Probabilistic Boolean Networks: Models, Algorithms and Applications

Wai-Ki CHING*
Department of Mathematics
The University of Hong Kong

Abstract

We consider modeling genetic regulatory networks by using Boolean Networks (BNs) and its extension Probabilistic Boolean Networks (PBNs). BNs (deterministic models) and PBNs (stochastic models) are useful and effective tools for studying genetic regulatory networks. A PBN is essentially a collection of BNs driven by a Markov chain (a random process). A BN is characterized by its attractor cycles and a PBN is characterized by its steady-state distribution. We review some algorithms for finding attractor cycles and steady-state distributions for BNs and PBNs, respectively. We then discuss an inverse problem, the problem of constructing a PBN given a set of BNs. It is well-known that the control of a genetic regulatory network is useful for avoiding undesirable states associated with diseases and this results in a control problem. We formulate both problems as optimization problems and efficient algorithms are also presented to solve them. Other applications of PBNs will also be discussed.

*Research Support in Part by HK RGC Grant and HKU Strategic Research Theme Fund on Computation and Information.

The Outline

(1) Boolean Networks (BNs)

(2) Probabilistic Boolean Networks (PBNs)

(3) The Construction Problem.

(4) The Control Problem.

(5) Concluding Remarks and Related Problems.

1. Boolean Networks and Probabilistic Boolean Networks.
1.1 Motivations and Objectives.

• An important issue in **systems biology** is to model and understand the **mechanism** in which the cells execute and control a large number of operations for their **normal functions** and also the way in which they fail in **diseases** such as cancer (25000 genes in human Genome). Eventually to design some **control strategy** to avoid the **undesirable** state/situation.

• **Genetic Network:** A set of molecular components such as genes, proteins and other molecules, and interactions between them that collectively carry out some cellular function.

• **Gene Expression:** The regulation of gene transcription. The expression of a gene may be controlled during RNA processing and transport, RNA translation, and the post-translational modification of proteins.

- **Mathematical Modeling** and **Computational Study**.

(i) Gaining an understanding of the complex patterns of behavior from the interactions between genes poses a huge scientific challenge with potentially **high industrial payoffs**.

(ii) **Massive data** and **advanced experimental techniques**, a computational approach is required.

(iii) Formal methods for the **modeling and simulation** of gene regulation processes are indispensable (**large and complex genetic regulatory system**).

• A Review of Mathematical Models (De Jong, 2002).

-**Boolean Networks (BNs) (Kaufman, 1969a, 1969b, 1974, 1993)**
-Differential and Partial Differential Equations (Mestl et al., 1995)
-**Probabilistic Boolean networks (PBNs) (Shmulevich et al., 2002)**
-Bayesian Networks (Friedman et al., 2000)
-Linear Models (van Someren et al., 2000)
-Petri Nets (Steggles et al. 2007)
. . . .

• Since genes exhibit "**switching behavior**", **BNs and PBNs** models have received much attention. We shall focus on these models.

1.1.1 The Lac Operon

• The **lac operon** * consists of a promoter/operator region and three structural genes: lacZ (Gene for β-galactosidase), lacY (Gene for β-galactoside permease), and lacA (Gene for β-galactoside transacetylase).

• **lac operon** is required for the **transport and metabolism of lactose in E. coli** (**Lactose** $\rightarrow$ **Glucose**). The **lac operon** is regulated by several factors including the availability of **Glucose** and **Lactose**.

Lactose	**Glucose**	**Expression**
Low	High	Off
Low	Low	Off
High	High	Off
High	**Low**	**On**

*lac operon is a well-known example of an inducible genetic circuit. It has been using as a model for understanding gene regulations since late 1950s. An operon is a cluster of structural genes that are expressed as a group and their associated promoter and operator. lacZ converts lactose into glucose and galactose and lacY transports lactose into the cell.

1.2 Boolean Networks

• In a BN, each **gene** is regarded as a **vertex** of the network and is then quantized into **two levels** only (**expressed: 1 or unexpressed: 0**) though the idea can be extended to the case of more than two levels.

• The **target gene** is predicted by several genes called its **input genes** via a **Boolean function**.

• If the input genes and the corresponding Boolean functions are given, a BN is said to defined and it can be considered as a **deterministic dynamical system**.

• The only **randomness** involved in the network is the **initial system state**.

1.2.1 An Example of a BN of Three Genes

$$v_i(t+1) = f^{(i)}(v_1(t), v_2(t), v_3(t)), \quad i = 1, 2, 3.$$

State	$v_1(t)$	$v_2(t)$	$v_3(t)$	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$
1	0	0	0	0	1	1
2	0	0	1	1	0	1
3	0	1	0	1	1	0
4	0	1	1	0	1	1
5	1	0	0	0	1	0
6	1	0	1	1	0	0
7	1	1	0	1	0	1
8	1	1	1	1	1	0

$$(0,0,0) \rightarrow (0,1,1) \leftrightarrow (0,1,1),$$

$$(1,0,1) \rightarrow (1,0,0) \rightarrow (0,1,0) \rightarrow (1,1,0) \rightarrow (1,0,1),$$

$$(0,0,1) \rightarrow (1,0,1), \qquad (1,1,1) \rightarrow (1,1,0).$$

- The **transition probability matrix** of the 3-gene BN is given by

$$A_3 = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left(\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{array}.$$

- We note that each column has **only one non-zero element** and **column sum is one**. Thus the matrix is very **sparse**.

• Given an initial state, the system will eventually enters into a set of stable states denoted as **attractor cycle**. In addition, **basin of attractor** is defined as those set of states leading the system to a specific attractor cycle.

• Since each attractor may consist of one or many states, an attractor containing only one state is called a **singleton attractor**. Otherwise, if an attractor contains p **states**, it is called an **attractor cycle** of **period** p.

• There are 2 interpretations for the biological function of attractors.

- The first one follows Kauffman is that one attractor should correspond to a **cell type** (Kauffman 1993).

- The second interpretation is that they correspond to different **cell states**: growth, differentiation (liver cell, nerve cell etc.) and apoptosis (Huang 1999).

1.2.2 A Review on BNs

- A BN $G(V, F)$ consists of **a set of vertices** $V = \{v_1, v_2, \ldots, v_n\}$. We define $v_i(t)$ to be the state (0 or 1) of the vertex v_i at time t.

- There is also **a list of Boolean functions** ($f_i : \{0,1\}^n \to \{0,1\}$): $F = \{f_1, f_2, \ldots, f_n\}$ to represent the rules of the regulatory interactions among the genes:

$$\boxed{v_i(t+1) = f_i(\mathbf{v}(t)), \quad i = 1, 2, \ldots, n}$$

where

$$\mathbf{v}(t) = (v_1(t), v_2(t), \ldots, v_n(t))^T$$

is called the **Gene Activity Profile** (GAP).

- The GAP can take any possible form (states) from the set

$$S = \{(v_1, v_2, \ldots, v_n)^T : v_i \in \{0,1\}\}$$

and thus totally there are 2^n possible states.

1.2.3 Finding Attractor Cycles in BNs

- Finding a singleton attractor can be a difficult task (NP-hard) (Akutsu et al. 1998).

- An algorithm is developed for identifying **singleton attractors** and **short period attractors** in BNs (Zhang et. al 2007a).

Results:

-We show that the algorithm for finding singleton attractors works in $O(1.19^n)$ time for $K = 2$, which is much faster than the naive $O(2^n)$ time algorithm. Here n is the number of genes and K is the maximum indegree.

-We also show that finding an attractor with the shortest period is NP-hard.

2.1 Probabilistic Boolean Networks (PBNs)

• Since BN is a **deterministic model**, to overcome this deterministic rigidity, an extension to a probabilistic model is natural.

• Reasons for a **stochastic model**:

- The biological system has its **stochastic** nature;

- The microarray data sets used to infer the network structure are usually not accurate because of the **experimental noise** in the complex measurement process.

• A probabilistic model, Probabilistic Boolean Networks (PBNs), was proposed by Shmulevich et al. (2002a), (2002b), (2002c), (2002d). Using PBN, they are able to construct a small size gene network using gene expression data, Shmulevich et al. (2003).

• For each vertex v_i in a PBN, instead of having only one Boolean function as in BN, there are a number of Boolean functions (**predictor functions**)

$$\boxed{f_j^{(i)} (j = 1, 2, \ldots, l(i))}$$

to be chosen for determining the state of gene v_i.

• The probability of choosing $f_j^{(i)}$ as the **predictor function** is

$$\boxed{c_j^{(i)}, 0 \le c_j^{(i)} \le 1 \quad \text{and} \quad \sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad \text{for} \quad i = 1, 2, \ldots, n.}$$

• The probability $c_j^{(i)}$ can be estimated by using the method of **Coefficient of Determination** (COD) (Dougherty et al. (2000)) with real gene expression data sets.

Remark:

-CoD is a measure of **relative decrease in error** from estimating transcriptional levels of a target gene through the levels of its predictor genes in the absence of the predictor genes. The CoDs obtained are then translated to the predictor probabilities.

-However, the maximum number of possible predictors for each gene and the number of their corresponding probabilities is equal to $O(2^{2^n})$ where n is the number of nodes. This means the number of parameters in the PBN model is $O(n2^{2^n})$.

-The major drawback of using CoD is the **model complexity** and **imprecisions** due to **insufficient data sample size**. Very often, we may have to impose some constraints on the maximum size of predictors for each gene.

• If we let f_j be the jth possible realization,

$$f_j = (f^{(1)}_{j_1}, f^{(2)}_{j_2}, \ldots, f^{(n)}_{j_n}), \quad 1 \le j_i \le l(i), \quad i = 1, 2, \ldots, n.$$

In an **independent PBN** (the selection of the Boolean function is independent), the probability of choosing the jth BN q_j is

$$\boxed{q_j = \prod_{i=1}^{n} c^{(i)}_{j_i}, \quad 1, 2, \ldots, N.} \tag{1}$$

• The maximum number of different possible realizations of BNs is

$$\boxed{N = \prod_{i=1}^{n} l(i).} \tag{2}$$

• There are other types of PBNs such as context-sensitive PBNs and instantaneously random PBNs (Pal et al., 2005). Here we only focus on **independent PBNs**.

• We note that the **transition process** among the states in the set S is a **Markov chain process**. Let $\mathbf{a}$ and $\mathbf{b}$ be any two column vectors in the set S. Then the transition probability

$$
\begin{aligned}
&\text{Prob } \{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}\} \\
= &\sum_{j=1}^{N} \text{Prob } \{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}, \text{the } j\text{th network is selected } \} \cdot q_j.
\end{aligned}
$$

• By letting $\mathbf{a}$ and $\mathbf{b}$ to take all the possible states in S, one can get the transition probability matrix for the process. The transition matrix is given by

$$\boxed{A = q_1 A_1 + q_2 A_2 + \cdots + q_N A_N.}$$

Here A_j is the corresponding transition probability matrix of the j-th BN.

• There are at most $N2^n$ nonzero entries for the transition probability matrix A.

2.2 Computation of the Steady-state Distribution of a PBN

• Evolution of the network is according to the transition probability matrix. Under some condition, the network will become "**steady**". To find the steady-state distribution (long run behavior) is a key aspect in studying the dynamics of genetic regulatory networks. The matrix size is **large** and **sparse**, i.e., there are many zero entries.

Results:

-An **efficient matrix method** (Zhang et al. 2007b) was proposed to **construct** the sparse transition probability matrix, and **power method** based on the sparse matrix-vector multiplication is applied to compute the steady-state probability distribution.

-The method provides a way for studying the sensitivity of steady-state distribution to the influence of different genes and BNs.

2.3 Approximation of the Steady-state distribution of a PBN

• A **matrix approximation method** for computing the steady-state distribution of a n-**gene** PBN was proposed (Ching et al. 2007).

• The idea is to **neglect** some Boolean networks (BNs) with very **slim selection probabilities**, say less than a **threshold value** τ during the construction of the transition probability matrix.

Results:

- We gave an error analysis of this approximation method and a theoretical result on the distribution of BNs in a PBN with at most **two Boolean functions** (random selection probability) for one gene.

-The selection probability distribution of a BN is given by

$$\boxed{\frac{(-\ln(\tau))^{n-1}}{(n-1)!} \quad \text{for} \quad \tau \in (0,1).}$$

2.4 Parsimonious Multivariate Markov Chain Model for PBNs

• A parsimonious multivariate Markov model was proposed (Ching et al. 2005) for approximating PBNs that can describe the dynamics of a genetic network using gene expression sequences.

Results:

-The parsimonious model is able to preserve the strength of PBNs and reduce the complexity of the networks. The number of parameters of the model is $O(n^2)$ where n is the number of genes involved.

-Efficient **estimation methods** for solving the model parameters were given. Numerical examples on synthetic data sets and practical yeast data sequences (Yeung and Ruzzo, 2001) are given to demonstrate the effectiveness of the proposed model.

2.5 Finding Attractor Cycles in a PBN

• We studied the expected number of singleton attractors in an independent PBN and designed algorithms for identifying singleton and small attractor cycles (Hayashida et al., 2009).

Results:

-The **expected number** is $(2 - (1/2)^{L-1})^n$, where n is the number of nodes in a PBN and L is the number of Boolean functions assigned to each node. In the case of $L = 2$, this number can be simplified to 1.5^n.

-The **average case time complexities** for identifying singleton attractors of a PBN with $L = 2$ and $L = 3$ are $O(1.601^n)$ and $O(1.763^n)$, respectively. The results of computational experiments suggest that these algorithms are much more efficient than the naive algorithm that examines all possible $O(2^n)$ states.

3. The Construction Problem

3.1 The Motivation

• We consider an inverse problem of constructing a PBN from a given **steady-state distribution** (Zhang et al., 2010) or a **transition probability matrix** (Chen et al. 2011).

• Such problems are very useful for network inference from steady-state data, as most microarray data sets are assumed to be obtained from sampling the steady-state data or time-series data.

• This is **an inverse problem** of huge problem size. The inverse problem is **ill-posed**, meaning that there will be many networks or no network having the desirable properties.

• Here we focus on constructing a PBN based on a given **transition probability matrix** and **a given set of BNs** (Chen et al. 2011).

3.2 The Problem Formulation

• Suppose that the possible BNs constituting the PBN are known and their BN matrices are denoted by

$$\{A_1, A_2, \ldots, A_N\}.$$

• Transition probability matrix is observed and they are related as follows:

$$A = \sum_{i=1}^{N} q_i A_i. \tag{3}$$

• We are interested in getting the **parameters q_i (selection probabilities)**, $i = 1, 2, \ldots, N$ when A is given.

• The problem size is huge and A is usually **sparse**. Here we assume that each column of A has at most m non-zero entries. In this case, we have $N = m^{2^n}$ and we can order $A_1, A_2, \cdots, A_{m^{2^n}}$ systematically.

• We note that q_i and A_i are non-negative and there are only $m \cdot 2^n$ non-zero entries in A. Thus we have at most $m \cdot 2^n$ equations for m^{2^n} unknowns.

• To reconstruct the PBN, one possible way to get q_i is to consider the following minimization problem:

$$\boxed{\begin{array}{l} \qquad\qquad \min_{\mathbf{q}} \left\| A - \sum_{i=1}^{m^{2^n}} q_i A_i \right\|_F^2 \\ \text{subject to} \\ \qquad\qquad 0 \le q_i \le 1 \quad \text{and} \quad \sum_{i=1}^{m^{2^n}} q_i = 1. \end{array}} \tag{4}$$

- Here $\|\cdot\|_F$ is the **Frobenius norm** of a matrix. We define a mapping F from the set of $l \times l$ square matrices to the set of $l^2 \times 1$ vectors by

$$F\left(\begin{pmatrix} a_{11} & \cdots & a_{1l} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ a_{l1} & \cdots & a_{ll} \end{pmatrix}\right) = (a_{11}, \ldots, a_{l1}, a_{12}, \ldots, a_{l2}, \ldots, \ldots, a_{1l}, \ldots a_{ll})^T. \tag{5}$$

- If we let

$$U = [F(A_1), F(A_2), \ldots, F(A_{m^{2^n}})] \quad \text{and} \quad \mathbf{p} = F(A) \tag{6}$$

then the minimization problem in Eq. (4) becomes

$$\boxed{\begin{array}{l} \qquad\qquad \min_{\mathbf{q}} \|U\mathbf{q} - \mathbf{p}\|_2^2 \\ \text{subject to} \\ \qquad\qquad 0 \le q_i \le 1 \quad \text{and} \quad \sum_{i=1}^{m^{2^n}} q_i = 1. \end{array}} \tag{7}$$

- Since

$$||U\mathbf{q}-\mathbf{p}||_2^2 = (U\mathbf{q}-\mathbf{p})^T(U\mathbf{q}-\mathbf{p}) \tag{8}$$

we have

$$\min_{\mathbf{q}} ||U\mathbf{q}-\mathbf{p}||_2^2 = \min_{\mathbf{q}}\{\mathbf{q}^T U^T U\mathbf{q} - 2\mathbf{q}^T U^T\mathbf{p} + \mathbf{p}^T\mathbf{p}\}. \tag{9}$$

- Minimization problem (9) **without constraints** is equivalent to

$$\boxed{\min_{\mathbf{q}}\{\mathbf{q}^T U^T U\mathbf{q} - 2\mathbf{q}^T U^T\mathbf{p}\}.} \tag{10}$$

- The matrix U^TU is a **symmetric positive semi-definite matrix**. Solving minimization problem (10) without constraints is equivalent to solving the linear system

$$\boxed{U^TU\mathbf{q} = U^T\mathbf{p}} \tag{11}$$

with the **Conjugate Gradient (CG) method**. The CG method yields different solutions with different initial guesses. Therefore there are **many solutions** for our problem.

3.3 The Maximum Entropy Approach

• This method can give a solution of the inverse problem. But usually there are **too many solutions**. Extra constraint or criterion has to be introduced in order to narrow down the set of solutions or even a unique solution.

• In (Chen et al., 2011) it was proposed to consider the solution which gives the **largest Entropy** as $\mathbf{q}$ itself can be considered as a probability distribution.

• This means we are to find $\mathbf{q}$ such that it maximizes

$$\boxed{-\sum_{i=1}^{m2^n} q_i \log(q_i).} \tag{12}$$

• The Entropy attains its **maximum** when $\mathbf{q}$ is the **uniform distribution**.

• We recall that for the inverse problem, we have $m \cdot 2^n$ equations for m^{2^n} unknowns. Thus one may have infinitely many solutions.

• Since $\mathbf{q}$ can be viewed as a probability distribution, one possible way to get a better choice of q_i is to consider maximizing the entropy of $\mathbf{q}$ subject to the given constraints, i.e., the following maximization problem:

$$\boxed{\begin{array}{ll} & \max_{\mathbf{q}} \left\{ \sum_{i=1}^{m^{2^n}} (-q_i \log q_i) \right\} \\ \text{subject to} & \\ & \bar{U}\mathbf{q} = \bar{\mathbf{p}} \quad \text{and} \quad 0 \le q_i \quad i = 1, 2, \ldots, m^{2^n}. \end{array}} \tag{13}$$

• We remark that the **constraints that $q_i \le 1$ can be discarded** as we required that

$$\sum_{i=1}^{m^{2^n}} q_i = 1 \quad \text{and} \quad 0 \le q_i \quad i = 1, 2, \ldots, m^{2^n}.$$

- The dual problem of (13) is of the type

$$\min_{\mathbf{y}} \max_{\mathbf{q}} L(\mathbf{q}, \mathbf{y}) \tag{14}$$

where $\mathbf{y}$ is the **multiplier** and $L(\cdot, \cdot)$ is the **Lagrangian function**

$$L(\mathbf{q}, \mathbf{y}) = \sum_{i=1}^{m^{2^n}} (-q_i \log q_i) + \mathbf{y}^T (\bar{\mathbf{p}} - \bar{U}\mathbf{q}). \tag{15}$$

- The optimal solution $\mathbf{q}^*(\mathbf{y})$ of the inner maximization problem of (14) solves the equations

$$\nabla_{q_i} L(\mathbf{q}, \mathbf{y}) = -\log q_i - 1 - \mathbf{y}^T \bar{U}_{\cdot i} = 0, \quad i = 1, 2, \ldots, m^{2^n}$$

and is thus of the form:

$$q_i^*(\mathbf{y}) = e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}}, \quad i = 1, 2, \ldots, m^{2^n} \tag{16}$$

where $\bar{U}_{\cdot i}$ is the ith column of the matrix $\bar{U}$.

• After substituting $\mathbf{q}^*(\mathbf{y})$ back into (15) the **dual problem** (14) can be simplified to

$$\boxed{\min_{\mathbf{y}} \left\{ \sum_{i=1}^{m^{2^n}} e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}} + \mathbf{y}^T \bar{\mathbf{p}} \right\}.} \tag{17}$$

• The solution of the primal problem (14) is obtained from the solution of the **dual problem** (16) through Problem (17).

• Thus we have transformed a **constrained maximization problem with m^{2^n} variables** into an **unconstrained minimization problem of $m \cdot 2^n + 1$ variables**.

• We will then apply **Newton's method** in conjunction with **Conjugate Gradient (CG) method** to solving the dual problem.

3.4 Newton's Method

- We denote

$$\boxed{f(\mathbf{y}) = \sum_{i=1}^{m2^n} e^{-1-\mathbf{y}^T \bar{U}_{\cdot i}} + \mathbf{y}^T \bar{\mathbf{p}}} \qquad (18)$$

the function we want to minimize.

- In Newton's method, we have to compute the **gradient** and the **Hessian** of f. They are given, respectively, as follow:

$$\boxed{\nabla f(\mathbf{y}) = -\bar{U}\mathbf{q}^*(\mathbf{y}) + \bar{\mathbf{p}}} \qquad (19)$$

and

$$\boxed{\nabla^2 f(\mathbf{y}) = \bar{U} \cdot \mathrm{diag}(\mathbf{q}^*(\mathbf{y})) \cdot \bar{U}^T} \qquad (20)$$

where $\mathbf{q}^*(\mathbf{y})$ is as defined in Eq. (16) and diag$(\mathbf{q}^*(\mathbf{y}))$ is the diagonal matrix with diagonal entries $(\mathbf{q}^*(\mathbf{y}))$.

Newton's Method

```
Choose starting point $\mathbf{y}_0 \in Im(\bar{U})$
Initialize : $k = 1$;
    While $||\nabla f(\mathbf{y}_k)||_2 >$ tolerance
        Find $(\mathbf{p}_k)$ with $\nabla^2 f(\mathbf{y}_{k-1})\mathbf{p}_k = -\nabla f(\mathbf{y}_{k-1})$;
        Set $\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{p}_k$;
        Set $k = k + 1$;
    End.
```

• From Eq. (20), we observe that f is **strictly convex** on the subspace $Im(\bar{U})$.

• Newton's method will produce a sequence of points $\mathbf{y}_k$ according to the iteration $\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{p}_k$, where the Newton step $\mathbf{p}_k$ is the solution of the Hessian matrix system:

$$\nabla^2 f(\mathbf{y}_{k-1})\mathbf{p}_k = -\nabla f(\mathbf{y}_{k-1}). \tag{21}$$

• We note that $\nabla^2 f(\mathbf{y}_{k-1})$ is a **one-to-one mapping** of the concerned subspace onto itself.

• Moreover, from Eq. (19) $\nabla f(\mathbf{y}) \in Im(\bar{U})$ as we have $\bar{\mathbf{p}} \in Im(\bar{U})$ (from Eq. (13)). Hence, Eq. (21) has an unique solution and therefore Newton's method for minimizing f is well defined.

• If we start with $\mathbf{y}_0 \in Im(\bar{U})$, the Newton's sequence will remain in the subspace. It will **converge** locally at a **quadratic rate**.

• To enforce global convergence one may wish to resort to line search or trust region techniques. However, we did not find this necessary in our computational experiments.

3.5 Conjugate Gradient Method

• In each iteration of the Newton's method, one has to solve the linear system of the form in Eq. (21). We propose to employ the **Conjugate Gradient (CG) method**.

• The convergence rate of CG method depends on the **effective condition number**

$$\frac{\lambda_1(\nabla^2 f(\mathbf{y}))}{\lambda_s(\nabla^2 f(\mathbf{y}))} \tag{22}$$

of $\nabla^2 f(\mathbf{y})$. Since $\nabla^2 f(\mathbf{y})$ is **singular** we have to consider the second smallest eigenvalue $\lambda_s(\nabla^2 f(\mathbf{y}))$.

Results: For the Hessian matrix $\nabla^2 f(\mathbf{y})$, we have

$$2^n \cdot e^{-2(m\cdot 2^n+1)\cdot\|y\|_\infty} \leq \frac{\lambda_1(\nabla^2 f(\mathbf{y}))}{\lambda_s(\nabla^2 f(\mathbf{y}))} \leq \left(\sqrt{2^n}+\sqrt{m}\right)^2 \cdot e^{2(m\cdot 2^n+1)\cdot\|y\|_\infty}.$$

3.6 Some PBN Examples

• For Newton's method, we set the tolerance to be 10^{-7} while the tolerance of CG method is 10^{-10}.

Example 3.1 In the first example, we consider the case $n = 2$ and $m = 2$ and we suppose that the observed/estimated transition probability matrix of the PBN is given as follows:

$$A_{2,2} = \begin{pmatrix} 0.1 & 0.3 & 0.5 & 0.6 \\ 0.0 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.4 \end{pmatrix}. \qquad (23)$$

• Then there are 16 possible BNs for constituting the PBN and they are listed below:

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} A_3 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_6 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} A_7 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} A_8 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_9 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{10} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} A_{11} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{12} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{13} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{14} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} A_{15} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} A_{16} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

• Suppose we have

$$A = \sum_{i=1}^{16} q_i A_i$$

and the followings are the 8 equations governing q_i (cf. Eq. (6)):

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
\hline
1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\hline
1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\hline
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1
\end{array}\right)
\left(\begin{array}{c}
q_1\\ q_2\\ q_3\\ q_4\\ \hline q_5\\ q_6\\ q_7\\ q_8\\ \hline q_9\\ q_{10}\\ q_{11}\\ q_{12}\\ \hline q_{13}\\ q_{14}\\ q_{15}\\ q_{16}
\end{array}\right)
=
\left(\begin{array}{c}
0.1\\ 0.0\\ 0.0\\ 0.9\\ \hline 0.3\\ 0.7\\ 0.0\\ 0.0\\ \hline 0.5\\ 0.0\\ 0.5\\ 0.0\\ \hline 0.6\\ 0.0\\ 0.0\\ 0.4
\end{array}\right).$$

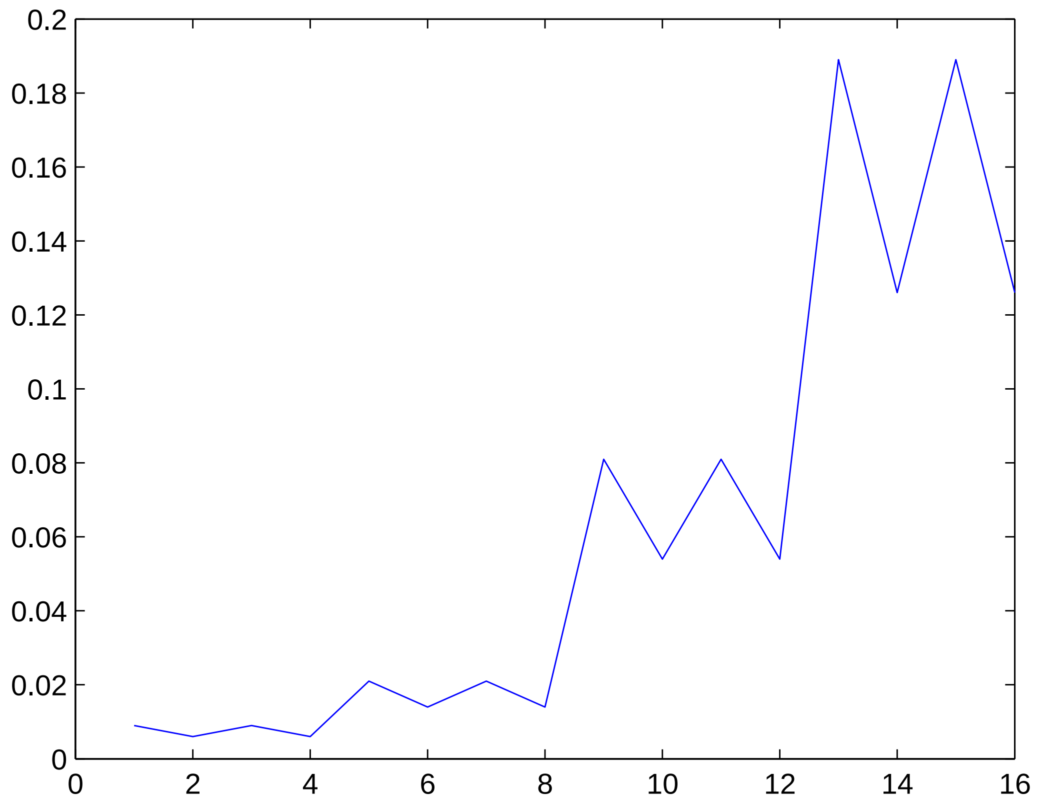


Fig. 3.1. The Probability Distribution q for the case of $A_{2,2}$.

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	1	1
2	0	1	0	1
3	1	0	0	0
4	1	1	0	0

Table 3.1: The Truth Table for A_{13}.

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	1	1
2	0	1	0	1
3	1	0	0	0
4	1	1	1	1

Table 3.2: The Truth Table for A_{14}.

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	1	1
2	0	1	0	1
3	1	0	1	0
4	1	1	0	0

Table 3.3 : The Truth Table for A_{15}.

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	1	1
2	0	1	0	1
3	1	0	1	0
4	1	1	1	1

Table 3.4 : The Truth Table for A_{16}.

Example 3.2 We then consider the case $n = 3$ and $m = 2$ and we suppose that the observed transition matrix of the PBN is given as follows:

$$A_{3,2} = \begin{pmatrix} 0.1 & 0.3 & 0.5 & 0.6 & 0.2 & 0.1 & 0.6 & 0.8 \\ 0.0 & 0.7 & 0.0 & 0.0 & 0.8 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.9 & 0.0 & 0.0 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.0 \end{pmatrix}.$$

• There are 256 possible BNs for constituting the PBN. The solution is shown in Figure 3.2. We note that the PBN is dominated (over 60%) by 25 BNs.

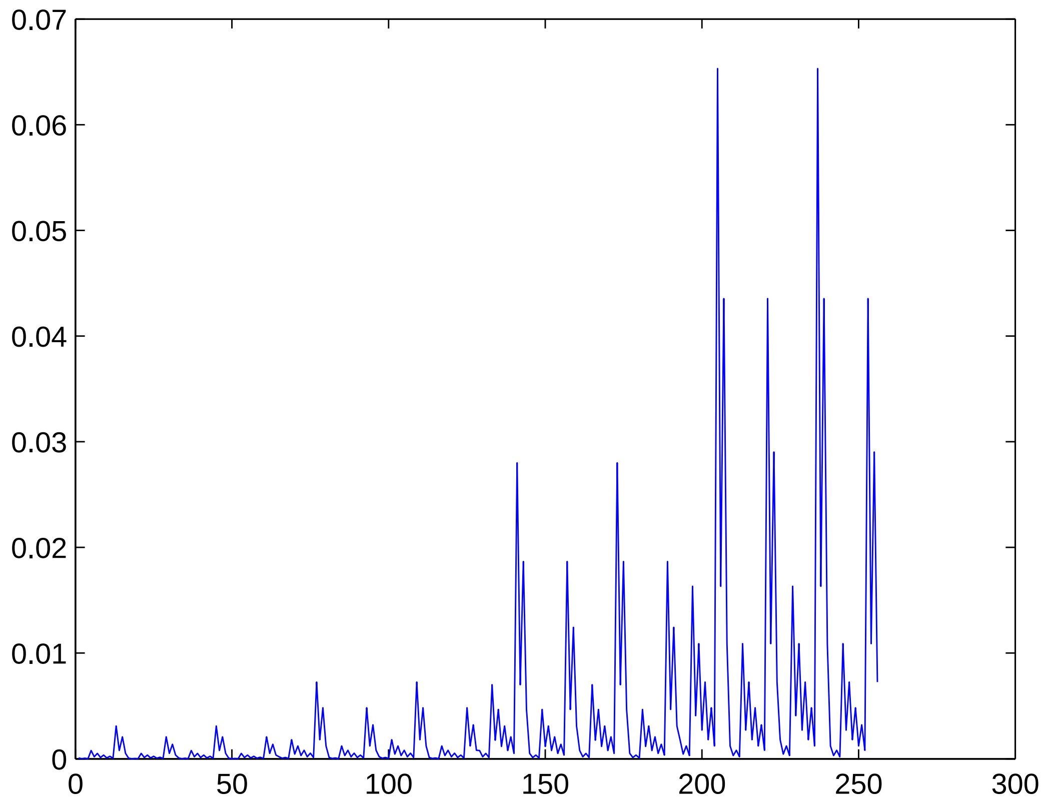


Fig. 3.2. The Probability Distribution q for the case of $A_{3,2}$.

Example 3.3 We consider a popular PBN (Shmulevich et al. (2002a)):

Network State	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

Table 3.5 : Truth Table (Taken from Shmulevich et al. (2002a)).

BN_1	1	7	7	6	3	8	6	8
BN_2	1	7	7	5	3	7	5	8
BN_3	1	7	7	2	3	8	6	8
BN_4	1	7	7	1	3	7	5	8

Table 3.6 : The Four BNs.

• We consider **adding some perturbations** to the **first two rows** and the **non-zeros entries** of the transition probability matrix $A_{4,4}$ as follows:

$$\begin{pmatrix}
1.0-\delta & \delta & \delta & 0.2+\delta & \delta & \delta & \delta & \delta \\
\delta & \delta & \delta & 0.2+\delta & \delta & \delta & \delta & \delta \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0-2\delta & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.3-\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 \\
0.0 & 0.0 & 0.0 & 0.3-\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 \\
0.0 & 1.0-2\delta & 1.0-2\delta & 0.0 & 0.0 & 0.5-\delta & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5-\delta & 0.0 & 1.0-2\delta
\end{pmatrix}.$$

• For $\delta = 0.01, 0.02, 0.03$ and 0.04, we apply our algorithm and obtain **16 major BNs** (out of 10368 BNs) (Table 3.7) and these BNs actually contribute, respectively, 94%, 90%, 84% and 79% of the network.

• We note that the 1st, 8th, 9th and the last major BNs match with the four BNs (BN_1, BN_2, BN_3, BN_4) in Table 3.7.

BNs									$\delta = 0.01$	$\delta = 0.02$	$\delta = 0.03$	$\delta = 0.04$
1*	1	7	7	1	3	7	5	8	0.047	0.045	0.042	0.040
2	1	7	7	1	3	7	6	8	0.047	0.045	0.042	0.040
3	1	7	7	1	3	8	5	8	0.047	0.045	0.042	0.040
4	1	7	7	1	3	8	6	8	0.047	0.045	0.042	0.040
5	1	7	7	2	3	7	5	8	0.047	0.045	0.042	0.040
6	1	7	7	2	3	7	6	8	0.047	0.045	0.042	0.040
7	1	7	7	2	3	8	5	8	0.047	0.045	0.042	0.040
8*	1	7	7	2	3	8	6	8	0.047	0.045	0.042	0.040
9*	1	7	7	5	3	7	5	8	0.071	0.067	0.063	0.059
10	1	7	7	5	3	7	6	8	0.071	0.067	0.063	0.059
11	1	7	7	5	3	8	5	8	0.071	0.067	0.063	0.059
12	1	7	7	5	3	8	6	8	0.071	0.067	0.063	0.059
13	1	7	7	6	3	7	5	8	0.071	0.067	0.063	0.059
14	1	7	7	6	3	7	6	8	0.071	0.067	0.063	0.059
15	1	7	7	6	3	8	5	8	0.071	0.067	0.063	0.059
16*	1	7	7	6	3	8	6	8	0.071	0.067	0.063	0.059

Table 3.7: The 16 Major BNs and their Selection Probabilities.

3.7 The performance of Newton's Method and CG Method

We also present the number of Newton's iterations required for convergence and the average number of CG iterations in each Newton's iteration in the following table.

n	m	Number of BNs	Newton's Iterations	Average Number of CG Iterations
2	2	16	9	9
2	3	81	7	9
3	2	256	7	7
3	3	6561	11	13

Table 3.8 : Number of Iterations.

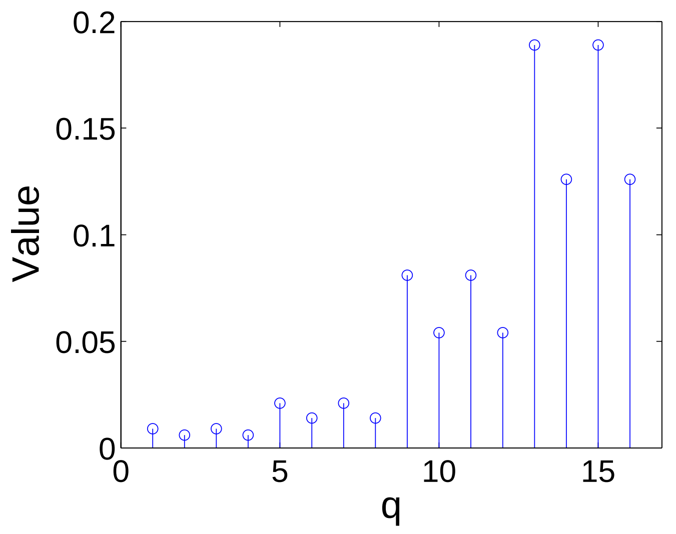
3.8 The Projection-based Gradient Descent Method

• We developed a **projection-based gradient descent** method (Wen et. al. (2015)) for solving the following problem:

$$\boxed{\begin{array}{ll} & \displaystyle\min_{\mathbf{q}\in\Omega} \phi(\mathbf{q}) \equiv \frac{1}{2}\|U\mathbf{q}-\mathbf{p}\|_2^2 \\ \text{where} & \\ & \Omega = \{\mathbf{q} : \mathbf{q}_i \geq 0, \sum_i \mathbf{q}_i = 1\}\,. \end{array}} \tag{24}$$

• The challenge comes from the fact that **the matrix U is huge** in practice such that it is **not desirable to store the matrix**. A **matrix free method** is therefore desirable for computational purpose.

Results: We prove its convergence and apply it to the PBN construction problem. Similar but **more sparse** solutions are obtained.



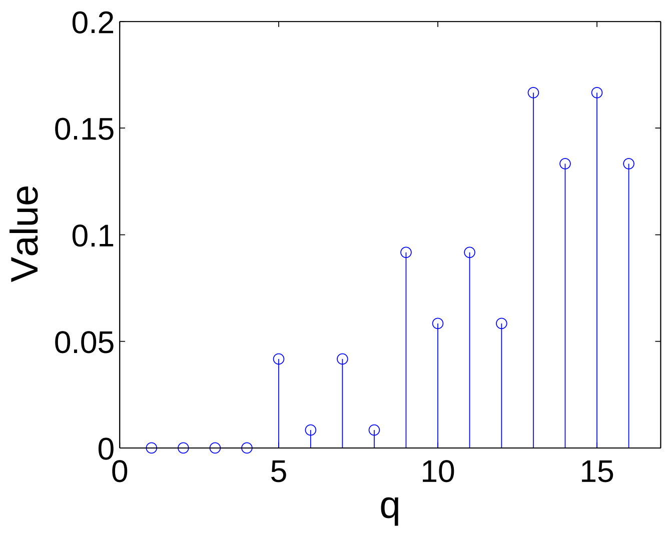


Fig. 3.3 The probability distribution $\mathbf{q}$: Entropy Method (Left), Projection-based Method (Right)

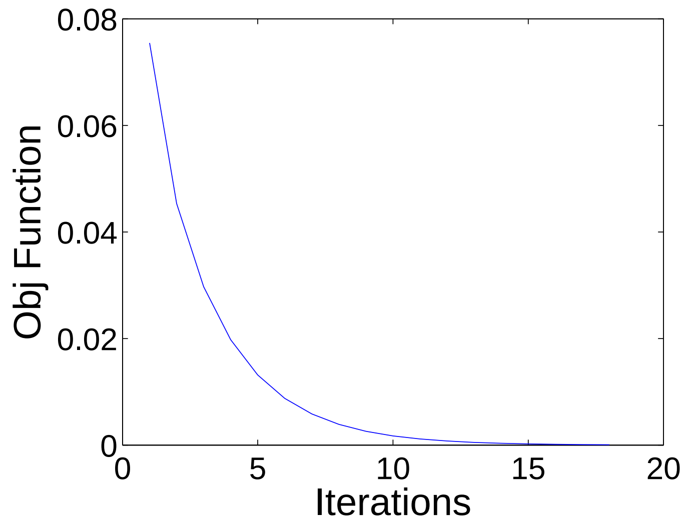


Fig. 3.4 The convergence curve of our method.

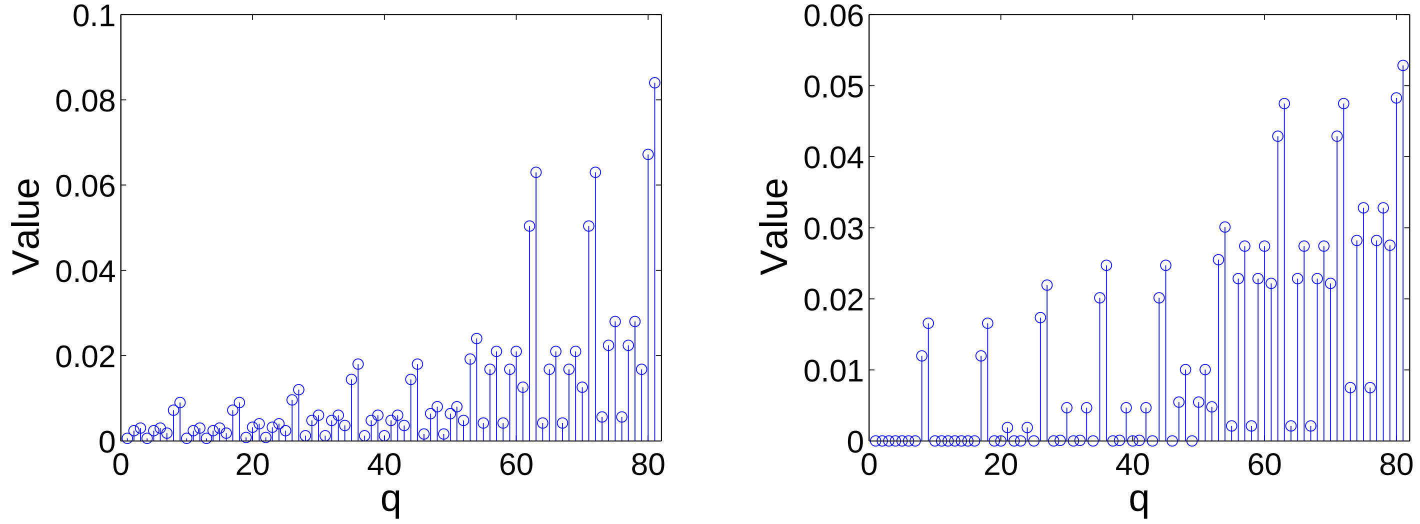


Fig. 3.5 The probability distribution $\mathbf{q}$: Entropy Method (Left), Projection-based method (Right)

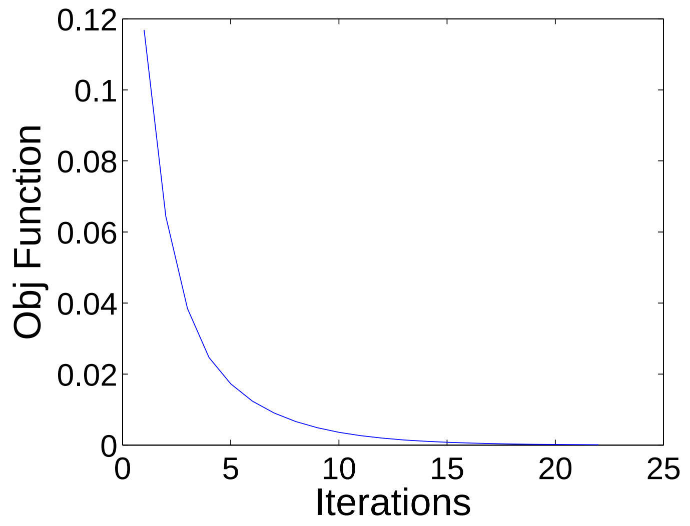


Fig. 3. The convergence curve of our method.

3.9 A Heuristic Method

• A **heuristic method** was proposed for the PBN construction problem in (Ching et al., 2009a).

Results:

-By exploiting the fact that the transition probability matrix can be written as a weighted sum of Boolean network matrices, a heuristic algorithm of $O(m2^n)$ complexity was proposed.

Here n and m are the number of genes, and the number of non-zero entries in the transition matrix, respectively.

-The algorithm was also analyzed numerically.

4. The Optimal Control Problem

4.1 The Motivation

• **Intervention** can be achieved through **removing a gene** or **transplanting a gene**, as done in **gene therapy**. It is an efficient way to generate **mutations** and also hoped to be an efficient way for correcting mutation (therapy).

• **Simian Virus 40** (SV40) was discovered in 1950s during the development of vaccine for certain poliovirus.

- It was found that SV40 can develop tumors when injected in mice. Moreover, SV40 DNA was also found in some human brain tumors and cause cancer in mouse cell.

- Large T-antigen, the protein encoded by SV40 **inactivate** the function of **Gene p53**. It is likely that p53 is a **key gene** for controlling the **global behavior** of the genetic network.

4.2 The Formulation

• Given a PBN, one can consider **structural intervention** (permanently alter the underlying network structure with minimal structural modification) (Pal et al., 2005) and **external control** (apply external perturbation to alter the network dynamics) (Pal et al., 2006) of the network. Here shall consider the later one.

• Here we study a **discrete time control problem**, (Ching et al. 2009b). Beginning with an initial probability distribution $\mathbf{v}_0$ the PBN evolves according to two possible transition probability matrices P_0 and P_1.

• **Without any external control**, we assume that the PBN evolves according to a **fixed transition probability matrix** P_0.

• When a **control is applied** to the network, the PBN will then evolve according to another **transition probability matrix** P_1 (with more favorable steady states or a BN) but it will return back to P_0 again when no more control is applied to the network.

• We remark that one can have more than one type of control, i.e., more than one transition probability matrix P_1 to choose in each time step.

• We then suppose that the **maximum number of controls** that can be applied to the network during the **finite investigation period** T (finite-horizon) is K where $K \leq T$.

• The objective here is to find an optimal control policy such that the state of the network is **close to a target state vector $\mathbf{z}$**.

• Here $\mathbf{z}$ can be a unit vector (**a desirable state**) or a **probability distribution (a weighted average of desirable states)**.

• To facilitate our discussion, we define the following state probability distribution vectors

$$\boxed{\mathbf{v}(i_k i_{k-1} \ldots i_1) = P_{i_k} \cdots P_{i_1} \mathbf{v}_0}$$

to represent all the possible network state probability distribution vectors up to time k. Here

$$\boxed{i_1, i_2, \ldots, i_k \in \{0, 1\} \quad \text{and} \quad \sum_{j=1}^{k} i_j \le K}$$

and $i_k i_{k-1} \ldots i_1$ is a Boolean string of size k.

• We then define

$$\boxed{U(k) = \{\mathbf{v}(i_k i_{k-1} \ldots i_1) : i_1, \ldots, i_k \in \{0, 1\} \quad \text{and} \quad \sum_{j=1}^{k} i_j \le K\}}$$

to be the set containing all the possible state probability vectors up to time k.

• We note that one can conduct a forward calculation to compute recursively all the state vectors in the sets

$$\boxed{U(1), U(2), \ldots, U(T).}$$

• Beginning with $\mathbf{v}_0$, we have

$$\mathbf{v}(0) = P_0\mathbf{v}_0 \quad \text{and} \quad \mathbf{v}(1) = P_1\mathbf{v}_0$$

and therefore

$$U(1) = \{\mathbf{v}(0), \mathbf{v}(1)\} = \{P_0\mathbf{v}_0, P_1\mathbf{v}_0\}.$$

Similarly, we have

$$\begin{aligned} U(2) &= \{\mathbf{v}(00), \mathbf{v}(01), \mathbf{v}(10), \mathbf{v}(11)\} \\ &= \{P_0P_0\mathbf{v}_0, P_1P_0\mathbf{v}_0, P_0P_1\mathbf{v}_0, P_1P_1\mathbf{v}_0\}. \end{aligned}$$

Recursively, one can compute

$$\boxed{U(3), U(4), \ldots, U(T).}$$

• The main computational cost is the **matrix-vector multiplication**.

• We adopted the **sparse matrix structure** for the transition probability matrix, thus the computational cost for each matrix-vector multiplication is at most $O(N2^n)$ where n is the number of genes and N is the number of BNs in the network.

• We don't need to compute and store all the 2^T vectors as some state probability distribution does not exist (the maximum number of controls is K). The total number of vectors involved is

$$\boxed{\sum_{j=0}^{K} \frac{T!}{j!(T-j)!}.}$$

• For example, if $K = 1$ and the number of nonzero entries in the control matrix is no more than $N2^n$, then the complexity of the algorithm is $O(TN2^n)$.

4.3. Two Examples of Objective Functions

• The first objective that one can try is to minimize the **terminal distance** with a target vector $\mathbf{z}$, i.e.,

$$\boxed{\min_{\mathbf{v}(i_T i_{T-1} \dots i_1) \in U(T)} \|\mathbf{v}(i_T i_{T-1} \dots i_1) - \mathbf{z}\|.} \tag{25}$$

• The second possible objective is to minimize the **overall average of the distances** of the state vectors $\mathbf{v}(i_t \dots i_1)$ $(t = 1, \dots, T)$ to a target vector $\mathbf{z}$:

$$\boxed{\min_{\mathbf{v}(i_t i_{t-1} \dots i_1) \in U(T)} \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{v}(i_t \dots i_1) - \mathbf{z}\|.} \tag{26}$$

• For the first optimal control problem (25), once we compute all the feasible state vectors $U(T)$, we can then compute the minimum of the following:

$$\boxed{\min\{\|\mathbf{v}(i_T i_{T-1} \ldots i_1) - \mathbf{z}\|\}.}$$

The optimal control policy can be found accordingly.

• For the second optimal control problem (26), we define the following cost function:

$$\boxed{D(\mathbf{v}(\mathbf{w}_t), t, k), \quad 1 \le t \le T, \quad 0 \le k \le K}$$

as the minimum total distance to the terminal time T when beginning with state distribution vector $\mathbf{v}(\mathbf{w}_t)$ at time t and that the number of controls used is k.

• Here $\mathbf{w}_t$ is a Boolean string of length t.

• To reduce the duplication in the calculation of distances, we consider the following **Dynamic Programming (DP) formulation:**

$$D(\mathbf{v}(\mathbf{w}_{t-1}), t-1, k) = \min\{\|\mathbf{v}(0\mathbf{w}_{t-1}) - \mathbf{z}\| + D(\mathbf{v}(0\mathbf{w}_{t-1}), t, k), \\ \|\mathbf{v}(1\mathbf{w}_{t-1}) - \mathbf{z}\| + D(\mathbf{v}(1\mathbf{w}_{t-1}), t, k+1)\}. \quad (27)$$

Here $0\mathbf{w}_{t-1}$ and $1\mathbf{w}_{t-1}$ are Boolean strings of size t.

• The **first term** in the right-hand-side of (27) is the cost (distance) when no control is applied at time t.

• The **second term** is the cost when a control is applied. The optimal control policy can be obtained during the process of solving Eq. (27).

- To solve our optimal control problem:

$$\min_{0 \le k \le K} \{D(\mathbf{v}_0, 0, k)\}$$

we need the following **boundary conditions**:

$$D(\mathbf{v}(\mathbf{w}_t), t, K+1) = \infty \quad \text{for all } \mathbf{w}_t \text{ and } t.$$

For $k = 0, 1, \ldots, K$,

$$D(\mathbf{v}(\mathbf{w}_T), T, k) = ||\mathbf{v}(\mathbf{w}_T) - \mathbf{z}||$$

for

$$\mathbf{w}_T = i_T \ldots i_1, \text{ and } \sum_{j=1}^{T} i_j \le K.$$

4.4 An Approximation Method

• One of the main cost in the algorithm is the matrix-vector multiplication. We proposed an approximation method based on the idea discussed in (Ching et al. 2007).

• The idea of the approximation method here is to **neglect those BNs with small selection probabilities**. Theoretical results on the distribution of BNs in a PBN has shown that there are a large amount of BNs will be chosen with very small probabilities.

• We assume

$$P_0 = \sum_{i=1}^{N} q_i A_i$$

where A_i is the transition probability of a BN and q_i is the probability of choosing the corresponding BN in the PBN.

• The idea is to remove those BNs whose selection probability is less than a certain **threshold value** τ. After removing such BNs, the number of nonzero entries in the PBN will be reduced significantly.

• Suppose that there are n_τ **Boolean networks** being removed whose corresponding transition matrices are

$$A_1, A_2, \ldots, A_{n_\tau}$$

and their selection probabilities are given by

$$q_1, q_2, \ldots, q_{n_\tau}$$

respectively.

• Then after the removal of these n_τ Boolean networks and making a normalization, the transition probability matrix becomes

$$\boxed{\widetilde{P}_0 = \frac{(P_0 - (q_1 A_1 + q_2 A_2 + \ldots + q_{n_\tau} A_{n_\tau}))}{1 - (q_1 + q_2 + \ldots + q_{n_\tau})}.} \qquad (28)$$

4.5 The Error Bounds

Results: If $\widetilde{P}_0$ is used instead of P_0, the errors of the optimal objective values are bounded above, respectively, by

$$\boxed{(1+\|P_0-\widetilde{P}_0\|_1)^{T-k}-1}$$

and

$$\boxed{\frac{1}{T}\left(k[(\|P_0-\widetilde{P}_0\|_1+1)^{T-k}-1]+\sum_{i=1}^{T-k}[(\|P_0-\widetilde{P}_0\|_1+1)^{i}-1]\right)}$$

when objective functions (25) and (26) are applied. Here k ($0 \leq k \leq K$) is the number of controls applied in the control policy. The two bounds can be further approximated, respectively, by

$$\boxed{(T-k)\|P_0-\widetilde{P}_0\|_1 \quad \text{and} \quad \frac{\|P_0-\widetilde{P}_0\|_1}{2T}(T-k)(T+k+1)}$$

when $\|P_0-\widetilde{P}_0\|_1 \approx 0$.

4.6 Numerical Experiments

4.6.1 A small network example

- We first consider a small **three-gene network** whose transition probability matrix (column sum is one) when there is **no control** is given by

$$P_0 = \begin{pmatrix} 0.3 & 0.3 & 0.0 & 0.2 & 0.3 & 0.0 & 0.0 & 0.2 \\ 0.3 & 0.0 & 0.0 & 0.3 & 0.1 & 0.0 & 0.2 & 0.3 \\ 0.1 & 0.0 & 0.5 & 0.0 & 0.1 & 0.3 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\ 0.0 & 0.1 & 0.0 & 0.3 & 0.2 & 0.1 & 0.2 & 0.1 \\ 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.1 & 0.1 & 0.2 \\ 0.3 & 0.5 & 0.0 & 0.0 & 0.3 & 0.0 & 0.2 & 0.0 \end{pmatrix}.$$

- The transition probability matrix when the **control is applied** is given by

$$P_1 = \begin{pmatrix} 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \end{pmatrix}.$$

- The **target state vector** is

$$\boxed{\mathbf{z} = 0.3(1,0,0,0,0,0,0,0)^T + 0.7(0,0,0,0,0,0,0,1)^T}$$

a probability distribution or actually a weighted average of two state vectors.

- The **initial state vector** is assumed to be the **uniform distribution vector**

$$\boxed{\mathbf{v}_0 = \frac{1}{8}(1,1,1,1,1,1,1,1)^T.}$$

- We assume that the total time T to be 12.

- We try several different maximum number of controls $K = 1,2,3,4,5$.

• Tables 4.1 and 4.2 report the numerical results when two different objective functions (25) and (26) are used, respectively.

• All computations were done in a **PC Pentium 4HT with MATLAB 7.0.** The computational time for solving the optimal policy in all the cases is **less than two seconds**.

K	1	2	3	4	5
Control	[10]	[10]	[7,9,10]	[2,4,6,10]	[2,4,6,8,10]
Objective	0.486	0.486	0.486	0.486	0.486

Table 4.1: A small example of three-gene network when Objective function (25) is used.

K	1	2	3	4	5
Control	[1]	[1,4]	[1,4,12]	[1,4,7,10]	[1,2,10,11,12]
Objective	0.655	0.652	0.649	0.646	0.643

Table 4.2: A small example of three-gene network when Objective function (26) is used.

4.6.2 A 12-gene Network Example

• We give a more complex example, a **12-gene** network is randomly generated. For each gene i, there are **two** Boolean functions $f_1^{(i)}$ and $f_2^{(i)}$.

• All the Boolean functions and their variables are generated randomly.

• Thus there are total **4096 BNs** involved in the PBN. The two objective functions (25) and (26) are used.

• Again we assume $T = 12$. Tables 4.3 and 4.4 give the numerical results for $\tau = 10^{-4}$, where τ is the threshold value. The number of **removed BNs is 2499**.

• The target vector is $\frac{1}{2^n}(0, \ldots, 0, 1, \ldots, 1)^T$. When a control is applied it will turn off the first gene and make no change to other genes.

K	1	2	3	4	5
Control$_{\text{Ori}}$	[10]	[8,10]	[6,8,10]	[4,6,8,10]	[2,4,6,8,10]
Control$_{\text{App}}$	[10]	[8,10]	[6,8,10]	[4,6,8,10]	[2,4,6,8,10]
Error$_{\|\cdot\|_2}$	0	0	0	0	0
Control$_{\text{Ori}}$	[12]	[9,12]	[5,7,12]	[5,7,9,12]	[3,5,7,9,12]
Control$_{\text{App}}$	[12]	[9,12]	[5,7,12]	[5,7,9,12]	[3,5,7,9,12]
Error$_{\|\cdot\|_1}$	0	0	0	0	0
Control$_{\text{Ori}}$	[10]	[8,10]	[8,10]	[4,6,8,10]	[2,4,6,8,10]
Control$_{\text{App}}$	[10]	[8,10]	[8,10]	[4,6,8,10]	[2,4,6,8,10]
Error$_{\|\cdot\|_\infty}$	0	0	0	0	0

Table 4.3: A 12-gene network with objective function (25)

K	1	2	3	4	5
Control$_{\text{Ori}}$	[1]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Control$_{\text{App}}$	[1]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Error$_{\|\cdot\|_2}$	0	0	0	0	0
Control$_{\text{Ori}}$	[1]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Control$_{\text{App}}$	[1]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Error$_{\|\cdot\|_1}$	0	0	0	0	0
Control$_{\text{Ori}}$	[1]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Control$_{\text{App}}$	[2]	[1,2]	[1,2,3]	[1,2,3,4]	[1,2,3,4,5]
Error$_{\|\cdot\|_\infty}$	0.0005	0	0	0	0

Table 4.4: A 12-gene network with objective function (26)

	K	1	2	3	4	5
Objective function (3)	Time_{Ori}	1.72	2.43	4.22	7.44	11.73
	Time_{App}	1.65	2.17	3.45	5.77	8.86
Objective function (4)	Time_{Ori}	1.74	2.49	4.35	7.69	12.15
	Time_{App}	1.70	2.29	3.79	6.48	10.06

Table 4.5: Time consumed for the 12-gene network (in sec.)

Finite-horizon Control of PBNs with Multiple Hard-constraints

• The problem was further studied in (Cong et al., 2009) and (Chen et al., 2013)

Results:

-The case of multiple hard-constraints was studied, an algorithm was developed for finding all optimal control policies.

-A heuristic approach was developed in order to deal with large size networks. A different and more efficient algorithm, using integer linear programming with hard constraints, was presented later by (Chen et al., 2013) using WNT5A network (Kim et al., 2002) as a typical example.

4.8 State Reduction Approach for Optimal Control Policy in a PBN

• A state reduction method (Qian et al., 2010), an approximation method was introduced in (Chen et al., 2012) to find the optimal control policy for a PBN so as to avoid the network from entering into undesirable states. Such problem is NP-hard in general (Akutsu et al., 2007).

Results:

-Inspired by the state reduction strategies (instead of deleting the nodes in a network, we delete the out-most states having less influence to the network), **Dynamic Programming** (DP) method was adopted in conjunction with state reduction approach to reduce the computational cost of the DP method.

-Numerical examples are given to demonstrate both the effectiveness and the efficiency of our proposed method.

5. Concluding Remarks

• In this talk, we reviewed the BN and PBN approach for modeling genetic networks. They can be studied in a Markov chain context. We then introduced some major issues in studying PBNs, they include the computation of the **steady-state distribution**, **finding attractor cycles**, **formulation and algorithms for construction and control for PBNs**.

• We remark that PBNs are not only useful for analyzing complex biological systems at **molecular level** but also at physiological level with potential application at **clinical level**.

-For example, (Ma et al., 2008) applied PBN in processing functional Magnetic Resonance Imaging (fMRI) signals to infer a brain connectivity network for Parkinson disease patients.

• Boolean network model is constructed for representation of the evolution of patterns of economic behavior in financial crisis (Caetano and Yoneyama, 2015).

• Application of PBNs in investigating the relationship between correlated defaults of different industrial sectors and business cycles as well as the impacts of business cycles on modeling and predicting correlated defaults. These are central issues in credit risk measurement and management (Gu et al., 2013).

• Boolean network models produce realistic behavior and also some insights into the reasons for stability in industrial networks (Easton et al., 2008).

• PBNs were employed to study the relationship between machine components, their reliability and function (Torres et al., 2015). The modeling of manufacturing processes through PBNs assists the design of new systems for predictions of future behavior, identifies improvement areas, and evaluates changes to existing systems.

References

• Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998) A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions, *Genome Informatics*, **9**, 151–160.

• Akutsu, T., Hayashida, M., Ching, W., and Ng, M. (2007) Control of Boolean networks: hardness results and algorithms for tree structured networks, *Journal of Theoretical Biology*, **244**, 670–79.

• Caetano, M. and Yoneyama T. (2015) Boolean network representation of contagion dynamics during a financial crisis, *Physica A*, **417**, 1–6.

• Chen, X., Akutsu, T., Tamura, T. and Ching, W. (2013) Finding optimal control policy in probabilistic Boolean networks with hard constraints by using integer programming and dynamic programming, *International Journal of Data Mining and Bioinformatics*, **7**, 322–343.

• Chen, X., Ching, W., Chen, X.S. Cong, Y. and Tsing, N. (2011) Construction of probabilistic Boolean networks from a prescribed transition probability matrix: A maximum entropy rate approach, *East Asian Journal on Applied Mathematics*, **1**, 121–154.

• Chen, X., Jiang, H., Qiu, Y. and Ching W. (2012). On optimal control policy for probabilistic Boolean network: a state reduction approach, *BMC Systems Biology*, 6(Suppl 1):S8.
http://www.biomedcentral.com/1752-0509/6/S1/S8

• Ching, W., Ng, M., Fung, S. and Akutsu, T. (2005) On construction of stochastic genetic networks based on gene expression sequences, *International Journal of Neural Systems*, **15**, 297–310.

• Ching, W, Zhang, S, Ng, M., and Akutsu, T. (2007) An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks, *Bioinformatics*, **23** (2)12, 1511–1518.

• Ching, W., Chen, X., and Tsing, N. (2009a) Generating probabilistic Boolean networks from a prescribed transition probability matrix. *IET Systems Biology*, **3**(6), 453–464.

• Ching, W., Zhang, S., Jiao, Y., Akutsu, T., Tsing, N., and Wong, A. (2009b) Optimal control policy for probabilistic Boolean networks with hard constraints. *IET Systems Biology*, **3**(2), 90–99.

• Cong, Y., Ching, W., Tsing, N., Leung, H. (2010) On finite-horizon control of genetic regulatory networks with multiple hard-constraints. *BMC Systems Biology*, 4(Suppl 2). Article S14.

• De Jong, H. (2002) Modeling and simulation of genetic regulatory systems: A literature review, *Journal of Computational Biology*, **9**, 67–103.

• Dougherty, E., Kim, S., and Chen, Y. (2000) Coefficient of determination in nonlinear signal processing. *Signal Processing*, **80**, 2219–2235.

• Easton, G., Brooks, R., Georgieva, K. and Wilkinson, I. (2008) Understanding the dynamics of industrial networks using Kauffman Boolean networks, *Advances in Complex Systems*, **11** (01) 139–164.

• Friedman, N., Linia, M., Nachman, I. and Peer, D. (2000) Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, **7**, (3-4), 601–620.

• Gu, J., Ching, W., Siu, T. and Zheng, H. (2013) On modeling credit defaults: A probabilistic Boolean network approach. Risk and Decision Analysis, **4**, 119-129.

• Hayashida M, Tamura, T, Akutsu, T, Ching, W, and Cong, Y. (2009) Distribution and enumeration of attractors in probabilistic Boolean networks. *IET Systems Biology*, **3**, (6), 465–474.

• Huang, S. (1999) Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery, *Journal of Molecular Medicine*, **77**, 469–480.

• Kauffman, S. (1969a) Metabolic stability and epigenesis in randomly constructed genetic nets, *Journal of Theoretical Biology*, **22**, 437–467.

• Kauffman, S. (1969b) Homeostasis and differentiation in random genetic control networks, *Nature*, **224**, 177–178.

• Kauffman, S. (1974) The large scale structure and dynamics of gene control circuits: an ensemble approach, *Journal of Theoretical Biology*, **44**, 167–190.

• Kauffman, S. (1993) The origins of order: Self-organization and selection in evolution, New York: Oxford Univ. Press.

• Kim, S., Li, H., Dougherty, E.R., Cao, N., Chen, Y., Bittner, M. and Suh, E. (2002) Can Markov chain models mimic biological regulations? *Journal of Biological Systems*, **10**, 337–357.

• Ma, Z., Wang, Z., McKeown, M. (2008) Probabilistic Boolean network analysis of brain connectivity in Parkinsons disease. *IEEE J Selected Topics in Signal Process*, **2**(6), 975–985.

• Mestl, T., Plahte, E. and Omholts, S. (1995) A Mathematical Framework for Describing and Analyzing Gene Regulatory Networks, *Journal of Theoretical Biology*, **176**, 291–300.

• Pal R., Datta A., Bittner, M., Dougherty, E. (2005) Intervention in context-sensitive probabilistic Boolean networks, *Bioinformatics*, **21 (7)**, 1211–1218.

• Qian X, Ghaffari N, Ivanov I, Dougherty ER (2010) State reduction for network intervention in probabilistic Boolean networks, *Bioinformatics*, **26**, 3098–3104.

• Pal, R., Datta, A., Dougherty, E. (2006) Optimal infinite-Horizon control for probabilistic Boolean networks, *IEEE Tran. Signal Processing*, **54** (6), 2375–2387.

• Shmulevich I., Dougherty E., Kim S., and Zhang W. (2002a) Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks, *Bioinformatics*, **18**, 261–274.

• Shmulevich I., Dougherty E., Kim S., and Zhang W. (2002b) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, *Proceedings of the IEEE*, **90**, 1778–1792.

• Shmulevich I., Dougherty E., Kim S., and Zhang W. (2002c) Gene perturbation and intervention in probabilistic Boolean networks, *Bioinformatics*, **18**, 1319–1331.

• Shmulevich I., Dougherty E., Kim S., and Zhang W. (2002d) Control of stationary behavior in probabilistic Boolean networks by means of structural intervention, *Biological Systems*, **10**, 431–446.

• Shmulevich, I., Gluhovsky, I., Hashimoto, R., Dougherty, E., and Zhang (2003) Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks, *Comparative and Functional Genomics*, **4**, 601–608.

• Shmulevich, I. and Dougherty, E. (2010) *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*, Philadelphia: SIAM.

• Steggles, L., Banks, R., Shaw, O. and Wipat, A. (2007) Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach, *Bioinformatics*, **23**, 336–343.

• Torres, P., Mercado, E. and Rifon, L. (2015) Probabilistic Boolean network modeling of an industrial machine, *Journal of Intelligent Manufacturing*, DOI 10.1007/s10845-015-1143-4

• Trairatphisan, P., Mizera, A., Pang, J., Tantar, A., Schneider, J. and Sauter, T. (2013) *Recent Development and Biomedical Applications of Probabilistic Boolean Networks*, Cell Communication and Signaling, 11:46. (http://www.biosignaling.com/content/11/1/46).

• Wen, Y., Wang, M., Cao, Z., Cheng, X., Ching, W and Vassiliadis, V. (2015) Sparse Solution of Non-negative Least Squares Problems with Applications in the Construction of Probabilistic Boolean Networks, *Journal of Numerical Linear Algebra with Applications*, **22**, 883–889.

• Someren, V., Wessels, E. and Reinders M. (2000) Linear Modeling of Genetic Networks from Experimental Data, *Intelligent Systems for Molecular Biology* (ISMB 2000) August 19-23, San Diego CA.

• Yeung, K. and Ruzzo, W. (2001) An Empirical Study on Principal Component Analysis for Clustering Gene Expression Data, *Bioinformatics*, **17**, 763–774.

• Zhang, S., Hayashida, M., Akutsu, T., Ching, W. and Ng, M. (2007a) Algorithms for finding small attractors in Boolean networks, *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 20180.

• Zhang, S., Ching, W., Ng, M. and Akutsu, T. (2007b) Simulation study in probabilistic Boolean network models for genetic regulatory networks, *Journal of Data Mining and Bioinformatics*, **1**, 217–240.

• Zhang S, Ching W, Chen X, Tsing N. (2010). Generating probabilistic Boolean networks from a prescribed stationary distribution. *Information Science*, **180**, 2560–2570.