A Recurrent Fuzzy Neural Model of a Gene Regulatory Network for Knowledge Extraction Using Differential Evolution

[1]Debasish Datta, [2]Sheli Sinha Choudhuri, [3]Amit Konar, [4]Atulya Nagar, and [5]Swgatam Das
{[1]debasish_datta23@rediffmail.com, [2]shelism@redifmail.com, [3]konaramit@yahoo.co.in, [4]nagara@hope.ac.uk, [5]swagatamdas19@yahoo.co.in}

Abstract— **A gene regulatory network describes the influence of genes over others. This paper attempts to model gene regulatory network by a recurrent neural net with fuzzy membership distribution of weights. A cost function is designed to match the response of neurons in the network with the gene expression data, and a differential evolution algorithm is used to minimize the cost function. The minimization yields fuzzy membership distribution of weights, which on de-fuzzification provides the desired signed weights of the gene regulatory network. Computer simulation reveals that the proposed method outperforms existing techniques in detecting sign, and magnitude of weights of the regulatory network.**

Index Terms – **gene regulatory network, fuzzy recurrent neural network, time series gene expression data, differential evolution algorithm.**

I. INTRODUCTION

Living beings adapt themselves to an ever-changing environment because of the changes in their genetic information stored in DNA [23]. When a new generation evolves from an old one, the information stored in the parent's DNA is copied into child's DNA. Most importantly, DNA contains a few regulatory genes, which on being influenced by environmental factors (such as temperature, atmospheric pressure etc.) attempt to control the transfer of genetic information from one generation to the next by interacting with other genes. This interaction among genes can be visualized as a virtual network, called gene regulatory network.

This regulatory mechanism is important from the point of view of genetic engineering [31], [32], as it provides insight into the interaction between different genes. Currently, with the advancement of the DNA micro array technology, it has become possible to simulate gene regulatory network from gene expression time series data. Researchers are taking a keen interest in modeling gene regulatory network by soft computing techniques [33], [34], [35], and have attempted several approaches, including Boolean networks [7], [8], Linear differential model [9],[10], Bayesian networks [11],[12], linear additive regulation models [19], [20], [21], [22], and the like . A brief review of these models is given below.

Boolean networks consider two state of a gene: 1 for active and 0 for inactive. Boolean functions are used to describe the state change of a gene due to the interaction of other genes. The main drawback of Boolean network lies in ignorance of the effect of gene at intermediate levels, causing information loss because of binary quantization process**.** In addition, Boolean networks presume that the transition between genes' activation states are synchronous, which is biologically implausible. However, a study of dynamic behavior modeling using the Boolean network is under investigation [14], [15].

Bayesian models have also been employed to represent gene regulatory network. A gene regulatory network under the Bayesian framework is considered a directed acyclic graph, where the vertices are represented as genes, and the edges represent the conditional dependence relation between genes. Bayesian networks [11], [12] are effective in handling with noise, incompleteness, and stochastic nature of gene expression data. However they are unable to represent the dynamical aspects of gene regulation. In recent times, dynamic Bayesian networks [16], [17], [18] have been employed to handle the temporal information of gene regulatory network. They are also capable of handling 'hidden variable', 'prior knowledge', and 'missing data'.

Linear additive regulation models [19], [20], [21], [22] also are effective to describe the expression level of a gene at time t as a weighted sum of expression levels of other genes at time (t -1). The main drawback of the linear additive network model lies in its incapability to capture nonlinear dynamics between gene regulations.

None of the above technique could provide a fool-proof solution to the problem of gene regulatory network [36]. In this paper, we propose a mathematical model for gene regulatory mechanism using fuzzy recurrent neural network (RNN) [39], and attempt to determine the numerical interaction values between genes. We have chosen recurrent neural network [37] as the basic framework of our model because of its capability to handle dynamic aspects. In a recurrent neural network, the current state of a neuron is determined by the previous states of all or most of the neurons in the network. As a result, the model provides dynamic aspect, which is most essential for the gene regulatory network. The weight between neurons

Manuscript received November 17, 2008; revised February 24, 2009.

Debasish Datta, Sheli Snha Choudhuri, Amit Konar and Swagatam Das are with the Department of Electronics And Telecommunication Engineering, Jadavpur University, Kolkata-700032, India.

Atulya Nagar is with Intelligence, and Distributed System Lab, Liverpool Hope University, Liverpool L169JD

978-1-4244-2959-2/09/$25.00 © 2009 IEEE

gives the numeric interaction values between genes. Thereby, if it is possible to find those weight values between neurons from the time series data available for genes, then the real interaction between genes can be revealed. Moreover, different techniques are available to find these interactions between neurons [28], [29], [30] for example back propagation through time [38], [40]. In this paper we, however, have used differential evolution algorithm to determine weights.

Gene expression profiling or micro array [6] analysis has enabled us to measure thousands of genes in a single RNA sample. There are a variety of micro array platforms that have been developed to accomplish this gene expression profiling, and the basic idea for each is simple: a glass slide or membrane is spotted or "arrayed" with DNA fragments or oligonucleotides that represent a specific gene coding region. Purified RNA is then fluorescently- or radioactively labeled, and hybridized to the slide/membrane. In some cases, hybridization is done simultaneously with reference RNA to facilitate the comparison of data across multiple experiments. After thorough washing, the raw data is obtained by laser scanning or auto radiographic imaging. At this point, the data may be entered into a database, and analyzed by a number of statistical methods.

The main problem associated with the approaches used to find those interaction values between genes is that the only information available in reality is the gene expression time series data, which are also limited, and erroneous. Since *a priory* knowledge about the network topology or its parameters are not available, it is difficult to model the problem in a deterministic framework. Further, gene-regulatory network weights usually are multimodal functions of the gene expression time-series data. While modeling gene regulatory network as an optimization problem, researchers attempt to minimize the mean square error obtained by taking the squared difference of the measured time-series data, and desired response of the network nodes. Since the error function too is a nonlinear multimodal surface, the solution sets of weights is non-unique, and naturally the solution does not guarantee the optimal selection of weights of the network. An alternative formulation of the problem is to presume that the weights have a stochastic/fuzzy distribution. Although there is a scope of multiplicity of such distributions for individual weights, the justification of introducing stochastic/fuzzy distribution lies in searching a narrow space for individual probability or fuzzy memberships in [0, 1], thereby minimizing the possibility of multiplicity, and also cost of determining the right value of a weight w_{ji} in [-30, 30] say, can be reduced significantly by interval based fuzzy system, where we need to determine the memberships in [0, 1] for fixed weights in a set S = {-30,-25,0,25,30}.

In this paper, we represent the connection weights of the regulatory network using fuzzy membership distributions, and employed a differential evolution algorithm to find the optimal membership distribution of weights. The membership distributions thus obtained are then defuzzified by centroidal defuzzification technique, and the results are found to be promising. The sign of almost all the weights obtained by the network can be retrieved correctly for known networks, which indeed is a challenge to the works undertaken in the existing literature [1],[24], [25],[26],[27] where they missed to retrieve the correct sign of many network weights.

The paper is organized as follows. In section II, we propose a fuzzy extension to the existing framework of gene regulatory network model using recurrent neural network. Section III presents a scheme for generation of gene expression time series data for known gene-regulatory network. We have generated the gene expressions of a 4-gene network, and shown them graphically to mimic the real gene expression time series data, with some predefined values. Section IV describes cost function used in our model. In section V, we describe the differential evolutionary algorithm used to find the simulated network parameters. In section VI, we presented the simulated results, and in section VII we used our model to simulate a gene regulatory network using real gene expression time series data.

II. PROPOSED EXTENDED FUZZY RECURRENT NEURAL NETWORK (FRNN) MODEL

Currently available time series data from gene microarray gives the idea that the value of gene expression changes with time. To incorporate the time aspect in the proposed model, and the dynamic aspect of recurrent neural network (as introduced earlier), we have chosen the differential equation (1) as our model, where each gene expression is differentiated with respect to time.

$$T_i \frac{dg_i}{dt} = f\Big(\sum_{j=1}^{N} \overset{*}{w}_{ji} g_i - b_i\Big) - k_i g_i \qquad (1)$$

Here the weight w_{ji} from neuron j to i has a fuzzy membership distribution μ ($*w_{ji}$), and the corresponding fuzzy set is given by a doublet $\{w^k_{ji} \mid \mu(w^k_{ji})\}$. The $*w_{ji}$ of equation (1) is evaluated by centroidal defuzzification procedure given by (2).

$$\overset{*}{w}_{ji}(t) = \frac{\sum\limits_{k=1}^{F} w^k_{ji} \times \mu(w^k_{ji})(t)}{\sum\limits_{k=1}^{F} \mu(w^k_{ji})(t)} \qquad (2)$$

As an example let F=3; so that a particular weight can be represented as {30/0.35, 0.001/0.89,-30/0.2}, and after the de-fuzzification it becomes (30×0.35 + 0.001×0.89-30×0.2)/ (0.35+0.89+0.2) = 3.1256. At this point, we want the attention of the reader on the above fuzzy set; the set members are 30, 0.001, and -30. We have chosen the middle value as 0.001 instead of 0. Because if it is zero the value will not have any effect in de-fuzzification. To make it up we took the value of a weight as zero at the final step, when its calculated value after de-fuzzification is less or equal to 0.001. The bias b_i, and time constant T_i are also represented in a similar manner like w_{ij}^* (see Fig.2.)

Let g_i is the expression of i^th gene, T_i is the time constant, w_{ji}^* be a defuzzified weight from neuron j to neuron i in the neural net representation of a recurrent neural network ; w_{ji}^* can be positive, negative or zero depending on whether j^th gene is activating, inhibiting gene i or doesn't have any effect on it at all, 'b_i', 'k_i', and 'N' represent the bias term for i^th gene, decay constant for g_i, and total number of genes present in the network, $f(x)=1/(1+e^{-x})$ is the nonlinear function used to get the output of each gene.

The available gene time series data we get in reality is in discrete form, and the model shown in equation (1) is a continuous one. To incorporate the discrete feature in our model with respect to time we have changed it as follows:

$$T_i \frac{dg_i}{dt} = f\left(\sum_{j=1}^{N} \overset{*}{w}_{ij}\, g_i(t) - b_i\right) - k_i g_i(t)$$

$$\Rightarrow T_i \frac{g_i(t+\Delta t) - g_i(t)}{\Delta t} = f\left(\sum_{j=1}^{N} \overset{*}{w}_{ji}\, g_i(t) - b_i\right) - k_i g_i(t)$$

$$\Rightarrow g_i(t+\Delta t) - g_i(t) = \frac{\Delta t}{T_i} f\left(\sum_{j=1}^{N} \overset{*}{w}_{ji} g_i(t) - b_i\right) - \frac{\Delta t}{T_i} k_i g_i(t)$$

$$\Rightarrow g_i(t+\Delta t) = \frac{\Delta t}{T_i} f\left(\sum_{j=1}^{N} \overset{*}{w}_{ji} g_i(t) - b_i\right) + g_i(t)\left(1 - \frac{\Delta t}{T_i} k_i\right) \qquad \text{……….(3)}$$

Equation (3) demonstrates how expression of a particular gene changes with time in response to the other genes present in the network.

III. GENE EXPRESSION PRODUCTION WITH PREDEFINED VALUES

In section III we attempted to generate artificial gene expression time series data to test the accuracy of our method. Using the model in equation (3), we have generated time series data using the parameter values of TABLE I of a 4-gene network;

TABLE I
PARAMETER VALUES OF A 4-GENE NETWORK

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	20.0	-20.0	0.0	0.0	0.0	10.0
Gene2	15.0	-10.0	0.0	0.0	-5.0	5.0
Gene3	0.0	-8.0	12.0	0.0	0.0	5.0
Gene4	0.0	0.0	8.0	-12.0	0.0	5.0

The interpretation of the above weight values are as follows. From TABLE I we can see that box (1, 2) = -20.00, the meaning is that gene1 has -20.00 unit of effect on gene2. We have chosen the same set of values as in paper [1] to compare the accuracy of our model. The generated time series data for the genes are shown in Fig.1.

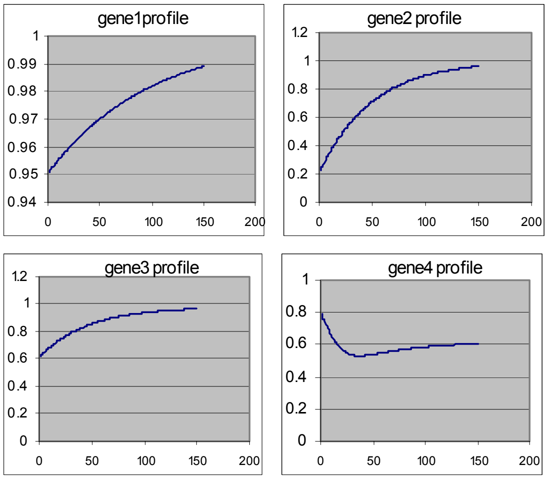

Fig.1. Expression profile of gene1, gene2, gene3, and gene4, respectively

X-axis represents the time points of observations of genes expression and Y-axis represents gene expressions. As can be seen from Fig.1. that nearly after150 points, the expression of all the genes get saturated, therefore this is the region from where we can extract maximum information. Because of this reason we have used 150 data points for each gene profile.

IV. PROPOSED COST FUNCTION

Accuracy of gene regulatory network (GRN) design mainly depends on two issues (i) how well we can measure the accuracy of the existing connection values of the network, and (ii) how well we can measure the accuracy of the skeletal structure (network topology) of the simulated network. The first issue is related to fine tuning of the values of the existing or detected connections, and the second issue is related to the detection of the sparse structure of the network. Handling both issues simultaneously is a tough job, because we do not have any knowledge except the available gene expression time series data, which is also limited. Therefore a judicious choice of cost function can greatly influence the accuracy of the simulated network.

In our proposed cost function, we tried to incorporate both the issues. To meet the first issue, we evaluated the accuracy of the produced gene expression of our simulated network by comparing it with the gene expression produced using the network parameters of TABLE I with the hope that if the network parameters of our simulated network is closer to the parameters of TABLE I then the difference (error) between these two set of gene expression will be less. That error has been calculated using the equation (4).

$$C_1 = \frac{1}{\Gamma NM} \sum_{k=1}^{M} \sum_{t=1}^{\Gamma} \sum_{i=1}^{N} \{[g^i_{org}(t)]_k - [g^i_{cal}(t)]_k\}^2 \qquad (4)$$

This expression is basically the squared error between the original gene expression, and experimental gene expression. Here M is the number of time series used; r is the number of data point in each time series data, and N is the number of gene present in the network. $[g^i_{org}(t)]_k$ is the original expression of i^{th} gene at t^{th} time instance in k^{th} time series, and $[g^i_{cal}(t)]_k$ is the calculated expression of the same using our simulated network.

Designing a cost function to detect the accurate sparse structure of the simulated network (which is the second issue) is much more difficult. Fortunately, the study of genetics revels that in a gene regulatory network it is unlikely that all the genes interact with each other; rather few genes are involve in regulation of a gene. Considering this practical phenomenon, we designed the cost function given by equation (5).

$$C_2 = p \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{|\overset{*}{w}_{ij}|}{|1 + \overset{*}{w}_{ij}|} \qquad (5)$$

Here w^*_{ji} is the connection value between gene j, and gene i, and p is a constant. The idea behind C_2 is outlined here. The natural phenomenon is that a particular gene is influenced by few genes; therefore the simulated network structures are penalized according to their number of nonzero connections. Introduction of this term in cost function is risky, because it may lead the whole system to an all zero solution. To tackle this situation, and gain some sort of control over C_2, we kept p as constant. Choosing proper value of p is also tricky. An appropriate value will lead to a good solution or it may mislead the system, its value should be such that C_2 can't override C_1.

Our final cost function is shown in equation (6).

$$C = C_1 + C_2$$

$$C = \frac{1}{\Gamma NM} \sum_{k=1}^{M} \sum_{t=1}^{\Gamma} \sum_{i=1}^{N} \{[g^i_{org}(t)]_k - [g^i_{cal}(t)]_k\}^2 + p \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{|\overset{*}{w}_{ij}|}{|1 + \overset{*}{w}_{ij}|} \qquad (6)$$

Using this cost function, we will select the solution with the smallest cost value as the final solution i.e. if cost of solution_1 is less than that of solution_2 then solution_1 is our final solution.

V. EXTRACTION OF FUZZY MEMBERSHIP DISTRIBUTION OF WEIGHTS USING EXTENDED DIFFERENTIAL EVOLUTION ALGORITHM

In this paper, we have used the well known differential evolutionary (DE) algorithm to find the simulated network. To improve the outcome of DE we have included a new initialization technique. The idea behind it is to spread the initial candidate solutions as far possible in the search space. As a result, we hope that some of the solutions may be close to the original solution. To incorporate this feature into our evolutionary algorithm, we have used a chaos system [2], [3]. The process of producing the chaos is as follows:

$$Z_{k+1} = \mu Z_k(1 - Z_k) \qquad (7)$$

where $k = 0, 1, 2, 3, \ldots, \Theta$, Θ is the number of chaotic iteration, μ is the control parameter. Z_k takes any value between 0 and 1; it is the selected value in the kth iteration. When we set $\mu=4$, and $Z_0 \neq \{0, 0.25, 0.5, 0.75, 1\}$, the value of Z_k distributes with proper randomness, and irregularity. We indeed found that this initialization improve the overall convergence rate of the differential evolution algorithm.

We now briefly outline the differential evolution algorithm [4] introduced here. It is a population based evolution method. A population of solution vectors is successively updated by addition, subtraction, and component swapping, until the population converges, hopefully to the optimum. The steps of the differential evolution algorithm have shown below:

(i) Initialize the population pool with N number of random individuals

(ii) For each candidate solution C_i in the population pool form a mutant vector $V_i = C_{r1} + \lambda\, (C_{r2} - C_{r3})$ where r1, r2, r3 are three mutually distinct randomly drawn indices from 1 to N, and λ is the mutation factor, $0 < \lambda <= 2$.

(iii) For each candidate solution C_i, and corresponding mutant vector V_i form a trial vector α_i as follows: for each component of the candidate solution draw a random number τ_i between 0 and 1. If $\tau_i <= P_r$, then the corresponding component of V_i appears in α_i, otherwise component of candidate solution appears in α_i. P_r is a predefined value, and typically $P_r = 0.9$.

(iv) If COST (α_i) is better than COST (C_i) then α_i replaces C_i in the next generation otherwise keep C_i. COST () is the defined cost function for a particular problem.

(v) If stop criterion is met then show the result otherwise go to step (ii)

Each individual candidate of differential evolution algorithm represents a complete solution. As an example one solution of the above 4-gene network contains 16 connection weights, 4 bias terms, and 4 time constants, with total of 24 fields. Each field of every individual represented by a fuzzy set, has number of element = C_f. Again for an example if $C_f = 5$ then one single individual solution will contain 24 x 5=120 number of fields for that 4-gene network. We maintain a pop_size number of individual all the time in the population pool. The population pool of the differential evolutionary algorithm for the four gene network with $C_f = 5$ can be represented pictorially as in Fig.2.

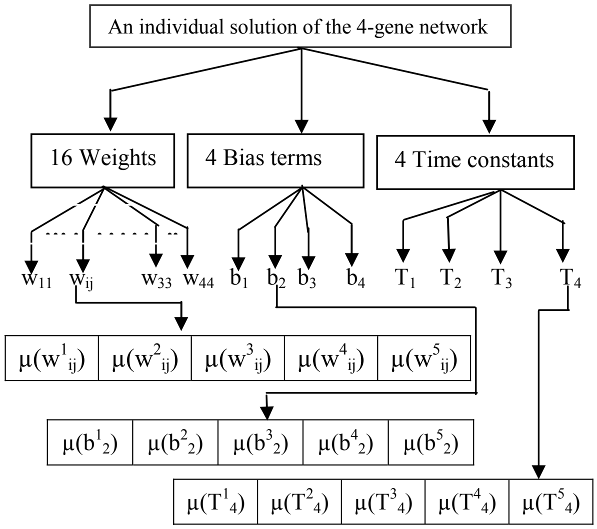
An individual solution of the 4-gene network
16 Weights
4 Bias terms
4 Time constants
w11 wij w33 w44
b1 b2 b3 b4
T1 T2 T3 T4
μ(w1ij) μ(w2ij) μ(w3ij) μ(w4ij) μ(w5ij)
μ(b12) μ(b22) μ(b32) μ(b42) μ(b52)
μ(T14) μ(T24) μ(T34) μ(T44) μ(T54)

Fig.2. Pictorial representation of a single individual solution of differential evolution algorithm.

In Fig.2. $C_f = 5$, $\mu(w^1_{ij})$, $\mu(w^2_{ij})$, $\mu(w^3_{ij})$, $\mu(w^4_{ij})$, $\mu(w^5_{ij})$ represents the fuzzy membership values of the weight w_{ij} of any individual solution. $\mu(b^1_2)$, $\mu(b^2_2)$, $\mu(b^3_2)$, $\mu(b^4_2)$, $\mu(b^5_2)$ represents the fuzzy membership value of the bias terms b_2, and $\mu(T^1_4)$, $\mu(T^2_4)$, $\mu(T^3_4)$, $\mu(T^4_4)$, $\mu(T^5_4)$ represents the fuzzy membership of the time constants T_4. At each step of differential evaluation, we defuzzify each individual using the equation (2), generate gene expression using equation (3), calculate its cost value using equation (6), and make the appropriate decision whether to keep that particular individual for the next generation or not. In some cases after de-fuzzification the time constant may becomes zero. As can be seen from equation (3) that time constants are used as denominator thereby it can't be zero. To handle this situation we randomly re-initialize the fuzzy membership values of the time constants when it becomes zero.

V. RESULTS

The results are shown in TABLES II, and VI. For convenience, we have shown TABLE I once more; the results are generated using four time series data. We have chosen 1500 iterations of the differential evolution algorithm, 300 iterations for the chaotic initialization algorithm. Our algorithm takes nearly 27 minuets on a Pentium dual core computer containing 1 GB RAM.

TABLE I

	Gene1	Gene2	Gene3	Gene4	bi	Ti
Gene1	20.00	-20.00	0.00	0.00	0.00	10.00
Gene2	15.00	-10.00	0.00	0.00	-5.00	5.00
Gene3	0.00	-8.00	12.00	0.00	0.00	5.00
Gene4	0.00	0.00	8.00	-12.00	0.00	5.00

TABLE II
RESULTS AFER RUN1 USING 4 TIME SERIES DATA, 70 POPULATION, C_f =9, TOPOLOGY CONSTANT P= 0.1

	Gene1	Gene2	Gene3	Gene4	bi	Ti
Gene1	16.8	-0.6	0.0	0.0	11.1	14.6
Gene2	9.1	-14.8	24.5	3.8	-18.9	1.8
Gene3	7.8	-4.8	16.2	0.0	-0.8	5.7
Gene4	0.0	-0.1	7.3	-6.5	-22.2	27.7

TABLE III
RESULTS AFTER RUN2 USING 4-TIME SERIES DATA, 50 POPULATION, C_f= 5, TOPOLOGY CONSTANT P= 0.1

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	6.1	-13.6	0.0	0.0	-18.9	9.2
Gene2	18.3	-06.7	10.3	-0.01	-7.3	1.8
Gene3	1.4	-00.5	1.1	-0.5	-4.7	6.6
Gene4	8.4	14.2	-19.6	8.7	5.01	6.6

As can be seen from TABLEs I through III, connection weights are identified with sufficient accuracy, all the negative and positive signs, and some nonexistent weights are identified correctly in TABLE II. However, the sign of one weight is not correct in TABLE III. At this point, we want to remind the reader once again that we don't had any prior knowledge about the network structure. The results can be shown as below for better understanding.

TABLE IV
SIGNS OF TABLE I

	Gene1	Gene2	Gene3	Gene4	bi	Ti
Gene1	+	-	0.00	0.00	0.00	10.0
Gene2	+	-	0.00	0.00	-	5.0
Gene3	0.00	-	+	0.00	0.00	5.0
Gene4	0.00	0.00	+	-	0.00	5.0

TABLE V
SIGNS OF TABLE II

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0.00	0.00	11.1	14.6
Gene2	+	-	24.5	3.8	-	1.8
Gene3	7.86	-	+	0.00	-0.84	5.7
Gene4	0.00	-0.1	+	-	-22.2	27.7

TABLE VI
SIGNS OF TABLE III

	Gene1	Gene2	Gene3	Gene4	b_i	T_i
Gene1	+	-	0.0	0.00	-18.9	9.2
Gene2	+	-	10.3	-0.01	-	1.8
Gene3	1.4	-	+	-0.5	-4.7	6.6
Gene4	8.4	14.2	-19.6	+ (wrong)	5.0	6.6

For comparison of our results of TABLE V, we present the results calculated in paper [1] in TABLE VII. It has used the same set of network parameters introduced here.

TABLE VII
PARAMETER OBTAINED IN REFERENCE [1]

	Gene1	Gene2	Gene3	Gene4
Gene1	+(29.0)	+(34.0)[wrong]	0.0(5.5)	0.0(7.0)
Gene2	+(83.5)	-(98.0)	0.0(3.5)	0.0(3.0)
Gene3	0.0(16.5)	-(60.5)	+(99.5)	0.0(15.0)
Gene4	0.0(10.0)	0.0(8.5)	+(100.0)	- (100.0)

The result shown in TABLE VII was obtained using 5-time series data. In this table, the values in the parentheses indicate the percentage of occurrence of connections in the network within 200 runs. Clearly only two signs of the weights are determined correctly, and above all those values are among 200 runs. Therefore, no specific knowledge can be gained from that. On the contrary, the values of weights obtained in our method are due to execution of a single run, and from that one can easily have an idea about the nature of interaction among the genes. The results are pictorially shown in Fig.3. and Fig.4.

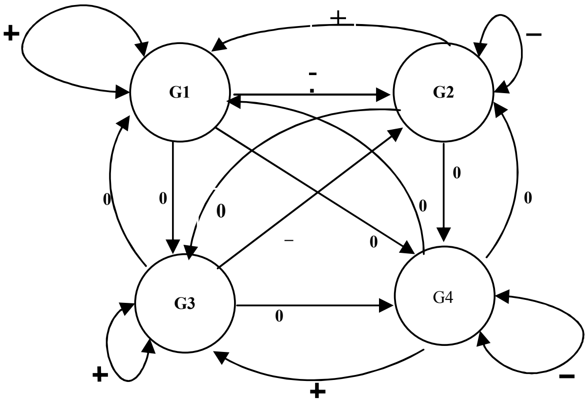

Fig.3. Original gene regulatory network of TABLE IV

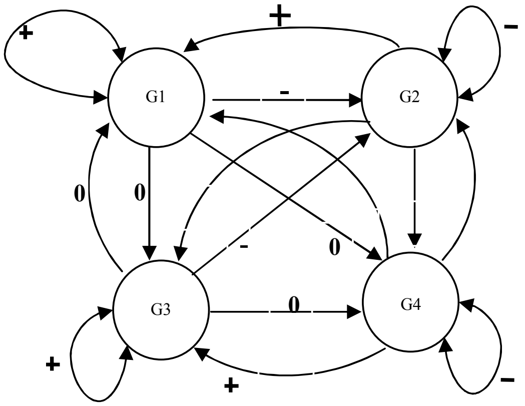

Fig.4. Simulated gene regulatory network of TABLE V

Although all the positive and negative signs of the weight values are identified correctly, our model fails to identify some non-existing connections. The unmarked connection weights of Fig.4. are not correct as they were non existent connections.

VI. INFERRING GENE REGULATORY NETWORK USING REAL DATA SET

We have used our model to infer the gene regulatory network of e.coli. bacteria S.O.S DNA repair network [5]. This network consists of nearly 30 genes regulated at the transcription level. Four experiments have been conducted with different UV light intensities. Experiment 1, and 2 using UV= 5 jm-2, and experiment 3, and 4 using UV=20jm-2. Using these experiments, expressions of eight major genes have been documented. These genes are uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, polB.

The function of the e.coli. S.O.S. DNA repair network is as follows: lexA acts as a balancing factor for the whole network. In absence of any DNA damage it binds to the promoter region of the genes, suppressing the S.O.S genes in the network. When DNA damage occurs, RecA (one of the S.O.S protein) becomes activated. It decreases the level of lexA. As a result the S.O.S genes become activated. Once the damage has been repaired or bypassed, the levels of RecA decrease. As a result, the level of lexA increases, and again deactivates the S.O.S genes. This regulatory mechanism is shown pictorially in Fig.5.

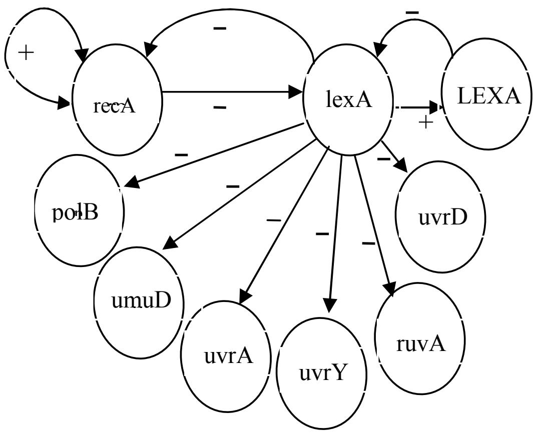

Fig.5. E.coli S.O.S. DNA repair network, activation is represented by '+' sign, and inhibitation by'-'; genes are written with small letter, and protein with capital letter,

Gene profiles of E.coli S.O.S DNA repair network consists of 50 data points, sampled every 6 minutes. The data set consists of four 8 × 50 matrix. Each column represents the observation of expression value at a particular time instant of eight genes, and each row represents the fifty expression value of a particular gene at different time instants. This data set is available in the website (http://www.weizmann.ac.il/mcb/UriAlon/). It is one of the best data sets, which fits our model. We have conducted the same experiment, as with the above artificial data, and the identified interaction values between genes are represented in TABLE VIII.

TABLE VIII
IDENTIFIED INTERECTION VALUES OF E.COLI. S.O.S DNA REPAIR NETWORK

	uvrD	lexA	umuD	recA	uvrA	uvrY	ruvA	polB
uvrD	-14.8	6.7	-10.4	17.0	0.0	-5.4	-15.5	16.1
lexA	-14.1	-7.9	0.0	-6.8	-12.3	13.9	1.1	-14.6
umuD	-18.5	-10.2	-24.8	9.2	-17.3	-21.2	-11.2	-20.2
recA	-31.8	-3.8	-11.6	-8.0	-12.8	-7.7	2.6	-16.1
uvrA	4.9	6.7	13.1	17.8	13.0	-20.6	-6.6	12.4
uvrY	-0.2	-7.5	-14.1	5.2	-12.5	0.0	-19.9	3.1
ruvA	-10.0	12.7	0.0	0.0	-7.6	-29.2	19.1	3.9
polB	-2.3	1.5	7.9	-3.6	-6.1	1.7	-18.8	0.0

We have fixed the lower and upper range of the interaction value as -30, and +30; the values of other parameters of the algorithm are the same as used in our model. The result of our algorithm has been shown in TABLE VIII. Here the reader should keep in mind that the cost function used contains multiple minima, and also the available gene expression time series data contains only 50 data points.

VIII. CONCLUSION

The only information available for inferring gene regulatory network is the gene expression time series data, which are also erroneous. There is no guideline regarding the structure of a particular gene regulatory network. Hence, the result obtained is not 100% accurate, but considering the challenges, and limited availability of information, our model provides a good result. If more knowledge can be incorporated in the cost function, then the model will be able to provide more accurate results. Our next goal is to incorporate a few more constraints in our cost function to make it more accurate.

REFERENCE

[1] Rui Xu, Donald C. Wunsch II and Ronald L. Frank, " Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization" IEEE/ACM transaction on computational biology and bioinformatics,vol.4, No.4,pp 681-692, 2007.

[2] C. Lng, S.Q. Li. Chaotic spreading sequences with multiple access performance better than random sequences. IEEE transaction on Circuit and System - I, Fundamental Theory and Application, 47(3):394-397, 2000.

[3] R.May. Simple mathematical models with very complicated dynamics. Nature, 261: 459 - 467, 1976.

[4] Storn, Rainer, and Kenneth Price. Differential Evolution –A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization11, 1997, pp. 341-359.

[5] B. Perrin, L. Ralaivola, A. Mazurie, S. Battani, J. Mallet, and F.d'Alche´-Buc, "Gene Networks Inference Using Dynamic Bayesian Networks," Bioinformatics, vol. 19, pp. ii138-ii148, supplement 2, 2003.

[6] L. F. A. Wessels, E. P. Van Someren, M.J.T. Reinders, "A comparison of genetic network models," Pac Symp Biocomput, 508-519, 2001.

[7] T. Akutsu, S. Miyano, S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," Pac Symp Biocomput, 17-28, 1999.

[8] S. Liang, S. Fuhrman, R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," Pac Symp Biocomput, 18-29, 1998.

[9] J. L. Michael De Hoon, S. Imota, K. Kobayashi, N. Ogasawara, S.Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtills using differential equations," Pac Symp Biocomput, 17-28, 2003.

[10] T. Chen, H. L. He, G. M. Church, "Modeling gene expression with differential equations," Pac Symp

Biocomput, 4:29-40, 1999.

[11] N. Friedman, M. Linial, I. Nachman, D. Pe'er, "Using Bayesian net work to analyze expression data," J. Comp. Biol., 7, 601-620, 2000.

[12] S. Imoto, T. Gota, S. Miyano, "Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression," Pac Symp Biocomput, 175-186, 2002.

[13] R. Xu, X. Hu, D. Wunsch, "Inference of genetic regulatory networks with recurrent neural network models," Engineering in Medicine and Biology Society, 2004. EMBC 2004. Conference Proceedings. 26th Annual International Conference of the Volume 2, 1-5 Sept. 2004 Page(s):2905 –2908 Vol.4.

[14] J. Hallinan and J. Wiles, "Evolving Genetic Regulatory Networks Using an Artificial Genome," Proc. Second Asia-Pacific Bioinformatics Conf., vol. 29, pp. 291-296, 2004.

[15] J. Hallinan and J. Wiles, "Asynchronous Dynamics of an Artificial Genetic Regulatory Network," Proc. Ninth Int'l Conf. Simulation and Synthesis of Living Systems, 2004.

[16] D. Husmeier, "Sensitivity and Specificity of Inferring Genetic Regulatory Interactions from Micro array Experiments with Dynamic Bayesian Networks," Bioinformatics, vol. 19, no. 17, pp. 2271-2282, 2003.

[17] B. Perrin, L. Ralaivola, A. Mazurie, S. Battani, J. Mallet, and F. d'Alche´-Buc, "Gene Networks Inference Using Dynamic Bayesian Networks," Bioinformatics, vol. 19, pp. ii138-ii148, supplement 2, 2003.

[18] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, and S. Miyano, "Estimating Gene Networks from Gene Expression Data by Combining Bayesian Network Model with Promoter Element Detection," Bioinformatics, vol. 19, pp. ii227-ii236, supplement 2, 2003.

[19] P. D'haeseleer, "Reconstructing Gene Network from Large Scale Gene Expression Data," PhD dissertation, Univ. of New Mexico, 2000.

[20] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modelling of mRNA Expression Levels during CNS Development and Injury," Proc. Pacific Symp. Bio computing, pp. 41-52, 1999.

[21] E. van Someren, L.Wessels, and M. Reinders, "Linear Modeling of Genetic Networks from Experimental Data," Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology, pp. 355-366, 2000.

[22] E. van Someren, L. Wessels, and M. Reinders, "Genetic Network Models: A Comparative Study," Proc. SPIE, Micro-Arrays: Optical Technologies and Informatics, pp. 236-247, 2001.

[23] The Modern Synthesis of Genetics and Evolution Copyright © 1993-1997 by Laurence Moran. [http://www.talkorigins.org/faqs/modern-synthesis.html]

[24] Carlos Epinosa-soto, Pablo Padilla-Longoria and Elena R.Alvarez-Buylla, "A Gene Regulatory Network Model for Cell-Fate Determination during Arabidopsis thaliana Flower Development That Is Robust and Recovers Experimental Gene Expression Profiles." The Plant Cell, Vol. 16, PP: 2923–2939, November 2004, www.plantcell.org ^{a} 2004 American Society of Plant Biologists,

[25] Thomas Schlitt and Alvis Brazma, "Current approaches to gene regulatory network modeling", BMC Bioinformatics 2007, 8(Suppl 6):S9 doi:10.1186/1471-2105-8-S6-S9, This article is available from: http://www.biomedcentral.com/1471-2105/8/S6/S9

[26] Rui Xu, Ganesh K.Venayagamoorthy, Donald C. Wunsch II, "Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization", Science Direct, Neural networks 20 (2007) 917- 927, online available at www.sciencedirect.com

[27] Vohradsky J, "Neural Network Model of Gene Expression", the FASEB journal, 15(2001) 846-354.

[28] Shuhui Li, Donald C. Wunsch, Edgar O'Hair, Michael G. Giesselman, "Extended Kalman filter training of neural network on SIMD parallel machine", journal of Parallel and Distributed Computing , Volume 62, issue 4 (April 2002), pages: 544 – 562, ISSN: 0743-7315.

[29] Pinaki Roychowdhuri, Y.P. Singh, and R.A Chanskar, "Dynamic Tunneling Technique for efficient Training of Multilayer Perceptrons", IEEE transaction on Neural Networks, Vol.10, NO. 1, January 1999.

[30] S.F. Masri, A.W. Smyth, A.G Chassiakos, M. Nakamura, T.K. Caughey , "Training Neural Networks By Adaptive Random Search Technique ", Journal of Engineering Mechanics, vol. 125, No. 2, pp-123-132, February 1999.

[31] Bar-Joseph, Z. (2004). "Analyzing time series gene expression data", Bioinformatics, 20 (16), 2493-2503.

[32] De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. Journal of Computational Biology, 9, 67-103.

[33] Wahde, M., & Hartz, J. (2000). Coarse-grained reverse engineering of genetic regulatory networks. Biosystems, 55, 129- 136.

[34] Wahde, M., & Hartz, J. (2001). Modeling genetic regulatory dynamics in neural development. Journal of computational Biology, 8,429-442.

[35] VanBogelen, R.A., Greis, K.D., Blumenthal, R.M., Tani, T.H., and Matthews, R.G. (1999), "Mapping regulatory networks in microbial cells", Trends Microbial. 7, 320 – 328.

[36] Somogyi, R., and Sniegoski, C.A. (1996) "Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation", complexity 1, 45-63.

[37] Bose, N.K., and Liang, P. (1996), "Neural Network Fundamentals with Graphs. Algorithms, and Applications", p.312, McGraw-Hill, New York.

[38] Werbos, P. (1990). "Backpropagation through time: what it does and how to do it", Proceedings of IEEE, 78(10), 1550-1560.

[39] Hassoun, M.H. (1995), "Fundamentals of Artificial neural network", MIT Press, Cambridge, Mass.

[40] D.E Rumelhart, G.E Hinton, and R.J. Williams, "Learning representation by backpropagation errors", Nature, Vol.323, pp. 533 – 536, 1986.