

doi: 10.3969/j.issn.0490-6756.2020.01.010

嵌入社区半径的力引导与径向树混合布局算法

任淑霞, 吴涛, 张书博

(天津工业大学计算机科学与软件学院, 天津 300387)

摘要: 力引导布局算法存在无法展示复杂网络社区结构的缺陷, 虽引入聚类的方式来展示社区结构, 但社区内节点拥挤且排列无序, 不利于观察社区内节点的结构特征与连边关系, 为此提出嵌入社区半径的力引导与径向树混合布局算法. 该算法首先采用 K-means 算法对网络节点进行社区划分; 然后, 用社区内节点数量确定社区半径, 并将社区半径嵌入到社区斥力、引力中来展示社区结构; 最后, 采用径向树布局分层可视化各社区内节点. 实验中使用拥挤区域占比、点分布偏差、节点偏差等指标验证了本算法既能降低拥挤度又能减少节点布局偏差, 可视化结果显示, 本算法布局社区结构明显, 节点层次分明, 易于理解.

关键词: 力引导布局算法; 社区半径; 径向树; 混合布局; 复杂网络

中图分类号: TP393.02 **文献标识码:** A **文章编号:** 0490-6756(2020)01-0073-09

Community radius embedded in force-directed and radial tree hybrid layout algorithm

REN Shu-Xia, WU Tao, ZHANG Shu-Bo

(Department of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China)

Abstract: Force-directed layout has a defect that can't display complex network community structure. Although cluster layout algorithm is introduced to visualize community structure, nodes in the community are crowded and disordered, which is not conducive to observing structural features and nodes connection relationship in the community. Therefore, community radius embedded in force-directed and radial tree hybrid layout algorithm is proposed. Firstly, the algorithm uses the K-means algorithm to divide the network nodes into community. Then, the community radius is determined by the number of nodes in each community, and the community radius is embedded into the repulsion and gravity to achieve the effect of cluster layout. Finally, radial tree layout is used for each community to hierarchically visualize the nodes within the community. In the experiment, crowded area ratio, point distribution deviation, node deviation and other indicators are used to show that the algorithm can reduce congestion and node layout deviation. Visual results demonstrate that layout structure of the algorithm is obvious, and nodes are clearly structured and easy to understand.

Keywords: Force-directed layout algorithm; Community radius; Radial tree; Hybrid layout; Complex network

1 引言

计算机技术的快速发展, 让社会关系网络、生

物网络、疾病传播网络等网络分析在现实生活中起到重要的作用. 但要正确地解读和理解这些网络中包含的信息却非常困难. 因此, 如何描绘网络和节

收稿日期: 2019-01-03

基金项目: 国家自然科学基金(61403278)

作者简介: 任淑霞(1973-), 女, 山东梁山人, 副教授, 博士生, 研究领域为数据挖掘, 大数据分析等. E-mail: t_rsx@126.com

点结构,让其便于理解和使用一直是复杂网络布局算法领域研究的热门.

最早对复杂网络进行布局的是 1984 年 Peter Eades 提出的弹力模型^[1]. Eades 的算法虽然简单,但是效率非常低. Fruchterman 和 Reingold 提出 FR(Fruchterman-Reingold)^[2] 布局算法. FR 算法易于实现、降低了计算复杂度. 由于 FR 算法强调节点均匀、边长一致等美学标准,因此上述改进算法均存在阻碍网络社区结构形成的缺陷.

于是,学者开始关注于将聚类方法应用于布局算法的研究, Linlog 模型^[3]是一种计算最小能量来呈现社区结构的模型,虽然它已经实现聚类效果,但没办法证明社区划分的有效性. 吴渝等^[4]提出了社团引力导引的布局算法,该算法不仅为每个节点加入社团引力,还结合 K-means 算法让节点向社团的中心位置聚拢,该算法无需对节点分类,可以在布局的同时对节点聚类. Zhou 等^[5]提出了基于度中心性的社团力导引改进算法,从而完成对复杂网络的聚类布局. 通过观察发现聚类布局算法均能达成社区划分且社区之间没有重叠的效果. 但是,社区内的节点由于太过靠拢,用户很难发现社区内节点之间的关系,不利于用户对社区内部的信息的探索.

基于上述问题,本文提出嵌入社区半径的力引导与径向树混合布局算法 ERRTH (community radius embedded in force-directed and radial tree hybrid layout algorithm),针对 FR 布局算法单纯追求美学标准而不利于发现网络中的社区结构这一问题,本文采用 K-means 算法快速进行社区划分,再用嵌入^[6]社区半径的社区引力和斥力来分离社区,从而完成复杂网络社区结构的展示. 为弥补聚类布局算法在社区内部节点结构展示上的缺陷,本文混合径向树来层次化显示社区内节点结构. 布局结果不仅符合对称性和美学标准,同时也便于用户观察和分析复杂网络中的信息.

2 嵌入社区半径的力引导布局算法

2.1 FR 布局算法

FR 布局算法是一种力引导图布局算法. 它遵循两个简单的原则:有边连接的节点应该相互靠近;节点之间不能离得太近.

FR 算法的每次迭代主要分为两个部分:

1) 计算节点之间的斥力

$$f_r(i, j) = -u^2 / \text{dist}(i, j).$$

2) 计算有边连接的节点之间的吸引力

$$f_a(i, j) = \text{dist}(i, j)^2 / u$$

其中, u 代表两节点之间的理想距离.

最后,综合吸引力和斥力,通过最大位移来限制节点的移动距离. 当两个节点存在覆盖现象时, FR 算法会施加斥力来分开节点. FR 算法在布局时仅仅改变节点位置,避免重叠,并没有分离节点所属的社区,不利于用户的观察. 为达到分离社区的目的,本文提出嵌入社区半径的力引导布局算法.

2.2 嵌入社区半径的力引导布局算法

大量研究显示复杂网络存在社区内连边众多,而社区之间连边较少的特性,因此采用社区划分^[7]来构建社区网络. 当社区数量较多时易产生社区重叠或超出布局区域范围的现象,为解决该问题,本文提出社区半径并定义社区网络的社区斥力和社区引力,同时用社区半径对社区斥力和引力作用范围进行合理限制.

嵌入社区半径的力引导布局算法步骤如下:

(1) 构建社区网络.

采用 K-means 算法^[8],选取 m 个点作为 m 个社区的初始中心点,然后分配节点到最近的中心点所在的社区,重新计算中心点,如果中心点不变,则停止计算;否则继续直到社区中心不变为止. 将复杂网络划 $G(V, E)$ 分出 m 个社区 $\{H_1, H_2, \dots, H_m\}$, 构建社区网络 G' .

(2) 计算社区半径.

定义 1 社区半径 r_i . 社区半径 r_i 是社区 H_i 以社区中心为圆心的圆的半径,社区内节点越多, r_i 越大. 社区半径的计算公式为

$$\begin{cases} M_i = \sum_{i \in H_i} \frac{1}{2} \frac{x \cdot y}{\text{num}(v)} = \pi r_i^2 \\ r_i = \sqrt{\frac{M_i}{\pi}} \end{cases} \quad (1)$$

式中, x 和 y 表示画布的宽和高; $\text{num}(v)$ 表示网络中的节点数量; M_i 表示社区 H_i 的布局区域.

(3) 计算社区斥力.

定义 2 社区斥力 $E(H_A, H_B)$, 设社区斥力 $E(H_A, H_B)$ 是社区网络 G' 中社区 H_A 和社区 H_B 之间存在的一种斥力. $E(H_A, H_B)$ 的计算公式为

$$E(H_A, H_B) = \begin{cases} 0, l(e_{AB}) \geq 2d \\ k \frac{N(H_A)N(H_B)}{dl(e_{AB})^2}, d < l(e_{AB}) < 2d \\ h, l(e_{AB}) \leq d \end{cases} \quad (2)$$

其中, $l(e_{AB})$ 表示 A, B 社区中心的距离; $N(H_A)$ 表示社区 A 所包含的节点数.

社区网络中社区之间存在社区斥力, 社区斥力主要由社区半径来限制作用范围和计算. 由图 1 可以看出, 社区之间的社区斥力主要由 $l(e_{AB})$ 和 d 来确定的.

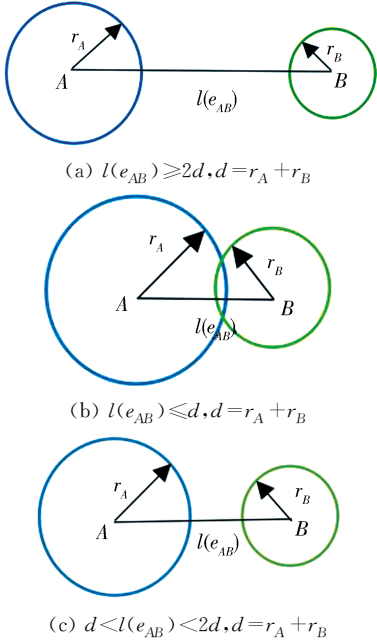


图 1 社区斥力作用范围

Fig. 1 Range of community repulsion

1) 如图 1(a) 所示, 当社区中心距离 $l(e_{AB}) \geq 2d$ 时, 两社区之间应该只有社区引力而没有社区斥力, 此时社区斥力 $E(H_A, H_B) = 0$.

2) 如图 1(b) 所示, 当社区中心距离 $l(e_{AB}) \leq d$ 时, 两社区会发生社区重叠. 因此, 需要施加一个固定常量的斥力 h 来分离两社区, 此时社区斥力 $E(H_A, H_B) = h$.

3) 如图 1(c) 所示, 社区中心距离 $l(e_{AB})$ 位于 d 和 $2d$ 之间. $N(H_A)$ 是社区 H_A 内节点的个数, 开始时社区内的节点越多, 得出社区斥力就越大, 随着 $l(e_{AB})$ 的增大, 两社区之间的社区斥力将变小. 同时, 需要一个斥力系数 k , 方便用户随时调节社区斥力的值, 调整布局结果. 此时社区斥力 $E(H_A, H_B) = k \frac{N(H_A)N(H_B)}{dl(e_{AB})^2}$.

因此, 社区半径能将社区斥力的作用范围限制在 $[0, 2(r_A + r_B))$ 之间, 社区斥力能够分离两社区并阻止它们之间的社区重叠.

4) 计算社区引力.

定义 3 社区引力 $G(H_A, H_B)$, 社区引力

$G(H_A, H_B)$ 是社区网络中社区和社区之间存在的一种引力. $G(H_A, H_B)$ 的计算公式为

$$G(H_A, H_B) = \begin{cases} 0, & l(e_{AB}) \leq d \\ g \frac{|N(H_A) - N(H_B)| l(e_{AB})^2}{d}, & l(e_{AB}) > d \end{cases} \quad (3)$$

社区网络中不仅存在社区斥力, 还存在社区引力, 也由社区半径来限制其作用范围和计算的. 如图 2 所示.

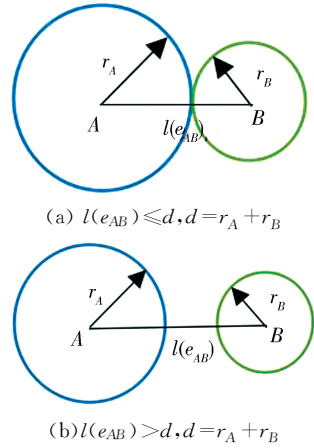


图 2 社区引力作用范围

Fig. 2 Range of community gravitation

1) 如图 2(a) 所示, 当社区中心距离 $l(e_{AB}) = d$ 时, 如果两社区存在社区引力, 那么它们将会发生社区重叠, 这时只有社区斥力而无社区引力. 因此, 当 $l(e_{AB}) \leq d$ 时, 社区引力 $G(H_A, H_B) = 0$.

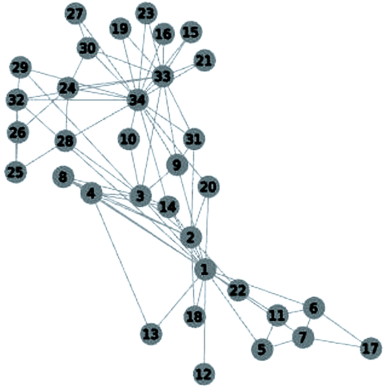
2) 如图 2(b) 所示, 当社区中心距离 $l(e_{AB}) > d$ 时, 两社区之间就会存在社区引力. 开始时, 社区引力与两个社区所包含的节点数之差成正比. 随着 $l(e_{AB})$ 增大, 社区引力将会变大, 而社区斥力将变小, 社区之间将会吸引来解决超出布局区域的问题. 同时, 需要存在一个引力系数 g , 方便用户随时调节社区引力的值来调整布局结果, 此时社区引力为

$$G(H_A, H_B) = g \frac{|N(H_A) - N(H_B)| l(e_{AB})^2}{d} \quad (4)$$

因此, 可得出社区引力的作用范围为 $(r_A + r_B, \infty)$, 社区引力能解决两社区相距太远而超出布局区域的问题.

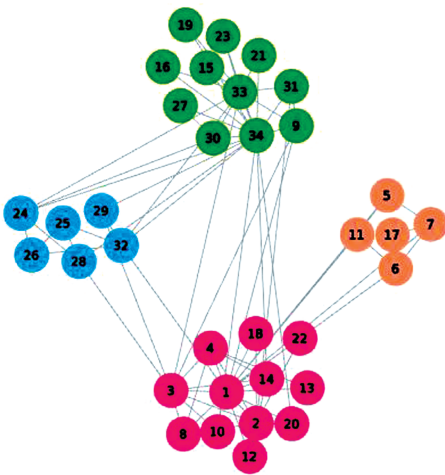
嵌入社区半径的力引导算法在 Karate 数据集的布局结果如图 3(b) 所示. 较之 FR 算法, 可以看出复杂网络被划分成不同社区, 社区之间相互远

离,社区网络的结构明显.但社区内节点相互拥挤在一起,不利于用户观察社区内结构特征与连边关系.因此,本文采用径向树布局排列各社区内节点,让节点满足层次结构布局的需求.



(a) FR 算法的布局结果

(a) FR algorithm layout result



(b) 嵌入社区半径的力引导算法的布局结果

(b) Layout result of community radius embedded in Force-Directed algorithm

图 3 布局结果对比图

Fig. 3 Layout result comparison graph

3 混合径向树的社区半径力引导算法

3.1 径向树

树图^[9]关注自动生成关系信息的几何表示,通常用于分层可视化.用于对分层信息进行建模的典型数据结构是顶点表示实体并且其连边对应于实体之间关系的树.层次结构布局可以向用户有效的传达信息.径向树^[10-11]作为树图的分支,在展示节点结构层次有着独特优势.

3.2 采用径向树排列社区节点

为直观展示社区内节点层次结构,本文采用径向树对社区内的节点进行排列,使所有节点都位于社区

中心聚焦的同心圆上.径向树排列节点的步骤如下.

(1) 如要快速搜索出社区内节点的全部层次结构 L ,则采用 DFS(深度搜索算法)来而不采用 BFS(广度搜索算法),并从社区内随机选出节点作为根节点放置在社区中心,设各级之间的距离 D 是将社区半径除以社区中的级数 L 来计算的.

(2) 将根节点放置在社区中心,此节点是下一级中其他节点的父节点.由于 1 级节点具有相同的父节点,可以分布在整个 2π 上.因此计算角度时,将 2π 除以 1 级节点的数量,这会在 1 级节点之间创建角度空间,然后遍历 1 级节点,使用式(5)计算 1 级节点的位置.

$$\begin{cases} N_x = D * \sin(N_{level} * A_s) \\ N_y = D * \cos(N_{level} * A_s) \end{cases} \quad (5)$$

其中, N_{level} 是层级内的节点索引; A_s 是角度空间.

(3) 计算切线角度.

定义 4 层级半径.节点所在的层级到根节点的距离叫做层级半径.

节点 N_0 的子节点必须落在其所属层级的一定范围内以防止重叠,切线垂直于 N_0 节点与根节点的连线,切线与子节点所在层级相交的点为切线切点.切线切点与 N_0 节点的父节点(根节点)连线的夹角作为切线角度,使用式(6)来计算切线角度.

$$\tan \text{An}(N_0) = 2 \arccos(R_{out}/R_{in}) \quad (6)$$

式中, R_{in} 是子节点所对应的层级半径; R_{out} 是父节点所对应的层级半径.

(4) 计算等分线角度.节点 N_0 计算等分线角度公式如下.

$$\text{BisAn}(N_0) = \frac{\text{Arc}(N_0, N_1) + \text{Arc}(N_0, N_2)}{2} \quad (7)$$

其中, N_1, N_2 是 N_0 的相邻节点; $\text{Arc}(N_0, N_1)$ 为图 4 中 N_0 与 N_1 的夹角.子节点所能分配的范围为等分线角度与切线角度的交集所对应的圆弧上,这能避免子节点分配重叠的现象.子节点的分配范围如图 4 所示.

(5) 在 2 级或更高级别上,由于父节点限制子节点的位置,首先得到待分配子节点的父节点集合.遍历该集合,得到父节点的子节点列表,明确子节点的分配范围,放置子节点在分配范围对应的圆弧上.

ERRTH 算法的最终布局结果如图 5 所示,较之图 3 展示的布局结果,可以看出社区内节点层次结构分明,节点之间拥挤程度不高.其中红色节点为根节点,其余节点分布在以根节点为圆心的同心

圆上, 布局结果明显, 且符合对称性和美学标准。

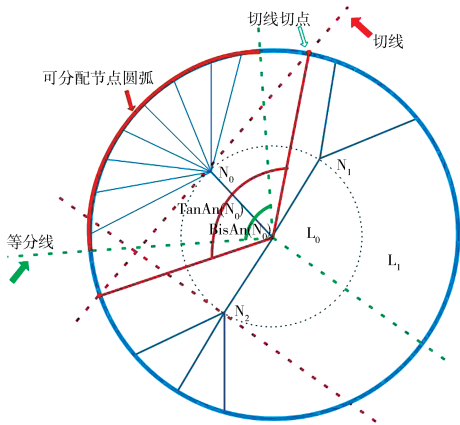


图 4 子节点的分配范围示意图

Fig. 4 Child node allocation range graph

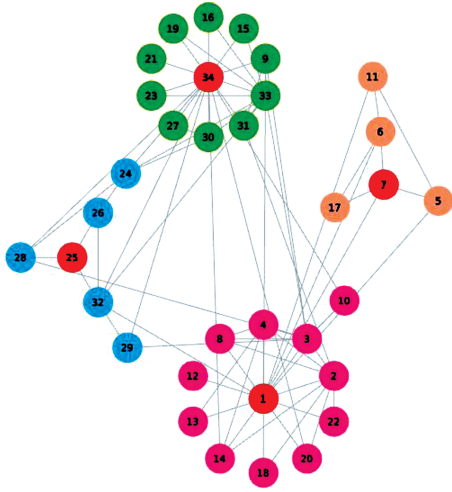


图 5 ERRTH 算法布局结果

Fig. 5 The layout result of ERRTH algorithm

3.3 ERRTH 算法步骤

ERRTH 算法步骤如算法 1 所示。

算法 1 ERRTH 算法

输入: 网络 $G(V, E)$, 最大迭代次数 N , 最大位移 M 。

输出: 网络节点位置坐标 $\{D_1, D_2, \dots, D_n\}$ 。

Begin

1) 初始化节点位置, 为节点生成随机的坐标。

2) 用 K-means 将网络 G 划分出 m 个社区, 构建社区网络, 确定各社区节点数量, 依据式(1)计算出社区半径。

3) 依据式(2)和式(3)计算社区网络中社区之间社区引力和斥力, 综合社区引力和斥力, 用最大位移 M 限制社区的最大移动距离, 社区所属节点随着各社区中心移动。

4) 若迭代次数大于最大迭代次数, 执行步骤 5), 否则, 循环步骤 3)。

5) 对于社区内节点, 用 DFS 算法明确各社区层级, 然后通过社区半径计算出级间距离, 放置根节点。

6) 对于社区内 1 级节点, 依据式(5)计算 1 级节点位置。

7) 对于社区内 2 级或更高级别的节点, 依据式(6)确定切线角度和式(7)确定等分线角度, 明确切线角度和等分线角度的交集范围, 放置节点到交集范围所对应的圆弧上, 循环放社区内节点至最后一层即可。

8) 输出节点位置 $\{D_1, D_2, \dots, D_n\}$ 。

End.

4 实验结果分析

为证明算法的合理性和有效性, 本文选用复杂网络中的 Dolphins^[12], Karate^[13], Football^[5] 网络来展示复杂网络社区结构, 选取 FR 算法和朱志良等人算法^[14] 的布局结果进行对比。为定量分析本文算法的合理性, 将 ERRTH 算法与 FR 算法、CGDA 算法^[13]、文献[14]算法和文献[5]算法采用相同数据集进行对比, 并用点分布方差^[14]、拥挤区域占比、边长偏差^[15]、节点分布偏差^[16] 和时间进行实验分析。实验的运行环境是 intel(R) Core (TM) 2 Quad CPU Q8300@2.50 GHz、内存为 16 G、64 位 Win10 的 PC。

4.1 布局结果分析

(1) Dolphins 网络是依据生活在新西兰的 62 只海豚而创建的社会关系网络, 该网络包含 62 个节点和 159 条边。

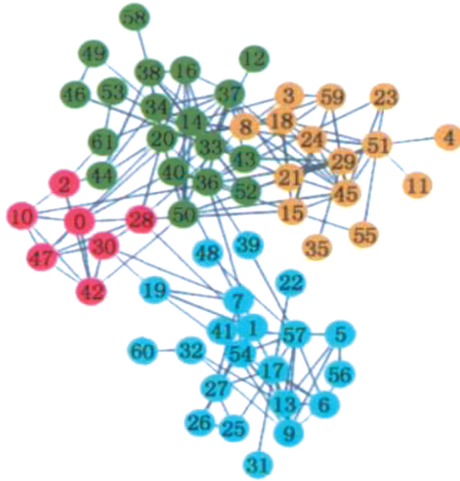
ERRTH 算法与朱志良等人的算法在 Dolphins 数据集上的布局结果如图 6 所示。图 6(a) 是文献[14]的可视化布局结果, 该算法虽然阻止不同社区节点的交错, 但社区位置过于靠近, 网络变得非常拥挤, 若没有颜色加以区分, 用户无法观察出社区结构。而图 6(b) 可以看出 ERRTH 算法不仅能够阻止不同社区的覆盖, 而且社区之间相互远离且层次结构明显。两者比较, ERRTH 算法能清晰地展示网络的社区结构和社区层次结构。

(2) Football 网络数据集, 该数据集表示美国大学生足球俱乐部比赛的网络, 该数据集包含 115 个节点和 616 条边。

图 7 为 Football 数据集的布局结果, 由于数据集节点与边规模比较大, 因此选择适当缩小节点来

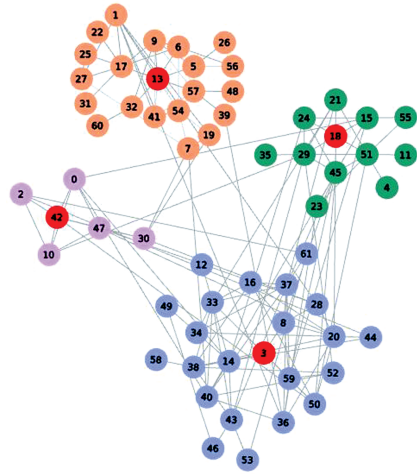
满足布局要求. 从图 7(a)可以看出 FR 算法已经不能满足复杂网络展示和美学需求, 节点之间层次结构, 用户很难明确社区结构和节点层次特征. 为了

能够在图 7(b)中清晰地观察社区结构, 本文选择放大蓝色方框内的布局结果如图 8 所示.



(a) 文献[14]的布局结果

(a) The layout result of document [14]

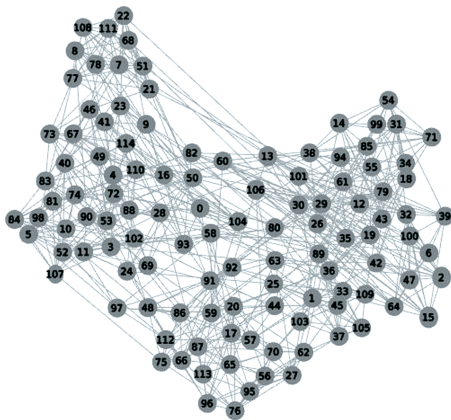


(b) ERRTH 算法布局结果

(b) The layout result of ERRTH

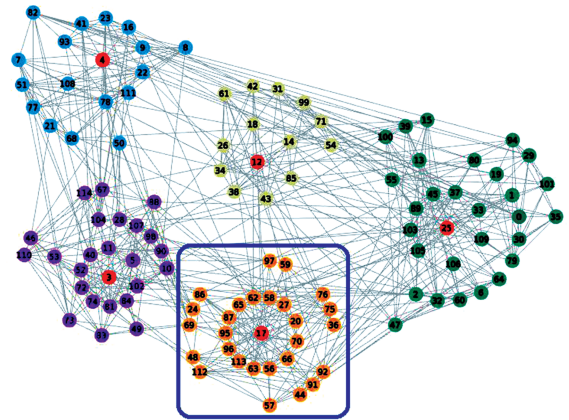
图 6 Dolphins 布局对比图

Fig. 6 Comparison graph of Dolphins layout



(a) FR 算法布局结果

(a) FR algorithm layout result

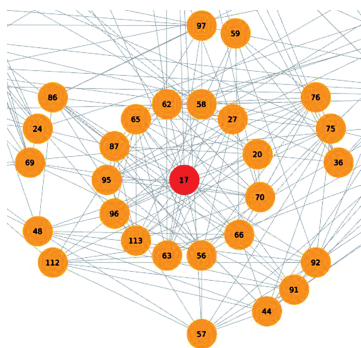


(b) ERRTH 算法布局结果

(b) The layout result of ERRTH

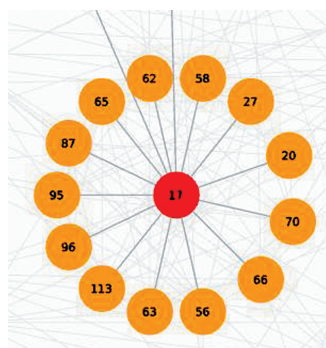
图 7 Football 布局对比图

Fig. 7 Comparison graph of Football layout



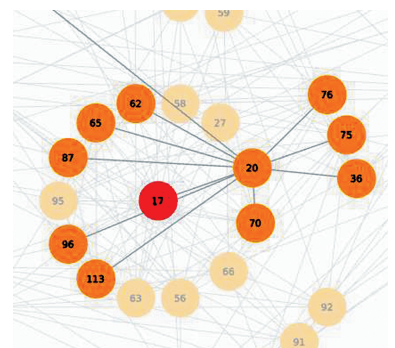
(a) 所有节点排列图

(a) All node arrangement graph



(b) 1级节点排列图

(b) 1-level node arrangement graph



(c) 节点 20 的层次连接图

(c) The level connection graph of node 20

图 8 蓝色区域节点层次连接关系图

Fig. 8 The node level connection graph of blue area

图 8(b)可以清晰显示根节点 17 与下一层级的 20、70 等节点相连,而图 8(c)节点 20 与相同层级的 62、65、87、70 等节点和下一层级的 76、75、36 等节点相连,用户分析网络时可以优先着眼于社区内部的关键信息,然后再与不同社区的节点联合分析.从而提高用户获得复杂网络关键信息的效率,方便用户的使用.最后,综合图 7(a)和(b)可以看出 ERRTH 算法不仅能够分离不同的社区,还能展示社区内的层次结构和连边关系,便于用户探索复杂网络的信息.

4.2 算法效率分析

4.2.1 时间复杂度分析

ERRTH 算法由嵌入社区半径的力引导和径向树布局算法两部分组成.针对嵌入社区半径的力引导布局算法,在每一次迭代过程中,社区网络中的 k 个社区都要计算各自的社区引力和斥力,社区内的节点 v' 采用引力的方式随着社区中心移动,所以这部分的时间复杂度为 $O(k^2 + k|v'|)$.针对径向树布局,每个社区内的节点都采用 DFS 算法明确节点层级结构.循环节点层级结构 L ,依据父节点 v' 来确定可分配节点的圆弧并放置相应的子节点 v^n ,径向树的时间复杂度为 $O(|v'|^2 + L|v'| |v^n|)$.由此可得,ERRTH 算法的时间复杂度为 $O(k^2 + k|v'|) + O(|v'|^2 + L|v'| |v^n|)$.

4.2.2 实验指标分析

拥挤度^[17]常被用于衡量集成电路布局设计是否合理的,拥挤度需要统计不同区域边缘的走线数 Se 和固定边缘走线数 De ,但在复杂网络中很难统计边缘走线数 Se 和 De ,因此不适用对复杂网络的布局结果进行评价.为对复杂网络的布局效果进行衡量,本文以区域内节点的数量来判定区域是否为拥挤区域,并提出拥挤区域占比来评价不同算法下的复杂网络布局效果.

定义 5 平均节点拥有量(avg_region).每个区域内拥有的平均节点数量,记为 avg_region .

$$\begin{cases} region = s \cdot t \\ avg_region = \frac{num(v)}{region} \end{cases} \quad (8)$$

式中, $num(v)$ 表示节点总数; $region$ 表示区域总数; s 和 t 表示布局图被划分出的行、列数.

定义 6 拥挤区域($crowd_area$).若划分进某区域内的节点数量高于平均节点拥有量,则此区域被认定拥挤区域,记为 $crowd_area$.

定义 7 拥挤区域占比($crowd_area_rate$).拥挤区域的数量占有所有区域的比例,记为 $crowd_are-$

a_rate .

$$crowd_area_rate = \frac{num(crowd_area)}{region} \quad (9)$$

如图 9 所示,给区域分配节点时,部分节点位于区域的边缘上,为将这部分节点划分至指定区域,本文提出一个“左部优先,上部优先”的节点划分策略.节点 1 位于 B, C 两个区域,按照“左部优先”原则,节点 1 划分进了 B 区域.同样,节点 3 因“上部优先”原则被划分进 E 区域.但少数特殊节点如节点 2,它位于区域 A, B, D, E 的边缘交叉处,按照划分策略,它会被划分进区域 A .循环处理边缘节点至每个节点都被分配到相应区域,并依据式(9)计算出拥挤区域占比.拥挤区域占比越高,说明复杂网络的部分区域相对较拥挤,节点过于密集,不利于用户观察复杂网络.

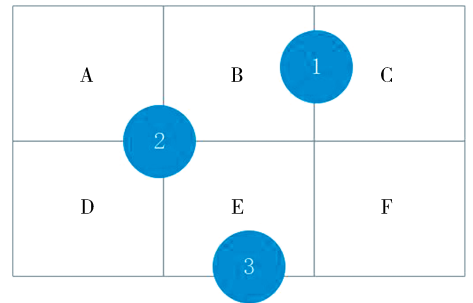


图 9 区域节点划分策略

Fig. 9 Regional node partitioning strategy

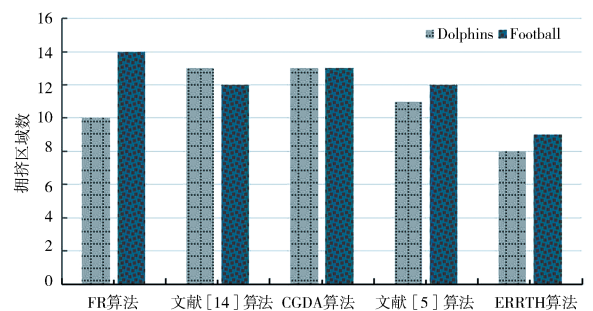


图 10 不同算法的拥挤区域统计图

Fig. 10 Crowded area statistics of different algorithms

实验中的指标数据均采用 python 语言编程获得,实验结果分别如表 1 和表 2 所示.

由表 1 和表 2 的实验结果,以及图 10 分析得出,在相同的数据集下,采用径向树的 ERRTH 算法产生的拥挤区域明显少于其他布局算法,说明 ERRTH 算法可以有效减少复杂网络中的拥挤区域.再从点分布方差来看,ERRTH 算法布局后的节点分布更为合理,可以降低节点拥挤导致的布局

混乱和区域内节点的密集程度。

ERRTH 算法的边长偏差和节点分布偏差低于其他布局算法,说明 ERRTH 算法相较于其他算法社区内的节点布局偏差较少,节点分布结果更均匀,更符合复杂网络布局的美学标。ERRTH 算法为达到明确社区内节点层次结构的目的而采用径向树进行二次布局,这部分占用时间过多导致 ERRTH 算法的时间略高于其他算法,但算法之间差距不大,并且尚在用户可容忍时间范围 2~8 s 之内。

表 1 Dolphins 网络的实验结果

Tab. 1 Experimental results of the Dolphins network

实验参数	FR 算法	文献[14] 算法	CGDA 算法	文献[5] 算法	ERRTH 算法
点分布方差	0.3882	0.4885	0.4822	0.4422	0.3326
拥挤区域占比	0.40	0.52	0.52	0.44	0.28
边长偏差	0.1752	0.1490	0.1551	0.1670	0.1460
节点分布偏差	0.1609	0.1608	0.1607	0.1605	0.1602
时间/s	1.274	1.648	1.241	1.524	1.715

表 2 Football 网络的实验结果

Tab. 2 Experimental results of the Football network

实验参数	FR 算法	文献[14] 算法	CGDA 算法	文献[5] 算法	ERRTH 算法
点分布方差	0.7074	0.6586	0.6771	0.6599	0.6001
拥挤区域占比	0.56	0.48	0.52	0.48	0.36
边长偏差	0.3690	0.3751	0.3467	0.3542	0.3216
节点分布偏差	0.8686	0.8745	0.8681	0.8680	0.8675
时间/s	5.329	7.952	5.259	7.173	8.882

此外,本文还采用 ERRTH 算法对线虫的神经网络(299 个节点,2 359 条边,数据集来源于 Newman 教授的个人网站)进行布局,运行时间稳定在 16 s 左右,拥挤区域的占比为 0.4,远低于其他布局算法。可以推出 ERRTH 算法尤其适用于 500 个以下节点的布局,布局后的拥挤区域较少,节点分布较为合理。

最终从布局分析和算法效率分析的结果来看,ERRTH 算法可以显著地分离两个社区,且高效地展示各个社区内网络层次结构,提高复杂网络社区结构的辨识度,对于网络布局较之传统布局有着较强可读性和可解释性,具有很强的使用价值。

5 结 论

本文基于 FR 算法,提出嵌入社区半径的社区

斥力和引力来分离复杂网络的社区,接着又采用径向树来排列各个社区内节点位置。前者可以避免各社区的聚拢和重叠,后者可以直观展示各个社区内层次结构,便于用户理解社区内节点之间的层次关系。与力引导算法和聚类布局算法相比,ERRTH 算法可避免力引导算法不能分离社区以及现有的聚类布局算法不能显示社区内部节点结构关系的缺陷。

在对复杂网络进行社区划分时,网络中可能会存在一些同时与几个社区有联系,但又不属于任何社区的节点,它常位于社区的边缘区域被称为边缘节点。由于本文采用 Kmeans 算法来划分社区,该方法不能准确识别边缘节点和需要先给出聚类数目 m 。因此,在下一步工作中,将选用能识别边缘节点的社区划分算法来替代 Kmeans 算法,同时将本文布局算法运用到时变复杂网络^[18]分析上去,使时变网络在布局时能保持网络结构和心智图^[19]的稳定。

参考文献:

- [1] Peter E. A heuristic for graph drawing [J]. Congr Numeranti, 1984, 42: 149.
- [2] Fruchterman T M J, Reingold E M. Graph drawing by force-directed placement [J]. Softw-Pract Experi, 1991, 21: 1129.
- [3] Noack A. Energy models for graph clustering [J]. J Graph Algorithms Appl, 2008, 11: 453.
- [4] Yu W, Li Z, Li H, et al. A community gravity directed algorithm for showing community structure of complex networks[J]. J Comput-Aide Des Comput Graph, 2015, 27: 1462.
- [5] Zhou X, Huang T L, Ling X. Network clustering layout algorithm based on detecting community structure [J]. Comput Moderniz, 2017, 2017: 1.
- [6] 汤颖, 汪斌, 范菁. 节点属性嵌入的改进图布局算法[J]. 计算机辅助设计与图形学学报, 2016, 28: 228.
- [7] 刘苗苗, 郭景峰, 马晓阳. 基于共邻节点相似度的加权网络社区发现方法[J]. 四川大学学报:自然科学版, 2018, 55: 89.
- [8] Zhong S, Lin S, Xu G. The expansibility research of K-Means algorithm under the GPU [C]//Proceedings of the International Conference on Software Engineering and Service Science. Beijing: IEEE, 2017.
- [9] Dang T, Forbes A. Cactus tree: a tree drawing approach for hierarchical edge bundling [C]// Pacific

- Visualization Symposium. Seoul, South Korea: IEEE, 2017.
- [10] Raj M, Whitaker R T. Anisotropic radial layout for visualizing centrality and structure in graphs [C]// Proceedings of the International Symposium on Graph Drawing and Network Visualization. Boston, MA, USA: Springer, 2017.
- [11] Brandes U, Pich C. More flexible radial layout [J]. J Graph Algor Appl, 2011, 15: 107.
- [12] Lusseau D, Schneider K, Boisseau O J. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations [J]. Behav Ecol Sociobiol, 2003, 54: 396.
- [13] 吴渝, 李藻旭, 李红波. 展示复杂网络社团结构的社团引力引导的布局算法[J]. 计算机辅助设计与图形学学报, 2015, 27: 1460.
- [14] 朱志良, 林森, 崔坤, 基于复杂网络社区划分的网络拓扑结构可视化布局算法[J]. 计算机辅助设计与图形学学报, 2011, 23: 1808.
- [15] Shen X, Han D, Zang G. 3D visualization layout algorithm for concept lattice based on virtual node [J]. J Henan University, 2018(04): 421.
- [16] 李海峰. 图布局力引导算法的研究与实现[D]. 镇江: 江苏大学, 2012.
- [17] Li Z, Wu W, Hong X. Incremental placement algorithm for timing and routability optimization [J]. Chinese J Semiconduct, 2004, 25: 158.
- [18] Wang X, Song H. A dynamic network layout visualization method based on structural similarity [C]//Proceedings of the 2016 International Conference on Cyber-worlds. Chongqing: IEEE, 2016.
- [19] Archambault D, Purchase H C. The “Map” in the mental map: experimental results in dynamic graph drawing [J]. Int J Humput St, 2013, 71: 1044.

引用本文格式:

中文: 任淑霞, 吴涛, 张书博. 嵌入社区半径的力引导与径向树混合布局算法[J]. 四川大学学报: 自然科学版, 2020, 57: 73.

英文: Ren S X, WU T, Zhang S B. Community radius embedded in force-directed and radial tree hybrid layout algorithm [J]. J Sichuan Univ; Nat Sci Ed, 2020, 57: 73.