



# 合作竞争网络下的多智能体系统链路故障检测

邵晋梁<sup>†</sup>, 石磊<sup>†</sup>, 李彤, 张希琳<sup>\*</sup>

电子科技大学自动化工程学院, 成都 611731

<sup>\*</sup> 通信作者. E-mail: xilinzhang@uestc.edu.cn

<sup>†</sup> 同等贡献

收稿日期: 2021-04-08; 修回日期: 2021-06-06; 接受日期: 2021-07-14; 网络出版日期: 2022-08-03

国家自然科学基金 (批准号: U1830207, 61772003, 61903066, 62103080)、四川省科技计划项目 (批准号: 2021YFH0042)、博士后创新人才支持计划 (批准号: BX2021056) 和博士后科学基金 (批准号: 2021M700696) 资助项目

**摘要** 本文提出分布式符号簇一致协议, 研究合作竞争网络下的多智能体系统链路故障检测问题. 首先, 应用符号图理论和非负矩阵理论的相关知识分析智能体在符号簇一致性协议下的渐近收敛性和最终收敛状态. 其次, 通过观察链路发生故障前后智能体收敛状态的变化, 给出精确判断故障链路的方法. 该方法同时适用于单链路故障和多链路故障的情形. 最后, 通过数值模拟验证该方法的有效性.

**关键词** 多智能体系统, 合作竞争网络, 链路故障检测, 一致性, 分布式协议

## 1 引言

受自然界中生物群体通过局部协作而获得群体优势的启发, 如鹿群编队迁徙、鱼群协同巡游、狼群协同围捕、蚁群协同搬运等, 多智能体系统 (multi-agent system, MAS) 协同控制问题已成为当前国际上的一个热点问题, 涉及控制、数学、物理、生物、计算机、通信和人工智能等多个学科的内容. 研究者通过设计相对简单的网络结构和控制机制, 使得个体的行为在局部相互作用下在宏观层面上产生复杂的涌现现象. 对此进行建模和分析, 不仅可以揭示生物集群行为的内在规律, 而且在许多工程领域中有着广泛的应用, 例如社会网络观点动力学、多移动机器人系统、无人驾驶机/车群系统、智能电网、人造卫星簇等复杂系统的智能自主协同控制以及通讯网络当中的拥塞控制等. 目前关于多智能体系统协同控制的研究主要涉及一致性控制<sup>[1~3]</sup>、跟踪控制<sup>[4,5]</sup>、包围控制<sup>[6,7]</sup>、蜂拥控制<sup>[8~10]</sup>等问题.

现有关于多智能体系统协同控制的研究成果大部分都是建立在完全合作网络的基础上的, 即智能体之间只存在合作关系. 也就是说, 在通信拓扑图中, 描述个体之间信息交流的边的权值都假设为非负. 在现实世界中, 网络中个体之间的合作和竞争通常是共同存在的. 例如, 由于生存资源的有限, 生物系统中物种内部和物种之间就会存在合作和竞争<sup>[11]</sup>. 当集群机器人进行多目标搜索时, 通过任务分工形成多个子群, 各子群分别针对一个意向目标协同搜索, 智能体之间和子群之间就会存在合作协同

**引用格式:** 邵晋梁, 石磊, 李彤, 等. 合作竞争网络下的多智能体系统链路故障检测. 中国科学: 信息科学, 2022, 52: 1500–1512, doi: 10.1360/SSI-2021-0120  
Shao J L, Shi L, Li T, et al. Link failure detection for multi-agent systems on cooperation-competition networks (in Chinese). *Sci Sin Inform*, 2022, 52: 1500–1512, doi: 10.1360/SSI-2021-0120

和竞争协同<sup>[12]</sup>. 网络中个体之间的竞争关系对多智能体系统动力学行为的演化有着重大影响. 因此, 在多智能体系统协作控制问题中考虑个体之间的竞争关系有着重要的理论意义和实际应用价值. 近些年, 合作竞争网络上的多智能体二分一致性控制<sup>[13,14]</sup>、二分跟踪控制<sup>[15,16]</sup>、二分包围控制<sup>[17,18]</sup>、二分蜂拥控制<sup>[19,20]</sup>等协同控制问题引起了众多研究者的关注.

许多使用多智能体系统表示的物理世界和网络进程都是在恶劣和不断变化的环境中运行的, 这使得智能体之间的通信链路经常会因为受到外部干扰而中断. 在这些实际的网络系统中, 一个小故障或错误可能导致灾难性的局面. 例如, 在因特网进行数据传输时, 链路的失效会影响整个网络系统的性能; 在通过电网网络、公路网、铁路网等现实世界网络进行全球范围内的资源转移时, 链路的失效会导致资源利用不足. 因此, 检测网络中的不规则或链路故障(如通信线路的短路、断路以及线路物理性质引起的通信信号衰减或失真)可以有效地改善受影响系统的性能, 是实现多智能体系统协同控制的基本要求之一. 目前为止, 已有一些工作对完全合作网络下多智能体系统实施过程中的链路故障检测进行了研究. 对于一个典型的线性一致性模型, 文献[21]考虑通过一个监视器来检测关键链路的故障, 该监视器对该过程进行噪声局部测量. 文献[22]引入了多智能体系统中链路相对于智能体的可检测性概念, 其中网络中的链路故障根据其可检测性来描述, 这取决于链路故障如何影响系统中特定节点的响应, 以及是否可以通过观察该节点的响应来确定链路故障. 分析结果为网络信息流在有向图中一组边的可检测性提供了充分的代数图条件. Dhal 等<sup>[23]</sup>中定义了一种从噪声测量中检测链路故障的方法. 文献[24]提出了一致性扩散协议用以检测多智能体系统演化过程中的链路故障. Parlangei 等<sup>[25]</sup>对离散时间多智能体一致性网络中链路突然发生故障的问题进行了研究, 给出了检测故障链路的有效方法.

值得注意的是, 现有的工作主要针对完全合作网络上的多智能体系统链路故障检测问题展开研究. 对于合作竞争网络上的多智能体系统链路故障检测问题, 目前为止还未有相应的理论研究结果. 由于智能体之间的竞争关系, 整个网络可能会涌现出二分一致性动力学. 因此现有文献[21~25]所采用的一致性轨迹链路故障判定方法对于合作竞争网络是不适用的. 受此启发, 本文的主要贡献是提出基于一种新颖的基于分布式符号簇一致性协议的链路故障检测方法, 用以判断合作竞争网络上多智能体系统协同控制中的链路故障. 所提方法的优点在于, 可以利用收敛状态偏移来检测网络中的失效链路. 也就是说, 只需要观察发生故障前后智能体最终收敛状态的变化, 即可准确找出发生故障的链路.

本文的结构安排如下: 第 2 节给出了一些符号定义和预备知识; 第 3 节介绍了符号簇一致性协议并对其收敛性进行了详细分析; 第 4 节基于符号簇一致性协议给出了判断链路故障的理论结果; 第 5 节通过数值仿真验证了理论结果的正确性; 最后, 第 6 节总结了本文所研究的内容.

## 2 预备知识与符号定义

一个无向符号图被定义为  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , 其中  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  和  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  分别表示节点集和边集, “ $\times$ ”表示笛卡尔积(Cartesian product). 连接节点  $v_j$  和  $v_i$  的边被表示为  $(v_j, v_i)$  或  $(v_i, v_j)$ . 每条边都与一个正号“+”或负号“-”关联.  $(v_j, v_i) \in \mathcal{E}$  当且仅当节点  $v_j$  是节点  $v_i$  的邻居.  $\mathcal{N}_i$  表示节点  $v_i$  的邻居集. 令  $\mathcal{A} = [a_{ij}]$  表示邻接矩阵, 其中  $a_{ij} > 0$  ( $a_{ij} < 0$ ) 当且仅当边  $(v_j, v_i)$  的符号为“+” (“-”), 否则  $a_{ij} = 0$ .  $\mathcal{L} = [l_{ij}]$  表示符号拉普拉斯(Laplace)矩阵, 其中

$$l_{ij} = -a_{ij}, \quad \forall j \neq i,$$

$$l_{ii} = \sum_{j \in \mathcal{N}_i} |a_{ij}|, \quad i = 1, 2, \dots, n.$$

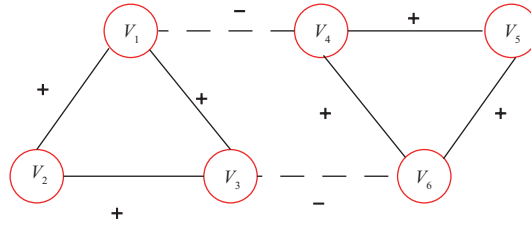


图 1 (网络版彩图) 一个结构平衡符号图

Figure 1 (Color online) A structurally balanced signed graph

从节点  $v_{i_1}$  到  $v_{i_z}$  的一条路径是一个有限的非空边序列, 表示为  $(v_{i_1}, v_{i_2})(v_{i_2}, v_{i_3}) \cdots (v_{i_{z-1}}, v_{i_z})$ , 其中  $v_{i_1}, v_{i_2}, \dots, v_{i_z} \in \mathcal{V}$  互不相同. 节点  $v_{i_1}$  到  $v_{i_z}$  的距离用  $d_{v_{i_1} \rightarrow v_{i_z}}$  表示, 它等于节点  $v_{i_1}$  与  $v_{i_z}$  之间最短路径的边的数量.

符号图的一个著名概念是结构平衡 (structural balance), 它的定义如下所示.

**定义1** ([26]) 如果存在互不相交的节点集  $\mathcal{V}_1, \mathcal{V}_2$  使得 (1)  $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}, \mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ ; (2) 对于任意  $(v_j, v_i) \in \mathcal{E}, a_{ij} > 0$  当且仅当  $v_i$  和  $v_j$  属于相同节点集, 否则  $a_{ij} < 0$ ; 然后称符号图  $\mathcal{G}$  是结构平衡的.

图 1 给出了一个包含 6 个节点的结构平衡符号图, 其中节点被划分为两个子集  $\mathcal{V}_1 = \{v_1, v_2, v_3\}$  和  $\mathcal{V}_2 = \{v_4, v_5, v_6\}$ . 属于同一子集的两个相邻节点之间具有正权重 (表示合作关系), 属于不同子集的两个相邻节点之间具有负权重 (表示竞争关系).

为了方便, 分别用  $\mathbb{N}, \mathbb{R}$  和  $\mathbb{R}^{n \times m}$  表示自然数集、实数集、 $n \times m$  维实矩阵集;  $I_n \in \mathbb{R}^{n \times n}$  表示单位矩阵;  $\mathbf{1}_n \in \mathbb{R}^{n \times 1}$  是一个所有元素都为 1 的列向量;  $\mathbf{0}$  表示一个所有元素都为 0 的矩阵;  $\otimes$  是 Kronecker 积;  $\text{diag}\{a_1, a_2, \dots, a_n\}$  表示一个对角元素依次为  $a_1, a_2, \dots, a_n$  的对角矩阵.  $\rho(Q)$  表示矩阵  $Q$  的谱半径.  $Q(i : j, p : s)$  是一个由矩阵  $Q$  的第  $i$  行到第  $j$  行、第  $p$  列到第  $s$  列的所有元素构成的一个块矩阵, 其中  $i \leq j, p \leq s$ .  $\prod_{w=1}^s Q_w = Q_s Q_{s-1} \cdots Q_1$  表示矩阵的左乘积.

### 3 符号簇一致性协议

考虑由  $n$  个智能体构成的系统. 假设所有智能体可以被划分为两个子集  $\mathcal{V}_1$  和  $\mathcal{V}_2$ , 属于同一子集内部的相邻智能体之间相互合作, 属于不同子集的相邻智能体之间彼此竞争. 所有智能体之间的信息交互用一个结构平衡符号图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  来描述, 其中  $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ , 两个智能体之间相互合作 (竞争) 当且仅当  $a_{ij} > 0$  ( $a_{ij} < 0$ ). 不失一般性, 假设  $\mathcal{V}_1 = \{v_1, v_2, \dots, v_m\}, \mathcal{V}_2 = \{v_{m+1}, v_{m+2}, \dots, v_n\}$ . 根据子集  $\mathcal{V}_1$  和  $\mathcal{V}_2$  的划分, 邻接矩阵  $\mathcal{A} = [a_{ij}]$  具有以下形式:

$$\mathcal{A} = \begin{pmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{pmatrix},$$

其中矩阵  $\mathcal{A}_{11} \geq 0$  和  $\mathcal{A}_{22} \geq 0$  分别被用来描述子集  $\mathcal{V}_1$  和  $\mathcal{V}_2$  内部的信息交互, 矩阵  $\mathcal{A}_{12} = \mathcal{A}_{21}^T \leq 0$  被用来描述两个子集之间的信息交互.

假设每个智能体  $v_i$  在  $k\tau$  时刻关联一个状态  $x_i[k] \in \mathbb{R}$ , 其中  $\tau$  是更新时间步长. 在文献 [27] 中被广泛研究的 Altifini 协议 (也称为符号一致性协议) 被描述为

$$x_i[k+1] = x_i[k] + \tau \sum_{v_j \in \mathcal{N}_i} (\text{sgn}(a_{ij})x_j[k] - x_i[k]). \tag{1}$$

在一个连通的结构平衡网络上实施协议 (1), 整个智能体系统会实现二分一致性, 即

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_j[k] - x_i[k]\| &= 0, \text{ 如果 } v_i, v_j \text{ 属于相同的节点子集,} \\ \lim_{k \rightarrow \infty} \|x_j[k] + x_i[k]\| &= 0, \text{ 如果 } v_i, v_j \text{ 属于不同的节点子集.} \end{aligned} \quad (2)$$

本文提出了一个更具普遍性的符号簇一致性协议:

$$x_i[k+1] = x_i[k] + \tau \sum_{v_j \in \mathcal{N}_i} \left( \text{sgn}(a_{ij}) \frac{x_j(k)}{c_j} - \frac{x_i(k)}{c_i} \right), \quad (3)$$

其中  $c_i > 0, i = 1, 2, \dots, n$  是正实数. 在一个连通的结构平衡网络上实施协议 (3), 那么整个多智能体系统会达到簇二分一致性, 即

$$\begin{aligned} \lim_{k \rightarrow \infty} \left\| \frac{x_j[k]}{c_j} - \frac{x_i[k]}{c_i} \right\| &= 0, \text{ 如果 } v_i, v_j \text{ 属于相同的节点子集,} \\ \lim_{k \rightarrow \infty} \left\| \frac{x_j[k]}{c_j} + \frac{x_i[k]}{c_i} \right\| &= 0, \text{ 如果 } v_i, v_j \text{ 属于不同的节点子集,} \end{aligned} \quad (4)$$

值得注意的是当  $c_i = 1, i = 1, 2, \dots, n$  时, 符号比例一致性协议 (3) 就转化为符号一致性协议 (1). 也就是说, 协议 (1) 可以看作是协议 (3) 的一种特殊情况.

**注释1** 符号一致性协议 (1) 作为合作竞争网络下多智能体系统的基本演化规律, 在最近的文献中得到了广泛的研究. 在符号一致性协议实施过程中, 由于传输线路的频带有限、智能体处理配置失效和网络攻击等因素, 网络中的通信链路可能会发生故障从而失效. 在文献 [24] 中, 作者通过观察链路故障发生前后智能体最终收敛状态的方法来检测失效链路. 然而这样的方法对于协议 (1) 来说并不适用. 这是因为对于给定的初始状态, 如果通信链路发生故障后网络是连通的, 那么所有智能体的状态在链路故障发生前后总是会收敛到完全相同的二分一致性状态, 也就是说我们不能从故障发生前后智能体的收敛状态来检测失效链路. 本文提出了一个新的符号簇一致性协议 (3). 协议 (3) 不仅是协议 (1) 的一个推广, 而且如果在协议 (3) 实施过程中发生了链路故障, 无论发生链路故障后网络是否连通, 我们都可以通过设置参数  $c_i > 0, i = 1, 2, \dots, n$  以及观察发生链路故障前后智能体最终收敛状态的变化精确找到失效链路.

下面对符号簇一致性协议 (3) 下智能体的最终收敛状态进行详细分析. 定义  $y_i[k]$  为智能体  $v_i$  的转换状态, 其中  $y_i[k] = x_i[k], v_i \in \mathcal{V}_1; y_i[k] = -x_i[k], v_i \in \mathcal{V}_2$ . 令

$$y(k) = [y_1[k], y_2[k], \dots, y_n[k]]^T,$$

协议 (3) 可以表示成以下形式:

$$y[k+1] = (I_n - \tau(D - \mathcal{A}^{\text{sgn}})D^{-1})y[k], \quad (5)$$

其中  $C = \text{diag}(c_1, c_2, \dots, c_n)$  且

$$\mathcal{A}^{\text{sgn}} = \begin{bmatrix} 0 & \text{sgn}(a_{12}) & \dots & \text{sgn}(a_{1n}) \\ \text{sgn}(a_{21}) & 0 & \dots & \text{sgn}(a_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{sgn}(a_{n1}) & \text{sgn}(a_{n2}) & \dots & 0 \end{bmatrix}.$$

为了表示方便, 令  $H = I_n - \tau(D - \mathcal{A}^{\text{sgn}})C^{-1}$ .

**定理1** 在符号簇一致性协议 (3) 中, 选择  $\tau < \frac{1}{c_{\max}}$ , 其中  $c_{\max} = \max\{c_i : i = 1, 2, \dots, n\}$ . 如果结构平衡符号图  $\mathcal{G}$  是连通的, 那么对于任意初始状态, 智能体最终收敛状态为

$$\begin{aligned} \lim_{k \rightarrow \infty} x_i[k] &= \frac{c_i}{\sum_{j=1}^n c_j} y^T[0] \mathbf{1}_n, \quad v_i \in \mathcal{V}_1, \\ \lim_{k \rightarrow \infty} x_i[k] &= \frac{-c_i}{\sum_{j=1}^n c_j} y^T[0] \mathbf{1}_n, \quad v_i \in \mathcal{V}_2. \end{aligned} \quad (6)$$

**证明** 在条件  $\tau < \frac{1}{c_{\max}}$  下, 式 (5) 中的系统矩阵  $H$  是一个非对称的列随机矩阵, 可得  $H^T$  是一个行随机矩阵. 在  $\mathcal{G}$  是连通符号图的情况下, 首先利用反证法证明矩阵  $H^T$  是不可约矩阵. 假设矩阵  $H^T$  是可约的, 那么存在一个置换矩阵  $P$  使得

$$P^T H^T P = \begin{bmatrix} B & F \\ \mathbf{0} & E \end{bmatrix},$$

对于任意  $i \neq j$ , 有  $[H]_{ij} = \tau[D - \mathcal{A}^{\text{sgn}}]_{ij} c_j^{-1}$ ,  $[H]_{ji} = \tau[D - \mathcal{A}^{\text{sgn}}]_{ji} c_i^{-1}$ . 因为  $D - \mathcal{A}^{\text{sgn}}$  是对称矩阵, 所以  $[D - \mathcal{A}^{\text{sgn}}]_{ij} = [D - \mathcal{A}^{\text{sgn}}]_{ji}$ , 再根据  $c_i > 0, c_j > 0$  可得  $[H]_{ij} > 0 \Leftrightarrow [H]_{ji} > 0$ ,  $[H]_{ij} = 0 \Leftrightarrow [H]_{ji} = 0$ . 等价地,  $[H^T]_{ij} > 0 \Leftrightarrow [H^T]_{ji} > 0$ ,  $[H^T]_{ij} = 0 \Leftrightarrow [H^T]_{ji} = 0$ . 进一步有  $[P^T H^T P]_{ij} > 0 \Leftrightarrow [P^T H^T P]_{ji} > 0$  且  $[P^T H^T P]_{ij} = 0 \Leftrightarrow [P^T H^T P]_{ji} = 0$ . 因此可得  $F = \mathbf{0}$ . 这意味着图  $\mathcal{G}[P^T H^T P]$  是不连通的, 进而图  $\mathcal{G}[H^T]$  是不连通的. 根据矩阵  $H$  的定义可知符号图  $\mathcal{G}$  是不连通的, 这与已知条件矛盾. 因此矩阵  $H^T$  是不可约的.

因为  $H^T$  是行随机矩阵, 所以  $H^T$  的特征值的最大模等于 1. 下面证明 1 是矩阵  $H^T$  唯一的具有最大模的特征值. 根据参数  $\tau$  的选择范围可知行随机矩阵  $H^T$  的所有对角元素均为正实数. 令  $\varsigma = \min\{[H]_{ii} : i = 1, 2, \dots, n\} > 0$  且  $B = H^T - \text{diag}([H]_{11}, [H]_{22}, \dots, [H]_{nn})$ , 进而有  $H^T \geq \varsigma I_n + B$ . 因为矩阵  $H^T$  是不可约的, 所以矩阵  $B$  是不可约的. 令  $\zeta = [\zeta_1, \zeta_2, \dots, \zeta_n]^T \geq \mathbf{0}$  是一个非负列向量,  $\theta = (\varsigma I_n + B)\zeta$ . 定义集合  $\mathcal{K} = \{i : x_i > 0\} \neq \emptyset$ . 因此, 有

$$\theta_i = \varsigma \zeta_i + \sum_{j=1}^n [B]_{ij} \zeta_j, \quad i = 1, 2, \dots, n$$

成立. 进一步可以得到

$$\begin{aligned} \theta_i &\geq \zeta_i > 0, \quad \forall i \in \mathcal{K}, \\ \theta_i &\geq \zeta_i = 0, \quad \forall i \notin \mathcal{K}. \end{aligned}$$

这意味着向量  $\theta$  中正元素的个数大于等于向量  $\zeta$  中正元素的个数. 若  $\theta$  中正元素的个数等于  $\zeta$  中正元素的个数, 有  $\theta_i = \zeta_i = 0, i \notin \mathcal{K}$ . 进而可以得到  $\sum_{j=1}^n [B]_{ij} \zeta_j = 0, \forall i \notin \mathcal{K}$ . 因为  $\zeta_i = 0, i \notin \mathcal{K}$ , 所以有  $\sum_{j \in \mathcal{K}} [B]_{ij} \zeta_j = 0, \forall i \notin \mathcal{K}$ . 这意味着对于任意  $i \notin \mathcal{K}, j \in \mathcal{K}$ , 有  $[B]_{ij} = 0$  成立. 也就是说  $B$  是一个可约矩阵, 与已知条件矛盾. 因此  $y$  中正元素的个数一定等于  $x$  中正元素的个数. 用  $e_i = [0, \dots, 0, 1, 0, \dots, 0]$  表示一个向量, 其中第  $i$  个元素等于 1, 其他元素都等于 0. 那么可以得知  $(\varsigma I_n + B)^{n-1} e_i$  中正元素的个数比  $e_i$  中正元素的个数至少多  $n-1$  个, 也就是说  $(\varsigma I_n + B)^{n-1} e_i$  的所有元素均大于 0. 因此, 有

$$(\varsigma I_n + B)^{n-1} [e_1, e_2, \dots, e_n] > \mathbf{0} \Rightarrow (\varsigma I_n + B)^{n-1} > \mathbf{0}.$$

根据 Perron-Frobenius 定理可知 1 是矩阵  $H^T$  唯一的具有最大模的特征值.

假设与特征值 1 相关的左特征向量和右特征向量分别是  $\mathbf{1}_n$  和  $\xi^T$ , 其中  $\xi^T \mathbf{1}_n = 1$ , 然后可得  $H^T \mathbf{1}_n = \mathbf{1}_n$ ,  $\xi H^T = \xi$ . 根据文献 [1] 中的结果, 可以得到当  $k \rightarrow \infty$  时,  $(H^T)^k \rightarrow \mathbf{1}_n \xi^T$ , 其中

$$\xi^T = \left[ \frac{c_1}{\sum_{j=1}^n c_j}, \frac{c_2}{\sum_{j=1}^n c_j}, \dots, \frac{c_n}{\sum_{j=1}^n c_j} \right].$$

因为  $H^k = ((H^T)^k)^T$ , 所以  $\lim_{k \rightarrow \infty} H^k = \xi \mathbf{1}_n^T$ , 进一步可知式 (6) 成立.

## 4 链路故障检测

本节分别给出两种情况下的通信链路故障检测方法: (1) 多智能体网络在链路发生故障后不连通; (2) 多智能体网络在链路发生故障后保持连通. 为了方便, 我们令  $y_i^*$  和  $y_i'$  分别表示链路发生故障前后智能体  $v_i$  的转换状态.

### 4.1 链路发生故障后网络不连通的情况

**定理 2** 假设多智能体网络在链路发生故障后不连通. 考虑符号簇一致性协议 (3), 其中令参数  $c_s = 1, s = 1, 2, \dots, n$ . 对于任意通信链路  $(v_i, v_j)$ , 下面的结果成立:

(1) 对于任意的初始状态  $y_s[0], s = 1, 2, \dots, n$ , 如果有  $y_i' \neq y_j'$  成立, 则可以判断链路  $(v_i, v_j)$  发生了故障.

(2) 对于任意的初始状态  $y_s[0], s = 1, 2, \dots, n$ , 如果有  $y_i' = y_j'$  成立, 但是改变智能体  $v_i$  或者  $v_j$  的初始状态后有  $y_i' \neq y_j'$  成立, 则可以判断链路  $(v_i, v_j)$  发生了故障.

**证明** 如果多智能体网络在链路发生故障后不连通, 则原网络将被划分为若干不相交的连通子网络. 发生故障后, 属于同一子网络中智能体的最终转换状态会保持相同, 而属于不同子网络的智能体的最终转换状态可能相同也可能不相同, 这取决于初始状态的选取. 对于任意通信链路  $(v_i, v_j)$ , 如果实施符号簇一致性协议后, 有  $y_i' \neq y_j'$  成立, 则可以确定智能体  $v_i$  和  $v_j$  属于不同子网络, 从而可以判断链路  $(v_i, v_j)$  发生了故障. 如果有  $y_i' = y_j'$  成立, 那么智能体  $v_i$  和  $v_j$  可能属于同一子网络也可能属于不同子网络. 因此并不能判断链路  $(v_i, v_j)$  是否发生了故障. 针对这种情况, 需要进行进一步的分析.

(1) 考虑智能体  $v_i$  和  $v_j$  属于同一子网络的情形. 由于子网络是连通的, 因此改变  $v_i$  或者  $v_j$  的初始状态后, 再一次实施符号簇一致性协议, 这两个智能体仍然会收敛到一致, 即  $y_i' = y_j'$ .

(2) 考虑节点  $v_i$  和  $v_j$  属于不同子网络的情形. 假设智能体  $v_i$  属于子网络  $\mathcal{G}_1$ , 智能体  $v_j$  属于子网络  $\mathcal{G}_2$ . 子网络  $\mathcal{G}_1$  和  $\mathcal{G}_2$  的节点集分别假设为  $\mathcal{V}_1 = \{v_i, v_{i1}, \dots, v_{ik}\}$  和  $\mathcal{V}_2 = \{v_j, v_{j1}, \dots, v_{jm}\}$ . 那么有

$$y_i' = \frac{y_i[0] + y_{i1}[0] + \dots + y_{ik}[0]}{k+1},$$

$$y_j' = \frac{y_j[0] + y_{j1}[0] + \dots + y_{jk}[0]}{m+1},$$

其中  $y_i' = y_j'$ . 改变智能体  $v_i$  的初始状态为  $\bar{y}_i[0] \neq y_i[0]$ , 而其他智能体的初始状态保持不变时, 再一次实施符号簇一致性协议可以得到

$$\bar{y}_i' = \frac{\bar{y}_i[0] + y_{i1}[0] + \dots + y_{ik}[0]}{k+1},$$

$$y_j' = \frac{y_j[0] + y_{j1}[0] + \dots + y_{jk}[0]}{m+1},$$

从而得到  $y'_i \neq y'_j$ .

通过对以上两种情形进行分析可以得知, 当第一次实施协议 (3) 后出现  $y'_i = y'_j$  的情况时, 可以通过改变  $v_i$  或  $v_j$  的初始状态后, 再一次实施协议 (3) 来判断链路  $(v_i, v_j)$  是否发生了故障.

**注释2** 在多智能体系统的符号簇一致性协议中, 一个附加在智能体上的观察者记录并显示智能体的状态值. 对于网络在通信链路发生故障后不连通的情况, 我们在定理 2 中给出了检测失效链路的方法. 然而, 定理 2 的方法只能检测出那些造成网络不连通的故障链路. 在这样的情况中, 只有故障链路相连的智能体可以立即检测到故障, 而其他智能体则需要等待, 直到收到由智能体 (属于失效链路的一部分) 发起的错误消息.

#### 4.2 链路发生故障后网络保持连通的情况

在协议 (3) 中, 如果令  $c_i = |\mathcal{N}_i|$ ,  $i = 1, 2, \dots, n$ , 其中  $|\mathcal{N}_i|$  是智能体  $v_i$  的邻居的数量, 那么从定理 1 可以得知, 在符号簇一致性协议 (3) 中, 智能体  $v_i$  的最终收敛状态由它的邻居数量  $|\mathcal{N}_i|$  以及所有智能体的邻居数量之和  $\sum_{j=1}^n |\mathcal{N}_j|$  所决定. 因此, 如果多智能体网络在链路发生故障后保持连通, 那么当网络中的其中一条通信链路  $(v_i, v_j)$  发生故障之后,  $|\mathcal{N}_i|$  与  $\sum_{l=1}^n |\mathcal{N}_l|$  均会减小, 从而使得智能体  $v_i$  的最终收敛状态发生变化. 下面给出定理分别展示单条和多条通信链路的失败对智能体最终收敛状态的影响.

**定理3** 假设多智能体网络在链路发生故障后保持连通. 考虑符号簇一致性协议 (3), 其中  $c_i = |\mathcal{N}_i|$ ,  $i = 1, 2, \dots, n$ . 如果只有一条通信链路  $(v_i, v_j)$  在协议执行期间发生了故障, 则以下不等式成立:

$$\begin{aligned} y_s^* &> y'_s, \quad s \in \{i, j\}, \\ y_k^* &< y'_k, \quad k \notin \{i, j\}. \end{aligned} \quad (7)$$

**证明** 首先分析智能体  $v_i$  和  $v_j$  在链路  $(v_i, v_j)$  发生故障前后的收敛状态. 根据等式 (6) 可知在链路  $(v_i, v_j)$  发生故障之前, 智能体  $v_s \in \{v_i, v_j\}$  的最终转换状态为

$$y_s^* = \frac{|\mathcal{N}_s|}{\sum_{l=1}^n |\mathcal{N}_l|} y^T [0] \mathbf{1}_n. \quad (8)$$

在链路  $(v_i, v_j)$  发生故障之后, 智能体  $v_s$  的邻居个数为  $|\mathcal{N}_s| - 1$ , 而所有智能体总的邻居变为  $\sum_{l=1}^n |\mathcal{N}_l| - 2$ . 因此, 在链路  $(v_i, v_j)$  发生故障之后, 智能体  $v_s$  的最终转换状态为

$$y'_s = \frac{|\mathcal{N}_s| - 1}{\sum_{l=1}^n |\mathcal{N}_l| - 2} y^T [0] \mathbf{1}_n. \quad (9)$$

下面分两种情况对  $|\mathcal{N}_s|$  和  $\sum_{l=1}^n |\mathcal{N}_l|$  的关系进行分析: (1) 如果网络中的所有边都与智能体  $v_s$  相连接, 那么  $v_s$  的邻居数量就等于网络中边的数量  $|\mathcal{E}|$ . 根据所有智能体的邻居数量与网络中边数量的关系  $\sum_{l=1}^n |\mathcal{N}_l| = 2|\mathcal{E}|$ , 有

$$\sum_{l=1}^n |\mathcal{N}_l| = 2|\mathcal{N}_s| \quad (10)$$

成立; (2) 如果网络中除了与智能体  $v_s$  相连接的边之外还有其他的边, 那么  $v_s$  的邻居数量就小于网络中边的数量, 即  $|\mathcal{N}_s| < |\mathcal{E}|$ . 进而可以得到

$$\sum_{l=1}^n |\mathcal{N}_l| > 2|\mathcal{N}_s|. \quad (11)$$

值得注意的是, 在情况 (1) 中如果链路发生了故障, 那么网络就不连通, 这不符合链路发生故障后网络仍然保持连通的假设. 因此, 不等式 (11) 成立. 因此, 根据式 (8) 和 (9), 有

$$\begin{aligned} y_s^* - y_s' &= \left( \frac{|\mathcal{N}_s|}{\sum_{l=1}^n |\mathcal{N}_l|} - \frac{|\mathcal{N}_s| - 1}{\sum_{l=1}^n |\mathcal{N}_l| - 2} \right) y^T[0] \mathbf{1}_n \\ &= \frac{\sum_{l=1}^n |\mathcal{N}_l| - 2|\mathcal{N}_s|}{\sum_{l=1}^n |\mathcal{N}_l| (\sum_{l=1}^n |\mathcal{N}_l| - 2)} y^T[0] \mathbf{1}_n > 0. \end{aligned}$$

下面分析在链路  $(v_i, v_j)$  发生故障前后智能体  $v_k \notin (v_i, v_j)$  的最终转换状态. 首先根据等式 (6) 可知在链路  $(v_i, v_j)$  发生故障之前, 智能体  $v_k$  的最终转换状态为

$$y_k^* = \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l|} y^T[0] \mathbf{1}_n. \quad (12)$$

在链路  $(v_i, v_j)$  发生故障之后, 智能体  $v_k$  的最终转换状态为

$$y_k' = \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| - 2} y^T[0] \mathbf{1}_n. \quad (13)$$

结合 (12) 和 (13) 可以得到

$$\begin{aligned} y_k^* - y_k' &= \left( \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l|} - \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| - 2} \right) y^T[0] \mathbf{1}_n \\ &= \frac{-2|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| (\sum_{l=1}^n |\mathcal{N}_l| - 2)} y^T[0] \mathbf{1}_n < 0. \end{aligned}$$

定理得证.

**注释3** 定理 3 展示了单条通信链路发生故障前后智能体收敛状态的变化. 在这种情况下, 网络中的所有智能体 (总共  $n$  个) 都会收到最终收敛状态的变化. 因此, 一个附加到任何智能体的观察者都能够本地感知网络中的链路故障. 为了准确地找出哪条链路发生了故障, 我们需要  $n - 1$  个观察者.

**定理4** 假设多智能体网络在具有共同节点  $v_i$  的多条通信链路  $(v_i, v_{j_1}), (v_i, v_{j_2}), \dots, (v_i, v_{j_r})$  发生故障后保持连通. 实施符号簇一致性协议 (3) 后, 以下结果成立:

- (1)  $y_i^* > y_i'$ .
- (2) 对于链路  $(v_i, v_j)$ ,  $j \in \{j_1, j_2, \dots, j_r\}$ , 如果  $y_j^* \geq y_j'$ , 则可以判断链路  $(v_i, v_j)$  发生了故障.
- (3) 对于链路  $(v_i, v_j)$ ,  $j \in \{j_1, j_2, \dots, j_r\}$ , 如果  $y_j^* < y_j'$ , 且存在  $v_k \notin \{v_{j_1}, v_{j_2}, \dots, v_{j_r}\}$  使得  $\frac{y_j'}{y_j^*} < \frac{y_k'}{y_k^*}$ , 则可以判断链路  $(v_i, v_j)$  发生了故障.
- (4) 如果对于任意  $j, s \in \{j_1, j_2, \dots, j_r\}$ , 有  $\frac{y_j'}{y_j^*} = \frac{y_s'}{y_s^*}$  成立, 则可以判断链路  $(v_i, v_{j_1}), (v_i, v_{j_2}), \dots, (v_i, v_{j_r})$  都发生了故障.

**证明** 在链路发生故障之前, 智能体  $v_i$  的最终转换状态为  $y_i^* = \frac{|\mathcal{N}_i|}{\sum_{l=1}^n |\mathcal{N}_l|} y^T[0] \mathbf{1}_n$ . 在链路发生故障之后, 智能体  $v_i$  的邻居个数为  $|\mathcal{N}_i| - r$ , 而所有智能体总的邻居变为  $\sum_{l=1}^n |\mathcal{N}_l| - 2r$ . 因此, 在链路发生故障之后, 智能体  $v_i$  的最终转换状态为  $y_i' = \frac{|\mathcal{N}_i| - r}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} y^T[0] \mathbf{1}_n$ . 通过计算可得

$$\begin{aligned} y_i^* - y_i' &= \left( \frac{|\mathcal{N}_i|}{\sum_{l=1}^n |\mathcal{N}_l|} - \frac{|\mathcal{N}_i| - r}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} \right) y^T[0] \mathbf{1}_n \\ &= \frac{\sum_{l=1}^n |\mathcal{N}_l| - 2r|\mathcal{N}_i|}{\sum_{l=1}^n |\mathcal{N}_l| (\sum_{l=1}^n |\mathcal{N}_l| - 2r)} y^T[0] \mathbf{1}_n > 0. \end{aligned}$$



因此, 结果 (1) 成立.

下面分析在链路发生故障前后智能体  $v_k \notin \{v_i, v_{j_1}, v_{j_2}, \dots, v_{j_r}\}$  的最终转换状态. 在链路发生故障之前, 智能体  $v_k$  的最终转换状态为  $y_k^* = \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l|} y^T[0] \mathbf{1}_n$ . 在链路发生故障之后, 智能体  $v_k$  的最终转换状态为  $y_k' = \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} y^T[0] \mathbf{1}_n$ . 通过计算可得

$$\begin{aligned} y_k^* - y_k' &= \left( \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l|} - \frac{|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} \right) y^T[0] \mathbf{1}_n \\ &= \frac{-2r|\mathcal{N}_k|}{\sum_{l=1}^n |\mathcal{N}_l| (\sum_{l=1}^n |\mathcal{N}_l| - 2)} y^T[0] \mathbf{1}_n < 0, \\ \frac{y_k'}{y_k^*} &= \frac{\sum_{l=1}^n |\mathcal{N}_l|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r}. \end{aligned}$$

也就是说, 如果与智能体  $v_k$  的通信链路没有发生链路故障, 那么  $v_k$  的状态一定满足  $y_k^* - y_k' < 0$  和  $\frac{y_k'}{y_k^*} = \frac{\sum_{l=1}^n |\mathcal{N}_l|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r}$ . 根据逆否命题的等价性可以得知, 对于通信链路  $(v_i, v_j)$ , 如果  $y_j^* \geq y_j'$ , 则可以判断链路  $(v_i, v_j)$  发生了故障. 结果 (2) 成立.

对于通信链路  $(v_i, v_j)$ , 当  $y_j^* < y_j'$  时, 并不能确定通信链路  $(v_i, v_j)$  是否发生了故障. 那么需要进一步对  $\frac{y_j'}{y_j^*}$  进行分析. 如果  $(v_i, v_j)$  发生了故障, 那么有

$$\frac{y_j'}{y_j^*} = \frac{\sum_{l=1}^n |\mathcal{N}_l|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} * \frac{|\mathcal{N}_j| - 1}{|\mathcal{N}_j|} < \frac{\sum_{l=1}^n |\mathcal{N}_l|}{\sum_{l=1}^n |\mathcal{N}_l| - 2r} = \frac{y_k'}{y_k^*}$$

成立. 也就是说, 当  $y_j^* < y_j'$  且存在  $v_k$  使得  $\frac{y_j'}{y_j^*} < \frac{y_k'}{y_k^*}$  时, 就可以判断链路  $(v_i, v_j)$  发生了故障. 结果 (3) 成立.

如果对于任意  $j, s \in \{j_1, j_2, \dots, j_r\}$ , 有  $\frac{y_j'}{y_j^*} = \frac{y_s'}{y_s^*}$  成立, 那么通信链路  $(v_i, v_{j_1}), (v_i, v_{j_2}), \dots, (v_i, v_{j_r})$  同时发生或没有发生故障. 因为从  $y_i^* - y_i'$  可以判断出存在与节点  $v_i$  相连的通信链路发生了故障, 所以可以判断通信链路  $(v_i, v_{j_1}), (v_i, v_{j_2}), \dots, (v_i, v_{j_r})$  都发生了故障. 结果 (4) 成立.

对于两类不同的链路故障情形, 即链路发生故障后网络不连通和网络保持连通, 定理 2 和 3 通过设置合适的参数  $c_i$ , 分别给出了判断链路故障的代数条件. 实际上, 即使网络中同时存在两类链路故障, 仍然可以通过定理 2 和 3 给出的判定方法精确定位故障链路. 下面给出详细的链路故障检测流程图.

**注释4** 在符号簇一致性协议 (3) 中, 可以通过设计不同的参数  $c_i, i = 1, 2, \dots, n$  观察链路故障发生前后智能体收敛状态的变化, 从而精确找到失效链路. 例如, 对于链路发生故障后网络保持连通和网络不连通的情形, 我们分别可以设计参数  $c_i = 1$  和  $c_i = |\mathcal{N}_i|$  来找到失效链路, 详情分别请看定理 2 和 3. 此外, 即使一条链路发生故障后网络是否连通未知, 也可以根据参数链路故障检测流程图 2 来精确找到失效链路.

**注释5** 现有文献 [21~24] 主要针对完全合作网络上的多智能体系统链路故障检测问题展开研究. 与现有文献相比, 本文的主要贡献是基于符号簇一致性协议提出了解决合作竞争网络上的多智能体系统链路故障检测方法, 该方法同时适用于单链路故障和多链路故障的情形. 此外, 值得注意的是本文的方法仅适用于结构平衡的符号图, 对于结构不平衡的符号图是无效的. 因此, 我们将来的工作是提出一种新的有效方法, 用来解决结构不平衡符号图的链路故障检测问题.

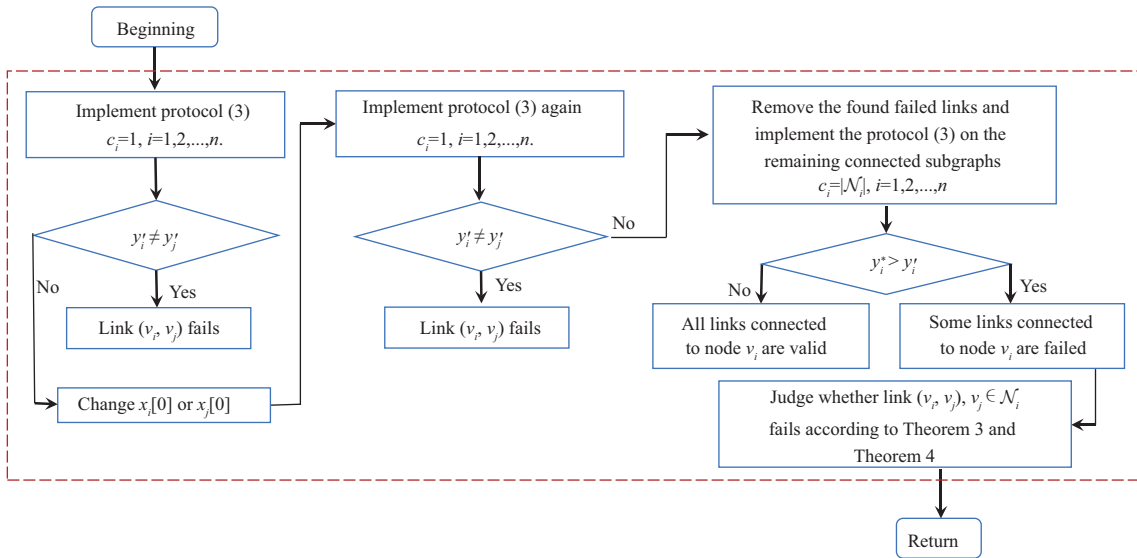


图 2 (网络版彩图) 链路故障检测流程图

Figure 2 (Color online) The flowchart of link failure detection

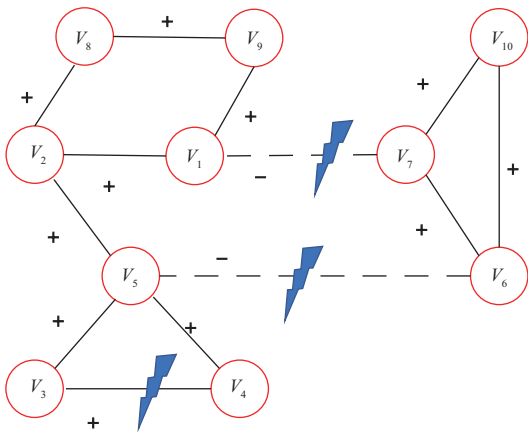


图 3 (网络版彩图) 由 10 个智能体组成的网络  $\mathcal{G}$ , 其中链路  $(v_1, v_7), (v_3, v_4), (v_5, v_6)$  发生了故障

Figure 3 (Color online) A network  $\mathcal{G}$  of ten agents in which the links  $(v_1, v_7), (v_3, v_4), (v_5, v_6)$  fail

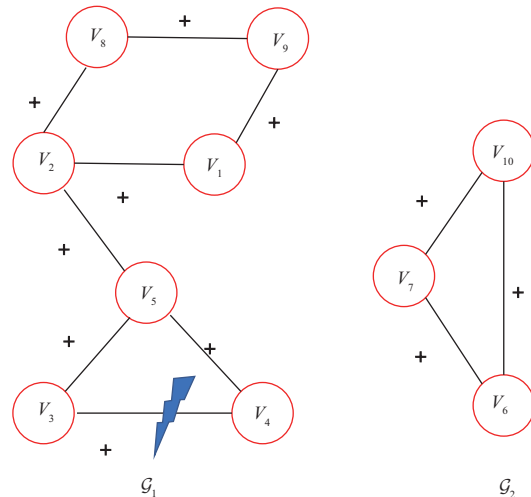


图 4 (网络版彩图) 网络  $\mathcal{G}$  移除故障链路  $(v_1, v_7), (v_5, v_6)$  后所得到的两个连通子网络  $\mathcal{G}_1$  和  $\mathcal{G}_2$

Figure 4 (Color online) Two connected subnetworks  $\mathcal{G}_1$  and  $\mathcal{G}_2$  obtained by removing the failed links  $(v_1, v_7), (v_5, v_6)$  from the network  $\mathcal{G}$

### 5 数值模拟

考虑一个由 10 个智能体构成的结构平衡网络  $\mathcal{G}$ , 如图 3 所示, 其中链路  $(v_1, v_7), (v_3, v_4), (v_5, v_6)$  发生了故障, “+” 和 “-” 分别表示智能体之间的合作和竞争关系. 智能体的初始状态被设置为  $x_1[0] = 0.5, x_2[0] = 1.5, x_3[0] = 2.5, x_4[0] = 3.5, x_5[0] = 4.5, x_6[0] = 1, x_7[0] = 2, x_8[0] = 3, x_9[0] = 2, x_{10}[0] = 4$ .

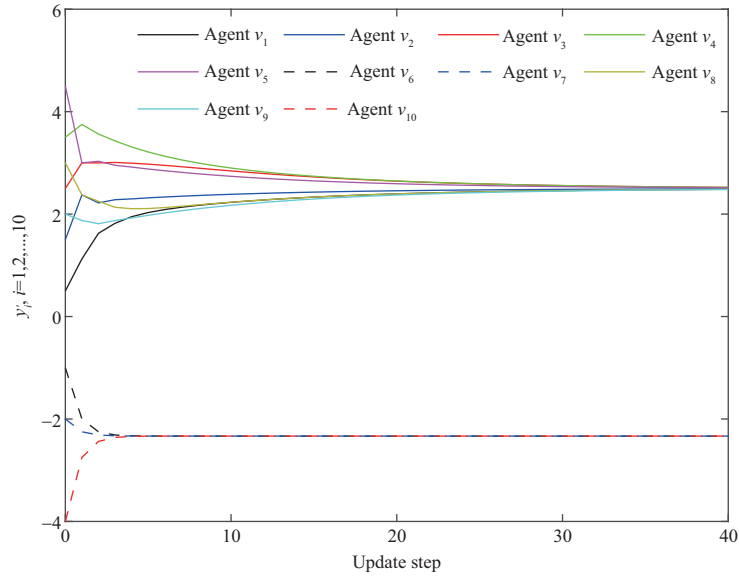


图 5 (网络版彩图) 在链路故障发生后的网络  $\mathcal{G}$  上实施协议 (3) 所得到的智能体转换状态轨迹

Figure 5 (Color online) The agents' transition state trajectories obtained by implementing protocol (3) on the network  $\mathcal{G}$  after link failures

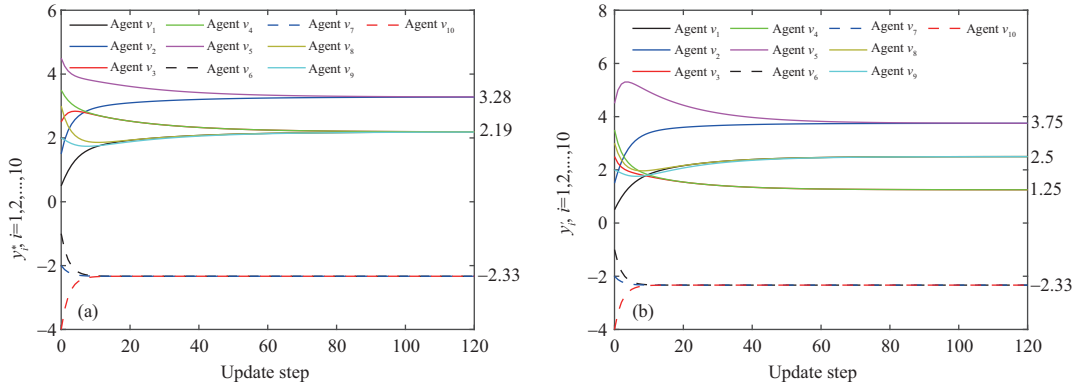


图 6 (网络版彩图) 链路发生故障前后在子网络  $\mathcal{G}_1$  和  $\mathcal{G}_2$  上实施协议 (3) 所得到的智能体转换状态轨迹

Figure 6 (Color online) The agents' transition state trajectories obtained by implementing protocol (3) on the sub-networks  $\mathcal{G}_1$  and  $\mathcal{G}_2$  before and after link failures

所有智能体被划分为两个子集  $\mathcal{V}_1 = \{v_1, v_2, v_3, v_4, v_5, v_8, v_9\}$  和  $\mathcal{V}_2 = \{v_6, v_7, v_{10}\}$ , 属于同一子集的相邻智能体之间相互合作, 属于不同子集的相邻智能体之间相互竞争. 因此, 智能体的初始转换状态被设置为  $y_i[0] = x_i[0], v_i \in \mathcal{V}_1, y_i[0] = -x_i[0], v_i \in \mathcal{V}_2$ . 选择时间步长  $\tau = \frac{1}{3.5}$ .

首先根据图 2 找出造成网络不连通的链路故障. 在网络上实施符号簇一致性协议 (3), 其中参数被设置为  $c_i = 1, i = 1, 2, \dots, 10$ . 从图 3 移除  $(v_1, v_7)$  和  $(v_5, v_6)$ , 可以得到两个连通子网络  $\mathcal{G}_1$  和  $\mathcal{G}_2$ , 如图 4 所示. 从图 5 可以看到在链路发生故障后智能体的最终收敛状态为  $y'_i = 2.5, v_i \in \mathcal{V}_1, y'_i = -2.33, v_i \in \mathcal{V}_2$ , 因此有  $y'_1 \neq y'_7, y'_5 \neq y'_6$ . 根据图 2 给出的链路故障检测流程, 可以判断链路  $(v_1, v_7)$  和  $(v_5, v_6)$  发生了故障. 这些连通子图中可能包含对网络连通性无影响的链路故障. 下面根据图 2 进行检测. 在子网络  $\mathcal{G}_1$  和  $\mathcal{G}_2$  上实施符号簇一致性协议 (3), 其中令  $c_i = |\mathcal{N}_i|, i = 1, 2, \dots, 10$ .

从图 6(a) 可以看到发生链路故障前智能体转换状态的收敛结果为  $y_1^* = y_3^* = y_4^* = y_8^* = y_9^* = 2.19$ ,  $y_2^* = y_5^* = 3.28$ ,  $y_6^* = y_7^* = y_{10}^* = -2.33$ , 从图 6(b) 可以看到发生链路故障后最终的智能体转换状态为  $y'_1 = y'_8 = y'_9 = 2.5$ ,  $y'_2 = y'_5 = 3.75$ ,  $y'_3 = y'_4 = 1.25$ ,  $y_6^* = y_7^* = y_{10}^* = -2.33$ . 通过计算可以得到  $y_i^* > y'_i$ ,  $i \neq 4, 5$  且  $y_i^* < y'_i$ ,  $i = 4, 5$ . 因此根据链路故障检测流程图可以判断链路  $(v_3, v_4)$  发生了故障.

## 6 总结

本文研究了合作竞争网络下的多智能体系统链路故障检测问题. 首先提出了一个符号簇一致性协议, 并分析了该协议下智能体的最终收敛状态. 然后通过判断链路发生故障前后智能体最终收敛状态的变化, 分别给出了合作竞争网络上判断单链路故障和多链路故障的检测方法. 与之前所提出检测方法的优点相比, 本文提出的算法可以利用收敛状态偏移来检测网络中的失效链路. 最终, 通过仿真实验验证了链路故障检测方法的有效性.

## 参考文献

- 1 Ren W, Beard R W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans Automat Contr*, 2005, 50: 655–661
- 2 Wang Z, Andrews M A, Wu Z X, et al. Coupled disease-behavior dynamics on complex networks: a review. *Phys Life Rev*, 2015, 15: 1–29
- 3 Ma L F, Wang Z D, Han Q-L, et al. Consensus control of stochastic multi-agent systems: a survey. *Sci China Inf Sci*, 2017, 60: 120201
- 4 Tang Y, Xing X, Karimi H R, et al. Tracking control of networked multi-agent systems under new characterizations of impulses and its applications in robotic systems. *IEEE Trans Ind Electron*, 2016, 63: 1299–1307
- 5 Shi L, Zheng W X, Shao J, et al. Scaled tracking consensus in discrete-time second-order multiagent systems with random packet dropouts. *IEEE Trans Syst Man Cybern Syst*, 2021, 51: 7745–7751
- 6 Shi L, Xiao Y, Shao J, et al. Containment control of asynchronous discrete-time general linear multiagent systems with arbitrary network topology. *IEEE Trans Cybern*, 2020, 50: 2546–2556
- 7 Shan Q H, Teng F, Li T S, et al. Containment control of multi-agent systems with nonvanishing disturbance via topology reconfiguration. *Sci China Inf Sci*, 2021, 64: 179203
- 8 Liu C, Wang M, Zeng Q, et al. Leader-following flocking for unmanned aerial vehicle swarm with distributed topology control. *Sci China Inf Sci*, 2020, 63: 140312
- 9 Shao J, Zheng W X, Shi L, et al. Leader-follower flocking for discrete-time Cucker-Smale models with lossy links and general weight function. *IEEE Trans Automat Contr*, 2021, 66: 4945–4951
- 10 Yu D, Chen C L P. Smooth transition in communication for swarm control with formation change. *IEEE Trans Ind Inf*, 2020, 16: 6962–6971
- 11 Chesson P, Kuang J J. The interaction between predation and competition. *Nature*, 2008, 456: 235–238
- 12 Calisi D, Farinelli A, Iocchi L, et al. Multi-objective exploration and search for autonomous rescue robots. *J Field Robot*, 2007, 24: 763–777
- 13 Tian L, Ji Z J, Hou T, et al. Bipartite consensus of edge dynamics on cooperation multi-agent systems. *Sci China Inf Sci*, 2019, 62: 229201
- 14 Pan L L, Shao H B, Xi Y G, et al. Bipartite consensus problem on matrix-valued weighted directed networks. *Sci China Inf Sci*, 2021, 64: 149204
- 15 Wen G, Wang H, Yu X, et al. Bipartite tracking consensus of linear multi-agent systems with a dynamic leader. *IEEE Trans Circ Syst II*, 2018, 65: 1204–1208
- 16 Shao J, Zheng W X, Shi L, et al. Bipartite tracking consensus of generic linear agents with discrete-time dynamics over cooperation-competition networks. *IEEE Trans Cybern*, 2021, 51: 5225–5235
- 17 Meng D. Bipartite containment tracking of signed networks. *Automatica*, 2017, 79: 282–289

- 18 Zhou Q, Wang W, Liang H, et al. Observer-based event-triggered fuzzy adaptive bipartite containment control of multiagent systems with input quantization. *IEEE Trans Fuzzy Syst*, 2021, 29: 372–384
- 19 Fan M C, Zhang H T, Wang M. Bipartite flocking for multi-agent systems. *Commun Nonlin Sci Numer Simul*, 2014, 19: 3313–3322
- 20 Xiao Q, Liu H, Wang X, et al. A note on the fixed-time bipartite flocking for nonlinear multi-agent systems. *Appl Math Lett*, 2020, 99: 105973
- 21 Gertler J J. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Syst Mag*, 1988, 8: 3–11
- 22 Rahimian M A, Ajourlou A, Aghdam A G. Characterization of link failures in multi-agent systems under the agreement protocol. In: *Proceedings of American Control Conference (ACC)*, 2012. 5258–5263
- 23 Dhal R, Torres J A, Roy S. Detecting link failures in complex network processes using remote monitoring. *Phys A-Stat Mech its Appl*, 2015, 437: 36–54
- 24 Pandey P K, Adhikari B, Chakraborty S. A diffusion protocol for detection of link failure and utilization of resources in multi-agent systems. *IEEE Trans Netw Sci Eng*, 2020, 7: 1493–1507
- 25 Parlangeli G, Valcher M E. On the detection and identification of edge disconnections in a multi-agent consensus network. 2021. ArXiv:2101.06728
- 26 Easley D, Kleinberg J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge: Cambridge University Press, 2010
- 27 Altafini C. Consensus problems on networks with antagonistic interactions. *IEEE Trans Automat Contr*, 2013, 58: 935–946

## Link failure detection for multi-agent systems on cooperation-competition networks

Jinliang SHAO<sup>†</sup>, Lei SHI<sup>†</sup>, Tong LI & Xilin ZHANG<sup>\*</sup>

*School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

\* Corresponding author. E-mail: xilinzhang@uestc.edu.cn

<sup>†</sup> Equal contribution

**Abstract** This paper proposes a distributed signed cluster consensus protocol to study the problem of link failure detection for multi-agent systems in cooperation-competition networks. Firstly, the asymptotic and final convergence states of agents under the signed cluster consensus protocol are analyzed using signed graph theory and nonnegative matrix theory. Secondly, by observing the change of the agent's convergence state before and after the link failure, a method for accurately judging the failed link is given. This method is suitable for both the cases of single link failure and multiple link failures. Finally, the effectiveness of the method is verified by numerical simulations.

**Keywords** multi-agent systems, cooperation-competition networks, link failure detection, consensus, distributed protocol