Complex Network Clustering by Multiobjective Discrete Particle Swarm Optimization Based on Decomposition

报告人：张天乐
导　师：公茂果

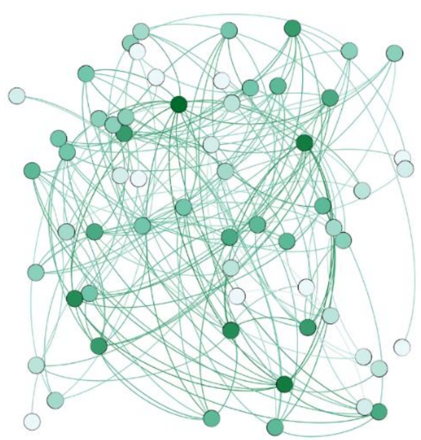
西安电子科技大学
XIDIAN UNIVERSITY

主要内容

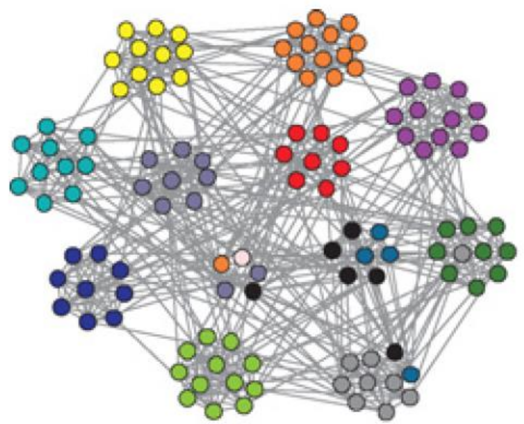
研 究 背 景

➢复杂网络

- 小世界
- 集群即集聚程度(clustering coefficient)
- 幂律（power law）



海豚社交网络



足球网络

研究背景

网络定义

1.非符号网络（unsigned network） G=(V,E)

强感知 $\forall i \in S,\ k_i^{in} > k_i^{out}$

弱感知 $\sum_{i \in S} k_i^{in} > \sum_{i \in S} k_i^{out}$

$$k_i^{in} = \sum_{i,j \in S} A_{ij}$$

$$k_i^{out} = \sum_{i,j \notin S} A_{ij}$$

2.符号网络（signed network） G=(V,PE,NE)

强感知 社区内，积极连接大于消极连接

$\forall i \in S, (k_i^+)^{in} > (k_i^-)^{in}$

弱感知 社区内，积极连接密集；社区间消极连接密集

$$\begin{cases} \sum_{i \in S} (k_i^+)^{in} > \sum_{i \in S} (k_i^+)^{out} \\ \sum_{i \in S} (k_i^-)^{out} > \sum_{i \in S} (k_i^-)^{in} \end{cases}$$

问题描述

➤网络聚类

Girvan & Newman 提出模块度函数（Q函数）

$$Q=\frac{1}{2m}\sum_{i,j}(A_{ij}-k_i\,k_j/2\,m)\delta(i,j)$$

G´omez 提出 （SQ函数）

$$SQ=\frac{1}{2\omega^{+}+2\omega^{-}}\sum_{i,j}\left(\omega_{ij}-(\frac{\omega_i^{+}\omega_j^{+}}{2\omega^{+}}-\frac{\omega_i^{-}\omega_j^{-}}{2\omega^{-}})\right)\delta(i,j)$$

存在局限性

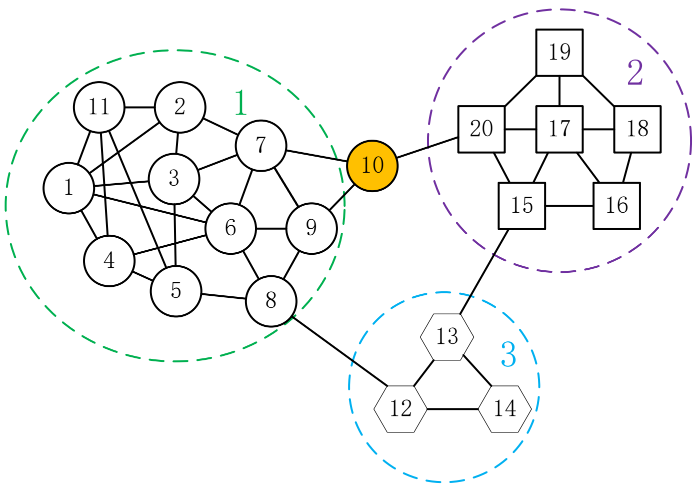
问题描述

- 模块密度(modularity density)

$$S = (V_1, V_2, \cdots, V_k)$$

maximize minimize

$$D = \sum_{i=1}^{k} \frac{L(V_i, V_i) - L(V_i, \bar{V_i})}{|V_i|}$$



$$L(V_1, \overline{V_2}) = \sum_{i \in V_1, j \in \overline{V_2}} A_{ij} \qquad L(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij}$$

M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Phys. A Stat. Mech. its Appl.*, vol. 391, no. 15, pp. 4050–4060, 2012.

问题描述

➢ 目标函数（非符号网络）

negative ratio association (NRA) → kernel k-means (KKM)

ratio cut (RC)

$$min = \begin{cases} NRA = -\sum_{i=1}^{k} \frac{L(V_i, V_i)}{|V_i|} \\ RC = \sum_{i=1}^{k} \frac{L(V_i, \overline{V_i})}{|V_i|} \end{cases}$$

→

$$KKM = 2(n-k) - \sum_{i=1}^{k} \frac{L(V_i, V_i)}{|V_i|}$$

问题描述

➢目标函数（符号网络）

$$\min = \begin{cases} SRA = -\sum_{i=1}^{k} \frac{L^+(V_i, V_i) - L^-(V_i, V_i)}{|V_i|} \\ RC = \sum_{i=1}^{k} \frac{L^+(V_i, \bar{V_i}) - L^-(V_i, \bar{V_i})}{|V_i|} \end{cases}$$

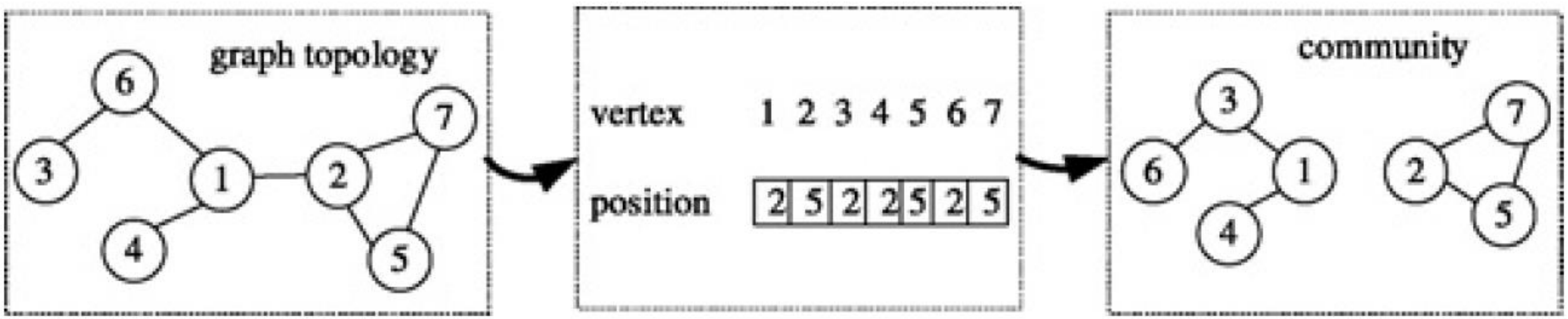
$$L^+(V_i, V_j) = \sum_{i \in V_i, j \in V j} A_{ij} \qquad A_{ij} > 0$$

$$L^-(V_i, V_j) = \sum_{i \in V_i, j \in V j} |A_{ij}| \qquad A_{ij} < 0$$

算法框架

粒子状态定义

位置定义



速度定义

$$V_i = \{v_1, v_2, \cdots, v_n\}$$

如果 $v_i = 1$ 粒子飞行

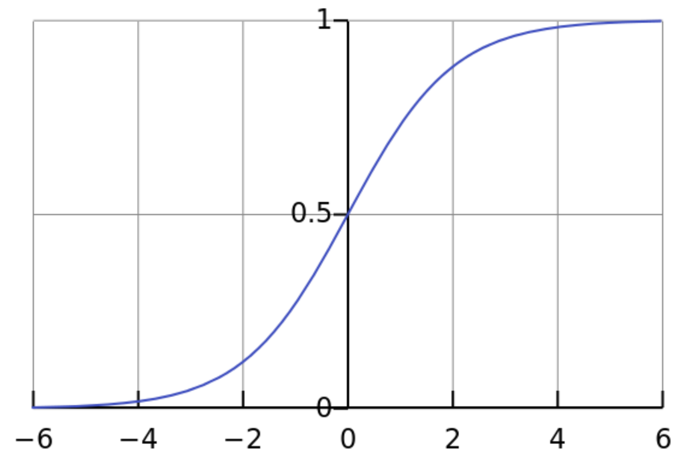
如果 $v_i = 0$ 粒子不动

算法框架

➢粒子状态更新

$$V_i = \textcolor{red}{sig}(\omega V_i + c_1 r_1 (X_i \oplus Pbest_i) + c_2 r_2 (X_i \oplus Gbest))$$

$$\begin{cases} y_i = 1 \ \ if \ \ rand(0,1) < sigmoid(x_i) \\ y_i = 0 \ \ if \ \ rand(0,1) \geq sigmoid(x_i) \end{cases}$$



$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

例：

Xi={1,**2**,**2**,**4**,2,2,**7**}

Pbest={1,**1**,**1**,**1**,2,2,**2**}

V1={0,**1**,**1**,**1**,0,0,**1**} r1c1=1.4

V2={1,0,0,1,0,1,0} r2c2=0.7

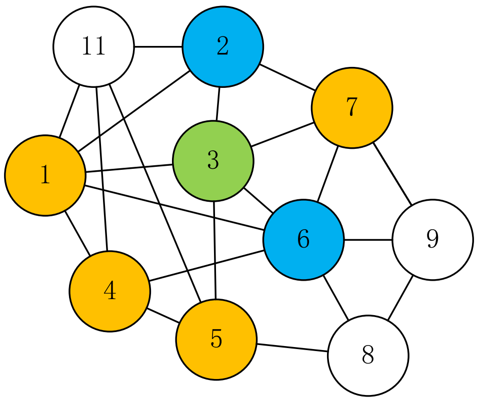
sig(X)={0,1,1,1,0,0,1}

粒子状态更新

$$x_i^t = x_i^t \otimes v_i^t$$

定义⊗操作，若**v=1**，**x**更新为其邻接节点所在最多的社区

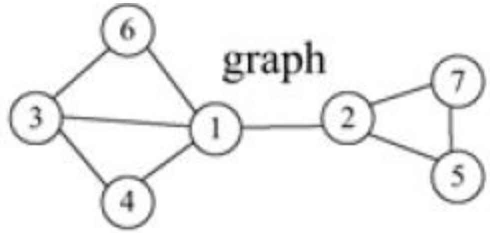
$$\begin{cases} x_{2i} = x_{1i} & if\ v_i = 0 \\ x_{2i} = Nbest_i & if\ v_i = 1 \end{cases}$$

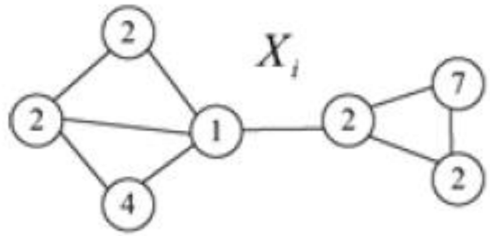


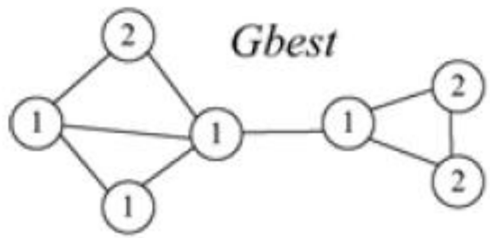
$$Nbest_i = \arg\max_r \sum_{j \in N} \varphi(x_{1j}, r)$$

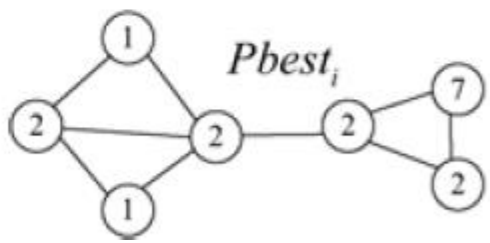
算法框架

➢粒子状态更新









$$X_i = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [1\ 2\ 2\ 4\ 2\ 2\ 7]$$

$$Gbest = [g_1, g_2, g_3, g_4, g_5, g_6, g_7] = [1\ 1\ 1\ 1\ 2\ 2\ 2]$$

$$Pbest_i = [p_1, p_2, p_3, p_4, p_5, p_6, p_7] = [2\ 2\ 2\ 1\ 2\ 1\ 7]$$

$$Gbest \oplus X_i = [0\ 1\ 1\ 1\ 0\ 0\ 1] = V_1$$

$$Pbest_i \oplus X_i = [1\ 0\ 0\ 1\ 0\ 1\ 0] = V_2$$

$$R_1 = r_1 c_1 = 1.4,\quad R_2 = r_2 c_2 = 0.7$$

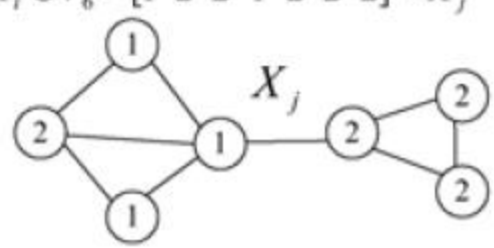
$$R_1 V_1 = [0\ 1.4\ 1.4\ 1.4\ 0\ 0\ 1.4] = V_3$$

$$R_2 V_2 = [0.7\ 0\ 0\ 0.7\ 0\ 0.7\ 0] = V_4$$

$$V_3 + V_4 = [0.7\ 1.4\ 1.4\ 2.1\ 0\ 0.7\ 1.4] = V_5$$

$$sig(V_5) = [0\ 1\ 1\ 1\ 0\ 0\ 1] = V_6$$

$$X_i \otimes V_6 = [1\ 2\ 2\ 1\ 2\ 2\ 2] = X_j$$



算法框架

- 粒子初始化

  PGLP算法

$$l(i) = \text{argmax}_r \sum_{j \in \Omega(i)} \delta(l(j), r)$$

Algorithm 2 Pseudo Code of PGLP

```
Begin
Input: label propagation iterations: iters.
    for each chrom(i)∈ population
      for(j = 1 : iters)
        for(k = 1 : vertexes)
          if(node[k].neighbor.size > 1)
            for(m = 1 : node[k].neighbor.size)
              Node[k].label ←formula (3);
          else Node[k].label ←node[k].neighbor.label;
        End for
      End for
    End for
Output: initialed chromosomes.
End
```

- Gbest

  从ns个corresponding neighbors随机选择

算法框架

➢ 契比雪夫方法 Tchebycheff approach

MOEA/D

- Weighted Sum Approach
- Tchebycheff Approach
- Boundary Intersection (BI) Approach

$$g^{te}(x|\omega, z^*) = \max_{1\le i\le k} \omega_i |f_i(x) - z_i^*|$$

$$x \in \Omega$$

$$z^* = (z_1^*, z_2^*, z_3^*, \cdots, z_k^*)$$

$$z_i^* = \{\min f_i(x) | x \in \Omega\}$$

算法框架

➢ Pbest

```
If    新生成的解X    支配    pbest
          Pbest = X;
else if    pbest   支配    X
          pbest保持其原始状态;
          else   //互相不支配
               // 那么我们使用聚合方法来确定是否更新pbet
             If
```

$$w_{i1}f(x_i^{t+1}) + w_{i2}f(x_i^{t+1}) < w_{i1}f(pbest_i) + w_{i2}f(pbest_i)$$

```
                  pbest = x;
             end if
       end if
end if
```

算法框架

➢ 震荡操作 turbulence operation

- ◆ 维持多样性
- ◆ 逃离局部最优

基于邻域的突变（NBM）

对每个粒子中的节点，如果随机数小于突变概率pm，则将NBM过程应用于该节点，即，将其标签标识符分配给其所有邻居。

Algorithm 3 Pseudo code of turbulence operation on one particle.

```
for i = 0; i < vertex; i++
    if rand(0, 1) < pm
        for j = 0; j < node[i].neighborsize; j++
            x[node[i].neighbor[j]] = x[i];
        end for
    end for
end for
```

算法框架

➤MODPOS算法整体框架

输入：网络的邻接矩阵A

输出：网络划分结果

Step1. 初始化

step 1.1 初始化种群 $P = \{x_1, x_3, \ldots, x_{pop}\}^T$

step 1.2 初始化速度 $V = \{v_1, v_2, \ldots, v_{pop}\}^T$

step 1.3 初始化分布均匀的权重向量

step 1.4 初始化pest，$pbest_i = x_i$

step 1.5 初始化参考点 z*

step 1.6 初始化neighborhood N 基于欧拉距离

$N = \{n1, n2, \ldots, npop\}^T$

Step 2.设置迭代次数标记 t = 0

Step 3.循环开始 for i = [1, pop] do

step 3.1 选择gbest

step 3.2 计算速度

step 3.3 计算位置

step 3.4 if (t < maxgen * pm) 突变

step 3.5 计算新解 $x(i)^{t+1}$

step 3.6 更新neighborhood

step 3.7 更新参考点z*

step 3.8 更新pest

step 4 if (t < maxgen) { t++; go to step3;}

else output

实验

规范化互信息 NMI

通过比较网络真实社区结构与MODPSO发现的网络社区结构的结构相似性来验证算法的有效性。

If NMI = 1 ， 两种社区结构完全一致
If NMI = 0 ， 两种社区结构完全不同

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.}\log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j}\log(C_{.j}/N)}$$

实验

➢实验结果

网络	节点数	边数	真实社区个数
dolphin	62	159	2
football	115	613	12

网络	指标	值
dolphin	NMImax	1
	NMIavg	1
	Qmax	0.4198
	Qavg	0.4198
football	NMImax	0.9289
	NMIavg	0.9278
	Qmax	0.6046
	Qavg	0.6035

总结与展望

总结

- 对于离散问题重新定义粒子的速度和位置
- 基于分解的多目标网络聚类方法
- 将网络拓展至符号网络

展望

- 引入分布式并行计算

Thank you !