物理学报 Acta Physica Sinica

Chinese Physical Society

Institute of Physics, CAS

复杂网络牵制控制优化选点算法及节点组重要性排序

刘慧　王炳琚　陆君安　李增扬

Node-set importance and optimization algorithm of nodes selection in complex networks based on pinning control

Liu Hui　Wang Bing-Jun　Lu Jun-An　Li Zeng-Yang

引用信息 Citation: Acta Physica Sinica, 70, 056401 (2021)　DOI: 10.7498/aps.70.20200872

在线阅读 View online: https://doi.org/10.7498/aps.70.20200872

当期内容 View table of contents: http://wulixb.iphy.ac.cn

复杂网络牵制控制优化选点算法及节点组重要性排序*

刘 慧[1)] 王炳珺[1)] 陆君安[2)†] 李增扬[3)]

1) (华中科技大学人工智能与自动化学院, 武汉 430074)

2) (武汉大学数学与统计学院, 武汉 430072)

3) (华中师范大学计算机学院, 武汉 430079)

(2020 年 6 月 9 日收到; 2020 年 9 月 4 日收到修改稿)

本文研究复杂网络动力学模型的无向网络牵制控制的优化选点及节点组重要性排序问题. 根据牵制控制的同步准则, 网络的牵制控制同步取决于网络的 Laplacian 删后矩阵的最小特征值. 因此, 通过合理选择受控节点集得到一个较大的 Laplacian 删后矩阵最小特征值, 是牵制控制优化选点问题的核心所在. 基于 Laplacian 删后矩阵最小特征值的图谱性质, 本文提出了多个受控节点选取的递归迭代算法, 该算法适用于任意类型的网络. 通过 BA 无标度网络、NW 小世界网络及一些实际网络中的仿真实验表明: 该算法在控制节点数较少时, 能有效找到最优受控节点集. 最后讨论了在复杂网络牵制控制背景下节点组重要性排序问题, 提出节点组的重要性排序与受控节点的数目有关.

关键词: 复杂动态网络, 牵制控制, 优化选点算法, 节点组重要性

PACS: 64.60.aq, 07.05.Dz, 87.55.kd, 89.75.–k **DOI:** 10.7498/aps.70.20200872

1 引 言

近年来, 复杂网络上的控制与优化引起各研究领域的兴趣[1–4]. 在许多现实场景中, 控制一个网络的所有节点几乎是不现实的, 特别在节点数多、连接复杂的大规模网络中. 为了节约控制成本, 可以通过只控制网络的部分节点并利用网络的连通性来达到控制整个网络的目的, 即牵制控制. 该研究方向受到广泛关注, 并取得了许多代表性的进展. 文献 [5,6] 提出可以通过对网络的部分节点施加控制器来实现整个网络的同步, 并研究了无标度网络的牵制控制, 发现牵制控制度大的节点比随机选点控制所需要的节点少. 文献 [7] 提出了自适应牵制控制同步判据, 给出了网络中牵制节点的数量和耦合强度的关系式. 文献 [8] 发现了网络在线性反馈的牵制控制下, 可以通过自适应调整耦合强度使网络达到同步. 文献 [9,10] 从网络的图谱性质出发, 研究了复杂网络的牵制控制的能控性. 其他扩展性工作还有许多, 这里不再一一赘述. 上述工作主要集中在提出牵制控制同步的准则上, 没有深入研究如何选择控制节点达到网络的优化控制. 至今, 牵制控制如何选择受控节点仍然是没有充分解决的问题, 需进一步理解牵制控制策略与网络结构特征的关系[11].

目前, 有一些工作从网络的拓扑度量出发给出网络牵制控制选点的策略. 比如, 文献 [1] 提出, 当网络受控节点较少时, 应当优先控制度大的节点, 当受控节点较多时, 应当优先控制度较小的节点. Wang 和 Chen[5], Song 和 Cao[12], 以及 Ali 和

* 国家自然科学基金 (批准号:61773175, 61702377, 61773294) 资助的课题.

† 通信作者. E-mail: jalu@whu.edu.cn

Soleyman[13] 研究了基于度指标的选点方案, 倾向于控制度最大的节点; Rong 等[14]、Jia 和 Li[15] 讨论基于介性中心度 (betweenness centrality, BC) 的牵制控制方案. 文献 [16] 提出了一种基于数据流的牵制控制策略, 该策略在实际网络中可以获得与基于 BC 的牵制方案相似的效果, 但计算量更小. 文献 [17] 使用 K-shell 分解法来寻找处于网络中心位置的节点, 发现 K-core 值更大的节点能更有利于网络传播. 文献 [18] 提出根据节点的森林距离大小来选择网络的控制节点, 优先选择森林距离较小的节点, 且该策略可以应用在有向网络及非连通网络中. 文献 [19] 依据矩阵扰动理论得出推论, 可以根据 Laplacian 矩阵的最大特征值对应特征向量的最大分量选择控制节点, 所得节点对网络 Laplacian 矩阵的特征值影响最大. 但是, 这些工作都只能从某个或某些拓扑度量出发指导性地建议牵制选点方案, 或者对某类特性的网络提供针对性的建议, 很难给出一个精准的方案, 在实际应用时效果不好把握. 本文在我们前期工作[1] 的基础上, 结合 Laplacian 删后矩阵最小特征值的图谱性质, 提出了牵制控制多个节点的节点组筛选算法, 能精准地找出最优的受控节点集合. 该方法不局限于特定属性的复杂网络, 对任何类型的网络都适用.

2 模型介绍及问题提出

先介绍复杂网络动力学模型, 并给出牵制控制同步准则. 考虑如下具有 N 个节点的连续时间动态网络模型[20], 其中网络的所有节点的自身动力学一致, 网络的状态方程描述如下:

$$\dot{x}_i = f(x_i) - c\sum_{j=1}^{N} l_{ij} P x_j + u_j(x_1, \cdots, x_N), \quad (1)$$

其中 i=1, 2, ···, N. 在上述方程中, $x_i \in \mathbb{R}^n$ 表示节点 i 的状态, $f(\cdot)$ 描述节点的自身动力学, 正常数 c 表示网络的全局耦合强度, u_i 是施加在节点 i 上的控制器, $P \in \mathbb{R}^{n\times n}$ 是内连耦合矩阵, 它是正定或者半正定的. $\boldsymbol{L}_N = [l_{ij}]_{N\times N}$ 为网络的 Laplacian 矩阵. 本文仅考虑无向网络的情况, 即 Laplacian 矩阵对角线上的元素为节点的度, 非对角线元素分别根据节点 i, j 之间是否存在连边而为–1 或 0.

假设网络的目标状态 $s(t)$ 满足:

$$\dot{s}(t) = f(s(t)),\ s(0) = s_0.$$

对复杂网络进行牵制控制的目的是: 通过控制网络中的部分节点, 使得网络中所有节点的状态都趋近于目标状态 $s(t)$. 由文献 [1] 可知, 对于一个无向网络, 当选定其受控节点集时, 可以通过如下代数不等式来判别网络能否通过牵制控制达到同步,

$$\lambda_1(\boldsymbol{L}_{N-l}) > \alpha/c, \quad (2)$$

其中, l 表示受控节点个数; $\boldsymbol{L}_{N-l}$ 是网络拉普拉斯矩阵 $\boldsymbol{L}_N$ 删去受控节点对应行和列所得到的矩阵, 即拉普拉斯矩阵的删后子矩阵, 也称为 Grounded-Laplacian 矩阵[21], N–l 表示删后子矩阵的维数; $\lambda_1(\boldsymbol{L}_{N-l})$ 表示该删后子矩阵的最小特征值. 此外, 式中 α 是由节点自身动力学函数 $f(\cdot)$ 和内连耦合矩阵 P 决定的, 与网络全局耦合强度 c 及网络结构无关.

接下来, 提出本文要研究的问题, 即优化 (2) 式左端项. 当网络的受控节点数 l 给定时, 如何确定受控节点或者受控节点集, 使得被牵制控制的网络的 $\lambda_1(\boldsymbol{L}_{N-l})$ 最大, 这就是牵制控制优化选点问题. 然而, 寻找网络合适的受控节点集使得 $\lambda_1(\boldsymbol{L}_{N-l})$ 最大化, 这个问题的计算量是庞大的. 计算量主要来自两部分, 第一部分是计算矩阵特征值的复杂度, 第二部分是受控节点的组合数量庞大而造成的复杂度. 特别是牵制控制多个节点的情况, 从所有节点中选择最优的受控节点集 $S = \{s_1, s_2, \cdots, s_l\}$, 从而得到最大的 $\lambda_1(\boldsymbol{L}_{N-l})$, 这是一个计算量巨大的 NP-hard 问题. 现有的研究对于给定的受控节点数 l, 仅能对可能的优化控制节点集给出宽泛建议[1]. 例如当受控节点数较少时, 应当优先牵制度大的节点, 当受控节点数较多时, 优先控制度小的节点; 或者仅能给出 $\lambda_1(\boldsymbol{L}_{N-l})$ 的上界及下界拓扑特征估计, 并不能得到精确的最优受控节点集. 本文结合删后矩阵 $\lambda_1(\boldsymbol{L}_{N-l})$ 的几个特征估计式, 推导出网络节点的筛选条件, 提出控制多个节点的牵制控制优化选点算法, 能够在控制节点数较少时准确找出最大 $\lambda_1(\boldsymbol{L}_{N-l})$, 并减少 $\lambda_1(\boldsymbol{L}_{N-l})$ 的计算次数.

3 牵制控制优化选点算法

先介绍本文算法的理论基础及推导过程. 该算法旨在筛除低价值受控节点及受控节点组合, 减少受控节点集 $\lambda_1(\boldsymbol{L}_{N-l})$ 的计算, 从而以较少的计算量得出最优受控节点集.

定理 1[1] 假设网络拓扑结构用图 G 表示. 图 G 的节点集为 $V(G)=\{1,\cdots,N\}$, 受控节点集 $S\subset V$, 受控节点数为 l, 其中 $s_1,\cdots,s_l$ 表示受控节点, 令 $p_1,\cdots,p_{N-l}$ 表示未受控的节点, 未受控节点集表示为 V/S, 并用符号 ω_{pj} 表示受控节点集 S 到未受控节点 p_j 的连边数, 则有

$$\lambda_1(\boldsymbol{L}_{N-l})\leqslant\frac{\omega_{p1}+\omega_{p2}+\cdots+\omega_{pN-l}}{N-l}. \tag{3}$$

下面对 (3) 式进行变形, 用符号 e 表示受控节点集内部连边数 (即受控节点与受控节点之间的连边数), k_{si} 表示受控节点 s_i 的度. 注意到, 有如下关系式成立:

$$\omega_{p1}+\omega_{p2}+\cdots+\omega_{pN-l}=k_{s1}+k_{s2}+\cdots+k_{sl}-2e. \tag{4}$$

根据 (3) 式和 (4) 式可得

$$\begin{aligned}\lambda_1(\boldsymbol{L}_{N-l})&\leqslant\frac{k_{s1}+k_{s2}+\cdots+k_{sl}-2e}{N-l}\\&\leqslant\frac{k_{s1}+k_{s2}+\cdots+k_{sl}}{N-l}.\end{aligned} \tag{5}$$

由此得到能减少最优 $\lambda_1(\boldsymbol{L}_{N-l})$ 的计算次数的一个过滤条件. (5) 式基于节点度进行筛选, 即寻找一个节点集, 使节点总度满足该式. 而在实际网络中, 对于某些度较小的节点, 即使将其搭配度最大的前 $l-1$ 个节点, 所组成的节点集可能也无法满足条件, 因此这种度小的节点无需考虑, 这就是节点初步筛选的依据, 即节点度不满足下式的节点将直接被排除:

$$k\geqslant(N-l)\times\lambda_1^*-\sum_{i=1}^{l-1}k_i, \tag{6}$$

其中 λ_1^* 是当前得到的 $\lambda_1(\boldsymbol{L}_{N-l})$ 最大值. 此步骤将多个节点总度的判断转化成了对单个节点度的判断, 有效减少低价值节点的配对组合的计算.

其次, 假设通过初步筛选之后剩余了 n 个节点, 在这 n 个节点中满足不等式 (5) 的节点组合, 理论上有 $R=C_n^l$ 种, 从这些组合中寻找最优组合仍然计算量较大. 但往往在剩余 n 个节点中有大部分都是度较小的节点, 它们基本只能和度最大的节点组成满足条件 (5) 式的组合. 根据排列组合可知, 在剩余 n 个节点中找到 l 个满足度筛选条件的节点组合可以分为两部分: 第一部分是包含度最小节点的组合, 组合数有 $\sum\limits_{i=1}^{w}(C_m^i\times C_{n-m}^{l-i})$ 种, 其中 $w=\min(l,\ m)$, m 表示当前最小度节点的个数; 第二部分是不包含度最小的节点的组合, 组合数有 C_{n-m}^l 种. 这两部分的组合数用公式表示如下:

$$C_n^l=\sum_{i=1}^{w}(C_m^i\times C_{n-m}^{l-i})+C_{n-m}^l, \tag{7}$$

其中 C_n^l 是从剩余 n 个节点中选择 l 个节点的组合数量, $\sum\limits_{i=1}^{w}(C_m^i\times C_{n-m}^{l-i})$ 表示包含当前度最小节点的组合, 但实际上此部分组合中满足条件 (5) 的组合数量远小于理论值 $\sum\limits_{i=1}^{w}(C_m^i\times C_{n-m}^{l-i})$, 因而可以优先将带有度较小节点且满足 (5) 式的组合列出, 然后不用再考虑带有最小度节点的组合, 以缩减待考虑组合的数量. 另外, C_{n-m}^l 表示不包含当前度最小的节点的组合数, 此部分组合也并非全部满足筛选条件 (5) 式. 上述两部分均可继续递归计算. 类似 (7) 式的分析, 能将 $R=C_n^l$ 的 NP-hard 问题逐步递归分解, 且递归的最大栈深不超过剩余节点中度互不相同的数量. 相比于一般的穷尽法 (列出所有的组合 C_n^l, 再一一计算排除), 本文所提出的递归算法能极大地减少计算量及内存消耗.

牵制控制多个节点算法最优 $\lambda_1(\boldsymbol{L}_{N-l})$ 的计算算法具体实现如下: 在节点组合筛选过程中, 首先选择 l 个节点作为初始受控节点, 通常选择度排序中的前 l 个节点. 计算控制这 l 个节点的 $\lambda_1(\boldsymbol{L}_{N-l})$ 作为筛选标准 λ^*, 根据 (3) 式有

$$\lambda^*\leqslant\frac{k_{s1}^*+k_{s2}^*+\cdots+k_{sl}^*-2e^*}{N-l}. \tag{8}$$

如果控制节点集 $\{s_1,s_2,\cdots,s_l\}$ 要得到一个更大的 $\lambda_1(\boldsymbol{L}_{N-l})$, 即

$$\lambda_1(\boldsymbol{L}_{N-l})>\lambda^*, \tag{9}$$

则必须有

$$\frac{k_{s1}+k_{s2}+\cdots+k_{sl}-2e}{N-l}>\lambda^*, \tag{10}$$

$$\frac{k_{s1}+k_{s2}+\cdots+k_{sl}}{N-l}>\lambda^*. \tag{11}$$

(10) 式和 (11) 式即是筛选控制节点集的依据. 只要在后续节点集 $\lambda_1(\boldsymbol{L}_{N-l})$ 计算中得到一个更大的 $\lambda_1(\boldsymbol{L}_{N-l})$, 则将这个 $\lambda_1(\boldsymbol{L}_{N-l})$ 当作新的 λ^* 来筛选剩余节点集. 最终对所有剩余组合计算筛选后得到的 λ^* 即为最优 $\lambda_1(\boldsymbol{L}_{N-l})$.

需要说明的是, 筛选条件 (11) 式计算简单, 仅需要知道节点的度即可. 而筛选条件 (10) 式计算量相对大些, 还需要计算受控节点集的内部连边数, 比筛选条件 (11) 式准确程度更高. 所以在实际

仿真中, 可以结合两者实现计算复杂度和精准程度上的平衡, 也可以只采用其中的一个式子. 基于以上实现过程, 现将选取多个牵制节点的算法列出如下.

算法 1

步骤 1 选择度排名前 l 的节点当作初始控制节点, 然后计算其特征值 $\lambda_1(\boldsymbol{L}_{N-l})$ 作为 λ^*;

步骤 2 将 λ^* 代入 (6) 式, 通过节点的度筛除不能满足 (6) 式的节点;

步骤 3 递归计算剩余节点中的满足条件 (11) 的节点组合;

步骤 4 在第 3 步的基础上, 计算剩余节点中的满足条件 (10) 的节点组合;

步骤 5 逐个计算剩余节点组合对应的删后矩阵的 λ_1, 如果计算出更大的 $\lambda_1(\lambda_1 > \lambda^*)$, 则将其当作新的筛选标准, 返回第 2 步继续迭代;如果未找到更大的 λ_1, 则当前 λ^* 即是最优 $\lambda_1(\boldsymbol{L}_{N-l})$, 当前控制节点集即是最优控制节点集.

4 实验仿真及算法有效性分析

本节将上述选点算法分别运用到生成网络如无标度网络、小世界网络, 以及真实数据网络如 Email 网络、Dolphin 网络. 通过数值仿真说明所提算法减少计算量的有效性, 通过与其他选点算法的对比说明算法 1 相较于其他选点策略在选择 $\lambda_1(\boldsymbol{L}_{N-l})$ 较大的节点集时更优.

4.1 算法初步应用和最优控制节点集的讨论

例 1 考虑参数设置为 $N = 1000$, $q = 5$ 的一个 BA 无标度网络, 该网络以 5 个节点的环状图作为初始网络, 每次增加一个新的节点, 且依照度优先连接策略连接到网络中已有的 q 个节点上. 生成的 BA 网络见图 1. 根据本文的算法, 将受控节点数 l 从 2 增加到 4, 来观察最优受控节点集的情况. 节点的度排序情况见表 1.

牵制控制两个节点, 即 $l = 2$ 的情况. 如果枚举法需计算 499500 次 998 阶矩阵的最小特征值. 根据算法 1, 首先选择节点集合{2, 11}控节点集, 计算此时的 $\lambda_1(\boldsymbol{L}_{N-l})$, 以此 $\lambda^* = 0.1986$ 作为初始的筛选标准, 可以得到节点{2, 11, 3, 18, 1, 9, 4}满足条件 (6) 式, 然后依次计算节点组合的 $\lambda_1(\boldsymbol{L}_{N-l})$, 发现遍历所有的组合也没有得到更大的 $\lambda_1(\boldsymbol{L}_{N-l})$, 则最初的 $\lambda^* = 0.1986$ 为最优的 $\lambda_1(\boldsymbol{L}_{N-l})$, 节点集合{2, 11}为牵制控制两个节点时的最优受控节点集合.

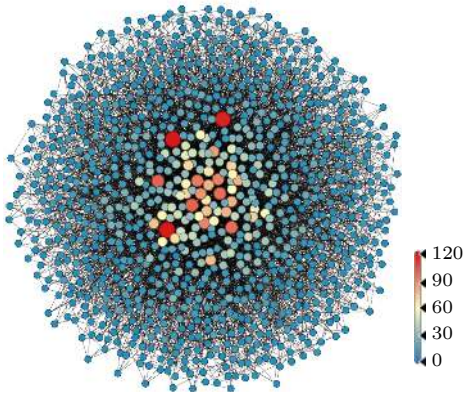

图 1 生成的 BA 网络. 不同颜色代表节点的度的大小, 红色表示度大的节点, 蓝色表示度小的节点

Fig. 1. A generated BA network. Different colors represent different node-degrees in the network; nodes in red have relative large degrees, and nodes in blue have small degrees.

牵制控制三个节点, 即 $l = 3$ 情况. 如果采用枚举法需计算 166167000 次 997 阶矩阵的最小特征值. 根据算法 1, 首先选择节点集合{2, 11, 3}作为初始的受控节点集合, 计算此时的 $\lambda^* = 0.3046$, 以此 λ^* 作为初始的筛选标准, 可以得到节点{2, 11, 3, 18, 1, 9, 4, 8, 14, 31}满足条件 (6) 式, 然后依次计算节点组合的 $\lambda_1(\boldsymbol{L}_{N-l})$, 发现节点组合{2, 3, 18}的 $\lambda_1(\boldsymbol{L}_{N-l}) = 0.3155$ 更大, 然后将此 $\lambda_1(\boldsymbol{L}_{N-l})$ 作为新的 λ^* 再进行筛选节点, 最终未能发现更大的 $\lambda_1(\boldsymbol{L}_{N-l})$, 即此时的节点集合{2, 3, 18}即为牵制控制三个节点时的最优受控节点集合.

牵制控制四个节点, 即 l=4 情况. 根据算法 1, 首先选择节点集合{2, 11, 3, 18}作为初始的受控制节点集, 计算此时的 $\lambda^* = 0.3894$, 以此 λ^* 作为初始的筛选标准, 发现节点的度排序前 17 的节点都满足筛选条件 (6), 依次计算节点组合的 $\lambda_1(\boldsymbol{L}_{N-l})$, 发现节点集合{2, 11, 18, 9}的 $\lambda_1(\boldsymbol{L}_{N-l}) = 0.3963$

表 1 节点度排序及节点编号 (BA 网络, $N = 1000$, $q = 5$)

Table 1. Degree ordering and node numbering in a BA network with $N = 1000$ and $q = 5$.

节点的度	128	122	103	98	87	86	75	63	60	58	53	51	51	49	47	43	42
节点编号	2	11	3	18	1	9	4	8	14	31	21	17	10	63	34	20	13

更大, 且将其作为新的筛选标准后, 没有发现更大的$\lambda_1(\boldsymbol{L}_{N-l})$, 即此时的节点集合{2, 11, 18, 9}即为牵制控制四个节点时的最优受控节点集合.

从上面的实验结果可以看出, 当受控节点数由少变多时, 比如从控制 3 个节点增加到控制 4 个节点时, 最优受控节点组合分别是{2, 3, 18}、{2, 11, 18, 9}, 优化的受控节点集合不是在控制 3 个节点的优化组合上加上一个节点, 而是在原有的基础上有所删减、有所变化的节点集合. 这就说明, 当受控节点数目 l 增加时, 最优受控节点集不能通过贪心选点策略得出.

4.2 算法 1 减少优化 $\lambda_1(\boldsymbol{L}_{N-l})$ 的计算量的有效性分析

为了客观评价本文算法减少计算量的效果, 这里给出两个指标:首次筛选剩余节点 n(算法 1 第二步), 度筛选后剩余节点组合 R(算法 1 第三步). 这两个指标能体现算法 1 在筛选中的效果. 这两个指标的计算结果均为 10 次重复实验取平均.

1) 首次筛选剩余节点数 n 相关仿真结果及分析

用 n 表示首次筛选(算法 1 第二步)后剩余的节点数, 实验结果在表 2 中列出. n 是一个重要的指标, 它很大程度决定了本文算法的计算复杂度, 因为$\lambda_1(\boldsymbol{L}_{N-l})$理论计算次数$C_n^l$与剩余节点 n 关系很大. 如果首次筛选不能将 n 缩减到 100 个节点左右的范围, 则待考虑的节点组合仍然会非常多, 后续的计算量比较大. 简而言之, n 越小算法效果越好. 由表 2 可见, 随着受控节点数 l 从 2 增加到 6, 剩余节点数 n 增加; 随着连边数 q 减少或者连接概率 P 减少, 剩余节点数 n 随之增加. 但实际上, 通过步骤 3 计算后的剩余节点组合远小于C_n^l. 因为即使是通过首次筛选后的节点, 它们当中度较小的节点也占相当大的比例, 它们往往只能搭配极少数度大的节点. 因此在C_n^l种组合中, 绝大部分组合都不满足条件, 详细见后文剩余节点组合 R 对应的实验数据.

2) 剩余节点的组合数量 R 相关仿真结果及分析

用 R 表示通过算法 1 第三步筛选后的节点组合数量. 指标 R 是$\lambda_1(\boldsymbol{L}_{N-l})$的实际计算次数的上限, 因为即使后续组合没有计算出更大的$\lambda_1(\boldsymbol{L}_{N-l})$来实现又一轮的节点组合筛选, 也仅需将剩余组合全部计算. 如果后续计算中得到了更大的$\lambda_1(\boldsymbol{L}_{N-l})$, 并依据算法 1 第二、三步再筛选掉一部分节点组合, 则计算次数将更少. 从表 3 中可以看到满足条件的组合数 R 远小于C_n^l, 这也说明了相比于穷举法, 本文的递归算法是有效的, 筛除了大量低价值节点组合.

表 2 通过步骤 2 筛选后的剩余节点数 n

Table 2. Number of remaining nodes n after Step 2 in Algorithm 1.

网络参数	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$
NW: $N=1000$, $P=0.05$	3.9	11.9	30.3	51.0	89.1
NW: $N=1000$, $P=0.025$	11.3	27.9	53.4	132.9	216.1
BA: $N=1000$, $q=10$	5.7	13.2	18.6	22.1	38.4
BA: $N=1000$, $q=8$	6.7	14.2	22.5	51.2	176.3
BA: $N=1000$, $q=5$	7.1	15.3	67.1	177.5	1000
BA: $N=1000$, $q=3$	10.2	55.7	1000.0	1000.0	1000.0

表 3 通过步骤 3 筛选后剩余节点的组合数量 R

Table 3. Number of combinations R of remaining nodes after Step 3 in Algorithm 1.

网络参数	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$
NW: $N=1000$, $P=0.05$	5.1	33.3	216.1	1136.5	4245.8
NW: $N=1000$, $P=0.025$	18.2	215.6	1009.5	1.1×10^4	4.7×10^4
BA: $N=1000$, $q=10$	3.3	11.5	34.3	163.1	1801.6
BA: $N=1000$, $q=8$	6.3	16.3	83.8	233.5	2583.7
BA: $N=1000$, $q=5$	21.6	69.5	306.5	2203.8	2.4×10^4
BA: $N=1000$, $q=3$	25.3	307.3	4376.2	2.2×10^5	4.5×10^6

4.3 本文算法与其他选点策略的对比

本节将对当前常用牵制控制选点策略与本文算法运用在实际网络中, 进行算法选点效果的对比. 首先介绍当前常用选点策略. 1) 度选点算法. 依据节点度排序来选择受控节点, 该算法无需额外计算量, 故通常在控制多个节点时, 采用度选点算法比较方便. 2) BC (betweenness centrality[14]) 选点策略. 该策略基于介数中心度选点, 旨在找出位于网络的重要路径上的节点. 3) K-shell 算法[22], 选取 K-core 值大的节点. 4) 最近提出的基于 ESI 指标的算法[19]. 5) 本文算法, 筛选计算出最优 $\lambda_1(\boldsymbol{L}_{N-l})$ 及受控节点.

将以上 5 种策略运用在实际网络 Dolphin 网络及 Email 网络[23] 中, 不同策略所得的选点及相应的 $\lambda_1(\boldsymbol{L}_{N-l})$ 见表 4 和表 5 所示. 表 4 和表 5 的实验结果表明: 本文算法得到的 $\lambda_1(\boldsymbol{L}_{N-l})$ 保持最优. 这验证了本文算法相比于其他算法更优. 另外, 把受控节点分布情况进行了可视化展示, 见图 2 和图 3. 图中蓝色节点为未受控节点, 黄色节点 (Dolphin 网络中) 及红色节点 (Email 网络中) 为受控节点, 节点尺寸越大, 表示节点度越大. 图 2 为 Dolphin 网络的情况, 4 个子图 (图 2(a)—图 2(d)) 给出了当受控节点数 $l=5$, 分别运用度算法、BC 算法、

表 4 不同算法在 Dolphin 网络中的选点及 $\lambda_1(\boldsymbol{L}_{N-l})$ 对比

Table 4. Node-selections and the corresponding $\lambda_1(\boldsymbol{L}_{N-l})$ under different algorithms on the dolphin network.

受控节点数	度算法	BC算法	K-shell算法	ESI算法	本文算法
$l=2$	(15, 46) $\lambda_1=0.1001$	(37, 2) $\lambda_1=0.1376$	(19, 30) $\lambda_1=0.0828$	(15, 38) $\lambda_1=0.0995$	(15, 18) $\lambda_1=0.2549$
$l=3$	(15, 46, 38) $\lambda_1=0.1053$	(37, 2, 41) $\lambda_1=0.2344$	(19, 30, 46) $\lambda_1=0.0935$	(15, 38, 46) $\lambda_1=0.1053$	(15, 14, 46) $\lambda_1=0.3664$
$l=4$	(15, 46, 38, 52) $\lambda_1=0.1064$	(37, 2, 41, 38) $\lambda_1=0.2511$	(19, 30, 46, 52) $\lambda_1=0.0950$	(15, 38, 46, 51) $\lambda_1=0.1069$	(62, 14, 46, 2) $\lambda_1=0.4662$
$l=5$	(15, 46, 38, 52, 34) $\lambda_1=0.1072$	(37, 2, 41, 38, 8) $\lambda_1=0.2710$	(19, 30, 46, 52, 22) $\lambda_1=0.0960$	(15, 38, 46, 51, 39) $\lambda_1=0.1078$	(15, 38, 52, 18, 14) $\lambda_1=0.5399$

表 5 不同算法在 Email 网络中的选点及 $\lambda_1(\boldsymbol{L}_{N-l})$ 对比

Table 5. Node-selections and the corresponding $\lambda_1(\boldsymbol{L}_{N-l})$ under different algorithms on the email network.

受控节点数	度算法	BC算法	K-shell算法	ESI算法	本文算法
$l=2$	(105, 333) $\lambda_1=0.0881$	(333, 105) $\lambda_1=0.0881$	(299, 389) $\lambda_1=0.0383$	(105, 42) $\lambda_1=0.0879$	(105, 23) $\lambda_1=0.0894$
$l=3$	(105, 333, 16) $\lambda_1=0.1169$	(333, 105, 23) $\lambda_1=0.1243$	(299, 389, 424) $\lambda_1=0.0392$	(105, 42, 333) $\lambda_1=0.1202$	(105, 333, 23) $\lambda_1=0.1243$
$l=4$	(105, 333, 16, 23) $\lambda_1=0.1518$	(333, 105, 23, 578) $\lambda_1=0.1490$	(299, 389, 424, 552) $\lambda_1=0.0494$	(105, 42, 333, 16) $\lambda_1=0.1481$	(105, 333, 23, 42) $\lambda_1=0.1535$
$l=5$	(105, 333, 16, 23, 42) $\lambda_1=0.1801$	(333, 105, 23, 578, 76) $\lambda_1=0.1774$	(299, 389, 424, 552, 571) $\lambda_1=0.0520$	(105, 42, 333, 16, 76) $\lambda_1=0.1770$	(105, 333, 23, 42, 41) $\lambda_1=0.1843$

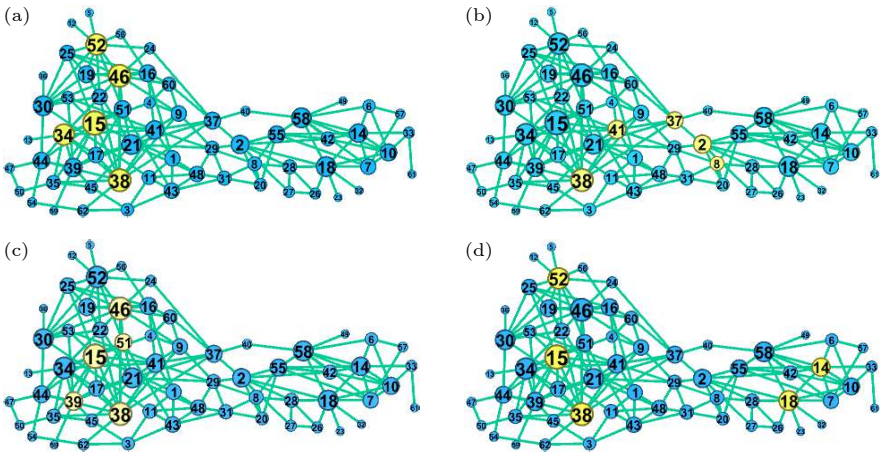

图 2 在 4 种不同算法下 Dolphin 网络的选点情况 (a) 度算法选点情况; (b) BC 算法选点情况; (c) ESI 算法选点情况; (d) 本文算法选点情况

Fig. 2. Visualization of nodes selections on the Dolphin network underfour strategies ($l=5$): (a) Using the degree-based pinning scheme; (b) using the BC-based pinning scheme; (c) using the ESI-based pinning scheme; (d) using our proposed algorithm.

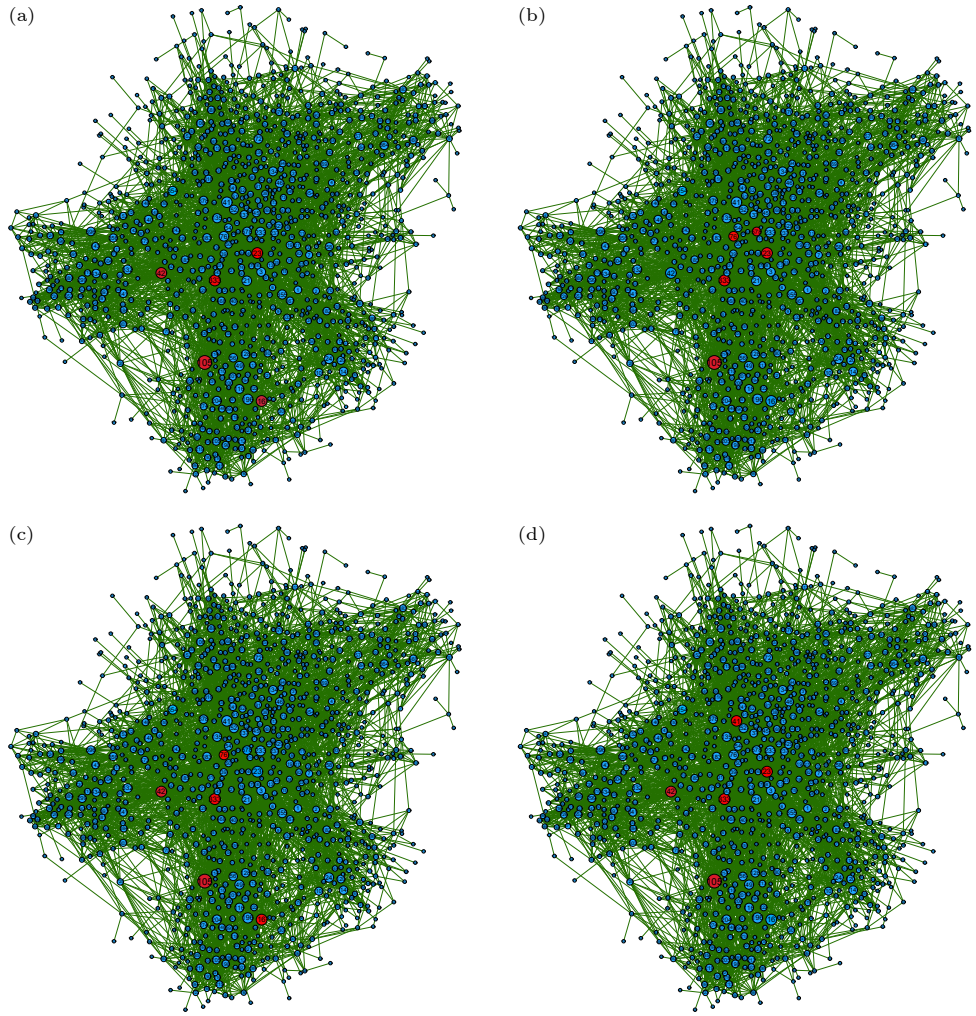

图 3 在 4 种不同算法下 Email 网络的选点情况 (a) 度算法选点情况; (b) BC 算法选点情况; (c) ESI 算法选点情况; (d) 本文算法选点情况

Fig. 3. Visualization of nodes selections on the Email network under four strategies: (a) Using the degree-based pinning scheme; (b) using the BC-based pinning scheme; (c) using the ESI-based pinning scheme; (d) using our proposed algorithm.

K-shell 算法、本文算法得到的受控节点分布情况. 图 3 给出 Email 网络的情况. 从上述网络选点情况可见, 度选点策略通常难以分辨节点的相对位置, 而不能准确地找到最优牵制控制节点集. BC 选点策略擅长寻找到关键路径上的节点, 而这样的节点容易在一条关键路径上前后成群出现, 也不够准确找到最优牵制控制的节点集. K-shell 选点策略擅长寻找在网络中的相对中心位置的节点, 集中性强, 当节点数较多时, 对网络边缘节点控制较差. ESI 算法是根据网络 Laplacian 矩阵的某个特征向量来选点, 故而其选点策略在节点的拓扑特征上不好分析, 但从实验结果发现该算法的效果不太稳定. 本文所提算法能够准确计算出控制节点数量较少时的最优受控节点, 这样的节点往往度较大, 在网络中的位置也相对分散, 是牵制控制的最优节点集.

在上述仿真中可见, 本文算法相较于其他算法, 能计算出控制节点数较少情况下($l \leqslant 6$)的最优控制节点集, 一定程度上填补了准确找出最优受控节点算法的空缺. 但本文算法针对控制节点较多情况, 尽管能够一定程度上缩减计算量, 但剩余计算量仍然较大, 难以直接计算.

5 节点组重要性排序

如何确定网络的重要节点组, 这在现实应用场景中广泛存在. 前面研究的确定网络重要节点组, 实际上就是提出了一种网络节点组重要性排序方法. 文献 [1] 提出对于无向网络确定重要节点组由网络 Laplacian 删后主子矩阵的最小特征值决定; 并且导出其上下界的精细估计, 指出按度大小牵制

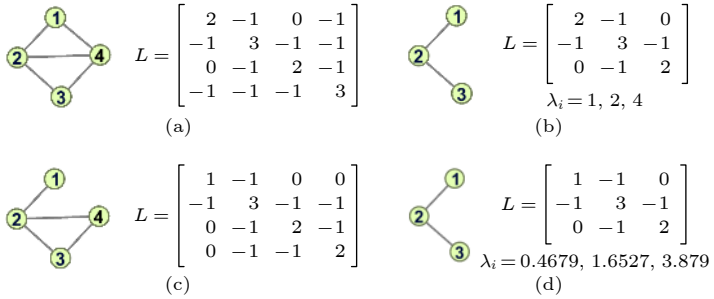

图 4 两个网络 A 与 B 结构不同, 但删去节点 4 后网络相同 (a) 网络 A 及其 Laplacian 矩阵; (b) 网络 A 删除节点 4 及其 Laplacian 矩阵; (c) 网络 B 及其 Laplacian 矩阵; (d) 网络 B 删除节点 4 及其 Laplacian 矩阵

Fig. 4. The structures of networks A and B are different, but the remaining structures are the same after deleting node 4. (a) network A and its Laplacian matrix; (b) network A deleting node 4 and its Laplacian matrix; (c) network B and its Laplacian matrix; (d) network B deleting node 4 and its Laplacian matrix.

控制并不是最优方法, 牵制控制度大还是度小的节点取决于控制节点的数目; 本文给出了一定条件下删后主子矩阵最小特征值一个优化算法.

通过下面的例 1 说明删后主子矩阵及其特征值蕴含了原矩阵的隐藏信息.

例 1 见图 4, 原网络 A, B 不同, 删去节点 4 后的网络相同, 但是删后子矩阵不同, 其特征值也不同. 网络 A 的删后矩阵最小特征值为 1, 网络 B 的删后矩阵最小特征值为 0.4679. 这说明了节点 4 在网络 A 中比在网络 B 中更重要. 在这个简单的例子中, 虽然删后网络相同, 但是删后主子矩阵特征值保留了原网络的隐藏信息.

下面给出一些例子说明删后矩阵最小特征值方法应用在节点组重要性排序上的有效性.

例 2 一个简单链状网络上的分析. 见图 5 所列 5 个节点的链状网络.

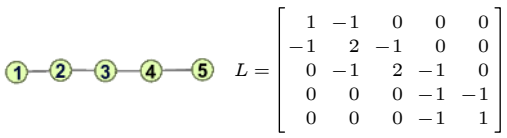

图 5 链状网络节点数 $N=5$ 及其拉普拉斯矩阵

Fig. 5. Chain graph with $N=5$ and its Laplacian matrix.

在该网络中, 对于一个节点的重要性排序, 节点 1, 2, 3 的 $\lambda_1(\boldsymbol{L}_{N-l})$分别为 0.1206, 0.1981, 0.3820, 所以单个节点排序中节点 3 最重要, 节点 2 和 4 其次, 节点 1 和 5 最差. 对于两个节点的节点组重要性排序: 节点组 (2, 4), (1, 4), (2, 5) 的 $\lambda_1(\boldsymbol{L}_{N-l})=1$(最重要), (1, 5) 为 0.5858 (其次重要), (2, 3), (1, 3), (3, 4), (3, 5) 为 0.3820 (较不重要), (1, 2), (4, 5) 为 0.1981(最不重要). 这一简单例子也说明, 单个节点排序最重要的节点不一定包含在两个节点的节点组的第一位.

例 3 一根水平樑如何选择两个基点 (l=2) 将它吊起呢? 假设樑长度为 1, 然后作细等分(只要细分足够密, 离散网络的节点组排序可以逼近连续问题的基点选择), 分点视为节点, 便成为 N 个节点的链. 设 N=82, 根据对称性, 第一个基点节点从节点 1—41 中选, 另一个从 82—42 中选. 图 6 刻画了 $\lambda_1(L_{N-2})$与所选节点位置的关系. 从图 6 可以看出, 当两个节点选取 (21, 62) 时特征值取得最大值. 当两个节点都取两端 (1, 82) 或中间 (41, 42) 时, 特征值达最小值. 也就是说对于链状图最优节点接近在 1/4 和 3/4 的位置. 例如取 $N=302$ 的链, 经计算最优节点为 (76, 227), 也符合上述 1/4 和 3/4 的规律.

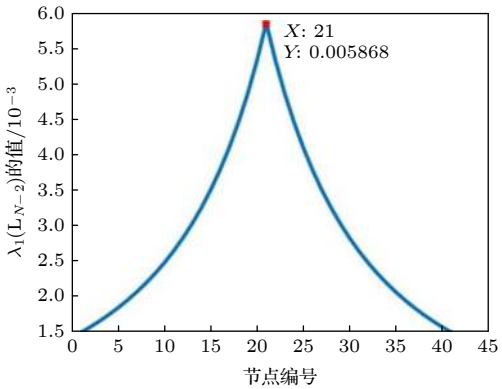

图 6 $\lambda_1(L_{N-2})$与左节点位置 (两节点对称选取) 的关系图, 这里 $N=82$

Fig. 6. The relationship of $\lambda_1(L_{N-2})$and the left node's position, where $N=82$ and the two nodes are selected symmetrically.

例 4 寻找正方形网络中最重要的 2 个节点组成的节点组. 在正方形网格中选两个最重要的

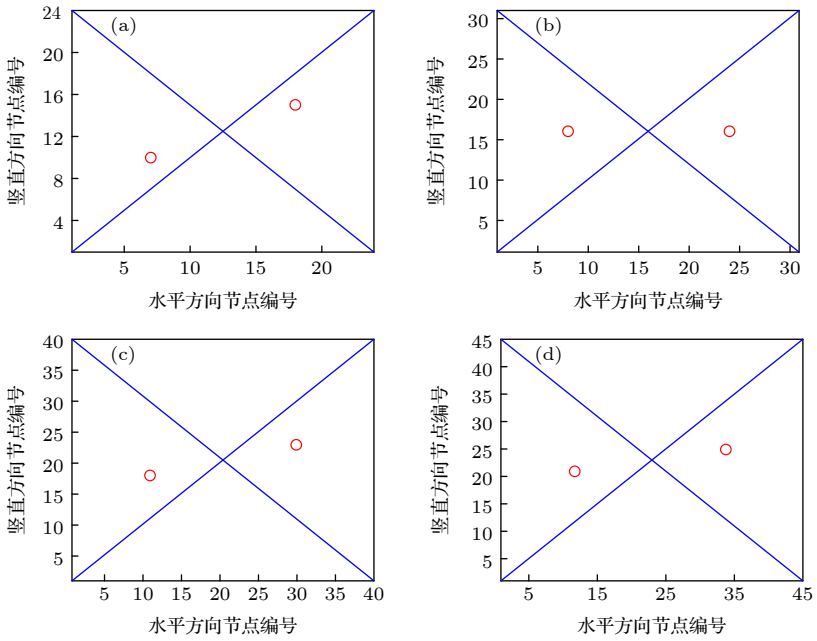

图 7 在正方形网格中选两个最重要的节点, 见图中红色的点 (a) 24 × 24 的正方形网格; (b) 31 × 31 的正方形网格; (c) 40 × 40 的正方形网格; (d) 45 × 45 的正方形网格

Fig. 7. Pinning two nodes in a square lattice. The optimal options are shown by red nodes: (a) The square with 24 × 24 nodes; (b) the square with 31 × 31 nodes; (c) the square with 40 × 40 nodes; (d) the square with 45 × 45 nodes.

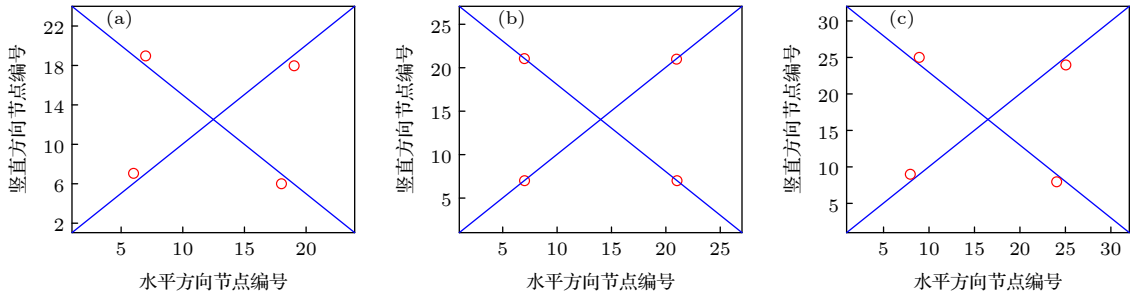

图 8 正方形网络选 4 个最重要的节点, 见图中红色的点 (a) 24 × 24 正方形网格; (b) 27 × 27 正方形网格; (c) 32 × 32 正方形网格

Fig. 8. Pinning four nodes in a square lattice. The optimal options are shown by red nodes: (a) The square with 24 × 24 nodes; (b) the square with 27 × 27 nodes; (c) the square with 32 × 32 nodes.

节点使得 $\lambda_1(L_{N-2})$最大. 图 7 中红色的点是计算得到的最优位置. 从图 7 发现, 控制节点处于对称位置, 却并不一定处于对角线上.

接着, 在正方形网络中寻找最重要的 4 个节点, 发现找到的最优节点组也处于对称位置, 但多数情况都不是位于对角线上, 大正方形网络分成 4 个小正方形网络, 最优控制节点在 4 个小正方形找中心节点偏移一格的位置上, 见图 8(a)和图 8(c)所示情形; 也有落在对角线上的情形, 见图 8(b).

例 5 社团的重要性. 比较三个社团在网络中的重要性, 见图 9, 其中红色 8 个节点, 蓝色 7 个节点, 绿色 6 个节点. 它们的 Laplacian 矩阵的删后主子矩阵最小特征值分别为 0.1905, 0.1792, 0.1597, 说明红色组最重要, 其次为蓝色, 最后为绿色.

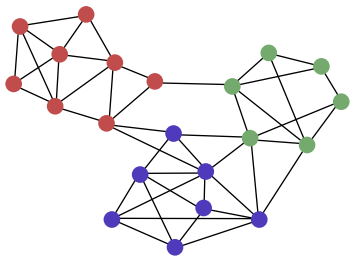

图 9 比较三个社团在网络中的重要性, 图中红蓝绿色标识了三个社团, 该图取自文献 [24]

Fig. 9. Compare the importance of three communities in a network, in which red, blue, and green colors implicit three different communities. This network is taken from Ref. [24].

6 结 论

复杂网络的牵制控制优化是一个十分有意义的研究方向. 但是由于控制多个节点时的最优受控节点集的选择是一个 NP-hard 问题, 如何找出最优受控节点集是一个富有挑战性的问题, 本文正是基于此开展工作. 基于 Laplacian 删后矩阵的图谱性质, 提出了选择最优受控节点集的算法, 该算法在受控节点数较少 (小于等于 6) 时, 能够准确地找到最优的受控节点集合. 进一步, 本文提出复杂网络节点组重要性的问题, 这与以往定义网络节点重要性不同: 本文所提出的节点组重要性, 节点的选取依赖于节点组所包含节点的数目;节点组包含节点的数目不同, 会有不同的重要节点选择方案及排序. 在以后的研究中, 我们将分析牵制控制策略及节点组重要性在多层网络[25]中的情况, 并希望进一步挖掘节点组重要性在实际网络中的应用.

参考文献

[1] Liu H, Xu X, Lu J A, Chen G, Zeng Z 2021 *IEEE Trans. Syst. Man Cybern. Syst.* **51** 786

[2] Wang K L, Wu C X, Ai J, Su S 2019 *Acta Phys. Sin.* **68** 196402 (in Chinese) [王凯莉, 邬春学, 艾均, 苏湛 2019 物理学报 **68** 196402]

[3] Han W T, Yi P, Ma H L, Zhang P, Tian L 2019 *Acta Phys. Sin.* **68** 186401 (in Chinese) [韩伟涛, 伊鹏, 马海龙, 张鹏, 田乐 2019 物理学报 **68** 186401]

[4] Li M, Wang B H 2014 *Chin. Phys. B* **23** 076402

[5] Wang X F, Chen G R 2002 *Phys. A* **310** 521

[6] Li X, Wang X F, Chen G R 2004 *IEEE Trans. Circuits Syst. Regul. Pap.* **51** 2074

[7] Zhou J, Lu J A, Lu J H 2008 *Automatica* **44** 996

[8] Yu W W, Chen G R, Lu J H 2009 *Automatica* **45** 429

[9] Francesco S, Mario D B, Franco G, Chen G R 2007 *Phys. Rev. E* **75** 046103

[10] Wang L, Dai H P, Dong H, Cao Y Y, Sun Y X 2008 *Eur. Phys. J. B* **61** 335

[11] Wang X F, Su H S 2014 *Annu Rev Control* **38** 103

[12] Song Q, Cao J D 2009 *IEEE Trans. Circuits Syst. Regul. Pap.* **57** 672

[13] Ali G, Soleyman A 2016 *Nonlinear Dyn.* **83** 1003

[14] Rong Z H, Li X, Lu W L 2009*Proc. IEEE Int. Symp. Circuits Syst.*Taipei, China, May 17–24, 2009 p1689

[15] Jia Z, Li X 2010 *29th Chinese Control Conference* Beijing, China, July 29–31, 2010 p4656

[16] Wang X Y, Liu X W 2018 *Nonlinear Dyn.* **92** 13

[17] Gong K, Kang L 2018 *J. Syst. Sci. Inf.* **6** 366

[18] Jin Y, Bao Q, Zhang Z 2019 *IEEE Int. Conference on Data Mining*, Beijing, China, November 8–11, 2019 p339

[19] Amani A M, Jalili M, Yu X, Stone L 2017 *IEEE Trans. Circuits Syst. Express Briefs* **64** 685

[20] Lu J A, Liu H, Chen J 2016 *Synchronization in Complex Dynamical Networks* (Vol. 1) (Beijing: Higher Education Press) p49 (in Chinese) [陆君安, 刘慧, 陈娟 2016 复杂动态网络的同步(第一版)(北京: 高等教育出版社)第49页]

[21] Pirani M, Sundaram S 2015 *IEEE Trans. Autom. Control* **61** 509

[22] Kitsak M, Gallos L, Havlin S, Liljeros F, Muchnik L, Stanley H E, Makse H A 2010 *Nat. Phys.* **6** 888

[23] Physicians network dataset, KONECT http://konect.uniko blenz.de/networks/ [2017.9.9]

[24] Danielle S B, Mason A P, Nicholas F W, Scott T G, Jean M C, Peter J M 2013 *Chaos* **23** 013142

[25] Xu M M, Lu J A, Zhou J 2016 *Acta Phys. Sin.* **65** 028902 (in Chinese) [徐明明, 陆君安, 周进 2016 物理学报 **65** 028902]

Node-set importance and optimization algorithm of nodes selection in complex networks based on pinning control*

Liu Hui [1)] Wang Bing-Jun [1)] Lu Jun-An [2)†] Li Zeng-Yang [3)]

1) (*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan* 430074, *China*)

2) (*School of Mathematics and Statistics, Wuhan University, Wuhan* 430072, *China*)

3) (*School of Computer Science, Central China Normal University, Wuhan* 430079, *China*)

(Received 9 June 2020; revised manuscript received 4 September 2020)

Abstract

Controlling a complex network to achieve a certain desired objective is an important task for various interacting systems. In many practical situations, it is expensive and unrealistic to control all nodes especially in a large-scale complex network. In order to reduce control cost, one turns to control a small part of nodes in the network, which is called pinning control. This research direction has been widely concerned and much representative progress has been achieved so far. However, to achieve an optimal performance, two key questions about the node-selection scheme remain open. One is how many nodes need controlling and the other is which nodes the controllers should be applied to. It has been revealed in our recent work that the effectiveness of node-selection scheme can be evaluated by the smallest eigenvalue λ_1 of the grounded Laplacian matrix obtained by deleting the rows and columns corresponding to the pinned nodes from the Laplacian matrix of the network. As a further study of our previous work, we study node selection algorithm for optimizing pinning control in depth, based on the proposed index λ_1 and its spectral properties. As is well known, it is an NP-hard problem to obtain the maximum of λ_1 by numerical calculations when the number of pinned nodes is given. To solve this challenge problem, in this paper a filtering algorithm is proposed to find most important nodes, which results in an optimal λ_1 when the number of pinned nodes is given. The method can be applied to any type of undirected networks. Furthermore, in this paper we propose the concept of node-set importance in complex networks from the perspective of network control, which is different from the existing definitions about node importance of complex networks: The importance of a node set and the selected nodes in this paper depends on the number of pinned nodes; if the number of pinned nodes is different, the selected nodes will be different. The concept of node-set importance reflects the effect of nodes' combination in a network. It is expected that the obtained results are helpful in guiding the optimal control problems in practical networks.

Keywords: complex dynamical networks, pinning control, optimization algorithm, node-set importance

PACS: 64.60.aq, 07.05.Dz, 87.55.kd, 89.75.–k **DOI:** 10.7498/aps.70.20200872

* Project supported by the National Natural Science Foundation of China (Grant Nos. 61773175, 61702377, 61773294).

† Corresponding author. E-mail: jalu@whu.edu.cn