Model Construction of Boolean Network via Observed Data

Daizhan Cheng, *Fellow, IEEE*, Hongsheng Qi, *Member, IEEE*, and Zhiqiang Li

***Abstract*— In this paper, a set of data is assumed to be obtained from an experiment that satisfies a Boolean dynamic process. For instance, the dataset can be obtained from the diagnosis of describing the diffusion process of cancer cells. With the observed datasets, several methods to construct the dynamic models for such Boolean networks are proposed. Instead of building the logical dynamics of a Boolean network directly, its algebraic form is constructed first and then is converted back to the logical form. Firstly, a general construction technique is proposed. To reduce the size of required data, the model with the known network graph is considered. Motivated by this, the least in-degree model is constructed that can reduce the size of required data set tremendously. Next, the uniform network is investigated. The number of required data points for identification of such networks is independent of the size of the network. Finally, some principles are proposed for dealing with data with errors.**

***Index Terms*— Algebraic form, identification, infection process, least in-degree model, uniform Boolean network.**

I. Introduction

The Boolean network was firstly proposed by Kauffman [1], and since then it has attracted great attentions from biologists, physicians, and systems scientists and became a hot topic in systems biology, physics, and systems science [2]–[7]. Some practical useful control techniques have also been proposed [8], [9].

The dynamics of a Boolean network is a logical system. A major difficulty in dealing with logical systems is that few analysis tools can be used. Recently, a new technique, called the semi-tensor product of matrices, was proposed. It converts the logical systems into standard discrete-time dynamic systems [10], [11]. It can also be used to deal with the control problems of Boolean networks [12], [13].

Network identification, which is also called the reverse engineering or network inference, investigates the algorithms to construct the dynamics of Boolean networks, such as genetic regulatory networks, etc. It is currently a topic of high interest of the community of mathematical biologists and particularly in the area of systems biology. The data can be obtained from variety of sources [14]. For instance, the Human Genome Project produces a gigantic set of human deoxy ribo nucleic acid data, and digesting it is essentially a modeling problem.

Various algorithms have been proposed for the network inference. For instance, in [15] a general reverse engineering algorithm has been proposed for inference of genetic network architecture. Identification by using a small number of gene expression patterns was proposed in [16]. Then another identification algorithm based on matrix multiplication and fingerprint function was proposed by them [17]. In [18] a randomized network search algorithm is presented, which requires less average time. In [19], a method is proposed to construct the network dynamics using prescribed attractor structure. Using computational algebra, the link adaptation algorithm was proposed by [20] and then the estimates for the expected amount of data were presented in [21]. The number of experiments required to characterize a network was also investigated in [22].

In this paper, a new method is proposed to build a model of Boolean network by using experimental data. The approach is based on the framework proposed by the authors recently [10]–[13]. In this approach, the logical dynamics of a Boolean network are converted into its algebraic forms, either componentwise or as a whole. The algebraic form is a standard discrete-time system described as a difference equation. These algebraic forms are equivalent to the original logical dynamic system. Hence, instead of identifying the original system, it is needed only to identify its algebraic forms.

Most of existing identification methods for Boolean networks are based on certain proposed algorithms. They provide only sufficient conditions. One of the advantages of such numerical methods is that they may be used for comparatively large networks. Compared to the existing methods, our approach emphasizes on the theoretical analysis. It provides a rigorous model when the data are sufficient. One of the advantages of our approach is that in this approach only parameters of algebraic equations need to be identified. This is much simpler than identifying logical relations directly. Another advantage is that the method can be easily extended to the multivalued case. Its critical defect is the computational complexity. However, when some special structure properties are known, the number of required data points could be reduced.

The basic tool in this approach is the semi-tensor product of matrices, which is a generalization of the conventional matrix product [23]. Let A and B be two arbitrary matrices of

Manuscript received August 28, 2010; revised January 5, 2011; accepted January 7, 2011. Date of publication February 22, 2011; date of current version April 6, 2011. This work was supported in part by the National Natural Science Foundation of China under Grant 61074114, Grant 60736022, and Grant 60221301.

D. Cheng and H. Qi are with the Key Laboratory of Systems and Control, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dcheng@iss.ac.cn; qihongsh@amss.ac.cn).

Z. Li is with the Department of Mathematics and Information Science, Henan University of Economics and Law, Zhengzhou 450002, China (e-mail: lizhiqiang@amss.ac.cn).

Digital Object Identifier 10.1109/TNN.2011.2106512

dimensions $m \times n$ and $p \times q$, respectively, and $s = \mathrm{lcm}\{n, p\}$ be the least common multiple of n and p. The semi-tensor product of A and B is defined as

$$A \ltimes B := \left(A \otimes I_{s/n}\right)\left(B \otimes I_{s/p}\right). \tag{1}$$

Using it, for each logical function there is a matrix, called its structure matrix (SM), the function can be expressed as the product of the SM with its arguments. This is the key for converting a logical dynamic system into a discrete-time system. Throughout this paper, the matrix product is assumed to be the semi-tensor product. Moreover, in most cases the symbol $\ltimes$ is omitted.

The method proposed in this paper can be used to identify any Boolean network as well as any multivalued logical network. Particularly, certain special types of Boolean network models, such as networks with known network graph (equivalently, incidence matrix), least in-degree model, and uniform networks, are proposed. To identify them, the number of required experimental data can be reduced tremendously.

The rest of this paper is organized as follows. In Section II, the necessary preliminaries, such as showing what is the (component-wise) algebraic form of a Boolean network and how to convert its dynamics into its (component-wise) algebraic form and vise versa, are introduced. In Section III, a general algorithm is provided to identify a Boolean network through the observed data. It is shown that at least $2^n + 1$ data points are required. Section IV shows that, when the network graph is known, the number of required data could be tremendously reduced. The least in-degree model is discussed in Section V. Theoretically, no matter how small the data size is, a least in-degree model can always be built (may not be unique). Section VI considers a model called the uniform Boolean network. It is shown that this model commonly exists. Its identification is independent of the network size. In Section VII, some principles are proposed to deal with data errors. Section VIII includes the concluding remarks.

II. Preliminaries

First, the following notations are used.

1) $\mathcal{D}_k := \{0, 1/(k-1), \ldots, (k-2)/(k-1), 1\}$, $k \geq 2$; $\mathcal{D} := \mathcal{D}_2 = \{0, 1\}$.
2) Let δ_n^i be the ith column of the identity matrix I_n, and $\Delta_n := \{\delta_n^1, \delta_n^2, \ldots, \delta_n^n\}$. When $n = 2$, simply use $\Delta := \Delta_2$.
3) Assume a matrix $M = [\delta_n^{i_1}\ \delta_n^{i_2}\ \ldots\ \delta_n^{i_s}] \in \mathcal{M}_{n\times s}$, i.e., its columns, $\mathrm{Col}(M) \subset \Delta_n$. M is called a logical matrix, and can be simply denoted as

$$M = \delta_n[i_1\ i_2\ \ldots\ i_s].$$

The set of $n \times s$ logical matrices is denoted by $\mathcal{L}_{n\times s}$.

Consider a Boolean network. Its dynamics is described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t)) \\ x_2(t+1) = f_2(x_1(t), \ldots, x_n(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t)) \end{cases} \tag{2}$$

TABLE I
SMs of Logical Operators (LOs)

LO	$\neg$	$\wedge$	$\vee$
SM	$M_n = \delta_2[2\ 1]$	$M_c = \delta_2[1\ 2\ 2\ 2]$	$M_d = \delta_2[1\ 1\ 1\ 2]$
LO	$\rightarrow$	$\leftrightarrow$	$\bar{\vee}$
SM	$M_i = \delta_2[1\ 2\ 1\ 1]$	$M_e = \delta_2[1\ 2\ 2\ 1]$	$M_p = \delta_2[2\ 1\ 1\ 2]$

where $x_i(t) \in \mathcal{D}$ are state variables, $f_i : \mathcal{D}^n \to \mathcal{D}$ are logical functions.

To use the matrix expression of logic, identifying $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$ is necessary. Then, $x_i(t) \in \Delta$ and $f_i : \Delta^n \to \Delta$. Define $x(t) := \ltimes_{i=1}^n x_i(t)$. It was proved in [10] and [11] that the system (2) can be expressed as

$$\begin{cases} x_1(t+1) = M_1 x(t) \\ x_2(t+1) = M_2 x(t) \\ \vdots \\ x_n(t+1) = M_n x(t) \end{cases} \tag{3}$$

where $M_i \in \mathcal{L}_{2\times 2^n}$, called the SM of f_i. Equation (3) is called the component-wise algebraic form of (2). Moreover, (3) can further be converted as

$$x(t+1) = L x(t) \tag{4}$$

where $L \in \mathcal{L}_{2^n\times 2^n}$ is called the transition matrix of the system. Equation (4) is called the algebraic form of (2).

For convenience, the SMs of some basic LOs are listed in Table I.

For instance, let $x = 1$ and $y = 0$. Then in vector form, $x = \delta_2^1$ and $y = \delta_2^2$. Now, using Table I, it follows that

$$\begin{aligned} &\neg x = M_n \ltimes x = \delta_2[2\ 1] \ltimes \delta_2^1 = \delta_2^2 \sim 0; \\ &x \wedge y = M_c \ltimes x \ltimes y = \delta_2[1\ 2\ 2\ 2] \ltimes \delta_2^1 \ltimes \delta_2^2 = \delta_2^2 \sim 0; \\ &x \vee y = M_d \ltimes x \ltimes y = \delta_2[1\ 1\ 1\ 2] \ltimes \delta_2^1 \ltimes \delta_2^2 = \delta_2^1 \sim 1; \\ &\ldots \end{aligned}$$

It was proved that (2)–(4) are equivalent to each other. Then it is not difficult to see that using experimental data to reconstruct model (3) or (4) is much easier than (2). This is because building model (3) or (4) is equivalent to calculating the SMs M_i (correspondingly, transition matrix L), while building model (2) directly seems much more difficult.

Finally, it is worth noting that some mechanical algorithms were proposed in [12] for the following conversions:

$$\text{C1: } (2) \to (3) \to (4) \quad \text{C2: } (4) \to (3) \to (2).$$

Before showing how to realize C1, some related properties of the semi-tensor product are briefly reviewed.

Proposition 2.1: The semi-tensor product has the following properties (in the following, the symbol $\ltimes$ is omitted).

1) Let $X \in \mathbb{R}^t$ be a column. Then

$$XA = (I_t \otimes A)X. \tag{5}$$

2) Let $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ be two columns. Then

$$W_{[m,n]}XY = YX \tag{6}$$

where $W_{[m,n]}$ is an $mn \times mn$ matrix, called the swap matrix [11].

3) Let $x \in \Delta$. Then
$$x^2 = M_r x \tag{7}$$
where $M_r = \delta_4[1\ 4]$ is called the power-reducing matrix [11].

C1 is based on the above properties of the semi-tensor product. A briefly explanation is given as follows.

Step 1: (2) $\to$ (3).

1) Using SMs of basic LOs to express $f_i(x_1,\dots,x_n)$ as a product of SMs M_j and arguments x_i.
For instance, consider
$$\begin{aligned} f(x_1,x_2) &= x_2 \leftrightarrow (x_1 \wedge x_2) \\ &= M_e x_2 M_c x_1 x_2. \end{aligned} \tag{8}$$
2) Using (5) to move all M_j to the front and all x_i to the rear.
For instance
$$M_e x_2 M_c x_1 x_2 = M_e(I_2 \otimes M_c) x_2 x_1 x_2. \tag{9}$$
3) Using (6) to re-order x_i.
For instance
$$x_2 x_1 x_2 = W_{[2,2]} x_1 x_2^2. \tag{10}$$
4) Using (7) to reduce the order of x_i. Previous steps may be used again to get the final form.
For instance
$$x_1 x_2^2 = x_1 M_r x_2 = (I_2 \otimes M_r) x_1 x_2. \tag{11}$$
Summarizing (8)–(11) yields
$$f(x_1,x_2) = M_f x_1 x_2 \tag{12}$$
where the SM M_f is
$$M_f = M_e(I_2 \otimes M_c) W_{[2,2]} (I_2 \otimes M_r).$$

Step 2: (3) $\to$ (4). Denote by $\mathrm{Col}_i(M)$ the ith column of matrix M. Then the coefficient matrix L in (4) can be obtained by [24]
$$\mathrm{Col}_i(L) = \ltimes_{j=1}^n \mathrm{Col}_i(M_j), \qquad i = 1,\dots,2^n. \tag{13}$$

Next consider C2. C2 is essential for this paper. It is briefly described as follows.

Step 1: (4) $\to$ (3). Define a set of matrices in $\mathcal{L}_{2\times 2^n}$, called the retrievers, as
$$\begin{aligned} S_1^n &= \delta_2[\underbrace{1,\dots,1}_{2^{n-1}},\underbrace{2,\dots,2}_{2^{n-1}}] \\ S_2^n &= \delta_2[\underbrace{1,\dots,1}_{2^{n-2}},\underbrace{2,\dots,2}_{2^{n-2}},\underbrace{1,\dots,1}_{2^{n-2}},\underbrace{2,\dots,2}_{2^{n-2}}] \\ &\ \vdots \\ S_n^n &= \delta_2[1,2,1,2,\dots,1,2]. \end{aligned} \tag{14}$$
Then
$$M_i = S_i^n L, \qquad i = 1,\dots,n. \tag{15}$$
Hence
$$x_i(t+1) = M_i \ltimes_{j=1}^n x_j(t), \qquad i = 1,\dots,n. \tag{16}$$

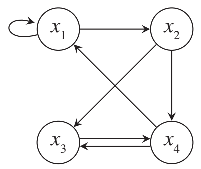


Fig. 1. Network graph of (21).

Step 2: Simplify (3). Use the following theorem to remove the fabricated variables.

Theorem 2.2: Let the algebraic form of a logical function be
$$f(x_1,\dots,x_n) = M \ltimes_{j=1}^n x_j. \tag{17}$$
f is independent of x_j (i.e., x_j is a fabricated variable), iff
$$M W_{[2,2^{j-1}]}(I - M_n) = 0 \tag{18}$$
where M_n is the SM of negation and $W_{[p,q]}$ is the swap matrix [23]. Moreover, if (18) holds, then the function can be expressed as
$$f(x_1,\dots,x_n) = \tilde{M} x_1 \dots x_{j-1} x_{j+1} \dots x_n$$
where
$$\tilde{M} = M W_{[2,2^{j-1}]} \delta_2^1.$$
Step 3: (3) $\to$ (2). Let
$$f(x_1,\dots,x_n) = M \ltimes_{i=1}^n x_i. \tag{19}$$
Split M into two equal parts as $M = [M^1, M^2]$. Then
$$\begin{aligned} f(x_1,\dots,x_n) = (x_1 \wedge f^1(x_2,\dots,x_n)) \\ \vee(\neg x_1 \wedge f^2(x_2,\dots,x_n)) \end{aligned} \tag{20}$$
where f^i has its SM as M^i, $i = 1,2$.

Using (20) repeatedly, the algebraic form (3) can finally be converted into its logical form as (2).

Remark 2.3: For k-valued logical networks, identify
$$\frac{i}{k-1} \sim \delta_k^{k-i}, \qquad i = 0,1,\dots,k-1.$$
Then with mild modifications, all the above forms and conversions remain true. For instance, we have (3) with $M_i \in \mathcal{L}_{k\times k^n}$, $i = 1,\dots,n$, and (4) with $L \in \mathcal{L}_{k^n\times k^n}$ [25]. So the following identification process remains applicable to k-valued logical networks.

The following example shows how to realize conversion C1.

Example 2.4: Consider the Boolean network shown in Fig. 1.

Its dynamics are described as
$$\begin{cases} x_1(t+1) = x_1(t) \to x_4(t) \\ x_2(t+1) = \neg x_1(t) \\ x_3(t+1) = x_2(t) \wedge x_4(t) \\ x_4(t+1) = x_2(t) \leftrightarrow x_3(t). \end{cases} \tag{21}$$

First, express the system (21) in its component-wise algebraic form as
$$\begin{cases} x_1(t+1) = M_i x_1(t) x_4(t) = \delta_2[1\ 2\ 1\ 1] x_1(t) x_4(t) \\ x_2(t+1) = M_n x_1(t) = \delta_2[2\ 1] x_1(t) \\ x_3(t+1) = M_c x_2(t) x_4(t) = \delta_2[1\ 2\ 2\ 2] x_2(t) x_4(t) \\ x_4(t+1) = M_e x_2(t) x_3(t) = \delta_2[1\ 2\ 2\ 1] x_2(t) x_3(t). \end{cases} \tag{22}$$

Setting $x = \ltimes_{i=1}^4 x_i(t)$, the system (21) can be expressed in its algebraic form as

$$\begin{aligned}x(t+1) &= M_i x_1(t)x_4(t)M_n x_1(t)M_c x_2(t)x_4(t)M_e x_2(t)x_3(t)\\ &= M_i(I_4 \otimes M_n)x_1(t)x_4(t)x_1(t)M_c x_2(t)x_4(t)M_e x_2(t)x_3(t)\\ &\;\vdots\\ &= Lx(t) \qquad (23)\end{aligned}$$

where

$$L = \delta_{16}[5\ 15\ 6\ 16\ 8\ 16\ 7\ 15\ 1\ 3\ 2\ 4\ 4\ 4\ 3\ 3]. \qquad (24)$$

The following example shows how to realize the conversion C2.

Example 2.5: Assume that a Boolean network with four nodes is considered, the algebraic form of the network dynamics is

$$x(t+1) = Lx(t) \qquad (25)$$

where

$$L = \delta_{16}[5\ 7\ 9\ 11\ 8\ 8\ 4\ 4\ 1\ 3\ 13\ 15\ 4\ 4\ 8\ 8].$$

Now, reconstruct its logical expression. The retrievers are calculated as

$$\begin{aligned}S_1^4 &= \delta_2[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]\\ S_2^4 &= \delta_2[1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2]\\ S_3^4 &= \delta_2[1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2]\\ S_4^4 &= \delta_2[1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2]. \qquad (26)\end{aligned}$$

Then the following component-wise algebraic form can be obtained as

$$\begin{cases}x_1(t+1) = M_1 x(t)\\ x_2(t+1) = M_2 x(t)\\ x_3(t+1) = M_3 x(t)\\ x_4(t+1) = M_4 x(t)\end{cases} \qquad (27)$$

where

$$\begin{aligned}M_1 = S_1^4 L &= \delta_2[1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1]\\ M_2 = S_2^4 L &= \delta_2[2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2]\\ M_3 = S_3^4 L &= \delta_2[1\ 2\ 1\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 2\ 2]\\ M_4 = S_4^4 L &= \delta_2[1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2]. \qquad (28)\end{aligned}$$

First, consider the logical expression of $x_1(t)$

$$x_1(t+1) = M_4 x(t) = \delta_2[1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1]x(t).$$

It is easy to verify that

$$\begin{aligned}M_1(M_n - I_2) &= 0\\ M_1 W_{[2,2]}(M_n - I_2) &\neq 0\\ M_1 W_{[2,4]}(M_n - I_2) &\neq 0\\ M_1 W_{[2,8]}(M_n - I_2) &= 0. \qquad (29)\end{aligned}$$

So $x_1(t)$ and $x_4(t)$ are fabricated variables in the dynamical equation of $x_1(t+1)$. Set $x_1(t) = x_4(t) = \delta_2^1$. Then

$$\begin{aligned}x_1(t+1) &= M_1 x_1(t)x_2(t)x_3(t)x_4(t)\\ &= M_1 W_{[2,8]} x_4(t)x_1(t)x_2(t)x_3(t)\\ &= \delta_2[1\ 2\ 1\ 1]x_2(t)x_3(t).\end{aligned}$$

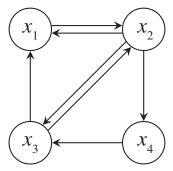


Fig. 2. Network graph of (30).

Hence the logical expression is

$$x_1(t+1) = x_2(t) \rightarrow x_3(t).$$

The same procedure can be used to reconstruct the logical expression of $x_2(t), x_3(t)$, and $x_4(t)$. Finally, from the SM (25), the following logical expression is obtained:

$$\begin{cases}x_1(t+1) = x_2(t) \rightarrow x_3(t)\\ x_2(t+1) = x_1(t)\bar{\vee}x_3(t)\\ x_3(t+1) = x_2(t) \wedge x_4(t)\\ x_4(t+1) = x_2(t)\end{cases} \qquad (30)$$

with its network graph depicted in Fig. 2.

III. Model Construction for General Networks

Assume a Boolean network consisting of n nodes. Let $X(t) = \{x_1(t), \ldots, x_n(t)\}$. Denote the observed data as $\{X(0), X(1), \ldots, X(N)\}$. A rigorous definition for the model construction is given below.

Definition 3.1: Assume that a set of observed data $\{X(0), X(1), \ldots, X(N)\}$ is given, where $X(t) = \{x_1(t), \ldots, x_n(t)\}$. The model construction problem is as follows. Find a logical dynamic system (2), such that the given data verify the dynamic equation.

A model that is verified by the given data is called a realization of the data.

From this definition, there is an immediate result.

Proposition 3.2: The system is uniquely identifiable, iff the set of data $\{X(0), X(1), \ldots, X(N-1)\}$ contains all possible states.

Proof: Convert the data into the vector form by using $x(t) := \ltimes_{i=1}^n x_i(t)$. Then in algebraic form, it follows that $x(t) = \delta_{2^n}^i$ and $x(t+1) = \delta_{2^n}^j$, iff the ith column of L is

$$\mathrm{Col}_i(L) = \delta_{2^n}^j. \qquad (31)$$

It follows that L is identifiable, iff in the vector form

$$\{x(0), x(1), \ldots, x(N-1)\} = \Delta_{2^n}.$$

The conclusion follows. ■

If the experiment has been carried out more than once, the following result is obvious.

Corollary 3.3: Assume the observed data consist of k groups as $\{X^i(0), X^i(1), \ldots, X^i(N_i)\}$, $i = 1, \ldots, k$. Then the system is uniquely identifiable, iff (in the vector form)

$$\left\{x^i(0), \ldots, x^i(N_i - 1) \middle| i = 1, 2, \ldots, k\right\} = \Delta_{2^n}. \qquad (32)$$

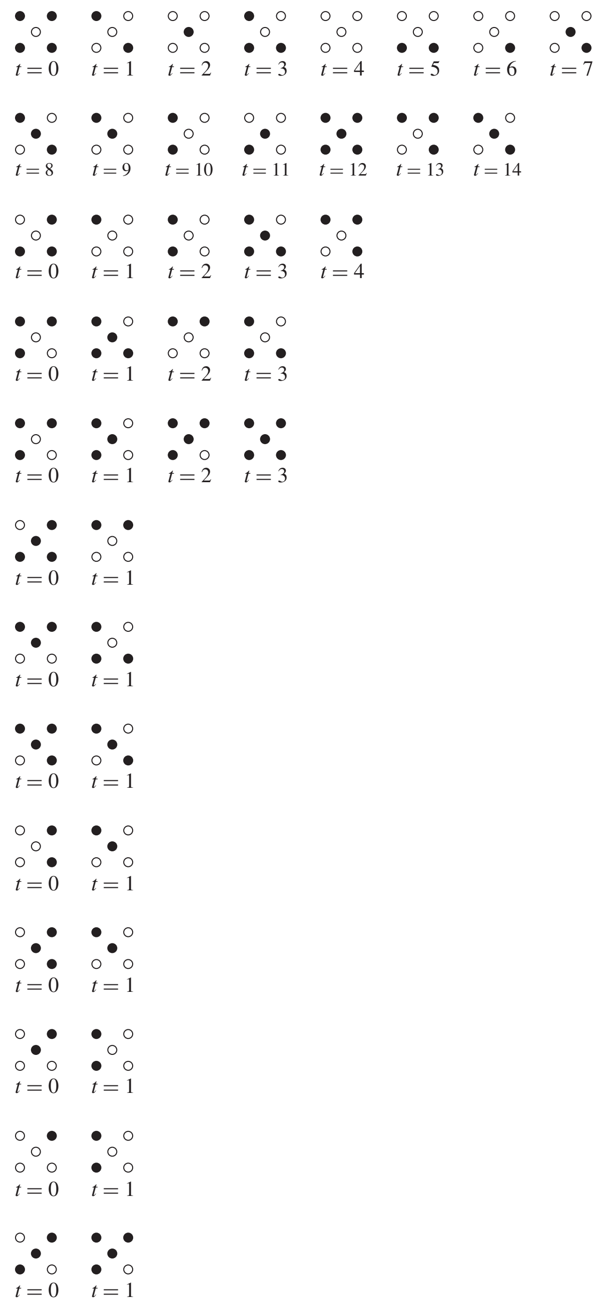


Fig. 3. Observed data from experiments in Example 3.5.

Remark 3.4:

1) From Proposition 3.2, one sees that, to identify a Boolean network of n nodes, at least $2^n + 1$ data are necessary.
2) If the data are not enough or do not satisfy the condition of Proposition 3.2, (31) still can be used to identify some columns. Then the model is not unique. Uncertain column(s) of L can be chosen arbitrarily.

Example 3.5: Assume a set of five cells is considered. The 12 groups of experimental data are shown in Fig. 3, where a white disc, numbered 1, represents a healthy cell, and a black disc, numbered by 0, represents an infected cell. Our goal is to build a dynamic model for the process of infection.

From the first experimental data, we have (where the nodes are ordered from left to right and then from top to bottom)

$$\begin{aligned}
&X^1(0) = (0,0,1,0,0); && X^1(1) = (0,1,1,1,0);\\
&X^1(2) = (1,1,0,1,1); && X^1(3) = (0,1,1,0,0);\\
&X^1(4) = (1,1,1,1,1); && X^1(5) = (1,1,1,0,0);\\
&X^1(6) = (1,1,1,1,0); && X^1(7) = (1,1,0,1,0);\\
&X^1(8) = (0,1,0,1,0); && X^1(9) = (0,1,0,1,1);\\
&X^1(10) = (0,1,1,0,1); && X^1(11) = (1,1,0,0,1);\\
&X^1(12) = (0,0,0,0,0); && X^1(13) = (0,0,1,1,0);\\
&X^1(14) = (0,1,0,1,0).
\end{aligned}$$

Now in the vector form, we have $X^1(0) = \delta_2[2,2,1,2,2]$ and

$$x^1(0) = \delta_2^2 \ltimes \delta_2^2 \ltimes \delta_2^1 \ltimes \delta_2^2 \ltimes \delta_2^2 = \delta_{32}^{28}.$$

Similarly, we can calculate

$$\begin{aligned}
&x^1(0) = \delta_{32}^{28}; \quad x^1(1) = \delta_{32}^{18}; \quad x^1(2) = \delta_{32}^{5}; \quad x^1(3) = \delta_{32}^{20};\\
&x^1(4) = \delta_{32}^{1}; \quad x^1(5) = \delta_{32}^{4}; \quad x^1(6) = \delta_{32}^{2}; \quad x^1(7) = \delta_{32}^{6};\\
&x^1(8) = \delta_{32}^{22}; \quad x^1(9) = \delta_{32}^{21}; \quad x^1(10) = \delta_{32}^{19}; \quad x^1(11) = \delta_{32}^{7};\\
&x^1(12) = \delta_{32}^{32}; \quad x^1(13) = \delta_{32}^{26}; \quad x^1(14) = \delta_{32}^{22}.
\end{aligned}$$

Using Proposition 3.2 [precisely, (31)], it is known that

$$\mathrm{Col}_{28}(L) = \delta_{32}^{18}; \quad \mathrm{Col}_{18}(L) = \delta_{32}^{5}; \quad \mathrm{Col}_{5}(L) = \delta_{32}^{20}; \quad \ldots$$

The 14 columns of L have been determined.

Using the same procedure to the other groups of data, certain values of columns of L can be figured out. Finally, we can easily obtain

$$\begin{aligned}
L = \delta_{32}[&4\ 6\ 8\ 2\ 20\ 22\ 32\ 26\ 19\ 21\ 23\ 17\ 19\ 21\ 31\ 25\\
&3\ 5\ 7\ 1\ 19\ 21\ 31\ 25\ 20\ 22\ 24\ 18\ 20\ 22\ 32\ 26]. \quad (33)
\end{aligned}$$

Hence, the algebraic form of the dynamics of the infection process from the experimental data is

$$x(t+1) = Lx(t) \tag{34}$$

where $L \in \mathcal{L}_{32\times 32}$ is shown in (33).

Next, construct its logical dynamic equation to see the interaction between cells. Using (14), the corresponding retrievers are

$$\begin{aligned}
S_1^5 = \delta_2[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\\
&2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]\\
S_2^5 = \delta_2[&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\
&1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]\\
S_3^5 = \delta_2[&1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\\
&1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2]\\
S_4^5 = \delta_2[&1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\\
&1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2]\\
S_5^5 = \delta_2[&1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\\
&1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2]. \quad (35)
\end{aligned}$$

Then the following component-wise algebraic form can be obtained as

$$\begin{cases} x_1(t+1) = M_1 x(t) \\ x_2(t+1) = M_2 x(t) \\ x_3(t+1) = M_3 x(t) \\ x_4(t+1) = M_4 x(t) \\ x_5(t+1) = M_5 x(t) \end{cases} \tag{36}$$

where

$$\begin{aligned} M_1 = S_1^5 L = \delta_2[&1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2] \\ M_2 = S_2^5 L = \delta_2[&1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2 \\ &1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2] \\ M_3 = S_3^5 L = \delta_2[&1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1 \\ &1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 1] \\ M_4 = S_4^5 L = \delta_2[&2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1 \\ &2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1\ 2\ 1] \\ M_5 = S_5^5 L = \delta_2[&2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]. \end{aligned} \tag{37}$$

First consider the logical expression of $x_1(t)$

$$\begin{aligned} x_1(t+1) = M_1 x(t) = \delta_2[&1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &1\ 1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]\ x(t). \end{aligned}$$

The fabricated variables can be removed by using Theorem 2.2. It is easy to verify that

$$\begin{aligned} M_1(M_n - I_2) &= 0 \\ M_1 W_{[2,2]}(M_n - I_2) &\neq 0 \\ M_1 W_{[2,4]}(M_n - I_2) &\neq 0 \\ M_1 W_{[2,8]}(M_n - I_2) &= 0 \\ M_1 W_{[2,16]}(M_n - I_2) &= 0. \end{aligned} \tag{38}$$

So $x_1(t)$, $x_4(t)$, and $x_5(t)$ are fabricated variables in the dynamic equation of $x_1(t+1)$. Setting $x_1(t) = x_4(t) = x_5(t) = \delta_2^1$ yields

$$\begin{aligned} x_1(t+1) &= M_1 x_1(t) x_2(t) x_3(t) x_4(t) x_5(t) \\ &= M_1 W_{[4,8]} x_4(t) x_5(t) x_1(t) x_2(t) x_3(t) \\ &= M_1 W_{[4,8]} (\delta_2^1)^3 x_2(t) x_3(t) \\ &= \delta_2[1\ 2\ 2\ 2] x_2(t) x_3(t). \end{aligned}$$

Hence its logical expression is

$$x_1(t+1) = x_2(t) \wedge x_3(t).$$

The same procedure can be used to reconstruct the logical expression of $x_2(t)$, $x_3(t)$, $x_4(t)$, and $x_5(t)$. Finally, the logical expression of the dynamics of the group of cells is obtained as

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ x_2(t+1) = x_3(t) \vee x_4(t) \\ x_3(t+1) = x_4(t) \leftrightarrow x_5(t) \\ x_4(t+1) = \neg x_5(t) \\ x_5(t+1) = x_1(t) \bar{\vee} x_2(t). \end{cases} \tag{39}$$

with its network graph depicted in Fig. 4.

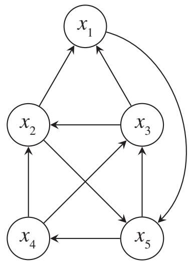


Fig. 4. Network graph of (39).

IV. Construction with Known Network Graph

In previous section, a general method is given to construct the dynamic model of a Boolean network from its experimental data. As pointed out before, in general at least $2^n + 1$ data points are necessary for uniquely determining the model. As n is not very small, this is a large number and in practical experiments such amount of data can hardly be obtained. In this section, the situation where the network graph is known is considered. In this case, the required data can be considerably reduced.

Note that when the number of network nodes is large, drawing its network graph is a heavy job. An alternative expression of the dynamic connection of nodes is the incidence matrix [26]. Consider an n-node network. An $n \times n$ matrix, $\mathcal{J} = (r_{i,j}) \in \mathcal{M}_{n\times n}$, is called its incidence matrix, where $r_{i,j} = 1$, if $x_i(t+1)$ depends on $x_j(t)$ directly, otherwise, $r_{i,j} = 0$. For instance, recall Example 2.4. Its incidence matrix is

$$\mathcal{J}|_{(21)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{40}$$

Consider the following example.

Example 4.1: Consider a network with 8 nodes. Its network graph is depicted in Fig. 5.

Assume that for this network there are experimental data as in Fig. 6.

To build its dynamic model, the component-wise algebraic form is used. That is

$$\begin{cases} x_1(t+1) = M_1 x_8(t) \\ x_2(t+1) = M_2 x_1(t) \\ x_3(t+1) = M_3 x_2(t) \\ x_4(t+1) = M_4 x_3(t) x_7(t) \\ x_5(t+1) = M_5 x_4(t) \\ x_6(t+1) = M_6 x_5(t) \\ x_7(t+1) = M_7 x_6(t) \\ x_8(t+1) = M_8 x_3(t) x_7(t). \end{cases} \tag{41}$$

From the data, it is easy to see that

$$\begin{aligned} x_8(0) = 0 &\Rightarrow x_1(1) = 1; \\ x_8(1) = 1 &\Rightarrow x_1(2) = 0; \quad \cdots \end{aligned}$$

Then in the vector form

$$\mathrm{Col}_2(M_1) = \delta_2^1; \quad \mathrm{Col}_1(M_1) = \delta_2^2; \quad \cdots$$

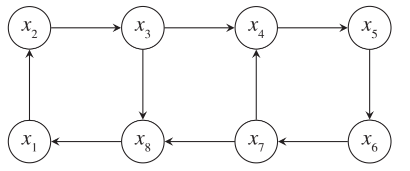


Fig. 5. Network graph of (41).

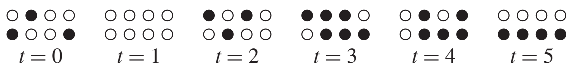


Fig. 6. Data of Network (41).

It is concluded that $M_1 = \delta_2[2, 1]$, and hence

$$x_1(t+1) = \neg x_8(t).$$

Similarly, the other M_i, $i = 2, 3, \ldots, 8$ can be calculated. Finally, the dynamics is obtained as

$$\begin{cases} x_1(t+1) = \neg x_8(t) \\ x_2(t+1) = x_1(t) \\ x_3(t+1) = \neg x_2(t) \\ x_4(t+1) = x_3(t) \vee x_7(t) \\ x_5(t+1) = x_4(t) \\ x_6(t+1) = \neg x_5(t) \\ x_7(t+1) = x_6(t) \\ x_8(t+1) = x_3(t) \wedge x_7(t). \end{cases} \tag{42}$$

Comparing this example with Example 3.5, it is obvious that, when the network graph (equivalently, the incidence matrix) is known, then the data needed to build the model will be much less.

V. Least In-Degree Model

Consider a network with n nodes. The in-degree of node k, denoted by $d_i(k)$, is the number of edges which point to node k. Consider the incidence matrix of the network. Then

$$d_i(k) = \sum_{j=1}^{n} r_{kj}, \qquad k = 1, \ldots, n. \tag{43}$$

For instance, consider Example 2.4. Its in-degrees are $d_i(1) = 2$, $d_i(2) = 1$, $d_i(3) = 2$, and $d_i(4) = 2$. Consider Example 3.5. Its in-degrees are $d_i(1) = d_i(2) = d_i(3) = d_i(5) = 2$, and $d_i(4) = 1$.

It is well known that [27] in an ordered network the in-degrees are much less than the number of nodes. In an experiment of random light-bulb networks, it was assumed that the number of nodes is $n = 1\,00\,000$ and the in-degree $d_i = 2$. In this section, the least in-degree model is considered.

Definition 5.1: Consider an n-node Boolean network with given experimental data. A realization with the in-degree $d_i^*(k)$, $k = 1, \ldots, n$, is called the least in-degree model, if for any other realization with in-degree $d_i(k)$, $k = 1, \ldots, n$, it is obtained that

$$d_i^*(k) \le d_i(k), \qquad k = 1, \ldots, n.$$

It is obvious that a least in-degree model requires much less data to identify a least in-degree model. Moreover, it is reasonable to assume a real practical network to be of least in-degree. In the following, we consider how to get a least in-degree realization. Start from the component-wise algebraic form (3). Denote a set of experimental data by $\{X(0), X(1), \ldots, X(N)\}$. Consider the dynamics of the ith node

$$x_i(t+1) = M_i x(t), \quad \text{where} \quad M_i \in \mathcal{L}_{2\times 2^n}. \tag{44}$$

Using this set of data, some columns of the SM M_i can be determined. For instance

$$M_i = [* \ldots * \, c_{i_1} * \ldots * \, c_{i_2} * \ldots * \ldots * \, c_{i_s} * \ldots *] \tag{45}$$

where c_{i_j}, $j = 1, \ldots, s$ are the identified columns and $*$ stands for the uncertain columns. Equation (45) is called the uncertain SM. Next, a set of matrices is constructed as

$$M_{i,j} := M_i W_{[2,2^{j-1}]}, \qquad j = 1, 2, \ldots, n.$$

Then split it into two equal parts as

$$M_{i,j} = \begin{bmatrix} M_{i,j}^1 & M_{i,j}^2 \end{bmatrix}. \tag{46}$$

Then the following result is obtained.

Proposition 5.2: f_i has a realization which is independent of x_j, iff

$$M_{i,j}^1 = M_{i,j}^2 \tag{47}$$

has a solution for the uncertain columns.

Proof: When $j = 1$, $M_{i,j} = M_i$ holds. Now split

$$M_i = \begin{bmatrix} M_i^1 & M_i^2 \end{bmatrix}. \tag{48}$$

If

$$M_i^1 = M_i^2 \tag{49}$$

has a solution for uncertain elements, then the solution makes $M_i^1 = M_i^2$. Using (18), it is easy to see that this realization is independent of x_1. Consider x_j, (44) can be rewritten as

$$x_i(t+1) = M_{i,j} x_j \ltimes_{k=1}^{j-1} x_k \ltimes_{k=j+1}^{n} x_k.$$

The same argument as for x_1 leads to the general conclusion. ■

Next, an algorithm is given for producing a least in-degree realization.

Algorithm 5.3:

Step 1: For each component-wise algebraic equation, use the observed data to identify part of its columns as (45). Define the incidence set as $S_i = \{1, 2, \ldots, n\}$, $i = 1, \ldots, n$.

Step 2: Construct (48) to check whether (49) has a solution. If "Yes," fix some uncertain columns and update the system to

$$x_i(t+1) = M_i^1 \ltimes_{j=2}^{n} x_j(t).$$

Go to the next step.

Step j: (where $j \le n$) Check whether (47) has a solution. If "Yes," fix some uncertain columns and update the system to

$$x_i(t+1) = M_{i,j}^1 \ltimes_{1\le k\le j-1,\, k\in S_i} x_k \ltimes_{k=j+1}^{n} x_k. \tag{50}$$

Replace S_i by $S_i \backslash \{j\}$.

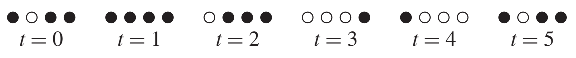


Fig. 7. Experimental data of Example 5.5.

The following conclusion comes from the design of the algorithm.

Proposition 5.4: Algorithm 5.3 yields least in-degree realization(s).

The following example is given to illustrate this.

Example 5.5: Consider the experiment data in Fig. 7.

The vector forms of the data are

$$x(0)=\delta_{16}^{12};\quad x(1)=\delta_{16}^{16};\quad x(2)=\delta_{16}^{8};$$
$$x(3)=\delta_{16}^{2};\quad x(4)=\delta_{16}^{10};\quad x(5)=\delta_{16}^{12}.$$

Using the technique developed in Section III, some columns of M_1 via the known data can be identified as

$$M_1=\delta_2[*\ 2\ *\ *\ *\ *\ *\ 1\ *\ 2\ *\ 2\ *\ *\ *\ 1].$$

Setting $M_1^1=M_1^2$ yields the solution as

$$M_1^1=M_1^2=\delta_2[*\ 2\ *\ 2\ *\ *\ *\ 1].$$

So the system can be simplified as

$$x_1(t+1)=\delta_2[*\ 2\ *\ 2\ *\ *\ *\ 1]x_2(t)x_3(t)x_4(t).$$

Splitting M_1^1 into two parts and considering the following equation:

$$\delta_2[*\ 2\ *\ 2]=\delta_2[*\ *\ *\ 1]$$

there is no solution. So the equation depends on x_2.

Consider

$$M_{1,2}=M_1^1W_{[2,2]}=\delta_2[*\ 2\ *\ *\ *\ 2\ *\ 1].$$

Then

$$\delta_2[*\ 2\ *\ *]=\delta_2[\ *\ 2\ *\ 1]$$

has a solution as

$$\delta_2[*\ 2\ *\ 1].$$

Then the original equation can be updated as

$$x_1(t+1)=\delta_2[*\ 2\ *\ 1]x_2(t)x_4(t).$$

Finally, check $x_4(t)$. Since

$$\delta_2[*\ 2\ *\ 1]W_{[2,2]}=\delta_2[*\ *\ 2\ 1]$$

and

$$\delta_2[*\ *]=\delta_2[2\ 1]$$

has a solution

$$\delta_2[2\ 1]$$

we obtain

$$x_1(t+1)=\delta_2[2\ 1]x_2(t).$$

That is

$$x_1(t+1)=\neg x_2(t).$$

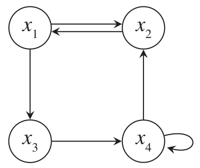


Fig. 8. Network graph of (51).

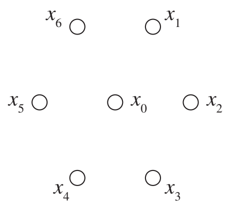


Fig. 10. Neighborhood of x_0.

Using the same procedure to three other equations, the least in-degree realization can finally be obtained as

$$\begin{cases} x_1(t+1)=\neg x_2(t)\\ x_2(t+1)=x_4(t)\vee x_1(t)\\ x_3(t+1)=x_1(t)\\ x_4(t+1)=x_3(t)\bar{\vee}x_4(t)\end{cases}\tag{51}$$

with its network graph depicted in Fig. 8.

It is easy to prove that, if

$$d_i(k)\le\mu,\qquad k=1,\dots,n$$

then the least number of data to identify the system is $2^\mu+1$, which is, in general, much less than 2^n+1.

VI. Construction of Uniform Boolean Network

In this section, the case where the network has a uniform dynamic structure is considered. The physical meaning is like this, assume there is a set of cells. Each cell might be infected only by its neighborhood. Moreover, the rule for a cell being infected is the same for all other cells. Then each cell has the same interactive pattern (i.e., same logical dynamics) with its neighborhoods.

Example 6.1: Let a set of experimental data be given as in Fig. 9. Assume that the infection process is uniform and each cell x_0 is affected only by its neighboring cells. Refer to Fig. 10, where the neighboring cells of x_0 are x_1, x_2, x_3, x_4, x_5, and x_6. Moreover, it is also reasonable to assume that the infection is isotropic. That is, the six neighborhood cells of a cell can be labeled in a clockwise fashion and start from any one cell. Then the dynamic equation becomes

$$x_0(t+1)=M\ltimes_{i=0}^{6}x_i(t),\quad \text{where}\quad M\in\mathcal{L}_{2\times 2^7}.\tag{52}$$

Our purpose is to identify M.

Some special points are checked here to demonstrate how to figure out M.

1) Consider $x_0=A$. Note that $x_0(0)=1$ and in its neighborhood we have $x_1(0)=1$, $x_2(0)=1$, $x_3(0)=1$, $x_4(0)=1$, $x_5(0)=1$, and $x_6(0)=1$: that is, $X(0)=(1,1,1,1,1,1,1)\sim\delta_{128}^1$. Since $x_0(1)=1$, it is concluded that $\mathrm{Col}_1(M)=\delta_2^1$.

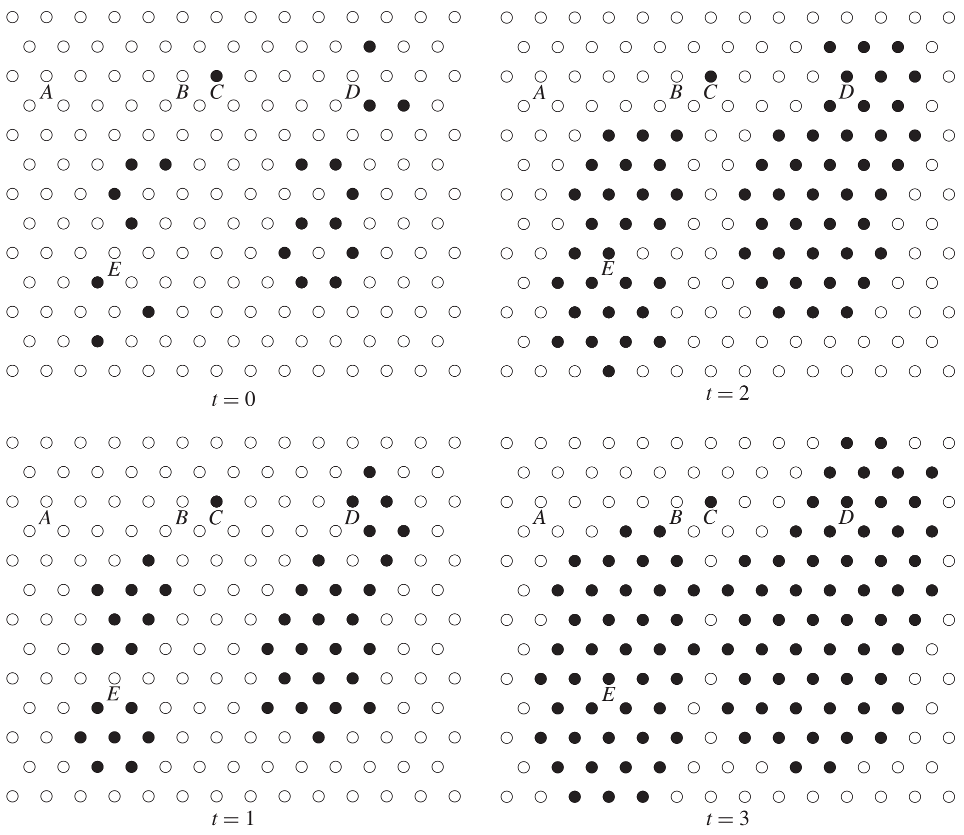


Fig. 9. Data of Example 6.1.

2) Consider $x_0 = B$. We have $x_0(0) = 1$ and in its neighborhood we have $x_1(0) = 1$, $x_2(0) = 1$, $x_3(0) = 1$, $x_4(0) = 1$, $x_5(0) = 1$, and $x_6(0) = 0$: that is, $X(0) = (1, 1, 1, 1, 1, 1, 0) \sim \delta_{128}^2$. Since $x_0(1) = 1$, it is concluded that $\mathrm{Col}_2(M) = \delta_2^1$. Moreover, by isotropy it can also be assumed that $X(0)$ are on all rotating positions to produce the same $x_0(1)$. That is

$$\begin{aligned}X(0) &= (1,1,1,1,1,0,1) \sim \delta_{128}^3 \Rightarrow \mathrm{Col}_3(M) = \delta_2^1\\ X(0) &= (1,1,1,1,0,1,1) \sim \delta_{128}^5 \Rightarrow \mathrm{Col}_5(M) = \delta_2^1\\ X(0) &= (1,1,1,0,1,1,1) \sim \delta_{128}^9 \Rightarrow \mathrm{Col}_9(M) = \delta_2^1\\ X(0) &= (1,1,0,1,1,1,1) \sim \delta_{128}^{17} \Rightarrow \mathrm{Col}_{17}(M) = \delta_2^1\\ X(0) &= (1,0,1,1,1,1,1) \sim \delta_{128}^{33} \Rightarrow \mathrm{Col}_{33}(M) = \delta_2^1.\end{aligned}$$

3) Consider $x_0 = C$. Since $x_0(0) = 0$ and $x_0(1) = 0$, we have $X(0) = (0, 1, 1, 1, 1, 1, 1) \sim \delta_{128}^{65}$, which implies $\mathrm{Col}_{65}(M) = \delta_2^2$.

4) Consider $x_0 = D$. Since $x_0(0) = 1$ and $x_0(1) = 0$, using isotropy, we have

$$\begin{aligned}X(0) &= (1,0,1,0,1,1,1) \sim \delta_{128}^{41} \Rightarrow \mathrm{Col}_{41}(M) = \delta_2^2\\ X(0) &= (1,1,0,1,1,1,0) \sim \delta_{128}^{18} \Rightarrow \mathrm{Col}_{18}(M) = \delta_2^2\\ X(0) &= (1,0,1,1,1,0,1) \sim \delta_{128}^{35} \Rightarrow \mathrm{Col}_{35}(M) = \delta_2^2\\ X(0) &= (1,1,1,1,0,1,0) \sim \delta_{128}^{6} \Rightarrow \mathrm{Col}_{6}(M) = \delta_2^2\\ X(0) &= (1,1,1,0,1,0,1) \sim \delta_{128}^{11} \Rightarrow \mathrm{Col}_{11}(M) = \delta_2^2\\ X(0) &= (1,1,0,1,0,1,1) \sim \delta_{128}^{21} \Rightarrow \mathrm{Col}_{21}(M) = \delta_2^2.\end{aligned}$$

5) Consider $x_0 = E$. Since $x_0(0) = 1$, $x_0(1) = 1$, and $x_0(2) = 0$, we have

$$\begin{aligned}X(0) &= (1,0,1,1,0,1,1) \sim \delta_{128}^{37} \Rightarrow \mathrm{Col}_{37}(M) = \delta_2^1\\ X(0) &= (1,1,1,0,1,1,0) \sim \delta_{128}^{10} \Rightarrow \mathrm{Col}_{10}(M) = \delta_2^1\\ X(0) &= (1,1,0,1,1,0,1) \sim \delta_{128}^{19} \Rightarrow \mathrm{Col}_{19}(M) = \delta_2^1\end{aligned}$$

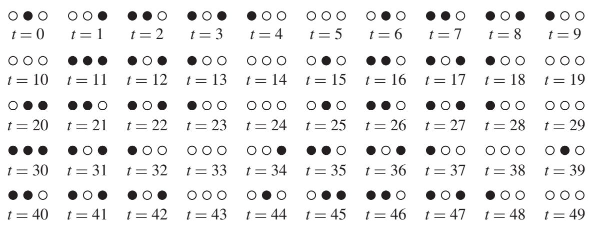
and

$$\begin{aligned}X(0) &= (1,0,1,0,0,1,1) \sim \delta_{128}^{45} \Rightarrow \mathrm{Col}_{45}(M) = \delta_2^2\\ X(0) &= (1,1,0,0,1,1,0) \sim \delta_{128}^{26} \Rightarrow \mathrm{Col}_{26}(M) = \delta_2^2\\ X(0) &= (1,0,0,1,0,0,1) \sim \delta_{128}^{51} \Rightarrow \mathrm{Col}_{51}(M) = \delta_2^2\\ X(0) &= (1,0,1,1,0,1,0) \sim \delta_{128}^{38} \Rightarrow \mathrm{Col}_{38}(M) = \delta_2^2\\ X(0) &= (1,1,1,0,1,0,0) \sim \delta_{128}^{12} \Rightarrow \mathrm{Col}_{12}(M) = \delta_2^2\\ X(0) &= (1,1,0,1,0,0,1) \sim \delta_{128}^{23} \Rightarrow \mathrm{Col}_{23}(M) = \delta_2^2.\end{aligned}$$

6) ...

Finally, all the columns of M can be identified as

$$\begin{aligned}M = \delta_2[&1\ 1\ 1\ 2\ 1\ 2\ 2\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\\ &1\ 2\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &1\ 2\ 2\ 2\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\\ &2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2].\end{aligned}$$



$t=0$ $t=1$ $t=2$ $t=3$ $t=4$ $t=5$ $t=6$ $t=7$ $t=8$ $t=9$
$t=10$ $t=11$ $t=12$ $t=13$ $t=14$ $t=15$ $t=16$ $t=17$ $t=18$ $t=19$
$t=20$ $t=21$ $t=22$ $t=23$ $t=24$ $t=25$ $t=26$ $t=27$ $t=28$ $t=29$
$t=30$ $t=31$ $t=32$ $t=33$ $t=34$ $t=35$ $t=36$ $t=37$ $t=38$ $t=39$
$t=40$ $t=41$ $t=42$ $t=43$ $t=44$ $t=45$ $t=46$ $t=47$ $t=48$ $t=49$

Fig. 11. Experimental data of Example 7.1.

Skipping the tedious and standard process, the algebraic form can finally be converted back to the logical form as

$$\begin{aligned}
&x_0(t+1)\\
&= x_0(t) \wedge (x_1(t) \vee x_2(t)) \wedge (x_2(t) \vee x_3(t)) \wedge (x_3(t) \vee x_4(t))\\
&\wedge (x_4(t) \vee x_5(t)) \wedge (x_5(t) \vee x_6(t)) \wedge (x_6(t) \vee x_1(t))\\
&\wedge (x_1(t) \vee x_3(t)) \wedge (x_3(t) \vee x_5(t)) \wedge (x_5(t) \vee x_1(t))\\
&\wedge (x_2(t) \vee x_4(t)) \wedge (x_4(t) \vee x_6(t)) \wedge (x_6(t) \vee x_2(t)). \qquad (53)
\end{aligned}$$

VII. Modeling via Data with Errors

Up to this stage, the data considered are precisely correct. In dealing with practical data, certain numerical methods should be used. In this section, some basic ideas for dealing with imperfect data are proposed.

If the data have some errors caused by measurement or others, in identification there may be conflicting data. For example, it is obtained from the data that

$$\mathrm{Col}_i(L) = \begin{cases} \delta_{2^n}^p, & k \text{ times} \\ \delta_{2^n}^q, & s \text{ times.} \end{cases} \qquad (54)$$

1) If $k \gg s$, then ignore $\delta_{2^n}^q$ and let $\mathrm{Col}_i(L) = \delta_{2^n}^p$.
2) If $k \ll s$, then ignore $\delta_{2^n}^p$ and let $\mathrm{Col}_i(L) = \delta_{2^n}^q$.
3) If $k \approx s$, then more data may be needed or (when data are already enough) it is concluded that the model is not acceptable.

Similar judgment can be made for each M_i in component-wise model.

For instance, the least in-degree model is considered. Set a threshold value k and denote by t_k^s the times for column i to be identified as s. Then the following principle is proposed for the identification. Let $\{I, J\}$ be a partition of index set $\{1, 2, \ldots, n\}$. Assume

$$\begin{cases} \mathrm{Col}_j(M) = \delta_2^{s_j}, & t_j^{s_j} \geq k, \ j \in J \\ \mathrm{Col}_i(M) = \delta_2^{s_i}, & t_i^{s_i} \ll k, \ i \in I. \end{cases} \qquad (55)$$

Then $\mathrm{Col}_i(M)$, $i \in I$ is considered as error columns and set $\mathrm{Col}_i(M) = *$, i.e., consider them as uncertain columns.

Roughly speaking, only the confident data should be taken for model construction. Many statistic testing methods can be used to tell whether a particular data is reliable or not. A simple example is given to depict it.

Example 7.1: Suppose there is a Boolean network with three nodes. The experimental data are depicted in Fig. 11.

The 50 experimental data can be converted into the vector form as

$$\begin{array}{llll}
x(0) = \delta_8^3; & x(1) = \delta_8^2; & x(2) = \delta_8^7; & x(3) = \delta_8^6; \\
x(4) = \delta_8^5; & x(5) = \delta_8^1; & x(6) = \delta_8^3; & x(7) = \delta_8^7; \\
x(8) = \delta_8^6; & x(9) = \delta_8^5; & x(10) = \delta_8^1; & x(11) = \delta_8^8; \\
x(12) = \delta_8^6; & x(13) = \delta_8^5; & x(14) = \delta_8^1; & x(15) = \delta_8^3; \\
x(16) = \delta_8^7; & x(17) = \delta_8^6; & x(18) = \delta_8^5; & x(19) = \delta_8^1; \\
x(20) = \delta_8^4; & x(21) = \delta_8^7; & x(22) = \delta_8^6; & x(23) = \delta_8^5; \\
x(24) = \delta_8^1; & x(25) = \delta_8^3; & x(26) = \delta_8^7; & x(27) = \delta_8^6; \\
x(38) = \delta_8^5; & x(29) = \delta_8^1; & x(30) = \delta_8^8; & x(31) = \delta_8^6; \\
x(32) = \delta_8^5; & x(33) = \delta_8^1; & x(34) = \delta_8^2; & x(35) = \delta_8^7; \\
x(36) = \delta_8^6; & x(37) = \delta_8^5; & x(38) = \delta_8^1; & x(39) = \delta_8^3; \\
x(40) = \delta_8^7; & x(41) = \delta_8^6; & x(42) = \delta_8^6; & x(43) = \delta_8^1; \\
x(44) = \delta_8^3; & x(45) = \delta_8^4; & x(46) = \delta_8^7; & x(47) = \delta_8^6; \\
x(48) = \delta_8^5; & x(49) = \delta_8^1. & &
\end{array}$$

Suppose the component-wise algebraic form of $x_1(t)$ is

$$x_1(t+1) = M_1 x(t) \qquad M_1 \in \mathcal{L}_{2\times 8}.$$

From the data, using the technique developed in Section III, it follows that:

$$\mathrm{Col}_1(M_1) = \begin{cases} \delta_2^1, & 8 \text{ times} \\ \delta_2^2, & 2 \text{ times.} \end{cases} \qquad (56)$$

Hence, set $\mathrm{Col}_1(M_1) = \delta_2^2$.

For the second to the eighth columns, it follows that

$$\begin{aligned}
\mathrm{Col}_2(M_1) &= \delta_2^2, \quad 2 \text{ times;} \\
\mathrm{Col}_3(M_1) &= \begin{cases} \delta_2^1, & 2 \text{ times} \\ \delta_2^2, & 4 \text{ times;} \end{cases} \\
\mathrm{Col}_4(M_1) &= \delta_2^2, \quad 2 \text{ times;} \\
\mathrm{Col}_5(M_1) &= \delta_2^1, \quad 9 \text{ times;} \qquad (57) \\
\mathrm{Col}_6(M_1) &= \begin{cases} \delta_2^1, & 1 \text{ time} \\ \delta_2^2, & 10 \text{ times;} \end{cases} \\
\mathrm{Col}_7(M_1) &= \delta_2^2, \quad 6 \text{ times;} \\
\mathrm{Col}_8(M_1) &= \delta_2^2, \quad 2 \text{ times.}
\end{aligned}$$

Hence, the matrix M_1 can be obtained as

$$M_1 = \delta_2[1\ 2\ 2\ 2\ 1\ 2\ 2\ 2].$$

Splitting M_1 as $M_1 = [M_{11}\ \ M_{12}]$, $M_{11} = M_{12}$ holds. The algebraic form of $x_1(t)$ is

$$x_1(t+1) = \delta_2[1\ 2\ 2\ 2] x_2(t) x_3(t).$$

Converting it into its logical form, we get

$$x_1(t+1) = x_2(t) \wedge x_3(t).$$

Using the same technique for $x_2(t)$ and $x_3(t)$, the logical expression from data is obtained as

$$\begin{cases} x_1(t+1) = x_2(t) \wedge x_3(t) \\ x_2(t+1) = \neg x_1(t) \\ x_3(t+1) = x_1(t) \vee x_2(t). \end{cases} \qquad (58)$$

Go back to the data, it is easy to check that there are eight wrong data points. The method seems relatively robust.

Finally, we would like to mention that, if a model is constructed and later on there are additional data, the model could be updated in the following way. If the kth equation verifies new data, it remains available. Otherwise, new identified columns could be added to the existing set and be used to build new SM M_k. Then new kth equation can be updated.

VIII. Conclusion

Assuming that for a Boolean network there are some observed data, the problem of constructing the dynamic model of the network via these data was considered. Instead of building the logical dynamic equations, we first identified its algebraic form and then converted the algebraic form back to the logical form. First, the general construction problem, which identifies the SM L of the network directly, was considered. Since it requires a large number of data, it is unrealistic for non-tiny networks. Then the case when the network graph is known was considered. It reduces the required data points tremendously. In fact, the data points required depend only on the in-degrees. Based on this observation, the least in-degree model that removes possible fabricated logical variables from each state equations was proposed. Theoretically, the least in-degree model can be obtained via any small number of data points. Finally, the case when the network has a uniform structure was considered. It is practically realistic because assuming all cells satisfying the same infection rule is reasonable. In this case, model construction is independent of the number of nodes.

The data considered in Sections II–VI are assumed to be precisely correct, except in last section (Section VII) in which some principles were given for data with error. In fact, many numerical methods can be used to deal with data error situations. But this is beyond the scope of this paper.

Finally, we would like to point out that the method proposed in this paper can be used for multivalued logical networks without any essential change.

References

[1] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Mar. 1969.

[2] T. Akutsu, M. Hayashida, W.-K. Ching, and M. K. Ng, "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *J. Theor. Biol.*, vol. 244, no. 4, pp. 670–679, Feb. 2007.

[3] R. Albert and H. G. Othmer, "The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster," *J. Theor. Biol.*, vol. 223, no. 1, pp. 1–18, 2003.

[4] M. Aldana, "Boolean dynamics of networks with scale-free topology," *Phys. D*, vol. 185, no. 1, pp. 45–66, Oct. 2003.

[5] A. Data, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: The imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, Jan. 2004.

[6] B. Drossel, T. Mihaljev, and F. Greil, "Number and length of attractors in a critical Kauffman model with connectivity one," *Phys. Rev. Lett.*, vol. 94, no. 8, pp. 088701-1–088701-4, Mar. 2005.

[7] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2375–2387, Jun. 2006.

[8] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. Bernardo, "How to infer gene networks from expression profiles," *Mol. Syst. Biol.*, vol. 3, no. 78, pp. 1–10, Feb. 2007.

[9] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. Bernardo, D. Bernardo, and M. P. Cosma, "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172–181, Mar. 2009.

[10] D. Cheng, "Input-state approach to Boolean networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 512–521, Mar. 2009.

[11] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2251–2258, Oct. 2010.

[12] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, Jul. 2009.

[13] D. Cheng, Z. Li, and H. Qi, "Realization of Boolean control networks," *Automatica*, vol. 46, no. 1, pp. 62–69, Jan. 2010.

[14] B. Palsson, *Systems Biology: Properties of Reconstructed Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[15] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL: A general reverse engineering algorithm for inference of genetic network architectures," in *Proc. Pacific Symp. Biocomput.*, vol. 3. 1998, pp. 18–29.

[16] 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 *Proc. Pacific Symp. Biocomput.*, 1999, pp. 17–28.

[17] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function," *J. Comput. Biol.*, vol. 7, nos. 3–4, pp. 331–343, 2000.

[18] D. Nam, S. Seo, and S. Kim, "An efficient top-down search algorithm for learning Boolean networks of gene expression," *Mach. Learn.*, vol. 65, no. 1, pp. 229–245, Oct. 2006.

[19] R. Pal, I. Ivanov, A. Datta, M. L. Bittner, and E. R. Dougherty, "Generating Boolean networks with a prescribed attractor structure," *Bioinformatics*, vol. 21, no. 21, pp. 4021–4025, Sep. 2005.

[20] R. Laubenbacher and B. Stigler, "A computational algebra approach to the reverse engineering of gene regulatory networks," *J. Theor. Biol.*, vol. 229, no. 4, pp. 523–537, Aug. 2004.

[21] W. Just, "Reverse engineering discrete dynamical systems from data sets with random input vectors," *J. Comput. Biol.*, vol. 13, no. 8, pp. 1435–1456, 2006.

[22] B. Krupa, "On the number of experiments required to find the causal structure of complex systems," *J. Theor. Biol.*, vol. 219, no. 2, pp. 257–267, Nov. 2002.

[23] D. Cheng, "Semi-tensor product of matrices and its applications–A survey," in *Proc. Int. Conf. Comput. Mach.*, vol. 3. 2007, pp. 641–668.

[24] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. New York: Springer-Verlag, 2010.

[25] Z. Li and D. Cheng, "Algebraic approach to dynamics of multivalued networks," *Int. J. Bifurc. Chaos*, vol. 20, no. 3, pp. 561–582, 2010.

[26] F. Robert, *Discrete Iterations* (Transl. by J. Rokne, Ed.). Berlin, Germany: Springer-Verlag, 1986.

[27] S. A. Kauffman, *At Home in the Universe*. London, U.K.: Oxford Univ. Press, 1995.

Daizhan Cheng (F'06) received the Ph.D. degree from Washington University, St. Louis, MO, in 1985.

He is currently a Professor with the Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His current research interests include nonlinear systems, numerical methods, and complex systems.

Prof. Cheng is a Fellow of the International Federation of Automatic Control.

Hongsheng Qi (M'10) received the Ph.D. degree in systems theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Assistant Professor with the Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. His current research interests include nonlinear control and complex systems.

Zhiqiang Li received the Ph.D. degree from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Lecturer with the Department of Mathematics and Information Science, Henan University of Economics and Law, Zhengzhou, China. His current research interests include nonlinear systems and complex systems.