计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20170751
56(2):281-292, 2019

基于多目标演化聚类的大规模动态网络社区检测

李 赫 印 莹 李 源 赵宇海 王国仁
(东北大学计算机科学与工程学院 沈阳 110819)
(15040107713@163.com)

Large-Scale Dynamic Network Community Detection by Multi-Objective Evolutionary Clustering

Li He, Yin Ying, Li Yuan, Zhao Yuhai, and Wang Guoren
(*College of Computer Science and Engineering*, *Northeastern University*, *Shenyang* 110819)

Abstract Evolutionary clustering is often utilized for dynamic network community detection to uncover the evolution of community structure over time. However, it has the following main problems: 1) The absence of error correction may lead to the result-drifting problem and the error accumulation problem; 2) the NP-hardness of modularity based community detection makes it inefficient to get an exact solution. In this paper, an efficient and effective multi-objective method, namely DYN-MODPSO(multi-objective discrete particle swarm optimization for dynamic network), is proposed, where the traditional evolutionary clustering framework and the particle swarm algorithm are modified and enhanced, respectively. The main work of this article is as follows: 1) A novel strategy, namely the recently future reference, is devised for the initial clustering result correction to make the dynamic community detection more effective; 2) the traditional particle swarm algorithm is modified so that it could be effectively integrated with the evolutionary clustering framework; 3) the de-redundancy random walk based initial population generation method is presented to improve the diversity and the initial precision of the individuals; 4) the multi-individual crossover operator and the improved interference operator are developed to enhance the local search and the convergence abilities of DYN-MODPSO. Extensive experiments conducted on the real and the synthetic dynamic networks show that the efficiency and the effectiveness of DYN-MODPSO are significantly better than those of the competitors.

Key words dynamic network community detection; evolutionary clustering; multi-objective optimization; random walk; particle swarm algorithm

摘 要 动态网络社区检测能揭示社区结构随时间演变的规律,是目前网络社区研究领域的热点之一.基于演化聚类的方法被广泛采用,但存在2个主要问题:1)缺乏结果校正机制,容易产生“结果漂移”和“误差累积”问题;2)问题的NP-难本质,导致基于模块度的精确社区结构检测在效率上存在很大问题.针对以上问题,通过对传统演化聚类框架和离散粒子群算法的改进及有效结合,提出一种高效且有效的

收稿日期:2017-09-30;**修回日期**:2018-04-08
基金项目:国家自然科学基金项目(61772124,61332014);中央高校基本科研业务费专项资金(N150404008,N150402002)
This work was supported by the National Natural Science Foundation of China (61772124, 61332014) and the Fundamental Research Funds for the Central Universities (N150404008, N150402002).
通信作者:印莹(yinying@cse.neu.edu.cn)

多目标动态社区检测方法(multi-objective discrete particle swarm optimization for dynamic network, DYN-MODPSO),主要工作包括:1)提出基于最近未来参考策略的初始聚类结果校正方法,提高动态社区检测结果的有效性;2)改进传统粒子群算法,使其能与演化聚类框架有效结合;3)提出基于去冗余的随机游走初始群体生成方法,提高传统粒子群算法中的个体多样性并保证个体的初始精度;4)提出多个体交叉算子及改进的干扰算子,提高算法的局部搜索能力与收敛能力.大量基于真实和人工动态网络数据的实验结果证实,提出的方法在效率和有效性方面,显著优于同类比较算法.

关键词 动态网络社区检测;演化聚类;多目标优化;随机行走;粒子群算法

中图法分类号 TP391

真实世界中,网络的节点和边通常会随时间动态地变化,这导致了网络中的社团结构也会随着时间发生改变.从 2005 年开始,对静态网络缺失的动态特征的研究逐渐成为了研究者们关注的热点[1].动态网络可以表示成由各个时刻静态网络组成的快照序列,动态网络社区检测的目的就是准确地挖掘出每一时刻快照的社区结构,从而可以分析社区结构随着时间的演变过程,这是无法通过静态网络社区检测洞察的.

动态网络社区检测有广泛的应用.在基因网络分析方面,动态网络社区检测能揭示特征基因集随时间的变化过程;在电商数据分析方面,动态网络社区检测能发现用户的偏好变化情况;在社交网络数据分析方面,动态网络社区检测能找出兴趣团体,预测团体可能参加的活动.此外,在新闻标题内容分析、论文作者合作关系分析、金融股市分析等方面,动态网络社区检测也有着广泛的应用.随着社区检测技术的发展,动态网络社区检测将会在越来越多的实际网络中得到应用,并发挥巨大的作用.

演化聚类框架是动态社区检测的主流方法之一,其基本思想是在第一时刻网络快照社区检测的基础上,根据社区结构的时间平滑性,用前一时刻的社区检测结果指导当前时刻的社区划分,以提高动态网络社区检测的效率.为了避免噪音等因素的影响,保证动态社区检测的准确性,许多文献把演化聚类框架引入到动态网络社区中来.但是,演化聚类存在以下 2 个问题:1)在有效性方面,若初始社区结构检测不准确,会导致后续社区结构检测的"结果漂移"和"误差累积"问题,即上一个结果不准确将导致下一个聚类不准确,并导致这种不准确性愈演愈烈;2)在效率方面,基于模块度优化的社区检测方法是 NP 难的[2],许多精确算法无法在合理的时间内解决该问题.

针对以上问题,本文提出一种基于改进演化聚类框架和离散粒子群算法的多目标动态社区检测方法,本文的主要贡献有 4 个方面:

1) 提出基于最近未来参考的演化聚类框架,提高初始聚类准确性,保证动态网络社区检测的可靠性.

2) 离散粒子群优化算法与基于精英策略的非支配排序遗传算法(NSGA-Ⅱ)结合,并基于演化聚类框架,利用前一时刻社区划分结果来快速指导粒子群算法搜索当前快照中的社团结构.

3) 提出基于去冗余策略的随机游走初始个体生成算法 DIGRW,对粒子群的位置进行初始化,提高了初始粒子群的个体多样性和个体精度.

4) 提出多个体交叉算子,增强算法的局部搜索能力,提高算法的收敛速度.

1 相关工作

由于动态网络社区检测能揭示静态网络检测无法洞察的社区结构随时间变化的规律,所以,自动态网络社区检测这一概念出现以来,一系列针对该问题的相关算法被先后提出.如 Sarkar 等人[3]提出利用数据挖掘技术分析动态网络的方法——基于潜在空间模型的动态网络社区检测方法.Cordeiro 等人[4]提出一种基于本地模块度优化的动态社区自适应发现算法.Li 等人[5]提出了一种基于增量识别的聚类方法来解决动态网络社区检测问题.Lander 等人[6]提出多目标图挖掘算法,用来在复杂动态网络中挖掘和检测社区结构.Ma 等人[7]提出进化非负矩阵因子分解算法来发现动态网络中社区的演变规律.

由于基于模块度优化的社区检测方法是经典 NP 难问题,上述精确算法在处理大规模动态网络社区检测时面临很大挑战.与精确算法相比,演化算法在处理大数据过程中具有较明显的优势[8].特别地,

由于演化算法具有高度伸缩性、灵活性、全局优化等能力,并在特征选择时可同时实现针对多目标的优化,因此演化聚类算法在动态网络分析领域中逐渐崭露头角.

Chakrabarti 等人[9]首次提出演化聚类框架,该框架指出动态网络具有时间平滑性,即相邻时刻动态网络前后变化差距不大. Kim 等人[10]为了提高效率,进一步提出基于演化聚类框架的动态网络社区检测方法,即粒子和密度的演化聚类方法来分析动态网络的社区结构. Pizzuti 等人[11]提出经典的 DYN-MOGA 算法. 首次将多目标优化方法和演化聚类框架结合用于动态网络社区检测,使算法具有隐并行性,既保证了当前时刻社区划分质量,又保证了当前时刻社区划分与前一时刻社区划分的相似度较大. 这些算法将演化聚类和演化算法结合,用于动态网络社区检测,在处理大规模网络效率上有所提高,但有效性仍存在问题.

针对现有同类算法存在的以上问题,本文提出一种基于多目标演化聚类的动态网络社区检测算法 DYN-MODPSO,既提高了效率,又保证了结果的有效性.

2　基本概念及问题定义

本节主要描述算法执行过程中涉及的基本概念,并对要解决的主要问题给出具体定义.

2.1　基本概念

动态网络社区检测主要涉及 2 个基本概念,一个是对某个社区划分好坏的评估,另一个是对不同社区划分相似性的评估. 以下分别描述这 2 个概念.

给定动态网络 $N=\{N^1,N^2,\cdots,N^m\}$,其中 N^t 表示时刻 t 的动态网络快照,$t=1,2,\cdots,m$. 记时刻 t 网络快照的社区划分为 $C^t=\{C_1^t,C_2^t,\cdots,C_k^t\}$,本文以 Pizzuti 提出的社区分数 CS(community score)作为社区划分 C^t 的评估函数[11]. 其中 CS 的适应度函数 F_{CS}的具体定义如下:

$$F_{CS}(C^t)=\sum_{i=1}^{k} score(C_i^t). \tag{1}$$

其中,函数 $score(C_i^t)$表示社区划分 C_i^t 的质量,公式如下:

$$score(C_i^t)=\frac{\sum_{m\in C_i^t}(\mu_m)^2}{|C_i^t|}\times\sum_{m,n\in C_i^t}A_{mn}^t, \tag{2}$$

$$\mu_m=\frac{1}{|C_i^t|}\sum_{n\in C_i^t}A_{mn}^t. \tag{3}$$

其中,C_i^t 表示时刻 t 第 i 个社区,μ_m 表示 C_i^t 内从节点 m 出发的位于 C_i^t 内部的边的分数,$\boldsymbol{A}^t$ 代表时刻 t 网络快照的邻接矩阵,$|C_i^t|$表示社区 C_i^t 内部的节点数,$score(C_i^t)$表示社区划分 C^t 中的社区 C_i^t 的得分.

从式(2)～(3)可以看出,社区 C_i^t 内部的边越密集,$score(C_i^t)$的值越大. $F_{CS}(C^t)$把划分 C^t 中各个社区得分情况 $score(C_i^t)$相加,若 $F_{CS}(C^t)$值越大,则表示每个社区的得分 CS 普遍偏大,社区内部连接越紧密,也说明了社区之间的连接越稀疏.

NMI(normalized mutual information)是归一化互信息函数,它用来测量 2 个社区划分的相似度[12]. 设时刻 t 网络划分为 $C^t=\{C_1^t,C_2^t,\cdots,C_k^t\}$,时刻 $t-1$ 网络划分方案为 $C^{t-1}=\{C_1^{t-1},C_2^{t-1},\cdots,C_k^{t-1}\}$,互信息 $N_{NMI}(C^t,C^{t-1})$定义如下:

$$N_{NMI}(C^t,C^{t-1})=\frac{-2\sum_{i=1}^{m}\sum_{j=1}^{n}L_{ij}\lg(L_{ij}N/L_{i.}L_{.j})}{\sum_{i=1}^{m}L_{i.}\lg\left(\frac{L_{i.}}{Num}\right)+\sum_{j=1}^{n}L_{.j}\lg\left(\frac{L_{.j}}{Num}\right)}. \tag{4}$$

其中,$\boldsymbol{L}$ 为混淆矩阵,L_{ij}表示既在社区 C_i^t 中,又在社区 C_j^{t-1} 中的节点数量,其中 $C_i^t\in C^t$,$C_j^{t-1}\in C^{t-1}$. $L_{i.}$是 $\boldsymbol{L}$ 中第 i 行元素的和,表示既属于当前社区 C_i^t 内部,又属于前一时刻网络快照节点的个数,即社区 C_i^t 内部节点的个数. 同理,$L_{.j}$是 $\boldsymbol{L}$ 中第 j 列元素的和,即社区 C_j^{t-1} 内部节点的个数. m 和 n 分别代表社区划分 C^t 和 C^{t-1}中的社区数量,Num 是社区划分 CR^t 和 CR^{t-1}所对应快照的节点数. $N_{NMI}(C^t,C^{t-1})\in[0,1]$,$N_{NMI}(C^t,C^{t-1})$越大,说明 C^t 和 C^{t-1}越相似.

2.2　问题定义

动态网络可以建模为图 $N=\{N^1,N^2,\cdots,N^T\}$的形式,其中 T 表示时刻,N^i 表示时刻 i 的网络快照. 社区是一系列的节点集合,这些集合内部节点具有强关联关系,集合间的节点具有弱关联关系. 动态网络社区检测就是要找出不同时刻网络快照中存在的真实社区结构. 为了定量地衡量社区结构划分好坏,本文采用社区分数 CS 来评估社区划分质量,用归一化互信息函数 NMI 来评估 2 种社区划分的相似性.

3 基于改进粒子群的社区检测算法 MRDPSO

基于模块度的动态社区检测是 NP 难问题,因此本文提出将粒子群优化算法和演化聚类框架相结合的有效解决方法. 传统粒子群算法主要用于处理静态网络数据,不能和演化聚类框架相结合. 本文对文献[11]提出的离散粒子群算法进行了改进,提出了算法 MRDPSO(multi-results discrete particle swarm optimization). MRDPSO 与现有的粒子群算法不同,主要从 3 个方面进行了改进:1)改进了粒子群算法的输出,可以输出多个最优解,避免了最优解的遗漏问题;2)用基于去冗余的随机游走初始群体生成方法初始化粒子群,提高粒子群中个体多样性并保证个体初始精度;3)提出多个体交叉算子(multi-individuals crossover operator, MICO)及改进的干扰算子 NBM^+ (neighbor based mutation),增强算法的局部搜索能力.

3.1 MRDPSO 总体描述

MRDPSO 伪代码如算法 1 所示. MRDPSO 框架由 3 个主要部分构成: 1)初始化阶段,利用基于去冗余策略的随机游走初始群体生成算法 DIGRW 初始化粒子群的位置信息,保证了初始种群个体具有一定精度和多样性,避免算法陷入早熟(行①～④);2)搜索阶段,对多目标粒子群社区检测方法进行改进,使得粒子群的全局最优位置都保留下来,使算法能够适应动态社区检测的需求(行⑤～⑯);3)突变阶段,提出多个体交叉算子 MICO(行⑭)及改进的干扰算子 NBM^+(行⑨),使得粒子群在向全局最优解靠近的同时,能够在小范围的精英粒子群体中进行局部搜索,提高算法发现全局最优解的效率. 3.2～3.4 节将对这 3 个主要步骤分别进行描述.

算法 1. MRDPSO.

输入:静态网络 $G=\{V,E\}$;

输出:静态网络的多个可能的社区划分方案 $g_{\text{best}}=\{\boldsymbol{g}_{\text{best}}^1,\boldsymbol{g}_{\text{best}}^2,\cdots,\boldsymbol{g}_{\text{best}}^k\}$.

① 粒子群位置初始化:利用 DIGRW 得到所有粒子的位置 $P=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{pop}\}$;

② 粒子群速度初始化:初始化所有粒子速度 $V=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_{pop}\}$,令 $\boldsymbol{v}_i=\boldsymbol{0}$,其中 $i=1,2,\cdots,pop$;

③ 初始化粒子群所有粒子的历史最优位置:$p_{\text{best}}=\{\boldsymbol{p}_{\text{best}}^1,\boldsymbol{p}_{\text{best}}^2,\cdots,\boldsymbol{p}_{\text{best}}^{pop}\}$;

④ 初始化粒子全局最优位置:计算所有粒子的适应度值,选出最大的粒子 i 作为最优粒子,最优位置 $g_{\text{best}}=\{\boldsymbol{p}_{\text{best}}^i\}$,可以有多个最优粒子;

⑤ 迭代,$t=0$;

⑥ for $i=1,2,\cdots,pop$ do

⑦ 更新第 i 个粒子的速度 $\boldsymbol{v}_i^{t+1}$;

⑧ 更新第 i 个粒子的位置 $\boldsymbol{x}_i^{t+1}$;

⑨ 以一定概率 pm 对 $\boldsymbol{x}_i^{t+1}$ 采用干扰算子 NBM^+;

⑩ 计算粒子 i 的适应度 $F_{\text{CS}}(\boldsymbol{x}_i^{t+1})$;

⑪ 若 $F_{\text{CS}}(\boldsymbol{x}_i^{t+1})>F_{\text{CS}}(\boldsymbol{p}_{\text{best}}^i)$,更新粒子的历史最优位置,令 $\boldsymbol{p}_{\text{best}}^i=\boldsymbol{x}_i^{t+1}$;

⑫ 若 $F_{\text{CS}}(\boldsymbol{x}_i^{t+1})=F_{\text{CS}}(\boldsymbol{g}_{\text{best}}^1)$,更新全局最优位置,把向量 $\boldsymbol{x}_i^{t+1}$ 加入到集合 g_{best} 中;

⑬ 若 $F_{\text{CS}}(\boldsymbol{x}_i^{t+1})>F_{\text{CS}}(\boldsymbol{g}_{\text{best}}^1)$,则 $g_{\text{best}}=\{\boldsymbol{x}_i^{t+1}\}$;

⑭ 对 $p_{\text{best}}=\{\boldsymbol{p}_{\text{best}}^1,\boldsymbol{p}_{\text{best}}^2,\cdots,\boldsymbol{p}_{\text{best}}^{pop}\}$ 采用交叉算子 MICO,若产生的个体优于 $\boldsymbol{g}_{\text{best}}^1$,则更新 g_{best};

⑮ 迭代终止条件:如果 $t<maxgen$,t++,转到⑥,否则停止算法并输出 g_{best};

⑯ end for

3.2 粒子群初始化

3.2.1 编码方式

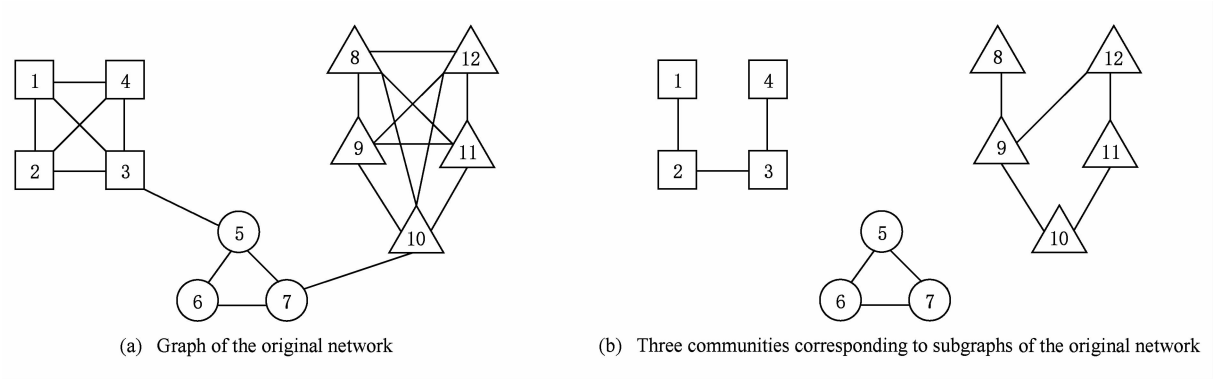
在用演化计算进行社区检测的过程中,每个个体作为一种社区划分方案,都以编码的方式存在. 目前的编码方式主要有字符串编码方式和位邻接编码方式[13].

图 1(a)为一个原始网络;图 1(b)为原始网络的 3 个子图,每个子图表示一个社区. 图 2 的位邻接编码对应图 1(b)所示的划分,图 2 中字符串编码为位邻接编码的解码. 在位邻接编码方式中,如果节点 i 的基因值是 j,则节点 i 与节点 j 位于相同社区;在字符串编码方式中,如果节点 i 的基因值为 k,则节点 i 在标号为 k 的社区中.

如图 2 所示,基于位邻接的编码方式先转换成字符串编码,然后再解码成社区划分 $C=\{C_1,C_2,C_3\}$ 的集合形式. 所以位邻接编码方式解码困难、效率低,故本文采用直观、高效的字符串编码方式.

3.2.2 初始化粒子群位置

如果初始粒子群有过多的冗余个体,就不能保证初始种群个体有较强的多样性. 如图 3 所示.



(a) Graph of the original network　(b) Three communities corresponding to subgraphs of the original network

Fig. 1　Community division of the network

图 1　网络的社区划分

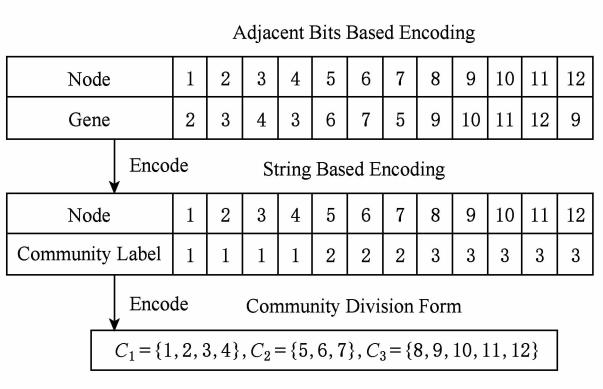
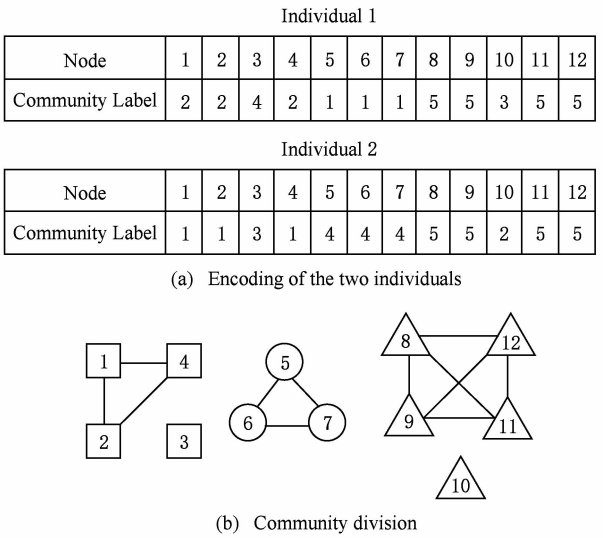


Fig. 2　The encoding scheme

图 2　编码方式



(a) Encoding of the two individuals

(b) Community division

Fig. 3　The encoding scheme and the corresponding community division

图 3　编码方式与对应的社区划分

图 3(a)中 2 种不同的字符串编码均对应如图 3(b)所示的划分,因此图 3 中的 2 种字符串编码称为冗余编码. 过多的冗余个体很容易导致算法陷入局部最优,或者算法的收敛速度降低.

针对这一问题,本文提出基于去冗余策略的随机游走初始群体生成算法 DIGRW(different individual generation based on random walk). 算法 DIGRW 大致过程有 4 个步骤:1)用基于随机行走的初始群体生成算法生成一个粒子位置 $\boldsymbol{p}$;2)用字符串去冗余方法(redundancy removal method, RRM)将个体 $\boldsymbol{p}$ 唯一化;3)若个体不重复,保留到种群中,否则删除;4)当种群个数达到上限时,停止生成个体. 后 2 个步骤比较直观,以下主要对 DIGRW 算法的前 2 步加以描述.

随机游走初始化社区划分的理论基础是:根据社区的连接属性,即社区内的连接密度远大于社区外的连接密度,那么如果起始节点和目的节点位于同一社区,则会有更多经过 k 步到达目的节点的路径存在;反之,经过 k 步到达目的节点概率很小. 基于随机游走的初始化社区划分基本过程为:1)随机选定一个目的节点 n,计算每个节点经过 k 步到 n 的概率;2)将所有节点依照概率值降序排列,在排序后的节点中找出二分分列点,使得模块度函数 Q 增加最大;3)如果不存在使函数 Q 增大的分裂点,递归终止,否则分裂点将当前网络分裂成 2 个子网络,并分别对其递归执行上述操作.

算法 2. $RRM(\boldsymbol{p})$.

输入:编码 $\boldsymbol{p}$;

输出:$\boldsymbol{p}$ 所对应的唯一编码 $\boldsymbol{u}$.

① $u[1]=1$;

② $\max=u[1]$;

③ for int $i=0,1,\cdots,|\boldsymbol{p}|-1$ do

④　for int $j=0,1,\cdots,i$ do

⑤　if($p[i]\neq p[j]$)

⑥　continue;

⑦　else

```
⑧          u[i]=u[j];
⑨          break;
⑩        end if
⑪        if((j==i) && (u[i]==0))
⑫          u[i]=++max;
⑬        end if
⑭      end for
⑮ end for
```

字符串去冗余策略 RRM 是将给定的编码标准化,使其成为能唯一表示该编码所对应社区划分的形式.算法 RRM 步骤如算法 1 所示.RRM 的本质是对节点所属的社区标号进行调整.在每个编码 $\boldsymbol{p}$ 中,规定节点 1 所属社区的标号为 1,即 $u[1]=1$(行①～②).从节点 2 开始,依次对节点的社区标号进行调整:如果节点 i 与前面的节点 j 处于相同社区,那么 $u[i]=u[j]$,否则如果节点 i 与前面的每个节点都不在一个社区,那么将当前最大的社区标号值加 1,作为 $u[i]$的社区标号(行③～⑮).

3.3 粒子群搜索

粒子群初始化完毕后,进入迭代搜索阶段.每一代粒子都会记录所有的全局最优位置 g_{best} 和每个粒子的历史最优位置 $\boldsymbol{p}_i^t$.在迭代过程中,粒子按照式(5)(6)更新粒子群速度和位置,

$$\boldsymbol{v}_i^{t+1}=sig(w\boldsymbol{v}_i^t+W_1r_1(\boldsymbol{p}_i^t\oplus\boldsymbol{x}_i^t)+W_2r_2(g_{\text{best}}\oplus\boldsymbol{x}_i^t)),\tag{5}$$

$$\boldsymbol{x}_i^{t+1}=\boldsymbol{x}_i^t\otimes\boldsymbol{v}_i^{t+1}.\tag{6}$$

其中,w 是惯性权重,W_1 是认知系数,W_2 是社会系数,r_1 和 r_2 分别是[0,1]之间的随机数,⊕是异或运算符.$\boldsymbol{p}_i^t$ 是粒子 i 在时刻 t 的最优位置,g_{best} 是所有粒子的最优位置集合,sig 和⊗对应文献[13]中的具体操作.对更新后的位置信息以一定概率进行突变,作为当代粒子的位置.计算粒子的适应度值,根据适应度值更新粒子的历史最优位置和全局最优位置.用交叉算子 MICO 处理所有粒子的历史最优位置集合 p_{best},得到新的位置后再次更新全局最优位置集合 g_{best}.粒子群通过迭代过程,不断更新优化 g_{best},当迭代终止时,输出 g_{best} 中的所有粒子位置作为多个可行的最优解.

3.4 干扰算子和交叉算子

粒子群优化算法可以通过结合遗传操作中的交叉和变异操作来保留最优粒子,增强种群多样性和增加跳出局部最优区域的能力.本研究中分别提出一种新的交叉算子 MICO 和一种新的干扰算子 NBM$^+$来实现粒子群优化过程中的个体交叉和变异操作.以下分别对这 2 个算子进行介绍.

3.4.1 多个体交叉算子 MICO

传统交叉算子仅是针对 2 个父代个体,与此不同,本文提出一种新的多个体交叉算子.受聚类融合算法平衡多个聚类结果获得更准确结果的思想启发[14-17],本文将遗传算法中传统 2 个父代个体交叉算子扩展成多个体交叉算子,提出多个体交叉算子 MICO.

MICO 交叉操作过程有 3 个:1)从所有粒子的历史最优位置 p_{best} 中随机挑选出 M 个位置;2)设交叉产生的新位置为 $\boldsymbol{x}$,将 $x[i]$赋值为 M 个位置中第 i 个元素上出现次数最多的社区标号,如果有多个出现最多的社区标号,则随机选一个赋值;3)交叉完成,产生新位置 $\boldsymbol{x}$,对 $\boldsymbol{x}$ 采用去冗余策略 RRM 去冗余.

如图 4 所示,若 $M=5$,则随机选出 5 个位置向量 $\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_5$,对它们进行多个体交叉操作,产生的新位置向量记为 $\boldsymbol{x}$.现在对基因位 $x[3]$进行赋值,可以看出 $\boldsymbol{x}_1$～$\boldsymbol{x}_5$ 的第 3 个基因值中的社区标号 2 出现次数最多,则令 $x[3]$的基因值为 2.以此类推,$\boldsymbol{x}$ 的所有基因位都按照这个规律赋值,最终 $\boldsymbol{x}=(1,1,2,1,2,3,3,3)$.

	1	2	3	4	5	6	7	8
$\boldsymbol{x}_1$	1	1	2	2	2	3	3	3
$\boldsymbol{x}_2$	1	1	2	1	2	3	3	3
$\boldsymbol{x}_3$	1	1	1	2	2	3	3	3
$\boldsymbol{x}_4$	1	2	2	1	2	3	3	3
$\boldsymbol{x}_5$	1	1	2	1	1	3	3	3
$\boldsymbol{x}$	1	1	2	1	2	3	3	3

Fig. 4 Search operation based on elitist-crossover

图 4 精英交叉搜索算子

3.4.2 干扰算子 NBM$^+$

为了提高粒子群算法的局部搜索能力,文献[14]提出干扰算子 NBM.然而,NBM 会产生冗余个体,故本文在此基础上加入了字符串去冗余策略,提出改进的干扰算子 NBM$^+$来增强解的多样性,避免算法陷入局部最优.

NBM$^+$的过程有 3 步:1)生成一个[0,1]之间的随机数 m;2)判断 m 与突变概率 pm 之间的关系,若 $m<pm$,则对粒子的位置向量进行突变操作,即随

机选择一个粒子位置向量的基因位,把该基因所对应节点的社区标号赋给它的所有邻居节点;3)对新产生的位置向量,采用去冗余策略 RRM 去冗余.

4　算法 DYN-MODPSO

算法 MODPSO 处理的是单时间片上静态网络聚类问题,本节基于前面提出的单快照社区检测算法 MRDPSO,进一步提出基于演化聚类框架的动态网络社区检测算法 DYN-MODPSO(multi-objective discrete particle swarm optimization for dynamic network)来处理动态网络社区检测问题.

4.1　算法总体描述

DYN-MODPSO 伪代码如算法 3 所示.

算法 3. DYN-MODPSO.

输入:动态图 $N=\{N^1,N^2,\cdots,N^T\}$,时刻 T,最大迭代次数 $genmax$;

输出:N^t 的聚类结果 C^t.

① while($t==1 \parallel t==2$)
② 　用MRDPSO 处理 N^t,得到一组社区划分方案 $D^t=\{D_1^t,D_2^t,\cdots,D_k^t\}$,$t$++;
③ 　if ($t==2$),计算 $N_{\mathrm{NMI}}(D_i^{t-1},D_j^t)$,$i,j\in[1,m]$,找出使 N_{NMI} 值最大的 D_i^{t-1},D_j^t,分别返回 N^1,N^2 的划分结果 $C^1=D_i^{t-1}$,$C^2=D_j^t$;
④ 　end if
⑤ 　for $t=3$ to T do
⑥ 　　利用 DIGRW 初始化粒子群的位置,初始化 $gen=1$,初始化粒子历史最优位置 $\boldsymbol{p}_{\mathrm{best}}^i=\boldsymbol{x}_i^t$,并将 $\boldsymbol{p}_{\mathrm{best}}^i$存入 $poplist$ 中,初始化粒子的速度向量和全局最优位置;
⑦ 　　while ($gen\leqslant genmax$)
　　/ * 粒子群迭代 * /
⑧ 　　for $i=1,2,\cdots,pop$ do
⑨ 　　更新第 i 个粒子的速度 $\boldsymbol{v}_i^{gen+1}$;
⑩ 　　更新第 i 个粒子的位置 $\boldsymbol{x}_i^{gen+1}$;
⑪ 　　依概率对 $\boldsymbol{x}_i^{gen+1}$ 用算子 NBM^+,更新 $\boldsymbol{p}_{\mathrm{best}}^i$:如果 $\boldsymbol{x}_i^{gen+1}\succ\boldsymbol{p}_{\mathrm{best}}^i$,则令 $\boldsymbol{p}_{\mathrm{best}}^i=\boldsymbol{x}_i^{gen+1}$,将更新后的 $\boldsymbol{p}_{\mathrm{best}}^i$存入 $poplist$;
⑫ 　　end for
⑬ 　　对 $p_{\mathrm{best}}=\{\boldsymbol{p}_{\mathrm{best}}^1,\boldsymbol{p}_{\mathrm{best}}^2,\cdots,\boldsymbol{p}_{\mathrm{best}}^{pop}\}$ 用算子 MICO,产新个体 $\boldsymbol{x}$,存入 $poplist$ 中;
⑭ 　　更新 gen 代粒子的全局最优位置:将 $poplist$ 中的位置解码,计算 $poplist$ 中粒子位置的 2 个目标函数 $F_{\mathrm{CS}}(\boldsymbol{x}_i^{gen})$ 和 $N_{\mathrm{NMI}}(\boldsymbol{x}_i^{gen-1},\boldsymbol{x}_i^{gen})$;
⑮ 　　对 $poplist$ 中的 $pop+1$ 个位置进行非支配排序,相同支配等级的个体再按拥挤距离排序;
⑯ 　　保留 $poplist$ 前 pop 位置,其余删除;
⑰ 　　更新 g_{best}:选 $poplist$ 中非支配等级为 1 的粒子位置,存入 g_{best}中;
⑱ 　　迭代终止条件:若 $gen<maxgen$,gen++,转到⑦,否则停止算法并从 g_{best}中找出使模块度值 Q 最大的个体输出;
⑲ 　　end while
⑳ 　　t++,若 $t>T$,退出,否则返回⑤;
㉑ 　end for
㉒ end while

算法 3 由 2 个主要部分构成:1)基准聚类校正,此阶段分别用 *MRDPSO* 处理动态网络中的前 2 张快照,通过计算不同快照中社区划分的相似性,基于时间平滑性原理对初始社区的检测结果进行校正,避免因快照 1 聚类结果不准导致快照 2 聚类结果不准的问题(行①～④);2)多目标演化聚类.此阶段在基准聚类校正的基础上,将多目标优化算法 *NSGA-*Ⅱ与 *MRDPSO* 融合,处理后续快照的社区检测问题(行⑤～㉒).以下各节将对这 2 个主要步骤分别进行描述.

4.2　基准聚类校正

前面改进的粒子群算法 MRDPSO 在与演化聚类结合处理动态社区检测的过程中,仍然会产生“结果漂移”和“误差累积”的问题.为解决该问题,本文提出基于最近未来参考策略的基准聚类校正方法,保证初始聚类结果的准确性,从而提高动态社区检测结果的有效性.

动态网络社区检测过程中初始聚类结果准确性非常重要,一旦初始聚类结果与真实结果存在着明显差异,那么根据时间平滑性假设,后续快照中的聚类结果也会与真实聚类结果存在显著差异,甚至这种差异会随着时步的增加越来越大,本研究提出的基本解决思路是:第 1 张快照和第 2 张快照同时进行社区检测,分别参照彼此的聚类信息.由于可以彼此参照相互的聚类信息,避免了第 1 张快照因社区检测过程中无参考而导致的聚类不准确问题.具体

实现过程中,用单时间片社区检测算法 MRDPSO 分别处理快照 1 和快照 2,得到对应的社区划分方案 $D^1=\{D_1^1,D_2^1,\cdots,D_m^1\}$ 和 $D^2=\{D_1^2,D_2^2,\cdots,D_n^2\}$,依次计算 $N_{\mathrm{NMI}}(D_i^1,D_j^2)$,其中 $i\in[1,m]$,$j\in[1,n]$.找出使 NMI 值最大的 2 个划分,分别作为快照 1 和快照 2 的聚类结果.注意:在实际动态网络分析中,可以根据用户需求,对指定的前 k 个快照执行基准校正过程.

4.3 多目标演化聚类

执行基准聚类校正后,对后续快照进行处理.首先用 DIGRW 对粒子群赋初值(行⑥～⑦),粒子群进行迭代搜索,每一代粒子更新位置向量,然后位置向量通过干扰算子 NBM^+ 进行突变.计算突变后粒子位置的 CS 与 NMI 值,如果突变后位置的 CS 与 NMI 都优于粒子历史最优位置的 CS 与 NMI 值,则更新粒子的自身最优位置 $\boldsymbol{p}_{\mathrm{best}}^i$(行⑦～⑬).把所有粒子的历史最优位置存入 *poplist*,对 *poplist* 采用交叉算子 MICO,产生的新个体加入 *poplist*.

对 *poplist* 采用多目标优化算法 NSGA-Ⅱ中的非支配排序和拥挤距离排序[18-20],根据动态网络社区的时间平滑性,选出社区划分质量好,同时又与前一张快照划分最相似的解作为当前快照的划分结果(行⑭～⑱).依照这样的规律,DYN-MODPSO 对每一张动态图快照都进行关联性地处理,直到处理完最后一个动态图快照 N^{T},输出最优社区划分方案 C^{T},算法结束(行⑲～㉒).粒子群的非支配排序过程,就是粒子通过两两比较 CS 与 NMI 后,按照适应度值从大到小进行的排序过程.粒子群的拥挤距离排序过程,就是在同一个支配等级中,即适应度值相同的条件下,选择互不相似的粒子排在前面,将比较与前面粒子比较相似的粒子排在后面,这样能够避免得到的解扎堆聚集,从而保证解的多样性.

5 实验结果与分析

5.1 实验环境配置

本文分别用人工网络和真实世界网络对算法进行了测试,对比算法是 DYN-MOGA,Kim-Han,IBEA.本实验所用的软硬件环境如下:Red Hat 64 位操作系统,16 核 CPU,主频 1.9 GHz,16 GB 内存,2 TB 硬盘;Eclipse 版本为 4.5.0,Java 版本为 1.8.0.

5.2 实验所用数据集

本文使用 Youtube,LiveJournal,DBLP,Flickr 这 4 个真实数据集和人工数据集进行实验[21-22].其中,Youtube 是用户到用户的链接关系网;DBLP 是作者合作关系网;LiveJournal 是在线社交关系网;Flickr 是一个分享网站的组员关系网.

人工数据集使用文献[21]中的算法生成.数据集 D_z($z=3,4,5,6$),其中 z 表示不同社区之间的边平均数.z 越大,社团间的连边越多,社团内的边越少,社团结构越不明显.数据集统计信息如表 1、表 2 所示.

Table 1 Real Datasets

表 1 真实数据集

Dataset	Number of Nodes	Number of Edges
Youtube	1 134 890	2 987 624
DBLP	317 080	1 049 866
LiveJournal	3 997 962	34 681 189
Flickr	2 302 925	33 140 017

Table 2 Synthetic Datasets

表 2 合成数据集

Dataset	Number of Nodes	Number of Edges
D_3	1 000 000	25 000 000
D_4	1 000 000	50 000 000
D_5	1 000 000	75 000 000
D_6	1 000 000	100 000 000

5.3 实验结果及分析

实验结果主要从 NMI 值、模块度、运行时间和收敛性 4 个方面验证算法的有效性.以下分别给出算法 DYN-MODPSO 在这 4 个指标下的度量结果.

5.3.1 NMI 值比较

由于人工网络的社区划分已经确定,所以本文选择第 2 节介绍的归一化互信息函数 NMI 作为指标,来评估这 3 个算法的社区划分结果和标准社区划分的相似性,从而检测算法的准确性.图 5 所示的是算法对人工网络 10 张快照检测结果的 NMI 值.

NMI 的值越接近 1,算法的检测结果越接近真实的社区划分.如图 5 所示,当 $z=3$ 时,DYN-MODPSO 的 NMI 值接近 1,而 DYN-MOGA 和 Kim-Han 的 NMI 值分别在 0.95 和 0.9 上下浮动,IBEA 的值初始接近 0.95,随着时间的推移,它的 NMI 值逐渐下降,在时间片 $T=10$ 时低于 0.9.当 $z=4,5$ 时,4 个算法的 NMI 值都下降,但是 DYN-MODPSO 的 NMI 值都稳定在 0.9,明显高于 DYN-MOGA,Kim-Han 和 IBEA.当 $z=6$ 时,DYN-MOGA,

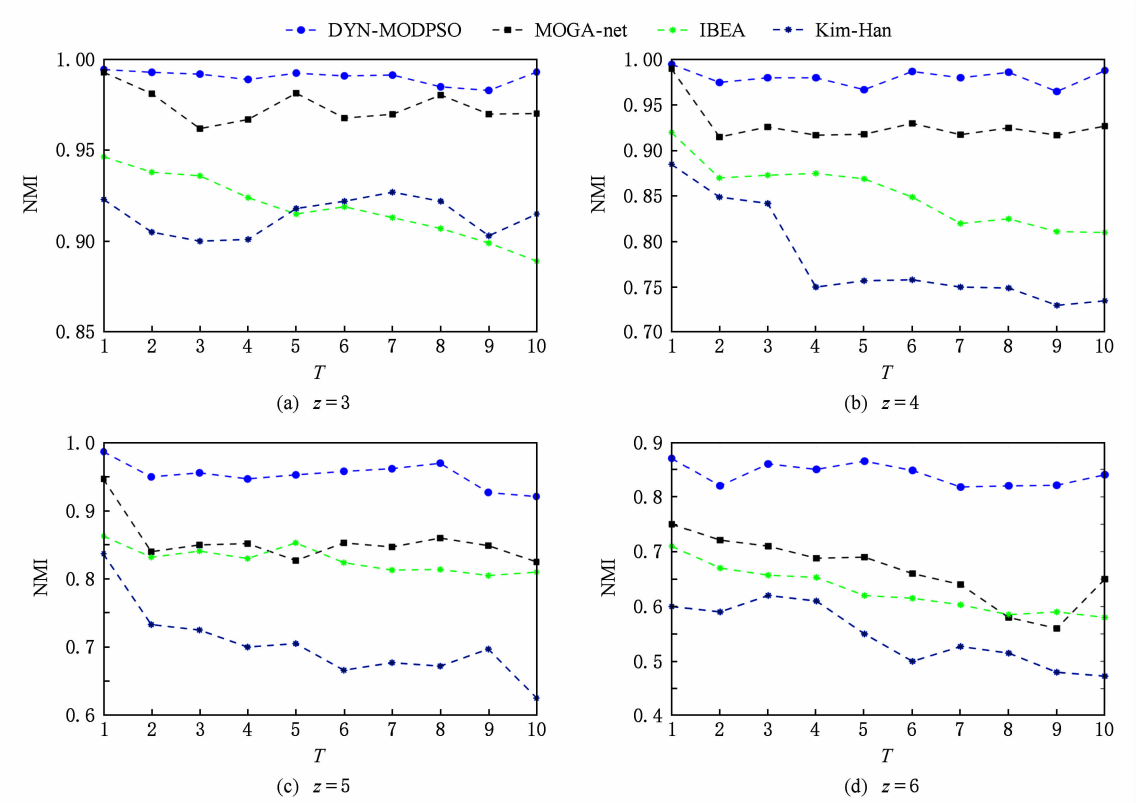


Fig. 5　NMI value of the synthetic dataset

图 5　人工数据集的 NMI 值

Kim-Han,IBEA 的 NMI 值已经接近 0.6,而 DYN-MODPSO 仍稳定在 0.85.

由此可以看出,当网络社区结构明显时,4 个算法都能检测到动态网络中准确的社区结构. 当动态网络社区结构变得模糊,4 个算法的社区检测能力均下降,但是 DYN-MODPSO 依然可以检测到相对准确的社区结构. 故算法 DYN-MODPSO 的检测能力都优于 DYN-MOGA,Kim-Han,IBEA,并且稳定性较强.

5.3.2　模块度比较

由于真实数据集的社区划分未知,所以用衡量社区划分质量的模块度函数来对实验结果进行评估. 模块度值越大,结果越接近真实的社区结构. 模块度函数记作 Q,定义为社区内实际的边数与随机连接情况下社区内期望的边数之差. 函数 Q 的计算公式如下:

$$Q=\frac{1}{2m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[A_{ij}-\frac{k_i k_j}{2m}\right]\delta(c_i,c_j). \qquad (7)$$

其中,$\boldsymbol{A}$ 是网络的邻接矩阵,m 是网络的总边数,k_i 表示节点 i 的度,c_i 表示节点 i 所在的社区标号. 如果 $i=j$,则 $\delta(i,j)=1$,否则 $\delta(i,j)=0$.

本文选择数据集的前 5 张快照,记录它们的模块度,如图 6 所示. 从图 6 中可以看出 DYN-MODPSO 的模块度大于 DYN-MOGA,Kim-Han,IBEA. 如图 6(a)(b),4 个算法的模块度都很高,DYN-MODPSO 稳定在 0.65 上下,随着数据集规模变大,如图 6(c)(d),4 个算法的模块度均减少,但是 DYN-MODPSO 的模块度值仍然在 0.53 上下. 所以,DYN-MODPSO 准确性很高,并且适合处理大数据网络图.

5.3.3　运行时间比较

图 7 所示的是算法的运行时间对比图. 算法在人工数据集上的平均运行时间如图 7(a)所示,可以看出 DYN-MODPSO 的运行时间小于 DYN-MOGA. 当 $z=3,4$ 时,算法 DYN-MODPSO 和 DYN-MOGA 的运行时间相差不大. 当 $z=5,6$ 时,网络中的社团结构变得模糊,此时 DYN-MOGA 的运行时间比 DYN-MOGA 将近缩短了一半.

算法在 4 个真实数据的平均运行时间如图 7(b)所示,DYN-MOGA 的运行时间比 DYN-MRDPSO 长,而且数据集越大,运行时间差越大,如图 7 所示 DYN-MOGA 在 LiveJournal 数据集的运行时间是

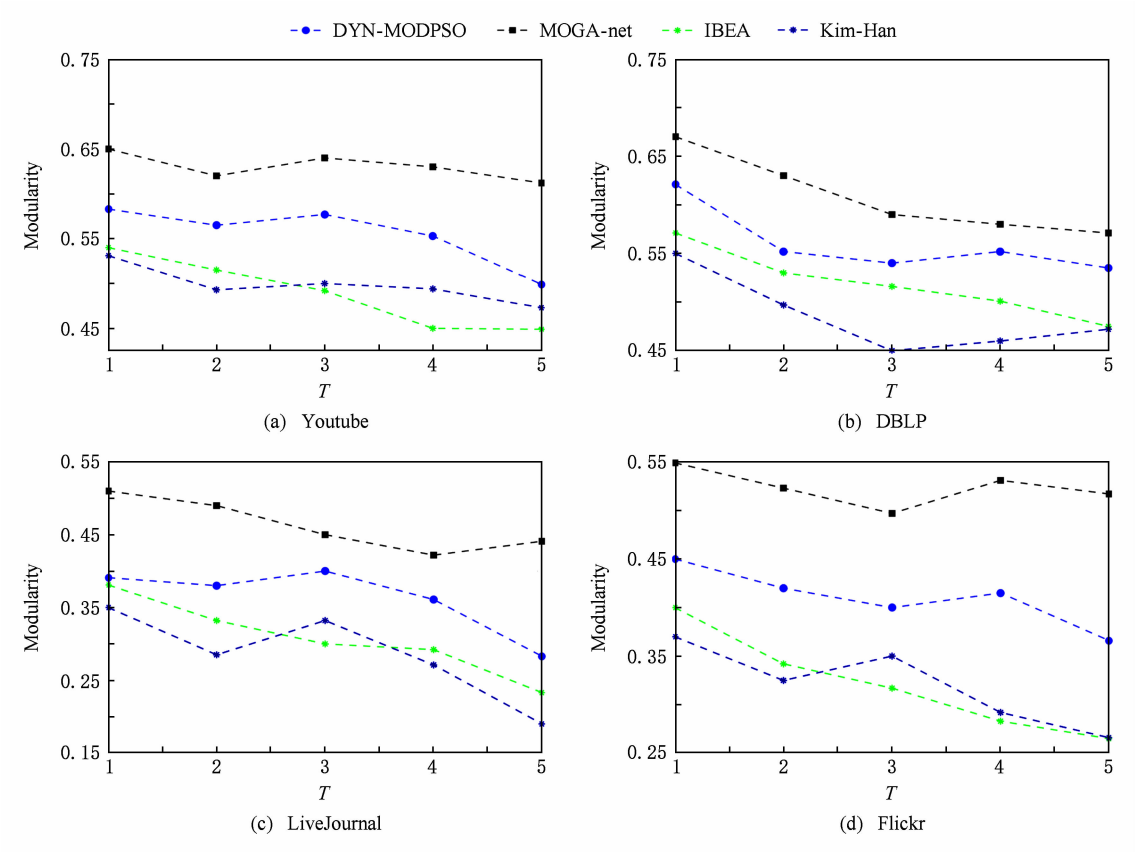


Fig. 6 The modular comparison of four real datasets

图 6 4 个真实数据集的模块度比较

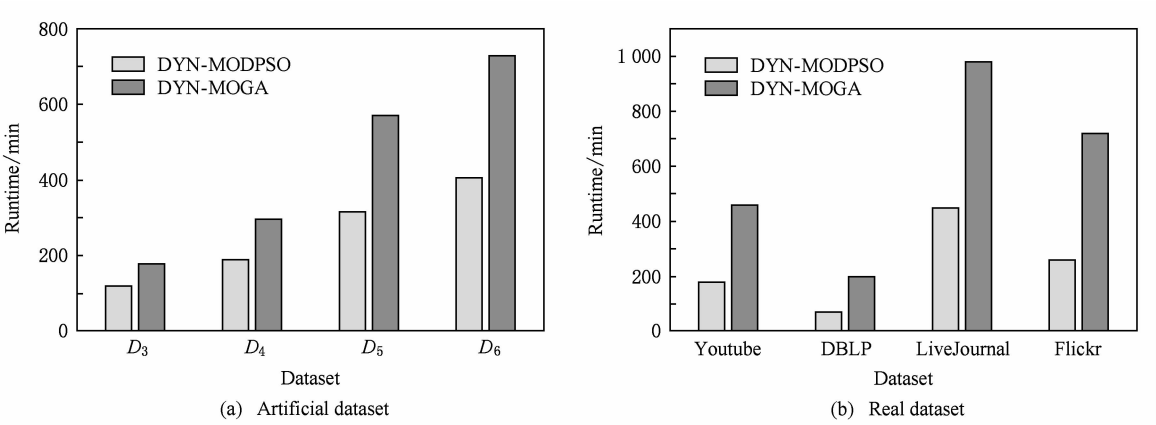


Fig. 7 Runtime comparison of the two algorithms

图 7 算法运行时间对比

DYN-MRDPSO 的 2 倍多. 所以 DYN-MRDPSO 是更高效的动态社区检测演化算法,更适合处理大规模数据.

5.3.4 收敛性比较

在社区检测方法中,DYN-MRDPSO 和 DYN-MOGA 都是演化算法. 收敛性是评估演化算法的一个指标. 在不断地迭代过程中,种群中不断优化最优解,当达到一定迭代次数时,最优解趋于稳定,不会随着迭代次数的增加而变化,这时算法收敛.

本文记录了 DYN-MRDPSO 和 DYN-MOGA 收敛时的最小迭代次数,如图 8 所示. DYN-MRDPSO 的迭代次数远小于 DYN-MOGA,说明 DYN-MRDPSO 具有更高的执行效率. 因为 DYN-MRDPSO 利用 DIGRW 来初始化种群,使种群在一定程度上接近最优解,而 DYN-MOGA 的初始种群是随机生成的,所以算法的迭代次数比 DYN-MRDPSO 多许多.

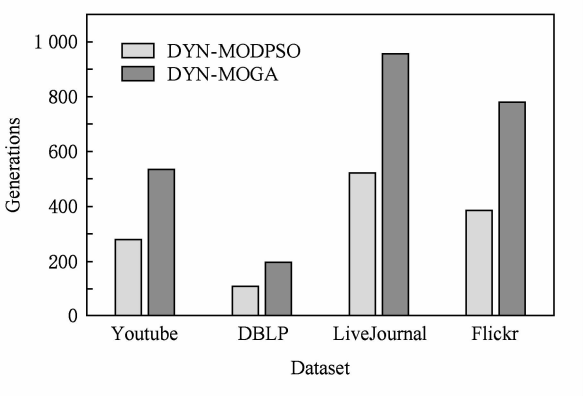


Fig. 8　Convergence comparison of algorithms

图 8　算法收敛性对比

6　总结与展望

本文提出一个参考最近未来的演化聚类框架,并将其引入粒子群社区检测方法中,提出了 DYN-MODPSO 算法.为了加快粒子群算法的收敛速度,本文对随机行走社区划分方法做了改进,提出了基于去冗余策略的随机行走初始个体生成算法,来初始化粒子群,使得粒子具有一定的精度和多样性.在粒子群的搜索过程中,本文引入多目标优化算法 NSGA-Ⅱ来同时优化 NMI 和 CS 这 2 个社区划分适应度函数,并加入干扰算子和多个体交叉算子来加强算法的局部搜索能力.

通过实验可以看出,在社区检测的性能方面,算法 DYN-MRDPSO 比 DYN-MOGA 和 Kim-Han 能检测到更准确的社区结构,是有效的.在社区检测的准确性方面,算法 DYN-MODPSO 的社区划分准确性高于 DYN-MOGA 和 Kim-Han,且具有较优的稳定性.故算法 DYN-MODPSO 是有应用意义的,可以用于动态网络社区检测.

参　考　文　献

[1] Cheng Xueqi, Jin Xiaolong, Wang Yuanzhuo, et al. Overview of big data systems and analytical techniques [J]. Jounal of Software, 2014, 3(9): 1889-1908 (in Chinese)
(程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述 [J]. 软件学报, 2014,3(9): 1889-1908)

[2] Fortunato S. Community detection in graphs [J]. Physics Reports, 2009, 486(3): 75-174

[3] Sarkar P, Moore A W. Dynamic social network analysis using latent space models [J]. ACM SIGKDD Explorations Newsletter, 2005, 7(2): 31-40

[4] Cordeiro M, Rui P S, Gama J. Dynamic community detection in evolving networks using locality modularity optimization [J]. Social Network Analysis & Mining, 2016, 6(1): 15-17

[5] Li Xiaoming, Wu Bin, Guo Qian, et al. Dynamic community detection algorithm based on incremental identification [C] // Proc of the 16th IEEE Int Conf Data Mining Workshop. Piscataway, NJ: IEEE, 2016: 900-907

[6] Lander B, Alejandro G. Multi-objective Graph Mining Algorithms for Detecting and Predicting Communities in Complex Dynamic Networks [M]. Raleigh: NCSU, 2017: 35-75

[7] Ma Xiaoke, Dong Di. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks [J]. IEEE Transactions on Knowledge & Data Engineering, 2017, 29(5): 1045-1058

[8] Stanovov V, Brester C, Kolehmainen M, et al. Why don't you use evolutionary algorithms in big data? [J]. Materials Science and Engineering, 2017, 173(1): 12-20

[9] Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering [C] //Proc of the 12th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2006: 554-560

[10] Kim M S, Han Jiawei. A particle-and-density based evolutionary clustering method for dynamic networks [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 622-633

[11] Folino F, Pizzuti C. An evolutionary multiobjective approach for community discovery in dynamic networks [J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1838-1852

[12] Hafez A I, Alshammari E T, Hassanien A E, et al. Genetic algorithms for multi-objective community detection in complex networks [C] //Proc of the 14th IEEE Int Conf on Intelligent Systems Design and Applications. New York: IEEE, 2014: 145-171

[13] Gong Maoguo, Cai Qing, Chen Xiaowei, et al. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 82-97

[14] Karimi-Majd A M, Fathian M, Amiri B. A hybrid artificial immune network for detecting communities in complex networks [J]. Computing, 2015, 97(5): 483-507

[15] Knowles J D, Corne D W. Approximating the nondominated front using the pareto archived evolution strategy [J]. Evolutionary Computation, 2014, 8(2): 149-172

[16] He Dongxiao, Zhou Xu, Wang Zuo, et al. Complex network community mining-genetic algorithm based on clustering fusion [J]. Acta Automatica Sinica, 2010, 36(8): 1160-1170 (in Chinese)
(何东晓, 周栩, 王佐, 等. 复杂网络社区挖掘—基于聚类融合的遗传算法[J]. 自动化学报, 2010, 36(8): 1160-1170)

[17] Xu Zhengguo, Zheng Hui, He Liang, et al. Self-adaptive clustering based on local density by descending search [J]. Journal of Computer Research and Development, 2016, 53(8): 1719-1728 (in Chinese)
(徐正国, 郑辉, 贺亮, 等. 基于局部密度下降搜索的自适应聚类方法[J]. 计算机研究与发展, 2016, 53(8): 1719-1728)

[18] Ahmed M M, Hafez A I, Elwakil M M, et al. A multi-objective genetic algorithm for community detection in multidimensional social network [C] //Proc of the 1st Int Conf on Advanced Intelligent System and Informatics. Berlin: Springer, 2016: 129-139

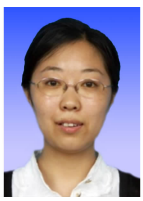
[19] Li Yangyang, Wang Yang, Chen Jing, et al. Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization [J]. Journal of Heuristics, 2015, 21(4): 549-575

[20] Chen Weineng, Yang Qiang. Probability distribution based evoluionary computation algorithms for multimodal optimization [J]. Journal of Computer Research and Development, 2017, 54(6): 1185-1197 (in Chinese)
(陈伟能, 杨强. 基于概率分布的多峰演化算法[J]. 计算机研究与发展, 2017, 54(6): 1185-1197)

[21] Newman M E, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 2004, 69(2): 26-53

[22] Institute of Web Science and Technologies at the University of Koblenz-Landau. Datasets [EB/OL]. (2017-04-27) [2017-07-21]. http://konect.uni-koblenz.de/networks/

Li He, born in 1991. Received her MEn degree in computer science from Northeastern University, China, in 2018. Her main research interests include big data mining and community detection.



Yin Ying, born in 1980. Received her BEn, MEn and PhD degrees in computer science from Northeastern University, China, in 2002, 2005 and 2008, respectively. Currently associate professor in the College of Computer Science and Engineering, Northeastern University, China. Member of IEEE, ACM and CCF. Her main research interests include data mining and machine learning. (yinying@cse.neu.edu.cn)

Li Yuan, born in 1986. Received his BEn and MEn degrees in computer science from Northeastern University (NEU), China, in 2009 and 2011, respectively. Currently PhD candidate in computer science, NEU. His main research interests include data mining and bioinformatics. (li888yuan@163.com)

Zhao Yuhai, born in 1975. Received his BEn, MEn and PhD degrees in computer science from Northeastern University (NEU), China, in 1999, 2004 and 2007, respectively. Currently associate professor in the College of Information Science and Engineering, NEU. Member of IEEE, ACM and CCF. His main research interests include data mining and bioinformatics. (zhaoyuhai@mail.neu.edu.cn)

Wang Guoren, born in 1966. Received his BSc, MSc and PhD degrees in computer science from Northeastern University (NEU), China in 1988, 1991 and 1996, respectively. Currently professor in the College of Computer Science and Engineering, NEU. Senior member of CCF. His main research interests include XML data management, query processing, optimization, high-dimensional indexing, parallel database systems, P2P data management and uncertain data management. (wanggr@mail.neu.edu.cn)