# A matrix approach to graph maximum stable set and coloring problems with application to multi-agent systems

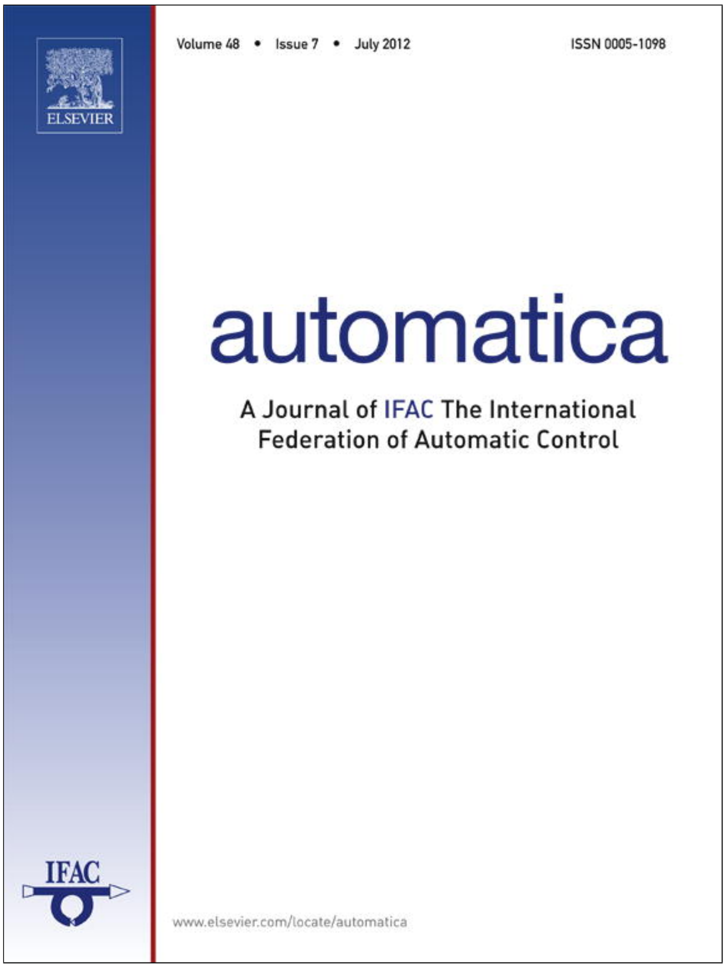
**ARTICLE** *in* AUTOMATICA · JULY 2012

Impact Factor: 3.02 · DOI: 10.1016/j.automatica.2012.03.024



Yuzhen Wang
Xi'an Jiaotong University


Zhenbin Liu
Shandong University


Volume 48 • Issue 7 • July 2012
ISSN 0005-1098
ELSEVIER
automatica
A Journal of IFAC The International
Federation of Automatic Control
IFAC
www.elsevier.com/locate/automatica

Automatica 48 (2012) 1227–1236

Contents lists available at SciVerse ScienceDirect

Automatica

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

# A matrix approach to graph maximum stable set and coloring problems with application to multi-agent systems[☆]

Yuzhen Wang [1], Chenghui Zhang, Zhenbin Liu

*School of Control Science and Engineering, Shandong University, Jinan 250061, PR China*

ARTICLE INFO

*Article history:*
Received 22 May 2011
Received in revised form
4 November 2011
Accepted 11 November 2011
Available online 16 April 2012

*Keywords:*
Graph
Maximum (weight) stable set
Vertex coloring
Necessary and sufficient condition
Algorithm
Multi-agent system
Group consensus

ABSTRACT

Using the semi-tensor product of matrices, this paper investigates the maximum (weight) stable set and vertex coloring problems of graphs with application to the group consensus of multi-agent systems, and presents a number of new results and algorithms. Firstly, by defining a characteristic logical vector and using the matrix expression of logical functions, an algebraic description is obtained for the internally stable set problem, based on which a new algorithm to find all the internally stable sets is established for any graph. Secondly, the maximum (weight) stable set problem is considered, and a necessary and sufficient condition is presented, by which an algorithm to find all the maximum (weight) stable sets is obtained. Thirdly, the vertex coloring problem is studied by using the semi-tensor product method, and two necessary and sufficient conditions are proposed for the colorability, based on which a new algorithm to find all the $k$-coloring schemes and minimum coloring partitions is put forward. Finally, the obtained results are applied to multi-agent systems, and a new protocol design procedure is presented for the group consensus problem. The study of illustrative examples shows that the results/algorithms presented in this paper are very effective.



## 1. Introduction

The maximum (weight) stable set (MSS) problem is a basic one in graph theory, and has found wide applications in many research fields such as computer science, operations research, and so on. The MSS problem is a classical NP-hard problem, and remains NP-complete even for triangle-free graphs and planar graphs of degree three (Garey & Johnson, 1979; Poljak, 1974). During the past decades, the MSS problem has been studied extensively and numerous approaches have been presented for solving or approximating the problem (Goemans & Williamson, 1995; Pardalos & Xue, 1994). In Pardalos and Xue (1994), a brief overview of progress on the MSS problem was given, where the authors described several different formulations of the MSS problem, and summarized a number of both exact algorithms (such as explicit and implicit enumeration) and heuristic algorithms (such as sequential greedy approaches, local and random searches) for the problem. Although many of these algorithms work well on certain classes of graphs, none of them is demonstrated to perform well for a general graph. Thus, it is necessary for us to establish a new formulation and algorithm to solve or approximate the MSS problem for general graphs.

Another basic and classical problem in graph theory is the coloring problem, which is one of the first problems proved to be NP-complete in the early 1970s. Graph coloring shows up in various forms such as vertex coloring, bandwidth coloring, list coloring, set coloring, $T$-coloring, lambda coloring, alpha coloring, and so on. Graph coloring is widely used in many real-life areas such as scheduling and timetabling in engineering, register allocation in compilers, air traffic flow management and frequency assignment in mobile (Barnier & Brisset, 2004; Carter, Laporte, & Lee, 1996; Chaitin, Auslander, & Chandra, 1981; Gamst, 1986). The vertex coloring problem is a well-known coloring problem, in which each vertex of a graph is assigned one color such that no adjacent vertices share the same color. For the coloring problem, two important issues should be considered: one is the $k$-colorability, that is, to determine whether a given graph can be legally colored with at most $k$ different colors, and the other is the so-called minimum coloring problem, namely, to find the smallest number of colors by which the graph can be legally colored. It is noted that there are numerous works on designing algorithms to solve the coloring problem (Blöchliger and Zufferey, 2008; Galinier, Hertz, & Zufferey, 2008; Glass & Pruegel-Bennett, 2005; Hertz, Plumettaz, & Zufferey, 2008; Malaguti, Monaci, & Toth, 2008; Torkestani & Meybodi, 2011; Yánez & Ramírez, 2003). The most popular coloring algorithms belong to three

☆ This work is supported by the National Natural Science Foundation of China (G61074068, G61034007), the Natural Science Foundation of Shandong Province (ZR2010FM013), the Research Fund for the Taishan Scholar Project of Shandong Province, and the Research Fund for the Doctoral Program of Chinese Higher Education (G200804220028). The material in this paper was not presented at any conference. This paper was recommended for publication under the direction of Editor Berç Rüstem.

*E-mail addresses:* yzwang@sdu.edu.cn (Y. Wang), zchui@sdu.edu.cn (C. Zhang), lzbzj@163.com (Z. Liu).

[1] Tel.: +86 531 88392515; fax: +86 531 88392205.

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.03.024

main solution approaches: sequential construction methods (very fast but not very efficient), local search methods (Blöchliger and Zufferey, 2008; Glass & Pruegel-Bennett, 2005), and population-based evolutionary methods, which need to utilize tables to reach the best results (Galinier et al., 2008; Malaguti et al., 2008).

Recently, a new powerful mathematical tool, called the semi-tensor product of matrices, was proposed by Cheng and Qi (2007) and Cheng, Qi, and Li (2011). This tool has been successfully applied to express and analyze Boolean control networks, and up to now many fundamental and landmark results have been presented on calculating fixed points and cycles of Boolean networks (Cheng & Qi, 2010), on the controllability and observability of Boolean networks (Cheng & Qi, 2009) and on their control design problems (Cheng, 2010; Cheng, Li, & Qi, 2010; Cheng, Qi, Li, & Liu, 2011). For other successful applications of the semi-tensor product, see Laschov and Margaliot (2011) and Li and Sun (2011).

In this paper, we investigate the MSS and vertex coloring problems by using the semi-tensor product of matrices, and present a new formulation to deal with these problems. First, by defining a characteristic logical vector and using the matrix expression of logical functions, an algebraic description is obtained for the internally stable set problem, based on which a new algorithm to find all the internally stable sets is established for any graph. Second, the maximum (weight) stable set problem is considered, and a necessary and sufficient condition is presented, by which an algorithm to find all the maximum (weight) stable sets is obtained. Third, the vertex coloring problem is studied by using the semi-tensor product method, and two necessary and sufficient conditions are proposed for the colorability, based on which a new algorithm to determine all the $k$-coloring schemes and minimum coloring partitions is established for any graph. Finally, as an application, we apply the obtained results to multi-agent systems, and present a new protocol design procedure for the group consensus problem of a class of linear multi-agent systems. The study of illustrative examples shows that the results/algorithms presented in this paper are very effective.

It is well worth pointing out that graph theory has been widely used in the analysis and synthesis of multi-agent systems, which are currently one of the hottest topics in the control field (Fax & Murray, 2004; Lin, Francis, & Maggiore, 2005; Lin & Jia, 2009; Su, Wang, & Lin, 2009). In particular, it is shown that the graph Laplacian plays an important role in control protocol designs for groups of agents with linear dynamics, and by it many useful results have been obtained in a series of recent works (Lee & Spong, 2007; Liu & Guo, 2009; Li & Zhang, 2009; Olfati-Saber, 2006; Olfati-Saber & Murray, 2004; Zhang & Tian, 2009). Can the results of this paper help to solve certain problems of multi-agent systems? This is the motivation for applying the main results obtained for graphs to the group consensus problem in this paper.

The main contributions of this paper are as follows. (i) A new mathematical formulation has been established to deal with the MSS and coloring problems, and a set of new theoretical results and algorithms has been presented under this formulation. (ii) A kind of new group consensus is presented for multi-agent systems by using the minimum coloring partition of graphs, and the corresponding protocol design procedure has been obtained for a class of linear multi-agent systems. Compared with the existing results on the MSS and coloring problems, our method has the following advantages. By our method, the problems of graphs can be exactly expressed in an algebraic form of matrices, which is quite different from the existing results (Blöchliger and Zufferey, 2008; Galinier et al., 2008; Goemans & Williamson, 1995; Pardalos & Xue, 1994; Torkestani & Meybodi, 2011; Yánez & Ramírez, 2003). According to our method, to check/determine whether a vertex subset is an MSS or a given graph is $k$-colorable, one only needs to compute a kind of structural matrix, with which the conclusion can be easily obtained. Our method expresses these problems in such a clear way that it may be very helpful for further study of the problems. Moreover, for a given graph with the number of vertices not too large, one can easily use our algorithms to find out all the maximum (weight) stable sets, $k$-coloring schemes or minimum colorings with the help of a computer, although our algorithms are not shown to be more efficient than the existing ones. It should be pointed out that the advantage of our approach lies in the mathematical formulation and exact decision results, not in reducing the computational complexity. In fact, the computational complexity of the MSS and coloring problems cannot be reduced by our algorithms, since the two problems are proved to be NP-hard/complete and it is impossible to design exact algorithms to solve them in polynomial time for a general graph.

The remainder of the paper is organized as follows. Section 2 gives the preliminaries on the semi-tensor product, the pseudo-Boolean function and graph theory. In Section 3, we investigate the MSS and vertex coloring problems, and present the main results of this paper. Section 4 studies the group consensus problem for a class of multi-agent systems. In Section 5, we give several illustrative examples to support our new results; this is followed by the conclusion in Section 6.

## 2. Preliminaries

In this section, we give some necessary preliminaries on the semi-tensor product, the pseudo-Boolean function, and graph theory, which will be used in the following sections.

**Definition 1** (*Cheng and Qi (2007)*).

1. Let $X$ be a row vector of dimension $np$, and $Y$ be a column vector with dimension $p$. Split $X$ into $p$ equal-size blocks as $X^1, \ldots, X^p$, which are $1 \times n$ row vectors. We define the (left) semi-tensor product, denoted by $\ltimes$, as
$$\begin{cases} X \ltimes Y = \sum_{i=1}^{p} X^i y_i \in \mathbb{R}^n, \\ Y^T \ltimes X^T = \sum_{i=1}^{p} y_i (X^i)^T \in \mathbb{R}^n, \end{cases} \tag{1}$$
where $y_i \in \mathbb{R}$ is the $i$-th component of $Y$.
2. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$. If either $n$ is a factor of $p$, say, $nt = p$ and denote it as $A \prec_t B$, or $p$ is a factor of $n$, say, $n = pt$ and denote it as $A \succ_t B$, then we define the (left) semi-tensor product of $A$ and $B$, denoted by $C = \{C^{ij}\} = A \ltimes B$, as follows: $C$ consists of $m \times q$ blocks and each block is defined as
$$C^{ij} = A^i \ltimes B_j, \quad i = 1, \ldots, m,\ j = 1, \ldots, q,$$
where $A^i$ is the $i$-th row of $A$ and $B_j$ is the $j$-th column of $B$.

**Remark 1.** The (left) semi-tensor product of matrices is a generalization of the conventional matrix product. Thus, we can simply call it a "product" and omit the symbol "$\ltimes$" if no confusion arises.

**Definition 2** (*Cheng and Qi (2007)*). A swap matrix $W_{[m,n]}$ is an $mn \times mn$ matrix, defined as follows: its rows and columns are labeled by double index $(i, j)$, the columns are arranged by the ordered multi-index $Id(i, j; m, n)$, and the rows are arranged by the ordered multi-index $Id(j, i; n, m)$. Then the element at the position $[(I, J), (i, j)]$ is
$$w_{(I,J),\,(i,j)} = \delta_{i,\,j}^{I,\,J} = \begin{cases} 1, & I = i \text{ and } J = j \\ 0, & \text{otherwise.} \end{cases}$$

**Remark 2.** From Definition 2, it is easy to see that (Cheng & Qi, 2007)
$$W_{[m,n]} XY = YX, \quad \forall X \in \mathbb{R}^m,\ \forall Y \in \mathbb{R}^n. \tag{2}$$

Let "1" and "0" represent the logical "True" and "False", respectively, and $\mathcal{D} := \{1, 0\}$. In many cases (Cheng et al., 2011) we also use the following two vectors to represent them:

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

where $\delta_n^i$ denotes the $i$-th column of the identity matrix $I_n$, and "$\sim$" stands for "identity" or "equivalence". Similarly, a $k$-valued logical variable $A \in \mathcal{D}_k$ can be equivalently represented with the following vectors:

$$\frac{k-i}{k-1} \sim \delta_k^i, \quad i = 1, 2, \ldots, k,$$

where

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

Set

$$\Delta_n := \{\delta_n^i \mid 1 \leqslant i \leqslant n\},$$

and for notational ease, let $\Delta := \Delta_2$ and $\Delta \sim \mathcal{D}$.

An $n \times t$ matrix $M$ is called a logical matrix if

$$M = [\delta_n^{i_1} \delta_n^{i_2} \cdots \delta_n^{i_t}],$$

and for compactness, we express $M$ briefly as

$$M = \delta_n[i_1 i_2 \cdots i_t].$$

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

**Lemma 1** (Cheng et al. (2011)). *Let $f(x_1, x_2, \ldots, x_s)$ be a Boolean function. Then, there exists a unique matrix $M_f \in \mathcal{L}^{2\times 2^s}$, called the structural matrix of $f$, such that*

$$f(x_1, x_2, \ldots, x_s) = M_f \ltimes_{i=1}^s x_i, \quad x_i \in \Delta, \tag{3}$$

*where $\ltimes_{i=1}^s x_i := x_1 \ltimes x_2 \ltimes \cdots \ltimes x_s$.*

**Remark 3.** The first row of the structural matrix $M_f$ corresponds to the truth values of the logical function $f(x_1, x_2, \ldots, x_s)$.

Now, we list the structural matrices of some basic Boolean operators (Cheng et al., 2011), which will be used later.

Negation ($\neg$): $M_n = \delta_2[2\ 1]$.

Conjunction ($\wedge$): $M_c = \delta_2[1\ 2\ 2\ 2]$.

Dummy operator ($\sigma_d$): $E_d = \delta_2[1\ 2\ 1\ 2]$.

The structural matrix $E_d$ has the following property: for any two logical variables $u, v \in \Delta$, $E_d uv = v$ or $E_d W_{[2,2]} uv = u$.

The following concept will be used in the next section.

**Definition 3** (Liu and Zhang (1993)). An $n$-ary pseudo-Boolean function $f(x_1, x_2, \ldots, x_n)$ is a mapping from $\mathcal{D}^n$ to $\mathbb{R}$, where $\mathcal{D}^n := \underbrace{\mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}}_{n}$.

A graph $\mathcal{G}$ consists of a vertex (node) set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. If each edge of $\mathcal{G}$, denoted by $e_{ij} = (v_i, v_j) \in \mathcal{E}$, is an ordered pair of two vertices of $\mathcal{V}$, we call $\mathcal{G}$ a directed graph (or digraph); if each edge $e_{ij} \in \mathcal{E}$ implies that $e_{ji} \in \mathcal{E}$, then we call $\mathcal{G}$ an undirected graph. A graph is called simple if each edge $e \in \mathcal{E}$ is described by a pair of two distinct vertices. In a digraph $\mathcal{G}$, a directed path is a sequence of ordered edges of the form $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots$. For node $v_i$, its neighbor set, $\mathcal{N}_i$, is defined as

$$\mathcal{N}_i := \{v_j \mid e_{ji} = (v_j, v_i) \in \mathcal{E}\}. \tag{4}$$

**Definition 4** (Minty (1980)). Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Given a vertex subset $S \subseteq \mathcal{V}$, if $v_i \notin \mathcal{N}_j$ and $v_j \notin \mathcal{N}_i$ hold for any $v_i, v_j \in S$ ($i \neq j$), then $S$ is called an internally stable set (an independent set, or a vertex packing) of $\mathcal{G}$. Furthermore, $S$ is called a maximum internally stable set, if any vertex subset strictly containing $S$ is not an internally stable set. An internally stable $S$ is called an absolutely maximum internally stable set if $|S|$ is the largest among those of all the internally stable sets of $\mathcal{G}$, and the largest $|S|$ is called the internally stable number of $\mathcal{G}$, denoted by $\alpha(\mathcal{G}) = |S|$, where $|S|$ stands for the cardinality of $S$.

**Remark 4.** From Definition 4 it is easy to see that any subset of an internally stable set is also an internally stable one, and any internally stable set can be enlarged to a maximum internally stable one. In particular, the empty set $\emptyset$ is regarded as an internally stable set of any graph.

**Definition 5** (Minty (1980)). Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Given a weight function $w : \mathcal{V} \mapsto \mathbb{R}$, a vertex subset $S \subseteq \mathcal{V}$ is called a maximum weight stable set if $S$ is an internally stable set and $\sum_{i\in S} w(v_i)$ is the largest among those of all the internally stable sets.

## 3. Main results

In this section, we investigate the maximum (weight) stable set and vertex coloring problems by the semi-tensor product method, and present the main results of this paper. First, we consider the internally stable set problem.

Consider a graph $\mathcal{G}$ with $n$ nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. Assume that the adjacency matrix, $A = [a_{ij}]$, of $\mathcal{G}$ is given as

$$a_{ij} = \begin{cases} 1, & v_j \in \mathcal{N}_i \\ 0, & v_j \notin \mathcal{N}_i. \end{cases} \tag{5}$$

**Remark 5.** If $\mathcal{G}$ is not a simple graph, say, there exists $v_i$ such that $(v_i, v_i) \in \mathcal{E}$, then we just let $a_{ii} = 0$ in our study. Thus, without loss of generality, we can assume that $a_{ii} = 0$ holds for all $i$ in the following.

Given a vertex subset $S \subseteq \mathcal{V}$, define a vector $V_S = [x_1, x_2, \ldots, x_n]$, called the characteristic logical vector of $S$, as follows:

$$x_i = \begin{cases} 1, & v_i \in S \\ 0, & v_i \notin S. \end{cases} \tag{6}$$

Let

$$y_i := \begin{bmatrix} x_i \\ \bar{x}_i \end{bmatrix}, \qquad \bar{x}_i = 1 - x_i, \quad i = 1, 2, \ldots, n,$$

$$Y_S := \ltimes_{i=1}^n y_i.$$

Then, we have the following result to determine whether or not $S$ is an internally stable set.

**Theorem 1.** *Assume that $Y_S = \delta_{2^n}^k$. Then, $S$ is an internally stable set of $\mathcal{G}$, if and only if the $k$-th component of the first row of matrix $M_S$ is zero, where*

$$\begin{cases} M_S = \sum_{i=1}^n \sum_{j\neq i} a_{ij} M_{ij}, \\ M_{ij} = M_{ji} = M_c (E_d)^{n-2} W_{[2^j, 2^{n-j}]} W_{[2^i, 2^{j-i-1}]}, \\ \quad (i < j). \end{cases} \tag{7}$$

**Proof.** ($\Rightarrow$) If $S$ is an internally stable set of $\mathcal{G}$, it is easy to see from Definition 4 that for any nodes $v_i, v_j \in \mathcal{V}$, if $a_{ij} = 1$, then either $v_i \notin S$ or $v_j \notin S$ holds, from which and (6) we have $x_i x_j = 0$. Thus, the characteristic vector $[x_1, x_2, \ldots, x_n]$ of $S$ satisfies

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j = \sum_{i=1}^n \sum_{j\neq i} a_{ij} x_i x_j = 0, \tag{8}$$

which is a Boolean equation. Since $x_i x_j = x_j x_i$ ($= x_i \wedge x_j$), without loss of generality, we assume that $i < j$. Then, using (2) and the dummy operator ($E_d$), we have

$$\begin{aligned} y_iy_j &= (E_d)^{n-2}y_{j+1}\dots y_ny_{i+1}\dots y_{j-1}y_1\dots y_{i-1}y_iy_j \\ &= (E_d)^{n-2}W_{[2^j,2^{n-j}]}y_{i+1}\dots y_{j-1}y_1\dots y_iy_jy_{j+1}\dots y_n \\ &= (E_d)^{n-2}W_{[2^j,2^{n-j}]}W_{[2^i,2^{j-i-1}]}y_1\dots y_iy_{i+1}\dots y_{j-1}y_j\dots y_n, \end{aligned}$$

where the product is "$\ltimes$". Thus, $x_ix_j = J_1M_{ij}\ltimes_{i=1}^n y_i$, where $J_1 = [1,\ 0]$, and

$$M_{ij} = M_c(E_d)^{n-2}W_{[2^j,2^{n-j}]}W_{[2^i,2^{j-i-1}]}.$$

Hence, Eq. (8) can be expressed as

$$\begin{aligned} 0 &= \sum_{i=1}^n\sum_{j\neq i} a_{ij}x_ix_j = J_1\sum_{i=1}^n\sum_{j\neq i} a_{ij}M_{ij}\ltimes_{i=1}^n y_i \\ &= J_1M_SY_S, \end{aligned} \tag{9}$$

which implies that $Y_S$ satisfies (9). Noticing that $Y_S = \delta_{2^n}^k \in \Delta_{2^n}$, the $k$-th component of $J_1M_S$ must be zero, that is, the $k$-th component of the first row of $M_S$ is zero. Thus, the necessity is proved.

($\Leftarrow$) If the $k$-th component of the first row of matrix $M_S$ is zero, then it can be seen from the assumption on $Y_S$ that $J_1M_SY_S = 0$. Equivalently, (8) has a solution $[x_1, x_2, \dots, x_n]$. Noticing $x_i \in \mathcal{D}$ and $a_{ij} \geqslant 0$, we know from (8) that $a_{ij}x_ix_j = 0$ holds for any $i \neq j$, which implies that either $v_i \notin S$ or $v_j \notin S$ holds when $a_{ij} = 1$. From the definition of internally stable sets, $S$ is an internally stable set. Thus, the proof is completed. □

From the above proof, we have the following corollary.

**Corollary 1.** *Consider the graph $\mathcal{G}$ in Theorem 1. For each node $v_i \in \mathcal{V}$, we assign it a characteristic logical variable $x_i \in \mathcal{D}$ and let $y_i = [x_i,\ \bar{x}_i]^T$. Then, $\mathcal{G}$ has a non-empty internally stable set if and only if the equation*

$$J_1M_S\ltimes_{i=1}^n y_i = 0 \tag{10}$$

*is solvable. Furthermore, the number of zero components of $J_1M_S$ is just the number of internally stable sets of $\mathcal{G}$.*

**Remark 6.** The structural matrix $M_{ij}$ or $M_S$ in Theorem 1 can be easily calculated with the help of a computer.[2]

Based on the proof of Theorem 1 and Corollary 1, we now establish an algorithm to find all the internally stable sets for a given graph $\mathcal{G}$.

**Algorithm 1.** Given a graph $\mathcal{G}$ with nodes $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, assume that its adjacency matrix is $A = [a_{ij}]$. For each node $v_i$, we assign it a characteristic logical variable $x_i \in \mathcal{D}$ and let $y_i = [x_i,\ \bar{x}_i]^T$. To find all the internally stable sets of $\mathcal{G}$, we can take the following steps.

(1) Compute the matrix $M_S = \sum_{i=1}^n\sum_{j\neq i} a_{ij}M_{ij}$ by (7).

(2) Extract the first row of $M_S$, and denote it by $\beta = [b_1, b_2, \dots, b_{2^n}]$. If for all $i$, $b_i \neq 0$, then $\mathcal{G}$ has no internally stable set and the calculation is stopped. Otherwise, find out all the zero components of $\beta$, and denote their positions in $\beta$ by $i_1, i_2, \dots, i_m$, that is, $b_{i_k} = 0$, $k = 1, 2, \dots, m$.

(3) For each index $i_k$, $k = 1, 2, \dots, m$, consider $\ltimes_{i=1}^n y_i = \delta_{2^n}^{i_k}$. Let (Cheng et al., 2011)

$$\begin{cases} S_1^n = \delta_2[\underbrace{1\cdots 1}_{2^{n-1}}\underbrace{2\cdots 2}_{2^{n-1}}], \\ S_2^n = \delta_2[\underbrace{1\cdots 1}_{2^{n-2}}\underbrace{2\cdots 2}_{2^{n-2}}\underbrace{1\cdots 1}_{2^{n-2}}\underbrace{2\cdots 2}_{2^{n-2}}], \\ \vdots \\ S_n^n = \delta_2[\underbrace{\underbrace{1\ 2}_{2}\cdots\underbrace{1\ 2}_{2}}_{2^{n-1}}], \end{cases} \tag{11}$$

then we have $y_i = S_i^n\ltimes_{j=1}^n y_j = S_i^n\delta_{2^n}^{i_k}$, $i = 1, 2, \dots, n$. Noticing that $y_i = [x_i,\ \bar{x}_i]^T$, we need to check whether $y_i = \delta_2^1$ or $x_i = 1$. Set

$$S(i_k) = \{v_i \mid y_i = \delta_2^1,\ 1 \leqslant i \leqslant n\}, \tag{12}$$

then $S(i_k) \subseteq \mathcal{V}$ is the internally stable set corresponding to $b_{i_k} = 0$, and all the internally stable sets of $\mathcal{G}$ are $\{S(i_k) \mid k = 1, 2, \dots, m\}$.

(4) Let

$$\alpha_0 = \max_{1\leqslant k\leqslant m}\{|S(i_k)|\},\quad \mathcal{S} = \{S(i_k) \mid |S(i_k)| = \alpha_0\},$$

then $\alpha_0$ is the internally stable number of $\mathcal{G}$, that is, $\alpha(\mathcal{G}) = \alpha_0$, and $\mathcal{S}$ is the set of all the absolutely maximum internally stable sets of $\mathcal{G}$.

**Remark 7.** In the above algorithm, with the product $\ltimes_{i=1}^n y_i$, each $y_i$ can be uniquely determined (return to its original value) by the formula $y_i = S_i^n\ltimes_{j=1}^n y_j$ (Cheng et al., 2011).

Algorithm 1 also provides a way to determine the absolutely maximum internally stable sets. In the following, we put forward another method to find all the absolutely maximum internally stable sets. To this end, we present a theorem first.

**Theorem 2.** *Consider a graph $\mathcal{G}$ with $n$ nodes $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$. Given a vertex subset $S \subseteq \mathcal{V}$ with its characteristic vector $V_S = [x_1, x_2, \dots, x_n]$, let $\ltimes_{i=1}^n[x_i,\ \bar{x}_i]^T = \delta_{2^n}^k$, where $1 \leqslant k \leqslant 2^n$ can be uniquely determined. Then, $S$ is an absolutely maximum internally stable set if and only if*

$$b_k = \max_{1\leqslant i\leqslant 2^n}\{b_i\} \geqslant 0, \tag{13}$$

*where*

$$\begin{cases} [b_1, b_2, \dots, b_k, \dots, b_{2^n}] := J_1\widetilde{M}, \\ \widetilde{M} = \sum_{i=1}^n M_i - (n+1)M_S, \\ M_i = (E_d)^{n-1}W_{[2^i,2^{n-i}]},\quad i = 1, 2, \dots, n, \end{cases} \tag{14}$$

*and $M_S$ and $J_1$ are the same as those in Theorem 1.*

**Proof.** From the proof of Theorem 1 and Definition 4, that $S$ is an absolutely maximum internally stable set is equivalent to that $V_S$ is a global solution to the following constrained optimization problem:

$$\max\sum_{i=1}^n x_i, \tag{15}$$

$$\text{s.t.}\ \sum_{i=1}^n\sum_{j\neq i} a_{ij}x_ix_j = 0. \tag{16}$$

[2] A MATLAB toolbox was provided by Professor Daizhan Cheng and his co-workers to deal with all the related computations on the semi-tensor product. For details, see http://lsc.amss.ac.cn/~dcheng/stp/STP.zip.

According to Liu and Zhang (1993), this constrained optimization problem can be changed to find a global maximum point $(x_1^*, \ldots, x_n^*)$ of the following function:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i - (n+1) \sum_{i=1}^{n} \sum_{j \neq i} a_{ij} x_i x_j \tag{17}$$

such that $f(x_1^*, \ldots, x_n^*) \geqslant 0$.

On the other hand, letting $y_i = [x_i, \bar{x}_i]^T$, $i = 1, 2, \ldots, n$, and using (2) and the dummy operator, we have

$$\begin{aligned} y_i &= (E_d)^{n-1} y_{i+1} \ldots y_n y_1 \ldots y_{i-1} y_i \\ &= (E_d)^{n-1} W_{[2^i, 2^{n-i}]} y_1 y_2 \ldots y_i y_{i+1} \ldots y_n \\ &= M_i \ltimes_{j=1}^{n} y_j, \end{aligned}$$

from which we obtain

$$x_i = J_1 M_i \ltimes_{j=1}^{n} y_j, \quad i = 1, 2, \ldots, n. \tag{18}$$

Thus, with (18) and the proof of Theorem 1, the pseudo-Boolean function $f(x_1, x_2, \ldots, x_n)$ can be expressed as

$$f(x_1, x_2, \ldots, x_n) = J_1 \widetilde{M} \ltimes_{i=1}^{n} y_i, \tag{19}$$

where $\widetilde{M}$ is given in (14).

Based on the above analysis and noticing that $\ltimes_{i=1}^{n} [x_i, \bar{x}_i]^T = \delta_{2^n}^k \in \Delta_{2^n}$, we conclude that if $S$ is an absolutely maximum internally stable set, then the $k$-th component of the row vector $J_1 \widetilde{M}$ is the largest among all the non-negative components, which is just what (13) says.

Conversely, if (13) holds, then the $k$-th component of $J_1 \widetilde{M}$ is the largest among all the non-negative components, which implies that $V_S = [x_1, \ldots, x_n]$ is the maximum point of $f(x_1, x_2, \ldots, x_n)$ such that $f(x_1, x_2, \ldots, x_n) \geqslant 0$. According to Liu and Zhang (1993), $V_S = [x_1, \ldots, x_n]$ is a global solution to the constrained optimization problem (15)–(16), which means that $S$ is an absolutely maximum internally stable set, and thus the proof is completed. □

The proof of Theorem 2 provides us with a way to find all the absolutely maximum internally stable sets for a given graph.

**Algorithm 2.** Given a graph $\mathcal{G}$ with nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, for each node $v_i$ we assign it a characteristic logical variable $x_i \in \mathcal{D}$ and let $y_i = [x_i, \bar{x}_i]^T$. To determine all the absolutely maximum internally stable sets of $\mathcal{G}$, we can take the following steps.

(1) Compute the matrix $\widetilde{M}$ given in (14).

(2) Extract the first row of $\widetilde{M}$, that is, $J_1 \widetilde{M}$, and denote it by $[b_1, b_2, \ldots, b_{2^n}]$. If for all $i$, $b_i < 0$, then $\mathcal{G}$ has no absolutely maximum internally stable set (except $\emptyset$) and the calculation is stopped. Otherwise, find out the maximum components of $[b_1, b_2, \ldots, b_{2^n}]$, and set

$$K = \left\{ i_k \mid b_{i_k} = \max_{1 \leqslant i \leqslant 2^n} \{b_i\},\ k = 1, 2, \ldots, m \right\}.$$

(3) For each index $i_k \in K$, $k = 1, 2, \ldots, m$, let $\ltimes_{i=1}^{n} y_i = \delta_{2^n}^{i_k}$. Using the formula (11), compute $y_i = S_i^n \delta_{2^n}^{i_k}$, $i = 1, 2, \ldots, n$. Set

$$S(i_k) = \{v_i | y_i = \delta_2^1,\ 1 \leqslant i \leqslant n\}, \tag{20}$$

then $S(i_k) \subseteq \mathcal{V}$ is the absolutely maximum internally stable set corresponding to $b_{i_k}$, and $\alpha(\mathcal{G}) = |S(i_k)|$. Thus, all the absolutely maximum internally stable sets of $\mathcal{G}$ are $\{S(i_k) \mid k = 1, 2, \ldots, m\}$.

Now, we consider the maximum weight stable set problem of graphs. For this problem, we have the following result.

**Theorem 3.** *Consider a graph $\mathcal{G}$ with $n$ nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, and let a non-negative weight function $w : \mathcal{V} \mapsto \mathbb{R}$ be given. Given a vertex subset $S \subseteq \mathcal{V}$ with its characteristic vector $V_S = [x_1, x_2, \ldots, x_n]$, let $\ltimes_{i=1}^{n} [x_i, \bar{x}_i]^T = \delta_{2^n}^k$, where $1 \leqslant k \leqslant 2^n$ can be uniquely determined. Then, $S$ is a maximum weight stable set if and only if*

$$col_k(\rho) = \max_{1 \leqslant i \leqslant 2^n} \{col_i(\rho)\} \geqslant 0, \tag{21}$$

*where*

$$\begin{aligned} &\rho = J_1 \widehat{M} \in \mathbb{R}^{2^n}, \\ &\widehat{M} = \sum_{i=1}^{n} w(v_i) M_i - \left( \sum_{i=1}^{n} w(v_i) + 1 \right) M_S, \end{aligned} \tag{22}$$

*$M_i$, $M_S$ and $J_1$ are the same as those in Theorem 2, and $col_i(\rho)$ stands for the $i$-th component of $\rho$.*

**Proof.** The proof is similar to that of Theorem 2, and thus it is omitted. □

**Remark 8.** Theorem 3 provides us with an algorithm to find all the maximum weight stable sets for a given graph. The algorithm is similar to Algorithm 2, and we omit it here.

In the following, we study the vertex coloring problem of graphs. First, we give some necessary concepts.

**Definition 6.** An $n$-ary $k$-valued pseudo-logic function $f(x_1, x_2, \ldots, x_n)$ is a mapping from $\Delta_k^n$ to $\mathbb{R}^k$.

Assume that $A = [a_{ij}] \in \mathbb{R}^{n \times m}$, $B = [b_{ij}] \in \mathbb{R}^{n \times m}$, then we can define

$$A \odot B = [c_{ij}], \quad c_{ij} = a_{ij} b_{ij},\ i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m,$$

which is called the Hadamard product of matrices/vectors.

Now we are ready to study the coloring problem. Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, and assume that its adjacency matrix is $A = [a_{ij}]$. Let a coloring mapping $\phi : \mathcal{V} \longmapsto N := \{c_1, \ldots, c_k\}$ be given, where $c_1, \ldots, c_k$ stand for $k$ kinds of different colors. Here, $\phi$ is not necessarily surjective.

*The coloring problem* is to find a suitable coloring mapping $\phi$ such that for any $v_i, v_j \in \mathcal{V}$, if $(v_i, v_j) \in \mathcal{E}$, then $\phi(v_i) \neq \phi(v_j)$.

For each vertex $v_i \in \mathcal{V}$, we assign it a $k$-valued characteristic logical variable $x_i \in \Delta_k$ as follows:

$$x_i = \delta_k^j, \quad \text{if } \phi(v_i) = c_j \in N,\ i = 1, 2, \ldots, n. \tag{23}$$

Then, we have the following results.

**Theorem 4.** *Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, and let a coloring mapping $\phi : \mathcal{V} \longmapsto N = \{c_1, \ldots, c_k\}$ be given. Then, the coloring problem is solvable with the given $\phi$, if and only if the $n$-ary $k$-valued pseudo-logic equation*

$$\sum_{i=1}^{n} \sum_{j \neq i} a_{ij} x_i \odot x_j = 0_k \tag{24}$$

*is solvable, where $0_k$ is the $k$-dimensional zero vector.*

**Proof.** ($\Rightarrow$) If the coloring problem is solvable with the given mapping $\phi$, then for any $v_i \neq v_j \in \mathcal{V}$, if $(v_i, v_j) \in \mathcal{E}$, then $\phi(v_i) \neq \phi(v_j)$. Thus, if $a_{ij} = 1$, then $x_i \neq x_j$, which implies that $x_i \odot x_j = 0_k$ or $a_{ij} x_i \odot x_j = 0_k$. With this, it is easy to see that (24) holds true. In other words, (24) is solvable.

($\Leftarrow$) Assume that the Eq. (24) has a solution $(x_1, \ldots, x_n)$. Since $a_{ij} \geqslant 0$ and $x_i \in \Delta_k$, this solution satisfies $a_{ij} x_i \odot x_j = 0_k$, $i, j = 1, 2 \ldots, n$, $i \neq j$. Thus, if $a_{ij} = 1$, then $x_i \odot x_j = 0_k$, which implies that $x_i \neq x_j$ or $\phi(v_i) \neq \phi(v_j)$. Hence, $\phi$ is a solution to the coloring problem. □

**Theorem 5.** *Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, and let a color set $N = \{c_1, \ldots, c_k\}$ be given. Then, the coloring problem of $\mathcal{G}$ is solvable with a mapping $\phi : \mathcal{V} \longmapsto N$, if and only if*

$$0_k \in Col(M) := \{col_1(M), \ldots, col_{k^n}(M)\}, \tag{25}$$

*where*

$$M = \sum_{i=1}^{n} \sum_{j \neq i} a_{ij} M_{ij}^H, \tag{26}$$

*and $M_{ij}^H \in \mathbb{R}^{k \times k^n}$, given in (30) below, is the structural matrix of the $k$-valued pseudo-logic function $x_i \odot x_j$.*

**Proof.** First, we show that

$$f(x_1, x_2, \ldots, x_n) := \sum_{i=1}^{n} \sum_{j \neq i} a_{ij} x_i \odot x_j$$

can be expressed as

$$f(x_1, x_2, \ldots, x_n) = M \ltimes_{i=1}^{n} x_i. \tag{27}$$

In fact, there exists a unique matrix $M_{ij}^H \in \mathbb{R}^{k \times k^n}$ (see Remark 9 below) such that

$$f_{ij}(x_1, \ldots, x_n) = x_i \odot x_j = M_{ij}^H \ltimes_{i=1}^{n} x_i.$$

With this, we have

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j \neq i} a_{ij} f_{ij}(x_1, x_2, \ldots, x_n) \\ &= \sum_{i=1}^{n} \sum_{j \neq i} a_{ij} M_{ij}^H \ltimes_{i=1}^{n} x_i = M \ltimes_{i=1}^{n} x_i. \end{aligned}$$

Thus, Eq. (24) can be rewritten as

$$M \ltimes_{i=1}^{n} x_i = 0_k. \tag{28}$$

Noticing that $\ltimes_{i=1}^{n} x_i \in \Delta_{k^n}$, Eq. (28) is solvable if and only if $0_k$ is one of columns of $M$. Thus, the theorem follows from Theorem 4. □

**Remark 9.** The structural matrix $M_{ij}^H$ in Theorem 5 can be calculated as follows. Set (Cheng et al., 2011)

$$\begin{cases} S_{1,k}^n = I_k \otimes \mathbf{1}_{k^{n-1}}, \\ S_{2,k}^n = [\underbrace{I_k \otimes \mathbf{1}_{k^{n-2}}, \ldots, I_k \otimes \mathbf{1}_{k^{n-2}}}_{k}], \\ \vdots \\ S_{n-1,k}^n = [\underbrace{I_k \otimes \mathbf{1}_k, \ldots, I_k \otimes \mathbf{1}_k}_{k^{n-2}}], \\ S_{n,k}^n = [\underbrace{I_k, \ldots, I_k}_{k^{n-1}}], \end{cases} \tag{29}$$

where $\mathbf{1}_j := [\underbrace{1, 1, \ldots, 1}_{j}]$ and $\otimes$ is the Kronecker product. Then,

$$\begin{aligned} f_{ij}(x_1, \ldots, x_n) &= x_i \odot x_j = H_k x_i \ltimes x_j \\ &= H_k S_{i,k}^n (\ltimes_{i=1}^{n} x_i) S_{j,k}^n (\ltimes_{i=1}^{n} x_i) \\ &= H_k S_{i,k}^n [I_{k^n} \otimes S_{j,k}^n] M_{r,k^n} \ltimes_{i=1}^{n} x_i, \end{aligned}$$

where $M_{r,k^n} = \mathrm{Diag}\{\delta_{k^n}^1, \delta_{k^n}^2, \ldots, \delta_{k^n}^{k^n}\}$ is the power-reducing matrix, and $H_k = \mathrm{Diag}\{E_1, E_2, \ldots, E_k\}$ with $E_i = (\delta_k^i)^T$, $i = 1, 2, \ldots, k$. Therefore,

$$M_{ij}^H = H_k S_{i,k}^n [I_{k^n} \otimes S_{j,k}^n] M_{r,k^n}. \tag{30}$$

The proof of Theorem 5 suggests an algorithm to find out all the coloring schemes with no more than $k$ colors for a given graph.

**Algorithm 3.** Assume that $\mathcal{G}$ is a graph with nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, and let a color set $N = \{c_1, c_2, \ldots, c_k\}$ be given. For each node $v_i$, we assign it a $k$-valued characteristic logical variable $x_i \in \Delta_k$. To find all the $k$-colorings of $\mathcal{G}$, that is, to find out all the coloring mappings such that the coloring problem of $\mathcal{G}$ is solved with no more than $k$ different colors, we can take the following steps.

(1) Compute the matrix $M$ given in (26).
(2) Check whether $0_k \in \mathrm{Col}(M)$. If not, the coloring problem with the given color set has no solution, and the computation is stopped. Otherwise, label the columns which equal $0_k$ and set

$$K = \{j \mid \mathrm{col}_j(M) = 0_k\}. \tag{31}$$

(3) For each index $j \in K$, let $\ltimes_{i=1}^{n} x_i = \delta_{k^n}^j$. Using (29), compute $x_i = S_{i,k}^n \ltimes_{i=1}^{n} x_i = S_{i,k}^n \delta_{k^n}^j$, $i = 1, 2, \ldots, n$. With the obtained solution $(x_1, \ldots, x_n)$, define

$$\begin{aligned} S_{c_1}^j &:= \{v_i \mid x_i = \delta_k^1, \ 1 \leqslant i \leqslant n\}, \\ S_{c_2}^j &:= \{v_i \mid x_i = \delta_k^2, \ 1 \leqslant i \leqslant n\}, \\ &\vdots \\ S_{c_k}^j &:= \{v_i \mid x_i = \delta_k^k, \ 1 \leqslant i \leqslant n\}, \end{aligned}$$

then a coloring scheme corresponding to index $j \in K$ is given as: all the vertices in $S_{c_i}^j$ are colored with color $c_i$, $i = 1, 2, \ldots, k$.

(4) All the coloring schemes are given as follows:

$$\phi_j(v_i) = c_s, \quad \text{if } v_i \in S_{c_s}^j,$$
$$i = 1, 2, \ldots, n; \ s = 1, 2, \ldots, k; \ j \in K. \tag{32}$$

That is,

| Color | | Vertices |
|---|---|---|
| $c_1$ | $\longrightarrow$ | $S_{c_1}^j$ |
| $c_2$ | $\longrightarrow$ | $S_{c_2}^j$ |
| $\vdots$ | | $\vdots$ |
| $c_k$ | $\longrightarrow$ | $S_{c_k}^j$ |

for all $j \in K$. The number of coloring schemes is $|K|$.

**Remark 10.** With $x_i \in \Delta_k$, $i = 1, 2, \ldots, n$, the product $\ltimes_{i=1}^{n} x_i$ is well defined. Conversely, with the product $\ltimes_{i=1}^{n} x_i$, each $x_i$ can be uniquely determined (return to its original value) by the formula $x_i = S_{i,k}^n \ltimes_{i=1}^{n} x_i$ (Cheng et al., 2011).

Based on Theorems 4 and 5, we have the following result for $S_{c_i}^j$ defined in Algorithm 3.

**Proposition 1.** *Assume that $0_k \in Col(M)$ holds. Then,*

(1) *each $S_{c_i}^j$ defined in Algorithm 3 is an internally stable set of $\mathcal{G}$, and*
(2) *for each index $j \in K$, $\{S_{c_1}^j, S_{c_2}^j, \ldots, S_{c_k}^j\}$ is a coloring partition of $\mathcal{G}$.*

**Proof.** (1) If $S_{c_i}^j = \emptyset$, it is naturally an internally stable set. Otherwise, we choose $v_s, v_t \in S_{c_i}^j$. If $a_{st} = 1$, then it is easy to know from Theorem 4 or Theorem 5 that $a_{st} x_s \odot x_t = 0$, which implies that $x_s \odot x_t = 0$. Noticing that $x_s, x_t \in \Delta_k$, we obtain $x_s \neq x_t$, which means that $v_s, v_t$ belong to two different sets of the form $S_{c_i}^j$. This is a contradiction with $v_s, v_t \in S_{c_i}^j$, and thus $a_{st} = 0$. Therefore, $S_{c_i}^j$ is an internally stable set of $\mathcal{G}$.

(2) With (1), to show that $\{S_{c_1}^j, S_{c_2}^j, \ldots, S_{c_k}^j\}$ is a coloring partition, we need to prove that (a) $S_{c_i}^j \cap S_{c_s}^j = \emptyset$, $i \neq s$, and (b) $\bigcup_{i=1}^{k} S_{c_i}^j = \mathcal{V}$. In fact, if $S_{c_i}^j \cap S_{c_s}^j \neq \emptyset$ $(i \neq s)$, then there

exists at least one vertex $v_t$ such that $v_t \in S^j_{c_i}$ and $v_t \in S^j_{c_s}$ hold simultaneously. Thus, from the construction of such sets, $\delta^i_k = x_t = \delta^s_k$, which implies that $i = s$. This is a contradiction, and then (a) holds true. Now, we show that (b) holds too. Fixing $j \in K$, from the construction of $S^j_{c_i}$, $i = 1, 2, \ldots, k$, it is easy to see that $\bigcup_{i=1}^k S^j_{c_i} \subseteq \mathcal{V}$. On the other hand, for any $v_s \in \mathcal{V}$, its $k$-valued characteristic logical variable $x_s \in \Delta_k$. Since the coloring problem is solvable, we can let $x_s := \delta^i_k$, from which and the construction of $S^j_{c_i}$ we have $v_s \in S^j_{c_i} \subseteq \bigcup_{i=1}^k S^j_{c_i}$. This implies that $\mathcal{V} \subseteq \bigcup_{i=1}^k S^j_{c_i}$. Thus, (b) follows from the above analysis. □

**Remark 11.** Noticing that $\phi$ in Theorem 4 or 5 is not necessarily surjective, it is easy to see that the coloring schemes $\phi_j$, $j \in K$ obtained by Algorithm 3 contain all the colorings with the number of colors being no more than $k$. Thus, if the above coloring problem is solvable, the coloring schemes contain the minimum coloring mappings, in other words, $\{S^j_{c_1}, S^j_{c_2}, \ldots, S^j_{c_k}\}$ (all $j \in K$) contain the minimum coloring partitions.

**Remark 12.** When the above coloring problem is solvable, the minimum coloring partition can be given as follows. Let

$$N_j = \{i \mid S^j_{c_i} = \emptyset,\ 1 \leqslant i \leqslant k\}, \qquad |N_{j_0}| = \max_{j \in K}\{|N_j|\},$$

then a minimum coloring partition is given as

$$\{S^{j_0}_{c_1}, S^{j_0}_{c_2}, \ldots, S^{j_0}_{c_k}\} \setminus \{S^{j_0}_{c_i} \mid S^{j_0}_{c_i} = \emptyset\}, \tag{33}$$

and the chromatic number of $\mathcal{G}$ is

$$\gamma(\mathcal{G}) = k - |N_{j_0}|. \tag{34}$$

## 4. Application to multi-agent systems

As mentioned earlier, graph theory has played an important role in the study of multi-agent systems, a basic issue of which is the consensus problem (Liu & Guo, 2009; Olfati-Saber, 2006; Zhang & Tian, 2009). Generally speaking, the main objective in the consensus problem is to design a suitable protocol such that a group of agents in a network converge to some consistent value such as attitude, position, velocity, and so on. It is noted that such a consensus requires all the agents in a network to share a common value. However, in complex practical applications, the agreement may be different with a change of environment, situation or cooperative tasks. Based on this, a more general consensus, called the group consensus, was studied in Yu and Wang (2010). The so-called group consensus problem is to design an appropriate protocol such that agents in a network reach more than one consistent value, i.e., different sub-groups reach different consistent values.

In this section, as an application, we use the results obtained in Section 3 to investigate a kind of group consensus problem for a class of multi-agent systems, and present a new control protocol design procedure.

Consider the following multi-agent system:

$$\begin{cases} \dot{x}_i = v_i, \\ \dot{v}_i = u_i, \quad i = 1, 2, \ldots, n, \end{cases} \tag{35}$$

where $x_i \in \mathbb{R}$, $v_i \in \mathbb{R}$ and $u_i \in \mathbb{R}$ are the position, velocity and control input of agent $i$, respectively.

Assume that the information topology of system (35) is described by a directed/undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ consists of all the agents and $v_i$ stands for agent $i$, $i = 1, 2, \ldots, n$. It is noted that $\mathcal{G}$ is not necessarily a balanced graph here when $\mathcal{G}$ is directed.

Applying Algorithm 3 with $k\ (\geqslant \gamma)$ colors to the graph $\mathcal{G}$, we can obtain a minimum coloring partition of $\mathcal{V}$, denoted by

$$S_1,\ S_2,\ \ldots,\ S_\gamma, \tag{36}$$

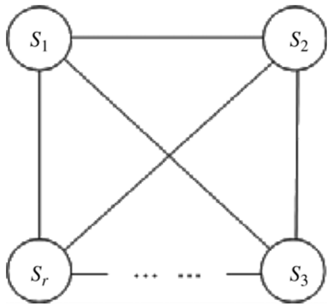



**Fig. 1.** A coloring partition of $\mathcal{G}$.

where $\gamma = \gamma(\mathcal{G})$ is the chromatic number of $\mathcal{G}$. From Proposition 1, each $S_i$ is an internally stable set, based on which the graph $\mathcal{G}$ can be equivalently changed as shown in Fig. 1.

For each agent $v_i$, define

$$\begin{aligned} N^F_i &:= S_j, \quad \text{if } v_i \in S_j, \\ N^E_i &:= \{v_j \mid (v_j, v_i) \in \mathcal{E},\ i \neq j\}, \end{aligned} \tag{37}$$

which are called the "Friends" and "Enemies" neighborhoods of agent $v_i$, respectively. Obviously, the agents belonging to the same internally stable set have the same "Friends" neighborhood, i.e., the internally stable set itself, and for agent $v_i \in S_j$, its enemies disperse in one or more $S_k$, $k \neq j$.

*The objective* of this part is to design a distributed protocol only based on the information from enemies such that all the agents are attracted to friends while trying to avoid conflict with enemies and keep the given distances from enemies. This can be described as the following *group consensus problem*. Given a required distance $d_{ij}$ to each pair $(S_i, S_j)$, $i \neq j$, design a linear control protocol

$$\begin{aligned} &u_i = u_i(x_i, x_{j_1}, \ldots, x_{j_{i_0}}), \qquad v_{j_k} \in N^E_i, \\ &\quad k = 1, 2, \ldots, i_0,\ i = 1, 2, \ldots, n \end{aligned} \tag{38}$$

such that for $\forall\, v_s \in S_i$ and $\forall\, v_k \in S_j$,

$$\begin{cases} \lim\limits_{t\to\infty} (x_k(t) - x_s(t)) = \begin{cases} d_{ij}, & i \neq j, \\ 0, & i = j, \end{cases} \\ \lim\limits_{t\to\infty} (v_k(t) - v_s(t)) = 0. \end{cases} \tag{39}$$

To facilitate the analysis, we regard $d_{ij}$ as a vector, i.e., we let $d_{ij} + d_{ji} = 0$. Then, we have the following results on the above group consensus problem.

**Theorem 6.** *Consider the multi-agent system (35) with its topology of graph $\mathcal{G}$. Assume that the required distances satisfy*

$$d_{ij} = d_{ik} + d_{kj}, \quad i, j, k \in \{1, 2, \ldots, \gamma\}, \tag{40}$$

*where $d_{ii} := 0$. Then, the above group consensus problem is solvable if and only if $\mathcal{G}$ has a directed spanning tree (or $\mathcal{G}$ is connected when $\mathcal{G}$ is an undirected graph).*

**Proof.** For each agent $v_i \in S_k$, let

$$y_i = x_i - d_{1k}, \quad i = 1, 2, \ldots, |S_k|, \quad k = 1, 2, \ldots, \gamma, \tag{41}$$

then the system (35) can be expressed as

$$\begin{cases} \dot{y}_i = v_i, \\ \dot{v}_i = u_i, \quad i = 1, 2, \ldots, n. \end{cases} \tag{42}$$

On the other hand, for $\forall\, v_s \in S_i$ and $\forall\, v_k \in S_j$, we obtain from (40) and (41) that

$$\begin{aligned} \lim_{t\to\infty} (y_k(t) - y_s(t)) &= \lim_{t\to\infty} [(x_k(t) - d_{1j}) - (x_s(t) - d_{1i})] \\ &= \lim_{t\to\infty} [(x_k(t) - x_s(t)) - (d_{1j} - d_{1i})] \\ &= \lim_{t\to\infty} [(x_k(t) - x_s(t)) - d_{ij}]. \end{aligned}$$

Thus,

$$\lim_{t\to\infty}(y_k(t)-y_s(t))=0 \Leftrightarrow \lim_{t\to\infty}[(x_k(t)-x_s(t))-d_{ij}]=0,$$

which implies that the group consensus problem is equivalent to the state consensus of the system (42). From Zhu, Tian, and Kuang (2009), the group consensus problem is solvable if and only if $\mathcal{G}$ has a directed spanning tree when $\mathcal{G}$ is directed (or $\mathcal{G}$ is connected when $\mathcal{G}$ is an undirected graph). □

**Remark 13.** The condition (40) is a natural one for the position requirement of $S_i$, $i=1,2,\ldots,\gamma$, and it also works in the case that the dimension of the agents' positions is larger than one. The results of this section can be generalized to the versions on high-order multi-agent systems.

**Theorem 7.** *Consider the multi-agent system (35) with its topology of graph $\mathcal{G}$. Assume that $\mathcal{G}$ has a directed spanning tree (or $\mathcal{G}$ is a connected undirected graph) and the required distances satisfy (40). Then, the group consensus protocol (38) can be designed as*

$$u_i=\sum_{j\in N_i^E}a(v_j-v_i)+\sum_{\substack{s=1\\ s\neq k}}^{\gamma}\sum_{j\in N_i^E\cap S_s}b \times(x_j-d_{1s}-x_i+d_{1k}),\quad \forall v_i\in S_k,\ k=1,2,\ldots,\gamma, \tag{43}$$

*where $a,b>0$ are real numbers satisfying $a^2>c_0b$, and $c_0>0$ is a sufficiently large real number.*

**Proof.** Applying (41) to each agent, the multi-agent system can be expressed as (42). On the other hand, by (41), the control (43) can be rewritten as

$$\begin{aligned}u_i&=\sum_{j\in N_i^E}a(v_j-v_i)+\sum_{\substack{s=1\\ s\neq k}}^{\gamma}\sum_{j\in N_i^E\cap S_s}b(y_j-y_i)\\&=\sum_{j\in N_i^E}[a(v_j-v_i)+b(y_j-y_i)],\\&\qquad v_i\in S_k,\ k=1,2,\ldots,\gamma.\end{aligned}\tag{44}$$

It can be seen from Zhu et al. (2009) and the conditions of the theorem that the system (42) can reach the state consensus under the control (44), with which and the proof of Theorem 6 we know that the group consensus of the system (35) is solved by the control protocol (43). Thus, the proof is completed. □

## 5. Illustrative examples

In this section, we give three examples to illustrate the effectiveness of the results/algorithms obtained in this paper.

**Example 1.** Consider the graph $G=\{\mathcal{V},\mathcal{E}\}$ shown in Fig. 2. We use Algorithm 2 to find all the absolutely maximum stable sets of the graph.

The adjacency matrix of this graph is as follows:

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

For each vertex $v_i$, we assign it a characteristic logical variable $x_i\in\mathcal{D}$ and let $y_i=[x_i,\ \bar{x}_i]^T$, $i=1,2,\ldots,6$. Since there are many '0's in $A$, we only need to calculate the structural matrices of $x_i$ and

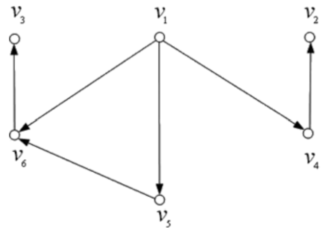



**Fig. 2.** The graph of Example 1.

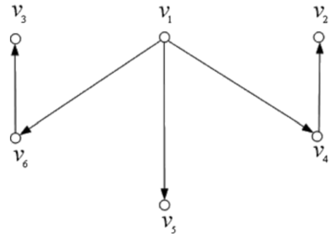



**Fig. 3.** The graph of Examples 2 and 3.

$a_{ij}x_ix_j$ with $a_{ij}=1$. By (7) and (14), and using the MATLAB toolbox provided by Cheng and his co-workers, we easily obtain

$$\begin{aligned}&[b_1\ b_2\ \cdots\ b_{64}]=J_1\widetilde{M}=J_1\\&\quad\times\left[\sum_{i=1}^{6}M_i-7(M_{24}+M_{36}+M_{41}+M_{51}+M_{61}+M_{65})\right]\\&=[-36\ -16\ -23\ -10\ -23\ -3\ -10\ 3\\&\quad -30\ -17\ -17\ -11\ -17\ -4\ -4\ 2\\&\quad -30\ -10\ -17\ -4\ -24\ -4\ -11\ 2\\&\quad -24\ -11\ -11\ -5\ -18\ -5\ -5\ 1\\&\quad -16\ -3\ -10\ -4\ -10\ 3\ -4\ 2\\&\quad -10\ -4\ -4\ -5\ -4\ 2\ 2\ 1\\&\quad -10\ 3\ -4\ 2\ -11\ 2\ -5\ 1\\&\quad -4\ 2\ 2\ 1\ -5\ 1\ 1\ 0],\end{aligned}$$

from which it is easy to see that

$$\max_{1\le i\le 64}\{b_i\}=3$$

and the corresponding column index set is

$$K=\{i_k\mid b_{i_k}=3\}=\{8,\ 38,\ 50\}.$$

For each index $i_k\in K$, let $\ltimes_{i=1}^{6}y_i=\delta_{26}^{i_k}$. By computing $y_i=S_i^6\delta_{26}^{i_k}$, $i=1,2,\ldots,6$ (see (11)), we have

$$\begin{aligned}&i_1=8\sim(x_1,x_2,x_3,x_4,x_5,x_6)=(1,\ 1,\ 1,\ 0,\ 0,\ 0),\\&i_2=38\sim(x_1,x_2,x_3,x_4,x_5,x_6)=(0,\ 1,\ 1,\ 0,\ 1,\ 0),\\&i_3=50\sim(x_1,x_2,x_3,x_4,x_5,x_6)=(0,\ 0,\ 1,\ 1,\ 1,\ 0).\end{aligned}$$

Thus, all the absolutely maximum stable sets are as follows:

$$\begin{aligned}&S(i_1)=\{v_1,v_2,v_3\},\qquad S(i_2)=\{v_2,v_3,v_5\},\\&S(i_3)=\{v_3,v_4,v_5\}.\end{aligned}$$

**Example 2.** Consider the graph $G=\{\mathcal{V},\mathcal{E}\}$ shown in Fig. 3. Letting a two-color set $N=\{c_1=\text{Red},\ c_2=\text{Blue}\}$ be given, we use Algorithm 3 to find out all the coloring schemes for $\mathcal{G}$.

For each node $v_i$, we assign it a characteristic logical variable $x_i \in \Delta$, $i = 1, 2, \ldots, 6$. The adjacency matrix of this graph is as follows:

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

By (26) and (30), and using the MATLAB toolbox provided by Cheng and his co-workers, we obtain

$$M = \sum_{i=1}^{6} \sum_{j \neq i} a_{ij} M_{ij}^{H}$$
$$= \left[\begin{matrix} 5\,3\,4\,2\,3\,1\,2\,0\,4\,3\,3\,2\,2\,1\,1\,0\,4\,2\,3\,1\,3\,1\,2 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,1\,1\,1 \end{matrix}\right.$$
$$\begin{matrix} 0\,3\,2\,2\,1\,2\,1\,1\,0\,2\,1\,2\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0\,0\,0 \\ 1\,0\,1\,0\,1\,1\,2\,1\,2\,0\,1\,1\,2\,1\,2\,2\,3\,0\,2\,1\,3\,1\,3\,2\,4 \end{matrix}$$
$$\left.\begin{matrix} 1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,1\,2\,2\,3\,3\,4\,0\,2\,1\,3\,2\,4\,3\,5 \end{matrix}\right].$$

It is observed that

$$\mathrm{col}_8(M) = \mathrm{col}_{57}(M) = 0_2,$$

and the corresponding column index set is

$$K = \{8,\ 57\}.$$

For each index $j \in K$, let $\ltimes_{i=1}^{6} x_i = \delta_{26}^{j}$. By computing $x_i = S_{i,2}^{6}\,\delta_{26}^{j}$, $i = 1, 2, \ldots, 6$ (see (29)), we have

$$\delta_{26}^{8} \sim [x_1, x_2, x_3, x_4, x_5, x_6] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$
$$\delta_{26}^{57} \sim [x_1, x_2, x_3, x_4, x_5, x_6] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

from which we obtain the following two coloring schemes:

$$S_{c_1}^{8} = \{v_1,\ v_2,\ v_3\}\ (\text{Red}), \qquad S_{c_2}^{8} = \{v_4,\ v_5,\ v_6\}\ (\text{Blue})$$

and

$$S_{c_1}^{57} = \{v_4,\ v_5,\ v_6\}\ (\text{Red}), \qquad S_{c_2}^{57} = \{v_1,\ v_2,\ v_3\}\ (\text{Blue}).$$

**Example 3.** Consider the following multi-agent system:

$$\begin{cases} \dot{x}_i = v_i, & \\ \dot{v}_i = u_i, & i = 1, 2, \ldots, 6, \end{cases} \tag{45}$$

where $x_i \in \mathbb{R}$, $v_i \in \mathbb{R}$ and $u_i \in \mathbb{R}$ are the position, velocity and control input of agent $i$, respectively.

Assume that the information topology of the system is given by the graph $\mathcal{G}$ shown in Fig. 3, where $v_i$ stands for agent $i$, $i = 1, 2, \ldots, 6$. From Example 2, we know that the minimum coloring partition of $\mathcal{G}$ is as follows: $S_1 = \{v_1, v_2, v_3\}$, $S_2 = \{v_4, v_5, v_6\}$. Given a required distance $d = 2$ to $(S_1, S_2)$, for each agent $v_i$ we design a control $u_i$ such that the whole system reaches group consensus, that is, for all $i, j = 1, 2, \ldots, 6$,

$$\lim_{t\to\infty} (x_j(t) - x_i(t)) = \begin{cases} d, & v_i \in S_1 \quad \text{and} \quad v_j \in S_2 \\ 0, & \text{both } v_i, v_j \in S_1 \quad \text{or} \quad S_2, \end{cases}$$
$$\lim_{t\to\infty} (v_j(t) - v_i(t)) = 0.$$

It is easy to check that all the conditions of Theorem 6 are satisfied, and thus the group consensus problem is solvable. By

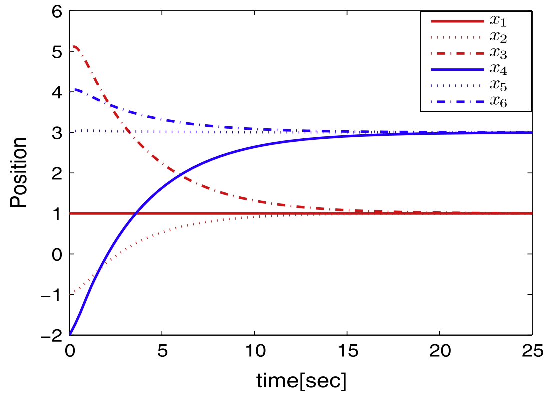



**Fig. 4.** Responses of the positions.

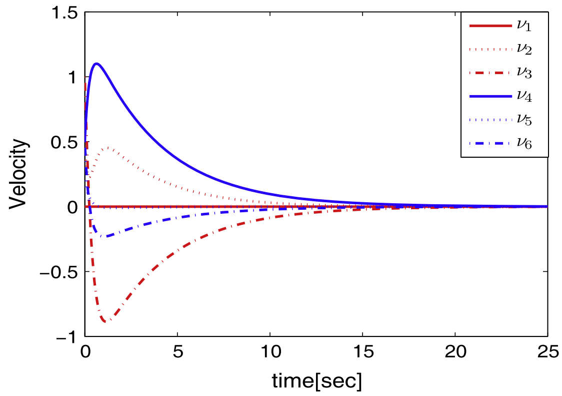



**Fig. 5.** Responses of the velocities.

Theorem 7, the desired control protocol is designed as

$$\begin{cases} u_1 = 0, \\ u_2 = a(v_4 - v_2) + b(x_4 - x_2 - d), \\ u_3 = a(v_6 - v_3) + b(x_6 - x_3 - d), \\ u_4 = a(v_1 - v_4) + b(x_1 - x_4 + d), \\ u_5 = a(v_1 - v_5) + b(x_1 - x_5 + d), \\ u_6 = a(v_1 - v_6) + b(x_1 - x_6 + d), \end{cases} \tag{46}$$

where $a, b > 0$ are real numbers, and $a$ is sufficiently large.

To show the effectiveness of the control (46), we carry out some simulation results with the following choices. Initial condition: $[x_1(0),\ x_2(0),\ x_3(0),\ x_4(0),\ x_5(0),\ x_6(0)] = [1,\ -1,\ 5,\ -2,\ 3,\ 4]$ and $[v_1(0),\ v_2(0),\ v_3(0),\ v_4(0),\ v_5(0),\ v_6(0)] = [0,\ 0.5,\ 1,\ 0.5,\ 0.2,\ 0.5]$; parameters: $a = 4$ and $b = 1$. The simulation results are shown in Figs. 4 and 5, which are the responses of the agents' positions and velocities, respectively.

It is observed from Figs. 4 and 5 that the group consensus with the given distance between the two groups is reached under the protocol (46). Simulation shows that the control design method given in this paper is very effective.

## 6. Conclusion

In this paper, we have studied the maximum (weight) stable set and vertex coloring problems with application to the group consensus of multi-agent systems, and presented a number of new results and algorithms by the matrix semi-tensor product method. By defining a characteristic logical vector and using the matrix expression of logical functions, an algebraic description has been obtained for the stable set problem, and a new algorithm is established to find all the internally stable sets. Based on the matrix expression, a necessary and sufficient condition is also obtained for the maximum (weight) stable set problem, and an algorithm to find all the maximum (weight) stable sets is designed. Moreover, by the semi-tensor product method, two necessary and sufficient conditions are proposed for the vertex coloring problem, based on which a new algorithm to find all the $k$-coloring schemes and minimum coloring partitions is established. In addition, the obtained results have been applied to multi-agent systems, and a new protocol design procedure has been obtained for the group consensus problem. The study of illustrative examples has shown that the results/algorithms presented in this paper are very effective.

## Acknowledgments

The authors would like to express their sincere appreciation to the anonymous reviewers for their constructive comments and suggestions on improving the quality and presentation of the paper.

## References


Barnier, N., & Brisset, P. (2004). Graph coloring for air traffic flow management. *Annals of Operations Research*, *130*(1-4), 163–178.

Blöchliger, I., & Zufferey, N. (2008). A graph coloring heuristic using partial solutions and reactive tabu scheme. *Computers and Operations Research*, *35*(3), 960–975.

Carter, M., Laporte, G., & Lee, S. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Resaerch Society*, *47*(3), 373–383.

Chaitin, G. J., Auslander, M. A., Chandra, A. K., et al. (1981). Register allocation via coloring. *Computer Languages*, *6*(1), 47–57.

Cheng, D. (2010). Disturbance decoupling of Boolean control networks. *IEEE Transactions Automatic Control*, *20*(3), 561–582.

Cheng, D., Li, Z., & Qi, H. (2010). Realization of Boolean control networks. *Automatica*, *46*(1), 62–69.

Cheng, D., & Qi, H. (2007). *Semi-Tensor product of matrices– theory and applications*. Beijing: Science Press (in Chinese).

Cheng, D., & Qi, H. (2010). A linear representation of dynamics of Boolean networks. *IEEE Transactions on Automatic Control*, *55*(10), 2251–2258.

Cheng, D., & Qi, H. (2009). Controllability and observability of Boolean control networks. *Automatica*, *45*(7), 1659–1667.

Cheng, D., Qi, H., & Li, Z. (2011). *Analysis and control of boolean networks: a semi-tensor product approach*. London: Springer.

Cheng, D., Qi, H., Li, Z., & Liu, J. B. (2011). Stability and stabilization of Boolean networks. *International Journal of Robust and Nonlinear Control*, *21*(2), 134–156.

Fax, J. A., & Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions Automatic Control*, *49*(9), 1465–1476.

Galinier, P., Hertz, A., & Zufferey, N. (2008). An adaptive menmory algorithm for the $k$-coloring problem. *Discrete Applied Mathematics*, *156*(2), 267–279.

Gamst, A. (1986). Some lower bounds for a class of frequency assignment problems. *IEEE Transactions of Vehicular Technology*, *35*(1), 8–14.

Garey, M.R., & Johnson, D.S. (1979).*Computers and intractability: a guide to the theory of NP-completeness* Eds. Freeman W.H. San Francisco.

Glass, C. A., & Pruegel-Bennett, A. (2005). A polynomially searchable exponential neighbourhood for graph coloring. *Journal of the Operational Research Society*, *56*(3), 324–330.

Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithm for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, *42*(6), 1115–1145.

Hertz, A., Plumettaz, A., & Zufferey, N. (2008). Variable space search for graph coloring. *Discrete Applied Mathematics*, *156*(13), 2551–2560.

Laschov, D., & Margaliot, M. (2011). A maximum principle for single-input Boolean control networks. *IEEE Transactions Automatic Control*, *56*(4), 913–917.

Lee, D., & Spong, M. W. (2007). Stable flocking of multiple inertial agents on balanced graphs. *IEEE Transactions Automatic Control*, *52*(8), 1469–1475.

Lin, Z., Francis, B., & Maggiore, M. (2005). Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions Automatic Control*, *50*(1), 121–127.

Lin, P., & Jia, Y. M. (2009). Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically chaning topologies. *Automatica*, *45*(9), 1425–1434.

Li, F., & Sun, J. (2011). Controllability of Boolean control networks with time delays in states. *Automatica*, *47*(3), 603–607.

Liu, Z. X., & Guo, L. (2009). Synchronization of multi-agent systems without connectivity assumptions. *Automatica*, *45*(12), 2744–2753.

Liu, Y. C., & Zhang, W. (1993). *Boolean methodology*. Shanghai: Shanghai Technology Literature Press, (in Chinese).

Li, T., & Zhang, J. F. (2009). Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions. *Automatica*, *45*(8), 1929–1936.

Malaguti, E., Monaci, M., & Toth, P. (2008). A metaheuristic approach for the vertex coloring problem. *Informs Journal on Computing*, *20*(2), 302–316.

Minty, G. J. (1980). On maximal independent sets of vertices in claw-free graphs. *Journal of Combination Theory Series B*, *28*, 284–304.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions Automatic Control*, *51*(3), 401–420.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions Automatic Control*, *49*(9), 1520–1533.

Pardalos, P., & Xue, J. (1994). The maximum clique problem. *Journal of Global Optimization*, *4*(3), 301–328.

Poljak, S. (1974). A note on stable sets and colorings of graphs. *Communication in Mathematics University Carolinae*, *15*(2), 307–309.

Su, H. S., Wang, X. F., & Lin, Z. L. (2009). Flocking of multi-agents with a virtual leader. *IEEE Transactions Automatic Control*, *54*(2), 293–307.

Torkestani, J. A., & Meybodi, M. R. (2011). A celluar learning automata-based algorithm for solving the vertex coloring problem. *Expert Systems with Applications*, *38*(8), 9237–9247.

Yánez, J., & Ramírez, J. (2003). The robust coloring problem. *European Journal of Operational Research*, *148*(3), 546–558.

Yu, J., & Wang, L. (2010). Group consensus in multi-agent systems with switching topologies and communication delays. *Systems and Control Letters*, *59*(6), 340–348.

Zhang, Y., & Tian, Y. P. (2009). Consentability and protocol design of multi-agent systems with stochastic switching topology. *Automatica*, *45*(5), 1195–1201.

Zhu, J., Tian, Y., & Kuang, J. (2009). On the general consensus protocol of multi-agent systems with double-integrator dynamics. *Linear Algebra and Its Applications*, *431*(5–7), 701–715.


**Yuzhen Wang** graduated from Tai'an Teachers College in 1986, received his M.S. degree from Shandong University of Science & Technology in 1995 and his Ph.D. degree from the Institute of Systems Science, Chinese Academy of Sciences in 2001. From 2001 to 2003, he worked as a Postdoctoral Fellow in Tsinghua University, Beijing, China. Since 2003, he has been a Professor with the School of Control Science and Engineering, Shandong University, China, and the director of the Institute of Automatic Control, Shandong University. From March 2004 to June 2004, from February 2006 to May 2006 and from November 2008 to January 2009 he visited City University of Hong Kong as a Research Fellow. From September 2004 to May 2005 he worked as a visiting Research Fellow at the National University of Singapore.

His research interests include nonlinear control systems, Hamiltonian systems, Boolean networks, and multi-agent systems. Prof. Wang received the Prize of Guan Zhaozhi in 2002, the Prize of Huawei from the Chinese Academy of Sciences in 2001, the Prize of Natural Science from the Chinese Education Ministry in 2005, and the National Prize of Natural Science of China in 2008. Currently, he is a Taishan Scholar of Sahndong Province, China, an Associate Editor of the Asian Journal of Control, and the IMA Journal of Mathematical Control and Information.

**Chenghui Zhang** received his B.S. and M.S. degrees from Shandong University of Technology, China, in 1985 and 1988, respectively, and his Ph.D. degree from Shandong University, China, in 2001. In 1988, he joined Shandong University, where he is currently a Full Professor with the School of Control Science and Engineering, and the director of Research Center of Power Electronics Energy-Saving Technology & Equipment of the Chinese Education Ministry. Prof. Zhang was selected as a Changjiang Scholar of the Education Ministry, and a Taishan Scholar of Shandong Province in 2009, respectively. His interests include optimal control of engineering, power electronics and motor drives, and energy-saving techniques.

**Zhenbin Liu** received his B.S. and M.S. degrees from the School of Mathematical Science, Qufu Normal University, China, in 2002 and 2006, respectively. Since 2009, he has been with Qingdao Agriculture University, China. He is currently pursuing a Ph.D. study at the School of Control Science and Engineering, Shandong University, China. His interests include logical systems, systems biology, semi-tensor product, etc.