SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

January 2014, Vol. 57 012203:1–012203:11
doi: 10.1007/s11432-012-4689-1

Cluster synchronization of a class of multi-agent systems with a bipartite graph topology

WANG Qiang & WANG YuZhen*

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

Received July 25, 2012; accepted October 25, 2012; published online November 30, 2012

Abstract This paper investigates cluster synchronization of a class of multi-agent systems with a directed bipartite graph topology, and presents a number of new results by using the neighbor's rules for the following two cases: I) there is competition among the agents of different clusters, and II) there are both competition and cooperation among the agents. Firstly, for case I), a linear control protocol is designed for cluster synchronization of multi-agent systems, and a method is presented to determine the final state with the initial conditions based on state-space decomposition. Secondly, we study case II), and design a control protocol based on the information of neighbors and that of two-hop neighbors (that is, neighbors' neighbors). Finally, two examples are studied by using our presented results. The study of illustrative examples with simulations shows that our results as well as designed control protocols work very well in studying the cluster synchronization of this class of multi-agent systems.

Keywords multi-agent systems, bipartite graph, clustering synchronization, competition, cooperation, state-space decomposition.

Citation Wang Q, Wang Y Z. Cluster synchronization of a class of multi-agent systems with a bipartite graph topology. Sci China Inf Sci, 2014, 57: 012203(11), doi: 10.1007/s11432-012-4689-1

1 Introduction

Recently, multi-agent systems have attracted a good deal of attention in the field of control and automation, partly because of their broad applications in many areas such as distributed formation control [1–4], flocking [5–9], and congestion control in communication networks [10,11]. The cooperation among agents plays an important role for the design of control protocols in many fields such as unmanned air vehicles, computer network [12–15] and distributed data fusion in sensor networks [16–18].

However, there exists another case completely different from the cooperation, i.e., the relationship among agents is not cooperation but competition with each other, or partly cooperation with some agents and partly competition against some other agents. This relationship was introduced, but not discussed deeply, in [19], where the authors believed that how to introduce competition to distributed coordination to represent more realistic scenarios is both interesting and important. Actually, this class of problems with competition exists extensively in reality such as the pursuer-invader problem, competition among several species in a specified area, etc. One of this class of problems is that agents in a coupled multi-agent system can realize cluster synchronization; that is, the system is required to split into several

*Corresponding author (email: yzwang@sdu.edu.cn)

clusters so that the agents in the same cluster can synchronize, while difference of interest quantities exists between each couple of different clusters [20]. To the authors' best knowledge, most researches in the literature are only based on local cooperation. Therefore, it is very meaningful and challenging for us to study this class of problems with competition relationship.

In this paper, we investigate cluster synchronization of a class of multi-agent systems with a directed bipartite graph topology, and present a number of new results by using the neighbor's rules for the following two cases: I) there is competition among the agents of different clusters, and II) there are both competition and cooperation among the agents. The main contributions of this paper are as follows: (i) a new method is presented to describe the competition among agents in a multi-agent network; (ii) based on the competition among agents, two kinds of control protocols are designed for a class of multi-agent systems with and without cooperation among the agents, respectively; (iii) a new method is obtained to determine the final states of the systems with the initial conditions by decomposition of the state space.

The remainder of the paper is organized as follows. Section 2 is the problem formulation and preliminaries. Section 3 is the main results of the paper. In this section, some properties on bipartite graphs are provided first, and two control protocols are designed. In Section 4, we give two illustrative examples to support our new results followed by the conclusion in Section 5.

2 Problem statement and preliminaries

In this section, we give the problem statement first, and then provide some preliminaries on algebraic graphs, which will be used in the sequel.

Consider the following system with n agents:

$$\begin{cases} \dot{x}_i = u_i, \\ x_i(t_0) = x_i^{(0)}, \qquad i = 1, 2, \ldots, n, \end{cases} \tag{1}$$

where $x_i \in \mathbb{R}$ is the state of agent i, $u_i \in \mathbb{R}$ is the control input of agent i, and $x_i(t_0) = x_i^{(0)}$ is the initial condition of agent i, $i = 1, 2, \ldots, n$.

The objective of this paper is to design control protocols such that system (1) can realize cluster synchronization in the following two cases, respectively:

Case I: The agents evolve on a bipartite graph only with a competitive relationship.

Case II: The agents evolve on a bipartite graph with both competition and cooperation among them.

Remark 1. It is noted that a bipartite graph is a special topology structure. In this paper, for ease of expression and analysis, we only consider the bipartite graph. In fact, for an arbitrary topology, cluster synchronization can hardly be achieved unless some strong additional conditions are added.

In the following, we recall some fundamental knowledge on algebraic graph theory and matrix theory, which will be used in the development of this research.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ be a directed graph of nth order with the set of nodes $\mathcal{V} := \{v_1, v_2, \ldots, v_n\}$, the set of edges (i.e., ordered pairs of the agents) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and $\mathcal{A} : \mathcal{E} \to \mathbb{R}^+$ assigning a (positive) weight to each edge such that if $e_{ij} := (v_i, v_j) \in \mathcal{E}, \mathcal{A}(e_{ij}) = a_{ij}$. The matrix $\boldsymbol{A} = [a_{ij}]_{n\times n}$ is named the adjacency matrix of the graph $\mathcal{G}$. For any $i, j \in \mathcal{V}$, $a_{ij} > 0$ if and only if $j \in \mathcal{N}_i$, where $\mathcal{N}_i = \{j \mid e_{ij} = (v_i, v_j) \in \mathcal{E}\}$. For notational convenience, we just consider the case of simple graphs in this study, that is, $e_{ii} \notin \mathcal{E}, i = 1, 2, \ldots, n$. The matrix $\boldsymbol{D} = [d_{ij}] \in \mathbb{R}^{n\times n}$ is the valency matrix of the topology $\mathcal{G}$, and d_{ij} is defined as

$$d_{ij} = \begin{cases} \displaystyle\sum_{k\in\mathcal{N}_i(t)} a_{ik}, & j = i, \\ 0, & j \neq i. \end{cases} \tag{2}$$

A directed tree is such a directed graph whose every vertex except the root, which has only children but no parent, has exactly one parent. A spanning tree of a digraph is a directed tree that contains all the vertices of the digraph [21]. The graph $\mathcal{G}$ is called a bipartite graph, if its vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V (See Figure 1).

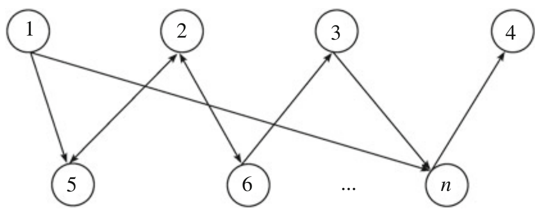

Figure 1 A bipartite digraph with n nodes.

We say that $\{X_1, X_2, \ldots, X_k\}$, $k > 1$, is a partition of set $X = \{x_1, x_2, \ldots, x_n\}$ if $X_i \neq \emptyset$, $X_i \bigcap X_j = \emptyset$, and $\bigcup_{i=1}^{k} X_i = X$. Thus, for any $x_i \in X$, there exists a unique subset X_j in the partition such that $x_i \in X_j$. Furthermore, for $x_i \in X$, we use $\hat{i}$ to denote the index of the subset in which x_i lies (that is, $x_i \in X_{\hat{i}}$), $1 \leqslant \hat{i} \leqslant k$. Obviously, if x_i and x_j are in the same cluster, then $\hat{i} = \hat{j}$.

Consider system (1), and assume that $\{X_1, X_2, \ldots, X_k\}$ is a partition of all the agents. The following definition is cited from [20].

Definition 1. Multi-agent system (1) is said to realize k-cluster synchronization with the partition $\{X_1, X_2, \ldots, X_k\}$ if $\lim_{t\to\infty} \|x_i(t) - x_j(t)\| = 0$ for $\hat{i} = \hat{j}$ and $\lim_{t\to\infty} \sup \|x_i(t) - x_j(t)\| > 0$ for $\hat{i} \neq \hat{j}$.

Remark 2 (See [20])**.** A similar concept "group consensus" of multi-agent systems was defined in [22]. The group consensus is weaker than the cluster synchronization defined here, because we require additionally that the differences between different clusters do not go to 0 as $t \to \infty$.

3 Main results

This section studies cluster synchronization of multi-agent system (1) with a directed bipartite graph topology (See Figure 1), and presents a number of new results for the following two cases: I) the agents compete against their adjacent agents; II) the agents compete against their neighbors and cooperate with two-hop neighbors at the same time. Meanwhile, we give a method to determine the final state with the initial condition by decomposing state space.

3.1 The case that agents compete against their neighbors

In this subsection, we consider the case in which there is only competition among the agents.

Consider system (1), and assume that its information topology is a directed bipartite graph which has a spanning tree, denoted by $\mathcal{G}$, and the adjacency matrix of $\mathcal{G}$ is $\boldsymbol{A} = [a_{ij}]_{n\times n}$.

Motivated by competition models of Ecology systems [23,24] and cooperation model [1], we can use $(x_i - x_j)$ to express the cooperation relationship between agents i and j, and $(x_i + x_j)$ to describe the competitive exclusion principle. Based on this, we design a linear cluster synchronization protocol for the case where there is only competition among the agents as follows:

$$u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}(x_i + x_j), \tag{3}$$

where a_{ij} is the weight between agents i and j.

Substituting (3) into system (1) yields

$$\dot{x}_i = -\sum_{j\in\mathcal{N}_i} a_{ij}(x_i + x_j), \quad i = 1, 2, \ldots, n, \tag{4}$$

which can be rewritten as

$$\dot{\boldsymbol{x}} = -(\boldsymbol{D} + \boldsymbol{A})\boldsymbol{x}, \tag{5}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$, and $\boldsymbol{D}$ is the valency matrix of the graph $\mathcal{G}$.

To study the cluster synchronization of system (5), we now present a lemma on the properties of bipartite graphs.

Lemma 1. Assume that $\mathcal{G}$ is a bipartite digraph with n nodes $\{v_1, \ldots, v_n\}$ and has a spanning tree. Then, $\mathrm{rank}(\boldsymbol{D}+\boldsymbol{A}) = n-1$, and each non-zero eigenvalue λ of matrix $\boldsymbol{D}+\boldsymbol{A}$ has a positive real part, where $\boldsymbol{D}$ is the valency matrix of $\mathcal{G}$, and $\boldsymbol{A} = [a_{ij}]_{n\times n}$ is its adjacency matrix.

Proof. For the detailed proof, please refer to Appendix.

Now, we consider cluster synchronization of system (5), and present the main result of this part.

Theorem 1. Consider the multi-agent system (1) with the bipartite digraph $\mathcal{G}$. Then, the system is stable and all the agents can realize cluster synchronization under control protocol (3).

Proof. Consider system (1). Since $\mathcal{G}$ is a bipartite digraph and has a spanning tree, according to Lemma 1, $\mathrm{rank}(\boldsymbol{D}+\boldsymbol{A}) = n-1$, and $\mathrm{Re}(\boldsymbol{D}+\boldsymbol{A}) \geqslant 0$.

Denote by $\lambda_1, \lambda_2, \ldots, \lambda_s$ and $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n$ the different eigenvalues and n linearly independent generalized eigenvectors of matrix $\boldsymbol{D}+\boldsymbol{A}$, respectively. According to Lemma 1, the zero eigenvalue of $\boldsymbol{D}+\boldsymbol{A}$ is simple. Without loss of generality, let $\lambda_1 = 0$ and

$$\boldsymbol{\xi}_1 = [\underbrace{1, \ldots, 1}_{r}, \underbrace{-1, \ldots, -1}_{n-r}]^{\mathrm{T}} \tag{6}$$

be the corresponding eigenvector. We construct matrix $\boldsymbol{T} := [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n]$, and take $\boldsymbol{z} = \boldsymbol{T}^{-1}\boldsymbol{x}$ as a coordinate transformation. Then, under the new coordinate transformation, system (5) can be changed into

$$\dot{\boldsymbol{z}} = -\boldsymbol{J}\boldsymbol{z}, \tag{7}$$

where $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^n$, and $\boldsymbol{J}$ is the Jordan canonical form of $\boldsymbol{D}+\boldsymbol{A}$.

Define $\boldsymbol{z}_e = [z_2, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^{n-1}$. Then $\boldsymbol{z} = [z_1, \boldsymbol{z}_e^{\mathrm{T}}]^{\mathrm{T}}$, with which closed-loop dynamics (7) can be rewritten as

$$\dot{z}_1 = 0, \tag{8}$$

$$\dot{\boldsymbol{z}}_e = -\boldsymbol{J}_e\boldsymbol{z}_e, \tag{9}$$

where $\boldsymbol{J}_e = \mathrm{diag}\{\boldsymbol{J}_2, \boldsymbol{J}_3, \ldots, \boldsymbol{J}_s\} \in \mathbb{R}^{(n-1)\times(n-1)}$, $\boldsymbol{J}_i$ is the Jordan canonical block with respect to the eigenvalue λ_i.

Consider subsystem (9). Since $\mathrm{Re}(\lambda_i) > 0$ for all $i \geqslant 2$, it is easy to see that $\boldsymbol{z}_e \to 0$. On the other hand, $\dot{z}_1(t) \equiv 0$ implies that $z_1(t) \equiv z_1(0)$. Thus, $\boldsymbol{z}(t) \to [z_1(0), 0, \ldots, 0]^{\mathrm{T}}$ as $t \to \infty$. With this, noticing $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{z}$ and the construction of $\boldsymbol{T}$, we have

$$\boldsymbol{x}(t) \to z_1(0)\boldsymbol{\xi}_1 \tag{10}$$

as $t \to \infty$, where $\boldsymbol{z}(0) = [z_1(0), z_2(0), \ldots, z_n(0)]^{\mathrm{T}} = \boldsymbol{T}^{-1}\boldsymbol{x}^{(0)}$, $\boldsymbol{x}^{(0)}$ is the initial condition of system (1).

Noticing $\boldsymbol{\xi}_1$ is given in (6), we see that system (5) can realize the cluster synchronization, and the final state is $z_1(0)\boldsymbol{\xi}_1$.

Remark 3. The proof of Theorem 1 itself provides a method to determine the final state when the system reaches cluster synchronization. The method contains the following steps:

1) Calculate the eigenvalues and linearly independent generalized eigenvectors of $(\boldsymbol{D}+\boldsymbol{A})$: $\{\lambda_1, \lambda_2, \ldots, \lambda_s\}$ and $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n\}$, where $\lambda_1 = 0$, $\boldsymbol{\xi}_1$ is given in (6);

2) Construct matrix $\boldsymbol{T} := [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n]$, and take $\boldsymbol{z} = \boldsymbol{T}^{-1}\boldsymbol{x}$;

3) Let $\boldsymbol{x}(0) = \boldsymbol{x}^{(0)} = [x_1(0), \ldots, x_n(0)]^{\mathrm{T}}$, and calculate $\boldsymbol{z}(0) = \boldsymbol{T}^{-1}\boldsymbol{x}(0) = [z_1(0), z_2(0), \ldots, z_n(0)]^{\mathrm{T}}$. Then $z_1(0)\boldsymbol{\xi}_1$ is the final state.

Remark 4. Theorem 1 is also applicable to m-dimensional systems. In fact, if $\boldsymbol{x} \in \mathbb{R}^m$, then system (5) will become $\dot{\boldsymbol{x}} = -[(\boldsymbol{D}+\boldsymbol{A}) \otimes \boldsymbol{I}_m]\boldsymbol{x}$. In this case, we can choose $\boldsymbol{z} = (\boldsymbol{T}^{-1} \otimes \boldsymbol{I}_m)\boldsymbol{x}$ to prove the m-dimensional version of Theorem 1.

3.2 The case that there exist competition and cooperation among agents

In this part, we consider the case where there exist both competition and cooperation among the agents, i.e., each agent competes with its neighbors and cooperates with its two-hop neighbors at the same time.

Consider system (1), and assume that its information topology is a directed bipartite graph which has a spanning tree, denoted by $\mathcal{G}$, and the adjacency matrix of $\mathcal{G}$ is $\boldsymbol{A}=[a_{ij}]_{n\times n}$. Motivated by [1,23,24], we design a control protocol as follows:

$$u_i=-\sum_{j\in\mathcal{N}_i}a_{ij}\Big[(x_i+x_j)+\sum_{k\in\mathcal{N}_j}a_{jk}(x_i-x_k)\Big],\quad i=1,2,\ldots,n, \tag{11}$$

where the term (x_i+x_j) stands for the competition between agents i and j, and the term $\sum_{k\in\mathcal{N}_j}a_{jk}(x_i-x_k)$ describes the cooperation of agent i with its two-hop neighbors.

Substituting (11) into system (1) yields

$$\begin{aligned}\dot{x}_i&=-\sum_{j\in\mathcal{N}_i}a_{ij}\Big[(x_j+x_i)+\sum_{k\in\mathcal{N}_j}a_{jk}(x_i-x_k)\Big]\\&=-\sum_{j\in\mathcal{N}_i}a_{ij}(x_j+x_i)-\sum_{j\in\mathcal{N}_i}a_{ij}\sum_{k\in\mathcal{N}_j}a_{jk}(x_i-x_k)\\&=-\sum_{j\in\mathcal{N}_i}a_{ij}(x_j+x_i)-\sum_{k\in\mathcal{N}_j}\sum_{j\in\mathcal{N}_i}a_{ij}a_{jk}(x_i-x_k),\quad i=1,2,\ldots,n.\end{aligned} \tag{12}$$

Let $\boldsymbol{A}^2:=[\tilde{a}_{ij}]_{n\times n}$, and $\boldsymbol{H}=\mathrm{diag}[h_1,h_2,\ldots,h_n]_{n\times n}$, where $h_i=\sum_{j=1}^n\tilde{a}_{ij}$. Then, system (12) can be rewritten as

$$\dot{\boldsymbol{x}}=-(\boldsymbol{D}+\boldsymbol{A})\boldsymbol{x}-(\boldsymbol{H}-\boldsymbol{A}^2)\boldsymbol{x}=-(\boldsymbol{D}+\boldsymbol{A}+\boldsymbol{H}-\boldsymbol{A}^2)\boldsymbol{x}, \tag{13}$$

where $\boldsymbol{x}=[x_1,x_2,\ldots,x_n]^{\mathrm{T}}$.

To study cluster synchronization of system (13), we give some lemmas first.

Lemma 2. Assume that $\mathcal{G}$ is a bipartite graph with n nodes $\{v_1,\ldots,v_n\}$, and $\boldsymbol{A}$ is its adjacency matrix. Then the matrix $\boldsymbol{A}^2$ is quasi-diagonal under a suitable labeling of nodes.

Proof. Since $\mathcal{G}$ is a bipartite graph, the matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A}:=\begin{bmatrix}\boldsymbol{0}_{r\times r} & \boldsymbol{A}_1\\ \boldsymbol{A}_2 & \boldsymbol{0}_{(n-r)\times(n-r)}\end{bmatrix}$$

under a suitable labeling of nodes. Thus, the matrix

$$\boldsymbol{A}^2=\begin{bmatrix}\boldsymbol{A}_1\boldsymbol{A}_2 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{A}_2\boldsymbol{A}_1\end{bmatrix},$$

which implies that the matrix $\boldsymbol{A}^2$ is quasi-diagonal under a proper labeling of nodes.

For convenience, in system (13) we denote the matrix $\boldsymbol{D}+\boldsymbol{A}+\boldsymbol{H}-\boldsymbol{A}^2:=\boldsymbol{B}=[b_{ij}]_{n\times n}$.

Lemma 3. Assume that $\mathcal{G}$ is a bipartite digraph with n nodes $\{v_1,\ldots,v_n\}$. Then, the matrix $\boldsymbol{B}$ defined as above has the following properties: a) $b_{ii}=\sum_{j\neq i}|b_{ij}|$; b) $\sum_{j=1}^r b_{ij}=\sum_{j=r+1}^n b_{ij}$; c) Furthermore, if $\mathcal{G}$ has a spanning tree, then $\mathrm{rank}(\boldsymbol{B})=n-1$, and $\mathrm{Re}(\lambda(\boldsymbol{B}))\geqslant 0$.

Proof. For the detailed proof, please refer to Appendix.

Now, we are ready to study cluster synchronization of system (13). Based on Lemma 1, Lemma 3 and Theorem 1, we have the following theorem.

Theorem 2. Consider multi-agent system (1) with the bipartite graph $\mathcal{G}$. Then, all the agents can realize cluster synchronization under control protocol (11), and the final states are determined by the initial conditions, that is, $\lim_{t\to\infty}\boldsymbol{x}(t)=z_1(0)\boldsymbol{\xi}_1$, where $z_1(0)$ and $\boldsymbol{\xi}_1$ are the same as those in Theorem 1.

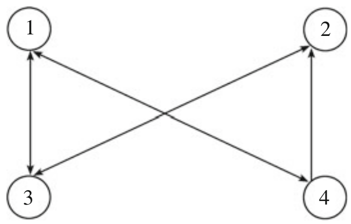

Figure 2 A bipartite graph.

Proof. Notice system (1) is changed into (13) under control protocol (11). Let $\boldsymbol{B} := \boldsymbol{D}+\boldsymbol{A}+\boldsymbol{H}-\boldsymbol{A}^2$. Then system (13) is expressed as $\dot{\boldsymbol{x}} = -\boldsymbol{B}\boldsymbol{x}$. From Lemma 3, $\mathrm{rank}(\boldsymbol{B}) = n-1$ and $\mathrm{Re}(\lambda(\boldsymbol{B})) \geqslant 0$. The next of the proof is similar to that of Theorem 2, and thus is omitted.

Remark 5. The cluster synchronization problem was first studied in [25], where several interesting results on the local synchronization of the reputation degrees were presented for Virtual Organizations. Compared with [25], our results are global ones, that is, all the agents reach cluster synchronization eventually, while the results of [25] can only guarantee some of the agents/entities to realize synchronization. Besides, this paper adopts a kind of control protocol of not only cooperation but also competition among the agents to realize cluster synchronization, and the used technique is different from that of [25].

4 Illustrative examples

In this section, we give two illustrative examples to show how to use the results obtained in this study in designing control protocols for cluster synchronization of multi-agent systems.

Example 1. Consider the following 4-agent system:

$$\dot{x}_i = u_i, \ i = 1, 2, 3, 4, \tag{14}$$

whose topology is shown as Figure 2 and initial condition is given as $x(0) = [x_1(0), x_2(0), x_3(0), x_4(0)]^{\mathrm{T}} = [8, -3, 6, -2]^{\mathrm{T}}$.

Now, we apply Theorem 1 to design a control protocol for cluster synchronization of the 4 agents only by using the competition among them. From Theorem 1, the desired protocol can be designed as

$$\begin{cases} u_1 = -2(x_1+x_3)-(x_1+x_4) = -3x_1-2x_3-x_4, \\ u_2 = -(x_2+x_3)-(x_2+x_4) = -2x_2-x_3-x_4, \\ u_3 = -2(x_3+x_2)-(x_3+x_1) = -x_1-2x_2-3x_3, \\ u_4 = -(x_4+x_1) = -x_1-x_4. \end{cases} \tag{15}$$

On the other hand, we readily have

$$\boldsymbol{D}+\boldsymbol{A} = \begin{pmatrix} 3 & 0 & 2 & 1 \\ 0 & 2 & 1 & 1 \\ 1 & 2 & 3 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}. \tag{16}$$

It is easy to check that the eigenvalues of the matrix $\boldsymbol{D}+\boldsymbol{A}$ are 0, 2, 2, 5 and the corresponding eigenvectors are $\boldsymbol{\xi}_1 = [1, 1, -1, -1]^{\mathrm{T}}, \ \boldsymbol{\xi}_2 = [0, 1, -1, 1]^{\mathrm{T}}, \ \boldsymbol{\xi}_3 = [1, 0, -1, 1]^{\mathrm{T}}, \ \boldsymbol{\xi}_4 = [8, 3, 7, 2]^{\mathrm{T}}$. According to Theorem 1 and Remark 3, multi-agent system (14) can realize cluster synchronization under the protocol (15) and the final states are

$$\mathrm{Row}_1(\boldsymbol{T}^{-1}\boldsymbol{x}(0))\boldsymbol{\xi}_1 = [0.6, 0.6, -0.6, -0.6]^{\mathrm{T}},$$

where $\boldsymbol{T} = [\boldsymbol{\xi}_1, \ \boldsymbol{\xi}_2, \ \boldsymbol{\xi}_3, \ \boldsymbol{\xi}_4]$, and $\mathrm{Row}_1(\boldsymbol{T}^{-1}\boldsymbol{x}(0))$ stands for the first component of $\boldsymbol{T}^{-1}\boldsymbol{x}(0)$.

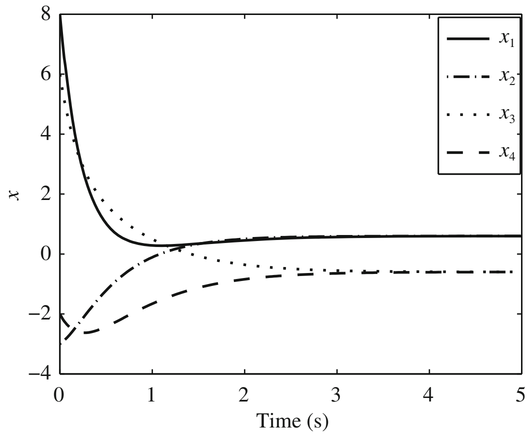

Figure 3 The state responses in Example 1.

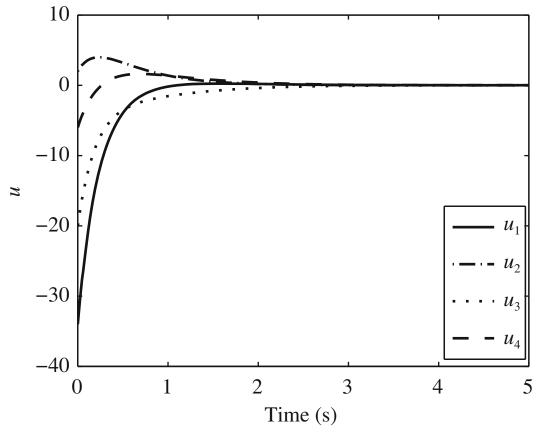

Figure 4 The controls used in Example 1.

To show the correctness of the above conclusion, we carry out some numerical simulations. The simulation results are shown in Figures 3 and 4, which are the swing cures of the states and control signals, respectively.

It can be seen from Figures 3 and 4 that the states of the 4 agents eventually realize cluster synchronization under the protocol (15) and the final states are the same as the theoretical analysis. Simulations show that our method is very effective in analyzing cluster synchronization of multi-agent system (14).

Example 2. Consider the multi-agent system (14), whose topology is shown as Figure 2 and initial condition is given as $\boldsymbol{x}(0) = [8, -3, 6, -2]^{\mathrm{T}}$. Assume that there are both competition and cooperation among the agents.

Now, we apply Theorem 2 to design a control protocol for cluster synchronization of the 4 agents. From Theorem 2, the desired protocol can be designed as

$$\begin{cases} u_1 = -2(x_1 + x_3) - (x_1 + x_4) - (x_1 - x_2) = -4x_1 + x_2 - 2x_3 - x_4, \\ u_2 = -(x_2 + x_3) - (x_2 + x_4) - (x_2 - x_1) = x_1 - 3x_2 - x_3 - x_4, \\ u_3 = -2(x_3 + x_2) - (x_3 + x_1) - 2(x_3 - x_4) = -x_1 - 2x_2 - 5x_3 + 2x_4, \\ u_4 = -(x_4 + x_1) - (x_4 - x_3) = -x_1 + x_3 - 2x_4, \end{cases} \tag{17}$$

which implies that the matrix $\boldsymbol{B}$ (see Theorem 2) is given by

$$\boldsymbol{B} = \begin{pmatrix} 4 & -1 & 2 & 1 \\ -1 & 3 & 1 & 1 \\ 1 & 2 & 5 & -2 \\ 1 & 0 & -1 & 2 \end{pmatrix}. \tag{18}$$

It is checked that the eigenvalues of $\boldsymbol{B}$ are $0, 3, 5, 6$ and the corresponding eigenvectors are $\boldsymbol{\xi}_1 = [1, 1, -1, -1]^{\mathrm{T}}$, $\boldsymbol{\xi}_2 = [2, 3, -1, 3]^{\mathrm{T}}$, $\boldsymbol{\xi}_3 = [4, -1, 1, 1]^{\mathrm{T}}$, $\boldsymbol{\xi}_4 = [1, 0, 1, 0]^{\mathrm{T}}$. According to Theorem 2, the multi-agent system (14) can realize cluster synchronization under the protocol (17) and the final states are

$$\mathrm{Row}_1(\boldsymbol{T}^{-1}\boldsymbol{x}(0))\boldsymbol{\xi}_1 = [0.6, 0.6, -0.6, -0.6]^{\mathrm{T}},$$

where $\boldsymbol{T} = [\boldsymbol{\xi}_1,\ \boldsymbol{\xi}_2,\ \boldsymbol{\xi}_3,\ \boldsymbol{\xi}_4]$, and $\mathrm{Row}_1(\boldsymbol{T}^{-1}\boldsymbol{x}(0))$ stands for the first component of $\boldsymbol{T}^{-1}\boldsymbol{x}(0)$.

To show the correctness of the above conclusion, we carry out some numerical simulations. The simulation results are shown in Figures 5 and 6, which are the swing cures of states and control signals, respectively.

It can be observed from Figure 5 and 6 that the states of the 4 agents eventually realize cluster synchronization under the protocol (17), and the final states coincide with the theoretical analysis. Moreover,

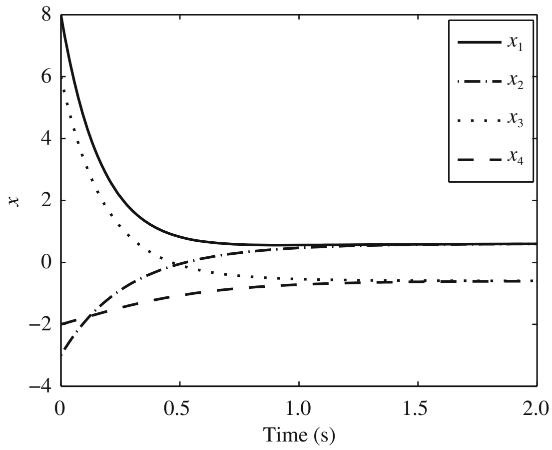

Figure 5 The state responses in Example 2.

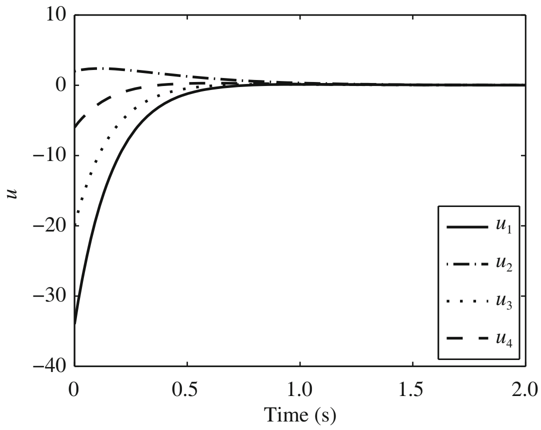

Figure 6 The controls used in Example 2.

comparing Figures 5 and 6 with Figures 3 and 4, we observe that the 4 agents reach cluster synchronization much faster than in Example 1. Simulations show that the protocol (17) is very effective in analyzing the cluster synchronization of the multi-agent system.

Remark 6. From the study of these two examples, we know that protocol (11) can make agents reach cluster synchronization faster than protocol (3). Protocol (3) needs less information of agents but has a relatively slow convergence, while protocol (11) has a faster convergence but needs more information of agents.

5 Conclusion

In this paper, we have investigated cluster synchronization of a class of multi-agent systems with a directed bipartite graph topology, and presented a number of new results by using the neighbor's rules. For the case where there is only competition among agents of different clusters, a linear control protocol was designed, and a new method was presented to determine the final state with the initial conditions based on state-space decomposition. When there are both competition and cooperation among the agents, a control protocol was designed based on the information of neighbors and that of two-hop neighbors. It has been shown that such a protocol could make the multi-agent system achieve the cluster synchronization faster. The study of illustrative examples with simulations has shown that our results as well as designed control protocols work very well in studying cluster synchronization of this class of multi-agent systems.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 61074068, 61174036, 61034007), Research Fund for the Taishan Scholar Project of Shandong Province of China, Natural Science Foundation of Shandong Province (Grant No. ZR2010FM013).

References

1 Fax J A, Murray R M. Information flow and cooperative control of vehicle formations. IEEE Trans Autom Control, 2004, 49: 1465–1476

2 Hao H, Barooah P, Mahta P G. Stability margin scaling laws for distributed formation control as a function of network structure. IEEE Trans Autom Control, 2011, 56: 923–929

3 Ni W, Cheng D Z. Leader-following consensus of multi-agent systems under fixed and switching topologies. Syst Control Lett, 2010, 59: 209–217

4 Wang Y, Zhang C, Liu Z. A matrix approach to graph maximum stable set and coloring problems with application to multi-agent systems. Automatica, 2012, 48: 1227–1236

5 Hong Y G, Hu J P, Gao L X. Tracking control for multi-agent consensus with an active leader and variable topology. Automatica, 2006, 42: 1177–1182
6 Tanner H G, Jadbabaie A, Pappas G J. Flocking in fixed and switching networks. IEEE Trans Autom Control, 2007, 52: 863–868
7 Su H S, Wang X F, Lin Z L. Flocking of multi-agents with a virtual leader. IEEE Trans Autom Control, 2009, 54: 293–307
8 Saber R O. Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans Autom Control, 2006, 51: 401–420
9 Lee D, Spong M W. Stable flocking of multiple inertial agents on balanced graphs. IEEE Trans Autom Control, 2007, 52: 1469–1475
10 Li T, Zhang J F. Asymptotically optimal decentralize control for large population stochastic multi-agent systems. IEEE Trans Autom Control, 2008, 53: 1643–1660
11 Zhang Y, Tian Y P. Consentability and protocol design of multi-agent systems with stochastic switching topology. Automatica, 2009, 45: 1195–1201
12 Zhang Y, Tian Y P. Consensus of data-sampled multi-agent systems with random communication delay and packet loss. IEEE Trans Autom Control, 2010, 55: 939–943
13 Kale N V, Salapaka S M. Maximum entropy principle based algorithm for simultaneous resource location and multi-hop routing in multi-agent networks. IEEE Trans Mob Comput, 2011, 1–12
14 Sun Y G, Wang L. Consensus of multi-agent systems in directed networks with nonuniform time-varying delays. IEEE Trans Autom Control, 2009, 54: 1607–1613
15 Ren W, Beard R W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans Autom Control, 2005, 50: 655–661
16 Saber R O. Distributed Kalman filter with embedded consensus filter. In: IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC'05), Seville, 2005. 8179–8184
17 Yang T, Sun Y F. The reconstruction of gene regulatory network based on multi-agent systems by fusing multiple data sources. In: IEEE International Conference on Computer Science and Automation Engineering (CSAE), Shanghai, 2011. 126–130
18 Saber R O, Shamma J S. Consensus filters for sensor networks and distributed sensor fusion. In: IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC'05), Seville, 2005. 6698–6703
19 Ren W, Cao Y C. Distributed Coordination of Multi-agent Networks. London: Springer-Verlag, 2010. 36–41
20 Xia W G, Cao M. Clustering in diffusively coupled networks. Automatica, 2011, 47: 2395–2405
21 Ma C Q, Zhang J F. Necessary and sufficient conditions for consensusability of linear multi-agent systems. IEEE Trans Autom Control, 2010, 55: 1263–1268
22 Yu J Y, Wang L. Group consensus in multi-agent systems with switching topologies and communication delays. Syst Control Lett, 2010, 59: 340–348
23 Xiao Y N, Chen L S. Stabilizing effect of cannibalism on structured competitive system (in Chinese). Acta Math Sci, 2002, 32: 210–216
24 Mao K, Li R H. The stability analysis of population competition model (in Chinese). J Biomath, 1999, 14: 288–292
25 Yu W W, Cao J D, Chen G R, et al. Local synchronization of a complex network model. IEEE Trans Syst Man Cybern Part B-Cybern, 2009, 39: 230–241

Appendix The proofs of Lemmas 1 and 3

(1) The proof of Lemma 1.

Since $\mathcal{G}$ is a bipartite graph and has a spanning tree, we assume that $\{U,\ V\}$ is a partition of the nodes, where $U=\{v_1,\ldots,v_r\}$ and $V=\{v_{r+1},\ldots,v_n\}$. Then, the adjacency matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A}:=\left[\begin{array}{cc}\mathbf{0}_{r\times r} & \boldsymbol{A}_1\\ \boldsymbol{A}_2 & \mathbf{0}_{(n-r)\times(n-r)}\end{array}\right],$$

and the last column of matrix $\boldsymbol{A}_1$ is not equal to zero. Without loss of generality, we denote the vertex v_n by the root vertex.

Letting

$$\boldsymbol{\xi}_1=[\underbrace{1,\ldots,1}_{r},\underbrace{-1,\ldots,-1}_{n-r}]^{\mathrm{T}},$$

according to the definitions of matrices $\boldsymbol{D}$ and $\boldsymbol{A}$, we have $(\boldsymbol{D}+\boldsymbol{A})\boldsymbol{\xi}_1=0$.

Next, we prove that there exists an $(n-1)$th order block matrix $\boldsymbol{M}$ of matrix $\boldsymbol{D}+\boldsymbol{A}$ such that $\text{rank}(\boldsymbol{M}) = n-1$. Let

$$\boldsymbol{M} := \begin{bmatrix} \boldsymbol{D}^{(1)}_{r\times r} & \tilde{\boldsymbol{A}}_1 \\ \tilde{\boldsymbol{A}}_2 & \boldsymbol{D}^{(2)}_{(n-r-1)\times(n-r-1)} \end{bmatrix}$$

be the $(n-1)$th order principal minor matrix of matrix $\boldsymbol{D}+\boldsymbol{A}$, where $\tilde{\boldsymbol{A}}_1$ is a matrix obtained from $\boldsymbol{A}_1$ by removing the last column, $\tilde{\boldsymbol{A}}_2$ is obtained from $\boldsymbol{A}_2$ by removing the last row, and $\boldsymbol{D}^{(1)}$ and $\boldsymbol{D}^{(2)}$ are the corresponding parts in $\boldsymbol{D}$. Since $\mathcal{G}$ has a spanning tree, and v_n is the root node, there is not a zero row in the matrix $\boldsymbol{M}$; thus, for any $1 \leqslant i \leqslant n-1$, $d_i + \sum_{j=1}^{n-1} a_{ij} \neq 0$. On the other hand, since $\mathcal{G}$ is a bipartite graph, there exists at least $1 \leqslant i \leqslant r$ such that $d_i > \sum_{j=r+1}^{n-1} a_{ij}$, and $d_i = \sum_{j=1}^{r} a_{ij}$ holds for $r+1 \leqslant i \leqslant n-1$. Let λ be any eigenvalue of $\boldsymbol{M}$, and $\boldsymbol{y} = [y_1, \ldots, y_{n-1}]^{\mathrm{T}}$ be the corresponding eigenvector, that is, $\boldsymbol{M}\boldsymbol{y} = \lambda\boldsymbol{y}$. Letting $|y_{r_0}| = \max_{1\leqslant i\leqslant n-1}\{|y_i|\}$, we now prove $\lambda \neq 0$ by the following three cases:

Case 1: $|y_1| = |y_2| = \cdots = |y_{n-1}|$. In this case, $\lambda \neq 0$. In fact, if $\lambda = 0$, according the definitions of $\boldsymbol{D}+\boldsymbol{A}$ and $\boldsymbol{M}$, for $r+1 \leqslant r_0 \leqslant n-1$, $\boldsymbol{M}\boldsymbol{y} = \lambda\boldsymbol{y} = 0$, which implies that

$$\boldsymbol{y} = \pm[\underbrace{1,\ldots,1}_{r}, \underbrace{-1,\ldots,-1}_{n-1-r}]^{\mathrm{T}}.$$

However, from the analysis as above, there exists at least $1 \leqslant r_0 \leqslant r$ such that $d_{r_0} > \sum_{j=r+1}^{n-1} a_{r_0 j}$, which implies that $\boldsymbol{M}\boldsymbol{y} \neq 0$, and this is a contradiction. Thus, $\lambda \neq 0$.

Case 2: $1 \leqslant r_0 \leqslant r$. In this case, it is easy to obtain

$$(\lambda - d_{r_0})y_{r_0} = \sum_{j=r+1}^{n-1} a_{r_0 j} y_j. \tag{A1}$$

Since $d_{r_0} + \sum_{j=1}^{n-1} a_{r_0 j} \neq 0$, if $\sum_{j=1}^{n-1} a_{r_0 j} = \sum_{j=r+1}^{n-1} a_{r_0 j} = 0$, then $\lambda = d_{r_0} \neq 0$; if not, (A1) implies that

$$|\lambda - d_{r_0}||y_{r_0}| = \left|\sum_{j=r+1}^{n-1} a_{r_0 j} y_j\right| \leqslant \left|\sum_{j=r+1}^{n-1} a_{r_0 j}\right||y_j| < \left|\sum_{j=r+1}^{n-1} a_{r_0 j}\right||y_{r_0}| \leqslant d_{r_0}|y_{r_0}|.$$

Thus, we have $|\lambda - d_{r_0}| < d_{r_0}$, which implies that $\lambda \neq 0$.

Case 3: $r+1 \leqslant r_0 \leqslant n-1$. In this case, we have

$$(\lambda - d_{r_0})y_{r_0} = \sum_{j=1}^{r} a_{r_0 j} y_j.$$

From Case 2, it is easy to know that if $\sum_{j=1}^{r} a_{r_0 j} = 0$, then $\lambda = d_{r_0} \neq 0$; if not,

$$|\lambda - d_{r_0}||y_{r_0}| = \left|\sum_{j=1}^{r} a_{r_0 j} y_j\right| \leqslant \left|\sum_{j=1}^{r} a_{r_0 j}\right||y_j| < \left|\sum_{j=1}^{r} a_{r_0 j}\right||y_{r_0}| = d_{r_0}|y_{r_0}|,$$

which implies that $\lambda \neq 0$.

Summing up these three cases, we have $\text{rank}(\boldsymbol{M}) = n-1$. Therefore, we have $\text{rank}(\boldsymbol{D}+\boldsymbol{A}) = n-1$. From Gerschgorin theorem[1)] , each non-zero eigenvalue of matrix $\boldsymbol{D}+\boldsymbol{A}$ has a positive real part.

(2) The proof of Lemma 3.

a) Since $\mathcal{G}$ is a bipartite graph, without loss of generality, we assume that $\{U,\ V\}$ is a partition of the nodes, where $U = \{v_1, \ldots, v_r\}$ and $V = \{v_{r+1}, \ldots, v_n\}$. According to Lemma 1, Lemma 2 and the definitions of $\boldsymbol{D}$ and $\boldsymbol{H}$, it is easy to know that

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{D_1} + \boldsymbol{H_1} - \boldsymbol{A_1}\boldsymbol{A_2} & \boldsymbol{A}_1 \\ \boldsymbol{A}_2 & \boldsymbol{D_2} + \boldsymbol{H_2} - \boldsymbol{A_2}\boldsymbol{A_1} \end{bmatrix} := \begin{bmatrix} \boldsymbol{B}^{(1)}_{r\times r} & \boldsymbol{A}_1 \\ \boldsymbol{A}_2 & \boldsymbol{B}^{(2)}_{(n-r)\times(n-r)} \end{bmatrix},$$

where $\boldsymbol{D} = \text{diag}[\boldsymbol{D}_1, \boldsymbol{D}_2]$, $\boldsymbol{H} = \text{diag}[\boldsymbol{H}_1, \boldsymbol{H}_2]$, $\boldsymbol{B}^{(1)} = \boldsymbol{D_1} + \boldsymbol{H_1} - \boldsymbol{A_1}\boldsymbol{A_2}$ and $\boldsymbol{B}^{(2)} = \boldsymbol{D_2} + \boldsymbol{H_2} - \boldsymbol{A_2}\boldsymbol{A_1}$. Then, we consider the following two cases:

Case A_1: $1 \leqslant i \leqslant r$. In this case, $b_{ii} = d_{ii} + h_{ii} - \tilde{a}_{ii} = \sum_{j\neq i}^{r} \tilde{a}_{ij} + \sum_{j=r+1}^{n} a_{ij} = \sum_{j\neq i} |b_{ij}|$.

Case A_2: $r+1 \leqslant i \leqslant n$. In this case, $b_{ii} = \sum_{j=1}^{r} a_{ij} + \sum_{j\geqslant r+1, j\neq i}^{n} \tilde{a}_{ij} = \sum_{j\neq i} |b_{ij}|$.

1) Horn R, Johnson C. Matrix Analysis. New York: Cambridge University Press, 1985

Summing up these two cases, we have $b_{ii} = \sum_{j\neq i} |b_{ij}|$.

b) Based on the definition of matrix $\boldsymbol{B}$ and a), we consider the following cases:

Case B_1: $1 \leqslant i \leqslant r$. In this case, $b_{ii} = \sum_{j\neq i}^{r} \tilde{a}_{ij} + \sum_{j=r+1}^{n} a_{ij} = -\sum_{j\neq i}^{r} b_{ij} + \sum_{j=r+1}^{n} b_{ij}$.

Case B_2: $r+1 \leqslant i \leqslant n$. In this case, $b_{ii} = \sum_{j=1}^{r} a_{ij} + \sum_{j=r+1,j\neq i}^{n} \tilde{a}_{ij} = \sum_{j=1}^{r} b_{ij} - \sum_{j=r+1,j\neq i}^{n} b_{ij}$.

Summing up these two cases, we have $\sum_{j=1}^{r} b_{ij} = \sum_{j=r+1}^{n} b_{ij}$.

c) From b), it can be seen that $\boldsymbol{B}\boldsymbol{\xi}_1 = 0$, where $\boldsymbol{\xi}_1$ is given in (6). Thus, $\mathrm{rank}(\boldsymbol{B}) \leqslant n-1$. Now, we prove that there exists an $(n-1)$-th order block matrix $\boldsymbol{N}$ of matrix $\boldsymbol{B}$ such that $\mathrm{rank}(\boldsymbol{N}) = n-1$.

Since $\mathcal{G}$ has a spanning tree, without loss of generality, we let the node v_n be the root node, and choose $\boldsymbol{N}$ as the $(n-1)$-th order principal minor matrix of matrix $\boldsymbol{B}$. Then $\boldsymbol{N}$ is expressed as

$$\boldsymbol{N} := \begin{bmatrix} \boldsymbol{B}^{(1)}_{r\times r} & \tilde{\boldsymbol{A}}_1 \\ \tilde{\boldsymbol{A}}_2 & \tilde{\boldsymbol{B}}^{(2)}_{(n-r-1)\times(n-r-1)} \end{bmatrix},$$

where $\tilde{\boldsymbol{A}}_1, \tilde{\boldsymbol{A}}_2$ are the same as in Lemma 1, and $\tilde{\boldsymbol{B}}^{(2)}$ is the corresponding part in $\boldsymbol{B}^{(2)}$. By Lemma 1, we have $\sum_{j=1}^{n-1} b_{ij} \neq 0$, for any $1 \leqslant i \leqslant n-1$. On the other hand, according to the properties of matrix $\boldsymbol{B}$, there at least exists $1 \leqslant i \leqslant n-1$ such that $b_{ii} > \sum_{j\neq i}^{n-1} |b_{ij}|$.

Let μ be any eigenvalue of the matrix $\boldsymbol{N}$, and let $\boldsymbol{y} = [y_1, \ldots, y_{n-1}]^{\mathrm{T}}$ be the corresponding eigenvector, that is, $\boldsymbol{N}\boldsymbol{y} = \mu\boldsymbol{y}$. Letting $|y_{r_0}| = \max\limits_{1\leqslant i\leqslant n-1}\{|y_i|\}$, we now prove $\mu \neq 0$ by the following two cases:

Case C_1: $|y_1| = |y_2| = \cdots = |y_{n-1}|$. In this case, according to the analysis above, for the matrix $\boldsymbol{N}$, there at least exists $1 \leqslant i \leqslant n-1$ such that $b_{ii} > \sum_{j\neq i}^{n-1} |b_{ij}|$, that is, $b_{ii}|y_i| > \sum_{j\neq i}^{n-1} |b_{ij}||y_j|$, which is a contradiction with $\boldsymbol{N}\boldsymbol{y} = 0$. Thus, $\mu \neq 0$.

Case C_2: There exist $1 \leqslant i, j \leqslant n-1$ such that $|y_i| \neq |y_j|$. In this case, it is easy to obtain

$$(\mu - b_{r_0r_0})y_{r_0} = \sum_{j\neq r_0}^{n-1} b_{r_0j}y_j. \tag{A2}$$

Since $\sum_{j=1}^{n-1} b_{ij} \neq 0$, if $\sum_{j\neq r_0}^{n-1} b_{r_0j} = 0$, $\mu = b_{r_0r_0} > 0$; if not, (A2) implies that

$$|\mu - b_{r_0r_0}||y_{r_0}| = \left|\sum_{j\neq r_0}^{n-1} b_{r_0j}y_j\right| \leqslant \left|\sum_{j\neq r_0}^{n-1} b_{r_0j}\right||y_j| < \left|\sum_{j\neq r_0}^{n-1} b_{r_0j}\right||y_{r_0}| \leqslant b_{r_0r_0}|y_{r_0}|.$$

Then we have $|\mu - b_{r_0r_0}| < b_{r_0r_0}$, which implies that $\mu \neq 0$.

Summing up these two cases, we have $\mathrm{rank}(\boldsymbol{N}) = n-1$. Therefore, we have $\mathrm{rank}(\boldsymbol{B}) = n-1$. On the other hand, from Gerschgorin theorem, each non-zero eigenvalue of matrix $\boldsymbol{B}$ has a positive real part.