Optimal Control of Logical Control Networks

Yin Zhao, Zhiqiang Li, *Student Member, IEEE*, and Daizhan Cheng, *Fellow, IEEE*

***Abstract*—This paper considers the infinite horizon optimal control of logical control networks, including Boolean control networks as a special case. Using the framework of game theory, the optimal control problem is formulated. In the sight of the algebraic form of a logical control network, its cycles can be calculated algebraically. Then the optimal control is revealed over a certain cycle. When the games, using memory $\mu > 1$ (which means the players only consider previous μ steps' action at each step), are considered, the higher order logical control network is introduced and its algebraic form is also presented, which corresponds to a conventional logical control network (i.e., $\mu = 1$). Then it is proved that the optimization technique developed for conventional logical control networks is also applicable to this μ-memory case.**

***Index Terms*—Boolean network, cycle, higher order logical control network, logical control network, optimal control.**

I. Introduction

In the investigation of cellular networks, Kauffman firstly introduced the Boolean network [1]. It has then caused an ever increasing interest in the study of Boolean networks, since it has been proved to be a useful tool in modeling of cell regulation [2]–[4], and also be used as models of some complex systems such as neural networks, social and economic networks [5], [6]. The first important topic is to find the topological structure of Boolean networks, such as fixed points, cycles, and basins of attractors, etc. [7]–[10]. Another challenging topic is its application to analyzing genetic networks [11]–[13]. To manipulate networks, the control of Boolean network is also a fundamental topic [14], [15].

A node of a Boolean network can take value 0 or 1. A logical network has the similar structure as the one of a Boolean network, but allows its nodes to take values from a finite set, say for a k-valued logical network its nodes can take values from $\{i/(k-1) | i = 0, 1, \cdots, k-1\}$. A k-valued logical network can approximate a real cellular regulatory network better than a Boolean network. We refer to [16] and [17] for k-valued logical networks.

The k-valued logical network is also an effective tool for solving some problems in game theory [18]–[22]. When the number of actions is 2, the Boolean control network is a proper model to describe the dynamics of an infinitely repeated game. Then the game is formulated as the optimal control problem of a Boolean control network.

Recently, a semi-tensor product approach to analysis and control of Boolean networks has been proposed. [23] is a proper reference for semi-tensor product. The framework and the structure analysis of Boolean networks are presented in [24] and [25]. Some fundamental control problems, such as controllability, observability, realization, and disturbance decoupling of Boolean control networks, are investigated in [26]–[28]. The method has also been extended to k-valued logical networks [29].

The purpose of this paper is to investigate the optimal control problem of logical control networks. The problem is motivated by the Boolean games and it is based on the framework of game theory. This model was firstly proposed in [30], in which the infinitely repeated game between a human and a machine based on the standard prisoners' dilemma (PD) model is considered. The purpose is to find a best human strategy when the machine strategy is fixed. We give an example to describe this.

Example 1.1: We consider the model of infinitely repeated PD [30].

The player 1 is a machine and player 2 is a person. Their actions can be

$$0: \quad \text{the player cooperates with the partner;}$$
$$1: \quad \text{the player defects the partner.}$$

TABLE I
PAYOFF BI-MATRIX

$P_1 \backslash P_2$	0	1
0	3,3	0,5
1	5,0	1,1

The payoff bi-matrix is assumed to be as shown in Table I.

Assume the machine strategy, which depends on the μ-memory, is fixed. It is defined as

$$m(t+1) = f_m\left(m(t-\mu+1), m(t-\mu+2), \cdots, m(t), h(t-\mu+1), h(t-\mu+2), \cdots, h(t)\right), \quad (1)$$

where the machine strategy $m(t)$ is considered as the state, f_m is a fixed logical function. The human strategy, $h(t)$, is considered as the control.

Denote by $p_h(t) := p_h(m(t), h(t))$ the payoff of the human. Our purpose is to design an optimal control to maximize the superior limit of the average of human payoff as time goes to infinity

$$J = \overline{\lim_{T\to\infty}} \frac{1}{T} \sum_{t=1}^{T} p_h(t). \quad (2)$$

Manuscript received December 29, 2009; revised October 18, 2010; accepted November 05, 2010. Date of publication November 15, 2010; date of current version August 03, 2011. This work is supported in part by the National Natural Science Foundation of China under Grant NNSF 60674022,Grant 60736022, Grant 60221301, and Grant 61074114 of China. Recommended by Associate Editor M. Egerstedt.

Y. Zhao and D. Cheng are with the Key Laboratory of Systems & Control, Institute of Systems Science, Academy of Mathematics & Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhaoyin@amss.ac.cn; dcheng@iss.ac.cn).

Z. Li was with the Key Laboratory of Systems & Control, Institute of Systems Science, Academy of Mathematics & Systems Science, Chinese Academy of Sciences, Beijing 100190, China. He is now with Department of Mathematics & Information Science, Henan University of Economics & Law, Zhengzhou 450002, China (e-mail: lizhiqiang@amss.ac.cn).

Digital Object Identifier 10.1109/TAC.2010.2092290

Note that when each player of a game has more than two possible actions, it is necessary to consider a k-valued logical control network.

Using the semi-tensor product approach, we first investigate the topological structure of logical control networks. Particularly, finding cycles via network algebraic form is investigated in detail. Then the optimal control is revealed by comparing average values over cycles. As a game using μ memory is considered, we propose higher order logical control network as its model [31]. To solve the optimal control problem of such a network, its algebraic form is reduced to one of conventional logical networks. Then it is proved that the approach developed for conventional logical network remains applicable.

The rest of the paper is organized as follows: Section II gives a formulation for the problem. In Section III we first discuss the topological structure of logical control networks. Then it is revealed that the optimal control is achieved on an optimal trajectory, which converges to a cycle, and hence it is computable. Section IV considers the optimal control of logical control networks. The higher order logical control networks, which correspond to the games using μ-memory, are discussed in Section V. Section VI is a brief conclusion.

II. Problem Formulation

For statement ease, we first introduce some notations.

- $\mathbf{1}_k := (\underbrace{1\ 1\ \cdots\ 1}_{k})^T$.
- $\mathcal{D}_k := \{0, 1/(k-1), 2/(k-1), \cdots, 1\}$; $\mathcal{D} := \mathcal{D}_2 = \{0, 1\}$.
- δ_n^i: the i-th column of the identity matrix I_n.
- $\Delta_n := \{\delta_n^i | i = 1, \cdots, n\}$; $\Delta_2 := \Delta$.
- $\mathrm{Col}_i(A)$: the i-th column of matrix A. Denoted by $\mathrm{Col}(A)$ the set of columns of A
- $\mathrm{Blk}_i(A)$: the i-th $n \times n$ block of $n \times mn$ matrix A.
- Denote by $M_{n\times r}$ the set of $n \times r$ matrices. $L \in M_{n\times r}$ is called a logical matrix if $\mathrm{Col}_i(L)$, $i = 1, \cdots, r$ have the form of δ_n^k. That is,

$$\mathrm{Col}(L) \subset \Delta_n.$$

  Denote by $\mathcal{L}_{n\times r}$ the set of $n \times r$ logical matrices.
- If $L \in \mathcal{L}_{n\times r}$, by definition it can be expressed as $L = [\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_r}]$. For the sake of compactness, it is briefly denoted as $L = \delta_n[i_1, i_2, \cdots, i_r]$. Its columns set can also be denoted as $\mathrm{Col}(L) = \delta_n\{i_1, i_2, \cdots, i_r\}$.
- $W_{[n,m]}$ is a swap matrix. We refer to [23] for its definition and properties.

Definition 2.1: [23]: Let $A \in M_{m\times n}$ and $B \in M_{p\times q}$ and denote the least common multiplier of n and p by $t = \mathrm{lcm}(n, p)$. Then the semi-tensor product of A and B is defined as

$$A \ltimes B := (A \otimes I_{t/n})(B \otimes I_{t/p}). \tag{3}$$

Remark 2.2: It is obvious that the semi-tensor product of matrices is a generalization of conventional matrix product, thus, the symbol $\ltimes$ can be omitted hereafter. We refer to [24] for the case when $n(p)$ is a multiplier of $p(n)$, and for some numerical examples. We also refer to [23] for its properties. In fact, all major properties of conventional matrix product remain unchanged under this generalization.

The dynamics of a Boolean network can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \cdots, x_n(t)) \\ x_2(t+1) = f_2(x_1(t), \cdots, x_n(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \cdots, x_n(t)) \end{cases} \tag{4}$$

where $x_i \in \mathcal{D}$ and $f_i : \mathcal{D}^n \to \mathcal{D}$ are logical functions.

A Boolean control network is a Boolean network with input(s) and output(s). Throughout this paper the outputs are not concerned, then its state dynamics can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)) \\ x_2(t+1) = f_2(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)) \end{cases} \tag{5}$$

where $x_i, u_i \in \mathcal{D}$, with x_i state variables, u_i controls and f_i logical functions.

We call the system (4) and (5) the conventional Boolean network and conventional Boolean control network, respectively. As mentioned in the introduction, they have been discussed widely.

A k-valued logical network (or k-valued logical control network) has the same form as (4) (or (5), respectively), except that the state variables x_i (and the controls u_i) take values from $\mathcal{D}_k$.

To get a matrix expression of the dynamics of a Boolean network, called the algebraic form of the network, we identify $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$. Then in vector form we have $x_i, u_i \in \Delta$. Under this vector form we denote $x = \ltimes_{i=1}^n x_i$. Then referring to [24], there exists a unique $L \in \mathcal{L}_{2^n\times 2^n}$, such that (4) can be expressed as

$$x(t+1) = Lx(t). \tag{6}$$

Equation (6) is called the algebraic form of (4).

For Boolean control network, we also denote $u = \ltimes_{i=1}^m u_i$. Then there exists a unique $L \in \mathcal{L}_{2^n\times 2^{n+m}}$ such that the algebraic form of (5) is expressed as

$$x(t+1) = Lu(t)x(t). \tag{7}$$

Similarly, to get the algebraic form of a k-valued logical (control) network, we identify

$$\frac{i}{k-1} \sim \delta_k^{k-i}, \quad i = 0, 1, \cdots, k-1.$$

Then one sees that the algebraic form of a k-valued logical (control) network is the same as (6) ((7)) except that $L \in \mathcal{L}_{k^n\times k^n}$ ($L \in \mathcal{L}_{k^n\times k^{m+n}}$).

We refer to [24], [25], [29] for calculating the algebraic forms.

When the updated values of the nodes of a logical network depend only on the current values, we have the aforementioned dynamic equations. But if the updated values depend on the past μ values, a μ-th order dynamic model is required.

A μ-th order logical network is described as

$$\begin{cases} x_1(t+1) = f_1\left(x_1(t),\cdots,x_n(t),\cdots,x_1(t-\mu+1),\cdots,\right. \\ \qquad\qquad \left. x_n(t-\mu+1)\right) \\ x_2(t+1) = f_2\left(x_1(t),\cdots,x_n(t),\cdots,x_1(t-\mu+1),\cdots,\right. \\ \qquad\qquad \left. x_n(t-\mu+1)\right) \\ \vdots \\ x_n(t+1) = f_n\left(x_1(t),\cdots,x_n(t),\cdots,x_1(t-\mu+1),\cdots,\right. \\ \qquad\qquad \left. x_n(t-\mu+1)\right) \end{cases} \tag{8}$$

Similarly, a μ-th order logical control network has its state dynamics as

$$\begin{cases} x_1(t+1) = f_1\left(x_1(t),\cdots,x_n(t)\cdots,x_1(t-\mu+1),\cdots,\right. \\ \qquad\qquad x_n(t-\mu+1),u_1(t),\cdots,u_m(t),\cdots, \\ \qquad\qquad \left. u_1(t-\mu+1),\cdots,u_m(t-\mu+1)\right) \\ \vdots \\ x_n(t+1) = f_n\left(x_1(t),\cdots,x_n(t),\cdots,x_1(t-\mu+1),\cdots,\right. \\ \qquad\qquad x_n(t-\mu+1),u_1(t),\cdots,u_m(t),\cdots, \\ \qquad\qquad \left. u_1(t-\mu+1),\cdots,u_m(t-\mu+1)\right) \end{cases} \tag{9}$$

For the system (7) (or (9)), consider the objective function as

$$J(u) = \overline{\lim_{T\to\infty}} \frac{1}{T}\sum_{t=1}^{T} P\left(x(t),u(t)\right) \tag{10}$$

where P is a function from $\mathcal{D}_k^{n+m}$ to $\mathbb{R}$. Our purpose is to find an optimal control $u^*(t)$ to maximize $J(u)$, that is

$$J^*(u) := J(u^*) = \max_u J(u).$$

III. Topological Structure of Logical Control Networks

To deal with the optimal control of a logical control network, its topological structure needs to be considered first. Particularly, later on, we will see that the optimal trajectory will converge to a certain cycle, so calculating cycles becomes a key issue.

A k-valued control network can be expressed as (5) with $x_i, u_i \in \mathcal{D}_k$. Its algebraic form is (7) with $L \in \mathcal{L}_{k^n\times k^{n+m}}$, and $x_i, u_i \in \Delta_k$.

Denote the control-state (product) space as

$$\mathcal{S}=\{(U,X)|U=(u_1,\cdots,u_m)\in\mathcal{D}_k^m, X=(x_1,\cdots,x_n)\in\mathcal{D}_k^n\}.$$

Using vector form, we set $s(t) = u(t)\ltimes x(t)$, then $s(t) \in \Delta_{k^{m+n}}$. Later on, we will see that the optimal control will be reached at an optimal trajectory which converges to a fixed cycle in $\mathcal{S}$. We, therefore, need to investigate the cycles in the control-state space $\mathcal{S}$.

In vector form, the graph in $\mathcal{S}$ has $\delta_{k^{m+n}}^i$, $i = 1,\cdots,k^{m+n}$ as its vertices. An edge $\overrightarrow{\delta_{k^{m+n}}^i\delta_{k^{m+n}}^j}$, which is briefly denoted as $\delta_{k^{m+n}}(\overrightarrow{ij})$, exists if $s(t+1) = \delta_{k^{m+n}}^j$ can be reached from $s(t) = \delta_{k^{m+n}}^i$ by choosing a matching $u(t+1)$. A cycle is a path $\{\delta_{k^{m+n}}^{i_1} \to \delta_{k^{m+n}}^{i_2} \to \cdots \to \delta_{k^{m+n}}^{i_d} \to \cdots\}$, in which there exists a integer d such that $\delta_{k^{m+n}}^{i_j} = \delta_{k^{m+n}}^{i_{j+d}}$, the smallest such d is called the length of the cycle.

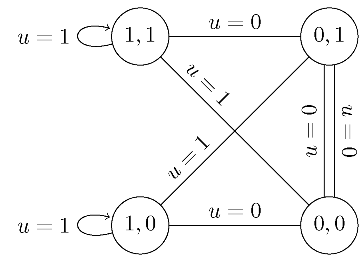


Fig. 1. State transfer graph.

For a cycle C of length d, because $s(t) = \delta_{k^{m+n}}^\ell$ can be decomposed uniquely to $u(t)x(t) = \delta_{k^m}^i\delta_{k^n}^j$, the cycle can be described as

$$C = \left\{\left(\delta_{k^m}^{i(t)},\delta_{k^n}^{j(t)}\right) \to \left(\delta_{k^m}^{i(t+1)},\delta_{k^n}^{j(t+1)}\right) \to \cdots \to \left(\delta_{k^m}^{i(t+d-1)},\delta_{k^n}^{j(t+d-1)}\right)\right\}.$$

For compactness, we denote it as

$$C = \delta_{k^m}\times\delta_{k^n}\left\{(i(t),j(t)) \to (i(t+1),j(t+1)) \to \cdots \to (i(t+d-1),j(t+d-1))\right\}. \tag{11}$$

Then we have the following results.

Proposition 3.1: An edge $\delta_{k^{m+n}}(\overrightarrow{ij})$ exists, if and only if

$$\mathrm{Col}_i(L) = \delta_{k^n}^\ell, \quad \text{where} \quad \ell = j(mod\, k^n). \tag{12}$$

Proof: By definition, the edge $\delta_{k^{m+n}}(\overrightarrow{ij})$ exists, if and only if there exists $u(t+1)$ such

$$u(t+1)L\delta_{k^{m+n}}^i = \delta_{k^{m+n}}^j. \tag{13}$$

It is easy to check that $L\delta_{k^{m+n}}^i = \mathrm{Col}_i(L)$, thus (13) yields

$$u(t+1)\mathrm{Col}_i(L) = \delta_{k^{m+n}}^j. \tag{14}$$

Note that $\delta_{k^{m+n}}^j$ can be factorized uniquely into $\delta_{k^m}^\xi\delta_{k^n}^\ell$, where $j = (\xi-1)k^n+\ell$. The proposition is proved. □

Example 3.2: Assume a Boolean control network is

$$x(t+1) = Lu(t)x(t) \tag{15}$$

where $u(t), x(t) \in \Delta$, and

$$L = \delta_2[1,2,2,1].$$

Note that $\delta_4^1 \sim (1,1), \delta_4^2 \sim (1,0), \delta_4^3 \sim (0,1), \delta_4^4 \sim (0,0)$, then we can get the state transfer graph as Fig. 1.

From Fig. 1 we can see that (1,1) and (1,0) are fixed points, $\{(0,1) \to (0,0)\}$, $\{(0,1) \to (1,0) \to (0,0)\}$, $\{(1,1) \to (0,1) \to (0,0)\}$, $\{(0,0) \to (1,1) \to (0,1) \to (1,0)\}$, $\{(1,1) \to (1,1) \to (0,1) \to (0,0)\}$, $\{(1,0) \to (1,0) \to (0,0) \to (0,1)\}$ are all the cycles with length less than or equal to 4.

In simple case, the fixed points and cycles can be found from the control-state graph directly. But when m and n are larger, it is difficult to draw the graph as above. Thus, we need to develop formulas to compute all the cycles algebraically.

From (7), we have

$$\begin{aligned} x(t+d) &= Lu(t+d-1)x(t+d-1) \\ &= Lu(t+d-1)Lu(t+d-2) \\ &\quad \cdots Lu(t+1)Lu(t)x(t) \\ &= L(I_{k^m} \otimes L)u(t+d-1)u(t+d-2) \\ &\quad \ltimes Lu(t+d-3)Lu(t+d-4)\cdots Lu(t)x(t) \\ &:= L_d \left(\ltimes_{\ell=1}^{d} u(t+d-\ell)\right) x(t) \qquad (16) \end{aligned}$$

where

$$L_d = \prod_{i=1}^{d} (I_{k^{(i-1)m}} \otimes L) \in \mathcal{L}_{k^n \times k^{dm+n}}.$$

Before calculating the cycles, we need some notations.

- Let $d \in \mathbb{Z}_+$, $\mathcal{P}(d)$ is the set of proper factors of d.
- Let $i, k, m \in \mathbb{Z}_+$, then

$$\theta_k^m(i,d) := \left\{(j,\ell) | \ell \in \mathcal{P}(d) \text{ and } j < k^{\ell m} \text{ such that } \delta_{k^{dm}}^i = \left(\delta_{k^{\ell m}}^j\right)^{\frac{d}{\ell}}\right\}. \qquad (17)$$

For finding $\theta_k^m(i,d)$, we can decompose $\delta_{k^{dm}}^i$ to $\ltimes_{\alpha=1}^{d} \delta_{k^m}^{i_\alpha}$ to see whether $\{i_1, i_2, \cdots, i_d\}$ is cyclic. We give an example to explain these notations.

Example 3.3:

- Let $d=6$. Then $\mathcal{P}(d) = \{1,2,3\}$.
- Let $m, k, d \in \mathbb{Z}_+$ be given. Then using an obvious formula $\delta_{k^\alpha}^a \delta_{k^\beta}^b = \delta_{k^{\alpha+\beta}}^{(a-1)k^\beta + b}$, there exists at most one j for every $\ell \in \mathcal{P}(d)$ such that $(j,\ell) \in \theta_k^m(i,d)$.

  Say, $m = n = k = 2$, $d = 6$.

  —Let $i = 1$, then $\delta_{k^{dm}}^i = \delta_{2^{12}}^1 = (\delta_{2^2}^1)^6 = (\delta_{2^4}^1)^3 = (\delta_{2^6}^1)^2$.
  Hence, $\theta_2^2(1,6) = \{(1,1),(1,2),(1,3)\}$.

  —Let $i=2$, $\delta_{2^{12}}^2 = (\delta_{2^2}^1)^5 \delta_{2^2}^2$, thus there is no solution.
  Hence, $\theta_2^2(2,6) = \emptyset$.

  —Let $i = 2^6+2$, $\delta_{2^{12}}^{2^6+2} = (\delta_{2^2}^1 \delta_{2^2}^1 \delta_{2^2}^2)^2 = (\delta_{2^6}^2)^2$.
  So $\theta_2^2(2^6+2,6) = \{(2,3)\}$.

In the following we simply use $\theta(i,d)$ for $\theta_k^m(i,d)$, the default k and m are assumed to be the type of logic and the number of inputs.

Theorem 3.4: The number of cycles of length d in the control-state graph of k-valued logical control network (7) is inductively determined by

$$N_d = \frac{1}{d} \sum_{i=1}^{k^{dm}} T\left(\mathrm{Blk}_i(L_d)\right) \qquad (18)$$

where

$$T\left(\mathrm{Blk}_i(L_d)\right) = \mathrm{Tr}\left(\mathrm{Blk}_i(L_d)\right) - \sum_{(j,\ell)\in\theta(i,d)} T\left(\mathrm{Blk}_j(L_\ell)\right)$$

Proof: Each cycle in $\mathcal{S}$ is a product of cycles in state space and control space, thus, we look for the cycle in state space first.

If $x(t)$ is in a cycle in state space of length d, from (16) we have

$$x(t) = L_d \left(\ltimes_{\ell=1}^{d} u(t+d-\ell)\right) x(t).$$

If $u(t+d-1), \cdots, u(t)$ are fixed, say $\ltimes_{\ell=1}^{d} u(t+d-\ell)) = \delta_{k^{dm}}^i$, then

$$x(t) = \mathrm{Blk}_i(L_d)x(t).$$

If $x(t) = \delta_{k^n}^j$, that means the (j,j)-th element of $\mathrm{Blk}_i(L_d)$ is 1. So the cycle with length d in state space under the given controls $u(t+d-1), \cdots, u(t)$ is $\{x(t) \to Lu(t)x(t) \to L_2u(t+1)u(t)x(t) \to \cdots \to L_d u(t+d-1)\cdots u(t)x(t)\}$. Thus, multiplying the cycle and the given u, we obtain a cycle of length d in control-state space. Hence, the number of length d cycles including multi-fold ones is $(1/d)\sum_{i=1}^{k^{dm}} \mathrm{Tr}(\mathrm{Blk}_i(L_d))$.

It is obvious that if ℓ is a proper factor of d, and $x(t)$ is in the cycle of length ℓ under $\widetilde{u}(t+\ell-1)\cdots\widetilde{u}(t) = \delta_{k^{\ell m}}^j$ and the cycle of length d under $u(t+d-1)\cdots u(t) = \delta_{k^{dm}}^i$ respectively, then we can obtain the same cycle in control-state space, if and only if $\delta_{k^{dm}}^i = (\delta_{k^{\ell m}}^j)^{d\ell}$. Moving away these multi-fold cycles, we obtain (18). □

From the proof of Theorem 3.4, we can see that the cycles can be found by the following algorithm:

1) For length d, calculate L_d.
2) Denote by $\ell_{jj}^{i,d}$ the (j,j)-th element of $\mathrm{Blk}_i(L_d)$. For $1 \le i \le k^{dm}$, check the diagonal elements of $\mathrm{Blk}_i(L_d)$, if $\ell_{jj}^{i,d} = 1$, and $\ell_{jj}^{\alpha,\ell} = 0$ for all $(\alpha,\ell) \in \theta(i,d)$, then $x(t) = \delta_k^j$ is in the cycle of length d under $u(t+d-1)\cdots u(t) = \delta_{k^{dm}}^i$. So the cycle is $\{u(t)x(t) \to u(t+1)Lu(t)x(t) \to \cdots \to u(t+d-1)L_d u(t+d-1)\cdots u(t)x(t)\}$.
3) If $i \le k^d - 1$, then set $i = i+1$, and return to 2); else set $d = d+1$, and return to 1).

Definition 3.5: A cycle $C = \delta_{k^m} \times \delta_{k^n}\{(i(t),j(t)) \to (i(t+1), j(t+1)) \to \cdots \to (i(t+d-1), j(t+d-1))\}$ is called a simple cycle, if it satisfies

$$j(\xi) \neq j(\ell), \quad t \le \xi < \ell \le t+d-1. \qquad (19)$$

Example 3.6: Recall Example 3.2. Since

$$L_1 = L = \delta_2[1,2,2,1]$$

we have $\mathrm{Tr}(\mathrm{Blk}_1(L_1)) = 2$, $\mathrm{Tr}(\mathrm{Blk}_2(L_1)) = 0$. Hence δ_2^1 and δ_2^2 are fixed points under control $u = \delta_2^1$. It follows that the fixed points in control-state space are

$$\delta_2 \times \delta_2\{(1,1)\}, \quad \delta_2 \times \delta_2\{(1,2)\}$$

which are simple ones. Next, since

$$L_2 = L(I_2 \otimes L) = \delta_2[1,2,2,1,2,1,1,2]$$

we have $\mathrm{Tr}(\mathrm{Blk}_1(L_2)) = \mathrm{Tr}(\mathrm{Blk}_4(L_2)) = 2$, $\mathrm{Tr}(\mathrm{Blk}_2(L_2)) = \mathrm{Tr}(\mathrm{Blk}_3(L_2)) = 0$, $\delta_4^1 = \delta_2^1\delta_2^1$, $\delta_4^4 = \delta_2^2\delta_2^2$ so

$$\begin{aligned} T\left(\mathrm{Blk}_1(L_2)\right) &= \mathrm{Tr}\left(\mathrm{Blk}_1(L_2)\right) - T\left(\mathrm{Blk}_1(L_1)\right) = 0 \\ T\left(\mathrm{Blk}_4(L_2)\right) &= \mathrm{Tr}\left(\mathrm{Blk}_4(L_2)\right) - T\left(\mathrm{Blk}_2(L_1)\right) = 2, \end{aligned}$$

$T(\mathrm{Blk}_2(L_2)) = T(\mathrm{Blk}_3(L_2)) = 0$, thus, $N_2 = 1$. δ_2^1 and δ_2^2 are in cycles of length 2 under $u(t+1)u(t) = \delta_2^2\delta_2^2$. We then can obtain a cycle of length 2 in control-state space as

$$\delta_2 \times \delta_2 \{(2,1) \to (2,2)\}$$

which is also simple. Consider

$$\begin{aligned} L_3 &= L(I_2 \otimes L)(I_4 \otimes L) = L_2(I_4 \otimes L) \\ &= \delta_2[1,2,2,1,2,1,1,2,2,1,1,2,1,2,2,1]. \end{aligned}$$

Since $\mathrm{Tr}(\mathrm{Blk}_1(L_3)) = \mathrm{Tr}(\mathrm{Blk}_4(L_3)) = \mathrm{Tr}(\mathrm{Blk}_6(L_3)) = \mathrm{Tr}(\mathrm{Blk}_7(L_3)) = 2$, so $T(\mathrm{Blk}_4(L_3)) = T(\mathrm{Blk}_6(L_3)) = T(\mathrm{Blk}_7(L_3)) = 2$, $T(\mathrm{Blk}_i(L_3)) = 0$, $i = 1, 2, 3, 5, 8$, we have, $N_3 = 2$. δ_2^1 and δ_2^2 are in cycles of length 3 under $u(t+2)u(t+1)u(t) = \delta_8^4 = \delta_2^1\delta_2^2\delta_2^2$, $\delta_8^6 = \delta_2^2\delta_2^1\delta_2^2$, and $\delta_8^7 = \delta_2^2\delta_2^2\delta_2^1$. We then can obtain the cycles of length 3 in control-state space as

$$\begin{aligned} &\delta_2 \times \delta_2 \{(1,1) \to (2,1) \to (2,2)\}, \\ &\delta_2 \times \delta_2 \{(2,1) \to (1,2) \to (2,2)\}. \end{aligned}$$

Finally, since

$$\begin{aligned} L_4 &= L_3(I_8 \otimes L) \\ &= \delta_2[1,2,2,1,2,1,1,2,2,1,1,2,1,2,2,1, \\ &\quad 2,1,1,2,1,2,2,1,1,2,2,1,2,1,1,2], \end{aligned}$$

we have $\mathrm{Tr}(\mathrm{Blk}_i(L_4)) = 2$, $i = 1, 4, 6, 7, 10, 11, 13, 16$, so $T(\mathrm{Blk}_i(L_4)) = 2$ for $i = 4, 6, 7, 10, 11, 13$, otherwise $T(\mathrm{Blk}_i(L_4)) = 0$, and hence $N_4 = 3$. δ_2^1 and δ_2^2 are in cycles of length 4 under $u(t+3)u(t+2)u(t+1)u(t) = \delta_{16}^4 = \delta_2^1\delta_2^1\delta_2^2\delta_2^2$, $\delta_{16}^6 = \delta_2^1\delta_2^2\delta_2^1\delta_2^2$, $\delta_{16}^7 = \delta_2^1\delta_2^2\delta_2^2\delta_2^1$, $\delta_{16}^{10} = \delta_2^2\delta_2^1\delta_2^1\delta_2^2$, $\delta_{16}^{11} = \delta_2^2\delta_2^1\delta_2^2\delta_2^1$, and $\delta_{16}^{13} = \delta_2^2\delta_2^2\delta_2^1\delta_2^1$. Then we can obtain the cycles of length 4 in control-state space as

$$\begin{aligned} &\delta_2 \times \delta_2 \{(1,1) \to (2,1) \to (1,2) \to (2,2)\}, \\ &\delta_2 \times \delta_2 \{(1,2) \to (1,2) \to (2,2) \to (2,1)\}, \\ &\delta_2 \times \delta_2 \{(1,1) \to (1,1) \to (2,1) \to (2,2)\}. \end{aligned}$$

This result is the same as what we observed from the graph in Example 3.2.

IV. Optimal Control of Logical Control Networks

In this section we consider the optimal control and the optimal trajectory of logical control networks. Similar to [30], we can prove the following result:

Theorem 4.1: For the k-valued control network (5) with the objective function (10), there exists an optimal control $u^*(t)$ such that the objective function is maximized and the trajectory of $s^*(t) = u^*(t)x^*(t)$ will become periodic after a finite time.

Remark 4.2: The difference between the above theorem and Theorem 2.1 in [30] for the case of 1-memory is that $x(t)$ and $u(t)$ here are multi-dimensional and have k values. But after converting them to the graph, there is no essential difference. So the proofs are the same. In the sequel, using the matrix expression of logical functions, we can give a method to find the optimal trajectory and obtain a G^* which is called the optimal control matrix, such that

$$u^*(t+1) = G^* u^*(t) x^*(t).$$

Proposition 4.3: The limit

$$J(u^*) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} P\left(x^*(t), u^*(t)\right) \tag{20}$$

always exits.

Proof: Consider the system (5). According to Theorem 4.1, an optimal trajectory will converge to an attractor. As a limit, $J(u^*)$ is the average of an attractor (fixed point or cycle). □

For a cycle $C = \delta_{k^m} \times \delta_{k^n}\{(i(t), j(t)) \to (i(t+1), j(t+1)) \to \cdots \to (i(t+d-1), j(t+d-1))\}$, denote

$$\begin{aligned} P(C) &= \frac{1}{d} \sum_{s(t)\in C} P\left(u(t), x(t)\right) \\ &= \frac{1}{d} \sum_{\ell=1}^{d} P\left(\delta_{k^m}^{i(t+\ell-1)}, \delta_{k^n}^{j(t+\ell-1)}\right). \end{aligned} \tag{21}$$

Proposition 4.4: Any cycle C contains a simple cycle C_s such that

$$P(C_s) \geq P(C) \tag{22}$$

Proof: Denote by $C = \delta_{k^m} \times \delta_{k^n}\{(i(t), j(t)) \to (i(t+1), j(t+1)) \to \cdots \to (i(t+d-1), j(t+d-1))\}$ an arbitrary cycle. If it is a simple cycle, the result is trivial. Otherwise, assume $\delta_{k^n}^{j(\xi)} = \delta_{k^n}^{j(\ell)}$, $\xi < \ell$, and $C_1 = \delta_{k^m} \times \delta_{k^n}\{(i(\xi), j(\xi)) \to \cdots (i(\ell-1), j(\ell-1))\}$ is a simple cycle. If $P(C_1) \geq P(C)$, we are done.

Otherwise, we remove C_1, then the remains form a new cycle C_1', because $L\delta_{k^m}^{i(\xi-1)}\delta_{k^n}^{j(\xi-1)} = \delta_{k^n}^{i(\xi)} = \delta_{k^n}^{i(\ell)}$. Now $P(C_1') > P(C)$. If C_1' is a simple cycle, we are done.

Otherwise, we can find a simple cycle C_2 such that either it satisfies (22) or remove it. Continuing this process, we can finally find a simple cycle C_s such that (22) holds. □

Denote by $R(x)$ the reachable set of the a state x, we say a cycle $C \subset R(x)$ if any element of C is in $R(x)$.

Definition 4.5: Giving the initial state x_0, a cycle C^* is called an optimal cycle if

$$C^* \in \arg\max_{C\subset R(x_0)} P(C). \tag{23}$$

By (16), at dth step, the initial state x_0 can reach

$$R_d(x_0) = \left\{u(d)L_d \ltimes_{\ell=1}^{d} u(d-\ell)x_0 | \forall u(\ell) \in \Delta_{k^m}, 0 \leq \ell \leq d\right\},$$

if $x_0 = \delta_{k^n}^{j(0)}$,

$$R_d(x_0) = \left\{u(d)\mathrm{Col}_\ell(L_d) | \forall u(d) \in \Delta_{k^m}, \ell = j(0)(mod\ k^n)\right\}.$$

If $\delta_{k^m}^i \delta_{k^n}^j$ is reached from x_0 at d-th step, $d > k^n$, the path from the initial state to $\delta_{k^m}^i \delta_{k^n}^j$ must at least pass a state twice. Similar to the proof of Proposition 4.4, we can reduce the path, finally,

$\delta_{k^m}^i \delta_{k^n}^j$ can be reached from x_0 at d'-th step, $1 \leq d' \leq k^n$. Thus,

$$R(x_0) = \cup_{d=1}^{k^n} R_d(x_0). \tag{24}$$

Since from x_0, $\{ux_0 | \forall u \in \mathcal{D}_k\}$ can be reached firstly, and for each $\delta_{k^m}^i \delta_{k^n}^j = \delta_{k^{m+n}}^\alpha$, it can reach $\{u\text{Col}_\alpha(L) | \forall u \in \mathcal{D}_k\}$. Thus, simply implementing DFS (Depth-First-Search) algorithm, we can get the reachable set.

According to the above argument, we can find the optimal cycle C^* only from all the simple cycles contained in $R(x_0)$. Denote the shortest path from initial state to C^* by

$$\delta_{k^m}^{i(0)} \delta_{k^n}^{j(0)} \to \delta_{k^m}^{i(1)} \delta_{k^n}^{j(1)} \to \cdots \to \delta_{k^m}^{i(T_0-1)} \delta_{k^n}^{j(T_0-1)} \to C^* \tag{25}$$

where

$$C^* = \delta_{k^m} \times \delta_{k^n} \{(i(T_0), j(T_0)) \to \cdots \to (i(T_0+d-1), j(T_0+d-1))\}.$$

We call (25) the optimal trajectory.

Next, we will prove the existence of the optimal control matrix G^*.

Theorem 4.6: Consider the k-valued logical control network (5) with the objective function (10). Let the optimal trajectory be (25), and the optimal control be $u^*(t)$. Then there exists a logical matrix $G^* \in \mathcal{L}_{k^m \times k^{m+n}}$, satisfying

$$\begin{cases} x^*(t+1) = Lu^*(t)x^*(t) \\ u^*(t+1) = G^* u^*(t) x^*(t). \end{cases} \tag{26}$$

Proof: According to Proposition 4.4, we can find an optimal cycle just from all simple cycles. Because the length of a simple cycle can not be greater than k^n, assume the initial state of a trajectory is $\delta_{k^n}^{j(0)}$, we can find all cycles with length less than or equal to k^n which can be reached from the initial state, and then find out the optimal trajectory (25). It is easy to know that $T_0 + d \leq k^{m+n}$, so we can get $T_0 + d$ columns of the optimal control matrix G^*, which satisfy

$$\text{Col}_i(G^*) = \begin{cases} \delta_{k^m}^{i(\ell+1)} & i = (i(\ell) - 1) k^n + j(\ell), \\ & \ell \leq T_0 + d - 2 \\ \delta_{k^m}^{i(T_0)} & i = (i(T_0 + d - 1) - 1) k^n \\ & \quad + j(T_0 + d - 1) \end{cases} \tag{27}$$

and the other columns of G^* $(\text{Col}(G^*) \subset \Delta_{k^m})$ can be arbitrary. Thus G^* is constructed. □

Example 4.7: Recall Example 3.2 and Example 3.6 again. Set

$$P(u(t), x(t)) = u'(t) \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} x(t).$$

Assume the initial state $x_0 = \delta_2^2$, from the result of Example 3.6 we can see $C^* = \delta_2 \times \delta_2 \{(2,1) \to (2,2)\}$ is obviously the optimal cycle. Choosing $u(0) = \delta_2^2$, the optimal cycle and the shortest path from δ_2^2 to the cycle is

$$\delta_2 \times \delta_2 \{(2,2) \to (2,1)\}.$$

Hence, $G^* = \delta_2[i, j, 2, 2]$, i, j can be either 1 or 2.

TABLE II
PAYOFF BI-MATRIX

$P_1 \backslash P_2$	L	M	R
L	$3,3$	$0,4$	$9,2$
M	$4,0$	$4,4$	$5,3$
R	$2,9$	$3,5$	$6,6$

Example 4.8: We consider the following infinitely repeated game. Both of player 1 and player 2 have three actions, {L, M, R}. The payoff bi-matrix is assumed to be the Table II.

It is easy to check that (M, M), which means player 1 choose M and player 2 also choose M, is the unique Nash equilibrium of the one-stage game, but it is obvious that (R, R) is more efficient than (M, M). In the infinitely repeated game, assume player 2's strategy is fixed to play R in the first stage, in the t-th stage, if the outcome in the $(t-1)$-th stage is (R, R) then plays R, otherwise, plays M. This strategy is called the "trigger strategy" [32].

Denote $L \sim 1$, $M \sim 0.5$, $R \sim 0$. The above game can be rewrote as

$$x(t+1) = Lu(t)x(t) \tag{28}$$

where

$$L = \delta_3[2, 2, 2, 2, 2, 2, 2, 2, 3],$$

$x(t) \in \Delta_3$, as the state, is the action of player 2 at t-th stage; $u(t) \in \Delta_3$, as the control, is the action of player 1 at t-th stage.

As we know, the trigger strategy is the Nash equilibrium of an infinitely repeated finite game in which the payoff function is

$$(1-\delta) \sum_{t=1}^{\infty} \delta^{t-1} \pi_t,$$

where π_t is payoff at t-th stage and δ is discount factor, when δ is sufficiently close to one [32].

Ignoring the discount factor, our payoff functions for player 1 and player 2 are

$$J_1 = \overline{\lim_{T\to\infty}} \frac{1}{T} \sum_{t=1}^{T} P_1(x(t), u(t)),$$

$$J_2 = \overline{\lim_{T\to\infty}} \frac{1}{T} \sum_{t=1}^{T} P_2(x(t), u(t)),$$

where

$$P_1(x(t), u(t)) = u'(t) \begin{bmatrix} 3 & 0 & 9 \\ 4 & 4 & 5 \\ 2 & 3 & 6 \end{bmatrix} x(t),$$

$$P_2(x(t), u(t)) = u'(t) \begin{bmatrix} 3 & 4 & 2 \\ 0 & 4 & 3 \\ 9 & 5 & 6 \end{bmatrix} x(t).$$

An natural question is whether the trigger strategy is still the Nash equilibrium in this game.

Player 2 has adopted the trigger strategy, we want to find the best response for player 1. Then the question is converted to finding the optimal control of 3-valued logical control network (28), which maximizes J_1.

Now we calculate the cycles

$$L_1 = L = \delta_3[2,2,2,2,2,2,2,2,3]$$

thus ,$\mathrm{Tr}(\mathrm{Blk}_1(L_1)) = 1$, $\mathrm{Tr}(\mathrm{Blk}_2(L_1)) = 1$, $\mathrm{Tr}(\mathrm{Blk}_3(L_1)) = 2$, $N_1 = 4$. δ_3^2 is a fixed point under $u = \delta_3^i$, $i = 1, 2, 3$, and δ_3^3 is a fixed point under $u = \delta_3^3$. so the fixed points of system (28) are

$$\begin{aligned} &\delta_3 \times \delta_3 \{(1,2)\}, \quad \delta_3 \times \delta_3 \{(2,2)\}, \\ &\delta_3 \times \delta_3 \{(3,2)\}, \quad \delta_3 \times \delta_3 \{(3,3)\}. \end{aligned}$$

$$\begin{aligned} L_2 &= L(I_3 \otimes L) \\ &= \delta_3[2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2, \\ &\quad 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2, \\ &\quad 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 3], \end{aligned}$$

$\mathrm{Tr}(\mathrm{Blk}_i(L_2)) = 1, i = 1, \ldots, 8$, $\mathrm{Tr}(\mathrm{Blk}_9(L_2)) = 2$. For $\delta_9^1 = \delta_3^1\delta_3^1$

$$T(\mathrm{Blk}_1(L_2)) = \mathrm{Tr}(\mathrm{Blk}_1(L_2)) - \mathrm{Tr}(\mathrm{Blk}_1(L_1)) = 0.$$

Similarly, we obtain $T(\mathrm{Blk}_1(L_2)) = T(\mathrm{Blk}_5(L_2)) = T(\mathrm{Blk}_9(L_2)) = 0$, $T(\mathrm{Blk}_i(L_2)) = 1$, $i = 2, 3, 4, 6, 7, 8$. Thus $N_2 = 3$. δ_3^2 is in cycles of length 2 with $u(t+1)u(t) = \delta_9^i$, $i = 2, 3, 4, 6, 7, 8$. Then we can find the cycles of length 2 as

$$\begin{aligned} &\delta_3 \times \delta_3 \{(1,2) \to (2,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (3,2)\}, \\ &\delta_3 \times \delta_3 \{(2,2) \to (3,2)\}. \end{aligned}$$

$$L_3 = L(I_3 \otimes L)(I_9 \otimes L) = \delta_{81}[\underbrace{2,\cdots,2}_{80},3].$$

By (18) we have $T(\mathrm{Blk}_i(L_3)) = 1$, $i = 2, \cdots, 13, 15, \cdots, 26$, $T(\mathrm{Blk}_i(L_3)) = 0$, $i = 1$, 14, 27, and $N_3 = 8$. δ_3^2 is in cycles of length 3 with $u(t+2)u(t+1)u(t) = \delta_{27}^i$, $i = 2, \cdots, 13, 15, \cdots, 26$. Then we can find the cycles of length 3 as

$$\begin{aligned} &\delta_3 \times \delta_3 \{(1,2) \to (1,2) \to (2,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (3,2) \to (2,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (1,2) \to (3,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (3,2) \to (3,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (2,2) \to (2,2)\}, \\ &\delta_3 \times \delta_3 \{(2,2) \to (2,2) \to (3,2)\}, \\ &\delta_3 \times \delta_3 \{(1,2) \to (2,2) \to (3,2)\}, \\ &\delta_3 \times \delta_3 \{(2,2) \to (3,2) \to (3,2)\}. \end{aligned}$$

There are also lots of cycles of length greater than or equal to 4. But we have proved that to deal with the optimal control of this game, finding all the cycles of length less than or equal to 3 is enough.

As a trigger strategy, the initial state is $x_0 = \delta_3^3$, its reachable set is

$$R(x_0) = \left\{\delta_3^1\delta_3^2, \delta_3^2\delta_3^2, \delta_3^3\delta_3^2, \delta_3^1\delta_3^3, \delta_3^2\delta_3^3, \delta_3^1\delta_3^3\right\}.$$

Using the result above, all the simple cycles contained in $R(x_0)$ are $\delta_3 \times \delta_3\{(1,2)\}$, $\delta_3 \times \delta_3\{(2,2)\}$, $\delta_3 \times \delta_3\{(3,2)\}$ and $\delta_3 \times \delta_3\{(3,3)\}$, and among them $\delta_3 \times \delta_3\{(3,3)\}$ is the optimal cycle. Choosing $u^*(0) = \delta_3^3$, then

$$G^* = \delta_3[*,*,*,*,*,*,*,*,3]$$

where the first eight columns can be arbitrary.

Thus we can choose

$$G^* = \delta_3[2,2,2,2,2,2,2,2,3],$$

which is the trigger strategy. We conclude that the best response for player 1 is to adopt the trigger strategy if player 2 has adopted the trigger strategy. Because the payoffs are symmetrical, so if player 1 has adopted the trigger strategy, the best response for player 2 is also trigger strategy. That means the trigger strategy is a Nash equilibrium of this game.

Remark 4.9: The major obstacle in applying above results to practical networks is the computational complexity. It is easy to see that the computational complexity depends on the computation of L_d, whose complexity is $O(k^{(2d-1)m+3n})$, $d \le k^n$. Thus, an efficient new numerical method has to be developed to deal with large scale networks in further works.

V. Optimal Control of Higher Order Logical Control Networks

To deal with μ-th order logical control networks, we first consider how to convert it to a first order form. We need the following lemma.

Lemma 5.1: [29]:

1) Let $x \in \Delta_k$. Then

$$x^2 = M_{r_k} x \tag{29}$$

where

$$M_{r_k} = \delta_{k^2}\left[1, k+2, 2k+3, \cdots, (k-1)k+k\right].$$

is called the base-k order reducing matrix.

2) Assume $x = \ltimes_{i=1}^{\mu} x_i \in \Delta_{k^\mu}$, then

$$x^2 = \Phi_{\mu_k} x \tag{30}$$

where

$$\Phi_{\mu_k} = \prod_{i=1}^{\mu} I_{k^{i-1}} \otimes \left[\left(I_k \otimes W_{[k,k^{\mu-i}]}\right) M_{r_k}\right] = M_{r_{k^\mu}}.$$

Lemma 5.2: Assume $x = \ltimes_{i=1}^{n} x_i \in \Delta_{k^n}$, set

$$\begin{aligned} F_{[m,n]_k} &= I_{k^m} \otimes \mathbf{1}_{k^{n-m}}^T, \\ E_{[m,n]_k} &= \mathbf{1}_{k^{n-m}}^T \otimes I_{k^m}. \end{aligned}$$

Then

$$F_{[m,n]_k} x = \ltimes_{i=1}^{m} x_i, \quad E_{[m,n]_k} x = \ltimes_{i=n-m+1}^{n} x_i.$$

Proof: If $\ltimes_{i=1}^{m} x_i = \delta_{k^m}^{j}$, then

$$\begin{aligned} F_{[m,n]_k} x &= \left(I_{k^m} \otimes \mathbf{1}_{k^{n-m}}^T\right) \ltimes \delta_{k^m}^{j} \ltimes_{i=m+1}^{n} x_i \\ &= \delta_{k^m}^{j} \ltimes \mathbf{1}_{k^{n-m}}^T \ltimes_{i=m+1}^{n} x_i \\ &= \delta_{k^m}^{j}. \end{aligned}$$

Whatever $\ltimes_{i=1}^{n-m} x_i$ is, we also have

$$\begin{aligned} E_{[m,n]_k} x &= E_{[m,n]_k} \ltimes_{i=1}^{n-m} x_i \ltimes_{i=n-m+1}^{n} x_i \\ &= I_{k^m} \ltimes_{i=n-m+1}^{n} x_i \\ &= \ltimes_{i=n-m+1}^{n} x_i. \end{aligned}$$

□

In the following we simply use M_r, Φ_μ, $F_{[m,n]}$ and $E_{[m,n]}$ for M_{r_k}, Φ_{μ_k}, $F_{[m,n]_k}$ and $E_{[m,n]_k}$ respectively, the default k is assumed to be the type of logic.

Denote $x(t) = \ltimes_{i=1}^{n} x_i(t)$, $u(t) = \ltimes_{i=1}^{m} u_i(t)$. Each equation of the μ-th order logical control network (9) can be written into its algebraic form as

$$\begin{cases} x_1(t+1) = M_1 u(t-\mu+1)\cdots u(t) x(t-\mu+1)\cdots x(t) \\ x_2(t+1) = M_2 u(t-\mu+1)\cdots u(t) x(t-\mu+1)\cdots x(t) \\ \vdots \\ x_n(t+1) = M_n u(t-\mu+1)\cdots u(t) x(t-\mu+1)\cdots x(t). \end{cases} \tag{31}$$

Multiplying the equations in (31) together, we obtain

$$x(t+1) = L \ltimes_{i=1}^{\mu} u(t-\mu+i) \ltimes_{i=1}^{\mu} x(t-\mu+i) \tag{32}$$

where

$$L = M_1 \prod_{j=2}^{n} \left[\left(I_{k^{\mu(m+n)}} \otimes M_j\right) \Phi_{\mu(m+n)}\right].$$

Denote $z(t) = \ltimes_{i=t}^{t+\mu-1} x(i)$, $v(t) = \ltimes_{i=t}^{t+\mu-1} u(i)$, then (32) can be converted to

$$x(t+1) = L v(t-\mu+1) z(t-\mu+1).$$

Then we have

$$\begin{aligned} z(t+1) &= \ltimes_{i=t+1}^{t+\mu} x(i) \\ &= \ltimes_{i=t+1}^{t+\mu-1} x(i) L v(t) z(t) \\ &= \left(I_{k^{(\mu-1)n}} \otimes L\right) \ltimes_{i=t+1}^{t+\mu-1} x(i) v(t) z(t) \\ &= \left(I_{k^{(\mu-1)n}} \otimes L\right) W_{\left[k^{\mu m+n}, k^{(\mu-1)n}\right]} \\ &\quad \ltimes v(t) z(t) \ltimes_{i=t+1}^{t+\mu-1} x(i) \\ &= \left(I_{k^{(\mu-1)n}} \otimes L\right) W_{\left[k^{\mu m+n}, k^{(\mu-1)n}\right]} \\ &\quad \ltimes v(t) x(t) \Phi_{(\mu-1)n} \ltimes_{i=t+1}^{t+\mu-1} x(i) \\ &:= \widetilde{L} v(t) z(t) \end{aligned} \tag{33}$$

where

$$\widetilde{L} = \left(I_{k^{(\mu-1)n}} \otimes L\right) W_{\left[k^{\mu m+n}, k^{(\mu-1)n}\right]} \left(I_{k^{\mu m+n}} \otimes \Phi_{(\mu-1)n}\right).$$

Note that $v(t), t = 0, 1, \cdots$ here is not completely independent, they should satisfy

$$F_{[(\mu-1)m, \mu m]} v(t+1) = E_{[(\mu-1)m, \mu m]} v(t).$$

Thus, (32) can be converted to

$$\begin{cases} z(t+1) = \widetilde{L} v(t) z(t) \\ F_{[(\mu-1)m, \mu m]} v(t+1) = E_{[(\mu-1)m, \mu m]} v(t) \end{cases}. \tag{34}$$

Similar to (16), if $z(t)$ is in a cycle of length d, we have

$$z(t) = z(t+d) = \widetilde{L}_d v(t+d-1) v(t+d-2) \cdots v(t) z(t), \tag{35}$$

where

$$\widetilde{L}_d = \prod_{i=1}^{d} \left(I_{k^{(i-1)\mu m}} \otimes \widetilde{L}\right).$$

$v(t+d-1)v(t+d-2)\cdots v(t)$ can be simplified as

$$\begin{aligned} & v(t+d-1) \cdots v(t) \\ &= \ltimes_{i=t+d-1}^{t+d+\mu-2} u(i) \ltimes_{i=t+d-2}^{t+d+\mu-3} u(i) \cdots \ltimes_{i=t}^{t+\mu-1} u(i) \\ &= W_{[k^{\mu m}]} \ltimes_{i=t+d-2}^{t+d+\mu-3} u(i) \ltimes_{i=t+d-1}^{t+d+\mu-2} u(i) \\ &\quad \ltimes \ltimes_{i=t+d-3}^{t+d+\mu-4} u(i) \cdots \ltimes_{i=t}^{t+\mu-1} u(i) \\ &= W_{[k^{\mu m}]} \left(u(t+d-2) \Phi_{(\mu-1)m} \ltimes_{i=t+d-1}^{t+d+\mu-3} u(i) \right. \\ &\qquad \left. \ltimes\ u(t+d+\mu-2)\right) \\ &\quad \ltimes \ltimes_{i=t+d-3}^{t+d+\mu-4} u(i) \cdots \ltimes_{i=t}^{t+\mu-1} u(i) \\ &= W_{[k^{\mu m}]} \left(I_{k^m} \otimes \Phi_{(\mu-1)m}\right) \ltimes_{i=t+d-2}^{t+d+\mu-2} u(i) \\ &\quad \ltimes \ltimes_{i=t+d-3}^{t+d+\mu-4} u(i) \cdots \ltimes_{i=t}^{t+\mu-1} u(i) \\ &\ \vdots \\ &= \prod_{i=1}^{d-1} \left(W_{\left[k^{\mu m}, k^{(\mu+i-1)m}\right]} \left(I_{k^m} \otimes \Phi_{(\mu-1)m}\right)\right) \\ &\quad \ltimes \ltimes_{i=t}^{t+\mu+d-2} u(i) \\ &:= R \ltimes_{i=t}^{t+\mu+d-2} u(i) \end{aligned}$$

where

$$R = \prod_{i=1}^{d-1} \left(W_{\left[k^{\mu m}, k^{(\mu+i-1)m}\right]} \left(I_{k^m} \otimes \Phi_{(\mu-1)m}\right)\right).$$

Moreover, $v(t+d) = v(t)$ must hold, that is

$$\ltimes_{i=t+d}^{t+d+\mu-1} u(i) = \ltimes_{i=t}^{t+\mu-1} u(i). \tag{36}$$

Assuming $\mu = sd + r$, where $s = [\mu/d]$, $\mu = r (mod\ d)$, the product $v(t+d-1)v(t+d-2)\cdots v(t)$ becomes as shown in the equation at top of the next page.

Then (35) is converted to

$$z(t) = \Psi_d \ltimes_{i=t}^{t+d-1} u(i) z(t) \tag{37}$$

$$
\begin{aligned}
v(t+d-1)\cdots v(t) &= R\ltimes_{i=t}^{t+d-1}u(i)\ltimes_{i=t+d}^{t+d+\mu-2}u(i)\\
&= R\ltimes_{i=t}^{t+d-1}u(i)\left(\ltimes_{i=t}^{t+d-1}u(i)\right)^{s-1}\ltimes_{i=t}^{t+d+r-2}u(i)\\
&= \begin{cases}
R(\Phi_{dm})^{s-1}\left(I_{k^{(d-1)m}}\otimes W_{\left[k^{(d-1)m},k^m\right]}\right)\cdot\Phi_{(d-1)m}\ltimes_{i=t}^{t+d-1}u(i), & r=0\\
R(\Phi_{dm})^{s}\ltimes_{i=t}^{t+d-1}u(i), & r=1\\
R(\Phi_{dm})^{s}\left(I_{k^{(r-1)m}}\otimes W_{\left[k^{(r-1)m},k^{(d-r+1)m}\right]}\right)\cdot\Phi_{(r-1)m}\ltimes_{i=t}^{t+d-1}u(i), & 2\le r\le d-1
\end{cases}
\end{aligned}
$$

where

$$
\Psi_d=\begin{cases}
\widetilde{L}_dR(\Phi_{dm})^{s-1}\\
\quad\ltimes\left(I_{k^{(d-1)m}}\otimes W_{\left[k^{(d-1)m},k^m\right]}\right)\\
\quad\ltimes\Phi_{(d-1)m}, & r=0\\
\widetilde{L}_dR(\Phi_{dm})^{s}, & r=1\\
\widetilde{L}_dR(\Phi_{dm})^{s}\\
\quad\ltimes\left(I_{k^{(r-1)m}}\otimes W_{\left[k^{(r-1)m},k^{(d-r+1)m}\right]}\right)\\
\quad\ltimes\Phi_{(r-1)m}, & 2\le r\le d-1.
\end{cases}
$$

Note that now in (37), $u(i), i=t,t+1,\cdots,t+d-1$ are independent. Refer to the method developed in Section IV, we can search the cycles of length d of (34) by using (37) and checking the trace of $\mathrm{Blk}_i(\Psi_d)$, if its (j,j)-th entry equals 1, $x(t)=\delta_{k^{\mu n}}^j$ is in cycle of length d under $u(t)u(t+1)\cdots u(t+d-1)=\delta_{k^{dm}}^i$. Using (36) we can get $v(t),\cdots,v(t+d-1)$, then we can obtain the cycle. Notice that when ℓ is a proper factor of d, and $z(t)$ is in the cycles of length ℓ and d simultaneously under $\ltimes_{\xi=t}^{t+\ell-1}\widetilde{u}(\xi)=\delta_{k^{\ell m}}^j$ and $\ltimes_{\xi=t}^{t+d-1}u(\xi)=\delta_{k^{dm}}^i$ respectively, then we have the same cycle in control-state space, if and only if $\delta_{k^{dm}}^i=(\delta_{k^{\ell m}}^j)^{d\ell}$. To count the number of cycles, we should take out these repeated cycles. Thus, similar to Theorem 3.4, we have the following theorem.

Theorem 5.3: The number of length d cycles of the logical control network (34) is inductively determined by

$$
N_d=\frac{1}{d}\sum_{i=1}^{k^{dm}}T\left(\mathrm{Blk}_i(\Psi_d)\right) \tag{38}
$$

where

$$
T\left(\mathrm{Blk}_i(\Psi_d)\right)=\mathrm{Tr}\left(\mathrm{Blk}_i(\Psi_d)\right)-\sum_{(j,\ell)\in\theta(i,d)}T\left(\mathrm{Blk}_j(\Psi_\ell)\right).
$$

Proposition 5.4: There is a one-to-one correspondence between the cycles of system (34) and the cycles of higher order logical control network (9).

Proof: Construct a function $\pi:\Delta_{k^{\mu(m+n)}}\to\Delta_{k^{m+n}}$ as follows: Since $\delta_{k^{\mu(m+n)}}^i$ can be decomposed uniquely to $\ltimes_{\ell=1}^{\mu}\delta_{k^m}^{i(\ell)}\ltimes_{\ell=1}^{\mu}\delta_{k^n}^{j(\ell)}$, set

$$
\begin{aligned}
\pi\left(\delta_{k^{\mu(m+n)}}^i\right) &:= F_{[m,\mu m]}\left(I_{k^{\mu m}}\otimes F_{[n,\mu n]}\right)\delta_{k^{\mu(m+n)}}^i\\
&=\delta_{k^m}^{i(1)}\delta_{k^n}^{j(1)}.
\end{aligned} \tag{39}
$$

Denote by Ω_{vz} and Ω_{ux} all the cycles of system (34) and the higher order logical control network (9) respectively. Then we define $\psi:\Omega_{vz}\to\Omega_{ux}$ as follows: For any $C=\{v(t)z(t)\to\cdots\to v(t+d-1)z(t+d-1)\}$,

$$
\psi(C):=\left\{\pi\left(v(t)z(t)\right)\to\cdots\to\pi\left(v(t+d-1)z(t+d-1)\right)\right\}. \tag{40}
$$

Set $u(\xi)=u(\ell)$, $x(\xi)=x(\ell)$, $v(\xi)=v(\ell)$, $z(\xi)=z(\ell)$, whenever $\xi=\ell(mod\ d)$. Because

$$
\begin{aligned}
L\ltimes_{i=t+d-\mu}^{t+d-1}\pi\left(v(i)z(i)\right) &= Lv(t+d-\mu)z(t+d-\mu)\\
&=F_{[n,\mu n]}z(t+d),
\end{aligned}
$$

$\psi(C)$ is a cycle in Ω_{ux}. Thus, ψ is well defined. Then we prove:

1) *ψ is surjective*. For any cycle $C\in\Omega_{ux}$, $C=\{u(t)x(t)\to u(t+1)x(t+1)\to\cdots\to u(t+d-1)x(t+d-1)\}$, let $C_1=\{v(t)z(t)\to\cdots\to v(t+d-1)z(t+d-1)\}$, where $v(i)=\ltimes_{\xi=i}^{i+d-1}u(\xi)$, $z(i)=\ltimes_{\xi=i}^{i+d-1}x(\xi)$. Then we can easily check that $\psi(C_1)=C$.
2) *ψ is injective*. If there is another cycle $C_2=\{\widetilde{v}(t)\widetilde{z}(t)\to\cdots\to\widetilde{v}(t+d-1)\widetilde{z}(t+d-1)\}$ such that $\psi(C_2)=C$, then there exists an $a\le d$, such that $\pi(\widetilde{v}(i)\widetilde{z}(i))=u(a+i-1)x(a+i-1)$, which means that the first m factors of $\widetilde{v}(i)$ form $u(a+i-1)$, and the first n factors of $\widetilde{z}(i)$ form $x(a+i-1)$. By (34) we know that the first $(k-1)m$ factors of $\widetilde{v}(i+1)$ equal to the last $(k-1)m$ factors of $\widetilde{v}(i)$, meanwhile the first $(k-1)n$ factors of $\widetilde{z}(i+1)$ equal to the last $(k-1)n$ factors of $\widetilde{z}(i)$. Thus we obtain

$$
\begin{aligned}
\widetilde{v}(i)&=\ltimes_{\xi=a+i-1}^{a+i+d-2}u(\xi)=v(a+i-1),\\
\widetilde{z}(i)&=\ltimes_{\xi=a+i-1}^{a+i+d-2}x(\xi)=z(a+i-1).
\end{aligned}
$$

Then it is obvious that $C_2=C_1$.

□

Now we consider the optimal control of μ-th order logical network. Set

$$
\widetilde{J}(v)=\overline{\lim}_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\widetilde{P}\left(z(t),v(t)\right) \tag{41}
$$

where

$$
\widetilde{P}\left(z(t),v(t)\right)=P\left(F_{[n,\mu n]}z(t),F_{[m,\mu m]}v(t)\right),
$$

By Lemma 5.2 it is easy to see $F_{[n,\mu n]}z(t)=x(t)$, $F_{[m,\mu m]}v(t)=u(t)$, then maximizing (41) is equivalent to maximizing (10).

Proposition 4.4 is no longer true, but it is easy to see that the optimal cycles (in Ω_{vz}) can be found in the cycles with no repeated element. Thus we can only search from the cycles of length less than or equal to the number of elements of the reachable set $R(z_0)$ of the initial state z_0.

Then, the following theorem can be obtained.

Theorem 5.5: For the μ-th order logical control network (9) with the objective function (10), there exists an optimal logic control matrix G^* such that the objective function is maximized and the trajectory of $s^*(t) = u^*(t)x^*(t)$ will become periodic after a certain finite time.

Proof: We can use (34) and (41) to replace (9) and (10) respectively to find the optimal control. (34) can also be described as a directed graph with finite vertices, so similar to Theorem 4.1, we can find the optimal cycle in Ω_{vz}. Then using (40), the optimal cycle in Ω_{ux} can be obtained. Denote the shortest path from initial state $\delta_{k^n}^{j(0)}, \cdots, \delta_{k^n}^{j(\mu-1)}$ to C^* by

$$\delta_{k^m}^{i(0)}\delta_{k^n}^{j(0)} \to \delta_{k^m}^{i(1)}\delta_{k^n}^{j(1)} \to \cdots \to \delta_{k^m}^{i(T_0-1)}\delta_{k^n}^{j(T_0-1)} \to C^*$$

where

$$C^* = \delta_{k^m} \times \delta_{k^n} \{(i(T_0), j(T_0)) \to (i(T_0+1), j(T_0+1)) \to \cdots \to (i(T_0+d-1), j(T_0+d-1))\}.$$

In the following, if $\ell = \xi (mod\ d)$, when $\ell \geq T_0$ and $T_0 \leq \xi \leq T_0 + d - 1$, then we set $i(\ell) = i(\xi)$. Using this convention, we can find G^*, satisfying

$$\mathrm{Col}_i(G^*) = \delta_{k^m}^{i(\ell+1)}, \quad \mu - 1 \leq \ell \leq T_0 + d + \mu - 2 \qquad (42)$$

where

$$i = \prod_{\xi=1}^{\mu} (i(\ell - \mu + \xi) - 1)\, k^{(\mu-\xi)m+\mu n} + \prod_{\zeta=1}^{\mu-1} (j(\ell - \mu + \zeta) - 1)\, k^{(\mu-\zeta)n} + j(\ell)$$

and the other columns of G^* $(\mathrm{Col}(G^*) \subset \Delta_{k^m})$ can be arbitrary. Then the higher order logical control network (9) is converted to

$$\begin{cases} x^*(t+1) = L \ltimes_{i=1}^{\mu} u^*(t-\mu+i) \ltimes_{j=1}^{\mu} x^*(t-\mu+j) \\ u^*(t+1) = G^* \ltimes_{i=1}^{\mu} u^*(t-\mu+i) \ltimes_{j=1}^{\mu} x^*(t-\mu+j). \end{cases} \qquad (43)$$

□

Example 5.6: Recall Example 1.1, and refer to [30]. Assume the machine uses the strategy "Two Tits For One Tat", it will take the action $m(t+1) = 0$ only when $h(t-1)h(t)m(t-1)m(t) = (0,0,1,1)$. Denote $1 \sim \delta_2^1$, $0 \sim \delta_2^2$ and let $(t, r, p, s) = (5, 3, 1, 0)$, and assume the initial state and control is $m(0) = m(1) = h(0) = h(1) = 0$, then (1) and the human payoff P_h can be rewritten as

$$m(t+1) = Lh(t-1)h(t)m(t-1)m(t), \qquad (44)$$

where

$$L = \delta_2[1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1],$$

and

$$P(m(t), h(t)) := P_h = h'(t) \begin{bmatrix} 1 & 5 \\ 0 & 3 \end{bmatrix} m(t).$$

Set $z(t) = m(t)m(t+1)$, $v(t) = h(t)h(t+1)$, $s(t) = v(t)z(t)$, from (33), (44) can be converted to

$$z(t+1) = \widetilde{L}v(t)z(t) \qquad (45)$$

where

$$\widetilde{L} = (I_2 \otimes L_m)W_{[8,2]}(I_8 \otimes MR) = \delta_4[1,3,1,3,1,3,1,3,1,3,1,3,2,3,1,3].$$

From (28), $\widetilde{P}(\delta_4^1\delta_4^1) = \widetilde{P}(\delta_4^1\delta_4^2) = \widetilde{P}(\delta_4^2\delta_4^1) = \widetilde{P}(\delta_4^2\delta_4^2) = 1$, $\widetilde{P}(\delta_4^1\delta_4^3) = \widetilde{P}(\delta_4^1\delta_4^4) = \widetilde{P}(\delta_4^2\delta_4^3) = \widetilde{P}(\delta_4^2\delta_4^4) = 5$, $\widetilde{P}(\delta_4^3\delta_4^1) = \widetilde{P}(\delta_4^3\delta_4^2) = \widetilde{P}(\delta_4^4\delta_4^1) = \widetilde{P}(\delta_4^4\delta_4^2) = 0$, $\widetilde{P}(\delta_4^3\delta_4^3) = \widetilde{P}(\delta_4^3\delta_4^4) = \widetilde{P}(\delta_4^4\delta_4^3) = \widetilde{P}(\delta_4^4\delta_4^4) = 3$.

It is easy to check the reachable set of the initial state $\delta_4^4\delta_4^4$ is

$$R\left(\delta_4^4\delta_4^4\right) = \{\delta_4^1\delta_4^1, \delta_4^1\delta_4^3, \delta_4^2\delta_4^1, \delta_4^2\delta_4^3, \delta_4^3\delta_4^1, d_4^3\delta_4^2, \delta_4^3\delta_4^3, \delta_4^4\delta_4^1, \delta_4^4\delta_4^2, \delta_4^4\delta_4^3\}.$$

which have ten elements.

By Theorem 5.3, we can obtain the cycles of length less than or equal to 10 with no repeated elements as

$$\begin{cases} C_1 = \delta_4 \times \delta_4 \{(1,1)\} \\ C_2 = \delta_4 \times \delta_4 \{(2,1) \to (3,1)\} \\ C_3^1 = \delta_4 \times \delta_4 \{(1,1) \to (2,1) \to (3,1)\} \\ C_3^2 = \delta_4 \times \delta_4 \{(2,3) \to (4,1) \to (3,2)\} \\ C_3^3 = \delta_4 \times \delta_4 \{(4,1) \to (4,2) \to (4,3)\} \\ C_4^1 = \delta_4 \times \delta_4 \{(1,3) \to (2,1) \to (4,1) \to (3,2)\} \\ C_4^2 = \delta_4 \times \delta_4 \{(2,1) \to (4,1) \to (4,2) \to (3,3)\} \\ C_5^1 = \delta_4 \times \delta_4 \{(1,1) \to (2,1) \to (4,1) \to (3,2) \to (1,3)\} \\ C_5^2 = \delta_4 \times \delta_4 \{(2,1) \to (4,1) \to (3,2) \to (2,3) \to (3,1)\} \\ C_5^3 = \delta_4 \times \delta_4 \{(1,1) \to (2,1) \to (4,1) \to (4,2) \to (3,3)\} \\ C_5^4 = \delta_4 \times \delta_4 \{(2,1) \to (4,1) \to (4,2) \to (4,3) \to (3,1)\} \\ C_6^1 = \delta_4 \times \delta_4 \{(1,1) \to (2,1) \to (4,1) \to (3,2) \to (2,3) \to (3,1)\} \\ C_6^2 = \delta_4 \times \delta_4 \{(1,1) \to (2,1) \to (4,1) \to (4,2) \to (4,3) \to (3,1)\}. \end{cases}$$

It is easy to calculate that the optimal cycle is C_3^2, which has the average human payoff 5/3. This result coincides with the one in [30].

The optimal trajectory for system (45) is

$$\delta_4^4\delta_4^4 \to \delta_4^4\delta_4^3 \to \delta_4 \times \delta_4 \{(4,1) \to (3,2) \to (2,3)\}.$$

Thus we can find the optimal trajectory for system (44) as

$$\delta_2^2\delta_2^2 \to \delta_2^2\delta_2^2 \to \delta_2 \times \delta_2 \{(2,1) \to (2,1) \to (1,2)\}.$$

Then

$$G^* = \delta_2[*,*,*,*,*,*,2,*,*,2,1,*,*,*,*,2,2],$$

where $*$ can be chosen arbitrarily from $\{1, 2\}$.

VI. Conclusion

This paper considered the infinity horizon optimal control of logical control networks, which are either conventional (with order one) or with higher orders. To develop a necessary tool, we first discussed the topological structure of logical control networks, and developed an algorithm with formulas to compute the cycles. For the optimal control under the given objective function, we obtained two main results: (1) the optimal control always exists and the trajectory under the optimal control will be periodic after finite time; (2) there is a logical matrix by which the the optimal control can be expressed. Then the problem was also investigated for higher order logical control networks. After certain algebraic transformation, using the properties of semi-tensor product and techniques developed in the previous work, the systems can be converted into conventional (i.e. first order) logical networks. Then the method developed in the first part remains applicable.

REFERENCES

[1] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theoretical Biol.*, vol. 22, no. 3, p. 437, 1969.

[2] S. Huang and I. Ingber, "Shape-dependent control of cell growth, differentiation, and apotosis: Switching between attractors in cell regulatory networks," *Exp. Cell Res.*, vol. 261, pp. 91–103, 2000.

[3] S. Huang, "Regulation of cellular states in mammalian cells from a genomewide view," *Gene Regulation and Metabolism: Postgenomic Computational Approaches*, 2002.

[4] C. Farrow, J. Heidel, H. Maloney, and J. Rogers, "Scalar equations for synchronous Boolean networks with biological applications," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 348–354, Feb. 2004.

[5] M. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, MA: MIT Press, 1995.

[6] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Random Boolean network models and the yeast transcriptional network," *Proc. Nat. Acad. Sci.*, vol. 100, no. 25, pp. 14 796–14 799, 2003.

[7] R. Albert and A.-L. Barabasi, "Dynamics of complex systems: Scaling laws or the period of Boolean networks," *Phys. Rev. Lett.*, vol. 84, no. 24, pp. 5660–5663, 2000.

[8] M. Aldana, "Boolean dynamics of networks with scale-free topology," *Physica D: Nonlinear Phenomena*, vol. 185, no. 1, pp. 45–66, 2003.

[9] B. Drossel, T. Mihaljev, and F. Greil, "Number and length of attractors in a critical Kauffman model with connectivity one," *Physi. Rev. Lett.*, vol. 94, no. 8, p. 88701, 2005.

[10] B. Samuelsson and C. Troein, "Superpolynomial growth in the number of attractors in Kauffman networks," *Physi. Rev. Lett.*, vol. 90, no. 9, p. 98701, 2003.

[11] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, no. 8, pp. 727–773, 2000.

[12] S. Harris, B. Sawhill, A. Wuensche, and S. Kauffman, "A model of transcriptional rgulatory networks based on biases in the observed regulation rules," *Complexity*, vol. 7, pp. 23–40, 2002.

[13] I. Shmulevich, E. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, p. 261, 2002.

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

[15] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life," *Annu. Rev. Genomics Hum. Genet*, vol. 2, pp. 343–372, 2001.

[16] A. Adamatzky, "On dynamically non-trivial three-valued logics: Oscillatory and bifurcatory species," *Chaos Solitons & Fractals*, vol. 18, no. 5, pp. 917–936, 2003.

[17] L. Volkert and M. Conrad, "The role of weak interactions in biological systems: The dual dynamics model," *J. Theoretical Biol.*, vol. 193, no. 2, pp. 287–306, 1998.

[18] J. Alexander, "Random Boolean networks and evolutionary game theory," *Philos. Sci.*, vol. 70, no. 5, pp. 1289–1304, 2003.

[19] D. Green, T. Leishman, and S. Sadedin, "The emergence of social consensus in Boolean networks," in *Proc. IEEE Symp. ALIFE '07*, 2007, pp. 402–408.

[20] D. Khanafiah and H. Situngkir, "Metabolism of social system: N-person iterated prisoner's dilemma analysis in random boolean network," *J. Soc. Complexity*, vol. 4, no. 1, pp. 20–33, 2004.

[21] J. Miller, "The coevolution of automata in the repeated prisoner's dilemma," *J. Econ. Behav. Organiz.*, vol. 29, no. 1, pp. 87–112, 1996.

[22] M. Piccione and A. Rubinstein, "Finite automata play a repeated extensive game," *J. Economic Theory*, 1993.

[23] D. Cheng, "Sime-tensor product of matrices and its applications—A survey," in *Proc. 4th Int. Congr. Chinese Mathematicians*, Hangzhou, 2007, pp. 641–668.

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

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

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

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

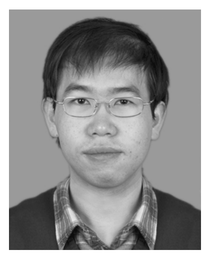
[28] D. Cheng, H. Qi, and Z. Li, "Disturbance decoupling of Boolean control networkss," *IEEE Trans. Aut. Contr.*, vol. 56, no. 1, pp. 2–10, Feb. 2011.

[29] Z. Li and D. Cheng, "Algebraic approach to dynamics of multi-valued networks," *Int. J. Bifurcation and Chaos*, vol. 20, no. 3, pp. 561–582, 2010.

[30] Y. Mu and L. Guo, "Optimization and identification in a non-equilibrium dynamic game," in *Proc. CDC-CCC'09*, 2009, pp. 5750–5755.

[31] Z. Li, Y. Zhao, and D. Cheng, "Structure of Higher Order Boolean Networks," *J. Grad. School, the Chinese Acad. Sci.*, accepted for publication.

[32] R. Gibbons, *A Primer in Game Theory*. Englewood Cliffs, NJ: Prentice Hall, 1992.



Yin Zhao received the B.S. degree in mathematics and applied mathematics from Tsinghua University, Beijing, China, in 2008. He is currently a Ph.D. candidate in Academy of Mathematics and Systems Science, Chinese Academy of Sciences.

His research interests include complex systems, game theory, etc.

Zhiqiang Li (S'09) received the Ph.D. degree in systems theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, in 2010.

He is currently with the Department of Mathematics & Information Science, Henan University of Economics & Law, Zhengzhou. His research interests include nonlinear systems control, complex systems, etc.

Daizhan Cheng (SM'01–F'06) received the B.S. degree from Department of Mechanics, Tsinghua University, in 1970, received the M.S. degree from Graduate School of Chinese Academy of Sciences in 1981, the Ph.D. degree from Washington University, St. Louis, in 1985.

Since 1990, he is a Professor with Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. His research interests include nonlinear control systems, switched systems, Hamiltonian systems-logical dynamic systems, and numerical realization for control design. He is the author/coauthor of over 200 journal papers, 9 books and 100 conference papers.

He was Associate Editor of the *International Journal of Mathematical Systems, Estimation and Control* (90–93); *Automatica* (99–02); the Asian *Journal of Control* (01–04); Subject Editor of the *International Journal of Robust and Nonlinear Control* (2005–2008). He is currently Editor-in-Chief of the *J. Control Theory and Applications* and Deputy Editor-in-Chief of *Control and Decision*. He was the Chairman of IEEE CSS Beijing Chapter (2006–2008), Chairman of Technical Committee on Control Theory, Chinese Association of Automation, Program Committee Chair of annual Chinese Control Conference (2003–2010), IEEE Fellow (2005–) and IFAC Fellow (2008–).