1)$ dimensional [10]. Incorporation of such decomposition technique will be tried in future to adapt the proposed method for larger networks. Application of the proposed method to the actual biological network will verify its effectiveness more comprehensively. Still many improvements in different aspects of the proposed method are needed for applying it to real microarray data obtained from large scale biological networks.

8. REFERENCES

[1] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of the Pacific Symposium on Biocomputing 4*, pages 17–28, January 1999.

[2] S. Ando, E. Sakamoto, and H. Iba. Evolutionary modeling and inference of gene network. *Information Sciences*, 145(3-4):237–259, September 2002.

[3] J. M. Bower and H. Bolouri. *Computational Modeling of Genetic and Biochemical Networks*. The MIT Press, 2001.

[4] H.-Y. Fan and J. Lampinen. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1):105–129, September 2003.

[5] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conf on Evolutionary Computation (ICEC '96)*, pages 312–317, May 1996.

[6] D. Irvine and M. Savageau. Efficient solution of nonlinear ordinary differential equations expressed in s-system canonical form. *SIAM Journal on Numerical Analysis*, 27(3):704–735, 1990.

[7] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic netowrks using genetic algorithm and s-sytem. *Bioinformatics*, 19(5):643–650, March 2003.

[8] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of s-system models of genetic networks using cooperative coevolutionary algorithm. *Bioinformatics Advance Access published online on October 28*, 2004.

[9] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi. Development of a system for the inference of large scale genetic networks. In *Proceedings of the Pacific Symposium on Biocomputing 6*, pages 446–458, January 2001.

[10] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, and Y. Eguchi. Inference of genetic network using the expression profile time course data of mouse p19 cells. In *Genome Informatics 13*, page 382383, December 2002.

[11] 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. In *Proceedings of Congress on Evolutionary Computations*, pages 615–622, December 2003.

[12] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. dAlchéBuc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(Supp 02):138–148, September 2003.

[13] E. Sakamoto and H. Iba. Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 720–726. IEEE Press, May 2001.

[14] M. A. Savageau. 20 years of s-systems. In *E. Voit, editor, Canonical Nonlinear Modeling. S-systems Approach to Understand Complexity*, pages 1–44. Van Nostrand Reinhold, 1991.

[15] C. Speith, F. Streichert, N. Speer, and A. Zell. Optimizing topology and parameters of gene regulatory network models from time-series experiments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 461–470. Springer, June 2004.

[16] R. Storn. System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, 3(1):22–34, April 1999.

[17] R. Storn and K. Price. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

[18] F. Streichert, H. Planatscher, C. Spieth, H. Ulmer, and A. Zell. Comparing genetic programming and evolution strategies on inferring gene regulatory networks. In *Genetic and Evolutionary Computation Conference (GECCO) Proceedings*, pages 471–480. Springer, June 2004.

[19] D. Tominaga, N. Koga, and M. Okamoto. Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 251–258. Van Nostrand Reinhold, July 2000.

[20] S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Genetic and Evolutionary Computation Conference (GECCO'99) Proceedings*, pages 657–664. Morgan Kaufmann, July 1999.