Neurocomputing 174 (2016) 588–596

Contents lists available at ScienceDirect

Neurocomputing

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

Finding graph minimum stable set and core via semi-tensor product approach

Jie Zhong [a,b], Jianquan Lu [b,c,*], Chi Huang [d], Lulu Li [e], Jinde Cao [b,f]

[a] Department of Mathematics, City University of Hong Kong, Kowloon, Hong Kong
[b] Department of Mathematics, Southeast University, Nanjing 210096, PR China
[c] School of Automation, Southeast University, Nanjing 210096, PR China
[d] College of Mathematics, Taiyuan University of Technology, Taiyuan 030024, PR China
[e] School of Mathematics, Hefei University of Technology, Hefei 230009, PR China
[f] Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 28 June 2015
Received in revised form
14 August 2015
Accepted 22 September 2015
Communicated by Liang Wang
Available online 21 October 2015

Keywords:
Graph
Minimum stable set
Graph core
Semi-tensor product
Complex networks

ABSTRACT

By resorting to a new matrix product, called semi-tensor product of matrices, this paper investigates the minimum stable set and core of graph, and also presents a number of results and algorithms. By defining a characteristic logical vector and using matrix expressions of logical functions, a new algebraic representation is derived for the externally stable set. Then, based on the algebraic representation, an algorithm is established to find all the externally stable sets. According to this algorithm, a new necessary and sufficient condition is obtained to determine whether a given vertex subset is an absolutely minimum externally stable set or not. Meanwhile, a new algorithm is also obtained to find all the absolutely minimum externally stable sets. Finally, the graph core, which is simultaneously externally stable set and internally stable set, is investigated. Here the internally stable set requires that internal nodes of this set are all disconnected with each other. Using semi-tensor product, some necessary and sufficient conditions are presented, and then an algorithm is established to find all the graph cores. The study of illustrative examples shows that the results/algorithms presented in this paper are effective.

1. Introduction

The development of graph theory was largely inspired and guided by the famous conjecture called Four-Colour Conjecture. The conjecture was solved by Appel and Haken in 1976 [1], which marked a big turning point in the history of graph theory. Since then, graph theory has experienced rapid and explosive growth, due to its important role in the applied mathematics [2–5]. With the rapid development of computer science and combinational optimization [6,7], the versatility of graph makes them indispensable tools to analyze and design the large-scale communication networks [8].

Recently, graph theory has also been widely used in the systematical analysis of neural networks, complex networks, and multi-agent systems, which are some of the hottest topics in the control field. For example, the Laplacian matrix of graph plays an extremely important role in control protocol designs for multi-agent with linear dynamics and synchronization of complex networks. Many fundamental and landmark results have been obtained regarding the Laplacian matrix [9]. A spanning tree T of an undirected graph $\mathcal{G}$ is a subgraph that includes all of the vertices of $\mathcal{G}$. The importance of spanning trees of various special types has been evident, which are widely used in the analysis of multi-agent systems.

The minimum (externally) stable set is another basic and classical problem in graph theory, which has wide applications in competitive markets, stochastic systems, clustering and so on [10–13]. Externally stable set requires that each node which is not belonging to the set is connected at least one node belonging to the set [14]. For example, externally stable set can be used to analyze the stability of 2-D systems [15]. As an opposite problem of graph, the investigation of internally stable set has been studied in [16], which can be seen as an independent set, or a vertex packing of graph. Externally stable set and internally stable set are two important and independent properties in graph theory.

Recently, a new powerful matrix product, called semi-tensor product of matrices was proposed by Cheng and his colleagues [17,18]. This new matrix product provides a way to multiply two matrices with arbitrary dimensions [17]. By resorting to semi-tensor product, a Boolean function can be converted into an algebraic form, and then a Boolean network can be expressed as a discrete algebraic system [17]. This original set-up opens new perspectives on systematical analysis of many problems for Boolean networks. And up to now, many fundamental and landmark results have been presented on calculating

* Corresponding author.
E-mail address: jqluma@seu.edu.cn (J. Lu).

http://dx.doi.org/10.1016/j.neucom.2015.09.073
0925-2312/

fixed points and cycles of Boolean networks [17,19], on the controllability and observability of Boolean networks [20–23], on the stability of Boolean networks [24], on the optimal control of Boolean control networks [25], on the synchronization of Boolean networks [26–32], on graph coloring problems [16], on Kalman decomposition of Boolean control networks [33], on networked evolutionary games [34], on nonlinear feedback shift registers [35] and so on.

It should be noted that one main drawback of the algebraic state expression of Boolean networks is its computational complexity. The algebraic state representation converts a Boolean network with n state-variables into a state-space of size 2^n. Thus, any algorithm based on this approach has an exponential time-complexity. Moreover, many problems like determining fixed points and observability of Boolean control networks have already been proved to be NP-hard. Hence, the computational complexity is intrinsic and also independent of the models adopted to describe Boolean networks. The main contributions of this paper are as follows: (i) some algebraic descriptions have been established to deal with the minimum stable set and graph core; (ii) a set of theoretical results and algorithms have been presented to determine the minimum stable set and graph core.

The rest of this paper is structured as follows. In Section 2, we present some preliminaries on semi-tensor product, k-valued logical variables, pseudo-Boolean function and graph theory, and definitions of (absolutely minimum) externally stable set and (absolutely maximum) internally stable set. In Section 3, we investigate (absolutely minimum) externally stable set and graph core, and then obtain some necessary and sufficient conditions and efficient algorithms to find the externally stable sets and graph cores. The study of numerical examples shows that the obtained results/algorithms are very effectiveness. The conclusion is presented in Section 4.

Notations: The standard notations will be used in this paper. Throughout this paper, $\mathbb{R}^{n\times m}$ denotes the set of real matrices of order $n\times m$, and $\mathbb{N}^+$ denotes the positive integers. $\mathbf{1}_n$ denotes the n-dimensional column vector with all entries being 1, and I_k is the identity matrix of order k. δ_k^j is the j-th column of identity matrix I_k, and Δ_k denotes the set of all k columns of I_k. Let $\mathrm{Col}_j(A)$ be the j-th column of matrix A, and $\mathrm{Col}(A)$ be the set of columns of matrix A.

2. Some preliminaries

In this section, we give an outline of semi-tensor product of matrices, k-valued logical variables, pseudo-Boolean function and graph theory, which will be used in the following sequels.

Now we are in the position to give the definition of semi-tensor product of matrices.

Definition 1 (*Cheng et al. [18]*). For a $n\times m$ matrix A and a $p\times q$ matrix B. Let l be the least common multiple of m and p. Then the semi-tensor product of A and B is defined as follows:

$$A\ltimes B=(A\otimes I_{l/m})(B\otimes I_{l/p}).$$

Here $\otimes$ is the Kronecker product of matrices. We can see that semi-tensor product of matrices is a generalization of conventional matrix product. So sometimes we can omit "$\ltimes$" for convenience.

Now we present some fundamental facts about semi-tensor product and logical matrices.

Definition 2 (*Cheng and Qi [17]*). An $mn\times mn$ matrix $W_{[m,n]}$ is called a swap matrix, if it is constructed in the following way: label its columns by the following sequence $(11,12,\ldots,1n,\ldots,m1,m2,\ldots,mn)$ and similarly label its rows by the following sequence $(11,21,\ldots,m1,\ldots,1n,2n,\ldots,mn)$. Then its element in the position $((I,J),(i,j))$ is assigned as

$$w_{(I,J),(i,j)}=\delta_{i,j}^{I,J}=\begin{cases}1, & I=i,J=j,\\ 0 & \text{otherwise}.\end{cases} \tag{1}$$

When $m=n$, we denote $W_{[m,n]}$ by $W_{[n]}$ or $W_{[m]}$.

Lemma 1 (*Cheng and Qi [17]*). *Let $X\in\Delta_m$ and $Y\in\Delta_n$ be two columns. Then*

$$W_{[m,n]}\ltimes X\ltimes Y=Y\ltimes X,\ W_{[n,m]}\ltimes Y\ltimes X=X\ltimes Y.$$

In this sequel, we denote "True" or "False" by "1" or "0" $\in\mathcal{D}=\{1,0\}$. In this paper, we use the following two logical vectors to represent them:

$$T=1\sim\delta_2^1,\quad F=0\sim\delta_2^2,$$

where $\delta_n{}^i$ denotes the i-th column of identity matrix I_n. Similarly, for a k-valued logical variable $z\in\mathcal{D}_k$, we can equivalently expressed by the following logical vectors:

$$\frac{k-i}{k-1}\sim\delta_k^i,\quad i=1,\ldots,k,$$

where

$$\mathcal{D}_k=\left\{1,\frac{k-2}{k-1},\ldots,\frac{1}{k-1},0\right\}.$$

Here, we set $\Delta_n=\{\delta_n^i\mid 1\le i\le n\}$ for notational ease. Then, when $n=2$, we denote $\Delta:=\Delta_2$, and $\Delta\sim\mathcal{D}$. A $n\times m$ matrix A is called a logical matrix if $A=[\delta_n^{i_1},\ldots,\delta_n^{i_m}]$ and for simplicity, we can express A by $A=\delta_n[i_1,\ldots,i_m]$. Denote the set of $n\times m$ logical matrices by $\mathcal{L}_{n\times m}$.

Lemma 2 (*Cheng and Qi [17]*). *Let $f:(\Delta_2)^n\to\Delta_2$ be a Boolean function. Then there exists a unique matrix $F\in\mathcal{L}_{2\times 2^n}$ such that $f(\sigma_1,\sigma_2,\ldots,\sigma_n)=F\ltimes\sigma_1\ltimes\sigma_2\ldots\ltimes\sigma_n$, for every $(\sigma_1,\sigma_2,\ldots,\sigma_n)\in(\Delta_2)^n$. F is called the structure matrix of f.*

Example 1. Consider the following two logical functions $f(x_1,x_2)=x_1\vee x_2$ and $g(x_1,x_2)=(\neg x_1)\vee x_2$. Then, according to Lemma 2, we can obtain its corresponding structure matrices M_f and M_g satisfying:

$$f(x_1,x_2)=M_f x_1\ltimes x_2=\begin{pmatrix}1&1&1&0\\0&0&0&1\end{pmatrix}x_1\ltimes x_2,$$

$$g(x_1,x_2)=M_g x_1\ltimes x_2=\begin{pmatrix}1&0&1&1\\0&1&0&0\end{pmatrix}x_1\ltimes x_2. \tag{2}$$

Here, we present the structure matrices of some basic logical operators (Negation: $\neg$, $M_n=\delta_2[2,1]$; Conjunction: $\wedge$, $M_c=\delta_2[1,2,2,2]$; Disjunction: $\vee$, $M_d=\delta_2[1,1,1,2]$; the dummy matrix: $E_d=\delta_2[1,2,1,2]$), which has the following property: for any two logical variables u and v, we have $E_d uv=v$, or $E_d W_{[2]}uv=u$. The group power-reducing matrix: $\Phi_n=\delta_{2^{2n}}[1,2^n+2,2\cdot 2^n+3,\ldots,(2^n-2)\cdot 2^n+2^n-1,2^{2n}]$ satisfies $\sigma\ltimes\sigma=\Phi_n\sigma$, for $\sigma\in\Delta_{2^n}$.

If $\ltimes_{i=1}^n\sigma_i=\delta_{2^n}^k$, how can we calculate each σ_i, $i=1,\ldots,n$ from $\ltimes_{i=1}^n\sigma_i=\delta_{2^n}^k$? Then, the following result shows how to calculate σ_i for all i. Firstly, we define the following matrices, which will be