Journal of Computer Applications
计算机应用,2012,32(2):317-321
ISSN 1001-9081
CODEN JYIIDU
2012-02-01
http://www.joca.cn

文章编号:1001-9081(2012)02-0317-05
doi:10.3724/SP.J.1087.2012.00317

# 基于基因表达式编程算法的复杂网络社区结构划分

罗锦坤*,元昌安,杨　文,胡卉颖,袁　晖
(广西师范学院 计算机与信息工程学院,南宁 530023)
(*通信作者电子邮箱 ljk_1985@126.com)

**摘　要**:由于复杂网络的不确定性,传统的复杂网络社区结构划分算法易造成过早收敛,使精度降低,且由于计算量大,时间复杂度较高。为克服以上不足,利用基因表达式编程(GEP)的自适应性和全局搜索能力强以及具有并行性计算等特点,优化网络社区结构的划分,提出了一种基于GEP的复杂网络社区结构划分算法,并通过实验验证了新算法的有效性。新算法在无先验信息情况下,可较准确地完成对复杂网络的社区划分。

**关键词**:复杂网络;社区划分;基因表达式编程
**中图分类号**:TP311.13　　**文献标志码**:A

# Community structure division in complex networks based on gene expression programming algorithm

LUO Jin-kun*, YUAN Chang-an, YANG Wen, HU Hui-ying, YUAN Hui
(*College of Computer and Information Engineering, Guangxi Teachers Education University, Nanning Guangxi 530023, China*)

**Abstract:** Due to the uncertainty of complex networks, traditional community structures division algorithm of the complex network could easily lead to premature convergence and decreased accuracy. And because of the large amount of computation, time complexity is high. To overcome the above shortcomings, the paper adopted GEP's global search ability and adaptability, and other characteristics with parallel calculations, optimized the network structure of the division of community, and proposed a community structure division algorithm of complex network based on GEP, and verified the validity of the new algorithm by experiment. The new algorithm has more accurate community division of the complex network in the case of no prior information.

**Key words:** complex network; community division; Gene Expression Programming (GEP)



## 0　引言

所谓复杂网络就是现实生活中复杂系统的抽象表示形式之一,由这些表示复杂系统独立个体的点和个体之间关系的边组成的网络。研究表明,很多复杂网络是异构的,复杂网络中的节点并不是全都具有统一的特征,而是许多不同类型节点的组合。相同类型的节点之间存在较多的连接,而不同类型的节点之间的连接则相对较少[1]。网络中的社区就是满足同一类型中的节点以及这些节点之间的边所构成的子图[2]。

社区结构作为复杂网络的一个重要特性,对于深入了解网络结构与分析网络特性具有重要意义。目前复杂网络的社区划分问题得到了大量的研究,传统的复杂网络社区结构发现算法有著名的Kernighan-Lin算法[3]、基于Laplace图特征值的谱平分法[4-5]、Girvan-Newman(GN)算法[6]等。Kernighan算法是一种基于贪婪算法原理,将网络分割为两个大小已知的社区的二分法,但该算法必须已知网络社区的确切规模才能得到正确结果,使其在实际网络分析中难以得到较好的应用。基于Laplace图特征值的谱平分法是利用网络结构的Laplace矩阵中不为零的特征值所对应的特征向量和同一个社区内的节点对应的元素近似相等的原理对网络社区进行划分,在具有$n$个节点的网络中,该算法的复杂度等价于求$n \times n$矩阵的特征向量的复杂度$O(n^3)$。GN算法则是一种分裂方法,它通过不断地从网络中移除边介数最大的边,将整个网络分解为多个子社区,边介数就是网络中经过每条边的最短路径的数目,它为区分社区的内部边和连接社区之间的边提供了一种有效的度量方法。它的时间复杂度为$O(e^3)$($e$表示网络中的边数)。虽然GN算法弥补了传统算法的一些不足,但是在不知道社区数目的情况下,无法知道这种分解到底要进行到哪一步终止。为克服以上不足,文献[1]又提出了一种基于DNA遗传算法的复杂网络社区结构发现算法,该算法尝试用DNA遗传算法来优化网络社区结构的划分,但该方法在现有的串行计算机上还无法体现DNA计算的并行性。而且由于遗传算法本身的特性,其个体由固定长度的线性串表示,由于线性串的长度在进化过程中固定不变,因而损失了功能复杂性;另外遗传算法的个体在遗传操作过程中,绝大部分都会死亡,这为检查个体存活性浪费了大量资源[7]。由此可见,当前大多数社区划分算法社区划分精度不高,且时间复杂度较高。针对以上情况,本文提出了一种基于基因表达式编程(Gene Expression Programming, GEP)算法的复杂网络社区发现(Community Structure Detection in complex networks based on GEP, GEP-CSD)方法,并且利用该社区划分算法对实际复杂网络数据节点进行划分和预测,既实现功能的复杂性,也让染色体在遗传操作下全部存活。同时该方法无需知

**收稿日期**:2011-08-01;**修回日期**:2011-09-02。　**基金项目**:国家自然科学基金资助项目(60763012);广西自然科学基金资助项目(2011GXNSFD018025);广西研究生教育创新计划资助项目(2011106030703M05)。

**作者简介**:罗锦坤(1985-),男,福建漳州人,硕士研究生,主要研究方向:智能计算、数据挖掘;　元昌安(1964-),男,安徽肥东人,教授,博士,CCF会员,主要研究方向:智能计算、数据挖掘;　杨文(1987-),男,湖南长沙人,硕士研究生,主要研究方向:并行计算;　胡卉颖(1986-),女,江西九江人,硕士研究生,主要研究方向:智能计算、数据挖掘;　袁晖(1987-),男,江苏南通人,硕士研究生,主要研究方向:智能计算、数据挖掘。

道社区先验信息,如社区大小,就可以快速有效地完成对复杂网络的社区划分。

本文算法主要特点如下:1)定义衡量网络社区划分优劣的度量函数作为染色体的适应值函数;2)改进传统算法,解决社区划分这种离散优化问题(传统算法不能很好解决社区划分过程中出现的离散孤立点问题);3)引入两种修复策略,修复在搜索过程中产生的社区孤立点,确保搜索到合理的解。验证实验结果表明:本文算法提高了网络社区划分精度。

## 1 相关工作

基因表达式编程[8-9]是借鉴生物遗传的基因表达规律提出的知识发现新技术。它是在遗传算法(Genetic Algorithm, GA)和遗传编程(Genetic Programming, GP)的基础上发展的新概念,首批研究成果于 2000 年 12 月在网上发表,在 2001 年 12 月正式发表[10]。

GEP 作为 GA 和 GP 的继承和发展,综合了二者的优点,具有更强的解决问题的能力[11]。它与 GA 和 GP 相比在主要步骤上都极为相似,但 GEP 克服了 GA 与 GP 的不足,更适合于函数关系的挖掘。它们最本质的区别在于:在 GA 中个体由固定长度的线性串(染色体)表示,在 GP 中个体由不同大小和形状的非线性实体(解析树)表示;而 GEP 将个体先编码为固定长度的线性串再表示成大小、形状都不同的非线性实体。这样,GEP 就克服了 GA 损失功能复杂性的可能和 GP 难以再产生新的变化的可能。

文献[12]给出了基本 GEP 算法中个体的组成,个体在 GEP 中又称为染色体(chromosome),染色体是由基因(gene)通过连接运算符(link operator)连接组成的,其最大的特点就是个体的基因型表示形式。GEP 的基因型个体由定长的头部和尾部组成,头部元素 ∈ {函数集 $F$} ∪ {终结符集 $T$},尾部元素 ∈ {终结符集 $T$}(其中函数集 $F$ 由求解问题需要的所有函数运算符组成,终止符集 $T$ 由描述问题的解的已知符号、变量或常数组成),并且头部长度 $h$ 和尾部长度 $t$ 须满足以下关系:

$$t = h(n_{max} - 1) + 1 \tag{1}$$

其中 $n_{max}$ 为函数集 $F$ 中函数的最大操目数,在个体表现型中表现为树型结构中节点的最大分支数,这一设计使得基于基因型个体的遗传操作都具有很好的合法性。GEP 的个体表现型被称为 ET 树。ET 树是通过顺序扫描基因型个体的字符,按照层次顺序构成。例如,若定义函数集 $F = \{*, /, +, -, Q\}$(依次为乘、除、加、减和开方运算),终止符集 $T = \{c, d, e, f\}$,则 $n_{max} = 2$,取头部长 $h = 5$,由式(1)得到尾部长 $t = 6$。假定有基因型个体:$Q/+-cfed$,它可以转化为如图 1 所示的 ET 树。

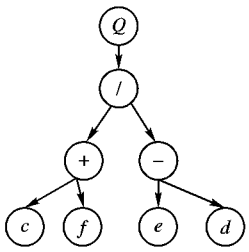



图 1 代数表达式 $\sqrt{(c+f)/(e-d)}$ 对应的表达式树

文献[10]给出了基本 GEP 算法的操作步骤:1)输入相关参数,创建初始化群体;2)计算每个个体的适应度函数,若不符合结束条件,继续下一步,否则结束;3)保留最好个体;4)选择操作;5)变异;6)插串操作(IS 插串、RIS 插串、Gene 插串);7)重组(1-点重组、2-点重组、多基因重组);8)若达到设定最大进化代数或计算精度,则进化结束,否则转到步骤 2)。

## 2 基于 GEP 算法的复杂网络社区结构划分

GEP 算法作为一种基于群智能的高效优化算法,对解决多维多目标复杂优化问题已经显示出一定的优势。GEP 具有极强的函数发现能力[13],它能揭示数据之间内在的本质,利用 GEP 对复杂网络的社区划分的思想就是用 GEP 来挖掘样本数据中蕴涵的函数关系,利用此函数关系式结合适当的阈值来达到划分复杂网络社区的目的。

本文提出的基于 GEP 算法的复杂网络社区结构划分需解决以下问题:一是定义评价复杂网络社区划分优劣的度量函数作为染色体的适应度函数;二是给出标准 GEP 选择策略;三是修复在社区划分过程中产生的孤立点以确保搜索到最佳解。

### 2.1 社区划分的 GEP 模型

#### 2.1.1 社区种群始初化及运算符定义

GEP 具有强大的自适应全局搜索能力,在网络社区划分应用方面具有强大的潜力。本节将介绍基于 GEP 算法的复杂网络社区发现(GEP-CSD)方法。每条染色体所对应的函数表达式将对应一种该网络结构下的社区划分方案。该方法首先采用社区分裂策略,将网络社区分为两个子社区,然后移除相连两个社区的边,最后对各个子社区进行划分,直到不能再划分为止。定义种群大小为 $N$,基因头长度为 $h$,基因尾长度为 $e$,函数最大操作目数为 $k$,最大迭代数为 $I_{max}$,终止迭代的适应度值为 $f_{max}$,变异率为 $Pmu$,插串率为 $Ptr$,重组率为 $Pre$。初始化种群,给每个节点随机分配一个社区号 $ID$。如表 1 所示,根据社区内节点的总数量为 $n$,建立一个初始化数组,其中染色体的数量为 $m$,根据求解的复杂情况而定。其中 $C\_nodenum(n)$ 为社区节点,$Chrom$ 为染色体。

表 1 种群初始化

| $C\_nodenum(1)$ | $C\_nodenum(2)$ | … | $C\_nodenum(n-1)$ | $C\_nodenum(n)$ | $Chrom$ |
|---|---|---|---|---|---|
| 18 | 33 | … | 27 | 10 | $Chrom\_1$ |
| 29 | 1 | … | 88 | 2 | $Chrom\_2$ |
| 25 | 26 | … | 66 | 3 | $Chrom\_3$ |
| ⋮ | ⋮ | | ⋮ | ⋮ | ⋮ |
| 11 | 79 | 36 | 56 | 4 | $Chrom\_m$ |

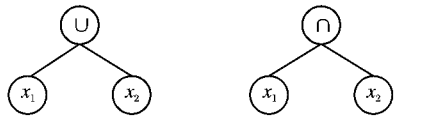
**定义 1** 社区中基因、群体、染色体。设 $F = \{f | f$ 为社区节点运算符 ∪ 和 ∩$\}$,$|F|$ 为 $F$ 中社区运算符个数,$T = \{x | x$ 为社区节点号即终结字符$\}$,$|T|$ 为 $T$ 中社区节点的个数,设社区 GEP 的基因表达式为 $f_1 f_2 \cdots f_h x_1 x_2 \cdots x_e$,其中 $f_i \in F \cup T$,$x_j \in T$,$h$ 为社区中基因头部的长度,$e$ 为社区中基因尾部的长度,且 $e = h(n-1) + 1$,$n$ 为社区节点运算符的最大操作数,则社区 GEP 群体表示为 $G = \{g | g = f_{11} f_{12} \cdots f_{1h} x_{11} x_{12} \cdots x_{1e} \cdots f_{k1} f_{k2} \cdots f_{kh} x_{k1} x_{k2} \cdots x_{ke}, f_{ij} \in F \cup T, x_{iq} \in T, i = 1,2,\cdots,k; j = 1,2,\cdots,h; q = 1,2,\cdots,e\}$,元素 $g$ 为社区染色体或社区群体的个体,$f_{ij}$ 为社区染色体的头分量,$x_{iq}$ 为

社区个体的尾分量。

**定义 2**　社区表达树 ET。在 GEP 结构中引入具有特殊功能的社区运算符 ∪ 和 ∩,使得构成的 ET 具有对网络节点进行自动社区划分的功能。引入了社区运算符的特殊 ET 称之为社区 ET。下面给出了符号定义:

∪表示其左右子树中的节点分属不同的社区,实现社区的分割,如图 2(a)所示;

∩表示其左右子树中的节点属于同一社区,实现社区的合并,如图 2(b)所示。




(a) $x_1$与$x_2$属于不同的社区　(b) $x_1$与$x_2$属于同一社区

图 2　社区运算符

### 2.1.2　社区染色体编码

由于社区划分的特殊性,染色体采用整数编码,并由单基因构成,单基因由基因头和基因尾构成,其中基因头部包含复杂网络中社区运算符“∪”和“∩”,基因尾部则从复杂网络中随机抽取无重复的节点序号构成。例如:用 $R1$ 表示染色体且 $R1 = \cup\cup\cap\cap\cup x_1x_2x_3x_4x_5x_6$,则 $R1$ 是一个合法的染色体。它描述的是进化过程中某一代的社区划分方案,其中,$x_i(i \in [1,n]$,$n$ 为网络中节点的总个数)是网络中每个节点对应的序号。

### 2.1.3　社区 ET 编码

在本算法中,染色体个体首先采用自顶向下,再从左到右的顺序编码为社区 ET,以便进行适应度计算。

### 2.1.4　个体适应度函数定义

对于网络社区结构划分问题,每条染色体代表每一种社区划分方案。适应值函数采用 Newman 等[11]提出的一种度量网络社区划分质量的标准。若将网络划分为 $K$ 个社区,定义一个 $K \times K$ 的对称矩阵 $\boldsymbol{E} = (e_{ij})$,其中元素 $e_{ij}$ 表示网络中连接 $i$ 社区和 $j$ 社区的边的数量占所有边的比例,这里所说的边是指在原始网络中,利用完整的原始网络计算的,并不是指在计算过程中被算法破坏的网络)。假设矩阵中对角线上各元素之和为 $\mathrm{Trace}(\boldsymbol{E}) = \sum_i e_{ii}$,$e_{ii}$ 表示网络中连接某一个社区内部各节点的边在所有边的数目中所占比例。定义每行(或每列)中各元素之和为 $a_i = \sum_j e_{ij}$,它表示与第 $i$ 个社区中的节点相连的边在所有边中所占的比例。在此基础上,用式(2)来定义网络社区划分的衡量标准。

**定义 3**　社区染色体个体适应度。设将网络划分为 $K$ 个社区,矩阵 $\boldsymbol{E} = (e_{ij})$ 为 $K \times K$ 阶的对称矩阵,其中 $e_{ij}$ 表示网络中连接 $i$ 社区和 $j$ 社区的边的数量占原始网络所有边的比例,矩阵 $\boldsymbol{E}$ 中对角线上各元素之和为 $\mathrm{Trace}(\boldsymbol{E})$,每行或每列各元素之和为 $a_i$,则社区染色体个体适应度函数 $f_i = Q$,其中 $Q$ 为式(2)所示。

$$Q = \sum_i (e_{ii} - a_i^2) = \mathrm{Trace}(\boldsymbol{E}) - \| \boldsymbol{E}^2 \| \qquad (2)$$

其中 $\| \boldsymbol{E}^2 \|$ 表示矩阵 $\boldsymbol{E}^2$ 中所有元素之和。式(2)表明网络中连接两个同类型节点的边的比例,减去在同样的社区结构下任意连接这两个节点的边的比例的期望值。$Q$ 越接近 1,表示社区结构越明显。通过这个度量标准就可以建立起社区划分质量的全局度量函数,即适应度函数 $f_i = Q$($i$ 表示第 $i$ 个染色体个体)。

### 2.1.5　社区遗传算子

1)选择操作。在进化每一代采用染色体的适应度来排序,染色体个体适应度的高低由网络模块度 $Q$ 来决定。采用被广泛使用的轮盘赌选择策略,同时使用“精英保留”策略来保持进化过程中出现的优良特性。

2)交叉算子。采用单点和双点交叉方式。

3)变异算子。根据文献[14],变异概率取 0.044。先随机选定要变异的基因位和基因码。基因头部的基因码根据定义 4 进行变异。基因尾部的基因码的变异采用先随机从网络社区节点中选取一个节点,并判断用此节点代替要变异的基因码后是否会产生重复基因码,若不产生重复则用新的基因码替代旧的基因码;否则继续寻找下一个不会产生重复基因码的节点来替代原基因码。

**定义 4**　基因头部的基因码的变异。若有 ∪→∩,∩→∪,则表示把基因头包含的运算符 ∪ 变异成 ∩,将 ∩ 变异成 ∪。

### 2.1.6　社区 ET 解码

染色体个体在经过多种遗传操作,完成每代进化之后,要计算评价其适应度,首先就需要对染色体所对应的社区 ET 进行解码,从而获取染色体所携带的网络社区节点序列信息。描述社区 ET 解码过程见算法 1。

算法 1　解码社区 ET(community decoding ET):分析出解码后的网路社区节点序列。

输入:输入染色体个体经过编码后得到的社区 ET;

输出:输出社区 ET 解码后对应的染色体所携带的社区节点序列。

```
1)  若 ET 为空则执行 11);
2)  Switch(ET 根节点):        //其中社区表达树 ET 按照定义 2
3)    Case '∪':
4.      剪枝左子树社区节点,将左子树社区节点放入集合 i 中
5)      剪枝右子树社区节点,将右子树社区节点放入集合 i 中
6)      FOR each SET i DO
7)      递归解码集合 i 中的每个子树
8)    Case '∩':
9)      递归解码社区节点左子树
10)     递归解码社区节点右子树
11) 返回染色体中携带的社区节点序列
```

### 2.1.7　基于 GEP 的网络社区结构划分算法

算法 2　基于 GEP 的网络社区结构划分算法(GEP-CSD)。

输入:训练集样例,最大操目数,交叉率,插串率,重组率,变异率;

输出:最后一代群体中最佳染色体所对应的最佳社区划分方案。

```
begin{
1)  gen = 0;                                  //表示循环的代数
2)  initialize();
        //按照表 1 初始化种群,其中染色体、基因按照定义 1
3)  statistics(fistpop);                      //统计第一代的情况
4)  report(fistpop);                          //报告第一代的情况
5)  while ( newpop [ maxpp ]. fitness < = Maxfitness or gen <
      Maxgen) {
     //个体的最大适应度值小于最大适应度值或者运行代数
     //小于最大代数时,则继续循环,其中适应度按照定义 3
6)    select();              //选择操作,采用轮盘赌选择法策略
7)    mutation();            //变异操作,变异算子按照定义 4
```

```
8)    inversion();                          //反串操作
9)    transposition();                      //插串操作;
10)   recombination();                      //重组操作
11)   statistic (newpop);                  //统计下一代的情况
12)   report(newpop);}                     //报告下一代的情况
end
```

### 2.2 社区划分孤立点修复策略

文献[15]提出两种修复策略,对于网络社区划分问题,作为一种随机自动基因表达式编程优化算法,其在搜索到的划分方案中可能存在孤立点的现象。所谓孤立点现象是指网络图中其邻接的图节点全部或绝大多数是与其相异组的成员的节点。可分为绝对孤立点和相对孤立点,孤立点的存在会影响算法搜索的效率,制定不同强度的修复策略可以提高算法的收敛速度;也可以使算法避免陷入局部最优。因此,本文引入以下两种方法作为社区划分的修复策略。

1)柔性修复策略[15]。如果网络图中存在其邻接的绝大多数图节点为与其相异的分组节点,则称其为相对孤立点。如在算法搜索过程中,搜索的划分方案中存在相对孤立点,则找出与该点邻接且同组的节点数目 $a$,与其相邻接的异组节点数目为 $b$,则定义 $L = b/a$。如果 $L$ 大于给定阈值 $k$,则强制其改变分组,否则保持分组不变。

2)绝对孤立点修复策略[15]。如果网络图中存在与其邻接的所有节点的分组相异的节点,称其为绝对孤立点。如在算法搜索过程中,搜索的网络社区划分方案中存在绝对孤立点,则强制其改变分组,转为与其邻接点相同的分组。

### 2.3 算法时间复杂度分析

命题 1 算法时间复杂度分析。设种群大小为 $P$,总进化代数为 $m$,$N_{max}$ 为种群最高适应度持续无法提高的进化代数的阈值,$L$ 为样本集的长度,把网络划分为 $K$ 个社区,网络节点数为 $n$,网络中边的数目为 $e$。则在最坏情况下,GEP-CSD 算法的时间复杂度为 $O(m \times P \times L \times K \times n \times e)$,在最好情况下,GEP-CSD 算法的时间复杂度是 $O(N_{max} \times P \times L \times K \times n \times e)$,在只考虑网络规模的情况下,算法的时间复杂度为 $O(n \times e)$。

证明 算法中,计算个体针对 $L$ 个训练样本的复杂度为 $O(L)$,算法需计算种群中每个个体的适应度值,而适应度值的计算涉及到 $K \times K$ 阶矩阵,以及需要计算各个社区之间和社区内部的节点与节点之间连接边的数目,还有社区节点的数目,故种群适应度计算的复杂度为 $O(P \times L \times K \times n \times e)$。在最坏的情况下,需要进化 $m$ 代,故在最坏的情况下算法的复杂度为 $O(m \times P \times L \times K \times n \times e)$,在最好的情况下,种群在连续 $N_{max}$ 代内其最高适应度无提高,此时进化过程结束,种群只进化了 $N_{max}$ 代,故在最好情况下,算法的复杂度为 $O(N_{max} \times P \times L \times K \times n \times e)$。在种群大小、总进化代数、进化代数的阈值、样本集长度及社区划分的数目固定的情况下,随着网络规模的不断扩大,算法的复杂度决定于网络节点数目和网络中的边数,因此,此时的算法复杂度为 $O(n \times e)$。

## 3 实验和结果

### 3.1 数据集及参数设置

该实验平台为:Intel Core2 Duo CPU T5670@1.80 GHz CPU;1 GB 内存;Windows XP Professional 操作系统,实验用 C# 语言实现,运行平台为 Visual Studio 2008。为了测试新算法的可行性和有效性,针对两个经典模型 Zachary Karate Club[16] 和 Krebs 采集的网络结构[17-18]进行了验证。实验结果表明,GEP 算法用于复杂网络社区发现问题的解决是行之有效的,具有很高的精度。以下两个实验 GEP-CSD 算法参数设置如下:进化代数为 500,种群大小为 20,函数集为 ∪ 和 ∩,终结符集为 $x_1, x_2, \cdots, x_i$($x_i$ 为第 $i$ 个节点在数据集中的序号),基因头长为 10,基因尾长为 11,基因数为 1,染色体长 21,变异率为 0.044,交叉率为 0.77。

### 3.2 实验结果

实验 1 Zachary Karate Club 网络。

本实验的目的是利用 GEP-CSD 算法来分析 Zachary Karate Club 网络。Zachary Karate Club 网络是 Zachary 用了两年的时间来观察美国一所大学中的空手道俱乐部成员间的相互社会关系,在调查过程中,该俱乐部的校长和主管因是否抬高俱乐部的收费问题而产生了争执,导致该俱乐部分裂成两个分别以主管和校长为中心的小俱乐部,基于这些成员在俱乐部内部及外部的社会关系,他构造了他们之间的关系网。在复杂网络的社区结构分析中 Zachary Karate Club 网络已经成为一个经典的问题。Zachary Karate Club 网络包含 34 个节点,77 条边,本实验利用本文算法把 Zachary Karate Club 网络划分成两个社区,一个是以主管为中心的社区(用圆圈表示),另一个是以校长为中心的社区(用方形表示),以此验证本文算法社区划分精确度。

为了评价 GEP-CSD 算法在精确率的表现,本文采用 GEP-CSD 与文献[19]中提出的一种社区划分的极值优化算法进行对比。该实验用 C# 语言实现,运行平台为 Visual Studio 2008。实验结果如图 3 和图 4 所示。图 3 是 GEP-CSD 算法某次计算时社区划分结果,图 4 是上述极值优化算法社区划分结果。

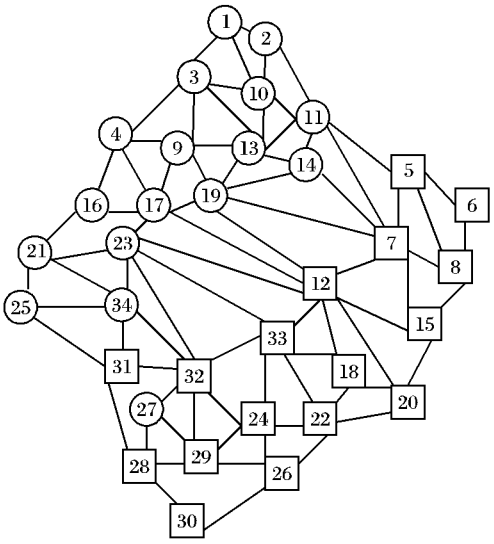



图 3 GEP-CSD 算法某次计算时社区划分结果

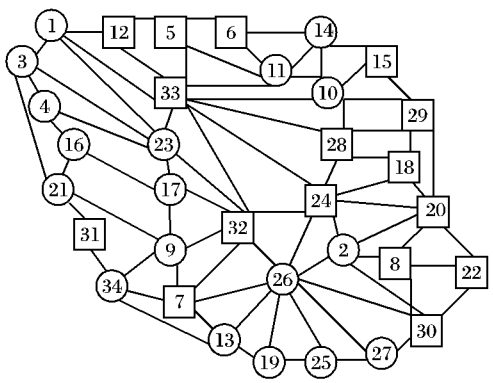



图 4 极值优化算法社区划分结果

实验 2　Krebs 网络。

本实验的目的是利用 GEP-CSD 算法划分 Krebs 网络社区结构,验证本文算法寻找最优解的全局收敛速度。Krebs 网络是一个由一些美国政治性书籍构成的关系网络,一共包括 152 个节点, 449 条边。网络中的节点分别代表带有明显自由主义、保守主义倾向和较为中庸的三种书籍;网络中的边表示其所连接的两本书至少被同一个读者所购买。

为了验证本文算法的全局收敛速度,本文采用 GEP-CSD 算法与文献[15]提出的基于粒子群算法的 Web 社区发现方法,进行了如下实验,表 2 给出了本文算法与 Web 网络社区结构发现算法平均收敛搜索时间的比较。经多次实验表明,本文算法在两种不同的修复策略下,具有较高的收敛速度。

**表 2　两种算法平均收敛搜索时间对比**　　ms

| 修复策略 | | GEP-CSD 算法 | Web 网络社区结构发现算法 |
|---|---|---|---|
| 绝对修复策略 | | 1276.02 | 69900.06 |
| 柔性修复策略 | $k=3.5$ | 978.71 | 4654.71 |
| | $k=3.0$ | 901.12 | 1394.54 |
| | $k=2.5$ | 803.31 | 1103.59 |
| | $k=2.0$ | 798.22 | 1081.45 |
| | $k=1.5$ | 768.27 | 748.29 |
| | $k=1.0$ | 650.67 | 550.93 |

### 3.3　实验分析

实验 1 结果表明,GEP-CSD 算法除了少数进化失败,也就是除了模块度 $Q=1$ 的情况,其他绝大部分进化对网络社区的划分精度超过 97%,只有少数节点被错分到其他社区中(节点 27 被错误分到其他社区中,其错误率小于 2.94%),而文献[19]提出的社区划分的极值优化算法的社区划分结果出现了两个孤立点(节点编号为 31 和 7)。

从实验 2 可以看出,在绝对修复策略下,GEP-CSD 算法总体平均搜索时间较短。当 $k=3.5,3.0,2.5,2.0$,寻找到社区划分最佳方案时,即 $Q$ 值达到最大值时,GEP-CSD 算法所耗费的平均搜索时间较少;在 $k=1.5,1.0$,GEP-CSD 算法达到 $Q$ 最大值时,其所耗费的时间较长。同时可以看出,Web 网络社区结构发现算法对柔性修复策略的阈值较敏感,其平均收敛搜索时间受其影响较大;而 GEP-CSD 算法则受其影响变化范围较小,具有较强的稳定性。由此可以得出结论:在 GEP-CSD 算法中根据不同的实际情况,选择不同的修复策略,对算法的运行效率具有一定的影响;而且在柔性修复策略下,选择适当的阈值,能够降低算法的收敛时间。但 GEP-CSD 算法也存在一定的不足,随着网络规模的扩大,网络节点数和边数增大,GEP-CSD 算法在寻找网络社区结构最佳方案时,所耗费时间也随之增大。

总之,GEP-CSD 方法可以有效地对网络社区进行划分,具有较高的社区划分精度,本文提出的算法时间复杂度总体上有一定的降低,由此可看出本文算法具有较高的全局优化能力,能有效解决局部收敛问题。

## 4　结语

复杂网络已成为当今非常热门的研究领域,相关的算法也日益成熟,本文尝试将 GEP 算法引入网络社区发现,并进一步通过实验进行分析。实验表明 GEP-CSD 算法能较好地完成对网络社区的划分,相比其他算法而言,本算法无需预先知道社区网络的大小,且具有较高的社区划分精度。GEP-CSD 算法还有待进一步研究,下一步的研究工作包括 $k$-社区划分方法,不同种群大小对算法的时间复杂度和收敛速度的影响,以及不同的修复策略下算法的社区划分精度等;同时还可以研究相应的并行 GEP 网络社区划分方法。

**参考文献:**


[1] DAI FEI-FEI, TANG PU-YING. Community structure detection in complex networks using DNA genetic algorithm [J]. Computer Engineering and Applications, 2008, 44(3): 53 – 56.

[2] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks [J]. Proceedings of National Academy of Science, 2002, 99(12): 7821 – 7826.

[3] KERNIGHAN B W, LIN S. An efficient heuristic procedure for partitioning graphs [J]. Bell System Technical Journal, 1970, 49(1): 291 – 307

[4] FIEDLER M. Algebraic connectivity of graphs [J]. Czechoslovak Mathematical Journal, 1973, 23(98): 298 – 305.

[5] POTHEN A, SIMON H, LIU P, *et al*. Partitioning sparse matrices with eigenvectors of graphs [J]. SIAM Journal on Matrix Analysis and Applications, 1990, 11(3): 430 – 452

[6] FOLEY T A. Local control of interval tension using weighted splines [J]. Computer Aided Geometric Design, 1986, 3(2): 281 – 294.

[7] 元昌安, 彭昱忠, 覃晓, 等. 基因表达式编程算法原理与应用[M]. 北京: 科学出版社, 2010.

[8] FERREIRA C. Gene expression programming: A new adaptive algorithm for solving problems [J]. Complex Systems, 2001, 13(2): 87 – 129.

[9] FERREIRA C. Gene expression programming in problem solving [C]// Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications. Berlin: Springer-Verlag, 2001: 635 – 654.

[10] 元昌安, 唐常杰, 左劼, 等, 基于基因表达式编程的函数挖掘-收敛性分析与残差制导进化算法[J]. 四川大学学报: 工程科学版, 2004, 36(6): 100 – 105.

[11] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69(2): 16.

[12] HAN JIAWEI, KAMBR M. Data mining—concepts and techniques [M]. 影印版. 北京: 高等教育出版社, 2001.

[13] 黄晓冬, 唐常杰, 李智, 等. 基于基因表达式编程挖掘函数关系[J]. 软件学报, 2004, 15(zk): 96 – 105.

[14] FERREIRA C. Gene expression programming: A new adaptive algorithm for solving problems[J]. Complex Systems, 2001, 13(2): 87 – 129.

[15] 段晓东, 王存睿, 刘向东, 等. 基于粒子群算法的 Web 社区发现[J]. 计算机科学, 2008, 35(3): 18 – 22.

[16] ZACHARY W W. An information flow model for conflict and fission in small groups [J]. Journal of Anthropological Research, 1977, 33(4): 452 – 473.

[17] NEWMAN M E J. Modularity and community structure in networks [J]. Proceedings of the National Academy of Sciences, 2006, 103 (23): 8577 – 8582.

[18] ADAMIC L A, GLANCE N. The political blogosphere and the 2004 U. S. election: Divided they blog [C]// LinkKDD '05: Proceedings of the 3rd International Workshop on Link Discovery. New York: ACM, 2005: 36 – 43.

[19] DUCH J, ARENAS A. Community detection in complex networks using extremal optimization [J]. Physical Review E, 2005, 72(2): 027104.