Chapter 7

Reconstructing the SOS System

This chapter presents the results of the experiments in which the developed system was evaluated by applying it to analyze some real microarray data. A well defined and widely studied network, the SOS DNA repair system of *Escherichia coli*, was attempted to reverse engineer from the expression profiles for demonstrating the capability of the developed methodology. First, a brief introduction to the SOS network is presented then the data that was used for reconstruction is evaluated and finally the reconstructed network is presented with discussion.

Simulation has become an effective and appropriate support for system design and verification. In contrast to analytical approach, the simulation approach gives more flexibility and convenience. Generally, simulation is used for different purposes such as convenience, cost effectiveness, safety, security and many more. However, in some cases, when there is no way to the absolute correct answers, simulation becomes the most powerful means of verification. For the particular case in hand, i.e. the gene regulatory network reconstruction problem, we don't know the correct regulatory structure existing among the genes in an organism, neither we know the type or strength of regulatory interactions. Therefore, simulation helps us most in this respect for verifying whether the developed system can identify the correct regulatory architecture and estimate the regulatory parameters of the system under consideration.

Still we need verification of the method or model applying in real scenario after we have gained enough support from simulated results. Because, simulation is just an imitation of the real situation and can not replace the real one. Therefore, finally the developed system was employed in the reconstruction task of a real biological network for verification purpose.

7.1 The SOS System

The SOS network in *Escherichia coli* was the first damage-inducible network to be identified. This is one of the most complex, best understood and widely studied gene regulatory system that is induced by DNA damage. The existence of the SOS system was discovered from studies on the effect of UV irradiation on *E. coli* and consideration of seemingly unconnected data.

Exposure of *E. coli* to agents or conditions that damage DNA or interfere with DNA replication, results in the induction of a diverse set of physiological responses termed as SOS responses [136]. These responses are due to the induction of more than 30 genes or proteins or operons which have often been referred to as *din* (damage-inducible) genes[33].

7.2 Model for SOS Regulation

The expression of the genes in the SOS regulatory network is controlled by a complex circuitry involving the RecA and LexA proteins [135]. A schematic diagram of the basic regulatory mechanism in the SOS system is presented in Fig. 7.1. In an uninduced cell, the product of the *lexA* gene acts as the repressor of more than 20 genes, including the *recA* and *lexA* genes, by binding to similar operator sequences upstream of each gene or operon. Many of these SOS genes are expressed at significant levels even in the repressed state. The amount of RecA protein present in an uninduced cell, 7,200 molecules per cell, is evidently enough to satisfy the requirement for this protein in homologous recombination.

In response to an SOS-inducing treatment or condition, a signal that leads to the expression of the SOS regulon is generated. A considerable body of evidence suggests that this signal consists of regions of single-stranded DNA. These may be generated when a cell attempts to replicate damaged DNA or under a variety of other circumstances. The binding of RecA to these regions of single-stranded DNA in the presence of a nucleoside triphosphate forms a nucleoprotein filament and converts RecA to an activated form (often referred to as RecA*). The interaction of activated RecA protein with the LexA protein results in the proteolytic cleavage of LexA: apparently, the activated RecA facilitates an otherwise latent capacity of LexA to autodigest. Activated RecA is also capable of mediating the cleavage of the repressors of the bacteriophages such as λ, P22, 434, and ϕ80, as well as the UmuD protein and its homologs. The cleavage of LexA occurs at a particular Ala-

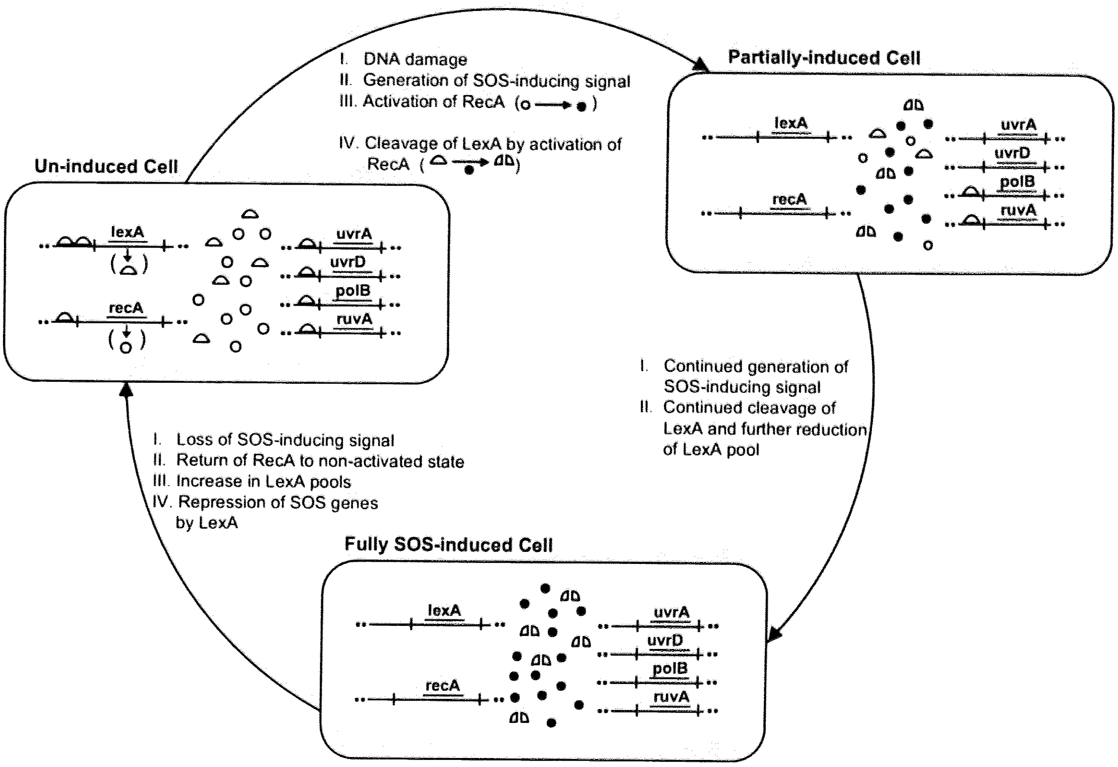

I. DNA damage
II. Generation of SOS-inducing signal
III. Activation of RecA
IV. Cleavage of LexA by activation of RecA
Partially-induced Cell
Un-induced Cell
lexA
recA
uvrA
uvrD
polB
ruvA
I. Continued generation of SOS-inducing signal
II. Continued cleavage of LexA and further reduction of LexA pool
I. Loss of SOS-inducing signal
II. Return of RecA to non-activated state
III. Increase in LexA pools
IV. Repression of SOS genes by LexA
Fully SOS-induced Cell

Figure 7.1: Model of SOS regulatory system (adapted from Ref. 41).

Gly-peptide bond near the middle of the protein and generates two polypeptide fragments. As the pools of intact LexA begin to decrease, various SOS genes, including the *recA* gene, are expressed at an increased level. Subsequently, the SOS responses mediated by the products of these genes begin to be observed. Genes with operators that bind LexA relatively weakly are the first to be expressed fully. If the inducing treatment is sufficiently strong, more molecules of RecA are activated, resulting in cleavage of more molecules of LexA. As the pools of LexA decline to very low levels, even genes whose operators bind LexA very tightly are expressed at maximal levels.

As the cell begins to recover from the inducing treatment, e.g., by DNA repair, the inducing signal is eliminated, and the RecA molecules return to their proteolytically inactive state. In the absence of the RecA protease, the continued synthesis of LexA molecules now leads to an increase in the LexA pools. This in turn leads to repression of the SOS genes and a return to the un-induced state [135, 98, 105].

Table 7.1: Some of the SOS genes in *E. coli* (adapted from [49])

Gene	Function
Expressed as the first	
lexA	Repressor of SOS genes
uvrA	UvrABC-excinuclease (NER repair)
uvrB	UvrABC-excinuclease (NER repair)
uvrD	Helicase II
polB	DNA polymerase II
ruvA	RuvAB-helicase
ruvB	Recombinational repair
dinI	Inhibitor of UmuD processing
Expressed as the second	
recA	SOS derepressor, recombinational repair
recN	RecN, recombinational repair
Expressed as the last	
sfiA	*sulA* cell division inhibitor
umuD	UmuD' (unit Pol V)
umuC	UmuC (Pol V)

7.3 Genes of SOS System

The functioning of the SOS system seems to be simple and, in general, operates as proposed by Little & Mount [67]. LexA repressor binds to SOS boxes with sequences 20-nucleotides long situated near or inside the promoter site of the SOS-induced genes, and its binding prevents accessibility to RNA polymerase.

Some of the SOS genes and the timing of their expression are shown in Table 7.1. In the first phase, among the expressed genes are *lexA*, encoding SOS repressor protein, genes *uvrA, uvrB, uvrD* whose products (together with *uvrC*-encoded protein) are involved in single strand nucleotide excision repair (NER), and *ruvA* and *ruvB* genes whose products are involved in recombinational DNA repair. One of the first expressed genes is also the *dinI* gene coding for an inhibitor of UmuD → UmuD' processing, and *polB* (*dinA*) encoding DNA polymerase II, enabling resumption of DNA synthesis when replication is stalled. In the next phase are expressed *recA* and *recN* genes, whose protein products are involved in DNA recombinational error-free repair. RecA, therefore, is involved in induction of the SOS response (via RecA*), in DNA recombination, single and double strand DNA repair, and recombination dependent replication [49]. RecN is involved in RecF-dependent recombination and double strand repair.

Knowledge of the role of RecA protein in recombination much preceded our knowledge of its role in de-repression of the SOS system and in SOS mutagenesis. Among the latest expressed genes are *sulA* (*sfiA*) encoding a cell division inhibitor and causing filamentous cell growth, *umuD* and *umuC* genes encoding error-prone DNA polymerase V, and genes *cea* and *caa* of colicinogenic plasmids coding for colicin E1 and colicin A, respectively. *cea* and *caa* can be regarded as apoptosis genes because their induction causes lysis and death of the cells. Only a few genes has more then one SOS box; namely *lexA*, (two sites) and *recN*, (three sites) included in Table 7.1 and (not indicated in Table 7.1) *ydjM* (two sites). These genes are regarded as being more tightly regulated [49].

7.4 Experimental Data set

The experimental data was downloaded from the homepage of Uri Alon Lab [43]. Ronen *et al.* [105] developed a system for real-time monitoring of the transcriptional activity of operons by means of low-copy reporter plasmids in which a promoter controls green fluorescent protein (GFP). In each plasmid a different promoter controls the transcription rate of the same reporter gene, gfp, and thus rate of transcript production from the promoter is proportional to the rate of GFP accumulation. By continuous measurements from living cells grown in a multiwell plate fluorimeter, high-resolution time courses of the promoter strength and cell density are obtained. With this method, temporal resolution of minutes can be achieved. This process complements, at higher accuracy, the genomic-scale perspective given by DNA microarrays. It was shown that this approach can be used to determine the order of genes in an assembly pathway [105].

Data are expression kinetics of 8 genes (*uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA* and *polB*) of the SOS DNA repair network. Measurements are done after irradiation of the DNA at the initial time with UV light. Four experiments are done for various light intensities (Exp. 1&2: 5 Jm^{-2}, Exp. 3&4: 20 Jm^{-2}). Each experiment composed of 50 instants evenly spaced by 6 minutes intervals and 8 genes are monitored. The expression levels for Exp. 1 & 2 are shown in Fig. 7.2 and 7.3.

The first hypothesis made by Alon *et al.* is that the GFP protein is very stable during the experiments: it is justified when comparing the typical GFP stability to the experiments length (300 minutes). By taking the derivative of the fluorescence amount with respect to time, Alon *et al.* have therefore access to the instantaneous protein production rate, since no protein degradation occurs during the experiments.

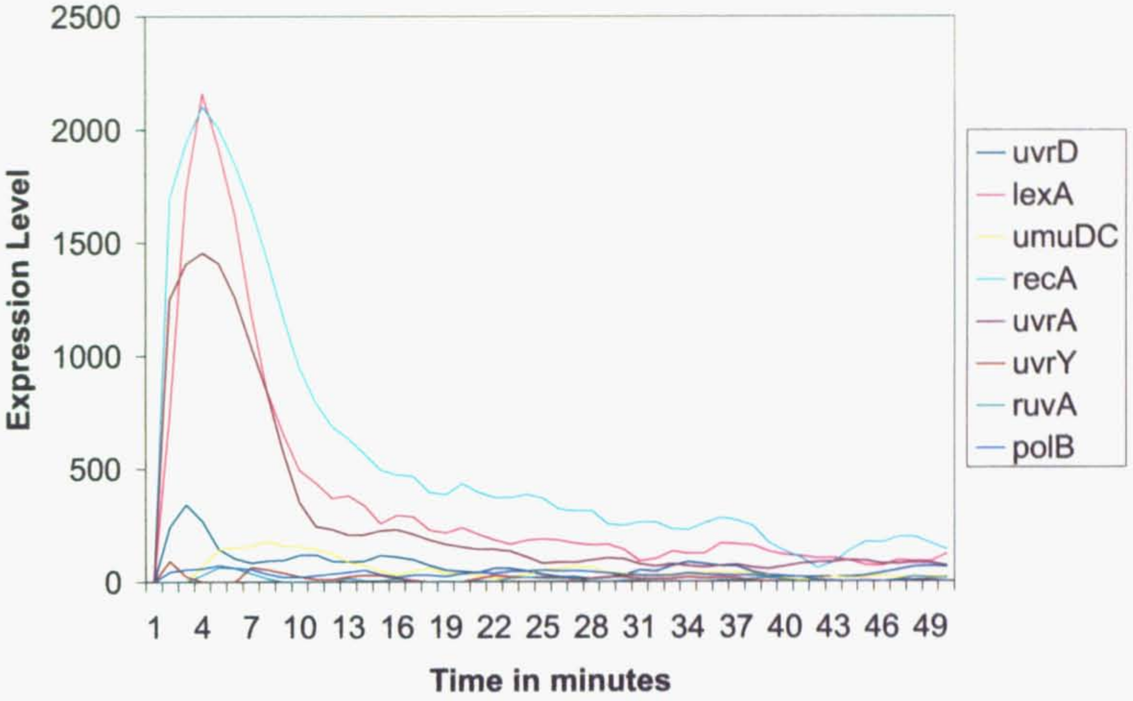


Figure 7.2: Expression profile of 8 SOS genes in experiment 1 (obtained from [43])

To use these data in the proposed model, the hypothesis made by Perrin *et al.* [98] about the stability of the mRNAs were adopted in this work. The hypothesis considers that mRNA molecules are degraded immediately after their production. Actually mRNA persistence depends on the nucleotide sequence; all mRNAs having the same sequence here because of the experimental technique, they all have the same persistence. It is hence sufficient to make the hypothesis that the turnover of the GFP mRNAs is very fast to ensure that there is a high unstability of the mRNAs. Therefore it is considered that the instantaneous promoter activity of each gene is also proportional to the present quantity of corresponding mRNA [98].

Using this hypothesis it is possible to assume that the data provided by Alon *et al.* directly indicate the observed mRNA quantities (also called expression levels) corresponding to each S.O.S. gene. In the experiment all the data from Alon's experiments were used, i.e. $50 \times 4 = 200$ sampling points were used for each gene. The data were normalized with in the range $(0, 1]$ and all the zero expression levels were replaced with a very small value. 5 runs were carried out to assure the statistical significance of the probabilistic search.

In this reconstruction experiments, the reverse engineering algorithm was used with TDE optimizer proposed by Fan and Lampinen [29] and the fitness function of Eq. (3.17) was used for evaluating the candidate networks.

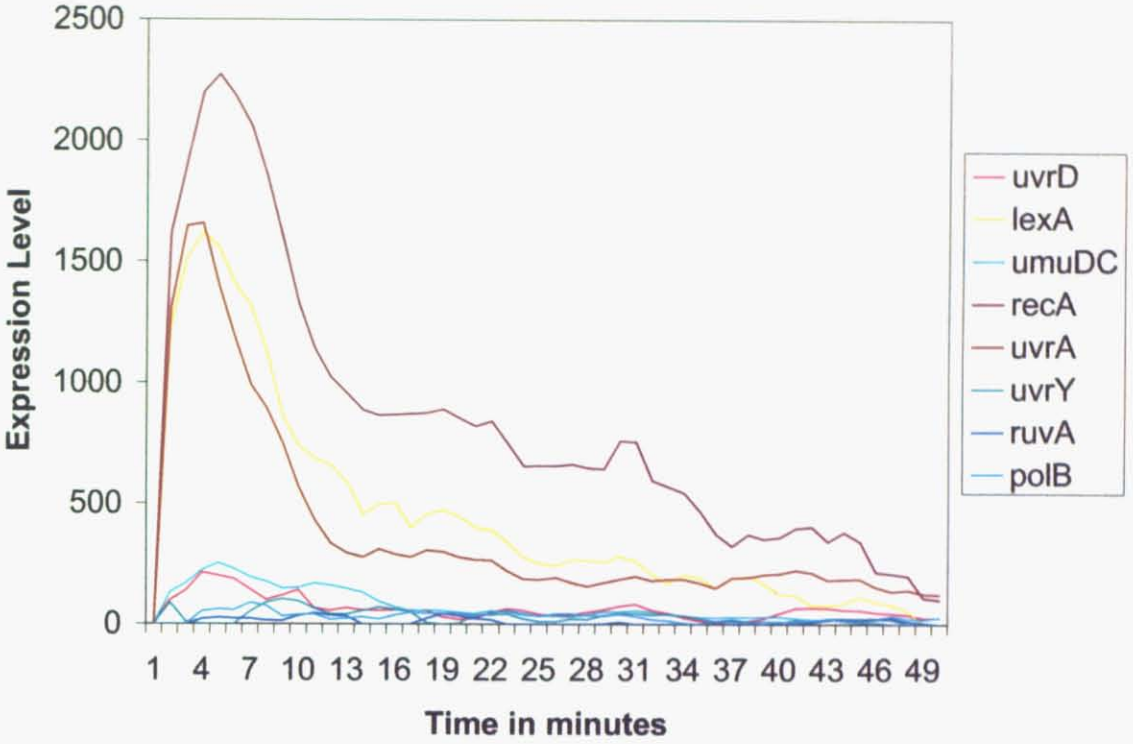


Figure 7.3: Expression profile of 8 SOS genes in experiment 2 (obtained from [43])

7.5 Reconstructed SOS system

As the gene expression data contained significant level of noise, the obtained results were very dispersed. Therefore, Z-score was applied to analyze which regulations are more significant and less diverse than others. For each parameter p in the model the mean magnitude μ_p and the standard deviation σ_p were calculated from the estimated values for that parameter at different experimental runs. Then the Z-score value was calculated as $Z_p = \mu_p/\sigma_p$ that can be used as a signal to noise measurement to imply robust parameters [6]. The analysis of Z-score is more qualitative but sufficient for comparing with the existing knowledge or suggesting new regulation [25]. The regulations which had Z-score value above the threshold $Z_{th} = 1.5$ were considered only, and the threshold level was set empirically.

Fig. 7.4 shows the structure of the SOS DNA repair network reconstructed by the proposed method. The true/known regulations are drawn using solid arcs labeled with T, the unknown regulations or possible false predictions are drawn using dashed arcs and the regulation drawn in solid arc with label P represents unknown regulation and predicted by others. Looking at the inferred structure it can be found that the *lexA* regulation of *lexA, umuD, recA, uvrA* and *uvrD* has been correctly identified. The activation of *lexA* by *recA* was also correctly

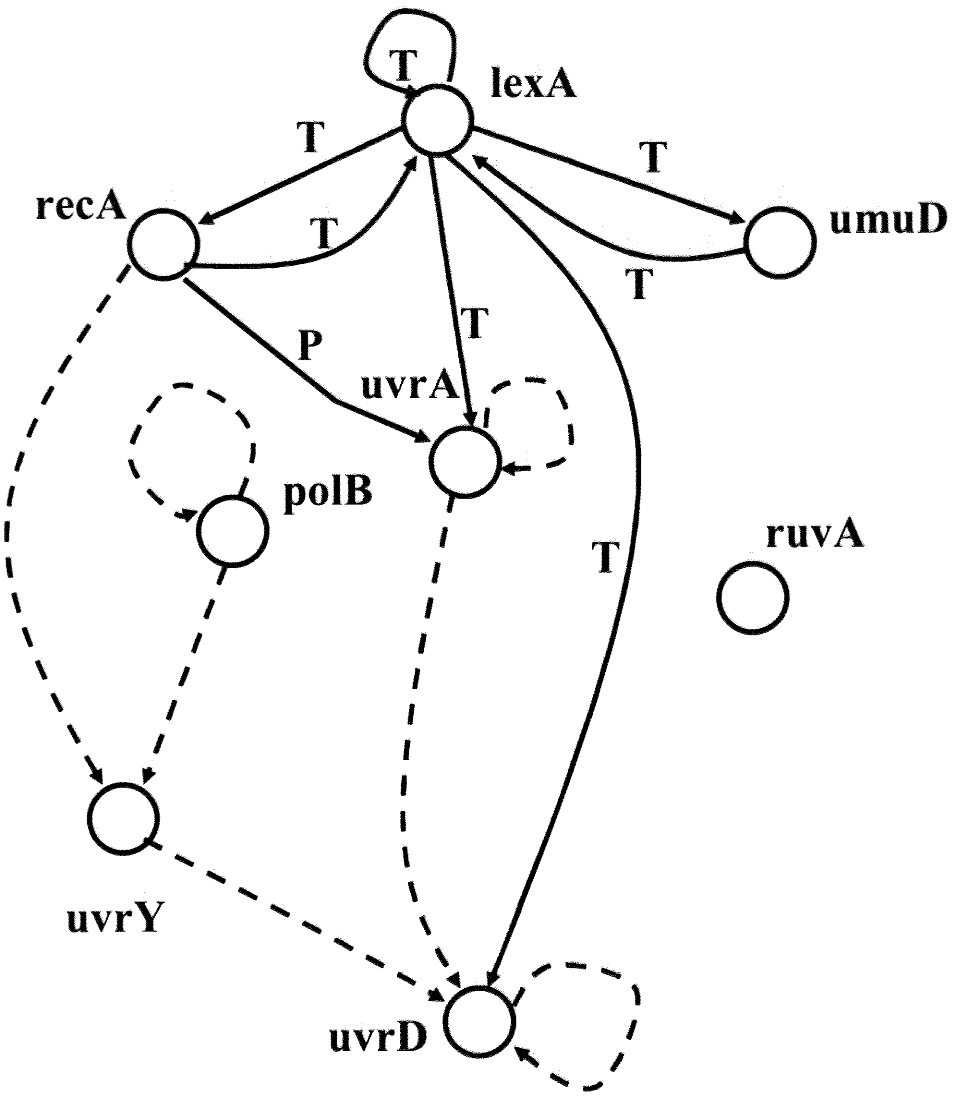


Figure 7.4: Estimated SOS network structure.

predicted. Regulation of *lexA* by *umuDC* is also known. An unknown regulation of *uvrA* by *recA* was identified. Perrin *et al.* [98] also identified this regulation in their experiment and hypothesized that this could correspond to the indirect regulation $recA \rightarrow RecA \dashv LexA \dashv uvrA$. Some other regulations were inferred, those are either novel regulatory pathways or false-positive findings. Moreover, in the microarray experiment the UV light shock was not sufficient to lead to the functioning of all SOS genes. From the expression profiles of Fig. 7.2 and 7.3 it can be seen that several genes were not induced sufficiently during the experiment. These genes are activated only when the damage is sufficiently high and their activation would have been useful to identify interrelation among the genes. Examining the

Table 7.2: Inferred SOS network by Perrin *et al.*. The j-th column shows all identified regulations exerced by j-th gene on other genes. [98]

	uvrD	*lexA*	*umuD*	*recA*	*uvrA*	*uvrY*	*ruvA*	*polB*
uvrD	0	0	0	0	0	0	0	0
lexA	0	–	+	+	–	0	0	0
umuD	0	0	0	0	0	0	0	0
recA	0	–	0	0	–	0	0	0
uvrA	0	–	0	X	X	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

expression profiles no significant change in the expression level of gene *ruvA* was observed. It was hypothesized that lack of significant change in the expression level is the reason why the proposed algorithm could not find any relation of gene *ruvA* with other genes in the network.

Perrin *et al.* [98] used the same data set to extract the regulations among the genes involved in the network. In their work, they used dynamic Bayesian network for modeling the gene regulation mechanism and used a statistical machine learning approach for the identification of the underlying network. The inferred network by their method is shown in Table 7.2.

When compared with the predicted network of Perrin *et al.*, the proposed method predicted almost all the regulations predicted by them. However, the regulations $uvrA \rightarrow lexA$ and $uvrA \rightarrow recA$ were not identified by the proposed method but the method of Perrrin *et al.* inferred them. On the other hand, the proposed method could identify some correct regulations which were not predicted by Perrin *et al.*: e.g., $lexA \rightarrow umuD$, $lexA \rightarrow uvrD$. Nevertheless, the overall result suggests that the algorithmic framework was useful to extract some correct regulation by analyzing the real gene expression data.

Chapter 8

Identifying the Regulators of Yeast Cell Cycle

In this chapter an attempt was made for connecting the cell cycle genes of budding yeast (*Saccharomyces cerevisiae*) in terms of their regulatory relationships. The molecular machinery of cell cycle control is known in more detail for budding yeast than for any other eukaryotic organism. Because of the supreme importance of cell cycle mechanism, a large number of genes and proteins are involved in maintaining the process under a constant state of supervision. After presenting the introduction to the budding yeast cell cycle system, the chapter presents the expression data that was used for inferring the regulators in the gene circuit. Finally, the results of the reconstruction experiments are presented, analyzed and compared with other works.

8.1 The Budding Yeast Cell Cycle

The cell cycle represents a fundamental driving force for differentiation and evolution in eukaryotes. The cell cycle consists of the series of events by which a cell grows and divides into two daughter cells each of which duplicates all the necessary mechanism for repeating the process from the mother cell. By the process of cell division all the essential component for the process is transferred to the daughter cells.

The yeast mitotic cell cycle is very similar to the cell cycle of other eukaryotic cells and is commonly broken down into the four standard phases G_1 (*gap 1*), S (*DNA synthesis*), G_2 (*gap 2*), and M (*mitosis*).

When the daughter cell is segregated from the mother cell, it is typically smaller than the mother cell. During G_1 the daughter cell will grow to reach a minimal size before a new cell cycle can being. The G_1 phase of the cell cycle is important for determining the fate of the cell. Depending on environmental conditions, the cell may go into stationary phase (sometimes designated G_0), a stage of no growth, and then re-enter the cell cycle when the environmental conditions become more favorable. The duplication of centrosome begins late in G_1 and is completed during S phase. The duplication poles then slowly begin to migrate to positions on opposite sides of the nucleus. Meantime, within the nucleus during the S phase, DNA replication takes place. The initiation of DNA replication is closely correlated with bud emergence, however, these two events are not dependent on each other. Completion of the DNA replication marks the beginning of G_2 phase. This phase defines the time after DNA replication but before mitosis. In *Saccharomyces cerevisiae*, the G_2 phase is relatively short. In M phase the cell undergoes mitosis, partitioning DNA between the mother cell and the daughter bud. One interesting difference between budding yeast and most other eukaryotic cells is that the mitosis takes place within the nucleus without breakdown of the nuclear envelope. After the DNA has been partitioned, the cell undergoes cytokinesis separating the mother cell from the daughter cell.

The different phases of the budding yeast cell cycle is depicted diagrammatically in Fig. 8.1. As the figure depicts, the bud first emerges shortly after entry into S phase and grows throughout the cell cycle and separated eventually. The status of a cell within the cell cycle can be approximately monitored visually by the size of the bud. The transitions from G_1 into S and from G_2 into M are called *checkpoints* because the transitions are delayed unless key processes have been completed.

8.2 The Transcriptional Program of Yeast Cell Cycle

The complete genomic sequence of *Saccharomyces cerevisiae* was published in 1996 which made the genome wide analysis of transcription patterns of all of the 6,183 open reading frames (ORFs) in a single microarray experiment. Such experiments by Spellman *et al.* [118] have shown that the transctiption levels of about 800 genes vary in a periodic or cyclic pattern through the cell cycle. The study considered the expression pattern of a gene over two cell cycles in synchronized cells. Spellman *et al.* [118] used DNA microarrays and samples from yeast cultures synchronized by three independent methods: a factor arrest, elutriation, and arrest of a *cdc15* temperature-sensitive mutant in their study. In a culture of synchronized cells, all cells are at the same stage in the cell cycle.

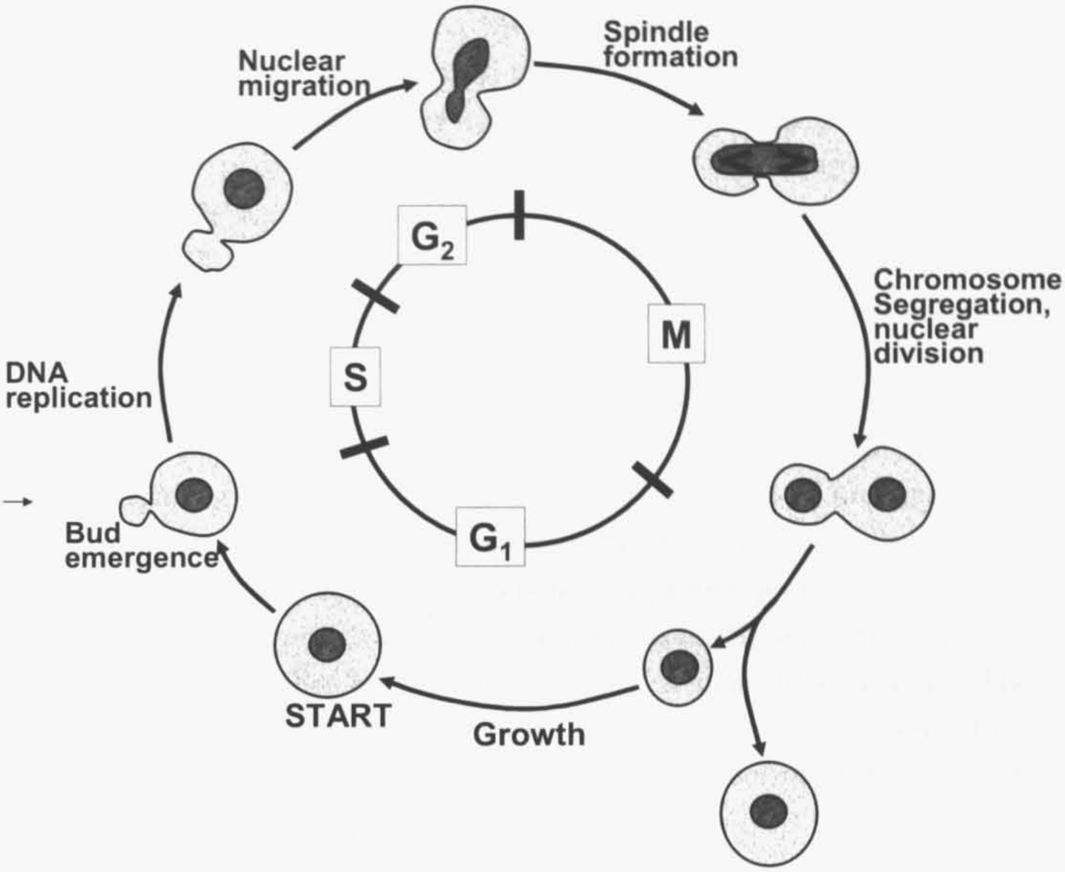


Figure 8.1: The cell cycle of budding yeast *Saccharomyces cerevisiae*.

Analysis of their experimental results showed that the transcription of each of the identified 800 genes is initiated once per cell cycle. The expression patterns found in their experiments indicated that some genes are transcribed in G_1, some in G_2, some in M and a few in S phase. Typically, genes of which products are needed at a specific part of the cell cycle are transcribed in the preceding period. For example, enzymes needed for synthesis of the trinucleotide precursors of DNA and for DNA replication are made in G_1 immediately prior to their use in S phase. Similarly, the histone proteins are synthesized during S phase immediately prior to their incorporation into chromatin and their use in chromosome condensation [34]. Further details about the study can be found in [118].

Cell-cycle studies over the past years, have mainly been concerned with identifying molecular machinery that are responsible for driving key chromosome cycle transitions. In early stages of the cell cycle, progression from one phase to the next is controlled by a family of protein kinase known as CDK (cyclin-dependent kinase). Cyclins are proteins involved in the regulation of the cell cycle whose concentration varies in a cyclical fashion during the cell cycle. In budding yeast there is only one CDK- Cdc28 that interacts with different cyclins at different times of the cell cycle [82]. In Phase G_1 Cdc28 associates with cyclins Cln1,Cln2,Cln3 for regulating growth and morphogenesis. In S phase the cyclin-CDK complex formed by Clb5, Clb6 and Cdc28 regulates the DNA replication and the complex formed between Clb2 and Cdc28 regulates mitosis in phase G_2/M [34]. And the activity of Cdc28 is usually terminated by cyclin degradation or inibotory phosphorylation. The activity of the Cdc28 rises and falls as the cell progresses through the cycle and the oscillations lead directly to cyclical change activation of certain proteins that initiate the major events of the cell cycle, for example an increase in Clb2/Cdc28 activity at the beginning of mitosis leads to increased activation of proteins that control chromosome condensation, nuclear envelope breakdown and spindle assembly.

The cell cycle control obviously depends on protein-protein interactions, which is also referred to as post-transcriptional mechanism. However, transcriptional regulation provides another level of control which is more fundamental. The genes peak in different phase during cell cycle are responsible for synthesis of cell cycle specific proteins. Some cyclin levels, for example, are controlled through cyclin gene transcription, since the genes mainly code for proteins.

8.3 Target Network

Though during the progression of the budding yeast cell cycle 800 genes were found to change their expression levels [118] the number of key regulators that are responsible for the control and regulation of this complex process is much smaller. Considering the capability of the proposed algorithm and the computational limitations, a part of the cell cycle pathway represented in the KEGG database [40] was considered for reconstruction in this experiment. The target network (shown in Fig. 8.2) consists of 14 well-described genes that play very important role in cell cycle regulation.

In the selected fragment of network, genes from different classes which are activated with their products at different stages of cell-cycle, were included. For example the network contains transcription factors ($SIW4$, $SWI6$, $MBP1$), cyclins ($CLN1$, $CLN2$, $CLN3$, $CLB5$, $CLB6$ which bind to the kinase $CDC28$) which are known to activate cell-cycle dependent genes. Among the others $SIC1$ and $CDC20$ act as inhibitors, degraders and competitors of the cyclin/$CDC28$. A brief description of the target network genes with their functions is presented in Table 8.1. The inter-

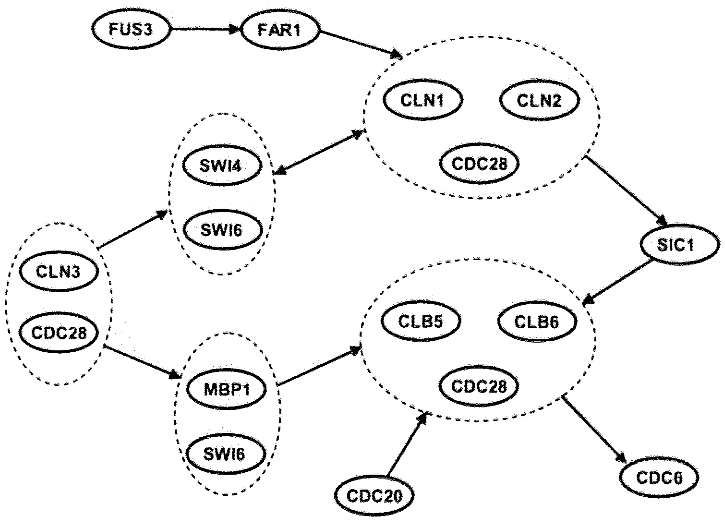


Figure 8.2: Target cell cycle network of *Saccharomyces cerevisiae* extracted from KEGG database [40]

Table 8.1: List of genes in *Saccharomyces cerevisiae* cell cycle network fragment considered for reconstruction

ORF	Gene Name	Description
YMR199W	*CLN1*	Cyclin, G_1/S specific
YPL256C	*CLN2*	Cyclin, G_1/S specific
YAL040C	*CLN3*	Cyclin, G_1/S specific
YPR120C	*CLB5*	Cyclin, B-type
YGR109C	*CLB6*	Cyclin, B-type
YBR160W	*CDC28*	Cyclin-dependent protein kinase
YER111C	*SWI4*	Transcription factor, subunit of SBF factor
YLR182W	*SWI6*	Transcription factor, subunit of SBF and MBF factor
YDL056W	*MBP1*	Transcription factor, subunit of MBF factor
YGL116W	*CDC20*	Cell division control protein
YLR079W	*SIC1*	Inhibitor of Cdc28p-Clb protein kinase complex
YJL194W	*CDC6*	ATP-binding protein required for DNA replication
YJL157C	*FAR1*	Cyclin-dependent kinase inhibitor
YBL016W	*FUS3*	Mitogen-activated protein kinase

actions among these genes are well defined and transcription activators are known. The same set of genes was used by Sugimoto and Iba for reconstructing the network by dynamic Bayesian network (DBN) and dynamic differential Bayesian network (DDBN) in [126].

8.4 The Gene Expression Data Set

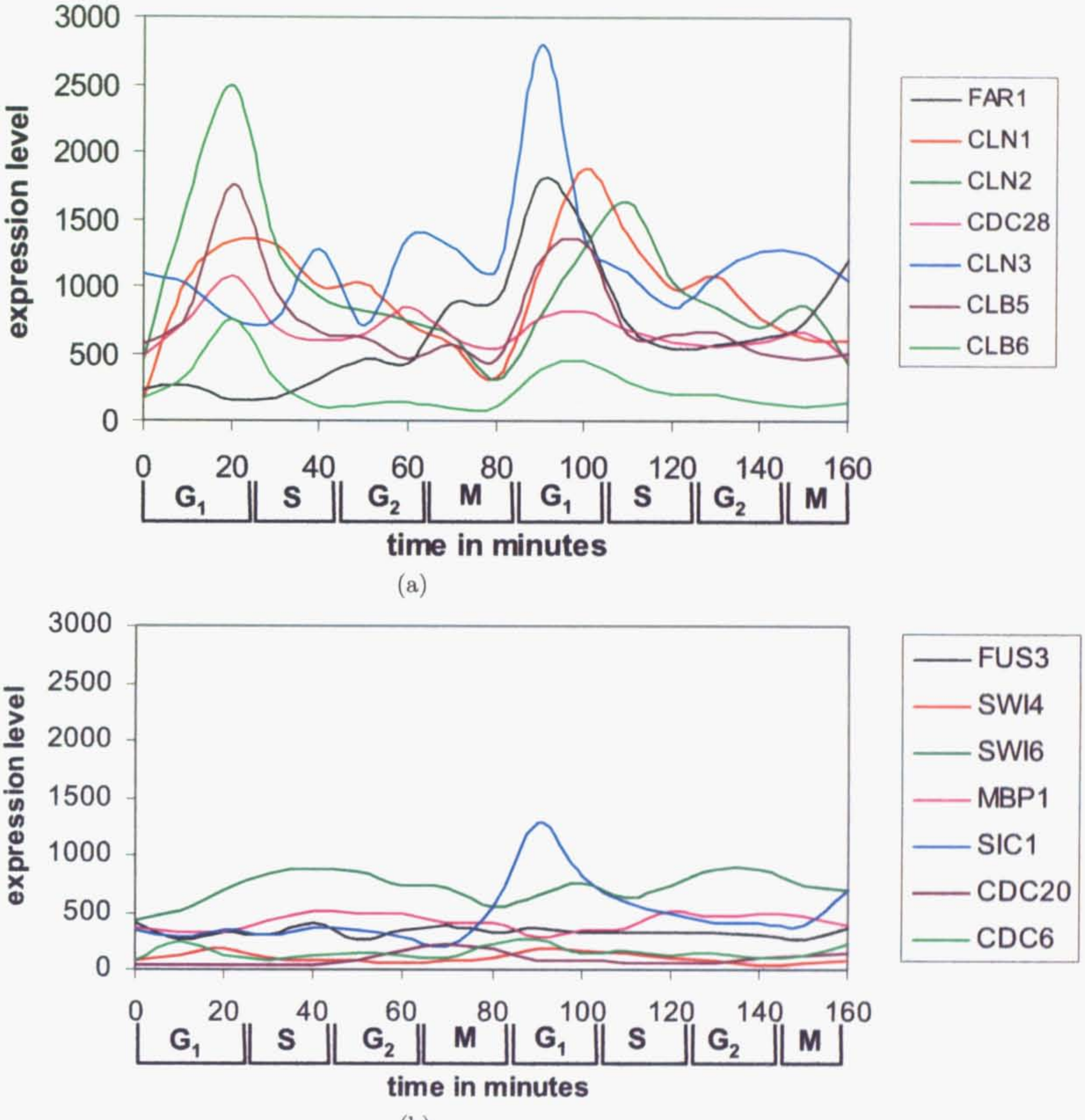
The microarray data from Cho *et al.* [18] obtained for *S. cerevisiae* cell cultures, synchronized by $CDC28$, were used in this study. In their study, commercially available high-density oligonucleotide arrays were used to quantitate mRNA transcript levels in synchronized yeast cells at regular intervals during the cell cycle. DNA oligonucleotide probes are directly synthesized on these arrays without individual manipulation or PCR amplification, minimizing the potential for cross-hybridization or clone error [18].

To obtain synchronous yeast culture, CDC28-13 cells were arrested in late G_1 at START by raising the temperature to $37°C$, and the cell cycle was reinitiated by shifting cells to $25°C$. Cells were collected at 17 time points taken at 10 min intervals, covering nearly two full cell cycles. Cells exhibited over 95% synchrony throughout the time course, as determined by bud size and nuclear position. The expression levels of the selected genes during these two cell cycles are shown in Fig. 8.3 separating into different phases of cell-cycle.

8.5 Experiment and Results

All the data(for each gene under consideration) from Cho's experiment were used for the reconstruction experiment, i.e. all the 17 sampling points for each gene were used. Again 10 repetition of inference were performed to assure the statistical significance. The reconstruction algorithm used the DEfirSPX optimizer developed in Chapter 4 and for evaluating the candidate networks the fitness function of Eq. (3.18) was used.

Probably because of insufficient amount of gene expression data, the high noise level present in expression data and above all the model flexibility the parameter values of the inferred networks were significantly different from trail to trial. In order to identify the core regulations, which contribute importantly to realize experimentally obtained time-course data, the results were analyzed using Z-score like that was done in previous chapter. To repeat, the analysis of Z-score is more qual-



(a)

(b)

Figure 8.3: Transcription levels of different genes during cell cycle of *Saccharomyces cerevisiae* from Cho *et al.* [18]

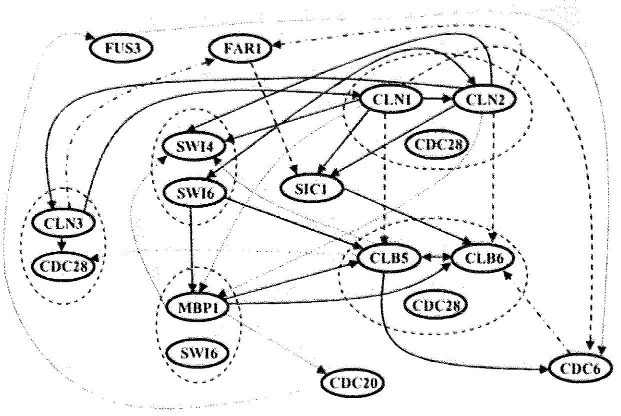


Figure 8.4: Reconstructed network of the yeast cell-cycle regulatory genes. Notation: solid arc ⇌ known regulation, dashed arc ⇌ indirect regulation, dash-dotted arc ⇌ inverse regulation and dotted arc ⇌ false-positive or novel regulation

itative but sufficient for comparing with the existing knowledge or suggesting new regulation [25]. For each parameter p in the model, the mean magnitude μ_p and the standard deviation σ_p were calculated from the estimated values for that parameter at different experimental runs. Then the Z-score value was calculated as $Z_p = \mu_p/\sigma_p$ that can be used as a signal to noise measurement to imply robust parameters [6].

Fig. 8.4 shows the reconstructed network structure where the interactions were inferred by the proposed algorithm more than 8 times within 10 runs and/or have a Z-score value above the threshold $Z_{th} = 1.5$, which was set empirically. Besides, the reconstructed method inferred self-degradation regulations for each gene except for $SWI4$, $MBP1$, and $CLB6$. These self-degradation regulations were not shown in the figure to keep it simple.

Inspecting the reconstructed network, it can be found that many of the regulations predicted by the proposed methodology conform with the target network of Fig. 8.2 and with the cell-cycle network given by Li *et al.* in [63]. The predicted self-degradation regulations were also mostly correct. However, the proposed method also identified some interactions which do not agree with currently available knowledge. The presence of some of these (possible) false positive interactions can be due to indirect regulations. For example, the regulation $CLN2 \rightarrow CLB6$ may be identified as a side-effect of indirect regulation $CLN2 \rightarrow SIC1 \rightarrow CLB6$. Such other regulations are $CLN1 \rightarrow CDC6$, $CLN1 \rightarrow CLB5$, $FAR1 \rightarrow SIC$. And a few interactions have been inferred inversely by the method e.g. $CLN2 \rightarrow FAR1$, $CLN3 \rightarrow FAR1$ and $CDC6 \rightarrow CLB6$. Most of the these indirectly and inversely predicted regulations, as well as many of the correct ones, had been predicted by

Sugimoto and Iba using DBN and DDBN in [126].

The proposed algorithm inferred 33 regulations (disregarding the self-degradatory regulations) among which 18 were predicted correctly (solid lines), 4 were indirect regulations (dashed lines), 3 were inversely predicted (dash-dotted lines) and 8 regulations were false positives or novel regulatory pathways (dotted lines). For the same target network Sugimoto and Iba could predict 19 and 18 regulations using DBN and DDBN respectively [126]. Using DBN their method predicted 7 correct, 1 indirect, 3 inverse and 8 false regulations and using DDBN 7 correct, 3 indirect, 5 inverse and 3 false regulations were inferred respectively. Soinov *et al.* [116] used a similar network of 20 genes which includes 11 genes (excludes FUS3, FAR1 and CDC6) from the network of Fig. 8.2 and additional 9 genes. Based on the highly accurate results from their classifiers, they identified 26 interactions among which 23 were correct. Among these 23 interactions 9 were questionable in their words because those failed in one or two of the accuracy tests. However, all the correct regulations (involving the genes of Fig. 8.2) inferred in their method were correctly identified in the proposed method. So the proposed method was successful not only to infer regulations those have been predicted by others, but also to identify some other correct interactions among the genes.

Examining all the extracted relations with respect to the known roles of selected genes, it was found that in most cases the predictions confirmed the prior knowledge which indicates the validity of the proposed approach. But the method was unsuccessful to identify a few correct regulations or predicted some wrong interactions in the cell-cycle network. Based on the experiments presented in previous chapters, some reasons behind such failure can be speculated. The primary reason was due to insufficient amount of gene expression data used for reconstruction, because the prediction power of the proposed method decreases with the decrease in supplied time-series data (Section 6.4). Secondarily, the level of noise present in the supplied data might had an adverse effect on the effectiveness of the method (Section 6.3). And finally, since only a fraction of the complete network was used, the absence of some genes may cause not only to lose some true interactions but also to emerge some false interactions.

Chapter 9

Discussion

With the increasingly available microarray data sets, identifying the regulatory machinery in a gene circuit is becoming more desirable compared to grouping of similar genes by clustering methods. Therefore, model-based reconstruction of gene regulatory network has become a popular research topic in the fields of computational biology and bioinformatics. Among different available models, the S-system has been found to provide valid representation in a large number of theoretical and practical studies. Moreover, the recent decoupling of the model has made its application computationally tractable in networks consisting of many metabolites. Nevertheless, the reverse engineering task has to cope with many technological, biological and algorithmic challenges. In this dissertation these challenges are addressed and some proposals are presented to solve them or at least to ameliorate the existing ways to deal with them.

9.1 Discussions on Memetic Optimizer

The problem space of gene network reconstruction using S-system model is highly multimodal, dynamic and deceptive. Because of the high complexity, any exhaustive or analytical method will prove unrealistic for finding the optimal solution for such a problem. Because of robust and reliable performance of EC in such problem domain, researchers used EC as a dependable alternative in recent years. However, none of the existing EAs proven it to be sufficient to find the global optimal solution for this problem. So, a substantial effort of this work has been expended to enhance the performance of an existing EA, namely DE, for multimodal problem optimization.

Neighborhood exploration has been found to be an effective way to improve the performance of an EA. Therefore, in an attempt to speed up DE, a memetic version of DE using crossover based local search (XLS) has been proposed. In the proposed XLS, named as FIR, the search space around the best individual is greedily explored in each generation.

As DE applies deterministic selection and lacks mutation operator, it tries to estimate the features of the search space iteratively based on the distribution of its individuals. There are EAs available in literature which exhibit very good performance without explicit mutation operation, e. g. MGG, G3. The reason behind their success is the use of XLS whose performance depends largely on the capability of the crossover operator used. Motivated by their methodology the FIR strategy was proposed for DE algorithm. Two implementations, DEfirDE and DEfirSPX, have been presented for the FIR where the first uses DE like recombination and the later uses simplex crossover (SPX) for searching the neighborhood of the best solution.

The ultimate target of designing this memetic version of DE is to increase the convergence speed of the algorithm in high dimensional optimization. Different experimental results, presented in Chapter 4, eminently shows that the FIR scheme increased the convergence velocity of the basic DE algorithm, especially in high dimensional search spaces.

Often the attempts towards a higher convergence rate simultaneously increase the risk that the algorithm will prematurely converge into a local optimum. Therefore, the robustness of the proposed DEFIR algorithms was analyzed by experimenting with various population sizes and different DE parameters. The results showed that the proposed memetic versions of DE can be used for higher convergence rate with increased search precision or search robustness.

Experimenting with random problems, yielded by a continuous landscape generator capable to differentiate between algorithms, the superiority of the proposed schemes were verified over their parent algorithm. It was found that in highly multimodal environment, the performance of DEfirDE and DEfirSPX were better than DE not only in terms of fitness achieved but also in terms of convergence speed. The suitability of the proposed schemes for optimizing the S-system model parameters was also studied. The results proved their superiority compared to basic DE in optimizing S-system model.

Between DEfirDE and DEfirSPX, the overall performance of the later was much better than that of the former. This is expected because SPX, a much sophisticated crossover operator that works well on functions having multimodality and/or epsitasis, will search the neighborhood of the best solution more effectively than the recombination process of DE. Therefore, the DEfirSPX scheme was more successful to accelerate the search-capability and to improve the performance of the algorithm.

Traditionally, hybrid EAs or MAs have been implemented by incorporating problem-dependent heuristics for refining the individuals (i.e. improving their fitness through fine tuning). However, the field of EA has always enjoyed the superior characteristic of being problem independent. Therefore, a recent interest is to include the local-search in EA in a problem independent manner. In an attempt to design a completely problem-independent crossover-based local search process, another scheme AHCXLS has been presented in this work. The AHCXLS scheme was designed by borrowing concepts from both LIPs and XLS, to take the advantages of both paradigms.

In different experimental results presented here, the proposed DEahcSPX outperformed the classic DE algorithm. The speedup of the algorithm has been also established by different results. The scalability study and the population size study highlighted the robustness of the proposed algorithm over original DE algorithm. Different experimental results and comparison with other MAs show that the performance of DEahcSPX is superior to many of the state-of-the-art EAs, particularly for multimodal problems, but it can also deal with the unimodal problems very competitively.

Generally, incorporation of a local search can not modify the overall behavior of an algorithm; however, can improve some of its characteristics. And more or less same phenomenon was observed in case of DEahcSPX. From different experimental results, and from the shape of the convergence graphs, it was found that for a particular class of problem the proposed memetic version of DE behaves similarly to its parent algorithm. However, in almost every case it exhibited a higher convergence velocity compared to DE.

The proposed AHCXLS was compared with other XLS applying in DE and it showed that the newly proposed local search scheme performs best. It can be hypothesized that the adaptive nature of the AHCXLS guides the algorithm to explore the neighborhood of each individual most effectively and locate the global optimum at a minimum cost. Furthermore, the scheme sets us free from the search for the best length for local search.

The principle of AHCXLS is so simple and generalized that it can be hybridized with any of the newly proposed DE variants without increasing the algorithm complexity. And in a brief study it was found that AHCXLS scheme can accelerate some other variants of basic DE algorithm proposed by Storn and Price. Experimental results also showed the prospect of the AHCXLS scheme in accelerating the self-adaptive variants of DE.

9.2 Discussions on Reverse Engineering Algorithm

After improving performance of DE, this enhanced optimizer was used for designing an evolutionary algorithm for inferring the transcriptional regulations in a biochemical network represented in the decoupled S-system form. Two major challenges in reconstructing GRNs modeled by S-system are (i) to detect the sparse topological architecture which is most commonly seen in biological networks and (ii) to estimate the kinetic parameters from limited amount of gene expression data corrupted with significant level of noise.

In order to cope with the first challenge, new fitness criteria were proposed for evaluating the candidate network models that are encountered during the evolutionary search. First, the MSE based fitness function was enhanced using a more effective penalty term. Grounded on the fact that not too many genes regulate both the synthesis and degradation process of other genes, an existing fitness function was modified/enhanced. And experimenting with this new MSE based fitness function it was found that it can reconstruct the network with fewer false positive predictions [88]. The proposed fitness function was also found more effective in estimating correct parameter values from fewer time series data.

Then another information criteria based fitness evaluation function was proposed for evaluating the network models. This fitness function was designed extending the AIC using the prior designed penalty term. However, a different penalty constant was used so that the complexity term can properly interact with the algorithm. An extensive study was performed on the performance of these two fitness functions varying the network size, noise-level and amount of gene expression data. In such studies both of the proposed fitness function showed brilliant performance in identifying the sparse network structure. Both of them also exhibited superior performance in estimating network architecture and kinetic parameter values compared to existing fitness functions. However, between these two fitness functions the overall performance of the newly proposed AIC based fitness criteria was better than the other.

The proposed memetic algorithm was designed targeting several issues such as identifying robust transcriptional regulations, estimating precise kinetic parameters, attaining skeletal network architecture and above all computational efficiency. The

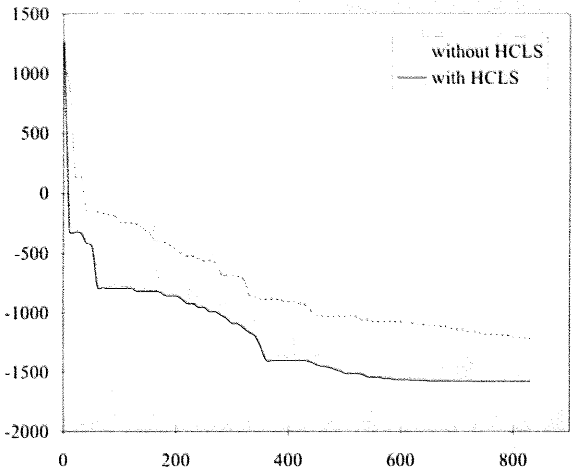


Figure 9.1: Performance comparison of the proposed algorithm with and without HCLS

optimization engine, the core of the algorithm, was implemented using the prior developed reliable and robust optimizer called DEfirSPX. However, the framework of the complete algorithm is so structured that any other optimizer can be used instead and TDE was used in some experiments. Besides, the exploitation capability of a hill climbing local-search heuristic was taken for efficient identification of the sparse network structure. Moreover, double optimization was used for detecting robust interactions and special measures were taken to maintain the population diversity for global convergence.

The suitability of the method is tested in gene circuit reconstruction experiments, varying the network dimension and/or characteristics, the amount of gene expression data used for inference and the noise level present in expression profiles. The reconstruction method inferred the network topology and the regulatory parameters with high accuracy. Nevertheless, the performance is limited to the amount of expression data used and the noise level present in the data. Furthermore, the proposed reconstruction algorithm was found more robust to noise and small sample size compared to other existing inference algorithms for gene networks. Finally, the methodology was verified by constructing the underlying networks of key regulators by analyzing different real gene expression data-sets.

Some empirical analysis of the proposed algorithm was performed to show that the different components of the algorithm were necessary for robust, efficient and accurate estimation of the regulatory parameters. All the results are based on the reconstruction experiments on NET1. As mentioned earlier, the purpose of the hill climbing local search (HCLS) was to identify the skeletal structure efficiently and

Table 9.1: Inferred parameters at different trials of Phase 1 and in Phase 2 for gene 3 of NET1 from 10% noisy data

	α_3	$g_{3,1}$	$g_{3,2}$	$g_{3,3}$	$g_{3,4}$	$g_{3,5}$	β_3	$h_{3,1}$	$h_{3,2}$	$h_{3,3}$	$h_{3,4}$	$h_{3,5}$
P1, T1	11.606	0.000	-0.755	0.000	0.000	0.000	12.930	-0.328	-0.509	2.248	0.000	0.000
P1, T2	11.605	0.000	-0.755	0.000	0.000	0.000	12.929	-0.328	-0.509	2.248	0.000	0.000
P1, T3	15.413	0.000	0.000	-0.464	0.000	0.000	20.405	-0.611	0.616	2.646	0.000	0.000
P1, T4	11.601	0.000	-0.755	0.000	0.000	0.000	12.925	-0.328	-0.509	2.249	0.000	0.000
P1, T5	11.320	0.000	-0.763	0.000	0.000	0.000	13.110	0.000	-0.567	2.213	0.000	0.000
P2	11.605	0.000	-0.755	0.000	0.000	0.000	12.929	-0.328	-0.509	2.248	0.000	0.000

effectively. In order to verify the usefulness of the HCLS component, the regulators of gene 1 were inferred using the proposed algorithm without the HCLS component. Performance of this modified algorithm is compared with that of the proposed algorithm in Fig. 9.1 in terms of convergence curves. From the graph it is very evident that HCLS was very effective in efficient optimization of the fitness function and thereby helpful in estimating the parameter values more accurately. In order to show the usefulness of double optimization in estimating the robust parameter values, the parameter values for gene 3, estimated at the end of different trials of phase 1 and after the optimization of phase 2, from 10% noisy expression data are presented in Table 9.1. The results presented in Table 9.1 clearly shows that different trials of optimization may converge to some local optimum and thereby some essential regulation may be lost (e.g. $g_{3,2}$ in Trial 3 of Phase 1) or type of the regulation may be changed (e.g. $h_{3,2}$ in Trial 3 of Phase 1) or the parameter values may be very different. A second optimization phase, starting from these local solutions can automatically detect all the robust interactions and give more robust values for the kinetic parameters. These empirical analyses of the algorithm were useful to show that the different components of the proposed algorithm were worthy to accomplish the purpose.

An AIC based fitness criterion was proposed in this work for evaluating the candidate solutions, during the search for the optimal set of parameters. In the proposed fitness evaluation function, the penalty term of AIC was extended. The purpose of this additional penalty term was to facilitate the selection of models with sparse network architecture. Following the guideline of previous work, this penalty term was designed so that it would penalize the fitness score of a candidate network model if it has regulators more than the maximum in-degree of that network. Therefore, this penalty term remains silent as long as the number of regulators of a gene does not exceed the maximum given limit. Otherwise, it penalizes the competing model and thus helps to identify the skeletal architecture. In this new

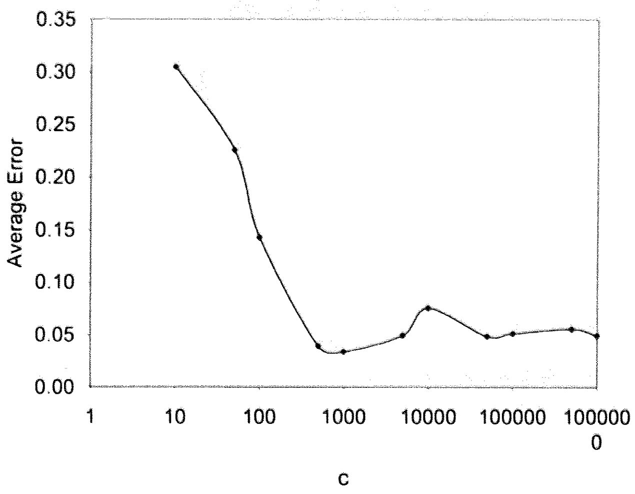


Figure 9.2: Study on the effect of penalty constant 'c'

penalty term there is an additional parameter 'c' which was given a value 1000. Choice of this parameter value was very straightforward and was determined as follows. As mentioned early, the algorithm performs a structure skeletalizing for reducing the computational burden by setting a parameter to zero if its absolute value is less than $\delta = 0.001$. It is needed to penalize the fitness score effectively for all additional regulators (that lie beyond the threshold of maximum in-degree) until their values go down below δ. Therefore, such a value for the parameter 'c' is quite natural and was found useful. In order to study the sensitivity of 'c', experiments were performed with different values for the parameter for NET1 under ideal condition. The result is shown in Fig. 9.2 in terms of average error value. As the figure shows, any value greater than 1000 for the parameter was suitable for penalizing the fitness score for additional regulators. And the specificity S_p of the reconstructed networks did not change significantly when a $c \geq 1000$ was chosen. This result also supports the anticipation about the choice of this penalty constant.

In a wide range of experiments, the reverse engineering algorithm showed its competence in inferring the correct transcriptional regulations and estimating the correct kinetic parameter values. If sufficient amount of gene expression data is given and the noise level is not very critical then the algorithm can identify almost all the correct regulatory interactions in a medium sized network. Also the kinetic parameters were estimated with pretty high accuracy. Because of the high flexibility of the S-system model, the performance of the reconstruction algorithm degraded with the decrease in microarray data sets used for inference. When inference was done from insufficient amount of experimental data, the number of correctly inferred regulations and the accuracy of the inferred parameter values decreased. One point should be noted here that the gene expression profiles, used in these experiments, were generated from random initial concentration levels. If data were collected for various stimuli or through gene disruption then the networks could be reconstructed from fewer data-sets because such data provide more information about the biochemical pathways existing in a group of closely coupled genes.

The performance of a reverse engineering algorithm has always been affected by the noise level in the experimental data and the proposed one is not an exception. Though the exact network topology and quasi accurate parameters were estimated in ideal noise free condition, the quality of the prediction deteriorated, both in terms of pathway identification and reaction kinetics estimation, with the increase in noise level. However, when compared with the predictions using other fitness evaluation criteria, the overall parameter accuracy and sensitivity/specificity were found better in the proposed fitness evaluation functions for different noise patterns. This empirically shows the superiority of the proposed fitness functions in evaluating the candidate networks.

In analysis of the real gene expression data, primarily because of insufficient experimental data with high noise, the proposed method could not get any consistent result for the estimated network in different experimental runs. Therefore, Z-score analysis was performed for a qualitative prediction of the bio-molecular pathways among the genes. Most of the pathways in the reconstructed networks were consistent with the results reported in the literature. The primary reason of erroneous predictions was deficiency in the time course data.

A summary of performance comparison between the proposed method and other reverse engineering algorithms based on the reported results for NET1 is presented in Table 9.2. As mentioned earlier, the proposed algorithm took approximately 12 minutes to solve a subproblem of NET1 in a 1.7 GHz Pentium processor. So the total time required for reconstructing the complete network was approximately 60 (5×12) minutes. According to Table 9.2, it is obvious that the proposed algorithm is more efficient than the other existing algorithms for the same task. The proposed algorithm not only took minimum time to reconstruct the network but also exhibited excellent performance in estimating nearly identical parameter values. The suitability for parallelization and the computational efficiency of the proposal make it most competitive for large scale genetic network reconstruction problems. Moreover, the algorithm exhibited such performance using fewer or equal amount of gene expression data used by other methods. And the ability of the method to identify all the regulations of NET1 correctly from expression data cluttered by 5% Gaussian noise,

Table 9.2: Performance comparison of different reconstruction algorithms for the problem of NET1

Method	ODE solving method	Time points used	CPU Time	Estimated results	Noisy exp considered?
This work	Num. intg. with spline	10 sets (10×11)	5×12 mins (Pen 1.7 GHz)	Nearly identical	Yes
HDE (Tsai and Wang [130])	Converted to algebraic eqn	10 sets (10×?)	170 mins (Pen 2.4 GHz)	Nearly identical	Not reported
GLSDC (Kimura *et al.* [57])	Num. intg. with local linear regression	15 sets (15×11)	8×89 mins (Pen 933 MHz)	Accurate	Not reported
PEACE1 (Kikuchi *et al.* [54])	Numerical integration	10 sets (10×10)	7×10hrs (1040× Pen 933 MHz)	Not exact structure	Not reported

makes it very apposite to work with the quality and quantity of gene expression data that the current technology offers.

In an overall, the results demonstrate that the proposed method can infer the regulatory relationships in a co-regulated module using gene expression data as input. However, the reconstruction algorithm may not be able to capture the complete network architecture in a single reconstruction attempt or may only be able to predict some qualitative interaction because of insufficient data availability or excessive noise in the data. Still this type of indication can be very useful for biologists to develop conjectures or to design additional experiments which may in turn help to identify new interactions.

Chapter 10

Conclusion

In this dissertation, a methodology has been presented to identify the molecular pathways of gene regulation from the gene expression data. The proposed algorithm was found competent to infer various types of transcriptional regulations, such as auto, cyclic and different types of feedback control, correctly with their kinetic parameters.

One of the objectives of the research was to develop a robust evolutionary optimizer capable of locating the global optimum in multimodal and deceptive problem domain, so that it can be used to estimate the parameter set for the disintegrated S-system model that was used for representing the mechanism of gene regulation. Therefore, some proposals are presented to enhance an existing evolutionary algorithm for real parameter optimization. In these efforts crossover-based local-search schemes have been hybridized with the classic differential evolution algorithm to accelerate it. Empirical studies showed that such memetization of the algorithm not only increases its convergence velocity but also increases its robustness and reliability. Hence, the proposed crossover based local searches were useful to improve the performance of the differential evolution algorithm.

Another theme of this research was to penalize the complex model structures to handle the high degree-of-freedom that the S-system model offers. Modifications and/or extensions have been proposed to the existing criteria with the help of some biological fact. One of the fitness criteria was based on MSE and the other was based on AIC. Experimental results suggest that these fitness functions were more effective to identify the skeletal structure and to estimate the precise parameter values.

Another the central idea proposed in this work was to use a hill climbing local search to accelerate and facilitate the evolution of the sparse candidate structures of the candidate models. The evolved network structures were sparse in nature which shows the competence of the algorithm to identify such skeletal architecture. And the empirical analysis of the algorithm confirms the local search's usefulness in accelerating the identification of sparse networks. The role of double optimization in identifying the robust biomolecular interactions from the considerably noisy gene-expression data has been also discerned.

In order to extract useful regulatory information from the gene expression data that the current technology offers, the reverse engineering algorithm should be robust against small gene expression data size and significant level of noise. Like other reconstruction methods, for the proposed algorithm the sensitivity and specificity of the reconstructed network reduced with decreasing sample size and increasing noise level. However, it was found that the reconstruction method can identify some regulations from very limited amount of expression data in presence of noise which indicates its robustness against small sample size and noise.

10.1 Future prospects

The meticulous identification of the genome-wide regulatory networks is one of the biggest challenges that the filed of systems biology is facing. This work has added some new guidelines towards the ultimate goal of automated reconstruction of cellular pathways. Yet, a long way to go before the use of such a system becomes widespread in drug design and disease treatment. Many scopes and guidelines for future studies have become perceptible from the discussions of the work. This section gives more specific directions to such prospects.

Reconstruction of a system-level gene regulatory network which includes thousands of genes and very complex interactions among them should make use of all possible genomic knowledge available rather than building the network from the scratch. Besides, for the system level integration of the biological organisms we should be able to use all sort of biological and genomic resources obtainable from different databases, such as molecular pathway database, nucleotide sequence database, protein sequence database, information about biomarkers, transcription locus, taxonomy, phylogeny, macromolecular structures, evolutionary relatedness or any form of information from any other source. However, such information can be used at different stages of the reconstruction process: to validate the prediction, reduce the scope and dimension of the search, increase the accuracy of the correct predictions, disprove the false regulations. Moreover, use of such prior knowledge may reduce the effect of noise and also the data requirement for the reconstruction process and lower the cost of expensive microarray analysis. Additional benefit of such knowledge incorporation is more efficient inference process. However, while use of such prior information is indispensable for very large gene network reconstruction, appropriate utilization of this knowledge should be guaranteed by refining ambiguous and conflicting knowledge from different databases and verifying the authenticity of the used knowledge.

Top-down or bottom-up reconstruction approaches can further facilitate the inference process for a real large gene network. Though, the model has already been disintegrated at gene level, when a system level network is considered, every gene will have hundreds of prospective regulators. Therefore, top-down or bottom up schemes by grouping of genes through clustering can reduce the problem dimensionality drastically and further assist the inference method.

Use of the framework of modular network can significantly reduce the complexity of reconstructing a gene network with hundreds or thousands of genes. Genes and proteins are organized into functional modular networks in which the network context of a gene or protein has implications for cellular function [53]. At the system level, a gene regulatory network consists of assemblages of these functional units each of which may consists of a group of 10 to 100 genes. Therefore, subdivision of the complete network into its modular components can be useful to identify higher level interactions among these modules. The top-down/modular analysis does not require complete information on the molecular interactions and has proven advantageous both for discovering unknown interactions and for estimating unknown kinetic parameters [53, 143]. Such techniques include clustering of genes and other statistical algorithms based on similarity in expression profiles.

The proposed method can also be integrated within the framework of other reconstruction method based on some more abstract models such as Boolean models. Such simpler models are capable of giving a first hand approximation of the structure or gene interaction from very limited data which can further be refined by using the proposed method. Such Boolean network models also prove a useful conceptual tool for investigating the principles of network organization and dynamics and can give information about the constraints in the number of inputs and outputs per gene, input and output sharing among genes evolved within a gene family or pathway, and restrictions on rule types etc. Such information can in turn facilitate the reconstruction process.

The memetic optimizer developed in this work has shown the promise of using crossover based local search strategies for Differential Evolution algorithm. So the prospect of the proposed local search strategies can be explored for other evolutionary algorithms particularly for the self-adaptive ones. Besides, most of the real-world problems are highly multimodal and the proposed memetic optimizer has proven its competence for multimodal problems. Therefore, this optimizer can be used as a handy tool for all sort of real world optimization problems.

Appendix A

DNA Microarray Experiment

Gene expression is a highly complex and tightly regulated process that allows a cell to respond dynamically both to environmental stimuli and to its own changing needs. Studying which genes are active and which are inactive in different cell types helps scientists to understand both how these cells function normally and how they are affected when various genes do not perform properly. In the past, scientists have only been able to conduct these genetic analyses on a few genes at once. With the development of DNA microarray technology, however, scientists can now examine how active thousands of genes are at any given time [115]. Gene expression profiling or microarray analysis have revolutionized the biomolecular research and have stimulated the further understanding of biological processes.

A microarray works by exploiting the ability of a given mRNA molecule to bind specifically to, or hybridize to, the DNA template from which it originated. By using an array containing many DNA samples, scientists can determine, in a single experiment, the expression levels of hundreds or thousands of genes within a cell by measuring the amount of mRNA bound to each site on the array. With the aid of a computer, the amount of mRNA bound to the spots on the microarray is precisely measured, generating a profile of gene expression in the cell [42].

A.1 Principle of Microarray Technology

Though the complete process of microarray experiments is quite complicated the basic principle underlying the microarray technology is very simple and is that complementary nucleic acid will hybridize [45]. Here are described the major steps involved in a microarray experiment (Fig. A.1).

(1) **Fabrication of a cDNA microarray:** The first step in cDNA microarray

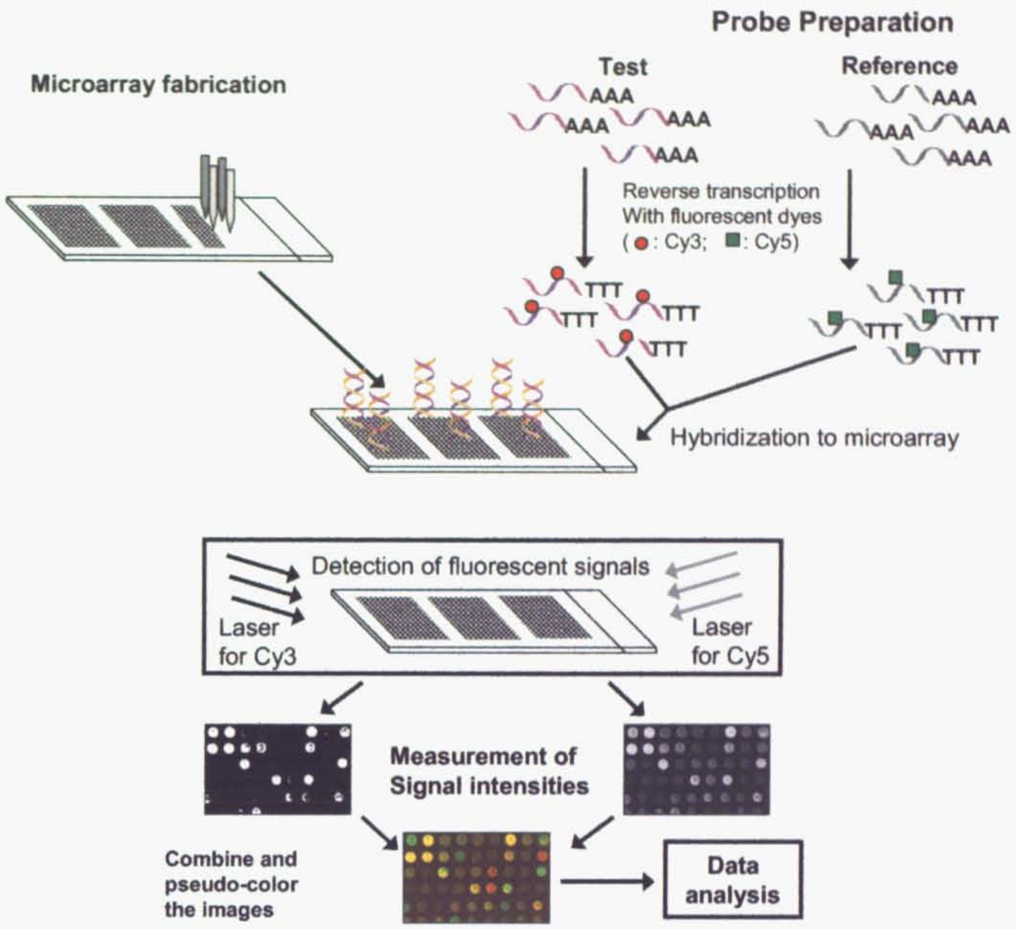


Figure A.1: DNA Microarray experiment overview (adapted from [45])

construction is the preparation of cDNA as arrayed targets. Any double-stranded cDNA, and probably single-stranded cDNA, can be used for the fabrication of a microarray. Conventional cloning techniques and PCR amplification are usually required for the target cDNA preparation. Target cDNAs are printed onto a poly-L-lysine-coated microscope glass slide by means of a robotic arrayer [45]. The coated surface provides attachment sites for the target cDNA so that it remains bound to the glass surface during hybridization and washing. Before hybridization, four steps of post-processing, namely rehydration, snap-drying, ultraviolet cross-linking and blocking of coated glass surface, of the microarray is required to minimize hybridization background noise.

(2) **Target preparation:** Messengers RNA are extracted from two probe samples which to be compared for expression level. mRNA from test and reference samples are then transformed in cDNA by reverse transcription and labeled with two different dyes [115]. Although various types of fluorescent labeling materials are now commercially available, Cy3- and Cy5-dUTP or dCTP are generally used for

this purpose.

(3) **Hybridization:** The separately labeled probes are pooled and concentrated. After concentration, successfully labeled probe can be identified by its color. Probes are suspended into the hybridization solution containing 3X SSC and 0.5% sodium dodecyl sulfate (SDS), and hybridized to the microarray under a coverslip in a specially designed hybridization chamber that is submerged in a 65°C water bath for 14-20 hrs [45]. At this temperature, a DNA strand that encounter the complementary strand, match together to create a double strand DNA. The fluorescent DNA will then hybridize on the spotted ones. After hybridization, the microarray is washed and air-dried.

(4) **Slide scanning:** A laser excites each spot and the fluorescent emission gather through a photo-multiplicator (PMT) coupled to a confocal microscope [115]. Two images are obtained where grey scales represent fluorescent intensities read. If grey scales are replaced by green scales for the first image and red scales for the second one, by superimposing the two images one image is obtained which composed of spots going from green ones (where only DNA from the first condition is fixed) to red (where only DNA from the second condition is fixed) passing through the yellow color (where DNA from the two conditions are fixed on equal amount).

(5) **Data analysis:** Now two microarray images are available from which the number of DNA molecules to be calculated in each experimental condition. To do so, the signal amount in the green dye emission wavelength and the signal amount in the red dye emission wavelength are measured. Then these signal amounts are normalized according to various parameters. It is supposed that the amount of fluorescent DNA fixed is proportional to the mRNA amount present in each cell at the beginning and we calculate the red/green fluorescence ratio. If this ratio is greater than 1 (red on the image), the gene expression is greater in the second experimental condition, if this ratio is smaller than 1 (green on the image), the gene expression is greater in the first condition.

A.2 Types of Microarray Technologies

Microarrays come in several varieties, each of which has specific advantages for research and screening. Two major types of microarrays are dominantly used: cDNA arrays and oligonucleotide arrays. The major difference between these two is in oligonucletide array, gene expression level in the sample is directly reflected instead of through the ratio of test and reference sample in the cDNA spotted arrays.

cDNA microarrays have evolved from Southern blots. It uses lengths of complementary DNA (or cDNA) produced from cellular messenger RNA using the reverse transcriptase polymerase chain reaction (RT-PCR).The probes from amplified PCR product are spotted onto a microscope slide, or alternative intermediate by a robot. Two cDNA samples from different source (such as normal and treatment) are labeled with Cy3 or Cy5 fluorescent dye and combined together to hybridize with the probes [115]. After laser excitation, the fluorescence is collected and the ratio is measured.

The other form of microarray consists of oligonucleotides or peptide nucleic acids synthesized either in situ on the chip or by conventional synthesis followed by immobilization on the chip. Scientists at Affymetrix developed this type of microarray, which is often called a DNA chip. Such High-density oligonucleotide probe array is the core technology of Affymetrix [39]. The probes are synthesized directly on a glass substrate. Millions of copies of a specific probe are located within a discrete area on the array called a probe cell. The probe pair consists of two probe cells - the perfect match (PM) and mismatch (MM). PM probes are designed to be complementary to a reference sequence, and MM probes are designed to be complementary to a reference sequence except for a homomeric base mismatch at the central position. And a transcript is represented as a probe set which is made up of probe pairs. The statistical algorithm uses probe pair intensity to generate a Detection call such as Present (P) and Absent (A).

Arrays can also use proteins. Now commercial protein chip are available that can be used in proteomics research. Some pharmaceutical companies are also using microarray formats to screen the activity of various chemical compounds against targets in their drug discovery programs. In addition, some researchers have begun to work with cell and tissue arrays. Several companies now offer these as part of their product lines.

These fundamental platforms for microscale experimentation have resulted from the marriage of several technologies. Robotic engineering, pin technology, molecular biology, DNA sequencing, optical and laser technology, and informatics have all contributed to the development of microarrays. Since few companies can develop expertise in all these areas, individual firms must partner with others to create compatible systems. Alternatively, researchers intent on preparing their own arrays must assemble components from several vendors.

A.3 Promise of Microarray Technology

Microarrays hold tremendous promise in health care and drug discovery. And some of its promises are already being fulfilled. The future of the array technology can be pictured as: a hand-held instrument that a physician could use to quickly diagnose cancer or other diseases during a routine office visit. The same instrument would also be able to facilitate a personalized treatment regimen, exactly right for you. Studies for system-level understanding of different organisms are currently going on in many laboratories around the world. Everything from mice to humans is being studied analyzing the gene expression obtained from different types of microarrays. In drug discovery, many pharmaceutical companies are using microarrays to study possible drug targets. Personalized drugs, molecular diagnostics, integration of diagnosis and therapeutics - these are the long-term promises of microarray technology. Many more are focusing on toxicogenomics and pharmacogenomics, two research areas based on DNA microarrays. In clinical research, many experiments in oncology, infectious disease and general disease classification and prognosis have been conducted. From their results, we now know many new subclasses of cancer and how a particular tumor may respond to treatment. And finally in clinical applications, the AmpliChip CYP450 by Roche and Affymetrix is intended to be an aid for physicians in individualizing treatment doses for patients on therapeutics metabolized through these genes. The AmpliChip CYP450 is already in use.

From the pure biological gene expression studies to clinical research, from drug discovery and development to personalized medicine, microarrays are already being used frequently by today's researchers.

A.4 Limitations of Microarray Technology

Despite the great promises of microarrays in health care, and their successes in both medical and biological research, the technology is still far away from daily use in the clinic. Why, despite the great potential of microarrays, is their use so far from the clinic?

While many laboratories have successfully implemented microarrays, there are many difficulties in the process. RNA extraction, amplification, and hybridization are all procedures involved in microarray research, and all have inherent problems leading to statistical errors. DNA microarrays are not accessible in abundant due to their high cost and difficulty of use. Besides, It is very difficult to obtain high quality tissues from hospitals after surgery for laboratories to use in microarray research. It is also difficult to extract enough RNA for microarray analysis from just a few cells. Lack of sufficient data is a result of the high cost of microarray experiments and the need for a database that researchers may quickly access for past experimental results. While the gene expression profile allows a whole genome approach to studying the cell, there are other factors effecting cell phenotype. Moreover, some expressed genes do not lead to proteins, as downstream regulation can occur within the cell. Furthermore, some researchers still do not trust the data obtained from microarrays because of the high variability. Repeated studies seem to yield significant differences in the scale factor and present call percentages. Still more problems exist. Once the microarray data, which is in the form of a gene list, is validated, a detailed analysis is necessary before useful information can be obtained.

However, as the microarray is becoming more mature, it is overcoming many of its shortcomings with the help of cutting-edge technologies and equipments. And it is expected that it will continue to play a key role in future genome research.

A.5 Microarray data to Regulatory Networks

Reverse engineering of regulatory networks relies upon the assumption that given enough data on actual genetic expression levels, we can deduce how genes are regulated. For this purpose it is practical to divide microarray data into two groups, steady state and time series.

Steady State

In a steady state experiment several closely related cell cultures are grown under very similar conditions, with expression data being collected only at one specific time point, usually during steady growth. Changes in expression levels are then identified by comparing across the cell cultures. A typical setup involves comparing a single gene knock-out mutant to the wild-type grown on the same media. The closer the two cultures are to each other, the better, since this eliminates noise from biological processes which are not related to the knocked-out gene. Ideally only those genes regulated either directly or indirectly (perhaps through another regulator) by the product of the gene which was knocked-out will show any significant change in expression level.

Time Series

In a time series experiment samples are taken at different time intervals from a single cell culture during growth. For each of these samples microarray data can be produced revealing the changes in expression levels over time. Using a mathmatical model it is now theoretically possible to reveal the underlying regulation. In reality, however, it is very difficult to obtain the exact networks from time series, because the whole thing relies on the time intervals being selected in accordance with the delay between the expression of the regulator and the effect it has on the regulated genes. Also, limitations in the experimental conditions means that it is difficult to obtain good quality data for more than approximately 20 time points, which are too few for most reverse engineering models.

Appendix B

Spline Interpolation

In engineering applications, data collected from the field are usually discrete and the physical meanings of the data are not always well known. To estimate the outcomes and, eventually, to have a better understanding of the physical phenomenon, a more analytically controllable function that fits the field data is desirable. The process of finding the coefficients for the fitting function is called **curve fitting**; the process of estimating the outcomes in between sampled data points is called **interpolation**; whereas the process of estimating the outcomes beyond the range covered by the existing data is called **extrapolation**.

Suppose we are given a sequence of n distinct numbers $x_1, x_2, x_3, \cdots, x_n$ (say $x_1 < x_2 < \cdots < x_n$) called **nodes** and for each x_i $(i = 1, 2, \cdots, n)$ a second number y_i $(i = 1, 2, \cdots, n)$ which are value of a function $f(x)$. We don't know any analytic expression for $f(x)$ using that we can calculate its value at any arbitrary point and we are looking for a function f so that

$$f(x) = y$$

A pair $\{x_i, y_i\}$ is called a **data point** and f is called the **interpolant** for the data points. Often x_i's are equally spaced, but not necessarily. The task is now to estimate $f(x)$ for arbitrary x by, in some sense, drawing a smooth curve through (and perhaps beyond) the x_i [100]. If the desired x is in between the largest and smallest of the x_i's, the problem is called interpolation.

A reasonable functional form should be used in an interpolation or extrapolation scheme in order to model the function within the given points and beyond. Such a functional form should be general enough to approximate wide range of functions that may be encountered in practice [100].

There are many different interpolation methods. When choosing an appropriate interpolation method the issues to be resolved are: How accurate is the method? How expensive is it? How smooth is the interpolant? How many data points are needed? [28].

One of the simplest methods is linear interpolation. It is easy and fast and pays its price by not being accurate. Polynomial interpolation is a generalization of linear interpolation. Note that the linear interpolant is a linear function. And in polynomial interpolation this linear function is replaced by a higher order polynomial. This is one of the most commonly used types of interpolation. Polynomial interpolation solves most of the problems of linear interpolation. But, polynomial interpolation also has some disadvantages. Calculating the interpolating polynomial is relatively very computationally expensive. Furthermore, polynomial interpolation may not be so exact after all, especially at the end points [28]. These disadvantages can be avoided by using spline interpolation.

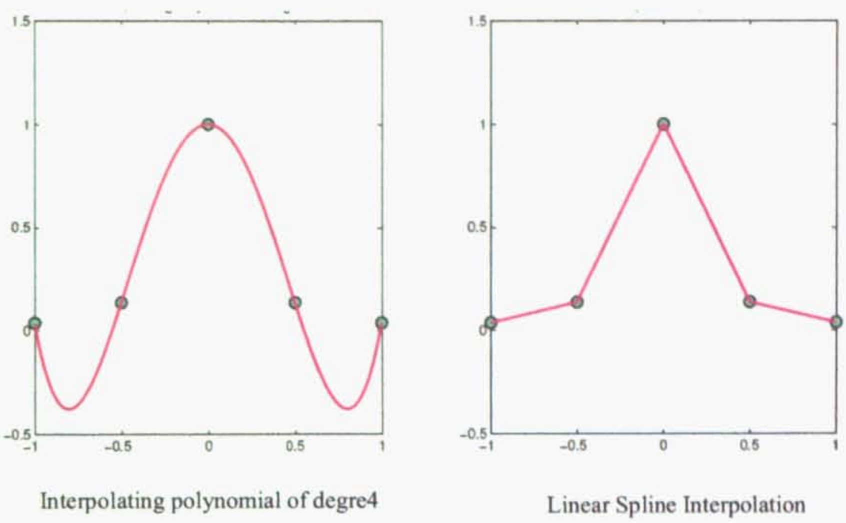
Spline Interpolation

It performs the interpolation of interorder background using smoothing spline polynomials. Spline interpolation consists of the approximation of a function by means of series of polynomials over adjacent intervals with continuous derivatives at the end-point of the intervals. That is, here, the interpolant is a special type of piecewise polynomical known as spline. Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline.

Given $k+1$ distinct knots x_i such that $x_0 < x_1 < \cdots < x_k$ with $k+1$ knot values y_i we are trying to find a spline function $S(x)$ of degree k such that

- The domain of S is an interval [a, b].
- $S, S', \cdots, S^{(k-1)}$ are continuous on [a,b].
- There are points x_i such that $a = x_0 < x_1 < \cdots < x_k = b$ and such that S is a polynomial of degree at most k on each $[x_i, x_{i+1}]$.

In other words, splines are piecewise polynomial functions whose derivatives obey certain continuity constraints. When k = 1, the splines are called linear splines. When k = 2, the splines are called quadratic splines, and when k = 3, the splines are called cubic splines.



Interpolating polynomial of degre4

Linear Spline Interpolation

Figure B.1: Linear Spline Interpolation

Linear Splines

Linear spline interpolation is the simplest form of spline interpolation. The data points are graphically connected by straight lines. The resultant spline is just a polygon (Fig. (B.1)). That is, here, we seek a linear spline function $S(x)$ such that $S(x_i) = y_i$ for $0 \leq i \leq n$.

Let

$$S(x) = \begin{cases} S_0(x) & x_0 \leq x \leq x_1 \\ S_1(x) & x_1 \leq x \leq x_2 \\ \vdots & \vdots \\ S_{k-1}(x) & x_{k-1} \leq x \leq x_k \end{cases} \tag{B.1}$$

where each $S_i(x)$ is linear.

Let us consider the linear function $S_i(x)$. The line passes through the points (x_i, y_i) and (x_{i+1}, y_{i+1}), so the slope of $S_i(x)$ is

$$m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{B.2}$$

Also, it can be stated that the line passes through the points (x_i, y_i) and $(x, S_i(x))$ for any x in the interval $[x_i, x_{i+1}]$, so for any x in this interval,

$$m_i = \frac{S_i(x) - y_i}{x - x_i} \tag{B.3}$$

which gives

$$S_i(x) = y_i + m_i(x - x_i) \tag{B.4}$$

Strategy for Linear Spline Interpolation

1. Compute m_i for $0 \leq i \leq n-1$ using (B.2)
2. For any $x \in [x_0; x_n]$, find the correct interval in which x lies, and evaluate the corresponding spline using (B.4).

Algorithm

(given x_i, y_i, m_i, for $0 \leq x \leq n-1$ and x)

```
for $i = 0 :\ n-1$
    if $x \leq x_{i+1}$
        break
    end
end
$S(x) = y_i + m_i(x - x_i)$
```

Compared to polynomial interpolation, linear spline interpolation is very cheap: the slopes can be computed in $\sim 3n$ flops, and given these slopes, the spline function can be evaluated in exactly 3 flops. Therefore, linear spline is free from the disadvantages of other splines, i.e. not expensive and easy to implement.

Cubic Spline

The cubic spline interpolation uses third degree polynomials to connect the data points which often results in strikingly smooth curve fits. It is assumed that the cubic polynomial function's first and second derivative is continuous at the interpolation nodes.

The cubic spline not only interpolates the data but matches the first and second derivatives at the knots. Notice, from the above definition, one is free to specify constraints on the endpoints. One common end point constraint is $S''(a) = 0S''(b) = 0$, which is called the natural spline. In other words, the natural cubic spline has zero second derivatives at the endpoints. It is the smoothest of all possible interpolating curves in the sense that it minimizes the integral of the square of the second derivative. Other popular choices are the clamped cubic spline, parabolically terminated spline and curvature-adjusted spline. Cubic splines are frequently used in numerical analysis to fit data. Compared to linear splines cubic splines are naturally more expensive to evaluate and difficult to implement.

Spline interpolation is a powerful data analysis tool. Splines correlate data efficiently and effectively, no matter how random the data may seem. Once the algorithm for spline generation is produced, interpolating data with a spline becomes an easy task.

Appendix C

Benchmark Functions

The test suite that was used for different experiments of Chapter 5, consists of 20 benchmark functions. The first 10 test functions of the suite are functions commonly found in the literature and the other benchmarks are the first 10 functions from the newly defined test suite for CEC 2005 special session on real-parameter optimization [125]. The test suite was as follows:

1. F_{sph}: Sphere Function
2. F_{ros}: Rosenbrock's Function
3. F_{ack}: Ackley's Function
4. F_{grw}: Griewank's Function
5. F_{ras}: Rastrigin's Function
6. F_{sch}: Generalized Schwefel's Problem 2.26
7. F_{sal}: Salomon's Function
8. F_{wht}: Whitely's Function
9. F_{pn1}: Generalized Penalized Function 1
10. F_{pn2}: Generalized Penalized Function 2
11. F_1: Shifted Sphere Function
12. F_2: Shifted Schwefel's Problem 1.2
13. F_3: Shifted Rotated High Conditioned Elliptic Function
14. F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness
15. F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds
16. F_6: Shifted Rosenbrock's Function
17. F_7: Shifted Rotated Griewank's Function without Bounds

18. F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds
19. F_9: Shifted Rastrigin's Function
20. F_{10}: Shifted Rotated Rastrigin's Function

Definitions of the first ten functions are as follows

$$F_{sph}(\vec{x}) = \sum_{i=1}^{N} x_i^2; \qquad -100 \leq x_i \leq 100; \qquad F_{sph}^* = F_{sph}(0,\cdots,0) = 0$$

$$F_{ros}(\vec{x}) = \sum_{i=1}^{N-1} (100(x_{i+1}-x_i^2)^2+(1-x_i)^2); \quad -100 \leq x_i \leq 100; \quad F_{ros}^* = F_{ros}(1,\cdots,1) = 0$$

$$F_{ack}(\vec{x}) = 20 + exp(1) - 20exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right);$$

$$-32 \leq x_i \leq 32; \qquad F_{ack}^* = F_{ack}(0,\cdots,0) = 0$$

$$F_{grw}(\vec{x}) = \sum_{i=1}^{N}\frac{x_i^2}{4000} - \prod_{i=1}^{N}\cos\frac{x_i}{\sqrt{i}} + 1; \qquad -600 \leq x_i \leq 600; \qquad F_{grw}^* = F_{grw}(0,\cdots,0) = 0$$

$$F_{ras}(\vec{x}) = 10N + \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i)); \qquad -5 \leq x_i \leq 5; \qquad F_{ras}^* = F_{ras}(0,\cdots,0) = 0$$

$$F_{sch}(\vec{x}) = 418.9829N - \sum_{i=1}^{N}\left(x_i \sin\left(\sqrt{|x_i|}\right)\right);$$

$$-500 \leq x_i \leq 500; \qquad F_{sch}^* = F_{sch}(420.9687,\cdots,420.9687) = 0$$

$$F_{sal}(\vec{x}) = -cos\left(2\pi\sqrt{\sum_{i=1}^{N} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{N} x_i^2} + 1;$$

$$-100 \leq x_i \leq 100; \qquad F_{sal}^* = F_{sal}(0,\cdots,0) = 0$$

$$F_{wht}(\vec{x}) = \sum_{j=1}^{N}\sum_{i=1}^{N}\left(\frac{y_{i,j}^2}{4000} - cos(y_{i,j}) + 1\right); \quad where \quad y_{i,j} = 100(x_j - x_i^2)^2 + (1-x_i)^2$$

$$-100 \leq x_i \leq 100; \qquad F_{wht}^* = F_{wht}(1,\cdots,1) = 0$$

$$F_{pn1}(\vec{x}) = \frac{\pi}{N}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{N-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_N - 1)^2\right\}$$

$$+\sum_{i=1}^{N} u(x_i, 10, 100, 4); \qquad where \;\; y_i = 1 + \frac{1}{4}(x_i + 1) \quad and$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$$

$$-50 \leq x_i \leq 50; \qquad F^*_{pn1} = F_{pn1}(-1, \cdots, -1) = 0$$

$$F_{pn2}(\vec{x}) = 0.1\left\{\sin^2(\pi 3x_1) + \sum_{i=1}^{N-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_N - 1)^2[1 + \sin^2(2\pi x_N)]\right\}$$

$$+\sum_{i=1}^{N} u(x_i, 5, 100, 4);$$

$$-50 \leq x_i \leq 50; \qquad F^*_{pn2} = F_{pn2}(1, \cdots, 1) = 0$$

Functions F_1 to F_{10} are designed by modifying classical benchmark functions to test the optimizers ability to locate a global optimum under a variety of circumstances such as translated and/or rotated landscape, optimum placed on bounds, Gaussian noise and/or bias added etc. [125]. A complete definition of these functions are available online at http://www.ntu.edu.sg/home/epnsugan and in [125]. And more detailed description of the other functions can be found in [146, 101]. In this test suit F_1 to F_5, F_{sph} and F_{ros} are unimodal and the rest are multimodal functions. All the chosen benchmarks are minimization problems.

References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second Int. Symposium on Information Theory*, pages 267–281, 1973.

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

[3] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734, 2000.

[4] J. S. Almeida and E. O. Voit. Neural-network-based parameter estimation in S-system models of biological networks. In *Genome Informatics 14*, pages 114–123, 2003.

[5] S. Ando. *Estimation of Gene Network Using Real-Coded Genetic Algorithm.* PhD thesis, School of Engineering, The University of Tokyo, 2004.

[6] S. Ando and H. Iba. Construction of genetic network using evolutionary algorithm and combined fitness function. In *Genome Informatics 14*, pages 94–103, June 2003.

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

[8] A. Arkin, J. Rossb, and H. H. McAdamsa. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, 149:1633–1648, 1998.

[9] M. Arnone and E. Davidson. The hardwiring of development: Organization and function of genomic regulatory systems. *Development*, 124(10):1851–1864, May 1997.

[10] A. Atherly, J. R. Girton, and J. F. McDonald. *The Science of Genetics.* Saunders College Publishing, USA, 1999.

[11] N. K. Bambha, S. S. Bhattacharyya, J. Teich, and E. Zitzler. Systematic integration of parameterized local search into evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):137–155, 2004.

[12] R. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, NJ, 1961.

[13] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[14] J. M. Bower and H. Bolouri, editors. *Computational Modeling of Genetic and Biochemical Networks.* The MIT Press, London, England, 2004.

[15] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.

[16] T. A. Brown. *Genome.* BIOS Scientific Publisher Ltd, New York, USA, 1999.

[17] R. Callard, A. J. T. George, and J. Stark. Cytokines, chaos, and complexity. *Immunity*, 11:507–513, 1999.

[18] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, July 1998.

[19] L. Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[20] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[21] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66:301–340, 2004.

[22] K. Deb, A. Anand, , and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):371–395, 2002.

[23] X. Deng, H. Geng, and H. Ali. EXAMINE: A computational approach to reconstructing gene regulatory networks. *Biosystems*, 81(2):125–136, 2005.

[24] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mRNA expression levels during CNS development and injury. In *Pacific Symposium on Biocomputing 4*, pages 41–52, 1999.

[25] P. D'Haeseller, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

[26] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Science*, 95(25):14863–14868, 1998.

[27] L. J. Eshelman and J. D. Schaffer. *Foundations of Genetic Algorithms 2*, chapter Real-Coded Genetic Algorithms and Interval Schemata, pages 187–202. Morgan Kaufmann, San Mateo, CA, 1993.

[28] R. L. Eubank. *Nonparametric Regression and Spline Smoothing.* Marcel Dekker, New York, 1995.

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

[30] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Computational Biology*, 7(3-4):601–620, 2003.

[31] S. Fujita, M. Matsui, H. Matsuno, and S. Miyano. Modeling and simulation of fission yeast cell cycle on hybrid functional petri net. *IEICE Trans. Fundamentals*, E86-A:2919–2928, 2003.

[32] R. Gämperle, S. D. Müller, and P. Koumoutsakos. A parameter study for differential evolution. In *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, pages 293–298, 2002.

[33] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[34] D. L. Hartl and E. W. Jones. *Genetics: Analysis of Genes and Genomes.* Jones & Bartlett Publishers, USA, 2004.

[35] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins. Computational studies of gene regulatory networks: In numero molecular biology. *Nature Review Genetics*, 2(4):268–279, 2001.

[36] D. Heckerman, C. Meek, and G. Cooper. *Computation, Causation and Discovery*, chapter A Bayesian approach to causal discovery. MIT Press, 1999.

[37] O. Hirose, R. Yoshida, S. Imoto, and S. Miyano. Inferring module-based gene networks from time course microarray data by state space models. In *Genome Informatics 16(2)*, page P157, 2005.

[38] R. Hofestädt. A petri net application to model metabolic processes. *Systems Analysis Modelling Simulation*, 16(2):113–122, 1994.

[39] http://www.gene chips.com/.

[40] http://www.genome.ad.jp/kegg/.

[41] http://www.honeylocust.com/RngPack/.

[42] http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html.

[43] http://www.weizmann.ac.il/mcb/UriAlon/.

[44] T. E. Ideker, V. Thorsson, and R. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Pacific Symposium on Biocomputing 5*, pages 302–313, 2000.

[45] K. Iida and I. Nishimura. Gene expression profiling by DNA microarray technology. *Crit. Rev. Oral Bilo. Med.*, 13(1):35–50, 2002.

[46] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. In *Pacific Symposium on Biocomputing 7*, pages 175–186, 2002.

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

[48] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transaction on Evolutionary Computation*, 7(2):204–223, 2003.

[49] C. Janion. Some aspects of the SOS response aystem - a critical survey. *Acta Biochimica Polonica*, 48(3):599–610, 2001.

[50] M. Kanehisa. *Post-genome Informatics.* Oxford University Press, New York, USA, 2000.

[51] S. A. Kauffman. *The Origins of Order, Self-Organization and Selection in Evolution.* Oxford University Press, USA, 1993.

[52] J. Kennedy and W. M. Spears. Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *International Conference on Evolutionary Computation (ICEC'98) Proceedings*, pages 78–83. IEEE, May 1998.

[53] B. N. Kholodenko, A. Kiyatkin, F. J. Bruggeman, E. Sontag, H. V. Westerhoff, and J. B. Hoek. Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *PNAS*, 99:12841–12846, 2002.

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

[55] S. Kim, S. Imoto, and S. Miyano. Dynamic bayesian network and nonparamtric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75:57–65, 2004.

[56] S. Kimura, M. Hatakeyama, and A. Konagaya. Inference of S-system models of genetic networks from noisy time-series data. *Chem-Bio Informatics Journal*, 4(1):1–14, 2004.

[57] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of S-system

models of genetic networks using cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154–1163, 2005.

[58] J. Kitagawa and H. Iba. *Evolutionary Computation in Bioinformatics*, chapter Identifying Metabolic Pathways and Gene Regulation Networks with Evolutionary Algorithms, pages 255–278. Morgan Kaufmann, 2002.

[59] H. Kitano, editor. *Foundations of Systems Biology*. The MIT Press, London, England, 2001.

[60] N. Krasnogor and J. Smith. A memetic algorithm with self-adaptive local search: TSP as a case study. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 987–994, 2000.

[61] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja. On learning gene regulatory networks under the boolean network model. *Machine Learning*, 52(1-2):147–167, 2003.

[62] B. Lewin. *Genes VIII*. Prentice Hall, New Jersy, 2003.

[63] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-clycle network is robustly designed. *PNAS*, 101(14):4781–4786, April 2004.

[64] Z. Li, S. M. Shaw, M. J. Yedwabnick, and C. Chan. Using a state-space model with hidden variables to infer transcription factor activities. *Bioinformatics*, 22(6):747–754, 2006.

[65] J. J. Liang and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *IEEE Congress on Evolutionary Computation*, pages 522–528, 2005.

[66] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific Symposium on Biocomputing 3*, pages 18–29, 1998.

[67] J. W. Little, S. H. Edmiston, L. Z. Pacelli, and D. W. Mount. Cleavage of the *Escherichia coli* lexA protein by the recA protease. *Proceedings of National Academy of Science (PNAS)*, 77(6):3225–3229, 1980.

[68] J. Liu and J. Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(6):448–642, 2005.

[69] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12(3):273–302, 2004.

[70] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, and Y. Eguchi. Inference of genetic network using the expression profile time course data of mouse P19 cells. In *Genome Informatics 13*, pages 382–383, December 2002.

[71] H. Matsuno, A. Doi, and M. Nagasaki. Hybrid petri net representation of genetic regulatory network. In *Pacific Symposium on Biocomputing 5*, pages 338–349, 2000.

[72] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano. Biopathway representation and simulation on hybrid functional petri net. *Silico Biology*, 3(3):389–404, 2003.

[73] M. M. Meysenburg and J. A. Foster. The quality of pseudorandom number generators and simple genetic algorithm performance. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 276–281, 1997.

[74] M. M. Meysenburg and J. A. Foster. Random generator quality and GP performance. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1121–1126, 1999.

[75] A. Mimura and H. Iba. Inference of a gene regulatory network by means of interactive evolutionary computing. *Information Sciences*, 145(3-4):225–236, September 2002.

[76] E. Mjolsness. *Computational Modeling of Genetic and Biochemical Networks*, chapter Trainable Gene Regulation Networks with Application to *Drosophila* Pattern Formation, pages 101–117. MIT Press, England, 2001.

[77] R. Morishita, H. Imade, I. Ono, N. Ono, and M. Okamoto. Finding multiple solutions based on an evolutionary algorithm for inference of genetic networks by S-system. In *Proceedings of Congress on Evolutionary Computations*, pages 615–622, December 2003.

[78] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. In *Technical Report 826, Caltech Concurrent Computation Program. California Institute of Technology, Pasadena, California*, 1989.

[79] P. Moscato and M. G. Norman. A "memetic" approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. In *Parallel Computing and Transputer Applications*, pages 177–186. IOS Press, 1992.

[80] K. P. Murphy and S. Mian. Modelling gene expression data using dynamic bayesian networks. Technical report, University of California at Berkeley, Department of Computer Science, 1999.

[81] M. Nakatsui, T. Ueda, I. Ono, and M. Okamoto. Efficient method for extracting common core binomial genetic interactions. In *Genome Informatics 16(2)*, page P148, 2005.

[82] K. Nasmyth. At the heart of the budding yeast cell cycle. *Trends in Genetics*, 12(10):405–412, December 1996.

[83] N. Noman and H. Iba. Accelerating differential evolution using an adaptive local search. submitted to IEEE Transaction on Evolutionary Computation.

[84] N. Noman and H. Iba. Inferring gene regulatory networks using differential evolution with local search heuristics. accepted in IEEE/ACM Transaction on Computational Biology and Bioinformatics.

[85] N. Noman and H. Iba. A memetic version of differential evolution algorithm for high dimensional optimization. submitted to Journal of Global Optimization.

[86] N. Noman and H. Iba. Enhancing differential evolution performance with local search for high dimensional function optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 967–974, June 2005.

[87] N. Noman and H. Iba. Inference of gene regulatory networks using S-system and differential evolution. In *Genetic and Evolutionary Computation Conference (GECCO) Proceedings*, pages 439–446, June 2005.

[88] N. Noman and H. Iba. Reverse engineering genetic networks using evolutionary computation. In *Genome Informatics 16(2)*, pages 205–214, December 2005.

[89] N. Noman and H. Iba. *Analysis of Biological Data: A Soft Computing Approach*, chapter Inferring Regulations in a Genomic Network from Gene Expression Profiles. World Scientific, 2006. in Press.

[90] N. Noman and H. Iba. Inference of genetic networks using S-system: Information criteria for model selection. In *Proceedings of the 2006 Conference on Genetic and Evolutionary Computation*, pages 263–270, July 2006.

[91] N. Noman and H. Iba. A new generation alternation model for differential evolution. In *Proceedings of the 2006 Conference on Genetic and Evolutionary Computation*, pages 1265–1272, July 2006.

[92] N. Noman and H. Iba. On the reconstruction of gene regulatory networks from noisy expression profiles. In *Proceedings of the World Congress on Computational Intelligence 2006*, pages 8712–8719, July 2006.

[93] I. M. Ong, J. D. Glasner, and D. Page. Modelling regulatory pathways in *E. coli* from time series expression profiles. *Bioinformatics*, 18(S1):241–248, 2002.

[94] Y.-S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, 2004.

[95] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong. Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions On Systems, Man and Cybernetics - Part B*, 36(1):141–152, 2006.

[96] I. Ono, H. Kita, and S. Kobayashi. *Advances in Evolutionary Computing*, chapter A Real-Coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover, pages 213–237. Springer, Jan 2003.

[97] S. Ott, S. Imoto, and S. Miyano. Finding optimal models for small gene networks. In *Pacific Symposium on Biocomputing 9*, pages 557–567, 2004.

[98] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. dAlchéBuc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(Suppl. 2):ii138–ii148, 2003.

[99] C. G. Peter Spirtes and R. Scheines. *Causation, Prediction, and Search.* MIT Press, USA, 2000.

[100] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipies in C, second edition.* Cambridge University Press, 1995.

[101] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization.* Springer, Berlin, Heidelberg, 2005.

[102] C. Rangel, J. Angus, F. Falciani, Z. Ghahramani, M. Lioumi, and D. Wild. Modeling T-cell activation using gene expression profiling and state space models. *Bioinformatics*, 20(9):1362–1372, 2004.

[103] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations of metabolic pathways. In *1st Int. Conf. on Intelligent Systems for Molecular Biology (ISMB '93)*, pages 328–336, 1993.

[104] R. J. Reece. *Analysis of Genes and Genomes.* John Wiley & Sons, USA, 2004.

[105] 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. *Proceedings of National Academy of Science (PNAS)*, 99(16):10555–10560, 2002.

[106] J. Rönkkönen, S. Kukkonen, and K. Price. Real-parameter optimization with differential evolution. In *Proceedings of 2005 IEEE Congress on Evolutionary Computation*, pages 506–513, September 2005.

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

[108] R. Salmon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *Biosystems*, 39(3), 1996. 263–278.

[109] H. Satoh, M. Yamamura, and S. Kobayashi. Minimal generation gap model for GAs considering both exploration and exploitation. In *Proceedings of IIZUKA'96*, pages 494–497, 1996.

[110] M. A. Savageau. Biochemical systems analysis. I. some mathematical properties of the rate law for the component enzymatic reactions. *Theoretical Biology*, 25:365–369, 1969.

[111] M. A. Savageau. Biochemical systems analysis. II. the steady-state solutions for an n-pool system using a power-law approximation. *Theoretical Biology*, 25:370–379, 1969.

[112] M. A. Savageau. *Biochemical Systems Analysis. A Study of Function and Design in Molecular Biology.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1976.

[113] M. A. Savageau. Power-law formalism: A canonical nonlinear approach to modeling and analysis. In *World Congress of Nonlinear Analysts*, pages 3323–3334, 1992.

[114] M. A. Savageau. Power-law formalism: A canonical nonlinear approach to modeling and analysis. In *World Congress of Nonlinear Analysts, 92, Vol.4*, pages 3323–3334, 1996.

[115] M. Schena, editor. *DNA Microarrays: A Practical Approach.* Oxford University Press, 1999.

[116] L. A. Soinov, M. A. Krestyaninova, and A. Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology*, 4, 2003. Article R6.

[117] R. Somogyi and C. A. Sniegoski. Modeling the complexity of genetic networks: Understanding multigenic and pleitropic regulation. *Complexity*, 1(6):45–63, 1996.

[118] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces Cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, December 1998.

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

[120] C. Spieth, F. Streichert, J. Supper, N. Speer, and A. Zell. Feedback memetic algorithms for modeling gene regulatory networks. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) 2005*, pages 61–67, 2005.

[121] J. Stark, D. Brewer, M. Barenco, D. Tomescu, R. Callard, and M. Hubank. Reconstructing gene networks: What are the limits? *Biochemical Society Transactions*, 31:1519–1525, 2003.

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

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

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

[125] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. In *Technical Report, Nanyang Technological University, Singapore, And KanGAL Report No 2005005, IIT Kanpur, India*, May 2005.

[126] N. Sugimoto and H. Iba. Inference of gene regulatory networks by means of dynamic differential bayesian networks and nonparametric regression. In *Genome Informatics 15(2)*, pages 121–130, 2004.

[127] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of National Academy of Science*, 96(6):2907–2912, 1999.

[128] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetic*, 22:281–285, 1999.

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

[130] K.-Y. Tsai and F.-S. Wang. Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, 21(7):1180–1188, 2005.

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

[132] E. P. van Someren, L. F. A. Wessels, and M. J. T. Reinders. Linear modeling of genetic networks from experimental data. In *In Intelligent Systems For Molecular Biology (ISMB 2000)*, pages 355–366, 2000.

[133] E. O. Voit and J. S. Almeida. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20(11):1670–1681, 2004.

[134] E. O. Voit and T. Radivoyevitch. Biochemical systems analysis of genome-wide expression data. *Bioinformatics*, 16:1023–1037, 2000.

[135] G. C. Walker. Mutagenesis and inducible responses to deoxyribonucleic acid damage in *Escherichia coli*. *Microbiology and Molecular Biology Reviews*, 48:60–93, 1984.

[136] G. C. Walker. Inducible DNA repair systems. *Annu. Rev. Biochem.*, 54:425–457, 1985.

[137] S.-C. Wang and S.-P. Li. *Lecture Notes in Computer Science, Volume 3141*, chapter Learning Bayesian Networks by Lamarckian Genetic Algorithm and Its Application to Yeast Cell-Cycle Gene Network Reconstruction from Time-Series Microarray Data, pages 49–62. Springer, 2004.

[138] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. Barker, and R. Somogyi. Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of National Academy of Science*, 95(1):334–339, 1998.

[139] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 1997. 67–82.

[140] F.-X. Wu, G. Poirier, and W. Zhang. Inferring gene regulatory networks with time delays using a genetic algorithm. *IEE Proceedings - Systems Biology*, 152(2):67–74, 2005.

[141] F.-X. Wu, W. Zhang, and A. Kusalik. Modeling gene expression from microarray expression data with state-space equations. In *Pacific Symposium on Biocomputing 9*, pages 581–592, 2004.

[142] F.-X. Wu, W. Zhang, and A. Kusalik. State-space model for gene regulatory networks with time delays. In *IEEE Computational Systems Bioinformatics Conference (CSB 2004)*, pages 454–455, 2004.

[143] N. Yalamanchili, D. Zak, B. Ogunnaike, J. Schwaber, A. Kriete, and B. Kholodenko. Quantifying gene network connectivity in silico: scalability and accuracy of a modular approach. *Systems Biology, IEE Proceedings*, 153(4):236–246, 2006.

[144] R. Yamaguchi and T. Higuchi. Comparative study of state space models for time-course gene expression data of yeast. In *Genome Informatics 16(2)*, page P151, 2005.

[145] J.-M. Yang and C.-Y. Kao. Integrating adaptive mutations and family competition into genetic algorithms as function optimizer. *Soft Computing*, 4:89–102, 2000.

[146] X. Yao, Y. Liu, and G. Liu. Evolutionary programming made faster. *IEEE Trans. on Evolutionary Computation*, 3(2):82–102, 1999.

[147] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[148] R. Yoshida, S. Imoto, T. Higuchi, and S. Miyano. State space model with markov switching for estimating time-dependent gene regulatory networks from time series microarray experiments. In *Genome Informatics 16(2)*, page P035, 2005.

[149] B. Yuan and M. Gallagher. On building a principled framework for evaluating and testing evolutionary algorithms: A continuous landscape generator. In *Congress on Evolutionary Computation (CEC2003)*, pages 451–458, 2003.

[150] D. Zaharie. Critical values for the control parameters of differential evolution algorithms. In *Proceedings of MENDEL 2002, 8th International Conference on Soft Computing*, pages 62–67, 2002.

List of Publications

Journal Paper(s)

J1. Nasimul Noman and Hitoshi Iba, "Inferring Gene Regulatory Networks using Differential Evolution with Local Search Heuristics", IEEE/ACM Transaction on Computational Biology and Bioinformatics, accepted for future publication.

J2. Nasimul Noman and Hitoshi Iba, "Accelerating Differential Evolution using an Adaptive Local Search", IEEE Transactions on Evolutionary Computation, accepted for future publication.

Book Chapter(s)

B1. Nasimul Noman and Hitoshi Iba, "Inferring Regulations in a Genomic Network from Gene Expression Profiles", in S. Bandyopadhyay, U. Maulik and J. T. L. Wang eds., *"Analysis of Biological Data: A Soft Computing Approach"*, World Scientific, (in press).

Conference Paper(s)

C1. Nasimul Noman and Hitoshi Iba, "On the Reconstruction of Gene Regulatory Networks from Noisy Expression Profiles", *World Congress on Computational Intelligence*, 2006, pp 8712-8719.

C2. Nasimul Noman and Hitoshi Iba, "Inference of Genetic Networks using S-system: Information Criteria for Model Selection", *Genetic and Evolutionary Computation Conference*, 2006, pp 263-270.

C3. Nasimul Noman and Hitoshi Iba, "A New Generation Alternation Model for Differential Evolution", *Genetic and Evolutionary Computation Conference*, 2006, pp. 1265-1272.

C4. Nasimul Noman and Hitoshi Iba, "Reverse Engineering Genetic Networks Using Evolutionary Computation", *Genome Informatics*, Vol. 16, No. 2, 2005, pp. 205-214.

C5. Nasimul Noman and Hitoshi Iba, "Inference of Gene Regulatory Networks Using S-system and Differential Evolution", *Genetic and Evolutionary Computation Conference*, 2005, pp. 439-446.

C6. Nasimul Noman and Hitoshi Iba, "Enhancing Differential Evolution Performance with Local Search for High Dimensional Function Optimization", *Genetic and Evolutionary Computation Conference*, 2005, pp. 967-974.

C7. Nasimul Noman, Kouichi Okada, Naoki Hosoyama and Hitoshi Iba, "Use of Clustering to Improve the Layout of Gene Network for Visualization", *Congress on Evolutionary Computation*, 2004, pp. 2068-2075.

C8. Naoki Hosoyama, Nasimul Noman and Hitoshi Iba, "Layout Search of a Gene Regulatory Network for 3-D Visualization", *The Fourteenth Workshop on Genome Informatics (GIW)*, 2003, pp. 104-113.

C9. Nasimul Noman, Md. Nurul Huda and M. Lutfar Rahman, "Simulation Study on the Suitability of DnP Scheme for Real-time Applications", *International Conference on Fundamentals of Electronics, Communications and Computer Sciences (ICFS)*, March, 2002, Tokyo (PP. 10-11 10-16).

C10. Nasimul Noman, Md. Nurul Huda and M. Lutfar Rahman, "An Improved Buffer Sharing Scheme for Shared Memory ATM switch with Multipriority Traffic", *International Conference on Computer and Information Technology (ICCIT)*, December, 2001, pp. 25-29.

C11. Nasimul Noman, Md. Nurul Huda and M. Lutfar Rahman, "Simulation Study of The Delay Performance of Dynamic Threshold with Native Pushout (DnP) Scheme under Bursty Traffic", *International Conference on Computer and Information Technology (ICCIT)*, December, 2001, pp. 42-46.

Poster(s)

P1. Kouichi Okada, Nasimul Noman and Naoki Hosoyama and Hitoshi Iba, "3-D Visualization Software of a Gene Regulatory Network", *The Fourteenth*

Workshop on Genome Informatics (GIW), 2003, pp. 386-387.

Other(s)

O1. Md. Julius Hossain, Md. Nurul Huda and Nasimul Noman, "BBPO - A New Buffer Sharing Scheme for ATM Networks", Dhaka University Journal of Science, Dhaka, Bangladesh, January 2003, pp 83-90.