基于复杂网络社区划分的网络拓扑结构可视化布局算法

朱志良，林 森，崔 坤，于 海
（东北大学软件学院 沈阳 110819）
（ls_ghost@yahoo.cn）

摘 要：许多真实的网络都可以用复杂网络的思想进行研究和解释，而社区结构是复杂网络的一个重要特征.为此，提出一种基于社区结构的网络布局算法.首先利用复杂网络社区发现算法对网络中的节点进行社区划分，并将一个社区抽象为一个节点，以社区间的关联为边构建新的网络；在此基础上，运用物理类比方法确定社区中心点的位置，并根据社区的规模确定社区的区域范围；最后运用条件择优的方式填充社区内部节点以完成网络拓扑的布局.仿真实验结果证明，该算法与传统的可视化布局算法相比，具有计算量更少、收敛速度快、结构清晰的特点，更具有实际应用的价值.

关键词：数据可视化；复杂网络；社区发现；布局算法
中图法分类号：TP 309.7

Network Topology Layout Algorithm Based on Community Detection of Complex Networks

Zhu Zhiliang, Lin Sen, Cui Kun, and Yu Hai
(*Software College, Northeastern University, Shenyang* 110819)

Abstract: Many real world networks can be studied and interpreted through the theory of complex network, and one important characteristic of it is a community structure. Based on the community structure of the network, a new network topology layout algorithm is proposed. The network is divided into several communities by community detecting algorithms. Furthermore, the new community network is constructed by nodes and edges abstracted from the communities and the correlations among communities respectively. Based on above, it performs physical analogy topology algorithm to determine the center position of each community and the area scope in terms of its scale. Finally it uses the conditional optimized method to fill up each community intern nodes in order to complete the layout of network topology. The simulation result shows that the proposed algorithm is more efficient, and it provides a better convergence rate, a more clear structure, and a more practical application value.

Key words: data visualization; complex network; community detection; layout algorithm

---

收稿日期：2011-03-30；修回日期：2011-06-20.基金项目：国家自然科学基金资助项目(60872040)；辽宁省自然科学基金资助项目(20082037)；辽宁省高等学校优秀人才支持计划资助项目；中央高校基本科研业务费资助项目(N100604007).朱志良(1962—)，男，博士，教授，博士生导师，CCF 会员，主要研究方向为混沌分形理论与应用、多媒体信息处理；林 森(1987—)，男，硕士，主要研究方向为复杂网络、并行计算；崔 坤(1986—)，男，硕士，主要研究方向为混沌保密通信、复杂网络；于 海(1971—)，男，博士，副教授，主要研究方向为混沌分形理论与应用、多媒体信息处理.

随着计算机技术的不断发展，网络在现实生活中起到了越来越重要的作用. 现实生活中的网络往往具有规模庞大、关系复杂等特点. 要正确地解读和理解网络中蕴含的复杂信息，往往需要经历从获取信息到抽取有用信息等过程. 对于这些抽取获得的有用信息，如何直观、高效地分析和研究这些复杂数据，已成为人们所热切关注的问题. 网络可视化模型研究正好满足了信息可视化的需求，它可以将上述复杂信息抽象为以个体为节点、个体之间关系为边的网络模型. 目前，网络的可视化已成为一种非常有用的信息表示方式.

在实际应用中，随着网络规模的扩大，网络中会出现大量的节点和边，从而导致可视化模型出现点覆盖、边交叉以及图形拥塞等现象，这将大大影响数据的可视化效果. 因而，如何解决这一问题是网络可视化模型研究的关键问题之一. 目前，解决这一问题通常是靠网络拓扑结构的布局算法来实现的，它主要使用的是物理类比方法. 使用该物理类比方法对网络进行布局的历史可以追溯到 1984 年由 Peter Eades 提出的弹力模型[1]，它将每个顶点看作为一个钢圈，顶点间的边看作是它们之间的弹簧. 开始时，顶点的位置是随机放置的；然后让弹力随机作用在顶点上，直到进行一定次数的迭代后算法停止. 对于不同的网络，该算法具有不同的收敛速度. 为此，Kamada 等[2]提出了对该模型的一种改进——KK 模型. KK 模型使布局算法的收敛速度有了明显的增加，但仍未达到令人满意的程度. 随后，Davidson 等[3]在其模型(简称 DH 模型)中首次提出了能量函数的思想，它将每一个布局形式赋予一个能量，较低的能量代表着更好的布局. DH 模型算法通过优化能量值来取得全局最优的布局，但其非常耗时.

在此基础上，Fruchteman 等[4]提出的一种经典的图布局算法——FR 算法，该它是在弹力模型的基础上做了一定的改进和优化. 在 FR 模型中，节点被看作是原子或天体. 根据物理学原理，原子内部粒子或是天体之间彼此会发生引力和斥力作用，这些力的作用将导致物体的运动，将物体拉近或推远. 与弹力模型不同的是，相邻的节点会彼此产生引力的作用，同时所有这些节点彼此之间还会产生斥力的作用，引力和斥力作用在不同的节点上，最终会达到一个平衡状态. FR 模型算法中还引入了模拟退火[5]的过程，其在每一次迭代的过程中都经历了 3 个步骤：1)计算所有节点来自邻居节点的引力的大小；2)计算节点间彼此斥力的大小；3)节点会在合力的作用下运动，而运动的幅度和频率会受到“温度”的限制，温度会随着迭代过程而下降，完成“退火”的过程，使节点最终达到静止，以完成图的自动布局.

FR 算法适用于结构简单、节点数少的简单网络，它具有易于实现、布局效果好等优点；且其每次迭代只需计算所有相连顶点间的吸引力和所有顶点间的斥力，因而降低了计算复杂度，是现在常用的网络布局算法之一. 在此之后的出现了对 FR 算法的改进算法一般期望最大化(general expectation maximization，GEM)算法[6]和 LinLog 算法[7]，近几年所提出的基于斥力张力模型的布局算法[8]、基于扩展力学模型的网络拓扑图布局算法[9]都在一定程度上借鉴了 FR 算法的思想，同时期提出的力导引布局算法[10]，快速多尺度(fast multi-scale，FMS)[11]算法以及快速多级力导引布局算法[12]也都是物理类比的布局算法. 此外，还有代数多网格布局(algebraic multigrid computation of eigenvectors，ACE)[13]算法和高维度嵌入布局(high-dimensional embedding，HDE)算法[14]都是从非物理类比的角度来对图进行布局的.

近年来人们对与拓扑算法研究的应用领域和适用场景大多数比较集中于对电子硬件元件的布局与布线算法，如对现场可编程门阵列(field programmable gate array，FPGA)的布局研究[15]或是芯片布线的研究[16]，可检索到的对网络拓扑结构的研究成果比较少. 与此同时，随着复杂网络的相关研究取得了迅猛发展，给网络布局算法提供了新的研究切入点，其不再局限于从物理类比的角度思考网络布局问题. 复杂网络是对复杂系统的高度抽象，其中许多性质如小世界性、无标度性、聚集性等均已得到了充分地研究. 真实世界中的许多复杂系统都可以表示成图或网络，包括社会网络、信息网络、生物网络和技术网络等. 分析表明，这些网络可以自然地分成一些节点组，同一个节点组内的 2 个节点之间比不同节点组的 2 个节点之间更倾向于有边相连，网络的这种拓扑特性被称为社区结构[17].

社区结构能够刻画网络中的连边关系的局部聚集特性，而对连边之间位置的布局正是网络图布局算法的难点和需要解决的问题. 一个好的布局算法应该能够揭示复杂的数据内涵，方便观察者理解网络的内在结构. 为此，本文在布局前引入社区划分这个步骤，经过划分社区的网络，连边的局部特性更能够得到细致的刻画，使在此基础上对网络进行布局会收到更好的效果. 由于真实世界中的网络普遍具有

社区结构的特性，即使没有社区特性，网络的节点也具有一定的聚类特性[18]，同样可以运用社区发现算法[19-22]对这些网络进行预处理. 因此，本文算法具有良好的适用性和普遍性.

1 算法思想及其设计实现

1.1 算法思想

设 G 为一个网络，则网络 G 可以表示为 $G(V, E)$，其中 V 为网络 G 中 n 个顶点的集合 $\{v_1, v_2, \cdots, v_n\}$，$E$ 为 m 条边的集合 $\{e_1, e_2, \cdots, e_m\}$；则网络拓扑可视化布局算法的目标是得到一个结构清晰、布局合理的拓扑图 G'[23]. G' 可以用如下4个指标来衡量：

1) 拥挤度. 合理的网络拓扑应当能够尽量避免或减少拥挤，而造成拥挤的原因是由于局部边过多. 本文使用了内外拥挤度估计函数[16]来评估一个网络拓扑布局的拥挤情况，它能够有效地检测出全部局部的拥挤区域并衡量全局的拥挤程度.

2) 边交叉数. 显然拓扑图 G' 的边交叉数越少，图看起来越清晰、布局越好[24].

3) 节点分布情况. 节点的分布应该尽可能地利用空间. 假设网络被分为大小相同的 n 个矩形区域，若记第 i 个区域中节点的数量为 x_i，所有区域内节点数量的平均值为 $\bar{x}$，则方差 $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ 应当尽可能地小[24].

4) 有连边的节点之间的相对位置：若一个节点有 n 个相邻节点，则这 n 个节点应当在空间上与该节点尽可能地相邻. 这样，点之间的关系才更清晰，同时也能够有效地减少边的交叉.

大量的研究表明，许多现实中的网络都可以用复杂网络的方式进行建模，即许多真实网络的特性都可以用复杂网络的理论进行解释和描述. 为了满足上述网络拓扑可视化布局算法的指标，可以运用社区发现算法将网络划分为的不同社区. 运用复杂网络社区的性质(社区内连边众多，而社区间连边相对较少)，因此可以首先将社区抽象为节点，社区间的关联为边构建社区网络；然后在这个网络的基础上运用传统的物理类比的方法对社区节点进行布局. 这样，在发挥了传统方法布局效果好、结构清晰的优点的同时，由于节点数少，又能够有效地减少运算时间，避免了在节点数过多的情况下算法收敛速度慢的情况. 之后，根据社区的规模确定不同社区的区域，并运用一定的条件择优算法填充内部节点，以完成网络拓扑图的布局.

1.2 算法设计

算法主要分为4个步骤：

Step1. 构建以社区为节点的网络.

为了得到良好的图布局效果，FR算法有两点约束，即有连线的节点之间需要在位置上邻近并且节点应避免过分地靠近[4].

由于社区内部的节点连边众多，出于社区连边的特点和对良好布局约束特征的考虑，可以将整个社区看作一个节点来构建新的网络进行布局，并在此布局的基础上确定每个社区的区域以填充社区内部邻近的节点. 这样做不仅符合良好图布局的规范，同时简化了后续的布局工作.

Step2. 运用物理类比方法对社区网络进行布局.

借鉴了FR算法的思想，根据物理学动量守恒定律，系统在不受外力的作用下，系统内部节点仅受到来自节点间内力的作用，系统的总动量保持不变. 这里系统初始动量为零，系统内力可看作是来自节点间的斥力和边对两端节点的张力作用. 那么经过一段时间以及内力作用后，系统内部节点会趋于静止或保持震荡. 利用这一性质，可完成对社区中心位置的布局.

系统内的物理量包括系统内的节点质量以及节点间的斥力与张力. 设 x 为网络 G 中的节点，则其质量可以用 $m(x)$ 表示，若 C_i 为网络中的第 i 个社区，则其抽象为节点后的质量 $m(c_i) = \sum_{i \neq j} P(e_{ij})$；其中 $P(x)$ 表示边 x 的权值，e_{ij} 表示一端节点为 c_i 的边.

根据FR算法，系统内节点间有斥力存在. 设 $R(x, y)$ 表示节点 x 与 y 之间的斥力，存在常量 f 和 g，C_i 和 C_j 分别为网络中的第 i 个和第 j 个社区，其抽象为节点后的斥力为

$$R(c_i, c_j) = \begin{cases} f, & d = 0 \\ \dfrac{g \cdot m(c_i) \cdot m(c_j)}{d^2}, & d \neq 0 \end{cases} \tag{1}$$

其中 $d = \|c_i - c_j\|$，d 为节点 c_i 与 c_j 的欧几里德距离. 通常，斥力越大，图的布局就会越稀疏. 这里，参数 f 和 g 需要根据实际网络数据的规模和画布的大小进行调节.

同理，边对节点的张力可以表示为

$$T(c_i, c_j, e_{ij}) = \begin{cases} 0, & l(e_{ij}) > d \\ k(d - l(e_{ij})), & l(e_{ij}) \leqslant d \end{cases} \tag{2}$$

其中 k 为弹性系数，C_i 和 C_j 为网络中的第 i 个和第

j 个社区，$l(e_{ij})$ 表示边 e_{ij} 的长度，且式(2)遵循胡克定律.

Step3. 确定社区区域范围.

经过前 2 步处理，新网络中每个节点的坐标对应着该节点所代表社区的中心点的坐标. 确定了社区中心节点的位置后，需要为每个社区分配一个区域以填充社区内部节点. 可以考虑使该区域为一个以中心节点为圆心，r 为半径的圆；为了使节点布置均匀，r 的确定需要与社区的规模相关联. 设 S_i 和 S_j 为第 i 个和第 j 个社区的区域面积，n_i 和 n_j 分别为其中的节点数量，则满足

$$\frac{S_i}{S_j}=\frac{n_i}{n_j}=\frac{\pi r_i^2}{\pi r_j^2} \tag{3}$$

Step4. 填充社区内部节点.

在确定社区中心节点和半径的基础上，对社区内部节点进行填充. 这里需要将社区内部的节点分为边缘节点和非边缘节点 2 类. 大体来说，边缘节点就是存在跨社区连边的节点，非边缘节点是指一个节点的所有连边的目标节点与该节点同属于一个社区. 对二者的具体定义将在算法的实现中进行说明. 为了使填充后图的整体结构清晰、布局均匀、线交叉数少，填充时边缘节点需要满足以下 3 个约束：

1) 边缘节点应尽量位于所在社区和与它有连接的其他社区的平面区域之间.

具体地，设 v_i 为社区 C_i 的第 i 个边缘节点，若社区 C_1，C_2，…，C_k 为 k 个与 v_i 有连接的社区，且其中心节点分别为 c_1，c_2，…，c_k，定义 $\boldsymbol{\alpha}_{ij}$，$j\in[1,2,\cdots,k]$ 表示从社区 C_i 的中心节点 c_i 到社区 C_1，C_2，…，C_k 的中心节点的矢量，则

$$\boldsymbol{\alpha}_i=\sum_{j=1}^{k}\boldsymbol{\alpha}_{ij} \tag{4}$$

设 $\boldsymbol{\beta}$ 为点 v_i 到中心点 c_i 的单位方向向量，那么向量 $\boldsymbol{\alpha}$ 和向量 $\boldsymbol{\beta}$ 之间的夹角是边缘点布局好坏的衡量标准之一. 夹角越小，证明该边缘点所处的位置越接近于其当前社区与它所连接的社区之间的空间位置的中心.

如图 1 所示，图中共有 4 个社区，分别为 c_1，c_2，c_3 和 c_4. 节点 v_1 存在于社区 c_1 中，假设已知节点 v_1 有与节点 v_2，v_3，v_4 的连边，且这 3 个节点分别位于社区 c_2，c_2，c_2 中，图中红色点代表着社区的中心坐标点；那么 $\boldsymbol{\alpha}_1=\boldsymbol{\alpha}_{12}+\boldsymbol{\alpha}_{13}+\boldsymbol{\alpha}_{14}$，如图 1 所示，$\boldsymbol{\beta}$ 即为节点 v_1 当前位置坐标点与社区中心节点之间的方向向量. 此处用 $\boldsymbol{\alpha}_1$ 和 $\boldsymbol{\beta}$ 的夹角来衡量节点 v_1 坐标的合理程度，记作 $r(v_i)=\frac{1-\boldsymbol{\alpha}\cdot\boldsymbol{\beta}}{|\boldsymbol{\alpha}|\cdot|\boldsymbol{\beta}|}$；越小的 $r(x)$ 值代表着更好的布局.

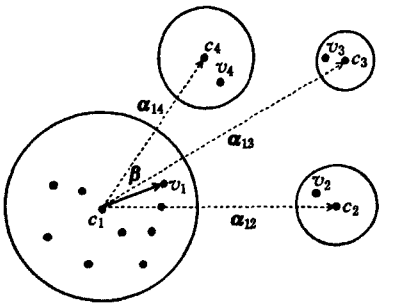


图 1　边缘节点向量计算示意图

2) 度数较大的节点应尽量靠近所属社区中心.

度数较大的节点通常与社区内部的节点有较多的连边，同时由于社区的聚类性，相连的节点彼此之间存在连边的几率较大. 因此，将度数较大的节点布局在靠近中心的位置能够有效地避免边交叉，同时使节点的层次清晰. 这里首先对社区内节点按照度数从小到大进行排列，记节点 v_i 的排序为 $\mathrm{rank}(v_i)$，那么该节点与中心节点的理想距离为

$$d(v_i)=\left(1-\frac{\mathrm{rank}(v_i)}{n}\right)\cdot r \tag{5}$$

其中 r 表示社区半径，n 表示社区内结点个数. 节点与所在社区中心的实际距离越接近于 d，说明布局越好，用函数 $e(x)$ 来衡量，记作 $e(v_i)=\ln(|D_i-d(v_i)|+1)$；其中 D_i 表示节点 v_i 与所属社区中心的实际距离，函数 $e(x)$ 取对数是为了降低实际距离与理想距离偏差对最终布局效果的敏感度. $e(v_i)$ 值越小，节点距所属社区中心节点的距离越趋近于理想距离.

3) 边缘节点应该尽量和与它有连边的非边缘节点靠近.

用边缘节点和与它连接的所有非边缘节点之间的平均最短路径来衡量边缘节点和与它存在连边的非边缘节点的靠近程度，记作 $l(v_i)=\sum_{n=1}^{k}d_{in}/k$；其中与节点 v_i 同社区且存在连边的节点有 k 个，d_{in} 表示节点 i 与节点 n 的距离；则 $l(v_i)$ 表示节点 v_i 距同社区内存在连边的节点的平均距离，$l(v_i)$ 越小，则布局效果越好.

综合考虑以上 3 个约束，定义函数 $U(v_i)$ 来度量每个社区填充点坐标的合理程度，即

$$U(v_i)=h\cdot r(v_i)+g\cdot e(v_i)+m\cdot l(v_i) \tag{6}$$

其中 h,g,m 为常数对应于 3 个约束对最终填充效果影响的权值,它们需要根据实际情况进行调节. 函数 $r(v_i)$,$e(v_i)$和 $l(v_i)$无法同时在给定的定义域范围内(社区坐标域)取到最小值,因此需要利用取条件极值的方式在给定区域坐标范围内不断地调整填充点的坐标,使每一个新插入的 v_i 节点的 $U(v_i)$的值在给定的区域坐标值范围内尽可能的小,从而保证得到满意的填充效果,这是保证最终得到一个布局合理的网络拓扑图的关键.

1.3 算法实现

算法总体流程图如图 2 所示.

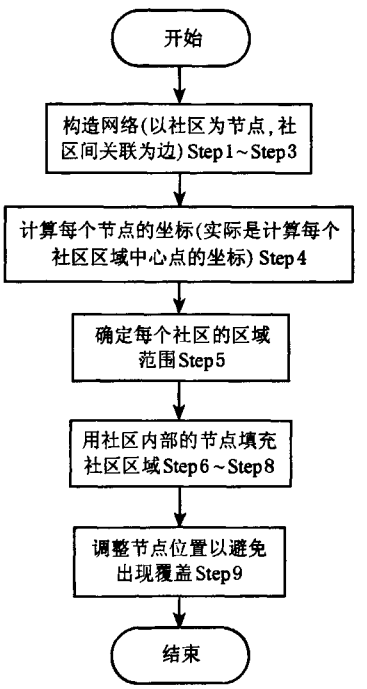


图 2 算法总体流程图

首先,通过 Step1~Step3 构建以社区为节点的网络,Step4 是为了确定社区中心的位置,Step5 是为了确定社区区域半径,至此社区的中心和区域范围都已确定. Step6~Step8 是为了填充社区内部的节点,Step9 用于最终调整节点以避免节点覆盖,最终完成整个网络拓扑图的布局.

算法的具体实现如下:

Step1. 确定边缘节点.

设 E 为网络中所有边的集合,节点 $v_i\in C_m$,$v_j\in C_n$,C_m 和 C_n 分别表示第 m 和第 n 个社区,且 $m\neq n$;e_{ij} 表示 v_i 和 v_j 之间的连边且 $i\neq j$,若 $\exists e_{ij}\in E$,则 v_i 和 v_j 为边缘节点.

Step2. 遍历所有边缘节点,确定社区间的关联. 对于边缘节点 v_i,$v_i\in C_m$,设 E_i 为节点 v_i 所在的所有连边的集合,若有 $e_{ij}\in E_i$ 且边缘节点 $v_j\in C_n$,则称 e_{ij} 为社区 m 到社区 n 的一条连边. 遍历所有节点直至找到所有社区间连边.

Step3. 构建以社区为节点的网络. 以 Step2 中找到的边缘节点所在的社区为节点,社区间的连边为边,连边个数为边的权值构建社区关联网络 $G(V,E)$.

Step4. 运用物理类比方式对 Step3 中构建的网络进行布局:

(a) 初始化节点. 节点坐标随机分配,所受合力 $\boldsymbol{f}=0$;

(b) 遍历每条边,根据公式(1)计算每个节点所受斥力 $\boldsymbol{R}$,$\boldsymbol{f}=\boldsymbol{f}+\boldsymbol{R}$;

(c) 根据公式(2)计算每个节点所受到的来自其他节点的张力 $\boldsymbol{T}$,$\boldsymbol{f}=\boldsymbol{f}+\boldsymbol{T}$;

(d) 使节点沿着 $\boldsymbol{f}$ 的方向运动距离 $d=k|\boldsymbol{f}|$,重置 $\boldsymbol{f}=0$;

(e) 若当前系统未达到平衡,则返回(b)继续执行.

系统是否平衡是由节点移动的位移来衡量的. 若系统所有节点的平均位移小于一个给定的阈值,则认为系统达到平衡. 这里由于所计算的节点其实是一个个社区,而社区的划分数量通常很少,所以节点在空间上的分布不需要过于精确的坐标,因此可以考虑牺牲一定的精度并通过增大阈值来提高算法的收敛速度.

Step5. 确定社区区域半径. 对于社区网络中的每一条边 e,设 l 为 e 的长度. 设这条边对应的 2 个社区 c_i 与 c_j 的半径分别为 r_i 和 r_j,社区内的节点个数分别为 n_i 和 n_j,r_i' 表示当前计算出的临时半径,对式(3)进行变换得出 r'_i,表示为 $r_i'=\dfrac{\sqrt{n_in_j}}{n_i-\sqrt{n_in_j}}$. 若 $r_i'<r_i$,则更新 $r_i=r_i'$. 用同样的方法遍历所有边,最终可确定每个社区区域的半径 r_i.

Step6. 对当前社区内所有节点进行排序,依据式(5)计算每个节点与中心节点的理想距离.

Step7. 对于非边缘节点 $v(x,y)$,

$$\begin{cases}x=x_0+r\cdot\cos\alpha\\ y=y_0+r\cdot\sin\alpha\\ \alpha=\alpha+\beta\end{cases}\tag{7}$$

其中 $v(x_0,y_0)$为社区中心点坐标,$\beta=2\pi/(n+1)$初始 $\alpha=0$,n 为社区内节点总数. 每插入一个新的非边缘节点,需根据式(7)计算其坐标,α 角每次增量为 β.

Step8. 对于边缘节点 $v'(x,y)$,首先根据边缘节点的理想方向向量(依据式(4)计算),理想距离(依据式(5)计算)和社区中心坐标(在 Step4 中已求出)算出点 $p(x,y)$,计算方法参照式(7)的前 2 个等式. 在以点 P 为中心,r 为半径的去心圆范围内,通过计算得到使 $U(v_i)$取最小或极小值时的节点 v_j,取 $v'(x,y)=v_j$.

出于对计算效率的考虑,可以不采用在给定区域内(社区范围)取极小值的方式. 这是因为极小值的坐标有很大几率会落在以点 p 为坐标,r(根据实际社区区域范围确定,这里取 $R-D$,即社区半径-理想距离)为半径的空间范围内.

因此，在这个区域内随机取若干个点的坐标代入式(6)进行计算，使 $U(v_n)$ 最小的点 v_n 也是一个接近于极小值的较优的解.

Step9. 检查节点是否存在覆盖现象. 若存在覆盖，则需要对相关节点的坐标进行调整，调整方法是遍历每个节点，并检查该节点与邻近节点的距离是否小于节点直径. 若小于，则证明存在覆盖，这时将该节点沿着二者圆心连线的方向向外侧移动圆心距离减去直径的距离，遍历所有节点直到不存在覆盖为止.

至此所有节点坐标确定，算法结束.

2　实验结果及讨论

为验证本文算法的有效性，我们在 Windows XP 操作平台上，使用 Eclipse 插件开发技术开发了一个自动绘制网络图的 RCP 插件进行实验测试，选用的测试数据源为复杂网络研究中常用的 Karate[①] 和 Dolphins 网络. 为了直观地展示本文算法的布局效果与特点，使用物理类比布局算法中具有代表性的基于斥力张力模型的网络拓扑图布局算法[8]与本文算法作为对照进行实验，运行环境为 Pentium Ⅳ 3.06 GHz Processor，2 GB RAM.

实验中，本文算法的运行时间包括算法初始处理过程中对网络社区划分的时间消耗以及后续节点位置计算的时间消耗.

Karate 网络描述了 1977 年美国一所跆拳道俱乐部 34 名成员之间的关系网络，其中有 34 个节点，78 条边. 实验结果如表 1 所示.

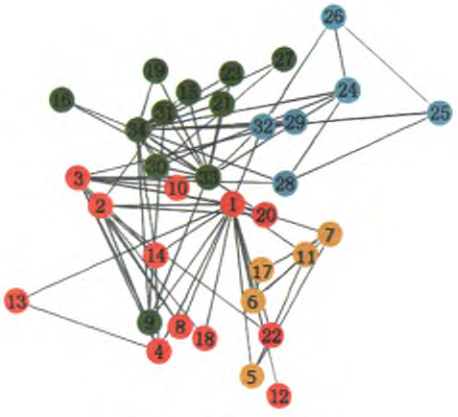
表 1　Karate 网络的实验统计结果

衡量指标	斥力张力布局算法	本文算法
边交叉数	94	48
拥挤度	0.791	0.753
点分布方差	1.25	0.75
运行时间/s	0.963	0.746

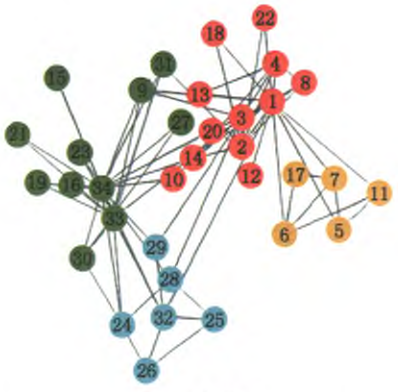
采用基于斥力张力的网络拓扑图布局算法和本文算法作用于 Karate 网络的布局效果如图 3 所示.

对比图 3a 和 3b 可以看出，本文的算法在边交叉数、拥挤度、点分布方差以及运行时间均小于斥力张力布局算法，说明其在效率上相对于传统的物理类比布局算法有所提高；同时从整个布局的边交叉数、拥挤度和点分布方差统计来看，边和点的布局更加清晰均匀.

以 Dolphins 网络为例进行实验，该网络是 Lusseau 等通过对海豚之间接触频率进行 7 年观察



a 基于斥力张力的布局算法



b 本文算法

图 3　Karate 网络布局实验对比图

和研究构建的. Dolphins 社会关系网络是由生活在新西兰 Doubtful Sound 峡湾的 62 只宽吻海豚统计数据构成，网络包含 62 个节点和 159 条边；其中节点代表海豚，连边代表海豚间的接触. 实验结果如表 2 所示.

表 2　Dolphins 网络的实验统计结果

衡量指标	斥力张力布局算法	本文提出的算法
边交叉数	214	143
拥挤度	1.032	0.892
点分布方差	4.25	2.25
运行时间/s	1.341	1.032

运用基于斥力张力的拓扑图布局算法对 Dolphins 网络进行实验，布局效果如图 4 所示.

相对于 Karate 网络而言，Dolphins 网络具有更多的节点和连边，更加容易出现布局的拥挤，这使整体图的结构显得有些混乱，社区效果不够明显，并且运行时间随着节点数的增加增长明显.

① http://www-personal.umich.edu/~mejn/netdata/.

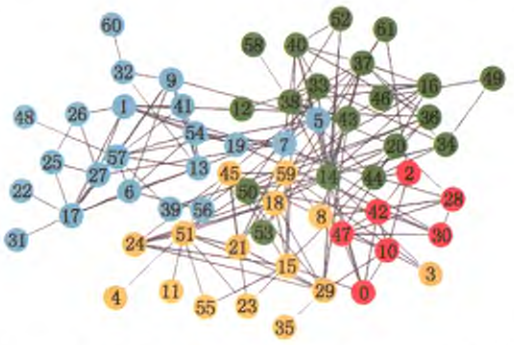


图 4 基于斥力张力的布局算法对 Dolphins 网络的布局效果图

图 5 是运用本文算法对 Dolphins 网络进行布局的效果. 经过统计，相比图 4，本文算法运行时间更短，并在一定程度上降低了整体布局的拥挤程度，节点分布更加均匀.

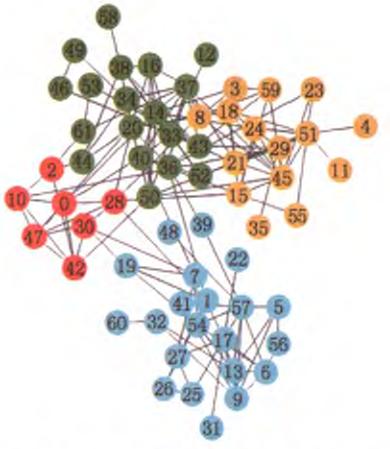


图 5 本文算法对 Dolphins 网络的布局效果图

实验表明，本文算法适用于处理规模在 3 000 个节点以下的网络，尤其适用于对 500 个以下节点的网络进行布局. 例如，采用本文算法曾对辽宁省沈阳市二环以内的交通网络(375 个节点，2 426 条边，数据从 Google Map 采集)进行布局测试，效果良好，运行时间稳定在 12 s 左右. 对于 500 个节点以上的网络，由于在布局效果的数据分析相对复杂，因此本文未做验证. 相比传统物理类比布局算法，本文算法有着更高的执行效率和更清晰的布局效果. 由于社区本身具有社区内部连边众多而社区间连边数明显少于社区内连边的性质，因而基于社区发现的布局算法可以有效地降低布局的拥塞现象并且结构清晰. 同时由于在后续的节点计算中采用了本文提出的收敛速度更快的社区内部节点填充的算法，因而其具有更快运算速度. 现实中，大规模复杂的网络往往具有明显的社区结构，因此本文算法在这方面具有明显的优势及更好的适应性，在处理大规模网络时具有速度快、布局效果好的特点，具有较强的实际应用价值.

3 结 语

针对复杂数据的可视化需求，本文提出了一种基于社区发现的网络拓扑结构可视化布局算法. 由于大多数的真实网络具有复杂网络的相关特性，如聚类性和存在社区结构，因此运用该算法进行布局在反映网络的社区结构同时，在后续计算中大大降低了运算的规模，提高了运算的速度. 仿真实验表明，算法效果较为理想，图布局结构清晰，在处理中小规模的网络数据时能满足布局的需要. 由于真实的数据网络自身具有社区特性或者聚类性，可以通过社区划分进行初始处理以运用本文算法进行布局. 对于具有明显异构性的网络，在布局时可以利用节点类别，种类等不同将同类节点归为一个社区，可以省去了社区划分这个耗时的步骤，从而大幅度地缩减运行时间. 经过实验表明，本文算法能够处理规模在 3 000 个节点以下未划分社区的网络，尤其在处理 500 个节点以下未划分社区的网络，能够在短时间内得出网络布局图，布局效果好，能够满足常规数据的可视化需求.

参考文献(References)：

[1] Carmel L, Harel D, Koren Y. Combining hierarchy and energy for drawing directed graphs [J]. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(1): 46-57

[2] Kamada T, Kawai S. An algorithm for drawing general undirected graphs [J]. Information Processing Letters, 1989, 31(12): 7-15

[3] Davidson R, Harel D. Drawing graphs nicely using simulated annealing [J]. ACM Transactions on Graphics, 1996, 15(4): 301-331

[4] Fruchterman T M J, Reingold E M. Graph drawing by force-directed placement [J]. Software-Practice and Experience, 1991, 21(11): 1129-1164

[5] Ingber L. Very fast simulated re-annealing[J]. Mathematical and Computer Modelling, 1989, 12(8): 967-973

[6] Frick A, Ludwig A, Mehldau H. A fast adaptive layout algorithm for undirected graphs [M] //Lecture Notes in Computer Science. Heidelberg: Springer, 1995, 894: 388-403

[7] Noack A. An energy model for visual graph clustering [M] // Lecture Notes in Computer Science. Heidelberg: Springer, 2004, 2912: 425-436

[8] Cheng Yuan, Yan Wei, Li Xiaoming. Network topology autolayout algorithm based on repulsion-tension model [J]. Computer Engineering, 2004, 30(3): 104-106 (in Chinese)
(程　远，严　伟，李晓明. 基于斥力-张力模型的网络拓扑图布局算法[J]. 计算机工程，2004，30(3)：104-106)

[9] Lu Liang, Lu Zexin, Li Sudan, *et al*. Network topology auto-layout algorithm based on extended mechanic model [J]. Application Research of Computers, 2010, 27(7): 2713-2715 (in Chinese)
(吕　亮，卢泽新，俪苏丹，等. 基于扩展力学模型的网络拓扑图布局算法[J]. 计算机应用研究，2010，27(7)：2713-2715)

[10] Gajer P, Kobourov S G. GRIP: graph drawing with intelligent placement [M] //Lecture Notes in Computer Science. Heidelberg: Springer, 2001, 1984: 104-109

[11] Harel D, Koren Y. A fast multi-scale method for drawing large graphs [M] //Lecture Notes in Computer Science. Heidelberg: Springer, 2001, 1984: 235-287

[12] Hachul S, Junger M. Drawing large graphs with a potential-field-based multilevel algorithm [M] //Lecture Notes in Computer Science. Heidelberg: Springer, 2005, 3383: 285-295

[13] Koren Y, Carmel L, Harel D. ACE: a fast multiscale eigenvectors computation for drawing huge graphs [C] // Proceedings of IEEE Symposium on Information Visualization. Los Alamitos: IEEE Computer Society Press, 2002: 137-142

[14] Harel D, Koren Y. Graph drawing by high-dimensional embedding [M] //Lecture Notes in Computer Science. Heidelberg: Springer, 2002, 2528: 299-345

[15] Dai Hui, Zhou Qiang, Bian Jinian, *et al*. Fast placement algorithm for hierarchical FPGAs [J]. Journal of Computer-Aided Design & Computer Graphics, 2010, 22(9): 1455-1462 (in Chinese)
(戴　晖，周　强，边计年，等. 层次式 FPGA 快速布局算法[J]. 计算机辅助设计与图形学学报，2010，22(9)：1455-1462)

[16] Li Zhuoyuan, Wu Weimin, Hong Xianlong. Incremental placement algorithm for wirelength and routability optimization [J]. Journal of Computer-Aided Design & Computer Graphics, 2003, 15(6): 651-655 (in Chinese)
(李卓远，吴为民，洪先龙. 优化线长和拥挤度的增量式布局算法[J]. 计算机辅助设计与图形学学报，2003，15(6)：651-655)

[17] Cheng Xueqi, Shen Huawei. Community structure of complex networks [J]. Complex Systems and Complexity Science, 2011, 8(1): 58-70 (in Chinese)
(程学旗，沈华伟. 复杂网络中的社区结构[J]. 复杂系统与复杂性科学，2011，8(1)：58-70)

[18] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69(2): Article No. 26113

[19] Hu Y Q, Li M H, Zhang P. Community detection by signaling on complex networks [J]. Physical Review E, 2008, 78(1): Article No. 016115

[20] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms [J]. Physical Review E, 2008, 78(4): Article No. 046110

[21] Leung L X Y, Pan H, Lio P, *et al*. Towards real-time community detection in large networks [J]. Rhysical Review E, 2009, 79(6): Article No. 066107

[22] Dorso C O, Medus A D. Community detection in networks [J]. International Journal of Bifurcation and Chaos, 2010, 20(2): 361-367

[23] Archambault D, Munzner T, Auber D. TopoLayout: multilevel graph layout by topological features [J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(2): 305-317

[24] von Landesberger T , Kuijper A, Schreck T, *et al*. Visual analysis of large graphs [J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 29(2): 37-60