Modeling and Simulation of Gene Regulatory Network: A Comprehensive Survey

Surama Biswas, Tamal Pal, Sriyankar Acharyya

Abstract—The Gene Regulatory Network (GRN) specifies the series of regulatory interactions between different genes. A target gene is interacted by a signal which is originated from the expression of its regulator gene. A gene is known to be expressed when it synthesizes a protein and the degree of synthesizing the protein determines the level of its expression. The same gene can behave as a 'target' in one state of interaction and a 'regulator' in the next state. If there are many interacting genes in a biological system, a network can be formed out of them where the genes are treated as nodes and interaction between any two genes is treated as an edge. This network is known as Gene Regulatory Network. Simulation of GRN addresses the issue of reconstructing the network on the basis of the expression levels of the interacting genes. Various mathematical tools are used to design the system and different optimization techniques are used to find the optimal design. The process of designing starts with time-dependent (Time-series) and condition-dependent (Steady state) gene expression data, available from micro-array chips. The target gene is activated depending on the collective interactions made to it. The problem can be modeled using Neural Network and application of Fuzzy logic may improve the design. There are two issues to discuss. One is related to uncover the parameters involved in GRN called parameter estimation problem. The other is to predict the network structure step by step while learning the parameters. Applications of meta-heuristic algorithms are proved to be efficient in resolving both the issues.

Index Terms—Bayesian network, correlation, direct scale free network, fuzzy logic, gene expression, microarray, neural network, optimization.

1 Introduction

GENES, the functional and primarily protein coding parts of the DNA are the blue prints of life in an organism. Gene expression is the way of response of a gene to a physiological phenomenon. The expression of a gene is composed of two main steps, viz., Transcription and Translation. Transcription is a bio-chemical change happening in the coding region of a gene (exon) by some external or internal influence which produces some intermediate products like primary RNA, messenger RNA or mRNA etc. [1], [2], [3]. Another bio-chemical reaction named translation when acted on the mRNA, the final product is formed that is known as protein. The extent of expression of a gene is a measurable quantity and named as expression level of the gene at the instance. Due to different cellular activities [4], [5], [12] the expression level of a gene may change and the protein thus produced may bind to the promoter site of any other gene causing it to be expressed. According to modern theory the sole cause of regulation is not the protein alone. In other words gene regulation does not occurs only in translational or post translational levels, it occurs also in transcriptional level as well by a short format RNA called micro RNA. The regulatory interaction may change the protein synthesis of the target gene in two ways, either to increase or to decrease [13], [15], [16]. They are named as enhancing or repressive regulatory interactions respectively.

The simulation of GRN can be done in two ways. One way is to construct the network from the microarray data by measuring similarity between expression levels at different time instances or conditions. This is called forward modeling. The other way of simulation is the Inverse modeling. In this process, network is constructed arbitrarily and expression level of the target gene is generated using certain mathematical formula. The error is calculated by comparing the simulated data with the original gene expression data. The structure of the gene regulatory network obtained from forward modeling provides a static road-map where the actual interest lies in the pattern of traffic, the cause of emerging such patterns, the way to control them so on and so forth. The structure of the network is similar to listing up the parts of a machine which does not give a clear picture of how the machine performs a particular function and how a subset of machine parts interact to each other when performing the function[11]. If the network is simulated in steps, the dynamics of the system may be understood. At each time step, a new edge is attached to the network. This depicts the dynamics of the gene interactions in cellular activities.

All the genes of an organism are sparsely present in the genome. The genome is embedded in each cell of the body of an organism [8], [9], [10]. But a gene may be expressed in one tissue and remain unexpressed in any other. In this context there are some questions to be answered. Which gene is expressed? When will the gene be expressed, in which tissue of the organism and to what extent [14]. For example, Insulin, the best studied poly-peptide hormone in human being, is produced by Human Insulin Gene (INS). It is produced as a result of increasing sugar level in blood and acts to lower the level of

- *Surama Biswas is currently working as a Research Assistant in Computer Science department in West Bengal University of Technology, WB, and India, PH-+ 919433045471. E-mail: surama.biswas@gmail.com.*
- *Tamal Pal is currently working as an Assistant Professor of Computer Science and Engineering department in Dream Institute of Technology, WB, and India. PH-+ 9163692941. E-mail: tamalpal91@yahoo.co.in.*
- *Dr. Sriyankar Acharyya is currently working as an Associate Professor in Computer Science department in West Bengal University of Technology, WB, and India, PH-+ 919903389062. E-mail: srikalpa8@gmail.com*

the sugar. The gene (INS) is expressed in the Beta Cells in Pancreas only, though it is present in any other cell of the body.

The study of the regulatory system consists of the following components [11]:

i) Structure of the System: The gene regulatory network (GRN) is represented by a graph G, where each gene i ∈ N (N is the maximum number of genes in the system) represent a node and regulatory interaction between i and j is an edge if gene j has any influence to the expression of gene i.

ii) System Dynamics: Time dependent (producing time series data) and condition dependent (producing steady state data) behavior of a set of genes is represented in system dynamics.

iii) Controlling Methods: The internal state of the cell may be controlled to minimize the malfunction as well as to facilitate drug finding applications.

iv) Design Methods: Simulation of GRN from gene expression data uses different probabilistic models and employs various local search algorithms to optimize the construction procedure.

Some applications of GRN are described as under:

a) Biomarker detection: Biomarker detection is an important application of the study of gene regulatory network. The gene regulatory network depicts the cellular activity undergoing in the tissues. The regulatory interaction pattern changes from normal to diseased tissues. The gene expression data from normal tissues and diseased tissues, tissues after therapeutic intervention and cure is obtained from microarray and regulatory interactions are modeled. It may be seen that a strongly connected component of the four different graphs corresponding to four different cases mentioned above, has been changed from one state to another. This sub-graph indicating the state of a disease is called bio-marker of the disease. Biomarker detection using Clustering algorithms has been done [49]. The study of gene regulatory network in the context of biomarker detection is being an interesting area of research [30].

b) Drug finding: The adjacency matrix of gene regulatory network is generally seen to be sparse in nature. Some nodes are strongly connected to the rest of the network. These genes are very few in number. They are called the hubs of the network. Drugs are designed concentrating on these hubs and aiming to reduce or increase the expression of the gene so as the disease can be taken under control. The hub genes detection in the study of gene regulatory network is especially interesting for drug finding.

c) Detection of side effects of a drug: After the application of a drug it gradually starts taking part in cellular activities. It may be considered as being a part of the regulatory network and sending regulatory interactions to the gene(s) targeted. But along with the targeted gene it may send undue regulatory interaction to other genes present in the network and even to the tissues not having affected by the disease. So side effect detection in the study of regulatory network is an important issue.

The rest of the paper is organized in following manner: Section 2 describes the GRN reconstruction framework which encompasses the data extraction and preprocessing, the co-expression matrix generation from processed data and finally probabilistic modeling. Section 3 defines the different soft-computing techniques including Recurrent Neural Network, Fuzzy Logic, Optimization Techniques and a software aiding the reconstruction, named as, GeneNetWeaver. In Section 4, the experimental issues including the source of data, the metrics concerned in the performance evaluation of the simulation and a performance analysis from reviewed literature.

2. GRN: MATHEMATICAL FRAMEWORK

The microarray data, collected from image analysis are prone to errors. So the data is processed before being used in the simulation. The preprocessed data is used for co-expression matrix generation which gives the possible relation between genes. The graph generation process is based on either association computation between two samples of gene expressions or probabilistic reconstruction of the graph. Mathematical modeling is required to be applied in Data extraction and preprocessing as described in Section 2.1, Co-expression matrix generation as described in 2.2 and Probabilistic graph generation as described in 2.3.

2.1 Data Extraction and Preprocessing

Gene expression level denotes a quantity which indicates the level of expression (protein generation process) for a particular gene in the cell and is represented in a floating point number obtained from microarray. The final input matrix contains N rows where each gene represents a row and T columns or samples where each column correspond either to a time point or to a condition. Here each entry represents the gene expression level of a gene i at sample t. But the raw data collected from microarray is prone to contain various noise elements. So there should be methods of extraction [30] of original gene expression levels sampled earlier.

Consider, Y_t denotes a set of microarray samples of all target genes measured at sampling time t and X_t be the gene expression levels at t, then Y_t is mapped to X_t. The expression level is sampled at certain time intervals to generate time series data. There are several models regarding extraction of gene expression from microarray data.

2.1.1 Fully correlated model

This model is based on the assumption that microarray data is fully identical with the actual gene expression i.e. $Y_t = X_t$ .This model is too simplified and does not correspond to real microarray experiments since it ignores noise factor.

2.1.2 Linear Gaussian model

White Gaussian noise is the linear combination of Gaussian noise (i.e. probability density function of the noise amplitude follows normal distribution) and white noise (noise with power carried by wave is constant per unit frequency).Y_t is related to X_t by a linear combination considering white Gaussian noise to be mixed with microarray data.

$$Y_t = M \cdot X_t + a_{obs} + v_t$$

M: Projection matrix is assumed to be identity matrix.

a_{obs}: A vector to adjust measurement error during observation.

v_t: White Gaussian noise vector

.

2.1.3 Gaussian model with discrete expression levels

In this model expression of each gene i at time t, $x_i(t)$ is related to corresponding micro-array data $y_i(t)$ with probability determined by Gaussian function. An individual probability for each gene indicates that this model accepts different noise levels for different genes of the network making the model nonlinear and potentially better.

After extraction of the data it undergoes some preprocessing operation by calculating signal to noise ratio (SNR) and normalization [49]. Suppose two data sets of normal and tumor cells named as class1 and class2 are present in hand, then SNR is defined as

$$(SNR) = \left(\frac{\big(mean(class1) - mean(class2)\big)}{\big(S.D(class1) + S.D(class2)\big)}\right)$$

Here S.D represents the standard deviation. After the calculation of SNR, the dataset is arranged in the descending order of SNR and a subset of the data with high SNR is collected. Next task is to normalize each expression data to scale in a certain range (say 0 to 1). If the j^{th} sample, named X_j contains x_{ij}, the expression level of ith gene, the formula of normalization of x_{ij} is given by:

$$Normalize\big(x_{ij}\big) = \frac{\big(x_{ij} - minimum\big(X_j\big)\big)}{\big(maximum\big(X_j\big) - minimum\big(X_j\big)\big)}$$

This normalized data is arranged into tabular form which is suitable for processing.

2.2 Co-expression Matrix Generation

The two dimensional array of gene expression data $x_{ij} \in X$ is available after pre-processing where each row represents a gene and each column (sample) represents either time points (in time series data) or conditions (in steady state data). Now the similarity or association between the gene expression levels of two genes i and j is to be estimated. If the value satisfies a predefined threshold then the two nodes may be connected by an edge. The adjacency matrix formed in this context is called co-expression matrix. The similarity is measured [31], [36], [44], [45], [46] by different means as under:

2.2.1 Pearson Correlation Co-efficient

The Pearson correlation coefficient $S(X_i, X_j)$ is estimated as:

$$S\big(X_i, X_j\big) = \frac{1}{N}\sum_{k=1,N}\frac{\big(x_{ik} - mean(X_i)\big)}{S.D(X_i)} \cdot \frac{\big(x_{jk} - mean\big(X_j\big)\big)}{S.D\big(X_j\big)}$$

X_i is the set of expression levels of gene i in different samples

N is the number of samples

x_{ik} is the expression level of gene i at k^{ith} sample

$mean(X_i)$ is the mean of N expression levels of gene i

$S.D(X_i)$ is the Standard deviation of N expression levels of gene i

The Pearson Correlation Coefficient is compared to a threshold which may be hard (in the range of .6 - .8) in case of un-weighted network or as a function of different parameters in case of weighted network. In both the cases the coefficient is compared with the threshold σ as under.

$$S\big(X_i, X_j\big) > \sigma$$

If the condition satisfies an edge will be drawn between the genes i and j.

2.2.2 Partial Correlation Coefficient

Correlation between two genes generally specifies either a direct relation between them or a common dependence on some third gene. But the Partial Correlation only considers the direct relation between two genes and disobeys the theory of joint dependence of any third gene.

2.2.3 Information Theoretic Approach

In information theoretic approach Mutual Information(MI) is the measure to determine the statistical dependence between two variables. If there are M states of a system A, i.e. {a1, a2, a3, . . ., a_m} MI is computed using Shannon Entropy H(A) which is given as:

$$H(A) = -\sum_{i=1}^{M_A} pb\,(a_i) \cdot log\big(pb(a_i)\big) \quad \text{(i)}$$

$pb(a_{ir})$ is the probability for choosing a_{ir}. Similarly joint entropy can be estimated for two systems A and B from joint probability $pb(a_i, b_j)$. The Mutual Information is defined as:

$$MI(A,B) = H(A) + H(B) - H(A,B) \geq 0 \quad \text{(ii)}$$

The systems A and B are independent if MI(A, B) is 0 and the statistical dependence increases with the increase in MI. Two genes are taken as system A and B which have either some continuous values in [0, 1] or some discrete value taken in the interval [0, 1].

2.3 Probabilistic Modeling

The network may be constructed either using forward modeling or inverse modeling. The basic graph representation techniques of GRN like Directed Graph and Directed Hyper-graph are discussed in subsections 2.3.1 and 2.3.2. Two Forward Modeling techniques, viz. Bayesian Network and Boolean Network and one Inverse Modeling technique, Directed Scale Free Graph are discussed in sections 2.3.3, 2.3.4 and 2.3.5 respectively. The reconstruction of regulatory network is based on a mathematical or probabilistic model. This is to avoid blind trial and error assumptions.

2.3.1 Directed graph

Here in the directed graph G = (V, E) each gene is represented as a node i ∈ V (V maps to N, is the maximum number of genes in the system) [14]. Each directed edge is represented by a tuple {i, j, sn}, where i and j are the target and regulator genes respectively and the symbol sn, (sn ∈ {+,-}) represents the sign of the interaction. The interaction is either enhancive (+) causing the rate of protein synthesis in target gene i to increase or repressive (-) causing the rate of protein synthesis in target gene i to decrease.

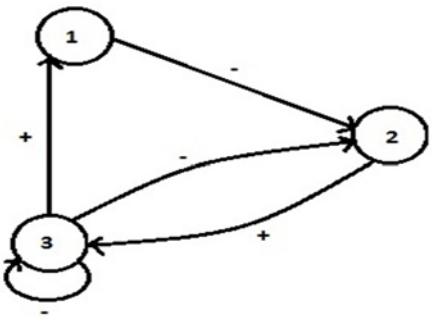

Figure 1: Directed Graph Representation of Gene Regulatory Network.

The directed graph representation of gene regulatory network is shown in figure 1. The set of vertices is represented as V = {1, 2, 3} and the set of edges E = {(2, 1, -), (1, 3, +), (3, 2, +), (2, 3, -), (3, 3, -)} where each edge is (i, j, sn). Her i and j are node numbers and sn is the sign of interaction shown in figure 1.

2.3.2 Directed Hyper-graph

The directed hyper-graph G' = (V', E') is composed of a node set V' where each node i ∈ V (V maps to N, is the maximum number of genes in the system) represents a gene as in 2.3.1 and each directed edge is represented by a tuple {i, J, Sn}. Here i is the target gene which is expressed as a result of the regulatory interactions made on it and J is the set of regulators of i, collective interactions of whom cause the expression of i and Sn is the list of signs of the regulatory influences [14]. If the graph of figure 1 is considered as the hyper-graph G', the set of vertices V' = {1, 2, 3} and set of edges E' = {(2, [1,3], [-, -]), (1, [3], [+]), (3, [2, 3], [+, -])}. For example the edge {(2, [1,3], [-, -]) in the edge set E represents that the target gene 2 in figure 1 is interacted by the regulators 1 and 3 and both the regulatory interactions are repressive i.e. having '-' sign.

2.3.3 Bayesian Network

It is a probabilistic approach of graph construction [14] [30] which has become one of the efficient representations of reconstruction of gene regulatory network. Bayesian network represents causal relationships between the nodes [37], [38], [39], [40], [41] rather than a flow of information. In gene regulatory network this causal relation is drawn between the expression levels X_i of the gene i involved in the system. In other words for a gene set the expression levels are obtained under different conditions. The causal relationship is reconstructed depending on some probability measures. The Bayesian network is a directed acyclic graph (DAG), G = (V, E) where each node i ∈ V is associated with the expression level of gene i, 1 ≤ i ≤ N where N is the maximum number of genes in the system. Each edge e_{ij}∈ E is an interaction between gene i and j. The two components of a Bayesian Network (BN) are the DAG and a joint probability distribution of the expression levels involved in the system. The conditional probability for each X_i is given by: pb(X_i | par(X_i)) where par(X_i) represents the parent of X_i in the graph G, is a candidate from the set of regulators in the system. Here actually we are computing the probability of par(Xi) being the parent of X_i from different microarray dataset of same genes. And an edge is drawn from parent to child if the probability exceeds a certain threshold.

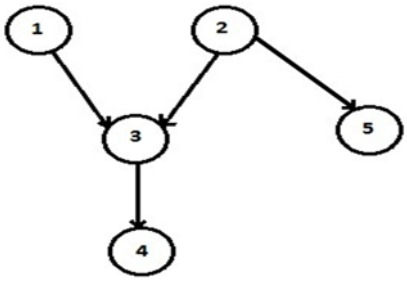

Figure 2: Sample network of 5 nodes. The conditional probability of each node of the network pb(Xi | par(Xi)) is determined.

The probability measures required for constructing such a graph is written as under:

The Conditional Probability measures of different variables as in figure 2 are described as under:

pb(X1), pb(X2), pb(X3 | X1, X2), pb(X4 | X3), pb(X5 | X4).

The Joint Probability which represents the network as a whole is given as in figure 2:

Pb(X1,X2, X3, X4, X5) = pb(X1)pb(X2)pb(X3 | X1, X2)pb(X4 | X3)pb(X5 | X4).

An important term involved in this context, Conditional Independence is given by the formula:

Ind(Xi : Non-descendants(Xi) | par(Xi)) which comes from the famous Marcov assumptions.

Statement of Marcov Assumptions: Each variable (Xi) is independent of the Non-descendant of (Xi) given par(Xi).

So the set of Conditional Independence according to figure 2 is given by:

Ind(X1:X2, X5), Ind(X2: X1), Ind(X3: X5 | X1, X2), Ind(X4: X1, X2, X5 | X3), Ind(X5: X1, X3, X4 | X2)

The probability distribution that obeys the Marcov assumption and the chain rules of conditional probability can be written as:

$$Pb(X) = \prod_{i=1,n} pb\left(X_i \vee par(X_i)\right)$$

Each node of the Bayesian network carry a probability table based on which expression level of a successor gene is deter-

mined. Consider an instance where x is regulated by y and z in figure 3.

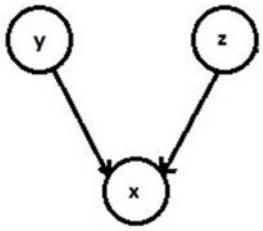

Figure 3: A gene regulating instance when genes y and z regulates x.

Expression levels are quantized into q=3 levels. The probability table of x looks like Table1. For the instance when y=1, z=2 then probability of x=1 is 0.3, x=2 is 0.5 and x=3 is 0.2.

TABLE 1
PROBABILITY FOR GENEX

y	z	x=1	x=2	x=3
1	1	0.6	0.3	0.1
1	2	0.3	0.5	0.2
1	3	0.8	0.2	0
2	1	0.15	0.35	0.5
2	2	0.7	0.2	0.1
2	3	0.1	0.6	0.3
3	1	0.5	0.2	0.3
3	2	0.2	0.4	0.6
3	3	0.17	0.23	0.6

Two Bayesian Network (BN) are said to be equivalent if they contain the same set of conditional independences, i.e. If Ind(G) = Ind(G') two BNs are equivalent. In this context the statement of Perl and Verma is important.

Statement of Perl and Verma [37]: Two DAGs are equivalent if and only if they have the same underlying undirected graph and same v-structures (i.e. same set of converging edges into one node. e.g. $X_j \rightarrow X_i \leftarrow X_k$).

This implies the equivalence graphs with some fixed directed edges like $X_j \rightarrow X_i$ and some undirected edges like $X_j - X_i$ which takes the form of $Xj \rightarrow X_i$ or $Xi \rightarrow X_j$. In the graph of figure 2 to keep the v structure $X_1 \rightarrow X_3 \leftarrow X_2$ intact, all the nodes should have the same direction as in the graph. But if a 6th node be connected from 4 then an equivalence set might be there with directed edge 4 to 6 or 6 to 4. Each member of the equivalence set of graphs is then evaluated for the score with respect to previous data D. The graph from the equivalence set with best score is the target network.

2.3.4 Boolean Network

Boolean network is a suitable medium of the reconstruction of gene regulatory network [14]. It is represented as graph G (V, E, F) where V is the set of nodes represented as genes. Each node $i \in V$ is associated with the expression level of gene i, $1 \leq i \leq N$ where N is the maximum number of genes in the system. Each edge $e_{ij} \in E$ is an interaction between gene i and j. At time instance t the state of gene i is denoted by $£_i \in \{0, 1\}$ which is determined by a Boolean function $f_i \in F$ involving k genes. So there are 2^K possible values of $£_i$. $£_i = 1$ means gene i is expressed and $£_i = 0$ means gene i is not expressed. An edge is extracted if gene i is expressed or $£_i$ flips from 0 to 1. Another function f_j, taking this output as an input along with (k-1) other inputs evaluates the state of some other gene. The genes $i \in \{1,2,....,n\}$ at time point t constitute an n element vector X(t) $\{£_1, £_2, ….,£_n\}$ which represent the state of the system at time point t. At time point t+1 system changes its state to X(t+1) by the set of functions F.

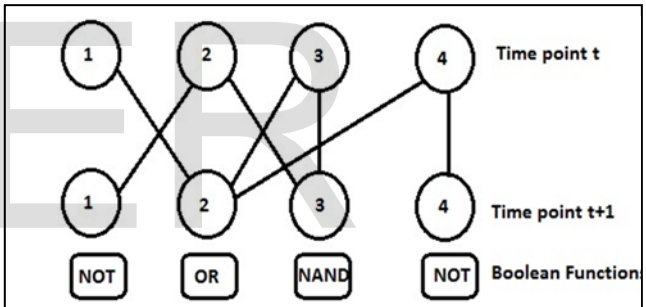

Figure 4: Wiring Diagram of a Sample Boolean Network

Figure 4 and Figure 5 shows the change of state from time point t, X(t) to the state at time point t+1, X(t+1). The four functions in the set F are some standard Boolean gates. In real practice they may be some complex Boolean functions.

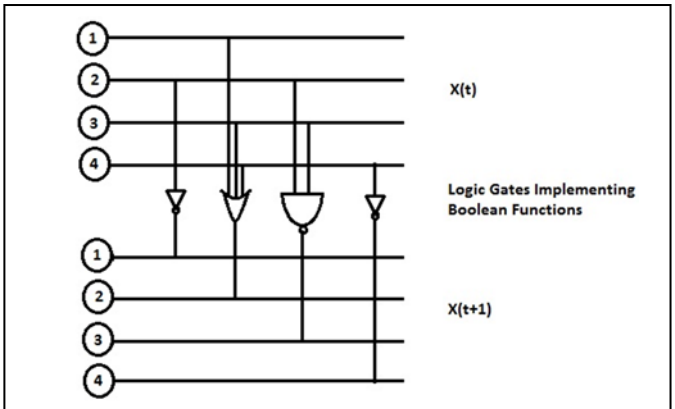

Figure 5: State Change from t to t+1 in the Circuit Diagram of a Sample Boolean Network

Each node of the Boolean network carry a data table based on which expression level of a successor gene is determined. Consider a regulatory interaction where x is regulated by y and z in figure 1.

Expression levels are quantized into 0 and 1 where zero indicates gene is not expressed and one indicates gene is expressed. The probabilities of x taking different values are presented in Table2. For instance at time t if y is not expressed and z is expressed then gene x is expressed at t+1.

TABLE 2
ACTIVATION OF GENE X INA BOOLEAN NETWORK

y	z	x
0	0	0
0	1	1
1	0	1
1	1	0

2.3.5 Direct Scale Free Network (DSF)

Scale free nature of directed graph obeying Power law has well been implemented in many types of graph representation [14]. Novel Graph construction applications like web graph, where websites or web pages are the vertices and the direct link between them represent the edges of the graph, the movie actor graph, scientific collaboration graph are examples where the scale free nature of directed graph are implemented. In gene regulatory network construction direct scale free graph is implemented in recent past [26]. Graph construction process of gene regulatory network is based on degree based preferential attachments. Initially we have a set of genes named N_{new} with which the GRN is to be constructed. The graph construction starts with an initial graph $G(t_0)$ at time point t_0. It has a single node randomly chosen from N_{new} with the initial degree ≥1 (to make the calculation meaningful). In each step a new edge (regulatory interaction between two genes) will be attached to the graph and after t time units the graph will exactly have t number of edges. Whenever a node is fetched from N_{new} and being used as the source (regulator) or target node of the edge to be attached it will be deleted from N_{new} and being placed in a set named N_{old}, the set of existing nodes which have already been used in the graph. To attach an edge to the graph G(t) there are three steps:

i) Choosing a regulator,
ii) Choosing a target,
iii) Placing a directed edge from regulator to target.

Here a new node (regulator/target) i $\in N_{new}$ having degree(i) = 0 and an existing node (regulator/target) j $\in N_{old}$ having degree(j) ≥1.

The phrase 'degree based preferential attachment' indicates that edge connection and the choice of regulator and target is entirely based on degree [42], [26]. In case that a new node is chosen the question of degree does not arises and therefore the selection of new node may be random. But in case that an existing node if chosen as the target the selection is based on its in-degree of the node and if chosen as the regulator the selection is based on the out-degree of the node.

Selection of regulator and target node (gene) may be done in three possible phases:

Case A: New regulator and existing target,
Case B: Existing regulator and existing target
Case C: Exiting regulator and new target

There are some probability constants like α, ß, γ, δ_{inp} and δ_{out} involved in this computation. They are non-negative real numbers provided α+ß+γ = 1 and δ_{inp}, $\delta_{out} \geq 0$. Each of the three cases A, B and C are executed with the probability α, ß and γ respectively. The above mentioned cases are described as under:

Case A: The case is chosen with probability α. New regulator v, is chosen from the set N_{new} randomly. In case of choosing existing node as target w from the set N_{old} a probability measure is to be estimated. The probability that a node w_i taken from the set N_{old} is the required target node w is given by:

$$Pb(w = w_i) = (Indeg(w_i) + \delta_{inp}) / (t + \delta_{inp}.n(t))$$

Here, $Indeg(w_i)$ is the indegree of the node w_i , t represents the time instance and n(t) represents number of nodes the graph contains in the current time point t.

Case B: The case is chosen with probability ß. The regulator v, being an existing node is chosen from the set N_{old} with some probability. The probability that a node v_i taken from the set N_{old} is the target node v of our interest is given by:

$$Pb(v = v_i) = (Outdeg(v_i) + \delta_{out}) / (t + \delta_{out}.n(t))$$

$Outdeg(v_i)$ is the out-degree of the node v_i in the current graph.

In case of choosing existing node as target w from the set Nold a probability is to be estimated. The probability that a node w_i taken from the set N_{old} is the required target nod w is given by:

$$Pb(w = w_i) = (Indeg(w_i) + \delta_{inp}) / (t + \delta_{inp}.n(t))$$

Case C: The case is chosen with probability Y. The regulator v, being an existing node is chosen from the set N_{old} with some probability. The probability that a node v_i taken from the set N_{old} is the target node v of our interest is given by:

$$Pb(v = v_i) = (Outdeg(v_i) + \delta_{out}) / (t + \delta_{out}.n(t))$$

The target w, being a new node is chosen from the set N_{new} either randomly or with some estimated probability.

3 SIMULATION

Reconstruction of Gene Regulatory network involves a step by step procedure of graph construction where at each instance a new weighted edge is added to the network. This may require a new node to be attached in the network. The final outcome is the simulated network with a set of parameters. The different parameters involved are i) weight parameters $w_{ij} \in W$, where W is an NxN matrix expressing the weight of each edge in the network, ii) Basal expression parameter b_i, expression level of gene i in non-excited condition which is an 1xN vector and iii) timing parameter t_i, the delay factor of gene i which is also an 1xN vector. Here N is the number of genes involved in the network. So total number of parameters are N^2+2N or $N(N+2)$. The problem domain may be decomposed into two parts, network reconstruction and parameter estimation. Different soft computing tools and techniques like Artificial Neural Network, Fuzzy logic and different advanced search and optimization techniques are implemented in this area. There are some softwares which aid the work by supplying ready to use data and platforms for comparing original and simulated network. The following subsections depict the different soft computing tools, algorithms and software which aids the reconstruction:

3.1 Useful Software

Since construction of Gene Regulatory Network is a well-known problem domain over one and half decades. There are softwares like GeneNetWeaver, GNA (Genetic Network Analyzer), GINsim, GeNESis etc. which have been developed to aid the GRN reconstruction. One such well known open source software is GeneNetWeaver [28], [29] available in http://sourceforge.net. The software supplies time series gene expression data of organisms like Yeast and E.coli for their known regulatory network. It extracts strongly connected subgraphs of the known genetic graphs of the aforesaid organisms. The number of nodes of the subgraphs may be specified by the user. It visualizes the original network and which can be used for future reference. It provides the facility for knock-out tests (removing a node or gene from the network to see change in behavior of other nodes) and knockdown tests (lowering the expression level for a specific gene of interest and to note the corresponding change of other gene expressions). It facilitates some of the reconstruction algorithm including ARACNE [15] for simulation. It also provides the comparison platform for original to simulated network and publishes the result as pdf.

3.2 Artificial Neural Network(ANN)

Here each gene is considered as an artificial neuron [12], [18], [19], [20] which is activated and send interaction to another gene by the additive effects of interactions of other genes. Here the genes of a certain layer of ANN not only can send interaction to the layer next, rather it can send the interaction even to a gene of a previous layer. So Recurrent Neural Network is the proper model applicable in this context. The variation of the expression level represented as the dynamics [43] of the system is given as under:

$$T_i \frac{dx_i}{dt} = f\left(\sum_{j=1}^{N} w_{ij} \cdot x_j(t) - b_i\right) - k_i \cdot x_i(t) \quad (1)$$

$$\frac{dx_i}{dt} = \frac{lim_{\Delta t \to 0}\, x_i(t+\Delta t) - x_i(t)}{\Delta t} \quad (2)$$

Substituting (2) to (1) we get

$$x_i(t+\Delta t) = f\left(\sum_{j=1}^{N} w_{ij} \cdot x_j(t) - b_i\right) - k_i \cdot x_i(t)\left(1 - \frac{\Delta t}{T_i} \cdot k_i\right) \quad (3)$$

$x_i(t)$ is the expression level of gene i at time instance t,
f(.) is a nonlinear sigmoid function $f(x) = 1/(1+e^{-x})$,
k_i is the decay constant of i^{th} gene,
T_i is the time constant of gene i,
b_i is the basal expression of gene i,
w_{ij} is the sigmoid weight from regulator j to target i,
N is the maximum number of genes in the system.

In each time step according to mathematical model a regulator and target is probabilistically chosen, parameters are imposed (initially randomly), the expression level of the target is determined by expression (3) specified above. Error is estimated by comparing the original gene expression value to the estimated gene expression value. The objective of optimization is to reduce the error. The expression level of the target gene and the RNN parameters of current iteration are making target gene capable to determine regulatory interaction.

3.3 Fuzzy Logic

Fuzzy logic has been successfully implemented [48] for parameter estimation problem and gene expression level estimation problem related to the present context [50]. It enhances the probability of choosing a value in the domain to be true. In other word it improves the Probability Mass Function. Suppose a parameter value is to be chosen from a range -30 to +30. Let us take discrete values in the range at 0.1 interval. The discrete set is {-30, -29.9, -29.8, . . . -0.1, 0, 0.1, . . . 29.8, 29.9, 30}. There are almost600 elements in the set. If we arbitrarily choose a value, the probability of right choice is 1/600. If we fuzzify the set in [0, 1] taking values of the interval 0.1, i.e. the typical membership values will lie in the range {0, 0.1, 0.2, 0.3, …., 0.9, 1}, the probability of choosing a value to be right value is almost 1/11 because of the presence of 11 elements in the fuzzy set. This probability improvement is the main cause of fuzzification. The work implements the parameter estimation using fuzzy set where the crisp set is taken as {-30, -15, 0.001, 15, 30}. 0.001 is taken instead of 0 to make the defuzzification meaningful. Fuzzification is done dividing the crisp values by random numbers. Defuzzification is done using centroidal defuzzification using the formula written below:

$$\widehat{w_{ij}}(t) = \frac{\sum_{m=1}^{L} \overset{m}{w}_{ij} \times \mu\left(\overset{m}{w}_{ij}\right)(t)}{\sum_{m=1}^{L} \mu\left(\overset{m}{w}_{ij}\right)(t)}$$

${}^{m}w_{ij}$is the crisp value,

$\mu\left({}^{m}_{w_{ij}}\right)(t)$is the fuzzy values generated in tth iteration

$\widehat{w_{ij}}(t)$is the defuzzified value of tth iteration.

In each iteration a set of five fuzzified values are generated for each of all N(N+2) parameters. After defuzzification it is been used for the gene expression value estimation and thereby error calculation.

3.4 Optimization Techniques

In both parameter estimation and graph construction different meta-heuristic algorithms [50] like Genetic Algorithm (GA), Simulated Annealing (SA), Differential Evolution (DE) [48], Particle Swarm Optimization (PSO) [21], [22], Ant Colony Optimization (ACO) [23] are successfully implemented. ACO has shown better result in graph reconstruction problem [27] whereas PSO has proven it's excellence in parameter estimation problem [26]. The optimization algorithms may be obtained in different publications [26], [27], [48], [50] etc. The optimization problem may be modeled either for step by step graph construction problem learning the parameter in each iteration or it may be presented as parameter estimation problem where whole set of parameters are initialized arbitrarily and iteratively trying for better solution minimizing the error comparing the simulated value of equation (3) to the corresponding micro-array gene expression data. The cost functions of the two types of problem specified above are written as mean squared error in equation (4) and (5) respectively.

$$e_i = \frac{1}{T} \cdot \sum_{t=1}^{T} \left(x_i(t) - \bar{x_i}(t)\right)^2 \quad (4)$$

$$\mathrm{E} = \frac{1}{NT} \cdot \sum_{t=1}^{T} \sum_{i=1}^{N} \left(x_i(t) - \bar{x_i}(t)\right)^2 \quad (5)$$

e_i : The predicted error of the temporal expression pattern of gene i,

E: Error associated with the predicted time series,

$x_i(t)$ and $\overline{x}_i(t)$: The original and estimated expression level of gene i at time point t respectively,

T: The number of time-points in the time-series micro array data,

N: is the number of genes in the system.

4 REVIEW OF EXPERIMENTAL RESULTS

This section provides the details of the source of microarray gene expression data. The measurement metrics for comparison of the simulated network to the original known network are discussed. Here original network may be obtained from the software like GeneNetWeaver. Lastly the listing of the applications of soft computing tools in different reviewed problems is presented.

4.1 Data Source

The starting point of the study lies in the biological data available from Micro-array. Micro-array is a suitable technology [6], [7] from where the gene expression data for large number of genes are available. Microarray is a glass slide where single stranded DNA molecules are attached by biological methods like spectrometry in fixed locations of the slide. These fixed locations are called spots. Each spot generally represent a gene. There may be thousands of genes represented in a microarray. The microarray data is obtained from different public repositories [32], [33], [34], [35] in different data formats like .cel, .tab etc. The data is then represented in some popular tabular format like spreadsheet by using different software like R, bio-conductor, orange etc. The data may be of two types: time series data and steady state data. Both are represented as 2D arrays [49] where rows are represented by N Number of genes and columns are represented either by M time points (in case of time series where gene expressions are plotted in different time points) or by M conditions (in case of steady state where gene expression for different conditions like normal tissues, diseased tissues, tissues after therapeutic intervention are plotted). Apart from genes there may be one extra column present in the raw data which correspond to the class label. The expression level of i^{th} gene at j^{th} time point is situated at the intersection point of i^{th} row and j^{th} column. There are two sub-categories of time series data: short time series (containing 3 – 8 time points) and long time series (more than 8 time points). But two important difficulties are involved here are: firstly, many biological and experimental noises are involved in the data and secondly, compared to the number of genes the number of time steps in which their expression levels measured are few. This phenomenon is called the "curse of dimensionality" [27]. The data thus collected from gene chip or microarray by using image analysis tools are needed to be properly extracted to get actual expression levels of the genes.

4.2 Metric

The performance of simulation is estimated by comparing the edges obtained from co-expression matrix to the edges of the inverse modeled network. Comparing the original adjacency matrix to the simulated one an edge e_{ij} may be named as true positive (tp), false positive (fp), true negative (tn) or false negative (fn). The Sensitivity of the simulation is defined by the True Positive Rate which indicates the share of correctly inferred edges over total number of edges that should be inferred (tp and fn) edges [27], [49]. On the other hand Specificity of the simulation is defined by the False Positive Rate of the predicted network which indicates the share of falsely inferred edges over total number of edges that should not be inferred (tp and tn). Finally the Precision of the simulated network, also referred as the Positive Predictive Value, indicates the share of correctly inferred edges over all inferred edges. The Sensitivity, Specificity and the Precision are defined as:

$$Sensitivity = \frac{tp}{tp + fn}$$

$$Specificity = \frac{fp}{fp + tn}$$

$$Precision = \frac{tp}{tp + fp}$$

tp: true positive edge,
fp: false positive edge,
tn: true negative edge,
fn: false negative edge.

4.3 Performance Analysis

In this paper the relative performances of soft computing tools applied in different works are presented. Different soft computing tools presented here are Recurrent Neural Network (RNN), Fuzzy Logic, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Differential Evolution (DE).The soft computing tools are applied in two types of problems, Parameter-set Generation problem and Dynamic Graph Generation Problem. Performances of different reviewed works are presented with the paper reference number in the table.

TABLE 3
PERFORMANCE ANALYSIS

Sl. No.	Soft-computing tools	Parameter-set Generation problem	Dynamic Graph Generation problem	
			Graph construction	Parameter learning
1	RNN	√, [48,50]	√, [26,27,43]	
2	Fuzzy Logic	√, [48, 50]		
3	PSO	√, [50]	√, [26]	√, [26,27]
4	ACO		√, [27]	
5	DE	√, [48]		

√ specifies good result,
[i] Denote the paper number in reference.

5 CONCLUSION

In this article the effort is made to present a complete relevance and procedure of gene regulatory network and its reconstruction. A clear explanation of what GRN is, how does the formation of GRN take place as a result of certain cellular activities, what will be its actual structure, and why the reconstruction process of such networks are so important is given. It is seen that the infant of an organism has the same orientation of genes in its genome as in the adult of that organism. The only thing differs in these two cases, is the way of expression of different genes and the resulting regulatory network of genes that causes the variation in their shapes, activities and other characteristics. The applications in detecting biomarker of a disease and finding drug for the disease have extensive importance in bio-informatics. But at the time of simulating the network or estimating the parameters for gene expression profiling we have only time series or steady state gene expression data at our hand which is again prone to errors. The prior knowledge of the structure of the graph is also not available. This situation makes the network prediction job a tedious one. In this article focus is made to each issue of modeling and simulation of the network and certain measures are suggested to reduce the complexity. Different mathematical models required for the simulation of GRN like data extraction procedure, co-expression network generation, different forward and reverse probabilistic models are discussed. The optimization techniques to reduce the state space search during graph reconstruction and parameter estimation, different soft computing techniques to facilitate the reconstruction are stated. The microarray technology, the public data sources of Time Series and Steady State gene expression data, different metrics to measure the performance of the simulation are presented and the application area is explained here. The software named GeneNetWeaver, discussed here, is not only useful as a platform for producing time series gene expression data needed for experiments but also provides a platform for different perturbation experiments like Knock Out and Knock Down tests. In this paper, the focus has been made on the optimal structure of GRN, but in future there is an ample scope for research in a particular disease domain, such as, to detect the biomarker of the disease, to find pathway for the disease, to find drug based on the simulation of GRN.

REFERENCES

[1] Jeremy M. Berg, John L. Tymoczko and Lubert Stryer, "Biochemistry", 5th Edition, W. H. Freeman..
[2] David L. Nelson and Michael M. Cox, "Principles of Biochemistry", 4th edition, Lehninger.
[3] H Lodish, A Berk, P Matsudaira, C. A. Kaiser, M Krieger, M. P. Scott, L Zipursky and J Darnell, "Molecular Cell Biology", 5th edition, W. H. Freeman.
[4] G.H. Bell, J. N. Davidson and H. Scarbrough, "Text Book of Physiology and Biochemistry", 6th edition, Livingstone.
[5] Richard Dawkins, "The Selfish Gene", 2nd edition, Oxford University Press.
[6] Jean-Michel Claverie and Cedric Notredame, "Bioinformatics for Dummies", 2nd edition, Wiley Publishing.
[7] Dov Stekel, "Microarray Bioinformatics", 1st edition, Cambridge University press.
[8] "DNA and RNA Introduction", Chemistry dept., Elmhurst College, . http://www.elmhurst.edu/~chm/vchembook/580DNA.htm
[9] "Genome Analysis", SCFBio, IIT Delhi, http://www.scfbio-iitd.res.in/research/genomeanalysis.htm.
[10] "Introduction to DNA Structure", Biology Learning Center at the University of Arizona , http://www.blc.arizona.edu/molecular_graphics/dna_structure/dna_tutorial.html.

[11] Hiroaki Kitano, "Systems Biology: A Brief Overview", Science, vol. 295, no. 5560, pp. 1662-1664, Mar. 2002.

[12] S. S. Ray, S. Bandyopadhyay, P. Mitra and S. K. Pal, "Bioinformatics in neurocomputing framework", IET Circuits, Devices and Systems. vol. 152, no. 5, pp. 556-564, Oct. 2005.

[13] John Grefenstette, Hadon Nash and Douglas Blair, "Comparing Algorithms for Large-Scale Sequence Analysis", Proc. 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering, pp. 89-96, Nov. 2001.

[14] Hidde De Jong, "Modeling and Simulation of Genetic Regulatory Systems: A Literature Review", Journal of Computational Biology. vol. 9, no. 1, pp. 69–105, 2002.

[15] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera and Andrea Califano, "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context ", BMC Bioinformatics. vol. 7.1, no. 7, 2006.

[16] Katia Basso, Adam Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera and Andrea Califano, "Reverse engineering of regulatory networks in human B cells", Nature Genetics, vol. 37, no. 4, pp. 382 – 390, 2005.

[17] A.K. Jain, J. Mao and K. M. Mohiuddin), "Artificial neural networks: a tutorial", The Computer Journal, vol. 29, no. 3, pp. 31-44,1996.

[18] Elaine Rich, Kevin Knight and Shivashankar B Nair, "Artificial Intelligence", 3rd Edition, Tata McGraw-hill.

[19] Jiri Vohradsky, "Neural Model of the Genetic Network", The Journal of Biological Chemistry, vol. 276, no. 39, pp. 36168 – 36173, 2001.

[20] Mattias Wahde and John Hertz, "Modeling Genetic Regulatory Dynamics in Neural Development", Journal of Computational Biology, vol. 8, no. 4, pp. 429 – 442, 2001.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.

[22] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization", EP '98 Proceedings of the 7th International Conference on Evolutionary Programming VII. Springer, pp. 591–600, 1998.

[23] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 26, no. 1, pp. 29–41, 1996.

[24] N. Noman and H. Iba, "Reverse engineering genetic networks using evolutionary computation", Genome Informatics, vol. 16, no. 2, 205–214, 2005.

[25] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics", PubMed Central (PMC), vol. 99, no. 16, pp. 10,555–10,560, 2002.

[26] Rui Xu, Wunsch D.C and Frank R.L, "Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization", Computational Biology and Bioinformatics. vol. 4, no. 4, pp. 681- 692, 2007.

[27] Kentzoglanakis K and Poole M, "A Swarm Intelligence Framework for Reconstructing Gene Networks: Searching for Biologically Plausible Architectures", IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 9, no. 2, pp. 358 – 371, 2012.

[28] Thomas Schaffter and Daniel Marbac, "GNW User Guide GeneNetWeaver", http://sourceforge.net, 2009.

[29] Thomas Schaffter, Daniel Marbach and Dario Floreano, "GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods", Bioinformatics, vol. 27, no. 16, pp. 2263–2270, 2011.

[30] Yufei Huang, Isabel M. Tienda-Luna, and Yufeng Wang, "Reverse Engineering Gene Regulatory Networks", IEEE Signal Processing Magazine, vol. 26, no. 1, pp. 76-97, 2009.

[31] Walid E. Gomaa, "Modeling Gene Regulatory Networks: A Survey", AICCSA IEEE/ACS International Conference, pp. 204-208, 2011.

[32] www.biolab.si/supp/bi-cancer/projections/info/DLBCL.html.

[33] www.biolab.si/supp/bi-cancer/projections/info/GSE349_350.html.

[34] www.biolab.si/supp/bi-cancer/projections/info/GSE412.html.

[35] www.datam.i2r.a-star.edu.sg/datasets/krbd.

[36] Jung Kyoon Choi, Ungsik Yu, Ook Joon Yoo and Sangsoo Kim, "Differential coexpression analysis using microarray data", Bioinformatics (Oxford Journals),vol. 21, no. 24,pp. 4348-4355, 2005.

[37] Judea Pearl and Stuart Russell, "Bayesian Networks", http://www.cs.berkeley.edu/~russel/papers/hbtnn-bn.pdf.

[38] Gregory F. Cooper and Edward Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data", Machine Learning, vol. 9, no. 4, pp. 309-347, 1992.

[39] David Heckerman, Dan Geiger and David M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data", Machine Learning, vol. 20, no. 3, pp. 197-243, 1995.

[40] Jing Yu, V. Anne Smith, Paul P. Wang, AlexanderJ Hartemink and Erich D. Jarvis, "Advances to Bayesian network inference for generating causal networks from observational biological data", Bioinformatics (Oxford Journals), vol. 20, no. 18,pp. 3594–3603, 2004.

[41] Changhe Yuan, Brandon Malone and Xiaojian Wu, "Learning Optimal Bayesian Networks Using A* Search", Proc. 22nd International Jt. Conference on Artificial Intelligence, 2011.

[42] B. Bollobas, C. Borgs, J. Chayes, and O. Riordan, "Directed scale- free graphs," in SODA'03: Proceedings of the 14th annual ACM-SIAM Symposium On Discrete Algorithms, pp. 132–139, , 2003.

[43] M. Wahde and J. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks", Biosystems, vol. 55 no. 1-3, pp. 129– 136, 2000.

[44] Haiying Wang, Francisco Azuaje, Olivier Bodenreider, Joaquín Dopazo, "Gene Expression Correlation and Gene Ontology-Based Similarity: An Assessment of Quantitative Relationships", IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB-2004), pp. 25-31, 2004.

[45] Carsten O Daub, Ralf Steuer, Joachim Selbig, Sebastian Kloska. "Estimating mutual information using B-spline functions – an improved similarity measure for analyzing gene expression data", BMC Bioinformatics, 5:118, 2004.

[46] Jyotsna Kasturi, Raj Zachary and Murali Ramanathan, "An information theoretic approach for analyzing temporal patterns of gene expression", Bioinformatics, vol. 19, no. 4, pp. 449–458, 2003.

[47] Peter J. Woolf and Yixin Wangi Physiol, "A fuzzy logic approach to analyzing gene expression data", Physiological Genomics, vol. 3, no. 1,pp. 9–15, 2000.

[48] D. Datta, S.S. Choudhuri, A. Konar, A.K Nagar and S. Das, "A Recurrent Fuzzy Neural Model of a Gene Regulatory Network for Knowledge Extraction Using Differential Evolution", In Proceedings of IEEE Congress on Evolutionary Computation, pp. 18-21, 2009.

[49] Monalisa Mandal and Anirban Mukhopadhyay, "A Multiobjective PSO-based Approach for Identifying Non-redundant Gene Markers from Microarray Gene Expression Data", IEEE International Confer-

ence of Computing, Communication and Application (ICCCA), pp. 1-6, 2012.

[50] Bijaya Ketan Panigrahi, Yuhui Shi, and Meng-Hiot, "Handbook of Swarm Intelligence: Concepts, Principles and Applications, Adaptation, Learning, and Optimization, vol. 8, 2011, Springer.