第 31 卷第 7 期
2014 年 7 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 31 No. 7
Jul. 2014

DOI: 10.7641/CTA.2014.40112

具有少量基本回路布尔网络的不动点

赵千川†

(清华大学 自动化系 智能与网络化系统研究中心; 清华信息科学与技术国家实验室(筹), 北京 100084)

摘要: 近来作为自然和人造非线性动态网络的一种紧凑模型, 布尔网络的研究受到广泛关注. 不动点和吸引子是预测布尔网络长期行为的关键. 本文针对具有少量基本回路的布尔网络, 提出了确定不动点的算法. 我们的方法是基于构成反馈顶点集的变量所满足的一组方程. 作为应用, 我们还给出了检验这类布尔网络全局稳定性的充要条件.

关键词: 不动点; 布尔网络; 反馈顶点集; 全局稳定性; NP–难性

中图分类号: TP273　　　**文献标识码**: A

Fixed points of Boolean networks with small number of elementary circuits

ZHAO Qian-chuan†

(Center for Intelligent and Networked Systems, Department of Automation;
Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 1000084, China)

Abstract: Boolean networks have been studied intensively recently due to their importance as a compact model for understanding both natural and man-made nonlinear dynamic networks. Fixed points and attractors are keys to predict long term behavior of Boolean networks. We develop algorithms for finding fixed point of Boolean networks with small number of elementary circuits, based on a set of equations on variables forming a feedback vertex set. As an application, we also present a sufficient and necessary condition for checking the global stability for such networks.

Key words: fixed point; Boolean network; feedback vertex set; global stability; NP–hardness

1 Introduction

Boolean networks have been studied intensively recently due to their importance as a compact model for understanding both natural and man-made nonlinear dynamic networks. Fixed points and attractors are key to predict long term behavior of Boolean networks. Among literature, using semi-tensor product, [1] establishes a powerful systematic theory on analysis and control Boolean networks and provides a linear characterization of fixed points; [2] employs graph theoretic methods to analyze complexity of Boolean networks; [3] introduces scalar equations to find attractors of Boolean networks and demonstrated inspiring simple and beautiful results. Unfortunately, since many analysis and control problems of Boolean networks have been shown to be NP–hard (see [2] for a list of them), it is important to identify tractable special classes of Boolean networks.

In this paper, we propose efficient fixed point finding algorithms for Boolean networks with small number of elementary circuits. Our method is based on a set of equations on variables forming a feedback vertex set[4]. Our result extends the work on a special class of Boolean networks known as Regulatory Boolean networks (REBN)[5] to general Boolean networks. As an application, we also present a sufficient and necessary condition for checking global stability[6] for such networks. It should be noted that the problem of finding minimum feedback vertex set for general Boolean networks is NP–complete[7]. In practice, it is sufficient to work on Boolean networks with small number of circuits.

The remaining part of the paper is organized as follows. In Section 2, we introduce notations and definitions to set up the fixed point problem for Boolean networks. In Sections 3, we will present the main results: conditions for finding fixed points of Boolean networks and checking global stability on reduced sets of variables (feedback vertex sets). In Section 4, we will show in general the fixed point problem for Boolean networks with many elementary circuits is NP–complete. In Section 5, we provide an example to demonstrate the theoretical results. Finally, Section 6 concludes the paper.

2 Definitions and system description

Definition 1 A (synchronized) Boolean network Σ is a directed graph $G = (V, E)$ whose individual vertices i are attached a Boolean state variable x_i,

Received 19 February 2014; accepted 30 May 2014.
†Corresponding author. E-mail: zhaoqc@tsinghua.edu.cn; Tel.: +86 10-62783612.
This work is supported by National Natural Science Foundation of China (Nos. 61074034, 61021063, 61174072, 61174105).

$i \in V = \{1, \cdots, n\}$ and updated by a Boolean equation

$$x_i(t+1)=f_i(x_j(t), j\in\Gamma(i)),\ \forall i\in V,\ t=0,1,\cdots, \tag{1}$$

where $f_i: B^{|\Gamma(i)|} \to B$ is a Boolean function with $B = \{0, 1\}$, and $\Gamma(i) = \{j \in V|(j, i) \in E\}$ is the set of head vertices of input arcs of vertex i, $|\Gamma(i)|$ is the size of $\Gamma(i)$. Some time, we also use the notion $f_i(X)$ to indicate f_i as a function of $X = (x_1, x_2, \cdots, x_n)$ with the understanding that $f_i(X) = f_i(x_j(t), j \in \Gamma(i))$. The set of functions $F = \{f_i, i \in V\}$ is called vertex function set of Σ. We also use the vector form of the state equation Eq.(1),

$$X(t+1) = F(X(t)),\ t = 0, 1, \cdots. \tag{2}$$

We use the notations $G_\Sigma, V_\Sigma, E_\Sigma, \Gamma_\Sigma(i), F_\Sigma$ to emphasis the dependent of $G, V, E, \Gamma(i), F$ on Σ.

Definition 2 A sequence of vertices $P = i_1, i_2, \cdots, i_p$ is called a path of Σ if $(i_j, i_{j+1}) \in E_\Sigma$ for $j = 1, 2, \cdots, p-1$.

Definition 3 A path $C = i_1, i_2, \cdots, i_c$ is called a circuit of Σ if $i_1 = i_c$.

Definition 4 A circuit $C = i_1, i_2, \cdots, i_c$ is called an elementary circuit of Σ if $i_j \neq i_k$ for all $j,\ k = 1, 2, \cdots, c-1$ if $j \neq k$.

Definition 5 A network is called acyclic, if it has no circuits.

Definition 6 A set of vertices V_e is called a feedback vertex set if the network generated by removing the vertices in V_e and arcs related to vertices in V_e is acyclic.

Definition 7 A state $X = (x_1, x_2, \cdots, x_n) \in B^n$ is called a fixed point of Σ, if $x_i = f_i(X_i)$, for all $i \in V_\Sigma$ and $X_i = (x^i_j, j \in \Gamma_\Sigma(i))$, or in vector form

$$X = F(X).$$

Definition 8 A sequence of states $X^1, X^2, \cdots, X^c \in B^n$ is called an attractor of Σ, if $x^j_i = f_i(X^{j-1})$, for all $i \in V_\Sigma$, $j = 2, \cdots, c$ and $x^1_i = f_i(X^c)$, for all $i \in V_\Sigma$, or in vector form

$$X^j = F(X^{j-1}),\ j = 2, \cdots, c$$

and

$$X^1 = F(X^c).$$

3 Elementary circuits and simplified conditions

Without loss of generality, we assume that the Boolean network under study is strongly connected (for all pair of vertices i, j, there is at least one path from i to j in the network).

We can define a new network from a given strongly connected Boolean network.

Algorithm 1 Constructing a new network Σ' based on a feedback vertex set V_e.

Input: a strongly connected Boolean network Σ whose graph is $G = (V, E)$ and a feedback vertex set V_e (formed by picking one vertex from each elementary circuit of G).

Output: an Boolean network Σ' having graph $G' = (V', E')$ with new set of vertex functions F'.

Step 1 Extend the network by adding a copy of vertex set $V_\mathrm{e} = \{i_1, \cdots, i_{n_\mathrm{e}}\}$ such that the new network has $n + n_\mathrm{e}$ vertices and assume the added vertices have indices $n+1, n+2, \cdots, n+n_\mathrm{e}$ respectively. Or formally, $V' = V \cup V'_\mathrm{e}$ where $V'_\mathrm{e} = \{n+1, n+2, \cdots, n+n_\mathrm{e}\}$.

Step 2 Redirect the arcs pointing to vertices in V_e to vertices in $V'_\mathrm{e} = \{n+1, n+2, \cdots, n+n_\mathrm{e}\}$ and add single input arcs from vertices in V'_e to V_e. Formally, we set $E' = E\backslash I(V_\mathrm{e}) \cup O(V'_\mathrm{e}) \cup D(V'_\mathrm{e}, V_\mathrm{e})$ where $I(V_\mathrm{e}) = \{(i, j)|i \in V, j \in V_\mathrm{e}, (i, j) \in E\}$, $O(V_\mathrm{e}) = \{(i, j+n)|i \in V, j \in V_\mathrm{e}\,(i, j) \in E\}$, and $D(V'_\mathrm{e}, V_\mathrm{e}) = \{(n+i, i)|i \in V_\mathrm{e}\}$.

Step 3 The set of vertex functions F' is obtained by associating vertex functions of vertices in V'_e with the vertex functions in V_e and change vertex functions of vertices in V_e to the single input function $f_i(x_{n+i})' = x_{n+i}$, for $i \in V_\mathrm{e}$. Formally, we set $f'_i = x_{n+i}$ if $i \in V_\mathrm{e}$; $f'_i = f_i$ if $i \in V\backslash V_\mathrm{e}$; $f'_{n+i} = f_i$ if $i \in V_\mathrm{e}$.

Remark An important property of the constructed new Boolean network Σ' is that it can be used to find all fixed points of Σ as shown in the following theorem.

Theorem 1 X is a fixed point of Σ iff (X, X_e) is a fixed point of Σ', where $X_\mathrm{e} = (x_i, i \in V_\mathrm{e})$ and Σ' is defined by Algorithm 1.

Proof It is straight forward to verify based on the construction of Σ'.

Algorithm 2 Constructing tree function for vertices in V'_e in Σ'.

Input: a network Σ'.

Output: tree functions of all vertices in V'_e.

Step 1 Remove the arcs in $D(V'_\mathrm{e}, V_\mathrm{e})$ and obtain an acyclic network Σ'_a.

Step 2 Define a vertex set $A = \emptyset$.

Step 3 Assign each vertex of V_e by the tree function $f^\mathrm{T}_i(X_\mathrm{e}) = x_i, i \in V_\mathrm{e}$ and add V_e to A.

Step 4 Assign each vertex i whose input arcs are directly from vertices in A (or equivalently, $\Gamma_{\Sigma'}(i) \subset A$) the tree function $f^\mathrm{T}_i(X_\mathrm{e}) = f_i(X^\mathrm{T}_\mathrm{e})$ to vertex i where $X^\mathrm{T}_\mathrm{e} = (f^\mathrm{T}_j(X_\mathrm{e}),\ j \in \Gamma_{\Sigma'}(i))$. Add vertex i to the set A.

Step 5 Repeat Step 4 until all vertices in V'_e are assigned.

For the set of fixed point equations in Σ', we have the following equivalent form.

Theorem 2 Let X^* be a fixed point of Σ and let $X^*_\mathrm{e} = (x^*_i, i \in V_\mathrm{e})$. Then X^*_e satisfies the following set of equations:

$$x_i^* = f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}}^*),\ i \in V_{\mathrm{e}}, \tag{3}$$

where f_{n+i}^{T} is the tree function obtained for V_{e}' by Algorithm 2.

Proof Note the assignment of tree function to vertices in Algorithm 2 is according to the topology sorting of vertices in Σ_a'. We will prove first that

$$f_i^{\mathrm{T}}(X_{\mathrm{e}}^*) = x_i^*,\ \forall i \in V \cup V_{\mathrm{e}}'.$$

This is true for vertices V_{e} according to the definition of the tree functions f_i^{T} for vertices in V_{e}. Next we use induction to show that this is also true for all vertices in $V \backslash V_{\mathrm{e}}$. Assume for vertices i, whose topology sorting position is no more than $k > 0$, it is true that

$$f_i^{\mathrm{T}}(X_{\mathrm{e}}^*) = x_i^*.$$

We will prove that for vertex j with topology sorting position $k+1$, it is also true that

$$f_j^{\mathrm{T}}(X_{\mathrm{e}}^*) = x_j^*.$$

In fact, we have

$$\begin{aligned}&\varGamma_{\Sigma'}(j) = \varGamma_\Sigma(j),\\ &f_j^{\mathrm{T}}(X_{\mathrm{e}}^*) = f_j(f_i^{\mathrm{T}}(X_{\mathrm{e}}^*), i \in \varGamma_\Sigma(j)) =\\ &f_j(x_i^*,\ i \in \varGamma_\Sigma(j)) = x_j^*.\end{aligned}$$

The last equation is due to the fact that X^* is a fixed point of Σ. According to induction, we thus established that fact that

$$f_i^{\mathrm{T}}(X_{\mathrm{e}}^*) = x_i^*,\ \forall i \in V.$$

Then for all vertices $i \in V_{\mathrm{e}}'$, we have $i - n \in V_{\mathrm{e}} \subset V$,

$$f_i^{\mathrm{T}}(X_{\mathrm{e}}^*) = f_{i-n}(f_j^{\mathrm{T}}(X_{\mathrm{e}}^*), j \in \varGamma_\Sigma(i-n)) = x_{i-n}^*.$$

The proof is completed.

Remark The importance of Theorem 2 lies in the key observation that the number of equations to check for a fixed point now reduces to the size of V_{e}. This extends the result for a special subclass of Boolean networks known as regulatory Boolean networks (REBN)[5] to the general Boolean networks. If the number of elementary circuits is small, the size of feedback vertex set is also small, and the finding of fixed point is tractable. It is possible to apply our results to networks with many arcs (at the order of $\mathrm{O}(n^2)$) since there are orientations of complete graphs such that they have no elementary circuits, i.e., being acyclic (see e.g. [8]). We can employ the powerful semi-tensor production theory developed in [1] to check the conditions in Theorem 2. We will demonstrate this in Section 5.

Theorem 3 Let $X_{\mathrm{e}}^* = (x_i^*, i \in V_{\mathrm{e}})$ be a vector satisfying the following set of equations:

$$x_i^* = f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}}^*),\ i \in V_{\mathrm{e}}, \tag{4}$$

where f_i^{T} is the tree function defined by Algorithm 2. Let $X^* = (x_1^*, x_2^*, \cdots, x_n^*)$ be a vector constructed as

$$x_i^* = f_i^{\mathrm{T}}(X_{\mathrm{e}}^*),\ i \in V.$$

Then X^* is a fixed point of Σ.

Proof We use induction to prove

$$f_i(x_j^*, j \in \varGamma_\Sigma(i)) = x_i^*,\ i \in V.$$

Because

$$\begin{aligned}&f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}}^*) = f_i(f_j^{\mathrm{T}}(X_{\mathrm{e}}^*), j \in \varGamma_\Sigma(i)),\ i \in V_{\mathrm{e}},\\ &x_i^* = f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}}^*), i \in V_{\mathrm{e}},\end{aligned}$$

and

$$x_i^* = f_i^{\mathrm{T}}(X_{\mathrm{e}}^*),\ i \in V,$$

we have

$$\begin{aligned}x_i^* = {}& f_i(f_j^{\mathrm{T}}(X_{\mathrm{e}}^*), j \in \varGamma_\Sigma(i)) =\\ & f_i(x_j^*, j \in \varGamma_\Sigma(i)),\ i \in V_{\mathrm{e}}.\end{aligned}$$

Assume for i whose topology sorting position is no more than $k > 0$ it holds that

$$f_i(x_j^*, j \in \varGamma_\Sigma(i)) = x_i^*.$$

We will prove that for vertex with topology sorting position is $k+1$, it is true that

$$f_i(x_j^*, j \in \varGamma_\Sigma(i)) = x_i^*.$$

In fact,

$$\begin{aligned}x_i^* = {}& f_i^{\mathrm{T}}(X_{\mathrm{e}}^*) = f_i(f_j^{\mathrm{T}}(X_{\mathrm{e}}^*), j \in \varGamma_\Sigma(i)) =\\ & f_i(x_j^*, j \in \varGamma_\Sigma(i)).\end{aligned}$$

The proof is completed.

Remark Algorithm 2 can be applied to find all attractors if we unfold the network k-times starting from V_{e} (denoted as Σ^k) until the tree functions repeats. Thus the algorithms in this section for fixed points can be extended to the study of global stability of the network.

Definition 9 A network is said to be globally stable, if it has a fixed point as its only attractor.

Corollary 1 Let V_{e} is a feedback vertex set of a strongly connected network Σ. Then the network Σ is globally stable, iff Σ' has a single fixed point and Σ^k, $k \geqslant 2$ do not have any fixed point.

Remark The number of k to check in the corollary is bounded above by $2^{|V_{\mathrm{e}}|^2} + 1$ since the set of Boolean functions on $B^{|V_{\mathrm{e}}|}$ are finite and all tree functions generated by unfolding finite times belong to this set. When k is small, we can employ the semi-tensor production theory to check the conditions in Corollary 1. This will be demonstrated by example in Section 5.

4 NP–hardness of Boolean networks with many elementary circuits

In this section, we show that the fixed point problem for Boolean networks with many elementary circuits is NP–complete. The proof is based on a reduction to the well-known NP–complete problem 3–SAT.

Theorem 4 For Boolean networks with no less than $\dfrac{1}{c}n$ elementary circuits ($\dfrac{1}{c} \geqslant 0.5$), the problem of whether they have fixed points is NP–complete.

Proof Please see the Appendix.

5 A numerical example

In this section, we give an example to show how the proposed algorithm works. The example comes from [9] as shown in Fig.1. It is an 8 vertex network Σ with vertex functions are given as

$$A(t+1)=\bar{C}(t),\ B(t+1)=\bar{A}(t),$$
$$C(t+1)=H(t)\cdot\bar{B}(t),\ D(t+1)=C(t),$$
$$E(t+1)=D(t),\ F(t+1)=\bar{E}(t),$$
$$G(t+1)=D(t)+E(t),\ H(t+1)=G(t)\cdot\bar{F}(t).$$

Here and below, we use A,B,C,D,E,F,G,H to label the 8 state variables $x_1,x_2,\cdots,x_8$ and vertex $1,2,\cdots,8$ to avoid complicated indices. So, for instance, by $A(t+1)=\bar{C}(t)$ we mean $x_1(t+1)=\bar{x_3}(t)$ which implies the vertex function f_1 of vertex 1 is $f_1(x_i, i\in\Gamma(1))=\bar{x}_3$, where $\Gamma(1)=\{3\}$.

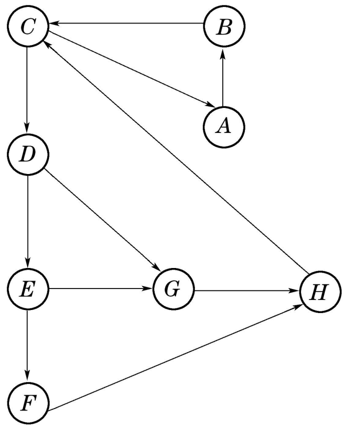

Fig. 1 An 8 vertex example

The network is strongly connected. It has a feedback vertex set $V_{\rm e}=\{C,E\}$. In fact the removal of $V_{\rm e}$ and related arcs (input or output arcs of vertices C and D) will lead to an acyclic graph. Apply Algorithm 1 and we obtain Σ' as shown in Fig.2 where $V_{\rm e}'=\{C',E'\}$ are the added vertices and dashed arcs are the set $D(V_{\rm e}',V_{\rm e})$.

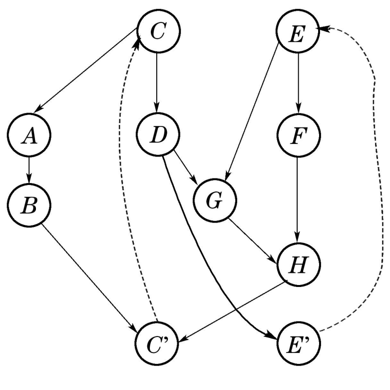

Fig. 2 Constructed Σ' for the network in Fig.1

Starting from $V_{\rm e}=\{C,E\}$, in Σ', we apply Algorithm 2 to obtain the tree functions for each vertex. The result is

$$f_A^{\rm T}(C,E)=\bar{C},\ f_D^{\rm T}(C,E)=C,$$
$$f_F^{\rm T}(C,E)=\bar{E},\ f_B^{\rm T}(C,E)=\bar{f}_A^{\rm T}=C,$$
$$f_G^{\rm T}(C,E)=f_D^{\rm T}(C,E)+E=C+E,$$
$$f_H^{\rm T}(C,E)=f_G^{\rm T}(C,E)\cdot\bar{f}_F^{\rm T}(C,E)=(C+E)\cdot E=E,$$
$$f_{C'}^{\rm T}(C,E)=f_H^{\rm T}(C,E)\cdot\bar{f}_B^{\rm T}(C,E)=E\cdot\bar{C},$$
$$f_{E'}^{\rm T}(C,E)=f_D^{\rm T}(C,E)=C.$$

Then according to Theorem 2, all fixed points of Σ satisfy the following set of equations given by the tree functions on $V_{\rm e}'$:

$$\begin{aligned}C&=f_{C'}^{\rm T}(C,E)=\bar{C}\cdot E,\\ E&=f_{E'}^{\rm T}C,E)=C.\end{aligned}\tag{5}$$

Since the size of $V_{\rm e}$ is small, it is ideal to employ semi-tensor production tools developed in [1] to do the analysis. As usual, we identify logical 1 and 0 respectively with the vectors

$$1\sim\begin{bmatrix}1\\0\end{bmatrix},\ 0\sim\begin{bmatrix}0\\1\end{bmatrix}.$$

Let δ_n^i be the i-th column vector of the identity matrix I_n of dimension n. For example, $\delta_2^1=\begin{bmatrix}1\\0\end{bmatrix}$ and $\delta_2^2=\begin{bmatrix}0\\1\end{bmatrix}$. Let $\Delta_n=\{\delta_n^i|, i=1,2,\cdots,n\}$ be the set of columns of the identity matrix I_n. A matrix L having r rows is called a logical matrix if its columns belong to Δ_n. If a logic matrix

$$L=[\delta_2^{i_1}\ \ \delta_2^{i_2}\ \ \cdots\ \ \delta_2^{i_m}]$$

for notational compactness, we write

$$L=\delta_n[i_1\ \ i_2\ \ \cdots\ \ i_m].$$

A 2×2^r matrix M_σ is said to be the structure matrix of the r-ary logical operator σ if

$$\sigma(p_1,\cdots,p_r)=M_\sigma\ltimes p_1\ltimes\cdots\ltimes p_r.$$

We have

$$M_n=\delta_2[2\ \ 1],$$
$$M_c=\delta_2[1\ \ 2\ \ 2\ \ 2].$$

So, the structure matrix form for Eq.(5) is

$$\begin{aligned}C&=M_cM_nCE=\delta_2[2\ \ 2\ \ 1\ \ 2]CE,\\ E&=C,\end{aligned}\tag{6}$$

and we have

$$CE=LCE=\delta_2[2\ \ 2\ \ 1\ \ 2]CEC.$$

We have $L=M_cM_nCEC=\delta_2[2\ \ 2\ \ 1\ \ 2](I_2\otimes W_{[2]})M_r=\delta_4[3\ \ 3\ \ 2\ \ 4]$, where $W_{[2]}$ is a swap matrix and M_r is a reduce matrix. It is easy to see due to fixed point theory developed in [1] (Chapter 5) that the number of solutions to set of equations Eq.(6) and Eq.(5) is equal to ${\rm tr}(L)$. Thus, they have a unique solution $(C,E)=(0,0)$ since ${\rm tr}(L)=1$. In fact, the only one appears on the diagonal of L is at $\delta_4^4\sim(0,0)$. Based on the solution, we can determine a corresponding fixed point of the original network Σ based on the evaluation of the tree functions at $(C,E)=(0,0)$ as

$$\begin{aligned} &f_A^{\mathrm{T}}(0,0)=\bar{0}=1,\ f_D^{\mathrm{T}}(0,0)=0,\\ &f_F^{\mathrm{T}}(0,0)=\bar{0}=1,\ f_B^{\mathrm{T}}(0,0)=0,\\ &f_G^{\mathrm{T}}(0,0)=0+0=0,\ f_H^{\mathrm{T}}(0,0)=0. \end{aligned}$$

The fixed point of Σ is $(A,B,C,D,E,F,G,H)=(1,0,0,0,0,1,0,0)$.

It is clear Σ has a single fixed point. We further check whether it is globally stable. To do this, we apply the corollary in Section 3. The k-th unfolding of Σ gives the k-fold composition of the tree functions

$$f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}})$$

as

$$g_i^k(X_{\mathrm{e}})=f_{n+i}^{\mathrm{T}}(g_i^{k-1}(X_{\mathrm{e}}))$$

with $g_i^1(X_{\mathrm{e}})=f_{n+i}^{\mathrm{T}}(X_{\mathrm{e}})$. Again, we employ semi-tensor product to facility the check of solutions of

$$\begin{aligned} C&=g_C^k(C,E),\\ E&=g_E^k(C,E), \end{aligned}$$

when $k=2,3,\cdots$. It is easy to see that we only need to check $\mathrm{tr}(L^k)$, $k=2,3,\cdots$. We have $L^2=\delta_4[2\ 2\ 3\ 4]$ with $\mathrm{tr}(L^2)=3$ and $L^3=L=\delta_4[3\ 3\ 2\ 4]$. Since $L^3=L$, we stop at $k=3$. From $\mathrm{tr}(L^2)=3$ we know that the network has two attractors beside the fixed point found before, and the system is not globally stable. The two solutions to $CE=L^2CE$ are δ_4^2 and δ_4^3 which corresponding to $(C,E)=(1,0)$ and $(C,E)=(0,1)$.

6 Conclusions

In this paper, we study the fixed points of Boolean networks with small number of elementary circuits. Reduced size vertex sets known as feedback vertex sets composed by picking up one vertex from one elementary circuits are used to find all fixed points of original networks. As an application, we introduce conditions to check global stability of such networks. We also prove the negative result that testing whether networks with large number of elementary circuits (no less than half of vertex set size) have fixed points is NP–complete. Future work includes the study of control and observation of Boolean networks with small number of elementary circuits. It is also interesting to investigate the stabilization problems asked in [6].

References:

[1] CHENG D Z, QI H S, LI Z Q. *Analysis and Control of Boolean networks* [M]. New York: Springer-Verlag, 2011.

[2] LASCHOV D, MARGALIOT M, EVEN G. Observability of boolean networks: a graph-theoretic approach [J]. *Automatica*, 2013, 49(8): 2351 – 2362.

[3] FARROW C, HEIDEL J, MALONEY J, et al. Scalar equations for synchronous boolean networks with biological applications [J]. *IEEE Transactions on Neural Networks*, 2004, 15(2): 348 – 354.

[4] SMITH G W, WALFORD R B. The identification of a minimal feedback vertex set of a directed graph [J]. *IEEE Transactions on Circuits Systems*, 1975, 22(1): 9 – 14.

[5] ARACENA J. Maximum number of fixed points in regulatory boolean networks [J]. *Bulletin of Mathematical Biology*, 2008, 70(5): 1398 – 1409.

[6] CHENG D Z, QI H S, LI Z Q, et al. Stability and stabilization of boolean networks [J]. *International Journal of Robust and Nonlinear Control*, 2011, 21(2): 134 – 156.

[7] KARP R M. Reducibility between combinatorial problems [M] //MILLER R E, THATCHER J W. *Complexity of Computer Computations*. New York: Plenum Press, 1972: 85 – 103.

[8] AIGNER M, TRIESCH E, TUZA Z. Searching for acyclic orientations of graphs [J]. *Discrete Mathematics*, 1995, 144(1/2/3): 3 – 10.

[9] HEIDEL J, MALONEY J, FARROW C. Finding cycles in synchronous boolean networks with applications to biochemical systems [J]. *International Journal of Bifurcation and Chaos*, 2003, 13(3): 535 – 552.

Appendix The proof of Theorem 4

Let $x_1,x_2,\cdots,x_n$ are n Boolean variables. Let $c_1,c_2,\cdots,c_m$ are a set of clauses defined by multiplication of three or less letters or their negation from $x_1,x_2,\cdots,x_n$. The 3–SAT problem is to determine, for the Boolean function f defined by the sum of the clauses as

$$f(X)=\sum_{j=1}^{m}c_j(X), \tag{A1}$$

whether there is an assignment of $X=(x_1,x_2,\cdots,x_n)$ on B^n such that

$$f(X)=1.$$

Here we use $c_j(X)$ to emphasis the dependence of c_j on X. An example is the function $f(x_1,x_2,x_3,x_4,x_5)=c_1\cdot c_2\cdot c_3$ where $c_1=x_1+x_2$, $c_2=x_2+\bar{x_3}$,$c_3=x_3+\bar{x_4}+x_5$, or

$$\begin{aligned} &f(x_1,x_2,x_3,x_4,x_5)=\\ &(x_1+x_2)\cdot(x_2+\bar{x_3})\cdot(x_3+\bar{x_4}+x_5), \end{aligned} \tag{A2}$$

where $\bar{a}$ stands for the negation of a. It is straightforward to see the problem in NP since the check of whether a given vector X is a fixed point is polynomial.

To proof the NP–hardness part of the theorem, we construction a Boolean network Σ_{f} for the Boolean function f defined in Eq.(A1) as follows. The Boolean network Σ_{f} has a vertex set $V=\{1,2,\cdots,2n+2m\}$ of size $n+m+m+n$ vertices which is divided as four subsets $V=V_1\cup V_2\cup V_3\cup V_4$. For vertices in the set $V_1=\{1,2,\cdots,n\}$, we label their state variables as $x_1,x_2,dots,x_n$. For vertices in the set $V_2=\{n+1,n+2,\cdots,n+m\}$, we label their state variables as $c_1,c_2,\cdots,c_m$. For vertices in the set $V_3=\{n+m+1,n+m+2,\cdots,n+m+m\}$, we label their state variables as $\sigma_1,\sigma_2,\cdots,\sigma_m$. For vertices in the set $V_4=\{n+2m+1,n+2m+2,\cdots,n+2m+n\}$, we label their state variables as $\psi_1,\psi_2,\cdots,\psi_n$. It has arc set defined as $E=E_1\cup E_2\cup E_3\cup E_4\cup E_5\cup E_6\cup E_7$ with

$$\begin{aligned} &E_1=\{(i,i)|i\in\{1,2,\cdots,n\}\},\\ &E_2=\{(i,n+j)|i\in\{1,2,\cdots,n\},\\ &\qquad x_i \text{ or its negation appears in } c_j\},\\ &E_3=\{(n+m+j,n+m+j+1)|j\in\{1,2,\cdots,m-1\}\}\\ &\qquad\cup\{(n+1,n+m+1)\},\\ &E_4=\{(n+m+j,n+2m+j)|j\in\{1,2,\cdots,m\}\},\\ &E_5=\{(n+m+2m,n+2m+1)\},\\ &E_6=\{(n+2m+i,n+2m+i+1)|i\in\{1,2,\cdots,n-1\}\},\\ &E_7=\{(n+2m+i,i)|i\in\{1,2,\cdots,n\}\}. \end{aligned}$$

The vertex function is defined as: for vertex i in V_1 with state variable x_i,

$$f_i = x_i \cdot \psi_i + \bar{x}_i \cdot \bar{\psi}_i;$$

for vertex $n+j$ in V_2 with state variable c_j,

$$f_{n+j} = c_j;$$

for vertex $n+m+j$ in V_3 with stat variable ω_j,

$$f_{n+m+1} = c_1,$$
$$f_{n+m+j} = \omega_{j-1} \cdot c_j$$

if $j = \{2, \cdots, m\}$; for vertex $n+2m+i$ in V_4 with stat variable ψ_i,

$$f_{n+2m+1} = \omega_m,$$
$$f_{n+2m+i} = \psi_{i-1}$$

if $i = \{2, \cdots, n\}$. For vertices in V_3, we can see easily their vertex functions can also be expressed as

$$f_{n+m+j} = \prod_{k=1}^{j} c_k.$$

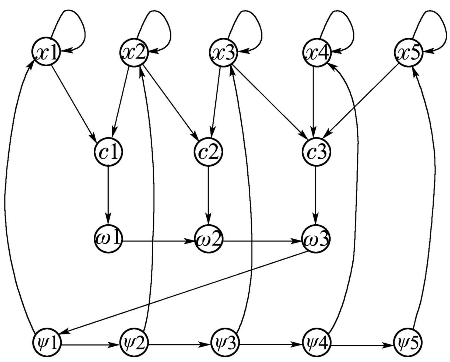

Fig. A Boolean network solving 3–SAT problem, a 5 variable example

To illustrate, we provide the construction for the example in Eq.(A2) in Fig.A.

Notice that the constructed Boolean network $\Sigma_{\rm f}$ has at least $n+m$ elementary circuits. So the number of elementary circuits compared with the scale of the network is at least $\dfrac{n+m}{2n+2m} = 0.5$. Below we will show that with such a network, we can solve the 3–SAT problem by testing whether the network has a fixed point.

On one hand, if there is an assignment of $X^0 = (x_1^0, x_2^0, \cdots, x_n^0)$ such that $f(X^0) = 1$, then we know there must be at least one j_0 such that $c_{j_0}(X^0) = 1$, as a result $\sum\limits_{k=1}^{m} c_k(X^0) = 1$. In the Boolean network $\Sigma_{\rm f}$, if we set the initial values of vertices in V_1 as $x_1^0, x_2^0, \cdots, x_n$, the initial values of vertices in V_2 as $c_1(X^0), c_2(X^0), \cdots, c_n(X^0)$, the initial values of vertices in V_3 as $\omega_j(0) = \prod\limits_{k=1}^{j} c_k(X^0), j = 1, \cdots, m$ the initial values of vertices in V_4 as $\psi_i(0) = 1, i = 1, \cdots, n$ then for vertex i in V_1, we have

$$x_i(1) = x_i(0) \cdot \psi_i(0) + \bar{x}_i(0) \cdot \bar{\psi}_i(0) =$$
$$x_i(0) \cdot 1 + \bar{x}_i(0) \cdot 0 = x_i(0) = x_i^0;$$

for vertex $n+j$, we have

$$c_j(1) = c_i(X(1)) = c_j(X^0) = c_j(0);$$

for vertex $n+m+j$, we have

$$\omega_j(1) = \prod_{k=1}^{j} c_k(X(1)) = \prod_{k=1}^{j} c_k(X^0) = \omega_j(0);$$

for vertex $n+2m+i$, we have

$$\psi_i(1) = \psi_1(1) = \omega_{\rm m}(1) = \omega_{\rm m}(0) = 1.$$

If we denote

$$C^0 = (c_1(X^0), c_2(X^0), \cdots, c_n(X^0)),$$
$$\Omega^0 = (\prod_{k=1}^{j} c_k(X^0), j = 1, 2, \cdots, m)$$

and $\Psi^0 = (1, 1, \cdots, 1)$, this implies that $X^0, C^0, \Omega^0, \Psi^0$ is a fixed point of $\Sigma_{\rm f}$.

Meanwhile, let $(X^0, C^0, \Omega^0, \Psi^0)$ be a fixed point of $\Sigma_{\rm f}$ where $X^0 = (x_1^0, x_2^0, \cdots, x_n^0)$, $C^0 = (c_1^0, c_2^0, \cdots, c_m^0)$, $\Omega^0 = (\omega_1^0, \omega_2^0, \cdots, \omega_m^0)$, $\Psi^0 = (\psi_1^0, \psi_2^0, \cdots, \psi_n^0)$ are vectors specify the state values of the fixed point in vertex sets V_1, V_2, V_3 and V_4, respectively. Set the fixed point as the initial state of the network. Since $(X^0, C^0, \Omega^0, \Psi^0)$ is a fixed point of the network, we have for vertices x_i in V_1 that

$$x_i^0 = x_i(1) = f_i(X^0, C^0, \Omega^0, \Psi^0) = x_i^0 \cdot \psi_i^0 + \bar{x}_i^0 \cdot \bar{\psi}_i^0.$$

This implies that $\psi_i^0 = 1, i = 1, 2, \cdots, n$. From $\psi_1^0 = 1$,

$$\psi_1^0 = \psi_1(1) = f_{n+2m+1}(X^0, C^0, \Omega^0, \Psi^0) = \omega_m(1),$$

we know that

$$\omega_m(1) = 1.$$

Furthermore, since

$$\omega_m(1) = f_{n+2m}(X^0, C^0, \Omega^0, \Psi^0) = \prod_{k=1}^{m} c_k(1).$$

We can establish that

$$\prod_{k=1}^{m} c_k(1) = 1.$$

According to the definition of the vertex function of c_j, we have

$$c_j(1) = f_{n+j_0}(X^0, C^0, \Omega^0, \Psi^0) = c_j(X^0).$$

This implies that

$$\prod_{j=1}^{m} c_j(X^0) = 1$$

and

$$f(X^0) = 1.$$

The proof is completed.

To illustrate, let us exam the example and the network in Fig.A again. It is easy to see that $X^0 = (1, 1, 1, 1, 1)$ satisfies $f(X^0)$. In the network, we have $C^0 = (1{+}1, 1{+}0, 1{+}0{+}1) = (1, 1, 1)$, $\Omega^0 = (1, 1, 1)$ and $\Psi^0 = (1, 1, 1, 1, 1)$.

作者简介:

赵千川 (1969–), 男, 第9届(2003年)《关肇直奖》获奖论文作者, 教授, 目前研究方向为网络化系统控制与优化, E-mail: zhaoqc@tsinghua.edu.cn.