第 34 卷第 4 期
2017 年 4 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 34 No. 4
Apr. 2017

DOI: 10.7641/CTA.2017.16033

k轨道任务分配问题的可解性条件: 图论方法

岳菊梅[1], 陈增强[2], 闫永义[3†], 金 鑫[1]

(1. 河南科技大学 农业装备工程学院, 河南 洛阳 471003;

2. 南开大学 计算机与控制工程学院, 天津 300071; 3. 河南科技大学 信息工程学院, 河南 洛阳 471003)

摘要: 将图论及一种新的数学分析工具——矩阵的半张量积(semi-tensor product of matrices, STP), 作为研究工具, 通过研究图的k内稳定集的充分必要条件, 研究了k轨道任务分配问题的可解性条件. 定义了图的顶点子集的特征向量, 利用STP方法得到图的k内稳定集新的若干充分必要条件. 基于这些新的充分必要条件, 建立了能够搜索出图的所有k内稳定集的两种算法. 进而将上述结果应用到k轨道任务分配问题, 得到了该问题可解性的两个充分必要条件. 此外, 通过这些充分必要条件, 也发现了一些有趣的现象. 例如, 完全最优方案(completely optimal schedules)的存在.

关键词: k轨道任务分配; k内稳定集; 可解性; 图论方法; 矩阵的半张量积

中图分类号: TP273 **文献标识码**: A

Solvability of k-track assignment problem: a graph approach

YUE Ju-mei[1], CHEN Zeng-qiang[2], YAN Yong-yi[3†], JIN Xin[1]

(1. College of Agricultural Engineering, Henan University of Science and Technology, Luoyang Henan 471003, China;
2. College of Computer and Control Engineering, Nankai University, Tianjin 300071, China;
3. College of Information Engineering, Henan University of Science and Technology, Luoyang Henan 471003, China)

Abstract: The theory of graph and a new mathematical analysis tool, semi-tensor product (STP) of matrices are applied to consider the solvability conditions of k-track assignment problem by investigating the necessary and sufficient conditions of k-internally stable sets of graphs (k-ISS). By defining characteristic vectors for vertex subsets of graphs and using the STP, several new necessary and sufficient conditions of k-ISS are obtained, based on which two algorithms able to find all the k-ISSs of a graph are established. The results obtained are further applied to the k-track assignment problem, and two necessary and sufficient conditions of the solvability of the problem are proposed; also some interesting phenomena such as completely optimal schedules are discovered by the new method.

Key words: k-track assignment; k-internally stable set; solvability; graph method; semi-tensor product of matrices

1 Introduction

The multi-track assignment problem is a key issue in operational research and control theory, and has been applied to many areas such as production scheduling and control, discrete event dynamic systems and optimal design of engineering. The problem has been investigated extensively in recent years. Cornelsen and Stefano[1] considered how to assign tracks of a station to some trains such that they can leave and enter without conflicting with any other one. Later, the problem of track assignment of a station was further examined by Demange, et al.[2]. The described this problem as an online coloring of circle graphs or permutation graphs under the assumption that a station is composed of serval parallel tracks and each track is approachable from one or both sides for one or multiple trains. As regard the optimization of track assignment, Severson and Paley[3] proposed a method to optimize the performance of assignment in the circumstance of multiple shipboard radar systems by maximizing the collective search area of the radars. As a continuation, Severson and Paley[4] discussed the optimization problem in the context of ballistic missile surveillance and tracking.

However, to the best of our knowledge, there has been no result on how to find out all the feasible schedules and all the optimal schedules for a given multi-track assignment problem. Finding all the feasible schedules and all the optimal schedules is useful to understand the inherent logical relationship of multi-track assignment problem. Hence it is meaningful to discuss complete solution sets of the problem, and of course it is a challenging work.

Received 23 February 2016; accepted 27 December 2016.
†Corresponding author. E-mail: yyyan@mail.nankai.edu.cn; Tel.: +86 13721603289.
Recommended by Associate Editor HU Xiao-ming.
Supported by Key Scientific Research Program of the Higher Education Institutions of Henan Educational Committee (15A416005), 2015 Science Foundation of Henan University of Science and Technology for Youths (2015QN016), National Natural Science Foundation of China (61573199) and Sub-project of National Key Research and Development Program (2016YFD0700103–2).

Graphs provide mathematical models to analyse successfully many concrete real-world problems. Regarding the problem of multi-track assignment, it can be modeled as a k–internally stable set of a graph (k–ISS). A k–internally stable set is an extension of internally stable set that there is no a path of length equal to or less than k between any two vertices of the set, where an internally stable set is a set of vertices in which no two vertices are adjacent, in other words, there is no path of length 1 between any two vertices of the set.

Unfortunately, little attention has been paid on how to find out all k-internally stable sets of graphs. Encouragingly, the theory of semi-tensor product (STP), which was proposed by D. Cheng and H. Qi[5] in recent years, provides a promising approach to model and analyse the structure of graphs. Y. Wang[6] is the first who introduces the STP to graph theory. Using the STP he investigated the problems of the maximum weight stable set and vertex colouring, and presented several new results and algorithms, which can be used to study some problems of multiple agent systems such as the group consensus. Later, M. Meng and J. Feng[7] applied the STP to the field of hypergraph. Using the STP, they studied the problems of stable set and colouring of hypergraphs and obtained some new results and algorithms that can find all the colouring schemes and minimum colouring partitions. Besides, STP has been applied successfully to many fields such as Boolean networks[8–11], game theory[12–13], nonlinear systems[14], fuzzy control systems[15] and finite automata[16–19]. Especially in the field of graph theory, as Y. Wang[6] said, the STP method can express graph problems in a clear way and is helpful for further study of graph problems.

Motivated by the above, one of the aims of our work is to investigate the k-internally stable sets of graphs and establish their new necessary and sufficient conditions and search algorithm by using STP as a main research tool. Another aim is to apply the results obtained to the k–track assignment problem, and further to establish an algorithm to find out all the feasible schedules and optimal schedules for a given multi-track assignment problem. Our main contributions are as follows. We provide a new mathematical formulation for k–internally stable sets of graphs, and obtain several new theoretical results and algorithms. These new results are further applied to consider the k–track assignment problem and two necessary and sufficient conditions of the solvability of the problem are proposed; also some interesting phenomena are discovered by the new method, which differs greatly from the existing results. It is worth noting that our focus of this paper is on the theoretical results and solution algorithms. How to reduce the computational complexity is our next work.

The paper contains the following contents. Section 2 gives some preliminaries on the STP and k–internally stable sets of graphs. Section 3 devotes to discuss how to search k–internally stable sets of graphs by the STP, and present serval new results on the issue. The solvability conditions of the k–track assignment problem is presented in Section 4. Section 5 tests the correctness of the results by an illustrative example; this is followed by some concluding remarks in Section 6.

2 Preliminaries

This section gives some necessary preliminaries on the STP, k–ISS, k–MISS and k–AMISS of graphs.

Definition 1[5] For $M \in \mathcal{M}_{m\times n}$ and $N \in \mathcal{M}_{p\times q}$, their semi-tensor product, denoted by $M \ltimes N$, is defined as follows: $M \ltimes N := (M \otimes I_{s/n})(N \otimes I_{s/p})$, where s is the least common multiple of n and p, and $\otimes$ is the Kronecker product.

Remark 1 STP is a generalization of the conventional matrix product, when $n = p$, it reduces to the latter. Not only do almost all the major properties of the conventional matrix product remain true for STP, for instance, the associative law is that for $A \in \mathcal{M}_{m\times n}$, $B \in \mathcal{M}_{p\times q}$, and $C \in \mathcal{M}_{r\times s}$, we have $(A \ltimes B) \ltimes C = A \ltimes (B \ltimes C)$, but STP can overcome some defects of the conventional matrix product; the following is an interesting example. For a detailed description, please refer to [5].

Definition 2[5] A swap matrix $W_{[m,n]}$ is an $mn \times mn$ matrix, which is defined as follows. Its rows and columns are labelled by double index (i, j), the rows are arranged by the ordered multi-index $\mathrm{Id}[(I, J), (i, j)]$, and the columns are arranged by the ordered multi-index $\mathrm{Id}[(I, J), (i, j)]$. 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 for any $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$, we have[5]

$$\begin{cases} W_{[m,n]} \ltimes X \ltimes Y = Y \ltimes X, \\ W_{[n,m]} \ltimes Y \ltimes X = XY. \end{cases} \tag{1}$$

This can be seen as the “quasi-commutative law” of the STP, which the conventional matrix product does not hold.

Let “1” and “0” represent the logical “True” and “False”, respectively, and $\mathcal{D} := \{0, 1\}$. In many cases, we use the following vectors to represent them. $\mathcal{T} := 1 \sim \delta_2^1$, $\mathcal{F} := 0 \sim \delta_2^2$, where δ_n^i is the ith column of the identity matrix I_n, and “$\sim$” denotes “identity”. Similarly, a k–valued logical variable $x \in \mathcal{D}_k$,

$$\mathcal{D}_k := \{0, \frac{1}{k-1}, \frac{2}{k-1}, \cdots, 1\},$$

can be represented with the vectors:

$$\frac{k-i}{k-1} \sim \delta_k^i, \ i = 1, 2, \cdots, k.$$

The following are some notations used in this paper:

$\varDelta_n = \{\delta_n^1, \cdots, \delta_n^n\}$. Especially $\varDelta_2 = \{\delta_2^1, \delta_2^2\}$ is the vector form of the logical range $\mathcal{D} := \{0, 1\}$.

$\boldsymbol{k}_n \in \mathbb{R}^n$ is a column vector with each element being real number k.

$\mathrm{col}_i(A)$ is the ith column of matrix A, $\mathrm{col}(A)$ is the set of the columns of A.

Definition 3[20] The Boolean product of matrices $A = [a_{ij}]_{p\times q}$ and $B = [b_{ij}]_{q\times r}$, denoted as $C = A\times_{\mathcal{B}} B = [c_{ij}]_{p\times r}$, is defined as follows: $c_{ij} = \overset{q}{\underset{l=1}{\vee}} (a_{il} \wedge b_{lj})$, where $\vee$ and $\wedge$ are the logic addition and multiplication on $\mathcal{D}$, respectively. Further the Boolean power of matrix A is defined as $A^{0_{\mathcal{B}}} = A, A^{k_{\mathcal{B}}} = A^{k-1_{\mathcal{B}}}\times_{\mathcal{B}} A, k = 1, 2, \cdots$.

A graph G consists of a finite nonempty set V of objects called vertices and a set E of 2-element subsets of V called edges. The sets V and E are the vertex set and edge set of G, respectively. So a graph G is an ordered pair of two sets V and E, denoted by $G = (V, E)$. For a graph $G = (V, E)$ with $V = \{v_1, v_2, \cdots, v_n\}$, if $e_{ij} = (v_i, v_j) \in E$ implies $e_{ji} = (v_j, v_i) \in E$, G is called an undirected graph; otherwise,G is called a directed graph. A graph $G' = (V', E')$ is called a sub-graph of G if $V' \subseteq V$ and $E' \subseteq E$. A path of a graph $G=(V, E)$ is a sub-graph $P = (V', E')$ of the form $V'=\{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$, $E'=\{(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \cdots, (v_{i_{k-1}}, v_{i_k})\}$. The number of the edges of a path from v_i to v_j is its length, denoted as $d(v_i, v_j)$. For $i = j$, we define $d(v_i, v_i) = 0$.

Definition 4[21] A set S of vertices of graph G is a k–internally stable set (k–ISS) of G if for every pair vertices v_i and v_j of S there is no path with $d(v_i, v_j) \leqslant k$ between them. A k–internally stable set S is said to be maximum (k–MISS) if any vertex subset strictly containing S is not a k–internally stable set. A k–internally stable set with the largest number of vertices is called a k–absolute maximum internally stable set (k–AMISS).

3 Algebraic approach to search k–ISSs and k–AMISSs

In this section we investigate how to search k–ISSs and k–AMISSs of graphs in a mathematical manner and present the main results of this paper.

3.1 Searching k–internally stable sets

Consider a graph $G= (V, E)$ with $V = \{v_1, v_2, \cdots, v_n\}$. The adjacency matrix $A = [a_{ij}]$ of G is defined by

$$a_{ij} = \begin{cases} 1, & (v_i, v_j) \in E, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

The k–adjacency matrix $A^{[k]}$ of G is defined as

$$A^{[k]} = A \vee A^{2_{\mathcal{B}}} \vee \cdots \vee A^{k_{\mathcal{B}}}, \tag{3}$$

where $A^{i_{\mathcal{B}}} (i = 1, 2, \cdots, k)$ are the Boolean power of matrix A, and $\vee$ is defined as follows. For $A = [a_{ij}]_{m\times n}$ and $B = [b_{ij}]_{m\times n}$, then $A \vee B = [a_{ij} \vee b_{ij}]_{m\times n}$.

For a given subset $S \subseteq V$, we define $V_S = [x_1, x_2, \cdots, x_n]$ as its characteristic vector, where

$$x_i = \begin{cases} 1, & \text{if } x_i \in S, \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

Further we define

$$Y_S = \ltimes_{i=1}^n y_i, \tag{5}$$

where $y_i = [x_i, \bar{x}_i]^{\mathrm{T}},\ \bar{x}_i = 1 - x_i,\ i = 1, 2, \cdots, n$.

Remark 3 According to [5], every y_i in Eq.(5) can be derived from Y_S. y_i is defined by x_i that is determined uniquely by the subset S. Thus Y_S is in one-to-one correspondence with V_S, and therefore Y_S can also be called the characteristic vector of S. In this paper Y_S and V_S can be used interchangeably without arising confusion. One can get a subset S if a characteristic vector Y_S is known.

We first introduce the following definition.

Definition 5 We call $E_{\text{d\_former}}(m, n)$ and $E_{\text{d\_latter}}(m, n)$ dummy operators, where

$$E_{\text{d\_former}}(m, n) = [\underbrace{I_m, \cdots, I_m}_{n}],$$
$$E_{\text{d\_latter}}(m, n) = [\underbrace{I_n, \cdots, I_n}_{m}].$$

The reason we call them dummy operators is that for m- and n-valued logical variables, $u \in \varDelta_m$ and $v \in \varDelta_n$, we have

$$\begin{cases} E_{\text{d\_former}}(m, n) \ltimes u \ltimes v = v, \\ E_{\text{d\_latter}}(m, n) \ltimes W_{[n,m]} \ltimes u \ltimes v = u. \end{cases} \tag{6}$$

For notation concision, when $m = n = 2$, denote $E_{\text{d\_former}}(2, 2)$ and $E_{\text{d\_latter}}(2, 2)$ as E_{d} and E'_{d}, respectively.

Theorem 1 Let $A^{[k]} = [a_{ij}^{[k]}]$ be the k–adjacency matrix of graph $G = (V, E)$ with $V = \{v_1, v_2, \cdots, v_n\}$. G contains a k–internally stable set if and only if there is a $j, 1 \leqslant j \leqslant 2^n$, such that

$$\mathrm{col}_j(M) = \mathbf{0}_n, \tag{7}$$

where

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix},\ M_i = Q \sum_{j=1}^n a_{ij}^{[k]} T_{ij},\ i = 1, 2, \cdots, n,$$

$$T_{ij} = (E_{\text{d}})^{n-2} \ltimes W_{[2^j, 2^{n-j}]} \ltimes W_{[2^i, 2^{j-i-1}]},$$
$$Q = [1\ 0\ 0\ 0].$$

Proof We first prove that there is a path with $d(v_i, v_j) \leqslant k$ from vertex v_i to v_j if and only if $a_{ij}^{[k]} = 1$. Consider $A^{l_{\mathcal{B}}} = [a_{ij}^l], 1 \leqslant l \leqslant k$. The definition of Boolean product of matrices implies that

$$a_{ij}^l = \underset{i_1, \cdots, i_l}{\vee} (a_{ii_1} \wedge a_{i_1 i_2} \wedge \cdots \wedge a_{i_{l-1} j}). \tag{8}$$

Observing Eq.(8), it is easy to see that $a_{ij}^l = 1$ iff there exist $l-1$ subscripts $i_1, i_2, \cdots, i_{l-1}$ such that

$$a_{ii_1} = a_{i_1i_2} = \cdots = a_{i_{l-1}j} = 1.$$

This together with Eq.(2) tell us that there exists an edge between vertices v_i and v_{i_1}, v_{i_1} and $v_{i_2}, \cdots, v_{i_{l-1}}$ and v_j, respectively. Thus $a_{ij}^l = 1$ is equivalent to that vertices v_i and v_j are connected by a path with $d(v_i, v_j) \leqslant l$; if the subscripts $i, i_1, i_2, \cdots, i_{l-1}, j$ are distinct from each other, the length of the path is l.

Now we consider $A^{[k]} = [a_{ij}^{[k]}]$. $a_{ij}^{[k]} = a_{ij}^1 \vee a_{ij}^2 \vee \cdots \vee a_{ij}^k = 1$ if and only if there is an $l, 1 \leqslant l \leqslant k$, such that $a_{ij}^l = 1$, i.e., there is a path with $d(v_i, v_j) \leqslant l$ from vertex v_i to v_j, therefore, $a_{ij}^{[k]} = 1$ if and only if there is a path with $d(v_i, v_j) \leqslant k$ from vertex v_i to v_j.

Next we prove Theorem 1. (Necessity). If G contains a k–internally stable set S with characteristic vector $V_S = [x_1, x_2, \cdots, x_n]$, it is easy to know from Definition 4 that for any two vertices $v_i, v_j \in V$, if $a_{ij}^{[k]} = 1$, that is, there is a path with $d(v_i, v_j) \leqslant k$ between v_i and v_j, then either $v_i \notin S$ or $v_j \notin S$, by which and Eq.(4) we have $x_ix_j = 0$. Thus the characteristic vector $V_S = [x_1, x_2, \cdots, x_n]$ of S satisfies the following equations

$$\sum_{j=1}^{n} a_{ij}^{[k]} x_i x_j = 0,\ i = 1, 2, \cdots, n. \tag{9}$$

Since $x_ix_j = x_jx_i$, without loss of generality, we assume $i < j$ in the following. Using Eqs. (1) and (6), for $y_i = [x_i, 1 - x_i]^{\mathrm{T}}$ and $y_j = [x_j, 1 - x_j]^{\mathrm{T}}$, we can get the following equation:

$$\begin{aligned}
&y_iy_j = \\
&(E_{\mathrm{d}})^{n-2} y_{j+1} \cdots y_n y_{i+1} \cdots y_{j-1} y_1 \cdots y_{i-1} y_i y_j = \\
&(E_{\mathrm{d}})^{n-2} W_{[2^j, 2^{n-j}]} y_{i+1} \cdots y_{j-1} y_1 \cdots y_n = \\
&(E_{\mathrm{d}})^{n-2} W_{[2^j, 2^{n-j}]} W_{[2^i, 2^{j-i-1}]} y_1 \cdots y_i y_{i+1} \cdots y_n = \\
&T_{ij} Y_S,
\end{aligned} \tag{10}$$

where $T_{ij} = (E_{\mathrm{d}})^{n-2} \ltimes W_{[2^j, 2^{n-j}]} \ltimes W_{[2^i, 2^{j-i-1}]}$, $Y_S = \ltimes_{i=1}^{n} y_i$.

Because $x_ix_j = Q(y_i \ltimes y_j)$, we then have

$$x_ix_j = Q(T_{ij} \ltimes Y_S), \tag{11}$$

where $Q = [1\ \ 0\ \ 0\ \ 0]$.

Equation (9) can be therefore rewritten as

$$\begin{aligned}
&\sum_{j=1}^{n} a_{ij}^{[k]} Q(T_{ij} \ltimes Y_S) = Q(\sum_{j=1}^{n} a_{ij}^{[k]} T_{ij}) \ltimes Y_S = \\
&M_i \ltimes Y_S = 0,\ i = 1, 2, \cdots, n,
\end{aligned} \tag{12}$$

where $M_i = Q(\sum_{j=1}^{n} a_{ij}^{[k]} T_{ij})$.

Note that Eq.(12) are equivalent to

$$\begin{cases} M_1 \ltimes Y_S = 0, \\ M_2 \ltimes Y_S = 0, \\ \qquad \vdots \\ M_n \ltimes Y_S = 0. \end{cases}$$

That is,

$$M \ltimes Y_S = \mathbf{0}_n, \tag{13}$$

where $M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix}$.

Now we can get that if G contains a k–internally stable set S with characteristic vector $V_S = [x_1, x_2, \cdots, x_n]$, then Eq.(9) is solvable, equivalently, Eq.(13) holds, which implies that there exists a column of M that is $\mathbf{0}_n$. The necessity is obtained.

(Sufficiency). If there is a j, $1 \leqslant j \leqslant 2^n$, satisfying $\mathrm{col}_j(M) = \mathbf{0}_n$, then the vector $Y_S = \delta_{2^n}^j$ satisfies Eq.(13). Hence, Eq.(9) has a solution $(x_1, x_2, \cdots, x_n)$, which corresponds to a characteristic vector of a subset of V, say, S. Since $x_i \in \mathcal{D}$ and $a_{ij}^{[k]} \geqslant 0$, we have $a_{ij}^{[k]} x_i x_j \geqslant 0$. This together with Eq.(9) indicates that $a_{ij}^{[k]} x_i x_j = 0$ holds for any $i \neq j$, which tell us that if $a_{ij}^{[k]} = 1$ then either $x_i = 0$ or $x_j = 0$, i.e., either $v_i \notin S$ or $v_j \notin S$. By the definition of k–internal stable set, S is a k–internal stable set of G. We then get the sufficiency. The proof is completed.

To construct, based on Theorem 1, an algorithm to search all k–internally stable sets of graphs, we express Theorem 1 in another form as follows:

Theorem 2 Consider a graph $G = (V, E)$ with the k–adjacency matrix $A^{[k]} = [a_{ij}^{[k]}]$. For a given subset $S \subseteq V$, let its characteristic vector be $V_S = [x_1, x_2, \cdots, x_n]$, and let $Y_S = \ltimes_{i=1}^{n} y_i = \delta_{2^n}^k$, $y_i = [x_i, \overline{x}_i]^{\mathrm{T}}$. Then S is a k–internally stable set of G iff

$$\mathrm{col}_k(M) = \mathbf{0}_n, \tag{14}$$

where

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix},\ M_i = Q \sum_{j=1}^{n} a_{ij}^{[k]} T_{ij},\ i = 1, 2, \cdots, n,$$

$$T_{ij} = (E_{\mathrm{d}})^{n-2} \ltimes W_{[2^j, 2^{n-j}]} \ltimes W_{[2^i, 2^{j-i-1}]},$$

$$Q = [1\ \ 0\ \ 0\ \ 0].$$

Proof (Necessity). If S is a k–internal stable set of G, from the proof of the necessity of Theorem 1, we know that the characteristic vector $Y_S = \delta_{2^n}^k$ of S satisfies Eq.(13), i.e., $M \ltimes \delta_{2^n}^k = \mathbf{0}_n$. Note that the dimension of M is $n \times 2^n$ and Y_S is of $2^n \times 1$ dimension. In this case the semi-tensor product of matrices reduces to the conventional product of matrices. Thus $M \ltimes \delta_{2^n}^k$ is just the kth column of M. The necessity is proved.

(Sufficiency). If $\mathrm{col}_k(M) = \mathbf{0}_n$, then the vector $Y_S = \delta_{2^n}^k$ satisfies Eq.(13). Recall the proof of the sufficiency of Theorem 1, and we know S is a k–internal stable set of G. The proof is then completed.

Based on the proofs of Theorems 1 and 2, the follow conclusion is obvious.

Corollary 1 For a given graph $G=(V,E)$, assume its k–adjacency matrix is $A^{[k]}=[a_{ij}^{[k]}]$. Assign each vertex $v_i\in V$ a characteristic variable x_i as described in Eq.(4) and define $y_i=[x_i,\overline{x}_i]^{\mathrm{T}}$. Then G contains a k–internally stable set if and only if the following equation

$$M\ltimes_{i=1}^{n}y_i=\mathbf{0}_n \tag{15}$$

has at least one solution, where M is given in Eq.(14). Moreover the number of k–internally stable set equals to the number of the solutions. One can get, according to Remark 3, a k–internally stable set from each solution.

Theorem 2 suggests an algorithm which can find out all k–internally stable sets of an arbitrary graph.

Algorithm 1 Given a graph $G=(V,E)$ with $A^{[k]}=[a_{ij}^{[k]}]$ as its k–adjacency matrix, assign each vertex $v_i\in V$ a characteristic variable x_i as described in Eq.(4) and define $y_i=[x_i,1-x_i]^{\mathrm{T}}$. Taking the following steps one can obtain all k–internally stable sets of G.

Step 1 Compute the matrix M in Theorem 2.

Step 2 Check whether there exists a zero-column $\mathbf{0}_n$ in M. If not, G has no k–internally stable set and the computation comes to end. Otherwise, set

$$K=\{i|\mathrm{col}_i(M)=\mathbf{0}_n\}. \tag{16}$$

Step 3 For each l in K, consider the equation $\ltimes_{i=1}^{n}y_i=\delta_{2^n}^{l}$. We define

$$\begin{cases}S_1^n=(E_{\mathrm{d}})^{n-1}\ltimes W_{[2,2^{n-1}]},\\ \quad\vdots\\ S_i^n=(E_{\mathrm{d}})^{n-1}\ltimes W_{[2^i,2^{n-i}]},\\ \quad\vdots\\ S_n^n=(E_{\mathrm{d}})^{n-1}.\end{cases} \tag{17}$$

Then y_i can be obtained by computing $y_i=S_i^n\ltimes\delta_{2^n}^{l}, i=1,2,\cdots,n$.

Step 4 Select $y_i=\delta_2^1$ and construct $S_l=\{v_i|y_i=\delta_2^1\}$. S_l is a k–internally stable set of G.

All k–internally stable sets of G are

$$\{S_l|l\in K, S_l\text{ is produced by Steps 3 and 4.}\}$$

The k–internally stable number of G is $\beta_k(G)=\max\limits_{l\in K}\{|S_l|\}$, where $|S_l|$ is the cardinality of S_l. All k–absolute maximum internally stable sets of G are

$$\zeta=\{S_l|\,|S_l|=\beta_k(G)\}.$$

Remark 4 Since the algorithm above can find all k–internally stable sets of a graph, we can get some k–internally stable sets with some special properties, such as, there is no common vertex between every two k–internally stable sets of a family of k–internally stable sets, say, $\{S_1,S_2,\cdots,S_m\}$.

Remark 5 All the operations related to STP in Algorithm 1 and Theorems 1 and 2 can be easily completed by the MATLAB toolbox developed by Professors D Z Cheng and H S Qi, which is accessible at: http: //lsc.amss.ac.cn/dcheng/stp/STP.zip.

3.2 Searching k–absolute maximum internally stable set

Algorithm 1 provides a way to find out all k-internally stable sets of graphs, of course, including k-absolute maximum internally stable sets (k-AMISSs). In this subsection, we investigate the problem separately and present a necessary and sufficient condition of such kind of subset and an algorithm able to search all k-AMISSs of an arbitrary graph.

Lemma 1[21] Let $A^{[k]}=[a_{ij}^{[k]}]$ be the k-adjacency matrix of graph $G=(V,E)$, $S\subseteq V$ is a given subset. Assign each vertex $v_i\in V$ a variable x_i that $x_i=1$ if $v_i\in S$ and $x_i=0$ if $v_i\notin S$, then S is a k-AMISS if and only only if $(x_1,x_2,\cdots,x_n)$ is a maximum point of the function

$$f(x_1,\cdots,x_n)=\sum_{i=1}^{n}x_i-(n+1)\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}a_{ij}^{[k]}x_ix_j, \tag{18}$$

and the maximum value of f is non-negative.

Theorem 3 Let $A^{[k]}=[a_{ij}^{[k]}]$ be the k-adjacency matrix of graph $G=(V,E)$. Consider a subset $S\subseteq V$ whose characteristic vector is $V_S=[x_1,x_2,\cdots,x_n]$. Assume that

$$Y_S=\ltimes_{i=1}^{n}y_i=\delta_{2^n}^{k},\ y_i=[x_i,\bar{x}_i]^{\mathrm{T}},$$

then S is a k-AMISS of G if and only if the kth component of M is maximum among all the non-negative components, where

$$M=P(\sum_{i=1}^{n}T_i)-(n+1)\bar{M}, \tag{19}$$

in which

$$P=[1\ \ 0],\ T_i=(E_{\mathrm{d}})^{n-1}\ltimes W_{[2^i,2^{n-i}]},$$
$$Q=[1\ \ 0\ \ 0\ \ 0],\ \bar{M}=Q(\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}a_{ij}^{[k]}T_{ij}),$$
$$T_{ij}=(E_{\mathrm{d}})^{n-2}\ltimes W_{[2^j,2^{n-j}]}\ltimes W_{[2^i,2^{j-i-1}]}.$$

Proof From the proof of Theorem 1 we know that

$$x_ix_j=Q(T_{ij}\ltimes Y_S).$$

Thus the part $\sum\limits_{i=1}^{n}\sum\limits_{j=1,j\neq i}^{n}a_{ij}^{[k]}x_ix_j$ of f can be expressed as

$$\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}a_{ij}^{[k]}x_ix_j=Q(\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}a_{ij}^{[k]}T_{ij})\ltimes Y_S. \tag{20}$$

Next, we consider another part $\sum\limits_{i=1}^{n}x_i$ of f.

Using Eqs.(1) and (6), we can get

$$y_i = (E_{\rm d})^{n-1} \ltimes y_{i+1} \ltimes \cdots y_n \ltimes y_1 \ltimes \cdots \ltimes y_i = (E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]} \ltimes y_1 \ltimes \cdots \ltimes y_n = (E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]} \ltimes Y_S.$$

Thus $x_i = Py_i = P((E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]} \ltimes Y_S)$. We then have

$$\sum_{i=1}^{n} x_i = P((\sum_{i=1}^{n} (E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]}) \ltimes Y_S). \quad (21)$$

Substituting Eqs.(20) and (21) to Eq.(18), we get

$$f(x_1,\cdots,x_n) = P((\sum_{i=1}^{n} (E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]}) - (n+1)Q(\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} a_{ij}^{[k]}T_{ij})) \ltimes Y_S = M \ltimes Y_S, \quad (22)$$

where M is given in Eq.(19).

Recall that $Y_S = \ltimes_{i=1}^{n} y_i$, $y_i = [x_i, \bar{x}_i]^{\rm T}$ and $(x_1, x_2, \cdots, x_n)$ are determined uniquely by each other, or say, they are in one-to-one correspondence with each other, by which we know from Lemma 1 that if S is a k-AMISS of G, then its characteristic vector $Y_S = \delta_{2^n}^k$ is a maximum point of Eq.(22) and the maximum value is non-negative, i.e., $M \ltimes Y_S$ is the maximum non-negative value of Eq.(22). Note that $M \ltimes Y_S = M \ltimes \delta_{2^n}^k$ is just the kth component of M, and that the range of f is the set consisting of all different components of M (because $Y_S = \ltimes_{i=1}^{n} y_i \in \Delta_{2^n}$). Therefore the kth component of M is a maximum one among all the non-negative components.

Conversely, if the kth component of M is a maximum component among all the non-negative components, from Eq.(22) and the assumption that the characteristic vector of S is $Y_S = \delta_{2^n}^k$, we know Y_S is a maximum point of Eq.(22) and the maximum value is non-negative. Therefore, according to the equivalence between Eq.(22) and Eq.(18) and the one-to-one correspondence between Y_S and $(x_1, x_2, \cdots, x_n)$, we know $(x_1, x_2, \cdots, x_n)$ is a maximum point of Eq.(18) and the maximum value is non-negative. By Lemma 1 we know that S is k-AMISS of G. The proof is completed.

Similar to the relationship between Algorithm 1 and Theorem 2, Theorem 3 provides a way to search all k-AMISSs of graphs.

Algorithm 2 Assume that $A^{[k]} = [a_{ij}^{[k]}]$ is the k-adjacency matrix of graph $G = (V, E)$, assign each vertex $v_i \in V$ a characteristic variable x_i as described in Eq.(4) and define $y_i = [x_i, 1-x_i]^{\rm T}$. To get all k-AMISSs of G, one can take the following steps.

Step 1 Compute the matrix M in Theorem 3.

Step 2 Check whether all components of M are negative, if yes, G has no k-AMISS and the computation comes to end. Otherwise, set

$$K = \{i|{\rm col}_i(M) = \max({\rm col}(M))\}. \quad (23)$$

Step 3 For each l in K, consider the equation $\ltimes_{i=1}^{n} y_i = \delta_{2^n}^l$. Define $S_i^n = (E_{\rm d})^{n-1} \ltimes W_{[2^i,2^{n-i}]}$, $i = 1, 2, \cdots, n$. Then y_i can be obtained by computing $y_i = S_i^n \ltimes \delta_{2^n}^l$,$i = 1, 2, \cdots, n$.

Step 4 Select $y_i = \delta_2^1$ and construct the set

$$S_l = \{v_i|y_i = \delta_2^1\}.$$

S is a k-AMISS of G. All k-AMISSs are

$$\{S_l|l \in K, S_l \text{ is produced by Steps 3 and 4.}\}.$$

The following conclusion on k-maximum weight internally stable set (k-MWISS) can be proved in a very similar way to that of Theorem 3, and the proof is omitted.

Theorem 4 Consider a graph $G = (V, E)$ with a non-negative function $\omega : V \to \mathbb{R}$. For a given subset $S \subseteq V$, suppose that the characteristic vector of S is $Y_S = \delta_{2^n}^k$. Then S is a k-maximum weight internally stable set of G if and only if the kth component of $\tilde{M}$ is the largest one, where

$$\tilde{M} = P(\sum_{i=1}^{n} \omega(v_i)T_i) - (1 + \sum_{i=1}^{n} \omega(v_i))\bar{M}, \quad (24)$$

in which P, T_i and $\bar{M}$ are given in Eq.(19).

Remark 6 From Theorem 4 we can establish an algorithm to find out all k-maximum weight internally stable sets of graphs, which is very similar to Algorithm 2, just replace M of Algorithm 2 by $\tilde{M}$ in Eq.(24), and therefore we omit it here.

4 Solvability of k-track assignment problem

The k-track assignment problem is a kind of resource allocation problem in operation research and scheduling theory with aiming to assign n jobs to k-machines. Each job starts and ends at specific times and can be processed by only one machine. Each machine operates in a certain track (period) beginning and finishing at certain times. Moreover each machine can handle only one job or is idle at a time. A schedule is an assignment of jobs to machines such that the time intervals of the jobs assigned to the same machine do not conflict with each other, and that these time intervals are contained in the working time interval of the machine.

Mathematically the k-track assignment problem can be expressed as follows. Let I be the set of all intervals corresponding to all jobs, and F_j be the set of the time intervals of the jobs which can be processed on the machine j. The k-track assignment problem is to find k disjoint sets $S_1, S_2, \cdots, S_k \subseteq I$ satisfying

1) $S_j \subseteq F_j$ for $j = 1, 2, \cdots, k$;

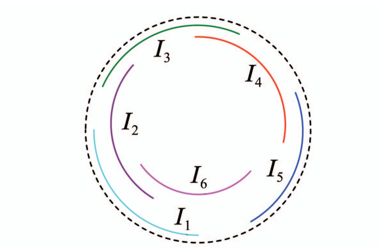
2) The intervals in S_j are not overlapped with each other;

3) $|S_1 \cup S_2 \cup \cdots \cup S_k|$ is maximal. $S = (S_1, S_2, \cdots, S_k)$ is called a feasible schedule if Eqs.(1)–

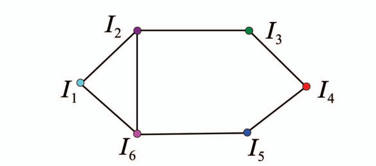
(2) hold. If Eq.(3) holds additionally, $S = (S_1, S_2, \cdots, S_k)$ is said to be optimal.

The k-track assignment problem can be modeled by circular-arc graphs. A circular-arc graph is the intersection graph of a set of arcs on a circle. It has one vertex for each arc in the set, and an edge between every pair of vertices corresponding to arcs that intersect.

Formally, let $I_1, I_2, \cdots, I_n$ be a set of arcs, the corresponding circular-arc graph is $G = (V, E)$ where $V = \{I_1, I_2, \cdots, I_n\}$ and $(I_i, I_j) \in E$ if and only if $(I_i \cap I_j) \neq \varnothing$. Fig.1 is an illustrative example.



(a) Set of arcs



(b) Corresponding circular–arc graph

Fig. 1 Example of circular-arc graph

For a given k-track assignment problem, using arcs to represent time intervals of jobs, the k-track assignment problem can be represented by a circular-arc graph. A schedule is then to find some group of disconnected vertices in the circular-arc graph, which corresponds to finding interval stable sets. To make the number of assigned jobs be largest, one needs to find some absolutely maximum internally stable sets in the circular-arc graph.

In this section, as an application of k-internally stable set and k-maximum internally stable set, we consider a simple case that the k machines have the same function, i.e., every job can be assigned to every machine. In this case, the k-track assignment problem is to find some disjoint internally stable sets or absolutely maximum internally stable sets in the set I, which can be achieved by the results/algorithms presented in Section 3. To solve such k-track assignment problem, we only need to set the k in the obtained results to be 1, consequently the k-internally stable set and k-absolutely maximum internally stable sets reduce to the internally stable set and absolutely maximum internally stable sets, respectively.

Based on the analysis above and the results of Section 3, we have the following conclusions.

Proposition 1 For a given k-track assignment problem, assume that its graph model is $G = (V, E)$ with $A^{[1]} = (a_{ij}^{[1]})$ being the 1-adjacency matrix. The k-track assignment problem has a feasible schedule if and only if there exist k column indices $1 \leqslant j_1, j_2, \cdots, j_k \leqslant 2^n$ such that

$$\mathrm{col}_{j_i}(M) = \mathbf{0}_n,\ i = 1, 2, \cdots, k. \tag{25}$$

where

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix},\ M_i = Q \sum_{j=1}^{n} a_{ij}^{[1]} T_{ij},\ i = 1, 2, \cdots, n,$$

$$T_{ij} = (E_{\mathrm{d}})^{n-2} \ltimes W_{[2^j, 2^{n-j}]} \ltimes W_{[2^i, 2^{j-i-1}]},$$
$$Q = [1, 0, 0, 0].$$

Proposition 2 The k-track assignment problem in Proposition 1 has an optimal schedule if and only if there are k maximum non-negative components in M,

$$M = P(\sum_{i=1}^{n} T_i) - (n+1)\bar{M}, \tag{26}$$

in which

$$P = [1, 0],\ T_i = (E_{\mathrm{d}})^{n-1} \ltimes W_{[2^i, 2^{n-i}]},$$
$$\bar{M} = Q(\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} a_{ij}^{[k]} T_{ij}),\ Q = [1, 0, 0, 0],$$
$$T_{ij} = (E_{\mathrm{d}})^{n-2} \ltimes W_{[2^j, 2^{n-j}]} \ltimes W_{[2^i, 2^{j-i-1}]}.$$

The proofs of Propositions 1 and 2 follow from Theorems 1 and 3 immediately and are omitted.

Remark 7 Based on Proposition 2, Algorithm 2 can be applied to solve the track assignment problem after a slight modification that in Step 1 of Algorithm 2 replace the matrix M by the matrix M in Proposition 2. In practice, we first use Algorithm 2 to find out all 1-absolute maximum internally stable sets of the graph that is the model of the track assignment problem to be solved, then collect the disjoint sets among these 1-absolute maximum internally stable sets, the resulting sets are the optimal schedules of the track assignment problem. Next we use an example in [21] to illustrate it.

Example 1 Consider the k-track assignment problem shown in Fig.2(a), which contains eight jobs varying from 8 a.m to 11 p.m, whose graph model is $G = (V, E)$, $V = \{1, 2, \cdots, 8\}$, as shown in Fig.2(b).

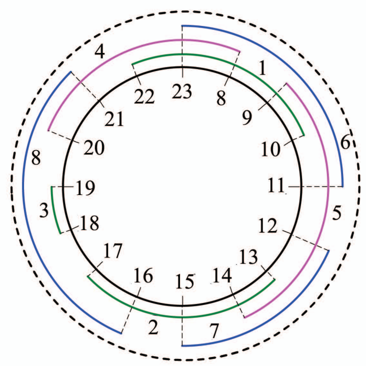


Fig.2(a) Jobs varing from 8 a.m. to 11 p.m.

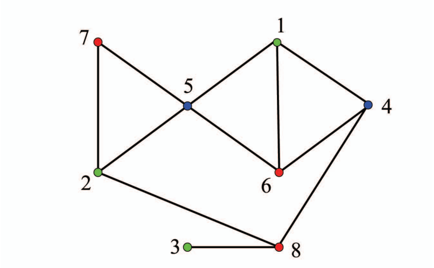


Fig.2(b) Corresponding graph model

We first give the 1-adjacency matrix $A^{[1]} = [a_{ij}^{[1]}]$ of G.

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

Step 1 According to (26) we can get the matrix M as follows:

$$\begin{aligned}
&M = \\
&[-190,-137,-155,-102,-137,-84,-102,-49, \\
&-119,-66,-102,-49,-84,-31,-67,-14, \\
&-137,-102,-102,-102,-67,-67,-32,-66,-31, \\
&-49,-14,-49,-14,-32,\underline{3},-173,-138,-138, \\
&-103,-120,-85,-85,-50,-102,-67,-85,-50, \\
&-67,-32,-50,-15,-120,-103,-85,-68,-85, \\
&-68,-50,-33,-49,-32,-32,-15,-32,-15, \\
&-15,2,-137,-102,-120,-85,-84,-49,-67,-32, \\
&-84,-49,-85,-50,-49,-14,-50,-15,-84,-67, \\
&-67,-50,-49,-32,-32,-15,-31,-14,-32, \\
&-15,-14,\underline{3},-15,2,-120,-103,-103,-86,-67, \\
&-50,-50,-33,-67,-50,-68,-51,-32,-15,-33, \\
&-16,-67,-68,-50,-51,-32,-33,-15,-16,-14, \\
&-15,-15,-16,\underline{3},2,2,1,-137,-84,-102,-49, \\
&-102,-49,-67,-14,-84,-31,-67,-14,-67, \\
&-14,-50,\underline{3},-102,-67,-67,-32,-85,-50,-50, \\
&-15,-49,-14,-32,\underline{3},-50,-15,-33,2,-120,-85, \\
&-85,-50,-85,-50,-50,-15,-67,-32,-50, \\
&-15,-50,-15,-33,2,-85,-68,-50,-33,-68, \\
&-51,-33,-16,-32,-15,-15,2,-33,-16,-16, \\
&1,-84,-49,-67,-32,-49,-14,-32,\underline{3},-49, \\
&-14,-50,-15,-32,\underline{3},-33,2,-49,-32,-32, \\
&-15,-32,-15,-15,2,-14,\underline{3},-15,-15,2,-16, \\
&1,-67,-50,-50,-33,-32,-15,-15,2,-32, \\
&-15,-33,-16,-15,2,-16,1,-32,-33,-15, \\
&-16,-15,1-16,2,\underline{3},2,2,1,2,1,1,0].
\end{aligned}$$

Step 2 There are nine maximum non-negative components 3 in M, which are marked by underlines. Their positions are 32, 94, 125, 144, 156, 200, 206, 218 and 249, respectively. Thus

$$K = \{32, 94, 125, 144, 156, 200, 206, 218, 249\}.$$

Step 3 For each element $l \in K$, computing $y_i = S_i^8 \ltimes \delta_{256}^l$, $i = 1, 2, \cdots, 8$, where S_i^8 are defined as in Eq.(17). Take $l = 32$ for example, we get

$$\begin{aligned}
&y_1 = S_1^8 \ltimes \delta_{256}^{32} = \delta_2^1,\ y_2 = S_2^8 \ltimes \delta_{256}^{32} = \delta_2^1, \\
&y_3 = S_3^8 \ltimes \delta_{256}^{32} = \delta_2^1,\ y_4 = S_4^8 \ltimes \delta_{256}^{32} = \delta_2^2, \\
&y_5 = S_5^8 \ltimes \delta_{256}^{32} = \delta_2^2,\ y_6 = S_6^8 \ltimes \delta_{256}^{32} = \delta_2^2, \\
&y_7 = S_7^8 \ltimes \delta_{256}^{32} = \delta_2^2,\ y_8 = S_8^8 \ltimes \delta_{256}^{32} = \delta_2^2.
\end{aligned}$$

Step 4 Select y_1, y_2, y_3 and construct the set $S_{32} = \{1, 2, 3\}$. S_{32} is a 1-absolute maximum internally stable set of G.

All the other 1-absolute maximum internally stable sets can be obtained in a similar way. All of them are listed as follows:

$$\begin{cases} S_{32} = \{1,2,3\},\ S_{94} = \{1,3,7\},\ S_{125} = \{1,7,8\}, \\ S_{144} = \{2,3,4\},\ S_{156} = \{2,3,6\},\ S_{200} = \{3,4,5\}, \\ S_{206} = \{3,4,7\},\ S_{218} = \{3,6,7\},\ S_{249} = \{6,7,8\}. \end{cases} \tag{27}$$

For the 2-track assignment problem, just choose two disjoint sets in Eq.(27), the collected sets correspond to an optimal schedule. All optimal schedules are as follows:

$$\begin{cases} (\{1,2,3\},\{6,7,8\}), (\{2,3,4\},\{6,7,8\}), \\ (\{3,4,5\},\{6,7,8\}), (\{1,7,8\},\{2,3,4\}), \\ (\{1,7,8\},\{2,3,6\}), (\{1,7,8\},\{3,4,5\}). \end{cases} \tag{28}$$

To obtain the optimal schedules of 3-track assignment problem, we need to know additional 1-internally stable sets, which can be achieved by Algorithm 1, all of them are listed in the following. (The detailed process is omitted, which is similar to the above analysis procedure).

$$\begin{cases} S_{64} = \{1,2\},\ S_{96} = \{1,3\},\ S_{128} = \{1\}, \\ S_{160} = \{2,3\},\ S_{176} = \{2,4\},\ S_{188} = \{2,6\}, \\ S_{192} = \{2\},\ S_{208} = \{3,4\},\ S_{216} = \{3,5\}, \\ S_{220} = \{3,6\},\ S_{222} = \{3,7\},\ S_{224} = \{3\}, \\ S_{232} = \{4,5\},\ S_{238} = \{4,7\},\ S_{240} = \{4\}, \\ S_{247} = \{5,8\},\ S_{248} = \{5\},\ S_{250} = \{6,7\}, \\ S_{251} = \{6,8\},\ S_{252} = \{6\},\ S_{253} = \{7,8\}, \\ S_{254} = \{7\},\ S_{255} = \{8\}. \end{cases} \tag{29}$$

Collecting two disjoint sets from (28) and (29), one from (28), the other from (29), we can get all optimal schedules of the 3-track assignment problem, which are listed as follows:

$$\begin{cases} (\{1,2,3\},\{4,5\},\{6,7,8\}), \\ (\{1,2\},\{3,4,5\},\{6,7,8\}), \\ (\{1,7,8\},\{2,6\},\{3,4,5\}), \\ (\{1,7,8\},\{4,5\},\{2,3,6\}). \end{cases} \tag{30}$$

Remark 8 It is interesting to find from Eq.(30) that each of the four optimal schedules contains all the jobs; hence three tracks (machines) are enough for the assignment problem. We call the schedule having the least number of tracks and containing the entire jobs completely optimal schedule. Our method provides a way to find all the completely optimal schedules for the k-track assignment problems.

Remark 9 The method described above can also be applied to solve other similar assignment problems which can be modeled as circular-arc graphs such as the frequency assignment problem, memory assignment problem and room assignment problem.

5 Illustrative example

This section uses an example in [21] to illustrate the procedure to search all k-internally stable sets of a graph by using the proposed method, and to verify the correctness of the results/algorithms presented in this paper.

Example 2 Consider the directed graph $G = (V, E)$ with $V = \{v_1, v_2, \cdots, v_8\}$ as shown in Fig.3.

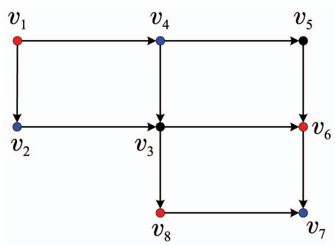


Fig. 3 The graph of Example 2

We use Algorithm 1 to search all k-internally stable sets and k-absolute maximum internally stable sets of G. Let us first consider the case of $k = 2$.

According to Eq.(3), it is easy to get the 2-adjacency matrix of G.

$$A^{[2]} = [a_{ij}^{[2]}] = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Steps 1 and 2 With $A^{[2]}$ we can have the matrix M in Eq.(7), in which the numbers of zero-columns are $123, 124, 126, 127, 128, 174, 176, 184, 190, 192, 216, 224, 238, 240, 247, 248, 251, 252, 254, 255$.

And thus, the set K is

$$K = \{123, 124, 126, 127, 128, 174, 176, 184, 190, 192, 216, 224, 238, 240, 247, 248, 251, 252, 254, 255\}.$$

By Theorem 1 we know that G contains a 2-internally stable set; by Corollary 1 we further know that these twenty zero-columns determine twenty 2-internally stable sets.

Step 3 For each element $l \in K$, computing $y_i = S_i^8 \ltimes \delta_{256}^l$, we can get y_i, $i = 1, 2, \cdots, 8$. Then the set $S_l = \{v_i | y_i = \delta_2^1\}$ is a 2-internally stable set.

Take $l = 123$ for example, we have

$$\begin{cases} y_1 = S_1^8 \ltimes \delta_{256}^{123} = \delta_2^1, \ y_2 = S_2^8 \ltimes \delta_{256}^{123} = \delta_2^2, \\ y_3 = S_3^8 \ltimes \delta_{256}^{123} = \delta_2^2, \ y_4 = S_4^8 \ltimes \delta_{256}^{123} = \delta_2^2, \\ y_5 = S_5^8 \ltimes \delta_{256}^{123} = \delta_2^2, \ y_6 = S_6^8 \ltimes \delta_{256}^{123} = \delta_2^1, \\ y_7 = S_7^8 \ltimes \delta_{256}^{123} = \delta_2^2, \ y_8 = S_8^8 \ltimes \delta_{256}^{123} = \delta_2^1. \end{cases} \tag{31}$$

Collecting y_1, y_6, and y_8, we then get a 2-internally stable set $S_{123} = \{v_1, v_6, v_8\}$, which is shown in red in Fig.3.

Consider another example, for $l = 174$, computing $y_i = S_i^8 \ltimes \delta_{256}^{174}$, $i = 1, 2, \cdots, 8$, we obtain the following:

$$\begin{cases} y_1 = S_1^8 \ltimes \delta_{256}^{174} = \delta_2^2, \ y_2 = S_2^8 \ltimes \delta_{256}^{174} = \delta_2^1, \\ y_3 = S_3^8 \ltimes \delta_{256}^{174} = \delta_2^2, \ y_4 = S_4^8 \ltimes \delta_{256}^{174} = \delta_2^1, \\ y_5 = S_5^8 \ltimes \delta_{256}^{174} = \delta_2^2, \ y_6 = S_6^8 \ltimes \delta_{256}^{174} = \delta_2^2, \\ y_7 = S_7^8 \ltimes \delta_{256}^{174} = \delta_2^1, \ y_8 = S_8^8 \ltimes \delta_{256}^{174} = \delta_2^2. \end{cases} \tag{32}$$

y_2, y_4 and y_7 indicate another 2-internally stable set $S_{174} = \{v_2, v_4, v_7\}$, which is marked in blue in Fig.3.

Similarly, by computing $y_i = S_i^8 \ltimes \delta_{256}^l$ for other ls in K, all the other 2-internally stable sets can be obtained. They are listed in the following.

$$\begin{aligned} & S_{124} = \{v_1, v_6\}, S_{126} = \{v_1, v_7\}, S_{127} = \{v_1, v_8\}, \\ & S_{128} = \{v_1\}, S_{174} = \{v_2, v_4\}, S_{176} = \{v_2, v_5\}, \\ & S_{184} = \{v_2, v_7\}, S_{190} = \{v_2\}, S_{192} = \{v_3, v_5\} \\ & S_{216} = \{v_3\}, \ S_{224} = \{v_4, v_7\}, S_{238} = \{v_4\}, \\ & S_{240} = \{v_5, v_8\}, S_{248} = \{v_5\}, S_{251} = \{v_6, v_8\}, \\ & S_{252} = \{v_6\}, \ S_{254} = \{v_7\}, S_{255} = \{v_8\}. \end{aligned}$$

Next, we consider the case of $k \geqslant 3$.

For $k = 3$, using the procedure similar to the case of $k = 2$, we can get all 3-internally stable sets. The following are the ones except single vertex (a single vertex is always a k-internally stable set for any integer k and any graphs): $\{1, 7\}, \{2, 4\}, \{2, 5\}, \{3, 5\}, \{5, 8\}, \{6, 8\}$.

For $k = 4$, all 4-internally stable sets except single vertex are $\{2, 4\}, \{2, 5\}, \{3, 5\}, \{5, 8\}, \{6, 8\}$.

Continually applying Algorithm 1, we find an interesting result that when $k \geqslant 4$ the k-internally stable sets are the same. Actually we can prove this by showing that $A^{[k]}$ remains unchanged when $k \geqslant 4$.

Now, we consider k-absolute maximum internal stable sets of G. For $k = 2$, the maximum non-negative element of the matrix M in Eq.(19) is 3, which occurs at the positions 123 and 174 of M. From Eqs.(31) and (32), we know that the sets $S_{123} = \{v_1, v_6, v_8\}$ and $S_{174} = \{v_2, v_4, v_7\}$ are 2-absolute maximum internally stable sets of M.

For $k = 3$, there are six maximum non-negative elements 2 in the matrix M in Eq.(19), whose positions are 126, 176, 184, 216, 247, 251, respectively. Thus the set K in Step 2 of Algorithm 2 is $K = \{126, 176, 184, 216, 247, 251\}$.

For each element $l \in K$, computing $y_i = S_i^8 \ltimes \delta_{256}^l$, we can get the following 3-absolute maximum internally stable sets: $\{1, 7\}, \{2, 4\}, \{2, 5\}, \{3, 5\}, \{5, 8\}, \{6, 8\}$.

When $k \geqslant 4$, similarly to the case of k-internal stable set, the k-absolute maximum internally stable sets of G also remain unchanged as follows: $\{2, 4\}$, $\{2, 5\}, \{3, 5\}, \{5, 8\}, \{6, 8\}$.

The results above are consistent with [21]. As for the Theorem 3 and Algorithm 2 regarding searching k-AMISSs of graphs, the correctness of them has also been verified by Example 1 in Section 4.

6 Conclusions

Graphs provide discrete mathematical models for many concrete real-world problems. This paper introduces STP to the field of graph theory and uses it to investigate the problems of k-internally stable set, k-maximum internally stable set, and k-absolute maximum internally stable set of graphs. These concepts serve as mathematical models for some real-world problems such as the k-track assignment problem. A set of new results are obtained with the help of STP, including three necessary and sufficient conditions of the three kinds of internally stable set. Based on these new conditions, two algorithms able to find all these subsets of graphs are established. Moreover, the new results are applied to the k-track assignment problem, and a new method is proposed for solving the problem. The approach of this paper provides a new angle and means to understand and analyse the structure of graphs and the related problems.

References:

[1] CORNELSEN S, STEFANO G D. Track assignment [J]. *Journal of Discrete Algorithms*, 2007, 5(2): 250 – 261.

[2] DEMANGE M, STEFANO G, LEROY-BEAULIEU B. On the online track assignment problem [J]. *Discrete Applied Mathemathics*, 2012, 160(7): 1072 – 1093.

[3] SEVERSON T A, PALEY D A. Optimal sensor coordination for multitarget search and track assignment [J]. *IEEE Transactions on Aeronautic Electric Systems*, 2014, 50(3): 2312 – 2319.

[4] SEVERSON T A, PALEY D A. Distributed multitarget search and track assignment wth consensus-based coordination [J]. *IEEE Sensors Journal*, 2015, 15(2): 864 – 875.

[5] CHENG D Z, QI H S, ZHAO Y. *An Introduction to Semi-tensor Product of Matrices and Its Applications* [M]. Singapore: World Scientific Publishing Co. Pte. Ltd, 2012.

[6] WANG Y Z, ZHANG C H, LIU Z B. A matrix approach to graph maximum stable set and coloring problems with application to multi-agent systems [J]. *Automatica*, 2012, 48(7): 1227 – 1236.

[7] MENG M, FENG J E. A matrix approach to hypergraph stable set and coloring problems with its application to storing problem [J]. *Journal of Applied Mathematics*, 2014, 34(4): 1 – 9.

[8] LI F F, YU Z. Feedback control and output feedback control for the stabilisation of switched Boolean networks [J]. *International Journal of Control*, 2016, 89(2): 1 – 9.

[9] LI H T, WANG Y Z, XIE L H, et al. Disturbance decoupling control design for switched Boolean control networks [J]. *System & Control Letters*, 2014, 72(72): 1 – 6.

[10] FORNASINI E, VALCHER M E. Finite-horizon optimal control of Boolean control networks [C] //*The 52nd IEEE Conference on Decision and Control*. Florence: IEEE, 2013: 3864 – 8369.

[11] LASCHOV D, MARGALIOT M. On Boolean control networks with maximal topological entropy [J]. *Automatica*, 2014, 50(10): 1654 – 1662.

[12] QI H S, LIU T, CHENG D A. Observability of dynamic control games [C] //*Proceedings of the Cybernetics and Intelligent Systems and IEEE Conference on Robotics, Automation and Mechatronics*. Angkor Wat: IEEE, 2015.

[13] LIU X Y, ZHU J D. On potential equations of finite games [J]. *Automatica*, 2016, 68(C): 245 – 253.

[14] ZHANG L J, ZHANG K Z. L2 stability, H_∞ control of switched homogeneous nonlinear systems and their semi-tensor product of matrices representationa [J]. *International Journal of Robust and Nonlinear Control*, 2013, 23(6): 638 – 652.

[15] FENG J E, LV H L, CHENG D Z. Multiple fuzzy relation and its application to coupled fuzzy control [J]. *Asian Jouranl of Control*, 2013, 15(5): 1313 – 1324.

[16] YAN Y Y, CHEN Z Q, LIU Z X. Semi-tensor product approach to controlliability and stabilizability of finite automata [J]. *Journal of Systems Engineering and Electronics*, 2015, 26(1): 134 – 141.

[17] YAN Y Y, CHEN Z Q, YUE J M, et al. STP approach to model controlled automata with application to reachability analysis of DEDS [J]. *Asian Journal of Control*, 2016, 18(6): 2027 – 2036.

[18] YAN Y Y, CHEN Z Q, YUE J M. Algebraic state space approach to model and control combined automata [J]. *Frontiers of Computer Science*, 2016: 1 –13.

[19] XU X R, HONG Y G. Matrix approach to model matching of asynchronous sequential machines [J]. *IEEE Transactions on Automatic Control*, 2013, 58(11): 2974 – 2979.

[20] CRAMA Y, HAMMER P L. *Boolean Functions: Theory, Algorithms, and Applications* [M]. New York: Cambridge University Press, 2011.

[21] HAMMER P L, RUDEANU S, BELLMAN R. *Boolean Methods in Operations Research and Related Areas* [M]. Berlin: Springer-Verlag, 1968.

作者简介:

岳菊梅 (1983–), 女, 博士, 目前研究方向为模糊逻辑与模糊控制系统、逻辑动态系统, E-mail: jmyue2014@163.com;

陈增强 (1964–), 男, 博士, 目前研究方向为智能预测控制与智能优化计算、复杂系统建模优化与控制、多智能体系统与复杂网络、离散事件逻辑动态系统建模与分析, E-mail: chenzq@nankai.edu.cn;

闫永义 (1980–), 男, 博士, 目前研究方向为复杂系统建模优化与控制、逻辑动态系统, E-mail: yyyan@mail.nankai.edu.cn;

金 鑫 (1986–), 男, 博士, 目前研究方向为高速移栽系统理论及应用领域中的土壤–植物–机器互作机理、智能化精准作业、移栽机运动参数与作业质量检测技术, E-mail: jx.771@163.com.