Systems & Control Letters 59 (2010) 767–774

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Input-state incidence matrix of Boolean control networks and its applications ☆

Yin Zhao *, Hongsheng Qi, Daizhan Cheng

The Key Laboratory of Systems & Control, Institute of Systems Science, Academy of Mathematics & Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 6 April 2010
Received in revised form
5 June 2010
Accepted 4 September 2010
Available online 16 October 2010

Keywords:
Boolean control network
Input-state incidence matrix
Reachability
Observability
Mix-valued logic

ABSTRACT

The input-state incidence matrix of a control Boolean network is proposed. It is shown that this matrix contains complete information of the input-state mapping. Using it, an easily verifiable necessary and sufficient condition for the controllability of a Boolean control network is obtained. The corresponding control which drives a point to a given reachable point is designed. Moreover, certain topological properties such as the fixed points and cycles of a Boolean control network are investigated. Then, as another application, a sufficient condition for the observability is presented. Finally, the results are extended to mix-valued logical control systems.

1. Introduction

The Boolean network was firstly proposed by Kauffman for modeling complex and nonlinear biological systems [1–3]. Since then, it has been studied widely and applied to some other systems. The first interesting topic is the topological structure of a Boolean network. There has been a great deal of progress in this subject [4–6]. Another challenging and important topic is the control of Boolean networks. Most effort has been towards the controllability of Boolean networks [7–9].

Recently, a new matrix product, called the semi-tensor product of matrices, has been proposed, and using it a logical mapping can be expressed into a matrix form [10]. Using them, a Boolean (control) network can be converted into a standard discrete-time dynamic (control) system [11,12].

In this paper the same framework is utilized. We, therefore, give a brief review on this new technique. For clarity, we introduce some notations first.

(i) Denote by $\mathrm{Col}(A)$ ($\mathrm{Row}(A)$) the set of columns (rows) of a matrix A, and $\mathrm{Col}_i(A)$ ($\mathrm{Row}_i(A)$) the i-th column (row) of A.

(ii) $\mathcal{D} := \{0, 1\}$.

(iii) Let δ_n^i be the i-th column of the identity matrix I_n, and $\Delta_n := \{\delta_n^1, \delta_n^2, \ldots, \delta_n^n\}$. When $n = 2$ we simply use $\Delta := \Delta_2$.

(iv) $\mathbf{1}_k := [\underbrace{11\ldots 1}_{k}]^T$.

(v) 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, $\mathrm{Col}(M) \subset \Delta_n$. Then M is called a logical matrix, and is simply denoted as

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

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

(vi) A matrix $B \in \mathcal{M}_{n\times s}$ is called a Boolean matrix, if its entries $b_{ij} \in \mathcal{D}$, $\forall\, i, j$. The set of $n \times s$ Boolean matrices is denoted by $\mathcal{B}_{n\times s}$.

(vii) Let $A \in \mathcal{M}_{n\times mn}$, denoted by $\mathrm{Blk}_i(A)$ the i-th $n \times n$ square block of A, $i = 1, 2, \ldots, m$.

A Boolean network with n nodes is described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t)), \quad x_i \in \mathcal{D}, \end{cases} \tag{1}$$

where $f_i : \mathcal{D}^n \to \mathcal{D},\ i = 1, \ldots, n$ are logical functions.

Similarly, a Boolean control network with n network nodes, m input nodes, and p outputs is described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \quad x_i, u_j \in \mathcal{D}, \end{cases} \tag{2}$$
$$y_j(t) = h_j(x_1(t), \ldots, x_n(t)), \quad j = 1, \ldots, p,$$

where $f_i : \mathcal{D}^{n+m} \to \mathcal{D},\ i = 1, \ldots, n$ and $h_j : \mathcal{D}^n \to \mathcal{D},\ j = 1, \ldots, p$ are logical functions.

☆ Supported partly by NNSF 60736022, 60221301 and 61074114 of China.

* Corresponding author.
E-mail addresses: zhaoyin@amss.ac.cn (Y. Zhao), qihongsh@amss.ac.cn (H. Qi), dcheng@iss.ac.cn (D. Cheng).

0167-6911/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sysconle.2010.09.002

Table 1
Truth table.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$p \bar{\vee} q$
1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1
0	1	1	0	1	1	0	1
0	0	1	0	0	1	1	0

Table 2
Matrix expressions of logical functions.

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

Throughout this paper the matrix product is assumed to be a semi-tensor product as $A \ltimes B$, and the symbol "$\ltimes$" is omitted in most places. We refer to [10] for details.

Identifying $1 \sim \delta_2^1,\ 0 \sim \delta_2^2$, and using the semi-tensor product of matrices, logic functions can be expressed by logical matrix, vectors, and their products. For example, the truth table of some common logical functions such as "negation ($\neg$)", "conjunction ($\wedge$)", "disjunction ($\vee$)", "conditional ($\rightarrow$)", "biconditional ($\leftrightarrow$)" and "exclusive or ($\bar{\vee}$)" is given in Table 1, and their structure matrices are given in Table 2.

Then, in vector form we have $\neg p = M_n \ltimes p$ and $p \wedge q = M_c \ltimes p \ltimes q$, etc.

Set $x = \ltimes_{i=1}^n x_i,\ u = \ltimes_{i=1}^m u_i$, and $y = \ltimes_{i=1}^p y_i$. Using vector form, (1) and (2) can be expressed by the following (3) and Eq. (4) respectively, which are called the algebraic forms of (1) and (2) respectively.

$$x(t+1) = Lx(t), \tag{3}$$

where $L \in \mathcal{L}_{2^n \times 2^n}$ is called the structure matrix of the Boolean network.

$$\begin{cases} x(t+1) = Lu(t)x(t) \\ y(t) = Hx(t), \end{cases} \tag{4}$$

where $L \in \mathcal{L}_{2^n \times 2^{n+m}}$, and $H \in \mathcal{L}_{2^p \times 2^n}$.

A matrix, $A = (a_{ij}) \in \mathcal{M}_{n\times n}$ is called the adjacency matrix (or incidence matrix) of the Boolean network (1), if

$$a_{ij} = \begin{cases} 1, & x_j(t+1) \text{ depends on } x_i(t) \\ 0, & \text{otherwise.} \end{cases}$$

If we consider the network graph, $a_{ij} \neq 0$, iff there is an edge from x_i to x_j. The difference between L of (3) and A is that L contains complete information of the network dynamics, precisely, (1) can be recovered from L easily, while A does not.

The controllability and observability of Boolean control networks were investigated under this framework [13]. As a related topic, the controllable normal form and the realization of Boolean networks were discussed in [14]. The topological structure of a Boolean network (without control) such as fixed points and cycles was investigated in [11]. We first give rigorous definitions for the controllability, observability, fixed points and cycles.

Definition 1.1. Consider system (2). Denote its state space as $\mathcal{X} = \mathcal{D}^n$, and let $X_0 \in \mathcal{X}$.

1. $X \in \mathcal{X}$ is said to be reachable from X_0 at time $s > 0$, if we can find a sequence of controls $U(0) = \{u_1(0), \dots, u_m(0)\}$, $U(1) = \{u_1(1), \dots, u_m(1)\}$, …, such that the trajectory of (2) with the initial value X_0 and the controls $\{U(t)\}$, $t = 0, 1, \dots$ will reach X at time $t = s$. The reachable set at time s is denoted by $R_s(X_0)$. The overall reachable set is denoted by
$$R(X_0) = \bigcup_{s=1}^{\infty} R_s(X_0).$$
2. System (2) is said to be controllable at X_0 if $R(X_0) = \mathcal{X}$. The system is said to be controllable if it is controllable at every $X \in \mathcal{X}$.

Definition 1.2. Consider system (2). Denote by $Y(t) = (y_1(t), \dots, y_p(t)) \in \mathcal{D}^p$.

1. X_1^0 and X_2^0 are said to be distinguishable, if there exists a control sequence $\{U(0), U(1), \dots, U(s)\}$, where $s \geq 0$, such that
$$\begin{aligned} Y^1(s+1) &= y^{s+1}(U(s), \dots, U(0), X_1^0) \\ &\neq Y^2(s+1) = y^{s+1}(U(s), \dots, U(0), X_2^0). \end{aligned} \tag{5}$$
2. The system is said to be observable, if any two initial points $X_1^0, X_2^0 \in \mathcal{X}$ are distinguishable.

Definition 1.3. Consider system (2). Denote the input-state (product) space by

$$\mathcal{S} = \left\{(U, X) \mid U = (u_1, \dots, u_m) \in \mathcal{D}^p,\ X = (x_1, \dots, x_n) \in \mathcal{D}^n\right\}.$$

Note that $|\mathcal{S}| = 2^{m+n}$.

1. Let $S_i = (U^i, X^i) \in \mathcal{S}$ and $S_j = (U^j, X^j) \in \mathcal{S}$. Denote by $U^i = (u_1^i, \dots, u_m^i)$, $X^i = (x_1^i, \dots, x_n^i)$, etc. (S_i, S_j), is said to be a directed edge, if X^i, U^i, X^j satisfy (2). Precisely,
$$x_k^j = f_k(x_1^i, \dots, x_n^i, u_1^i, \dots, u_m^i), \quad k = 1, \dots, n.$$
The set of edges is denoted by $\mathcal{E} \subset \mathcal{S} \times \mathcal{S}$.
2. The pair $(\mathcal{S}, \mathcal{E})$ forms a directed graph, which is called the input-state transfer graph.
3. $(S_1, S_2, \dots, S_\ell)$ is called a path, if $(S_i, S_{i+1}) \in \mathcal{E}$, $i = 1, 2, \dots, \ell - 1$.
4. A path $(S_1, S_2, \dots)$ is called a cycle, if $S_{i+\ell} = S_i$ for all i, the smallest ℓ is called the length of the cycle. In particular, the cycle of length 1 is called a fixed point.

Note that when the vector form of logical variables is used, the state space becomes $\mathcal{X} = \Delta_{2^n}$, similarly, the input-state space becomes $\mathcal{S} = \Delta_{2^{m+n}}$. According to the corresponding statement, it is easy to tell which form is used there.

In this paper we propose a matrix, called the input-state incidence matrix. Using it, a neat result about the controllability of Boolean control networks is presented by verifying the controllability matrix. The corresponding controller is also designed. Based on the structure of a reachable set, an easily verifiable sufficient condition for observability is also obtained. Then the input-state incidence matrix is also used to reveal some topological structures of Boolean control networks, including fixed points, cycles etc. Finally, the results are extended to general mix-valued logical dynamic systems. In the conclusion we compare this result with the result obtained in [13] to show that significant progress has been achieved in this paper.

The paper is organized as follows. Section 2 proposes the concept of incidence matrix of Boolean control networks, called the input-state incidence matrix. Its relationship with the algebraic form of the Boolean dynamics is revealed, which provides a convenient way to calculate it. Its basic properties are also investigated. Using the input-state incidence matrix, Section 3 provides a very convenient necessary and sufficient condition for the controllability of the Boolean control networks. Section 4 considers the control design from any point to a point in its reachable set. The corresponding trajectory is also provided. Section 5 gives an easily verifiable sufficient condition for the observability. Formulas for the numbers of fixed points and cycles of a Boolean control network are obtained in Section 6. All the aforementioned results have been extended to mix-valued logical control systems in Section 7. Section 8 is a brief conclusion, which contains a comparison with the existing results.

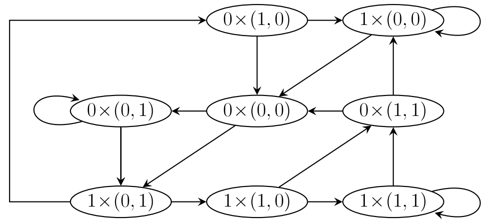

Fig. 1. Input-state dynamic graph.

2. Incidence matrix of Boolean control networks

The incidence matrix of a Boolean control network is used to describe the dynamic process of a Boolean control network. Roughly speaking, it is an algebraic version of the input-state dynamic graph. To begin with, we give a simple example to describe the input-state transfer graph.

Example 2.1. Consider a Boolean control network Σ as

$$\Sigma : \begin{cases} x_1(t+1) = (x_1(t) \vee x_2(t)) \wedge u(t) \\ x_2(t+1) = x_1(t) \leftrightarrow u(t). \end{cases} \quad (6)$$

Setting $x(t) = x_1(t) \ltimes x_2(t)$, it is easy to calculate that the algebraic form of Σ is

$$\Sigma : x(t+1) = Lu(t)x(t), \quad (7)$$

where

$$L = \delta_4 \begin{bmatrix} 1 & 1 & 2 & 4 & 4 & 4 & 3 & 3 \end{bmatrix}. \quad (8)$$

According to the dynamic equation (6) (equivalently, (7)), we can draw the flow of $(u(t), (x_1(t), x_2(t)))$ on the product space $\mathcal{U} \times \mathcal{X}$, called the input-state dynamic graph, as in Fig. 1.

Using vector form, the input-state product space becomes $\Delta \times \Delta_4$. We may define the points in the input-state product space as $P_1 = \delta_2^1 \times \delta_4^1,\ P_2 = \delta_2^1 \times \delta_4^2, \ldots, P_8 = \delta_2^2 \times \delta_4^4$.

Now we construct an 8×8 matrix, $\mathcal{J}(\Sigma)$ in the following way:

$$\mathcal{J}_{ij} = \begin{cases} 1, & \text{there exists an edge from } P_j \text{ to } P_i, \\ 0, & \text{otherwise.} \end{cases}$$

Then $\mathcal{J}(\Sigma)$, called the input-state incidence matrix of the Boolean control network Σ, is

$$\mathcal{J}(\Sigma) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \quad (9)$$

It is easy to see that the incidence matrix of a Boolean control network is indeed the transpose of the adjacency matrix of the input-state transfer graph (notice that it is different from the adjacency matrix of a Boolean network). However, it is very difficult to find this matrix by drawing the graph, since the graph will be very complex when n and m are not very small.

Next, we explore the structure of the input-state incidence matrix. Comparing (9) with (8), one might be surprised to find that

$$\mathcal{J}(\Sigma) = \begin{bmatrix} L \\ L \end{bmatrix}.$$

In fact, this is also true for general case. Consider Eq. (4). Note that, the j-th column of L corresponds to the "output" $x(t+1)$ for P_j-th "input" $(u(t), x(t))$ of the dynamic system. If this column $\mathrm{Col}_j(L) = \delta_{2^n}^i$, then it means that the output $x(t+1)$ is exactly the i-th element of $x(t) \in \Delta_{2^n}$. Now since $u(t+1)$ can be arbitrary, it follows that the input-state incidence matrix of system (4) is

$$\mathcal{J} := \mathcal{J}|_{(4)} = \left.\begin{bmatrix} L \\ L \\ \vdots \\ L \end{bmatrix}\right\} 2^m \in \mathcal{B}_{2^{m+n} \times 2^{m+n}}, \quad (10)$$

where the first block corresponds to $u(t+1) = \delta_{2^m}^1$, the second block corresponds to $u(t+1) = \delta_{2^m}^2$, and so on.

Next, we consider the properties of $\mathcal{J}$.

Definition 2.2. A matrix $A \in \mathcal{M}_{m\times n}$ is called a row-periodic matrix with period τ, if τ is a proper factor of m such that $\mathrm{Row}_{i+\tau}(A) = \mathrm{Row}_i(A),\ 1 \le i \le m - \tau$.

The following property can be verified via a straightforward computation.

Proposition 2.3. *1. $A \in \mathcal{M}_{m\times m}$ is a row-periodic matrix with period τ (where $m = \tau k$), iff*

$$A = \mathbf{1}_\tau A_0,$$

where $A_0 \in \mathcal{M}_{k\times m}$ consists of the first k rows of A, called the basic block of A.

2. $A \in \mathcal{M}_{m\times m}$ is a row-periodic matrix with period τ (where $m = \tau k$), then so is $A^s,\ s \in \mathbb{Z}_+$. ($\mathbb{Z}_+$ is the set of positive integers).

Applying Proposition 2.3 to the incidence matrix, we have

Corollary 2.4. *Consider system (4). Its input-state incidence matrix is*

$$\mathcal{J} = \mathbf{1}_{2^m} \ltimes \mathcal{J}_0, \quad \text{where } \mathcal{J}_0 = L. \quad (11)$$

Moreover, the basic block of $\mathcal{J}^s$ is

$$\mathcal{J}_0^s = L \ltimes (\mathbf{1}_{2^m} \ltimes L)^{s-1}. \quad (12)$$

Note that since $\mathcal{J}^s$ is a row-periodic matrix, it is easy to see that

$$\mathcal{J}_0^{s+1} = \mathcal{J}_0 \mathcal{J}^s = L\mathcal{J}^s = L\mathbf{1}_{2^m} \ltimes \mathcal{J}_0^s. \quad (13)$$

This equation shows that in calculating $\mathcal{J}_0^s$ we do not need to take the whole $\mathcal{J}$ into the calculation. We summarize it as follows.

Proposition 2.5.

$$\mathcal{J}_0^{s+1} = M^s L, \quad (14)$$

where

$$M = \sum_{i=1}^{2^m} \mathrm{Blk}_i(L).$$

Proof. . From (13) it is easy to see $\mathcal{J}_0^{s+1} = M\mathcal{J}_0^s$. Since $\mathcal{J}_0 = L$, (14) is obtained. □

3. Controllability

First, we consider the physical meaning of $\mathcal{J}^s$. When $s = 1$ we know that $\mathcal{J}_{ij}$ means whether there exists a set of controls such that P_i is reachable from P_j in one step by judging if $\mathcal{J}_{ij} = 1$ or not. Is there a similar meaning for $\mathcal{J}^s$? The following result answers this.

Theorem 3.1. *Consider system (4). Assume that the (i, j)-th element of the s-th power of its input-state incidence matrix, $\mathcal{J}_{ij}^s = c$. Then there are c paths from point P_j reach P_i at s-th step with proper controls.*

Proof. We prove it by mathematical induction. When $s = 1$ the conclusion follows from the definition of input-state incidence matrix.

Now assume $\mathcal{J}_{ij}^s$ is the number of the paths from P_j to P_i at the s-th step. Since a path from P_j to P_i at the $(s+1)$-th step can always be considered as a path from P_j to P_k at the s-th step and then from P_k to P_i at one step. It can be calculated as

$$c = \sum_{k=1}^{2^{m+n}} \mathcal{J}_{ik}\mathcal{J}_{kj}^s,$$

which is exactly $\mathcal{J}_{ij}^{s+1}$. □

From the above theorem the following result is obvious.

Corollary 3.2. *Consider system (4) with its input-state incidence matrix $\mathcal{J}$. P_i is reachable from P_j at the s-th step, iff $\mathcal{J}_{ij}^s > 0$.*

The above arguments showed that the whole controllability information is contained in $\{\mathcal{J}^s | s = 1, 2, \ldots\}$. By the Cayley–Hamilton Theorem in linear algebra, it is easy to see that if $\mathcal{J}_{ij}^s = 0$, $s \le 2^{m+n}$, then so are $\mathcal{J}^s$, $\forall s$. Next, we consider only $\{\mathcal{J}^s | s \le 2^{m+n}\}$. Since they are row-periodic matrices, we need only to consider their basic blocks $\mathcal{J}_0^s$. From Proposition 2.5, $\mathrm{Blk}_i(\mathcal{J}_0^s) = M^{s-1}\,\mathrm{Blk}_i(L)$. By the construction, it is clear that $\mathrm{Blk}_i(\mathcal{J}_0^s)$ corresponds to the i-th input $u = \delta_{2^m}^i$. Moreover, the j-th column of $\mathrm{Blk}_\mu(\mathcal{J}_0^s)$ corresponds to the initial value $x_0 = \delta_{2^n}^j$. Then the following conclusion is clear.

Theorem 3.3. *Consider system (4) with its input-state incidence matrix $\mathcal{J}$.*

1. *$x(s) = \delta_{2^n}^\alpha$ is reachable from $x(0) = \delta_{2^n}^j$ at s-th step, iff*
$$\sum_{i=1}^{2^m} (\mathrm{Blk}_i(\mathcal{J}_0^s))_{\alpha j} = (M^s)_{\alpha j} > 0. \tag{15}$$
2. *$x = \delta_{2^n}^\alpha$ is reachable from $x(0) = \delta_{2^n}^j$, iff*
$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} (\mathrm{Blk}_i(\mathcal{J}_0^s))_{\alpha j} = \sum_{s=1}^{2^{m+n}} (M^s)_{\alpha j} > 0. \tag{16}$$
3. *The system is controllable at $x(0) = \delta_{2^n}^j$, iff*
$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} \mathrm{Col}_j[\mathrm{Blk}_i(\mathcal{J}_0^s)] = \sum_{s=1}^{2^{m+n}} \mathrm{Col}_j(M^s) > 0. \tag{17}$$
4. *The system is controllable, iff*
$$\sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} \mathrm{Blk}_i(\mathcal{J}_0^s) = \sum_{s=1}^{2^{m+n}} M^s > 0. \tag{18}$$

Note that let $A \in \mathcal{M}_{m\times n}$. The inequality $A > 0$ means all the entries of A are positive, i.e., $a_{i,j} > 0$, $\forall i, j$.

When the controllability is considered, we do not need to consider how many paths from one state to the other. Hence the true number of $\mathcal{J}^s$ is less interesting. What we really need is whether it is positive or not. Hence, we can simply use the Boolean algebra in the above calculation.

We give a brief review on Boolean algebra in the following remark.

Remark 3.4. 1. If $a, b \in \mathcal{D}$, we can define the Boolean addition and the Boolean product respectively as

$$a +_{\mathcal{B}} b = a \vee b; \qquad a \times_{\mathcal{B}} b = a \wedge b.$$

$\{\mathcal{D}, +_{\mathcal{B}}, \times_{\mathcal{B}}\}$ forms an algebra, called the Boolean algebra.

2. Let $A = (a_{ij})$, $B = (b_{ij}) \in \mathcal{B}_{m\times n}$. Then we define.
$$A +_{\mathcal{B}} B = (a_{ij} +_{\mathcal{B}} b_{ij}).$$
3. Let $A \in \mathcal{B}_{m\times n}$ and $B \in \mathcal{B}_{n\times p}$. Then $A \times_{\mathcal{B}} B := C \in \mathcal{B}_{m\times p}$ as
$$c_{ij} = \sum_{k=1\,\mathcal{B}}^{n} a_{ik} \times_{\mathcal{B}} b_{kj}.$$
Particularly, Let $A \in \mathcal{B}_{n\times n}$. Then
$$A^{(2)} := A \times_{\mathcal{B}} A.$$

We use a simple example to illustrate the Boolean operations.

Example 3.5. Assume

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}; \qquad B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Then

$$A +_{\mathcal{B}} B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}; \qquad A \ltimes_{\mathcal{B}} B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$A^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}; \qquad A^{(s)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad s \ge 3.$$

Using Boolean algebra, we have the following corollary.

Corollary 3.6. *The results 1, 2, 3, and 4 of Theorem 3.3 remain true if in the corresponding conditions (15)–(18) $\mathcal{J}_0^s$ is replaced by $\mathcal{J}_0^{(s)}$.*

In particular, we call

$$\mathcal{C} := \sum_{s=1}^{2^{m+n}} \sum_{i=1}^{2^m} \mathrm{Blk}_i(\mathcal{J}_0^{(s)}) = \sum_{s=1}^{2^{m+n}} M^{(s)} \in \mathcal{B}_{2^n\times 2^n} \tag{19}$$

the controllability matrix, and denote $\mathcal{C} = (c_{i,j})$. Then

(i) *$\delta_{2^n}^i$ is reachable from $\delta_{2^n}^j$, iff $c_{i,j} > 0$;*
(ii) *The system is controllable at $\delta_{2^n}^j$, iff $\mathrm{Col}_j(\mathcal{C}) > 0$;*
(iii) *The system is controllable, iff $\mathcal{C} > 0$.*

The following example shows how to use Theorem 3.3 or Corollary 3.6.

Example 3.7. Consider the following Boolean control network

$$\begin{cases} x_1(t+1) = (x_1(t) \leftrightarrow x_2(t)) \vee u_1(t) \\ x_2(t+1) = \neg x_1(t) \wedge u_2(t), \end{cases} \tag{20}$$

$y(t) = x_1(t) \vee x_2(t)$.

Setting $x(t) = \ltimes_{i=1}^2 x_i(t)$, $u = \ltimes_{i=1}^2 u_i(t)$, we have

$$\begin{cases} x(t+1) = Lu(t)x(t) \\ y(t) = Hx(t), \end{cases} \tag{21}$$

where

$L = \delta_4[\,2 \ 2 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 4 \ 3 \ 1 \ 2 \ 4 \ 4 \ 2\,]$,
$H = \delta_2[\,1 \ 1 \ 1 \ 2\,]$.

For system (20), the basic block of its input-state incidence matrix $\mathcal{J}_0 = L$.

1. Is δ_4^1 reachable from $x(0) = \delta_4^2$?
After a straightforward computation, we have
$$(M^{(1)})_{12} = 0, \qquad (M^{(2)})_{12} > 0.$$
That means that $x(2) = \delta_4^1$ is reachable from $x(0) = \delta_4^2$ at 2nd step.

2. Is the system controllable or controllable at any point?
We check the controllability matrix:

$$\mathcal{C} = \sum_{s=1}^{2^4} M^{(s)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

According to Corollary 3.6, we conclude that
(i) The system is not controllable. It is controllable at $x_0 = \delta_4^3 \sim (0, 1)$.
(ii) $x_d = \delta_4^3 \sim (0, 1)$ is not reachable from $x_0 = \delta_4^1 \sim (1, 1)$, or $x_0 = \delta_4^2 \sim (1, 0)$, or $x_0 = \delta_4^4 \sim (0, 0)$.

4. Trajectory tracking and control design

Assume $x_d \in R(x_0)$. The purpose of this section is to find a control, which drives x_0 to x_d. Since the trajectory from x_0 to x_d (driven by a proper sequence of controls) is in general not unique, we only try to find the shortest one. A similar way can produce all the required trajectories.

Assume $x_0 = \delta_{2^n}^j$ and $x_d = \delta_{2^n}^i$. We give the following algorithm.

Algorithm 4.1. Assume the (i, j)-th element of the controllability matrix, $c_{i,j} > 0$.

- Step 1: Find the smallest s, such that in the block decomposed form
$$\mathcal{J}_0^s = \left[\text{Blk}_1(\mathcal{J}_0^s) \quad \text{Blk}_2(\mathcal{J}_0^s) \quad \cdots \quad \text{Blk}_{2^m}(\mathcal{J}_0^s)\right], \tag{22}$$
(where $\text{Blk}_i(\mathcal{J}_0^s) \in \mathcal{M}_{2^n \times 2^n}$) there exists a block, say, $\text{Blk}_\alpha\left(\mathcal{J}_0^{(s)}\right)$, which has its (i, j)-element
$$[\text{Blk}_\alpha(\mathcal{J}_0^{(s)})]_{ij} > 0. \tag{23}$$
Set $u(0) = \delta_{2^m}^\alpha$ and $x(s) = \delta_{2^n}^i$. If $s = 1$, stop. Else, go to the next step.
- Step 2: Find k, β, such that
$$[\text{Blk}_\beta(\mathcal{J}_0)]_{ik} > 0; \qquad [\text{Blk}_\alpha(\mathcal{J}_0^{s-1})]_{kj} > 0.$$
Set $u(s-1) = \delta_{2^m}^\beta$ and $x(s-1) = \delta_{2^n}^k$.
- Step 3: If $s - 1 = 1$, stop. Else, set $s = s - 1$, and $i = k$ (that is, replace s by $s - 1$ and replace i by k), and go back to Step 2.

Proposition 4.2. *As long as $x_d \in R(x_0)$ the control sequence $\{u(0), u(1), \ldots, u(s-1)\}$ generated by Algorithm 4.1 can drive the trajectory from x_0 to x_d. Moreover, the corresponding trajectory is $\{x(0) = x_0, x(1), \ldots, x(s) = x_d\}$, which is also produced from the algorithm.*

Proof. Since $x_d \in R(x_0)$, by the construction of controllability matrix $\mathcal{C}$, there exists the smallest s such that $\left[\text{Blk}_\alpha\left(\mathcal{J}_0^{(s)}\right)\right]_{i,j} > 0$. That means if $u(0) = \delta_{2^m}^\alpha, x(0) = \delta_{2^n}^j$, there exists at least one path from $x(0)$ to $x(s) = \delta_{2^n}^i$. Then we know x_0 can reach x_d at the s-th step if $u(0) = \delta_{2^m}^\alpha$. Hence, it is obvious that there must exist k such that x_0 can reach $\delta_{2^n}^k$ at $(s-1)$-th step with $u(0) = \delta_{2^m}^\alpha$, and β such that $u(s-1) = \delta_{2^m}^\beta$ which make $L\delta_{2^m}^\beta \delta_{2^n}^k = \delta_{2^n}^i$. Equivalently, we can find k, β, such that

$$[\text{Blk}_\beta(\mathcal{J}_0)]_{ik} > 0, \qquad [\text{Blk}_\alpha(\mathcal{J}_0^{s-1})]_{kj} > 0.$$

In the same way, we can found β' and k' such that $\delta_{2^n}^{k'}$ can be reached at $(s-2)$-th step and $L\delta_{2^m}^{\beta'}\delta_{2^n}^{k'} = \delta_{2^n}^k$. Continue this process, the sequence of controls and states from x_0 to x_d can be obtained. □

Example 4.3. Recall Example 3.7. For $x_0 = \delta_4^2$ and $x_d = \delta_4^1$, we want to find a trajectory from x_0 to x_d. We follow Algorithm 4.1 step by step as follows:

- Step 1: The smallest s is 2. We can calculate that
$[\text{Blk}_3(\mathcal{J}_0^2)]_{12} > 0.$
So $u(0) = \delta_4^3, \ x(2) = \delta_4^1$.
- Step 2: From a straightforward computation, we have
$[\text{Blk}_1(\mathcal{J}_0)]_{14} > 0; \quad [\text{Blk}_3(\mathcal{J}_0^{2-1})]_{42} > 0.$
So $u(1) = \delta_4^1, \ x(1) = \delta_4^4$.
- Step 3: Now $s - 1 = 1$, we stop the process.

Hence the control sequence for $x_0 = \delta_4^2 \sim (1, 0)$ and $x_d = \delta_4^1 \sim (1, 1)$ is $\{u(0) = \delta_4^3 \sim (0, 1), u(1) = \delta_4^1 \sim (1, 1)\}$, and the trajectory is $\{x(0) = \delta_4^2 \sim (1, 0), x(1) = \delta_4^4 \sim (0, 0), x(2) = \delta_4^1 \sim (1, 1)\}$. In general, the smallest step trajectory is not unique. For this example there are 4 ways that drive x_0 to x_d in 2 steps. By the same way, we can find the other 3 paths, which are

$$\{u(0) = \delta_4^3, u(1) = \delta_4^3\}, \qquad \{x(0) = \delta_4^2, x(1) = \delta_4^4, x(2) = \delta_4^1\};$$
$$\{u(0) = \delta_4^4, u(1) = \delta_4^1\}, \qquad \{x(0) = \delta_4^2, x(1) = \delta_4^4, x(2) = \delta_4^1\};$$
$$\{u(0) = \delta_4^4, u(1) = \delta_4^3\}, \qquad \{x(0) = \delta_4^2, x(1) = \delta_4^4, x(2) = \delta_4^1\}.$$

5. Observability

This section considers the observability of system (2). We need some new notations.

Definition 5.1. 1. Let $A = (a_{ij}) \in \mathcal{B}_{n\times s}$, σ is a unary operator. Then $\sigma : \mathcal{B}_{n\times s} \to \mathcal{B}_{n\times s}$ is defined as
$$\sigma A := (\sigma a_{ij}). \tag{24}$$
2. Let $A = (a_{i,j}), B = (b_{i,j}) \in \mathcal{B}_{n\times s}$, σ is a binary operator. Then $\sigma : \mathcal{B}_{n\times s} \times \mathcal{B}_{n\times s} \to \mathcal{B}_{n\times s}$ is defined as
$$A\sigma B := (a_{i,j}\sigma b_{i,j}). \tag{25}$$

We give a simple example for this.

Example 5.2. Assume.

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}; \qquad B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Then

$$\neg A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}; \qquad A\bar{\vee}B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

From the construction of $\mathcal{J}$ and the property of semi-tensor product, It is easy to see that $\text{Blk}_i(\mathcal{J}_0^{(s)})$ corresponds to the input $u(0) = \delta_{2^m}^i$. Moreover, each block $\text{Col}_j[\text{Blk}_i(\mathcal{J}_0^{(s)})]$ corresponds to $x_0 = \delta_{2^n}^j$. To exchange the running order of the indexes i and j, we use the swap matrix to define

$$\tilde{\mathcal{J}}_0^{(s)} := \mathcal{J}_0^{(s)} W_{[2^n, 2^m]}, \tag{26}$$

and then split it into 2^n blocks as

$$\tilde{\mathcal{J}}_0^{(s)} = [\text{Blk}_1(\tilde{\mathcal{J}}_0^{(s)})\ \text{Blk}_2(\tilde{\mathcal{J}}_0^{(s)}) \ldots \text{Blk}_{2^n}(\tilde{\mathcal{J}}_0^{(s)})], \tag{27}$$

where $\text{Blk}_i(\tilde{\mathcal{J}}_0^{(s)}) \in \mathcal{B}_{2^n\times 2^m}$, $i = 1, \ldots, 2^n$.

Now each block $\text{Blk}_i(\tilde{\mathcal{J}}_0^{(s)})$ corresponds to $x_0 = \delta_{2^n}^i$, and in each block $\text{Col}_j(\text{Blk}_i(\tilde{\mathcal{J}}_0^{(s)}))$ corresponds to $u(0) = \delta_{2^m}^j$.

Using the Boolean algebraic expression, we have the following sufficient condition for the observability.

Theorem 5.3. *Consider system (2) with its algebraic form (4). If*

$$\bigvee_{s=1}^{2^{m+n}} [(H \ltimes \mathrm{Blk}_i(\tilde{\mathcal{J}}_0^{(s)}))\bar{\vee}(H \ltimes \mathrm{Blk}_j(\tilde{\mathcal{J}}_0^{(s)}))] \neq 0,$$

$$1 \le i < j \le 2^n, \tag{28}$$

then the system is observable.

Proof. According to the construction and the above argument, it is easy to see that (28) implies that at least at one step the outputs corresponding to $x^0 = \delta_{2^m}^i$ and $x^0 = \delta_{2^m}^j$ are distinct. □

Remark 5.4. 1. Comparing with the result in [13], one of the advantages of this result is when the step s increasing, the corresponding matrices, concerned in the condition, do not increase their dimensions. So, it is easily computable. The major disadvantage is that this result is not necessary.
2. This observability result has been generated to mix-valued logical control systems. An numerical example for the observability of the generalized case can be found in Example 7.3.

6. Fixed points and cycles

The fixed points and cycles of an input-state dynamic graph are very important topological properties of a Boolean control network. For instance, the optimal control can always be realized over a fixed point or a cycle [15]. The input-state incidence matrix can also provide the information about this.

Taking the properties of $\mathcal{J}^s$ into consideration and recalling the argument for the fixed points and cycles for a free Boolean network (without control) [11], the following result is obvious.

Theorem 6.1. *Consider the state equation of system (2) with its input-state incidence matrix $\mathcal{J}$.*

1. *The number of the fixed points in the input-state dynamic graph is*

$$N_1 = \sum_{i=1}^{2^m} \mathrm{tr}(\mathrm{Blk}_i(\mathcal{J}_0)) = \mathrm{tr}(M). \tag{29}$$

2. *The number of length s cycles can be calculated inductively as*

$$N_s = \frac{\mathrm{tr}(M^s) - \sum_{k\in\mathcal{P}(s)} kN_k}{s}, \quad 2 \le s \le 2^{m+n}. \tag{30}$$

We use an example to depict it.

Example 6.2. Recall Example 2.1. We can calculate that

$\mathrm{tr}\, M = 3$, $\mathrm{tr}\, M^3 = 6$,
$\mathrm{tr}\, M^4 = 15$, $\mathrm{tr}\, M^5 = 33$,
$\mathrm{tr}\, M^6 = 66$, $\mathrm{tr}\, M^7 = 129$,
$\mathrm{tr}\, M^8 = 255$.

Using Theorem 6.1, we conclude that $N_1 = 3$, $N_3 = 1$, $N_4 = 3$, $N_5 = 6$, $N_6 = 10$, $N_7 = 18$, $N_8 = 30$. It is not an easy job to count them from the graph directly.

7. Mix-valued logical system

In multi-valued logic case, say, consider the k-valued logical network, we have $x_i, u_i \in \mathcal{D}_k$ [16]. When the infinitely repeated game is considered, the dynamics of the strategies, depending on one history, may be expressed as in (2) but $x_i \in \mathcal{D}_{k_i}$, and $u_i \in \mathcal{D}_{s_i}$. Such a dynamic system is called a mix-valued logical dynamic system.

Set

$$x = \ltimes_{i=1}^{n} x_i \in \prod_{i=1}^{n} \mathcal{D}_{k_i}; \qquad u = \ltimes_{\alpha=1}^{m} u_\alpha \in \prod_{\alpha=1}^{m} \mathcal{D}_{j_\alpha}.$$

Then in vector forms, we have $x_i \in \Delta_{k_i}$, and $u_\alpha \in \Delta_{j_\alpha}$. Set $k = \prod_{i=1}^{n} k_i$ and $j = \prod_{\alpha=1}^{m} j_\alpha$, then we have

$$x \in \Delta_k; \qquad u \in \Delta_j.$$

In this section, we claim that all the major results obtained in previous sections remain true for mix-valued logical dynamic systems (including multi-valued logical control networks as a particular case). We state it as a theorem and omit the proofs, because they are exactly the same.

Theorem 7.1. *Consider system (2), and assume it is a mix-valued logical dynamic system. Precisely, $x_i \in \mathcal{D}_{k_i}$, $i = 1, \dots, n$, $u_\alpha \in \mathcal{D}_{j_\alpha}$, $\alpha = 1, \dots, m$, $y_\beta \in \mathcal{D}_{\ell_\beta}$, $\beta = 1, \dots, p$. That is, each state x_i, each control u_α, and each output y_β can have different dimensions. Then we have the following generalizations.*

1. *Consider the controllability of this mix-valued logical dynamic system, Theorem 3.3 and Corollary 3.6 remain true.*
2. *Consider the observability of this mix-valued logical dynamic system, Theorem 5.3 remains true.*
3. *Consider the number of fixed points and the number of cycles of this mix-valued logical dynamic system, Theorem 6.1 remains true.*
4. *Consider the trajectories and the corresponding controls, Algorithm 4.1 remains available.*

To apply the extended results technically we need to solve the following problem: How to calculate $\{x_i\}$ from x and vise versa. Similarly, we have also to calculate $\{u_i\}$ from u and vise versa. We give the following formula.

Proposition 7.2. *Let $x_i = \delta_{k_i}^{\alpha_i}$, $i = 1, \dots, n$, and $x = \delta_k^{\alpha}$. Then*

1.

$$\alpha = (\alpha_1 - 1) \times \frac{k}{k_1} + (\alpha_2 - 1) \times \frac{k}{k_1 k_2} + \dots + (\alpha_{n-1} - 1) \times k_n + \alpha_n. \tag{31}$$

2.

$$\begin{cases} x_1 = \mathrm{diag}\left(\underbrace{\mathbf{1}_{k_1}^T, \dots, \mathbf{1}_{k_1}^T}_{k/k_1}\right) x \\ x_j = \mathrm{diag}\left(\underbrace{\mathbf{1}_{k_j}^T, \dots, \mathbf{1}_{k_j}^T}_{k/k_j}\right) W_{\left[\prod_{i=1}^{j-1} k_i, k_j\right]} x. \end{cases} \tag{32}$$

Proof. Eq. (31) can be proved via a straightforward computation. The first equality in (32) comes from the definition of the semi-tensor product. To prove the second one we have.

$$W_{\left[\prod_{i=1}^{j-1} k_i, k_j\right]} x = x_j x_1 \dots x_{j-1} x_{j+1} \dots x_n.$$

Using the first equality yields the second equality. □

We use the following example to demonstrate all the extended results in Theorem 7.1.

Example 7.3. Consider the following mix-valued dynamic system

$$\begin{cases} x_1(t+1) = f_1(u(t), x_1(t), x_2(t)) \\ x_2(t+1) = f_2(u(t), x_1(t), x_2(t)), \end{cases} \quad (33)$$
$$y(t) = h(x_1(t), x_2(t)),$$

where $x_1(t) \in \mathcal{D}_2,\ x_2(t) \in \mathcal{D}_3,\ u(t) \in \mathcal{D}_2,\ f_1 : \mathcal{D}_2^2 \times \mathcal{D}_3 \to \mathcal{D}_2,\ f_2 : \mathcal{D}_2^2 \times \mathcal{D}_3 \to \mathcal{D}_3$, and $h : \mathcal{D}_2 \times \mathcal{D}_3 \to \mathcal{D}_3$ are mix-valued logical functions.

Using vector form, system (33) can be expressed as

$$\begin{cases} x_1(t+1) = M_1 u(t) x_1(t) x_2(t), \\ x_2(t+1) = M_2 u(t) x_1(t) x_2(t), \quad x_1, u \in \Delta_2,\ x_2 \in \Delta_3, \end{cases} \quad (34)$$
$$y(t) = H x_1(t) x_2(t), \quad y \in \Delta_3.$$

In fact, in mix-valued cases, to describing a logical function is not an easy job. In general it should be described by a Truth Table. We skip this, and give their structure matrices directly. We assume the structure matrices of $f_1,\ f_2$, and h are $M_1,\ M_2$, and H respectively, where

$$M_1 = \delta_2 \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix};$$
$$M_2 = \delta_3 \begin{bmatrix} 3 & 1 & 3 & 2 & 2 & 1 & 3 & 2 & 1 & 3 & 3 & 3 \end{bmatrix};$$
$$H = \delta_3 \begin{bmatrix} 1 & 3 & 3 & 2 & 2 & 2 \end{bmatrix}.$$

Setting $x(t) = x_1(t)x_2(t)$, the algebraic form of (33) can be calculated as

$$\begin{cases} x(t+1) = L u(t) x(t) \\ y(t) = H x(t), \end{cases} \quad (35)$$

where

$$L = \delta_6[3\,1\,3\,5\,2\,4\,6\,5\,4\,6\,6\,6].$$

1. Consider the controllability of the system. The basic block of the input-state incidence matrix $\mathcal{J}_0 = L$. From a straightforward computation, we have the controllability matrix as
$$\mathcal{C} = \sum_{s=1}^{12} M^{(s)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} > 0.$$
We conclude that system (33) is controllable.
2. Given any two points, say, $x_0 = \delta_6^1 \sim (\delta_2^1, d_3^1)$ and $x_d = \delta_6^5 \sim (\delta_2^2, \delta_3^2)$. we want to find a trajectory from x_0 to x_d with proper controls.
   - Step 1: the smallest s is 3 for
     $[\mathrm{Blk}_1\, \mathcal{J}_0^{(3)}]_{51} > 0,$
     so $u(0) = \delta_2^1, x(3) = \delta_6^5$.
   - Step 2: we have
     $[\mathrm{Blk}_1\, \mathcal{J}_0]_{54} > 0, \qquad [\mathrm{Blk}_1\, \mathcal{J}_0^{3-1}]_{41} > 0,$
     so $u(2) = \delta_2^1, x(2) = \delta_6^4$. Then
     $[\mathrm{Blk}_2\, \mathcal{J}_0]_{43} > 0, \qquad [\mathrm{Blk}_1\, \mathcal{J}_0^{2-1}]_{31} > 0,$
     so $u(1) = \delta_2^2, x(1) = \delta_6^3$.
   - Step 3: $s - 1 = 1$, stop the process.

   Hence the control sequence leads $x_0 = \delta_6^1$ to $x_d = \delta_6^5$ is $\{u(0) = \delta_2^1, u(1) = \delta_2^2, u(2) = \delta_2^1\}$, and the trajectory is $\{x(0) = \delta_6^1, x(1) = \delta_6^3, x(2) = \delta_6^4, x(3) = \delta_6^5\}$.
3. Next, we calculate the number of fixed points and the numbers of cycles of different lengths.

   It is easy to calculate that
   $$\begin{aligned} &\mathrm{tr}(M) = 2, \quad \mathrm{tr}(M^2) = 6, \\ &\mathrm{tr}(M^3) = 8, \quad \mathrm{tr}(M^4) = 14, \\ &\mathrm{tr}(M^5) = 37, \quad \mathrm{tr}(M^6) = 60, \\ &\mathrm{tr}(M^7) = 135, \quad \mathrm{tr}(M^8) = 254, \\ &\mathrm{tr}(M^9) = 512, \quad \mathrm{tr}(M^{10}) = 1031, \\ &\mathrm{tr}(M^{11}) = 2037, \quad \mathrm{tr}(M^{12}) = 4112. \end{aligned}$$
   We conclude that there are $N_1 = 2$ fixed points; and N_i cycles of length $i,\ i = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$, where $N_2 = 2,\ N_3 = 2,\ N_4 = 2,\ N_5 = 7,\ N_6 = 8,\ N_7 = 19,\ N_8 = 30,\ N_9 = 56,\ N_{10} = 99,\ N_{11} = 185,\ N_{12} = 337$.
4. Finally, we consider the observability of the system.

   Denote
   $$O_{ij} = \bigvee_{s=1}^{2\times 6} [(H \ltimes \mathrm{Blk}_i(\tilde{\mathcal{J}}_0^{(s)})) \bar{\vee} (H \ltimes \mathrm{Blk}_j(\tilde{\mathcal{J}}_0^{(s)}))].$$
   A straightforward computation yields
   $$O_{12} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{13} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad O_{14} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix},$$
   $$O_{15} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{16} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{23} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix},$$
   $$O_{24} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad O_{25} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{26} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$
   $$O_{34} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad O_{35} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad O_{36} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix},$$
   $$O_{45} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{46} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad O_{56} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$
   Since $O_{ij} \neq 0,\ 1 \le i < j \le 6$, according to Theorems 5.3 and 7.1, the system is observable.

8. Conclusion

The controllability of a Boolean control network is investigated. The set of admissible controls are a sequence of m-dimensional Boolean numbers,

$$\{U(t) = (u_1(t), \ldots, u_m(t)) | t = 0, 1, 2, \ldots\}.$$

Formulas are obtained for reachable set of each point, for the controllability of each point and overall controllability. In particular the controllability matrix is constructed to verify the reachability from any initial x_0 to any destination x_d. Control design and the corresponding trajectory are presented. Formulas for calculating the numbers of fixed points and cycles are also obtained.

All the previous results have been extended to mix-valued logical control systems.

Comparing the new controllability result with the corresponding result in [13]. The main result for free sequences of controls in [13] is as follows:

Theorem 8.1. *Consider system (2). Assume its algebraic form is expressed as*

$$x(t+1) = L x(t) u(t), \quad (36)$$

where $L \in \mathcal{L}_{2^n \times 2^{m+n}}$. *Then the reachable set from* x_0 *is*

$$R(x_0) = \mathrm{Col}\left\{\bigcup_{i=1}^{2^n} L^i x_0\right\}. \tag{37}$$

Note that according to the semi-tensor product, $L^s \in \mathcal{L}_{2^n \times 2^{n+sm}}$. So, when step s is not small enough, the size of L^s will be too large to be calculated in a memory restricted computer. But the main result in this paper requires us to check $\mathcal{J}_0^{(s)}$. When applying to the simplest case — Boolean network, since $\mathcal{J}_0^{(s)} \in \mathcal{L}_{2^n \times 2^m}$, $\forall s$, it is always easily computable (as long as the first step is computable).

The major shortage of this result is still the computational complexity. Since the controllability matrix is of the dimension $2^n \times 2^n$, it is a large matrix when n is not small. Using our toolbox (http://lsc.amss.ac.cn/dcheng/STP/stp.zip) based on MatLab and calculated in PC, we can only deal with $n \leq 25$ or so.

References

[1] S. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets., J. Theoret. Biol. 22 (3) (1969) 437.
[2] S. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York, 1993.
[3] S. Kauffman, At Home in the Universe, Oxford Univ. Press, 1995.
[4] R. Albert, H. Othmer, The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in drosophila melanogaster, J. Theoret. Biol. 223 (1) (2003) 1–18.
[5] M. Aldana, Boolean dynamics of networks with scale-free topology, Physica D 185 (1) (2003) 45–66.
[6] B. Drossel, T. Mihaljev, F. Greil, Number and length of attractors in a critical Kauffman model with connectivity one, Phys. Rev. Lett. 94 (8) (2005) 88701.
[7] A. Datta, A. Choudhary, M. Bittner, E. Dougherty, External control in Markovian genetic regulatory networks: The imperfect information case, Bioinformatics 20 (2004) 924–930.
[8] R. Pal, A. Datta, M.L. Bittner, E.R. Dougherty, Optimal infinite horizon control for probabilistic Boolean networks, IEEE Trans. Signal Process. 54 (2006) 2375–2387.
[9] T. Akutsu, M. Hayashida, W. Ching, M. Ng, Control of Boolean networks: Hardness results and algorithms for tree structured networks, J. Theoret. Biol. 244 (4) (2007) 670–679.
[10] D. Cheng, Sime-Tensor Product of Matrices and its Applications—a Survey, in: Proc. 4th International Congress of Chinese Mathematicians, Higher Edu. Press, Int. Press, Hangzhou, 2007, pp. 641–668.
[11] D. Cheng, H. Qi, Z. Li, J.B. Liu, Stability and stabilization of boolean networks, Int. J. Robust Nonlinear Contr., in press (doi:10.1002/rnc.1581).
[12] D. Cheng, Input-state approach to boolean networks, IEEE Trans. Neural Netw. 20 (3) (2009) 512–521.
[13] D. Cheng, Z. Li, Solving logic equation via matrix expression, Front. Electr. Electron. Eng. China 4 (3) (2009) 259–269.
[14] D. Cheng, Z. Li, H. Qi, Realization of boolean control networks, Automatica 46 (1) (2010) 62–69.
[15] Y. Mu, L. Guo, Optimization and identification in a non-equilibrium dynamic game, in: Proc. CDC-CCC'09, 2009, pp. 5750–5755.
[16] Z. Li, D. Cheng, Algebraic approach to dynamics of multi-valued networks, Int. J. Bifurcation and Chaos 20 (3) (2010) 561–582.