An Improved Identification Technique of Gene Regulatory Network from Gene Expression Time Series Data Using Multi-Objective Differential Evolution

[1]Debasish Datta, [2]Amit Konar, [3]Atulya Nagar, [4]Archana Bisoyi

[1,2]Department of Electronics and telecommunication, Jadavpur University, Kolkata-32, [3]Intelligence, and Distributed System Lab, Liverpool Hope University, Liverpool L169JD,[4]Department of Electronics and Telecommunication Engineering, Institute of Advanced Engineering and Technology, Rayagada, Orissa.

debasishresearchpaper@rediffmail.com, konaramit@yahoo.co.in, nagara@hope.ac.uk,aranach.bisoyi88@gmail.com

Abstract – Gene regulatory network provides the knowledge of interaction strength among the genes in living organisms. Accurate identification of gene regulatory network is of prime interest to the researchers in recent time. Different researchers applied different optimization techniques to solve this problem. Most of these optimization techniques considers the square error between reference and simulated gene expression as their objective and minimize it to get a solution for the identification problem under consideration. But these techniques do not guarantee a unique set of network parameter, because the squared error is a non-linear multi-modal surface of network parameters. Therefore considering only square error as the objective function is not a good choice. An alternative way of formulation of this problem is to validate it from different perspective. In this paper, we propose a technique for identification of gene regulatory network using multiple objectives. The objectives are designed to make the identification technique more robust. Multi-objective differential evolution is used to find a set of pareto-optimal solutions with respect to the objective functions. Among those solutions, one is chosen according to some suitable criterion. Computer simulation has shown that the proposed technique can identify useful interaction information from gene expression time series data.

I. INTRODUCTION

Identification of gene regulatory network (GRN) is of prime interest to the researchers as it provides an insight to activation or inhibition of a gene in living organisms. Researchers attempted several soft computing techniques, including Boolean network, linear differential model, Bayesian network, linear additive model and the like. A brief discussion on these techniques is given in section II. The main problem of these existing techniques lies in inferring the gene regulatory network by minimizing the squared error between the desired response and the computed response of the simulated network. Since this square error is nonlinear multimodal surface, the solution set is non-unique, and naturally it does not guarantee the optimal selection of weights in the network. Researchers also applied one or more than one objective together with the squared error, and minimized the weighted sum of all these objectives for the accurate detection of the network parameters [14]. The main limitation of these techniques is the proper selection of weights for the objective functions. The performance of these techniques heavily depends upon the chosen set of weight for the objectives. To the best of our knowledge there is hardly any specific guidelines on the selection of weights in the existing literature. An alternative formulation of this problem is to find a set of non-dominated [17] solutions with respect to the objectives used for the identification problem. In this paper, we used a multi-objective version of differential evolution proposed in [16]. However, we made slight changes in this algorithm so as to suit it to our problem. The changes incorporated are discussed in section IV.

This paper is organised as follows. Section II provides a brief review of the existing techniques for generating inferences from a gene regulatory network. Section III introduces the model of gene regulatory network used in this paper. Section IV discussed the used objectives and their effectiveness in this perspective. Section V examines the scope of multi-objective differential evolution (MODE) algorithm in the present context. In section VI, we show the results of our proposed technique using a known artificial network. In section VII we draw inferences from a gene regulatory network using real world gene expression data.

II. REVIEW OF THE EXISTING TECHNIQUES

As discussed earlier there are different techniques available in the domain of gene regulatory network identification problem. A brief review of these techniques is given below,

Boolean networks [1], [12] is the first model introduced by the researchers to represent GRN. It considers two state of 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 of the network. The main drawback of the Boolean network lies in its ignorance about the state of the genes in the intermediate levels. Further, Boolean networks consider that the genes change their state synchronously, which is biologically implausible. Currently, the researchers are making attempts to model dynamic behaviour of the genes using Boolean network [9], [10].

Bayesian networks [7], [11] are most important among the probabilistic models used for the identification problem of gene regulatory network. While modelling gene regulatory network using Bayesian network, nodes are used to represent genes, and the conditional dependence relation among the genes is represented by the edges. Although Bayesian network model have some striking features that it can handle the incompleteness, noise and the stochastic nature of the gene expression data, it has a main drawback that it is acyclic in nature. Unfortunately, cycle is a very important property of the gene regulatory networks. Recently researchers have used dynamic Bayesian network [5], [10] to allow the cycle in the inferred model of gene regulatory network. Dynamic Bayesian network is capable of handling 'hidden variable', 'prior knowledge', and 'missing data'.

Linear additive models [6], [18], [19] are also used to identify the hidden network from gene expression data. This model describes the expression of gene at time point t as a weighted sum of the expression of other genes at time points (t-1). The main drawback of this model lies in the fact that it is not capable of handling the nonlinear dynamics of gene regulation.

Recently recurrent neural network (RNN) model [15], [3] is applied for the identification of gene regulatory network from gene expression time series data. To the best of our knowledge recurrent neural network model provides the most accurate result in the existing literature. Recurrent neural network is a kind of network where the outputs of neurons are feedback to the processing units; as a result it provides the dynamic aspect. RNN model considers the neuron as the gene and the interaction weights as the regulatory strength among the genes. According to this model expression of a gene changes with time as the time derivative of that gene expression in response to the nonlinear function of the weighted sum of the expression of all the genes present in the network.

Multi-objective optimization technique is also applied for inference of gene regulatory network. In [13] researchers applied S-system model for the gene dynamics. Three objective functions are designed which includes normalized square error between reference gene expression and calculated gene expression, square difference between slope of reference gene expression and slope of calculated gene expression, and the sum of kinetic orders. A hybrid differential evolution algorithm is used for minimizing the objective functions to get a set of Pareto optimal solution set.

After reviewing different techniques used for the inference problem of the gene regulatory network, we find that instead of identifying a single optimal solution for the gene regulatory network under consideration, it will be a better approach to find a set of solutions, which are Pareto optimal [17]. To achieve this goal we designed three objective functions for the accurate detection of the network. In this paper, we used the recurrent neural network for its accuracy and inherent dynamic aspect for modelling the nonlinear dynamics of gene regulation.

III. USED RECURRENT NEURAL NETWORK MODEL FOR GENE DYNAMICS

The model used in this paper for the gene dynamics is shown in (1). This is a popular model among the researchers [15], [3], [4]. One crucial reason of choosing this model is that there are different well-documented results available in the existing literature for this model. Beside that this model considers some important aspects of gene regulation, such as: its cyclic nature, nonlinearity, self- decay and the like.

$$T_i \frac{dg_i}{dt} = f\left(\sum_{j=1}^{N} w_{ji} g_i + b_i\right) - k_i g_i \qquad (1)$$

where,

$g_i(t)$ is the expression of i^{th} gene at time point t.

$f(\cdot)$ is a nonlinear sigmoid function, i.e. $f(x) = \frac{1}{1+e^{-x}}$,

k_i is a decay constant of i^{th} gene,

T_i is the time constant for gene i,

b_i denotes the bias term for gene i, which is used to control the operating point in the linear region of the sigmoid function,

w_{ij} is the signed weight from gene i to gene j, representing the influence of gene j to gene i, and

N is the number of genes present in the GRN under consideration.

Since the available gene expression time series data is in discrete form we need to discretize the dynamics (1). In limiting case the derivative of g_i can be written as shown in (2)

$$\frac{dg_i}{dt} = \lim_{\Delta t \to 0} \frac{g_i(t+\Delta t) - g_i(t)}{\Delta t} \qquad (2)$$

After some elementary algebra, and then substituting the resulting expression in (1), we obtain (3).

$$g_i(t+\Delta t) = \frac{\Delta t}{T_i} f\left(\sum_{j=1}^{Z} w_{ji} g_j(t) + b_i\right) + g_i(t)\left(1 - \frac{\Delta t}{T_i} k_i\right) \qquad (3)$$

IV. USED OBJECTIVE FUNCTIONS

The first objective function used in this work is the normalized square error between calculated and reference gene expression. Square error is a commonly used objective function. It is chosen as one of the objective in this work due to the reason that it can measure the goodness-of-fit of the inference model with the reference model. In this work, multiple time series are used, hence the squared error is calculated over all the time series for all the genes. The expression of normalized square error is shown in (4).

$$E_1 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{T_L} \left[\frac{(g_i^j(k))_{ref} - (g_i^j(k))_{cal}}{(g_i^j(k))_{ref}} \right]^2 \qquad (4)$$

Here, $(g_i^j(k))_{ref}$ and $(g_i^j(k))_{cal}$ are the reference and calculated expression of i[th] gene in j[th] time series at k[th] time point, N is number of genes present in the network, M is the number of gene expression time series used and each time series contains T_L number of time points. In this study $(g_i^j(k))_{cal}$ is calculated using expression (3) for given network parameters (e.g. network weights, bias terms and time constants). The extra superscript j is use to indicate the time series to which this value belongs, k is used to indicate the particular time point of the j[th] time series. In this study we applied our technique to infer a well-known artificial network. For this purpose we used five set of time series data. These data is calculated for different time durations.

An alternative useful information about the inferred network is the slope of the gene expression present in the network. In this study, we calculated the normalized square error between the calculated and reference gene expression. A same type of error function is used in [13] where authors studied s-system based gene regulatory network. In this work the used normalized square error of slope is shown in expression (5).

$$E_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{T_L} \left[\frac{(\dot{g}_i^j(k))_{ref} - (\dot{g}_i^j(k))_{cal}}{(\dot{g}_i^j(k))_{ref}} \right] \qquad (5)$$

Here, $(\dot{g}_i^j(k))_{ref}$ and $(\dot{g}_i^j(k))_{cal}$ are the reference and calculated gene expression slope of i[th] gene in j[th] time series at k[th] time point, the other symbols conveyed same meaning as for expression (4). Both of these values are calculated using expression (6) shown below.

$$(\dot{g}_i^j(k)) = \frac{g_i^j(k+1) - g_i^j(k)}{\Delta t_j} \qquad (6)$$

Here, Δt_j is the time duration in j[th] time series (i.e. the time duration between two consecutive time points at which gene expressions are observed).

Biologically it is known that a gene is regulated by very few genes in a regulatory network. To incorporate this biological fact into our model we used the expression of (7) as our third objective function.

$$E_3 = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{w_{ij}}{|1 + w_{ij}|} \qquad (7)$$

We used the expression (7) in our previous reported work [3], and achieved good result.

V. USED MULTIOBJECTIVE DIFFERENTIAL EVOLUTION (MODE)

In this study we used the multi objective differential evolution algorithm proposed in [16] to extract the network parameter from gene expression data. In this reported work, [16] authors proposed an enhanced version of classical de/rand/1 algorithm to incorporate the concept of non-dominated sorting. Instead of using the classical Pareto dominance [17] they used ϵ-dominance to retain the non-dominated solutions found and to distribute them in a uniform way. The concept of ϵ-dominance does not allow two solutions with a difference less than ϵ_i in the i[th] objective to be non-dominated with respect to each other thereby allowing a good spread of solutions. The algorithm of [16] is described in Fig1 using flow chart. For more details readers are requested to go through [16].

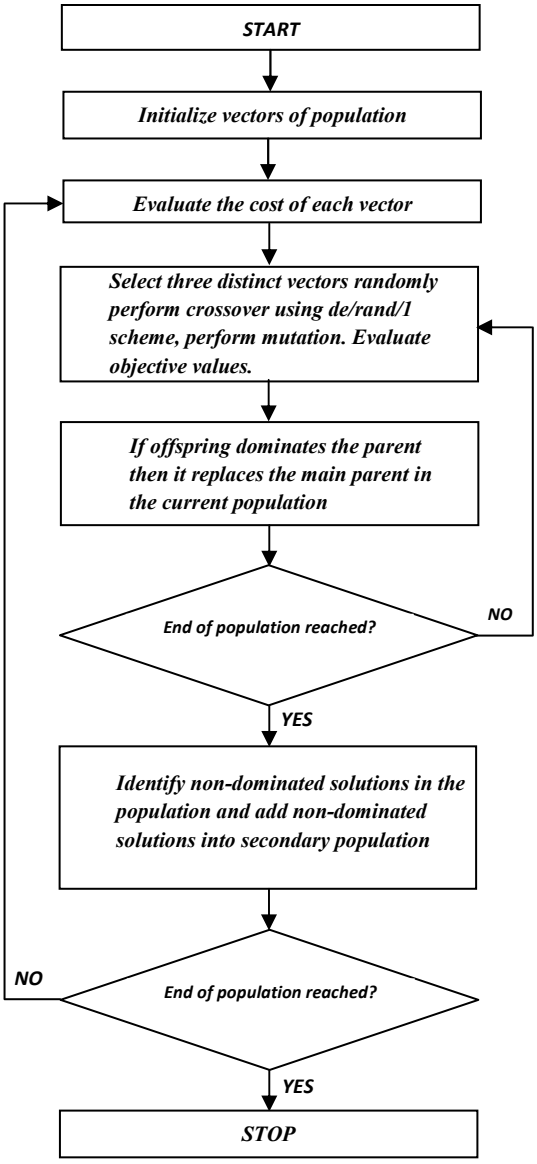

Fig1. Flow chart of the multi objective differential evolution algorithm of [16]

In this study, we, however, slightly changed the algorithm to fit our proposed model keeping the core evolutionary part intact. In [16] there is no guideline about the size of secondary population. In our algorithm we used a population of same size as of the original population to store the non-dominated elite parameter vectors. Other than that instead of ϵ-dominance described in [16] we use the classical Pareto optimality described in [17]. We also used a stopping criterion to choose the optimal network from the elite population pool. The used algorithm in this study is presented below:

Step1: initialize the population randomly between MAX_LIMIT, and MIN_LIMIT. MAX_LIMIT and MIN_LIMIT are the upper limit and lower limit of the value of fields of the parameter vectors.

Step2: calculate the current value of the scaling factor F = (U_L – L_L) × ((MAX_ITER – CURRENT_ITER + 1)/ MAX_ITER) + L_L. U_L, L_L are the upper limit and lower limit of the value of the scaling factor, MAX_ITER and CURRENT_ITER are the maximum allowable number of iteration and current iteration number.

Step3: if the CURRENT_ITER ≤ DECISION_ITER then choose three random parameter vectors r1, r2, and r3 from current population, otherwise choose three parameter vectors from ELIT_POPULATION. ELIT_POPULATION is the population of the Pareto optimal solutions.

Step4: generate the trial vector using scaling and recombination. Scaling and recombination operations are same as in de/rand/1.

Step5: if the trial vector dominates the current parameter vector with respect to the objective functions then it replaces the current parameter vector in the current population, otherwise current parameter vector retain its place.

Step6: repeat Step2 to Step5 for each of the parameter vector in the current population.

Step7: choose the Pareto optimal parameter vectors from the current population and store them in the ELIT_POPULATION if it is empty, otherwise choose the non-dominated parameter vectors from the combined population of chosen Pareto optimal parameter vectors from current population and the already existing parameter vectors of ELIT_POPULATION.

Step8: if the number of chosen Pareto optimal parameter vectors from current population is less than ELIT_POP_THRESOLD for certain iterations, then choose a parameter vector randomly from current population change the value of one field randomly. The field to be changed is chosen randomly.

Step9: repeat Step2 to Step9 until CURRENT_ITER becomes equal to MAX_ITER or values of the three objectives (equation 4, equation 5, and equation 7) of any parameter vector from ELIT_POPULATION become less than or equal to OBJECTIVE_TH1, OBJECTIVE_TH2, OBJECTIVE_TH3 respectively. In that case, choose that parameter vector as the final solution. OBJECTIVE_TH1, OBJECTIVE_TH2, OBJECTIVE_TH3 are three predefined threshold for the three objectives described in Section IV.

Step10: if CURRENT_ITER is equal to MAX_ITER then choose a parameter vector as the final solution from ELIT_POPULATION with minimum first objective value (equation 4)

During run of the above algorithm, we fixed a field of a parameter vector to zero if its value is in between -0.4 to +0.4 for computational feasibility

VI. RESULTS

We used our proposed technique with an artificial known network to test its accuracy. The known network used here is a well-known 4-gene network used in several reported works [15], [3], [4]. The used known network is shown below in TABLE1.

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

We used the gene network database of TABLE I to generate five set of gene expression time series data, each containing fifty time points using expression (3) and store them as reference. Then we applied the multi objective differential evolution algorithm described in section V to get back TABLEI. For this 4-gene network each parameter vector of the multi objective differential evolution contains 24 fields (16 fields for interaction value among the genes, 4 fields for bias terms, and 4 fields for time constants). As the differential evolution algorithm is a stochastic algorithm we run our algorithm 50 times and observed the obtained result. In each cell of TABLE II, the values in the braces indicate the percentage of time we observed the sign of a value. In TABLE II, we put the sign of the values, which occurs more number of time than other. For example in row number two and column number two, the positive value came 38 times out of 50 runs, where as negative and zero value came 10 and 2 times respectively. Therefore we decided it is more probable that this interaction value will be positive. The values in the other cell of TABLE II are obtained in the same way.

TABLE II

IDENTIFIED NETWORK USIN PROPOSED METHOD

	Gene1	Gene2	Gene3	Gene4
Gene1	+(76)	-(25)	0 (75)	0 (15)
Gene2	+(100)	-(75)	0(75)	0(75)
Gene3	0(75)	0(75)	0(75)	0(75)
Gene4	0(75)	0(50)	+(50)	0(75)

.

As can be seen, most of the signs of the interaction weights are identified with sufficient accuracy. We rounded off the percentage values for a convenient representation. The standard deviation of each weight over the 50 runs is also under predefined threshold. The code is written in C language and implemented on a laptop containing 1 GB ram with Pentium dual core processor (1.73 GHz). Each run of the program takes at an average 4.5 hours.

VII. INFERENCE OF ARTIFICIAL NETWORK USING REAL GENE EXPRESSION

We applied our proposed method to infer S.O.S DNA repair network [2] of E.Coli bacteria. 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. The network is shown in TABLE III.

TABLE III

ORIGINAL E.COLI S.O.S DNA REPAIR NRTWORK

	uvrD	lexA	umuD	recA	uvrA	uvrY	ruvA	polB
uvrD	0	0	0	0	0	0	0	0
lexA	-	-	-	-	-	-	-	-
umuD	0	0	0	0	0	0	0	0
recA	0	0	0	+	0	0	0	0
uvrA	0	0	0	0	0	0	0	0
uvrY	0	0	0	0	0	0	0	0
ruvA	0	0	0	0	0	0	0	0
polB	0	0	0	0	0	0	0	0

Zero values are used in some of the cells of TABLE III to represent that those genes do not interact with each others. We used our algorithm with the same setting used in

section VI for this inference problem. The result obtained is shown in TABLE IV. Those values appear more than 60% of time of total number of run of our algorithm we put it in the appropriate cell of TABLE IV. As can be seen from TABLE IV five nonexistent connections and six negative connections are detected correctly, which is a challenge in the existing literature.

TABLE IV

DECETED E.COLI S.O.S DNA REPAIR NRTWORK

	uvrD	lexA	umuD	recA	uvrA	uvrY	ruvA	polB
uvrD	-	+	-	+	**0**	-	-	+
lexA	-	-	**0**	-	-	-	+	-
umuD	-	-	-	+	-	-	-	-
recA	-	-	-	-	-	-	+	-
uvrA	+	+	+	+	+	-	-	+
uvrY	-	-	-	+	-	**0**	-	+
ruvA	-	+	**0**	**0**	-	-	+	+
polB	-	+	+	-	-	+	-	**0**

CONCLUSION

In this work, we presented a recurrent neural network model of GRN. We applied multi-objective differential evolution algorithm to identify the network parameters. The proposed technique eliminated the problem of using weighted sum of different objective values by using the Pareto optimality. The proposed technique provides good results.

REFERENCES

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

[2] 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.

[3] Datta, D., Choudhuri, S.S., Konar, A., Nagar, A.K and Das, S., "A Recurrent Fuzzy Neural Model of a Gene Regulatory Network for Knowledge Extraction Using Differential Evolution," Proc. of IEEE Congress on Evolutionary Computation, May 18 – 21,2009, Trondheim, Norway.

[4] Datta, Debasish, Konar, Amit and Janarthanan., R., "Extraction of Interaction Information among Genes from Gene Expression Time Series Data," NABIC 2009.

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

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

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

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

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

[10] Husmeier D., "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.

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

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

[13] Liu, Pang-Kai and Wang, Feng-Sheng., "Inference of Biochemical Network Models in S-System Using Multi Objective Optimization Approach,"Oxford Journal,March 5,2008.

[14] Noman, Nasimul and Iba, Hitoshi., "Inferring Gene Regulatory Networks Using Differential Evolution with Local Search Heuristics," IEEE/ACM transactions on computational biology and bioinformatics, vol.4, no.4,pp 634-647,Octobor-December 2007.

[15] 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.

[16] Santana-Quintero, Vicente Luis and Coello Coello, Carlos A., "An algorithm Based on Differential Evolution for Multi-Objective Problems," International Journal of Computational Intelligence Research, ISSN 0973-1873, Vol.1, No.2, (2005), pp 151-169.

[17] Srinivas, N and Deb, Kalyanmoy., "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," IEEE Evolutionary Computation, vol. 2,no. 3, pp 221-248,1994.

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

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