基于迁移学习和多目标进化算法的动态网络社区检测

邓高山

指导教师　林凡

厦门大学

厦門大學

本 科 毕 业 论 文 （设 计）

（软件工程）

基于迁移学习和多目标进化算法的动态网络社区检测

Transfer Learning Based Multiobjective Evolutionary Algorithm for Community Detection in Dynamic Networks

姓　　名：邓高山

学　　号：24320132202395

学　　院：软件学院

专　　业：软件工程

年　　级：2013 级

指导教师：林　凡　（副教授）

曾湘祥　（副教授）

二〇一七　年　月　日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年　　月　　日

致 谢

又到了一年凤凰花开的时节，大一入学时候稚嫩的面孔仿佛昨日，转眼却已到了告别时。大学四年，我从一个懵懂无知的青涩少年，成长为一个能够独当一面，有理想、有担当的成熟的成年人，这离不开我所处的环境的培养。

首先我要感谢党和国家，感谢你们对大学生发展的重视。大学开始逐渐接触到社会以后，深感一个和谐、稳定的国家，对于每个国人来说是多么重要。我对身在这样一个文明、民主的社会主义国家而感到骄傲与自豪。

我想对我的敬爱的母校-厦门大学表达我最真挚的祝福。她有着美丽的校园环境，认真踏实的学术氛围，使得我们能够蓬勃发展，学习专业知识，培养成人。提供给我们奖学金、补助，给我们平时的生活减少了很多负担。大力支持社团活动，使得我们能够早日与社会接轨，提高我们的人际交往能力。

我要感恩我的老师，感谢林凡老师在论文选题、修改上的大力指导，对我们严格要求，特意培养我们的独立科研的能力。感谢曾湘祥老师，对我来说，他既是一名长者，也像一位朋友，是他带领我进入科研领域，指导我进行实验并发表论文、参加国际会议。感谢辅导员李绍玉老师、刘锦锗老师，能够在我的学业生活中遇到任何困难及时给予帮助，热心解答我的各种问题。我的成长离不开各位老师的悉心关怀与认真教导，我会永远铭记你们的这份恩情。

我要感谢我的父母，是他们在经济、生活等各方面不遗余力地支持着我，并且始终维持着家庭的和睦，使得我能够在学校心无旁骛地学习知识、提高自我。同时，在我心情烦闷的时刻，能够及时开导我，尊重我的人生选择；感谢爷爷奶奶陪伴我度过了快乐的童年，感谢表姐给我分享她的人生经验帮我少走弯路。

感谢我的朋友，他们为我的大学生活增添了色彩。感谢我的舍友，在大学四年里能够和谐共处，没有矛盾，一起愉快地玩耍，度过了美好时光。感谢我的队友，跟我一起参加比赛，挑战自我，让我对技术更加感兴趣，也增加自己的信心。

大学四年时光如白驹过隙，太多的回忆涌上心头。往事已成追忆，未来的路还很漫长。相遇是缘，相离亦是缘，值此告别之际，唯有芙蓉隧道洞口的一句“我爱你，再见！”，方能表达自己对母校最真挚的感情。

摘　要

社区检测是近年来社交网络研究的热点和基础性问题。实际中的很多网络是动态变化的，因此研究动态网络中的社区检测具有重要的实际意义。在动态网络聚类问题中，我们面临着更大的挑战，既要最大化聚类的准确性，同时在网络结构变动后，也要最小化连续两次聚类结果之间的差异，这两个目标存在相互制约关系。本文针对动态社区检测问题的特性，将迁移学习中的特征提取思想与传统多目标进化算法框架结合，提出一种基于特征迁移的多目标进化算法（Feature Transfer based Multi-objective Optimization Genetic Algorithm， TMOGA）。其重要思想是，从过去已有的社区结构中提取特征，将有价值的特征信息保留下来，并且将这些特征信息迁移到当前以及未来的优化过程中，从而改进原有的进化算法。通过分别在小规模、大规模经典测试集以及真实问题上进行实验验证，最终结果表明，相比于目前最优秀的动态网络社区检测算法，我们的算法能够取得更好的聚类效果。

关键词：动态网络；迁移学习；社区检测；多目标进化算法

Abstract

Community detection is the hotspot and basic problem of social network research in recent years. Many of the real world networks are dynamic, so it is significative to study community detection in dynamic networks. In the dynamic network community detection, we face a greater challenge, it is necessary to maximize the accuracy of clustering while the network structure changes, but also to minimize the results of the two consecutive clustering differences between the two results. There is a trade-off relationship between these two objectives. In this paper, based on the characteristics of dynamic community detection problem, we proposes a feature transfer based Multi-objective optimization algorithm (TMOGA) based on trasfer learning and traditional multi-objective evolutionary algorithm framework. The main idea is to extract features from past community structures, retain valuable feature information, and migrate these feature information into current and future optimization processes to improve the original evolutionary algorithms. The results show that our algorithm can achieve better clustering effect compared with the best dynamic network community detection algorithm in the small scale, large scale classical benchmarks and real world problem.

Key words: Dynamic Networks; Transfer Learning; Community Detection; Multi-Objective Optimization

目 录

Content

第一章 绪论

1.1 研究背景与意义

网络结构是对于一个系统的抽象描述。现实中很多系统结构都可以用网络结构进行描述，例如社会关系网、科学家合作网、通信网、互联网、社交网络[1]等等。网络的社区检测算法是将一个网络划分成若干个紧密相连的小的团体，这样的团体又称作社区（Community）。同一种社区内部节点之间的关联紧密，而不同社区之间相互关联相对较稀疏。社区本质上是整个网络的一个强关联子集。

一个网络的图中各节点之间的相邻点个数、关联强度等信息属于网络结构的关键信息。而通过网络的社区结构恰好能够直观有效地反映出这些量化信息。举一个实际生活中的例子：新浪微博的用户数据。两个微博用户之间的互动得越频繁，他们的关联就越密切。通过社区检测算法，可以在整个新浪微博的人际关系网络中寻找小团体，可以针对这样的具有共同特征的团体进行更加精准地营销与推广，从而获得更好的经济效益。另外一个实际问题是学术论文的引用关系，我们可以通过文献作者之间的互相引用关系，确定学术研究领域的“小圈子”，从而了解学术界的哪些作者存在合作关系，并且掌握目前学术界的热门领域。

很多实际的网络的结构往往是动态变化的。比如在社交网络中，人与人之间的关系通常随时间发生改变，对应的网络结构也随之改变。在这种情景之下，动态网络模型[2]能够更准确地反映这种变化的网络结构。在动态网络模型的社区检测问题中，我们不仅仅要考虑静态网络模型中常考虑的聚类结果准确度，同时我们也要考虑第二个目标：最小化两个连续时间节点之间的聚类结果差异性。使得两次时间点之间的聚类结果尽量保持一致。第一个目标的现实意义很明确，只有保证准确度，才能够保证所求的结果与真实结果接近;第二个目标的意义在于在社区网络结构发生改变之后，使得变化后的聚类结果尽可能与之前相似，从而能够降低社区变化的成本。

静态网络社区检测问题，在文献[3]中已经被证明是一个N-P难的问题。动态网络社区检测问题又可以看作是一组静态网络问题的集合，因此算法的应用将变

得更加困难。多目标进化算法作为一种优秀的启发式算法，已经被广泛应用到了网络社区检测问题中，并且取得了相当良好的表现，关于这方面的相关研究在后续的章节有详细介绍。

我们认为动态网络模型不是毫无规律地随机变动，而是在变化过程中，部分社区仍然保留下了之前时刻的一些不变的特征。如果能够分辨并且提取这些特征，就能够为将来的优化过程提供帮助信息。我们受到迁移学习[4]中特征迁移的启发，将特征迁移的思想首次引入到解决动态网络问题中。对此，我们建立了一套从动态社区网络结构中提取特征，并且应用特征的机制。我们将这种特征迁移机制与经典的多目标进化算法NSGA-II[5]相结合，提出了一种基于特征迁移的多目标进化算法（Feature Transfer based Multi-objective Optimization Genetic Algorithm，TMOGA），大大提高了原本的NSGA-II算法在解决动态网络问题上的收敛速度与准确度。

本文的主要贡献如下：

1. 提出了基于迁移学习与多目标进化算法的动态网络社区检测框架，同时解决了将迁移学习结合到动态社区检测问题所面临的几个问题，包括特征定义，特征提取，特征筛选，特征迁移。

2. 提出与特征迁移机制配套的算法实现，包括：基于团的特征提取算法，基于启发式评估的特征筛选算法，将特征迁移到种群的算法。这三种算法结合起来，组成了一个完整的特征迁移机制，能够有效地将迁移学习的思想引入到解决动态社区检测问题中。

3. 对 TMOGA 算法的效果进行实验验证。实验结果表明，无论是在随机产生的测试集、经典测试集或者是实际应用问题上，我们的算法相较于目前前沿的其他同类算法都就有一定优势。同时，实验也探究了各种参数设定对于算法最终结果的影响。

1.2 论文组织结构

本章节主要指出对于动态网络聚类问题的研究背景与现实意义，并且陈述了我们做出的主要贡献。第二章节主要是对动态社区检测问题相关的知识进行介绍，

包括该问题的定义，同时给出了相关的研究工作。第三章详细介绍 TMOGA 算法的实现过程，并且给出了算法的具体实现伪代码。第四章是实验部分，包括介绍对于实验结果的评价指标，给出实验数据以及最后的实验结论。最后一章总结，同时提出未来可行的研究方向。

第二章 相关知识介绍

2.1 多目标优化

2.1.1 多目标优化问题

现实生活中的很多优化问题并不仅仅是单一目标，通常存在多个目标要求同时优化，而这些目标通常存在相互制约关系，因此求解过程也相对困难。多目标优化问题（Multi-Objective Optimization Problem， MOP）用可以表示成为如下的数学形式：

$$\max F(x) = \left(f_1(x)，f_1(x)，\ldots，f_1(x)\right)^T \tag{2-1}$$

$$\text{subject to} \quad g_i(x) = 0，i = 1，2，\ldots l$$

$$h_j(x) \geq 0，j = 1，2，\ldots n$$

其中，$g_i(x)$定义了 c 个约束式，F：$\Omega \to R^m$包含了 m 个目标函数，而R^m被称为目标空间。既找到一组满足约数式的解x，使得每个目标都求得最大或最小值。

2.1.2 帕累托最优

多目标问题所求得的解，往往是非单一解，因为单一解通常不可能在每个目标上都同时达到最优。多目标问题的解通常被描述成为一组相互之间满足帕累托支配关系的解的集合。帕累托最优关系[6]最初来自于经济学领域，后来被逐渐引入到了多目标优化领域中，成为多目标优化问题的一个最基本的概念。帕累托占优的定义是指，设$u，v \in R^m$，u 支配 v 当且仅当所有i $\in \{1，2，\ldots，m\}$，$u_i \geq v_i$同时至少存在j $\in \{1，2，\ldots，m\}$满足$u_i \geq v_i$，那么我们可以称解 u 支配解 v，记做$u \succcurlyeq v$。而当一个解集 P 中的任何一个解都不被其他解支配，我们称这个解集为帕累托前沿（Pareto Front），而这个帕累托前沿通常作为一个多目标优化问题的最终解。

2.1.3 多目标优化算法

传统的针对多目标优化问题的算法有基于近似估计法[7]，数学规划方法[8]，在还有经典的加权和方法[6]，边界交叉点方法[6]。而随着对多目标问题的进一步研究，多目标进化算法被逐渐引入到解决多目标问题，并且取得了良好的效果。

进化算法借鉴了自然界优胜劣汰的思想，将自然选择、染色体的自我复制、交叉互换、变异等自然现象应用到解决问题中。进化算法的特征是全局收敛速度快，对问题的解的具体分布不敏感，具有一定的随机性。作为一种元启发式优化算法，进化算法在很多优化问题上有着广泛的应用。尤其是针对大规模优化问题，当传统的优化手段难以在一定时间内获得足够优秀的解时，进化算法常常能够在这些问题上表现出色。

2.1.4 快速非支配排序（NSGA-II）算法

多目标优化领域最著名的算法是 2002 年由 Deb 提出的 NSGA-II[5]算法，该算法改进自 Deb 本人于 1994 年提出的非支配排序 NSGA[9]算法。NSGA-II 基于整个种群的支配关系提出了著名的快速非支配排序算法，大大降低了 NSGA 原本的时间复杂度。同时，提出了基于拥挤距离的种群多样性保持策略。NSGA-II 在提出以后的十多年来效果出色，作为多目标进化算法领域最经典的算法之一，一直被学者广泛地研究，目前的学术引用量已经超过两万次。

2.2 问题与目标函数定义

2.2.1 动态社区检测问题的数学描述

动态社区检测问题的目标，是在一个动态变化的社区网络结构中，找到每个时刻所对应的社区结构。定义 $\{1，2，\ldots，T\}$ 表示时间节点的集合，$V=\{1，2，\ldots，n\}$ 为一个静态网络结构的点的集合。一个动态网络问题是一组序列，定义为$N=\{N^1，N^2，\ldots，N^T\}$，其中 N^t 表示在时刻 t 的网络结构，可以用图 $G^t=(V^t，E^t)$ 来表示，其中 V^t 表示在 t 时刻结点的集合，而 E^t 表示在时间点 t 连接 V^t 中的两个结点的边的集合，如果在时间 t，点u与点v存在关联的边，则 $(u^t，v^t)\in E^t$。

一个静态网络$\mathrm{N^t}$中的社区结构，可以描述为一组结点 $V_i^t \in V^t$，且 V_i^t 内部有很高的边的密度。定义 C^t 为一个子图，代表着一个社区结构，或者称一个聚类。定义 $CR^t = \{C_1^t, C_2^t, \ldots, C_k^t\}$ 为 G^t 的一种分割方案，且保证 $\mathrm{C_1^t \cup C_2^t \ldots \cup C_k^t = V^t}$，同时$C_1^t \cap C_2^t \ldots \cap C_k^t = \emptyset$ 。

下图表示一个动态网络社区检测问题的简单模型，从图中可知 $T = 2$，$N = \{N^1, N^2\}$，$V^1 = V^2 = \{1, 2, 3, 4, 5, 6, 7\}$。$CR^1 = \{C_1^1, C_2^1\}$，$CR^2 = \{C_1^2, C_2^2\}$。其中$C_1^1 = \{1, 2, 3, 4\}$，$C_2^1 = \{5, 6, 7\}$，$C_1^2 = \{1, 2, 3\}$，$C_2^2 = \{4, 5, 6, 7\}$ 。

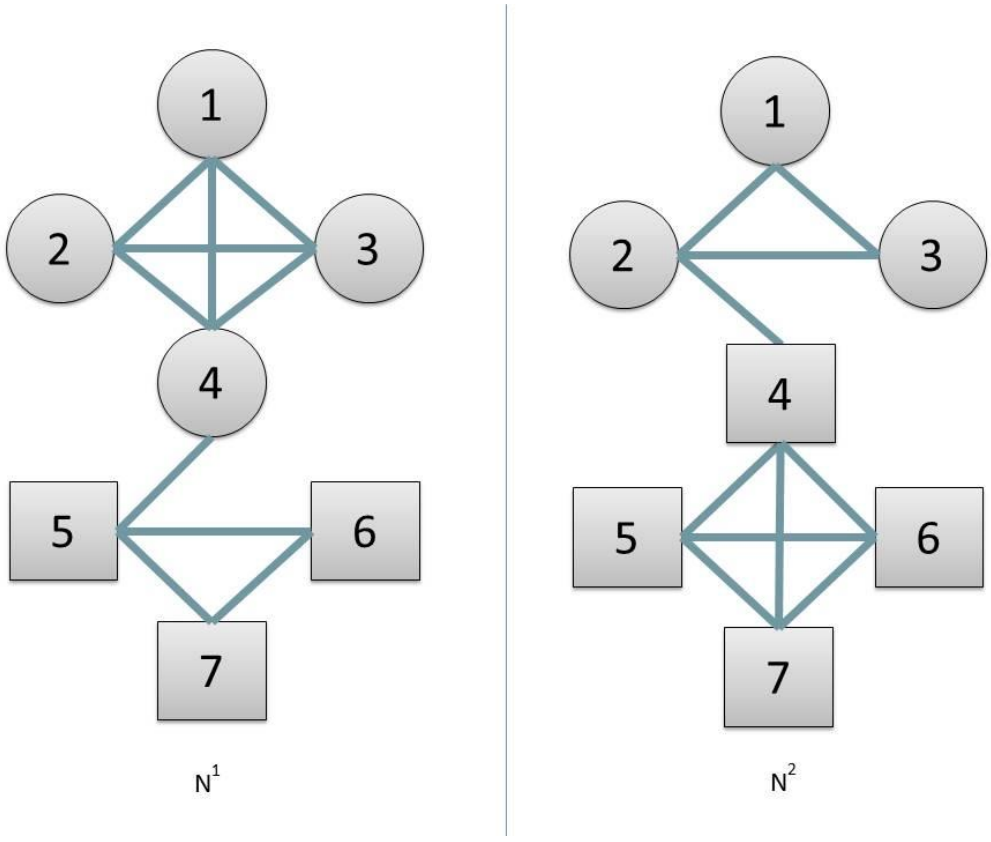


图 2- 1 一个简单的动态网络模型

2.2.2 聚类准确度指标：模块度

社区检测问题也可以被定义成为一个聚类问题，Girvan 与 Newman 在[10]提出了一个衡量最终的聚类结果好坏的指标：模块度（Modularity，也称作 Q 值）。模块度的含义是落在同一组内的边的比例减去对这些边进行随机分配所得到的

概率期望值，Q 值的数学定义为：

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

其中，m 是边的数量；A_{ij}是图的邻接矩阵中 i 行 j 列的值，表示连接 i 与 j 节点的边的权值；k_i是点 i 的度；$\delta(i,\ j)$ 表示如果点 i 与点 j 在同一个社区中，则$\delta(i,\ j) = 1$，否则$\delta(i,\ j) = 0$。

模块度值的大小主要取决于所求得的聚类结果的整体社区特性的强弱，因此可以用来定量地衡量网络社区划分的质量。模块度的取值范围是 [−1/2，1)，其值越接近 1，表示网络划分出的社区结构的强度越强，也就是划分质量越好。论文[10]中提到，当 Q 值在 0.3~0.7 之间时，说明聚类的效果已经很好。

2.2.3 内聚程度指标：核心连接均值

核心连接均值（kernel k-means, KKM）仅仅关注聚类内部的连接强度。它的定义如下：

$$KKM = 2(n - k) - \sum_{i=1}^{k}\frac{L(V_i, V_i)}{|V_i|} \tag{2-3}$$

其中$L(V_i, V_j) = \sum_{i\in V_i, j\in V_j} A_{ij}$ ，A 是邻接矩阵。V_i代表社区 i，$|V_i|$表示社区中的节点数量，KKM 反映了同一个社区之间的连接强度总和。

2.2.4 相似度度量指标

除了社区性质强度，还需要一个衡量相邻时间之间社区分类相似程度的指标。我们采用了归一化互信息指标（Normalized Mutual Information，NMI）[10]，NMI 的数学定义如下：

$$NMI(A,\ B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{C_A} C_{i.}\log(C_{i.}/N) + \sum_{j=1}^{C_B} C_{.j}\log(C_{.j}/N)} \tag{2-4}$$

其中，$C_A(C_B)$是指在 A 中的聚类的个数，$C_{i.}(C_{j.})$是C中第 i 行（或者第 j 行）的和，N 是点的个数。NMI 的取值范围为[0，1]，当$A = B$时，$NMI(A,\ B) = 1$；当 A，B 完全不同时，$NMI(A,\ B) = 0$。在动态问题中，我们的另一个目标是使

得两次相邻时间间隔中的聚类结果尽可能相似，即使NMI（CR^{t-1}，CR^{t}）的值最大。

2.2.5 社区检测问题定义

给定一个动态的网络结构，在 t 时刻的状态为N^t，我们的问题可以定义为，找出一个最优的划分方案CR^t，使得：

$$for\ each\ t: maximize \begin{cases} Q(\mathrm{CR^t}) \\ NMI(\mathrm{CR^t}, \mathrm{CR^{t-1}}) \end{cases} \quad (2\text{-}5)$$

因为进化算法通常优化最小值，因此为了方便进行实验，将两个目标都调整为求最小值的问题，最后所得的目标值计算公式为：

$$for\ each\ t: minimize \begin{cases} -Q(\mathrm{CR^t}) \\ 1 - NMI(\mathrm{CR^t}, \mathrm{CR^{t-1}}) \end{cases} \quad (2\text{-}6)$$

2.3 相关研究

近几年来，多目标优化算法被广泛应用到了优化领域中。多目标优化算法主要分成三大类：第一类是基于支配关系的算法，其中最著名的是 Deb 提出的NSGA-II[5]算法，同类的算法还有 SPEA2[11]，PAES2[12]。第二类是基于评价指标的算法，包括 HypE[13]；第三类是基于分解的方法，其中最著名的是张青富提出的MOEA/D[14]算法。2014 年，Deb 又提出了针对多维问题的NSGA-III[15]算法，将分解的思想跟参考点等策略加入到了 NSGA-II 中的一次改进。

这些多目标进化的算法很快就应用到了实际领域中，文献[16]将 NSGA-II 算法应用到了赛车设计问题中。这个问题总共有 24 个目标，其中 18 个是互相冲突的目标。文献[17]中将 MOEA/D 算法应用到解决分布式调度系统中，其中两个目标是系统运行时间与系统耗能总量。在各种优化领域中，多目标进化算法体现出了巨大的实用价值。

对于静态网络聚类问题也有很多研究，[18]提出的标签传播算法（LPA），采用随机标签传播策略快速求得聚类结果。2009 年提出的 LFM 算法[19]，是基于启发式的算法。

近年来，多目标优化算法被逐渐引入到静态社区检测问题中。在一些经典测

试集上，比如 LFR 问题[20]。相对于原来的经典方法，多目标优化算法在短时间内就能取得非常准确甚至完全一致的最终结果。2007 年，文献[21]将单目标进化算法首次引入到了社区检测问题中。随后，很多经典的多目标优化算法也逐渐被引入到解决静态社区检测问题中，比如 MOEA/D-net[22]算法，是基于 MOEA/D 算法解决动态问题。公茂果老师提出的基于分解的粒子群算法 DPSO[23]，将分解的思想同粒子群算法相结合，同时提出了基于标签传播的初始化种群方法，DPSO 在静态网络聚类问题取得了非常好的效果。

对动态社区检测问题的研究是一个新兴的研究方向，主要研究集中在中小规模、大规模的测试集，对于超大规模的问题的研究相对较少。目前的研究成果有，Kim 等人在 2009 年提出的 Kim-han 算法[24]，2014 年提出的 DYNMOEA[2]，将 NSGA-II 用来解决动态问题。2016 年，马晓科提出的 sE-NMF[25]，基于光谱聚类方法解决动态问题。

迁移学习近年来取得了突破性进展。迁移学习作为一种新的机器学习方法，其核心思想在于运用已存有的知识，对不同但相关领域问题进行求解。在迁移学习中，训练样本并不需要满足与新的测试样本独立同分布，同时可以利用少量的现有样本甚至不需要新的训练样本就能够解决新问题。迁移学习研究的代表人物是香港科技大学的杨强教授，其代表作有[4]。根据文献[26]，目前主流的迁移学习主要分成四类：样本迁移，特征迁移，基于模型的迁移学习，通过关系进行迁移。关于这几类方向，在研究上都取得了非常显著的效果。也有越来越多应用问题采用了迁移学习的算法。但是在动态社区网络聚类检测问题上，目前并没有相关的研究。因此，我们尝试将迁移学习中的特征迁移思想首次引入到动态问题中。

第三章 基于特征迁移的多目标进化算法

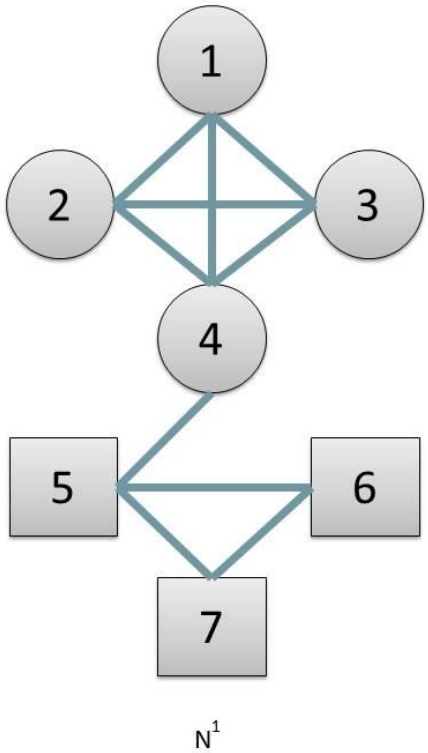
本章主要是介绍 TMOGA 算法的具体实现细节。首先介绍相关的基本知识，包括如何编码来表示一个动态聚类问题，以及对多目标进化算法中的一些基本算子进行自适应调整，包括种群产生，选择，交叉互换，变异，以及基于拥挤距离的快速非支配排序筛选。然后介绍了我们算法是如何进行特征迁移，详细介绍了特征提取，特征筛选，特征迁移三种算法。最后列出整个算法的实现伪代码。

3.1 算法基础部分

3.1.1 问题编码

进化算法中的编码是指将一个实际的问题表示成计算机可以理解的形式。编码是进化算法运行的基础。目前，对于社区聚类问题，已经提出了很多种不同的编码方案。而这些编码方案，可以大致分成两大类：第一类是直接编码，直接编码又叫做完全编码，通常直接用标签数字来代表每个点所属的社团编号，所有的节点组成一个标签序列。直接编码的优点是简单直白，并且不需要解码的过程，因此时间复杂度低。然而，直接编码的解在某些操作下，往往会损失一部分信息；第二类是间接编码，通常指需要进行解码操作才能将编码转换成对应问题的解的编码方案，比如基于核心的邻接编码[27]。由于这类编码往往能够更好地保留解整体结构特征不被轻易破坏，因此我们也采用了基于核心的邻接编码作为进化算法的编码方式。

下图说明了直接编码与间接编码的区别：



Direct representation :

1	2	3	4	5	6	7
1	1	1	1	2	2	2

Locus-based representation :

1	2	3	4	5	6	7
4	3	2	2	6	5	6

图 3- 1 直接编码与间接编码

如图 3-1 左表示N^1表示在 t =1 时刻的社区结构。对于直接编码方案，图 3-1 右上所示，可以通过节点的标签来直接区分不同的组，同在一个组的点标签相同，为该组的组号。直接编码的缺点在于，实际交叉过程中，即使父母都将某些节点同时分到一组，但是父母基因中对于原本属于同组的节点的标签往往不同，因此导致原本在同一组的节点在交叉操作之后，产生的后代由于父母标签不同，被误分在不同组。

基于核心的间接编码方案，如图 3-1 右下所示，该种编码方式不直接用标签来代表组，而是记录同在一个组的另一个点的索引位置来表示这两个点同在一个组。如图所示，按照基于核心的间接编码方式，图中表示 1 与 4 在同一组中，2 与 3 在同一组，以此类推。这样编码的好处在于交叉过程中不会破坏整体的组结构，在实验中也证明了这种编码方式对于复杂的网络聚类问题比一般的直接编码方式效果更好。

3.1.2 解码

对于间接编码，计算解的目标值之前需要进行解码操作。解码算法的流程是

先将每个结点单独分成组，然后逐渐遍历基因找到同属于同一组的节点，寻找到他们各自所属的组，并且将这些组归并成为一组，直到整个基因都被解码完毕，就可以还原出所有的社团结构。

3.1.3 种群的初始化

进化算法中的初始解是通过随机产生的，每个解的标签都有可能是自己，或者是等于它的某个邻居节点的标签。

3.1.4 交叉

在自然界中染色体交叉互换是亲代基因型产生子代基因型的主要方式，产生的子代往往会同时继承双亲的一些特性。受到这一生物进化自然现象的启发，交叉操作也成为了进化算法产生新解的一个最主要的算子。在进化算法的过程中，亲本的交叉的方式决定了搜索方向。关于遗传算法的交叉方法的研究早已成熟，目前主要包括单点交叉，两点交叉，均匀交叉，多亲本交叉。由于两点交叉能够最大程度保存亲本的点之间的整体特性不被破坏，因此这种交叉方式被广泛用在社团结构检测问题当中，我们的算法也采用了两点交叉的方式进行。两点交叉的主要方法是以其中一个亲本作为原型，在该亲本的基因型的基础上，随机选取两个交叉点，然后将这两点之间所截取的基因与另一亲本在这一段的基因进行整体的互换操作，从而产生一个新的子代。一般来说两个亲本能够分别作为原型，从而产生两个子代，两点交叉的过程如图 3-2 所示：

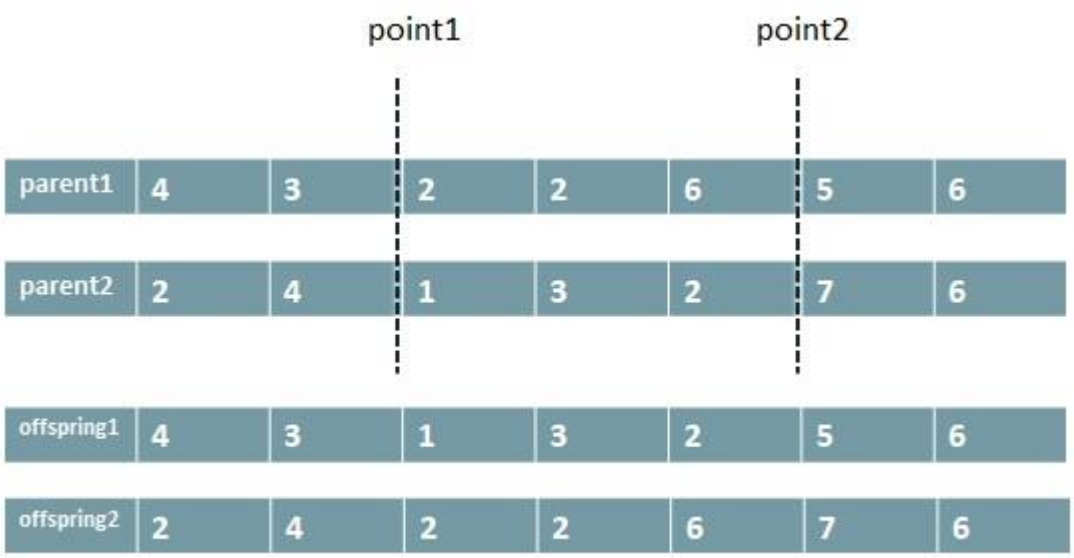


图 3- 2 交叉操作

3.1.5 变异

变异操作，在生物学中也被称为基因突变，虽然部分的变异在生物界中可能对后代产生有害的效果，然而变异仍然是作为一个物种进化过程中的一个关键性的步骤。在进化算法中，变异操作同样是保持解的多样性的关键部分，也是整个种群产生新解的重要方式。在产生一个子代之后，有一定概率发生变异。

我们采用的变异的方式是单点变异。单点变异的过程是，在产生了子代之后，对后代解中的每个节点，都有一定概率随机将该点的标签改变成为它的某一个邻居解的标签。

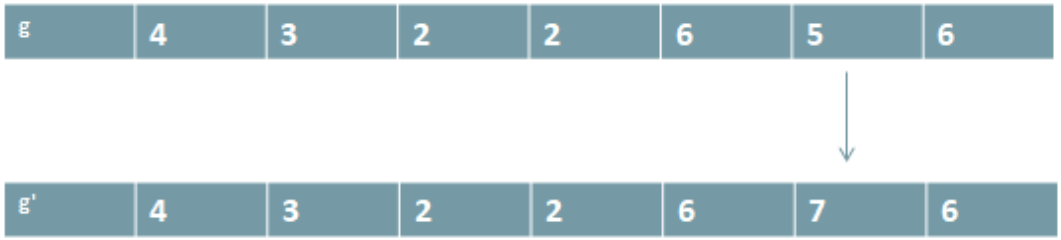


图 3- 3 单点随机变异

3.1.6 快速非支配排序

在种群产生子代之后，为了维持整个种群的规模不变，需要淘汰掉部分劣势的解。我们采用了 NSGA-II 中的基于拥挤距离的快速非支配排序来实现整个自

然选择过程。快速非支配排序的算法过程为，首先找出所有种群中没有被支配的所有解，从种群中移除掉这些解，并且加入到新种群中，这些解被称为非支配层级中的第一层。然后重复这个过程获得下一层，直到需要加入的解的数量大于整个种群的容量，然后对最后一层所得到的解应用基于拥挤距离的选择策略。

对于最后一层，引入拥挤距离是为了保证整个种群的多样性，使得种群尽量能够在解空间均匀分布。在二维的情况下，拥挤距离被定义为距离最近的附近的点之间的曼哈顿距离。解的周围其他解分布得越密集，拥挤距离就越小。而算法最终通过挑选一组解使得整体的拥挤距离最大，即保证解之间的多样性最大。

3.2 特征迁移机制在动态网络中的实现

TMOGA 算法的核心思想是将过去已知的解中的特征迁移到当前问题中。在迁移过程中，主要面临三个方面的问题：第一，如何定义特征并且从过去的解中提取特征；第二，如何筛选特征，即如何寻找出在当前问题仍然有效的特征；第三，在当前时刻如何利用从过去时刻中筛选出来的特征。本文针对这三个问题提出了相关的解决方案。

3.2.1 特征定义与提取

在社区聚类问题中，通常把一个紧密联系的团体作为一个整体来进行，原因在于这些团中往往联系紧密，算法[28]利用到了社区中的小团体特性。我们的特征定义也是基于团这个概念。一个社区的特征定义为在同一个社区里连接强度大于接受阈值 $p*F$ 的小团体，这个特征可以通过 2.2.3 节中的 KKM 值来衡量。阈值 p 代表特征提取的强度比，F 是当整个团体全连接时的 KKM 值，p*F 等于接受阈值。当阈值 p=0 时，每一个社区就是一个特征。我们挑选部分过去时刻的优秀解，将这些解中存在的特征的都选取出来。在图 3- 1 中，假定该图被分类到同一个社区，那么{1，2，3，4}构成一个特征，{5，6，7}构成一个特征，这些团的特征是内部联系紧密。特征提取特征的算法描述如下：

Feature Extraction Algorithm

Input: *S* is the solution array， n is the length of *S*, *F* is full connection's KKM.

Output: $O = \{O_1, \ldots, O_1\}$， O is the original feature sets of all extracted features.

```
for t ∈ [1， n] do
    aux[t] = t
end for
for i ∈ [1， n] do
    p = S[i]
    aux[i] = FINDSET(p)
end for
O = {}
for i ∈ [1， n] do
    j = FINDSET(S[i])
    add node i to group j in O
end for
return all f in O which KKM(f) < p * F
FINDSET(j):
while aux[j] != j do
    j = aux[j]
end while
return j
```

算法 1 特征提取

3.2.2 评估并筛选特征

随着时间的改变，部分特征可能会不再有参考价值，因此需要对于提取出的特征进行筛选操作。我们提出了基于启发式的特征筛选方法，通过一组对照解来对提取出的特征进行评估，只有通过了评估的特征才能够被利用到当前时刻。对

照解是通过一个快速的启发式算法LPA算法来产生。当获得了Rn个对照解之后，利用这 R 个对照解，分别对于每个特征进行评估。只有某个特征与对照解相似度大于50%，则认为对照解支持该特征。最后采用投票统计的方式，如果一个特征在所有的参照解中获得的支持次数超过一定的比例，这个比例称作通过率（Tr），我们认为该特征是当前时刻的有效特征，那么应当保留该特征。基于启发式的特征筛选算法描述如下：

Feature Selection Algorithm

Input: $O = \{O_1, \ldots, O_n\}$， O is the original features. H is the LPA results

Output: $F = \{F^1, \ldots, F^m\}$， F is the filter features.

```
01    aux[] is the zero array which length equals to n
02    for Oi ∈ O do
03        for Hj ∈ H do
04    if calculateSimilarity(OiHj) > 0.5:
05            aux[i] = aux[i] + 1
06    end for
07    for t ∈ [1, n] do
08        if aux[t] > Tr
09            F.add(Oi)
10    end for
11    return F
```

算法 2 特征筛选

3.2.3 标签传播算法

标签传播算法是一种快速随机的启发式算法，能够在相对短时间内获得一个相对较优解。由于需要对特征进行评估，因此标签传播算法被我们选择用来做初步地筛选。

标签传播算法的流程如下：首先为所有的节点指定一个唯一的标签，然后逐步刷新所有节点的标签，每一次刷新，将节点的标签改变成为它与它所有邻居节点中包含最多的标签。如果最多的标签不唯一，则随机选择一个标签。

3.2.4 特征团体的概率迁移

我们提出了特征迁移算法将特征加入到当前种群中。同时为了保证种群的多样性，我们只按照一定的概率（Tp）来对特征进行迁移。特征迁移算法的流程是，将每个特征集中的社团结构，以 Tp 的概率随机赋予到每一个种群中的解中，从而替换掉那些解原始的标签，算法的伪代码如下所示：

Feature Transfer Initialization Framework

Input: Given current network N， *t* is the current time， $F = \{F^1, \ldots, F^m\}$ is the set of selected feature， *s* is the maximum population size， *tp* is the transfer probability.

Output: A population p with a group of transferred solutions.

```
p = ∅
for t ∈ [1, s ] do
    create a random individual d
    for i ∈ [1, m ] do
        if random(0, 1) < tp
            set all nodes in F^i of d to the same group
        end if
    end for
    p.add(d)
end for
return p
```

算法 3 特征迁移

3.3 TMOGA 算法框架

到目前为止，整个 TMOGA 的算法的一些细节已经全部介绍完毕。这部分总结并且列举出整个算法的框架，算法可以看做是基于 NSGA-II 算法，特征迁移过程发生在 NSGAII 的种群初始化中。在第一个初始时刻，由于没有之前时刻能不能计算出相似度 *NMI*(*t-1*)的值，因此当 t=1 时刻，只能使用静态单目标网络的聚类的算法来获得初始解，这一部分被认为是初始化的过程。我们采用了单目标 DYNMOGA-single 来对 t=1 时刻获得最终的解。在后续时刻 t>2 时刻，开始进行特征提取，迁移算法。同时可以分别计算出两个目标值 Q 与 NMI。TMOGA 算法的框架如下：

TMOGA Framework

Input: Given a dynamic network $N = \{N^1, \ldots, N^T\}$, T is the time steps.

Output: A clustering for each network

Initialize: apply DYNMOGA-single to get S(1)
for t $\in$ [2, *T*] **do**
O(t) = featureExtraction(S(t-1))
    F(t) = featureSelection(O(t))
    B(t) = featureTransfer(F(t))
    while termination condition is not satisfied **do**
        parents = selection(B(t))
        offspring = crossover(parents)
        mutation(offspring)
        evaluate(offspring)
        assign offspring a rank
        combine parents and offspring
        Select points according to the crowded distant
    end while
    I(t) = the solution which have the maximum modularity value in B(t)
    S(t) = I(t)
end for
return *EP* ;

算法 4 TMOGA 框架

第四章 实验部分

本章主要通过实验来证明我们算法的实际效果。参与优化的两个目标是当前时刻的模块度（modularity 值），以及与前一时刻最优解的近似程度（NMI 值），这两个目标的计算方法在上文中已经给出。本章首先介绍了在实验中与我们算法进行对照的其他算法，然后介绍了评价实验结果的指标。4.4 节通过同其他经典算法之间的效果进行对比，证明了我们算法的分类准确度高。4.5 节研究在大规模网络问题中各种算法的表现，实验结果说明我们的算法在大规模问题中有更好的收敛性。4.6 节将我们的算法应用到实际问题中，表明了对于实际问题也有更高效的分类效果。

4.1 对照算法

DYNMOGA[2]算法，2014 年由 Folino 提出，基于 NSGA-II 框架，同时加入了一种动态自适应参数的方法，能够自动在聚类准确度与结果偏差之间找到一个最佳平衡。

Kim-han，[24]中提出的基于粒子群与密度的聚类进化算法，将一个网络结构当成是一个 nano 社团，最终由这些社团形成一个小整体来进行优化。

FacetNet 算法，[20]中提出的一种基于团块的数学方法，同样也被用于动态社区检测问题。

4.2 评价指标

我们的测试问题主要包括两类，一类是有标准解的问题，我们可以通过将聚类的结果同标准解的 NMI 进行对比，以及另一个参数错误率。另一类是没有标准解的实际问题，这类问题我们仅仅通过在规定时间内给出解的最终结果的目标值来评估。

错误率（Error Rate），我们采用多个参数来评价对于实验结果进行评价，其中之一是错误率。错误率度量了分类结果与真实聚类结果之间的空间距离，通过计算两个矩阵之间的差值矩阵的第二范式的值获得。计算方法是将得到聚类结果转化成一个 $n * k$ 的矩阵 Z，其中 n 代表点的个数，而 k 代表社团的个数，同时

将真实解的社区情况也同样做成一个矩阵 G，然后通过公式：

$$Error = || ZZ^T - GG^T ||$$ （3-1）

另一个指标是 NMI 值，上文中已经提到过。而这里的 NMI 是指与真实结果之间的 NMI 值，当 NMI 值为 1 时，说明获得的解与真实解等价，也就是效果最好。

4.3 实验一：标准测试集

本实验的第一个测试集来自 Lin 在 2009 年提出的测试集[20]，这个测试集是基于 Girvan 和 Newman 提出的经典测试集[1]生成方式产生的。总共包含 4 个社区，每个社区 32 个节点，总共 128 个节点，每个节点平均之间有 Z 条边与其他节点连接，Z 越大说明整个社区的噪声强度越大。由于该测试集是静态网络，为了引入动态特性，我们分别进行调整成 2 类测试集，分别称作 SYN-FIX 与 SYN-VAR 问题。

4.3.1 SYN-FIX 问题

SYN-FIX 问题是基于 Lin 提出的测试的基础上，做固定社区个数的动态变化。在每个时间点，从前一时刻的每个社区随机选择 3 个节点，将他们随机分配到其余的 3 个社区当中，形成新的社区网络。

4.3.2 SYN-VAR 问题

SYN-VAR 问题是对 Lin 测试集进行社区数量不固定的调整。这个问题的整个网络包含了 256 个节点，4 个社区，每个社区有 64 个节点。每个时刻从每个社区选出 8 个节点组成一个新社区，到 5 时刻时候，每个社区都只剩下 32 个节点，然后从第 6 时刻开始将节点返回到原来的社区，从而变成原来的状态，即整个社区的总数如下变化，从 4，5，6，7，8，8，7，6，5，4 从而产生共计 10 个时间点。

4.3.3 实验参数

种群大小	进化代数	变异率	交叉率	通过率	对照解数量	迁移概率
150	100	0.2	0.8	0.5	10	0.8

为了降低偶然性，每个算法，对于每组测试集，总共运行 20 次，记录最后获得的解与真实解的 NMI 的平均值。

4.3.4 实验结果

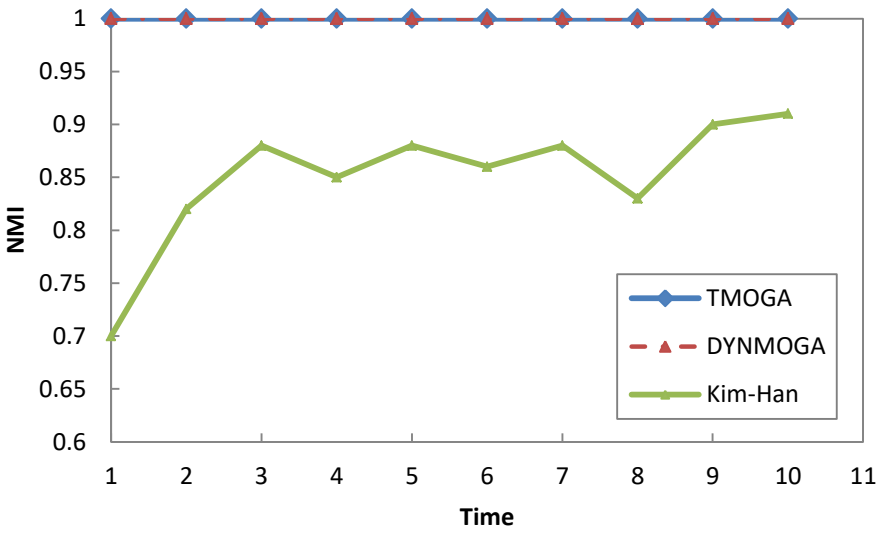


图 4- 1 SYN-FIX， Z=3

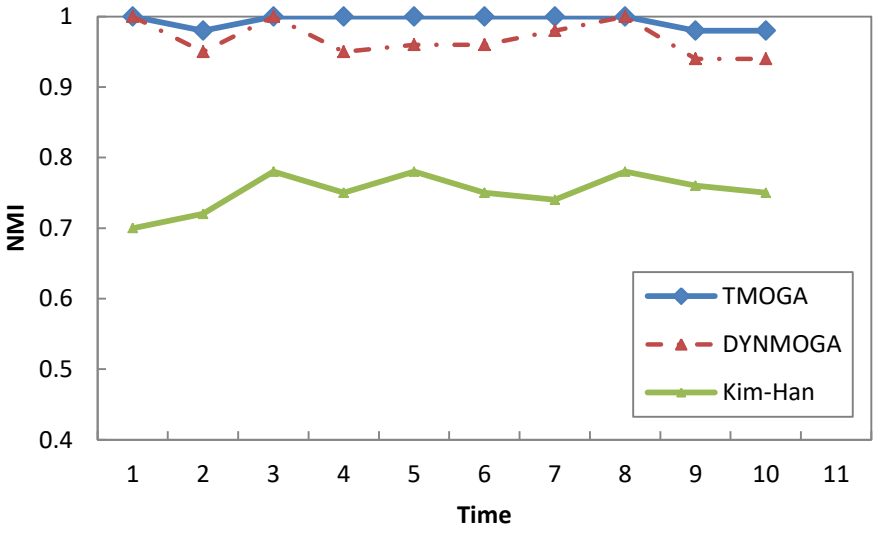


图 4- 2 SYN-FIX， Z=5

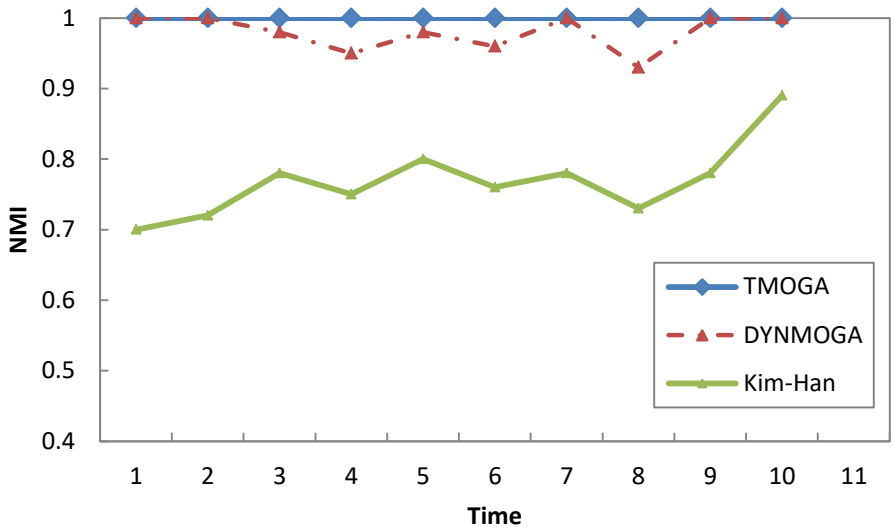


图 4- 3 SYN-VAR，Z=3

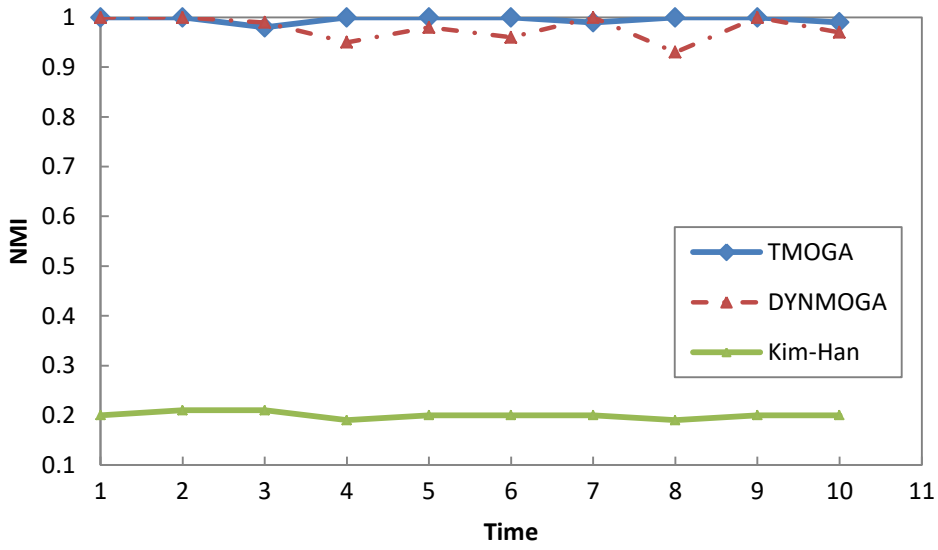


图 4- 4 SYN-VAR，Z=5

如图 4-1 所示，实验表明在 SYN-FIX 测试集上，当 Z 等于 3 的时候，我们的算法 TMOGA 总能够获得最优解，DYNMOGA 同样也能获得最优解，而 Kim-Han 算法并不能获得最优解。而当 Z=5 时，整个社区的噪声增大，如图 4-2 所示，TMOGA 仍然能找到最优解，而 DYNMOGA 在某几个时刻并不能获得最优解，但是总体远远优于 Kim-Han 算法。

而对于 FYN-VAR 问题，TMOGA 算法仍然能够获得接近于真实解的最终解，DYNMOGA 的效果也没有因为社区数量调整而减少过多。但是对于 Kim-Han 算法效果大打折扣。

这个实验说明我们的 TMOGA 算法在小规模问题上有着很好的收敛性，几乎总能获得最优解，而 DYNMOGA 效果也非常理想，但是对于 Kim-Han 算法，并不能获得最优的解。

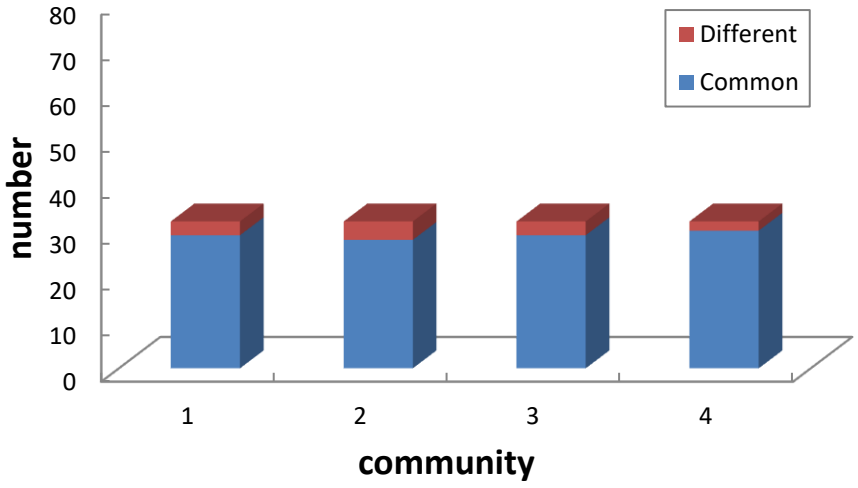


图 4- 5 SYN-FIX，Z=3，T=6，Common 代表实际标签与特征中相同的部分，Different 代表实际标签不在特征中的部分

从上图可以看出，对于 SYN-FIX 问题，保留下来的最终特征占据了每个社区总标签的大部分。说明我们的算法的初始解的效果已经接近真实解，在这个基础上进行搜索，能够将搜索范围缩小到仅仅需要对红色部分找到对应的社区位置，显然可以增加全局的搜索速度。

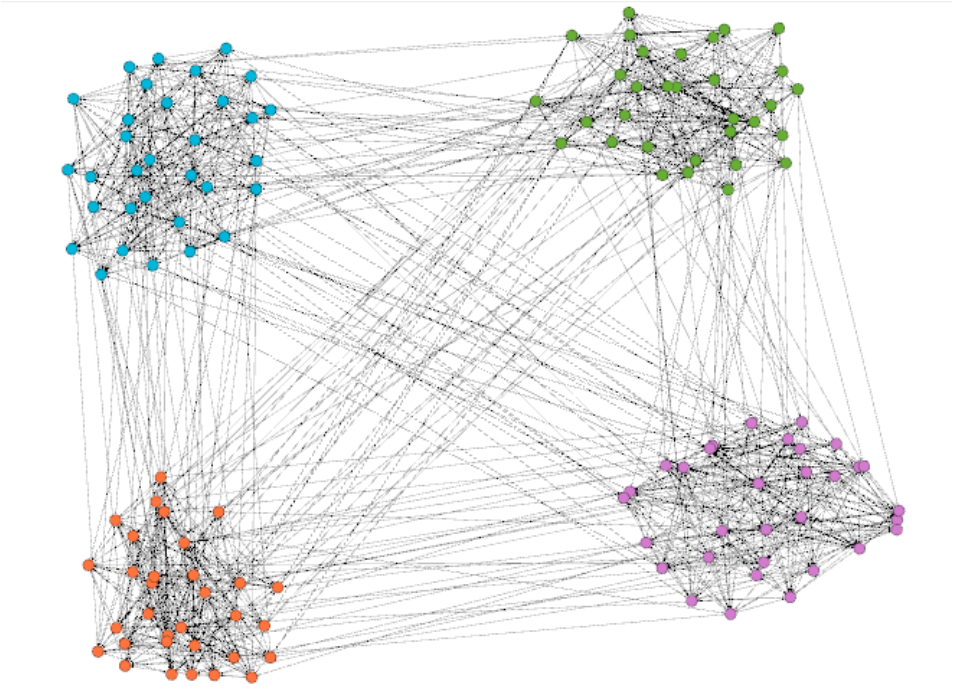


图 4- 6 SYN-FIX，Z=3，t=3 真实解

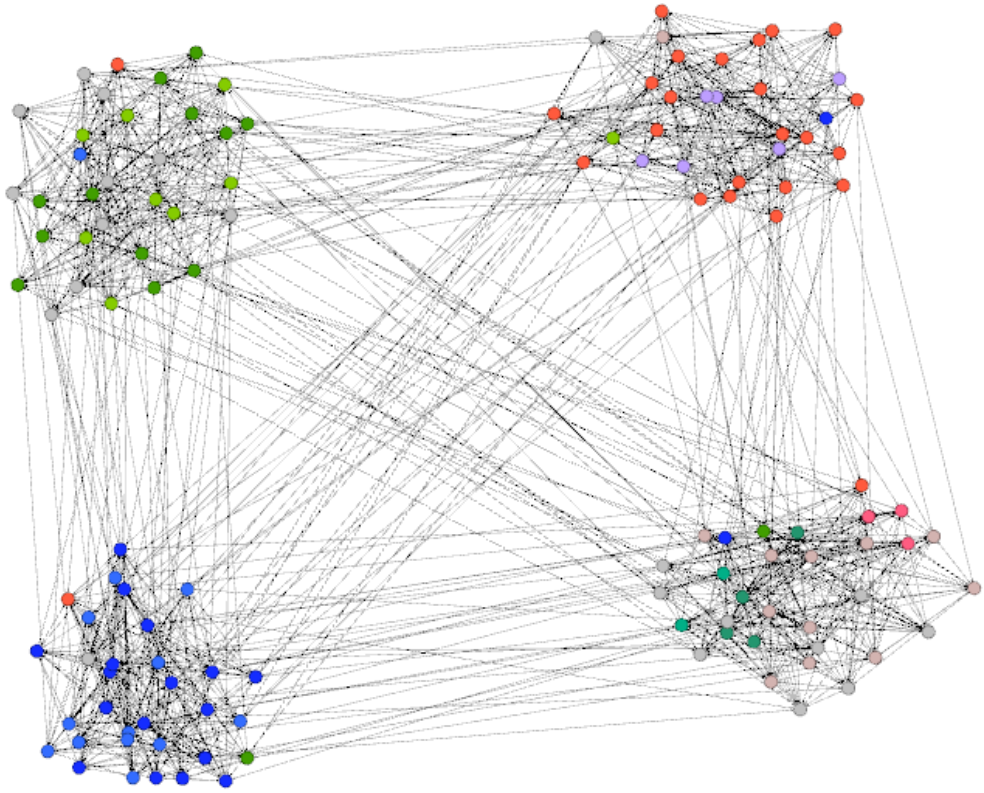


图 4- 7SYN-FIX，Z=3，t=4 时刻特征

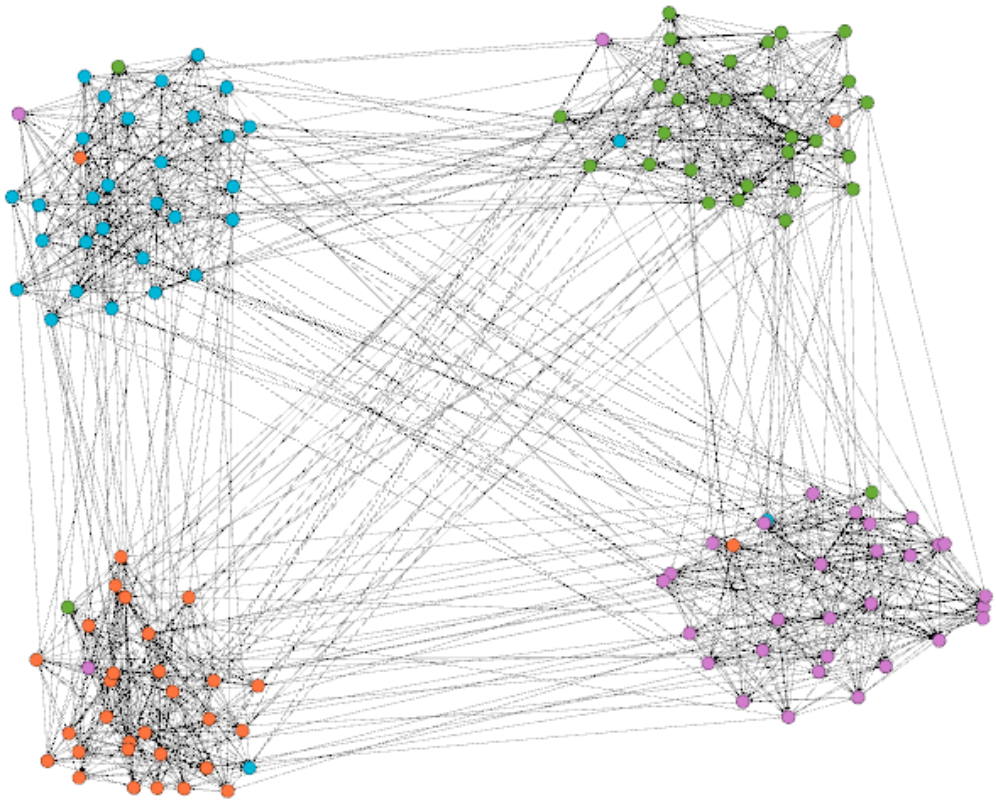


图 4- 8 SYN-FIX，Z=3，t=4，真实解

图 4- 6 展示了 SYN-FIX 问题中，t=3 时刻的聚类结果。而图 4- 8 是 t=4 时刻的聚类结果。对于 t=3 时刻的特征结果提取如图 4- 7 所示。可以看到左上，右下等几个大社团被聚类拆分成了几个子特征，但是整体的社团结构然接近。同标准结果进行对照发现，我们的特征迁移产生的初始解已经接近 t=4 时刻的真实解的标签分布情况。

4.4 实验二：大规模测试集

本部分实验验证算法在大规模测试集上的表现。常采用的数据集是基于 Greene 提出的动态网络社区演化模型[29]生成的。该模型提出了 4 种不同类型的社区演化的特点，这几个数据集分别是 Birth and Death， Expansion and Contraction， Intermittent communities， Merging and splitting。根据这 4 种演化特点，我们基于原始的 1000 个结点的测试集，平均每个节点的度为 15，最大的度为 50，总共的社区数量大约在 20 到 50 个之间，产生了相应的测试集。

4.4.1 模型描述

Birth and Death 模型：从第二时刻开始，将 10%的节点从已有的社区中拿取，从而产生新社区，同时 10%的社区被移除。

Expansion and Contraction 模型：每个时刻，选中原来社区的 10%的社区，将每个社区都扩张或者缩小 25%的原有规模，扩张时新的节点从原来的节点中随机抽取。

Intermittent Communities 模型：10%的来自第一时刻的社区被隐藏。

Merging and Splitting 模型：在每个时刻，10%的社区分裂，10%的社区被选中并且两两合并。

4.4.2 实验结果

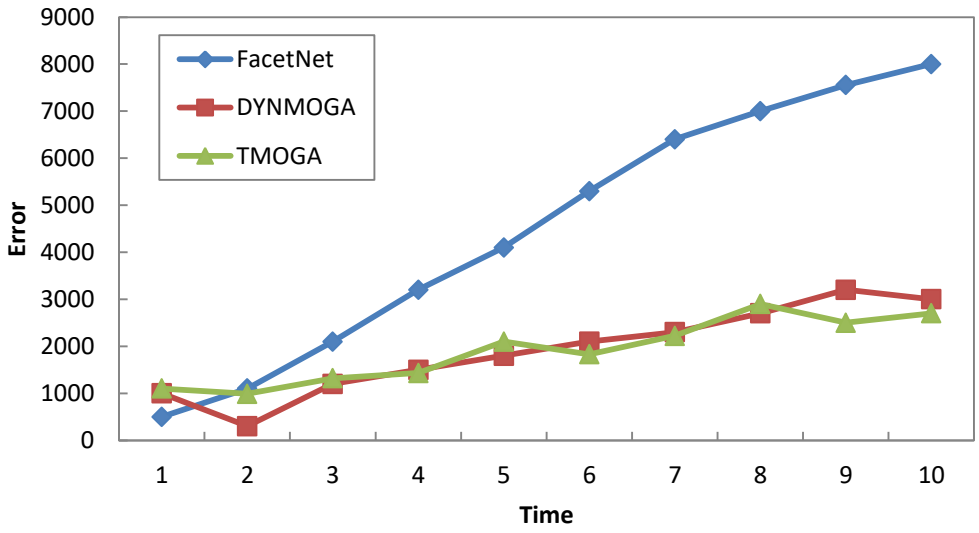


图 4- 9 Birth and Death

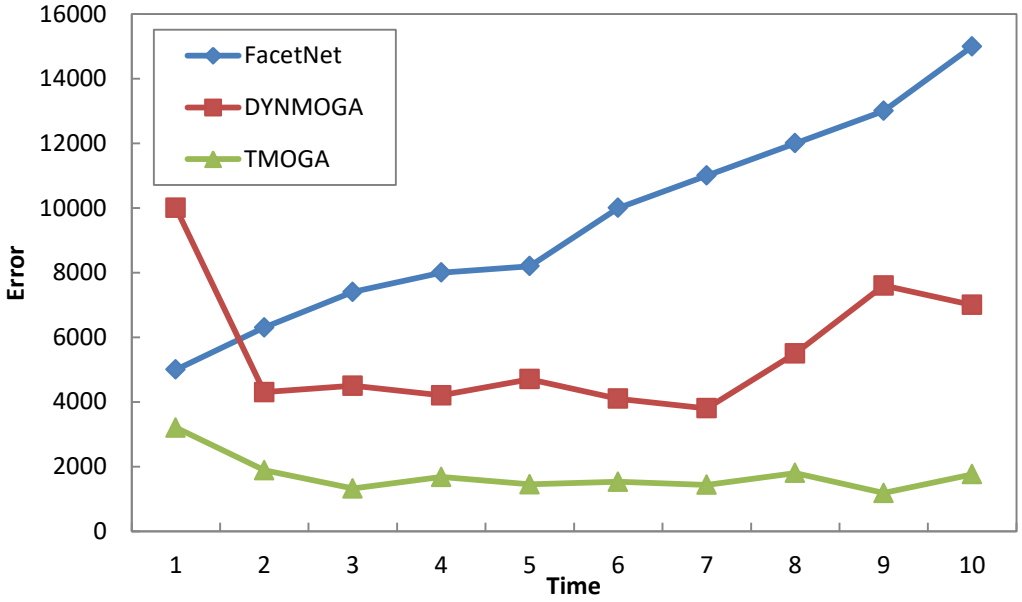


图 4- 10 Expansion and Contraction

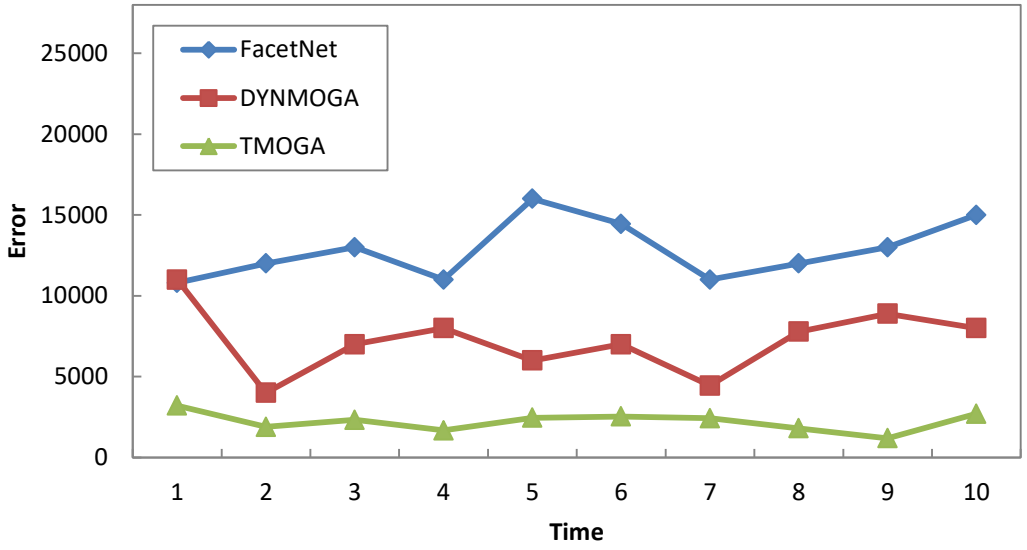


图 4- 11 Intermittent Communities

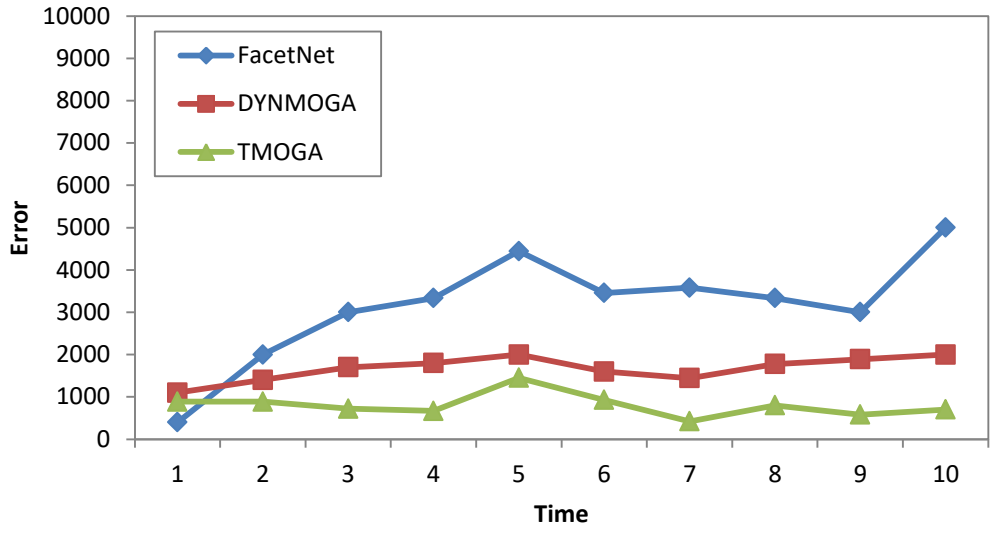


图 4- 12 Merging and Splitting

根据实验结果，总体来说我们的 TMOGA 算法优于其他两个算法。对于图 4-9 Birth and Death，TMOGA 算法的效果与 DYNMOGA 接近。因为该模型的每个社区都发生了一定的变动，所以整个社区特征相对来说保留下来的完整性减弱。但是变动的的比率只占据了整个社区节点数的小部分，因此经过特征迁移以后的种群仍然对进化有促进作用。对于 Expansion and Contraction 与 Intermittent communities 这两个实验模型，我们的算法明显优于其他两个算法。因为这两个模型仅仅只有少部分社区发生过变动，其他社区完整保留下来，因此这些社区特征也能够被完整地保存下来。因此，说明我们的算法的特征提取最适用于某些社区变动小甚至不变的情况，通过提取这些社区的特征，使得这些随着时间不变的有价值的信息能够检查并且保存到下一时刻中，这些特征融入到了种群的新一代作为初始解的特征，能够大大缩小整个进化算法的搜索空间，从而使得最后能够获得更好的结果。

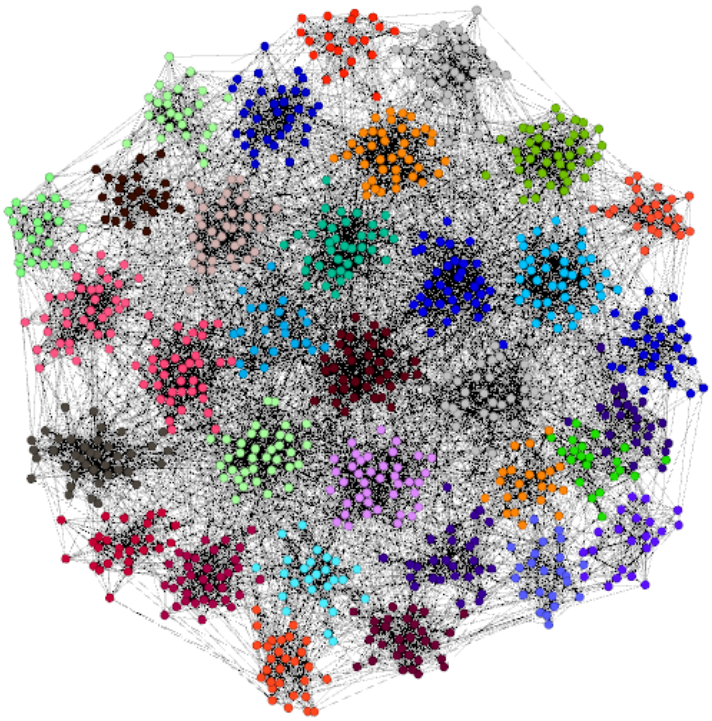


图 4- 13 Birth and Death，聚类结果

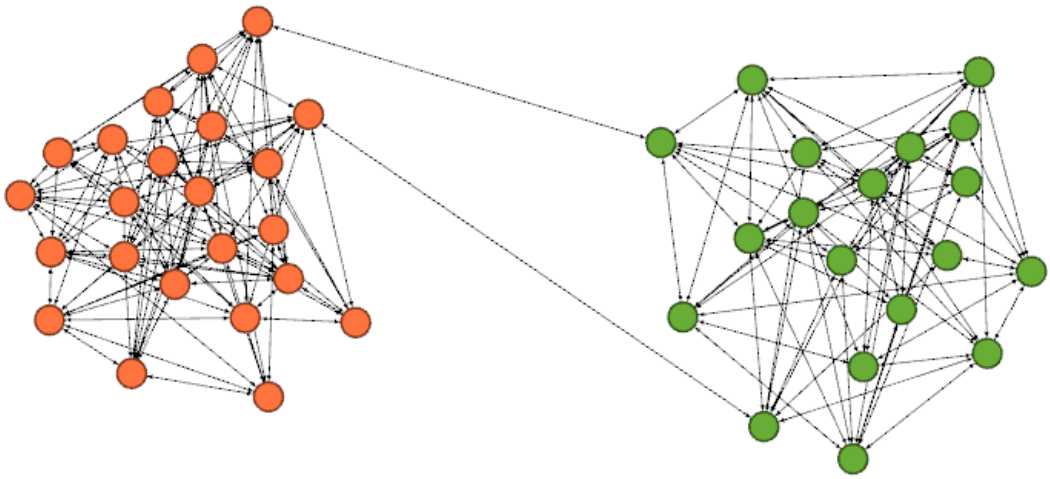


图 4- 14 选取 2 个社区的结果

4.5 实验三：实际生活中的问题

前两部分实验验证了我们算法无论是在小规模还是大规模问题上，相对于传统的算法都有更好的表现，而这一部分主要针对 TMOGA 算法是如何提高进化效果做进一步深入研究。我们选取了一个实际生活中的问题来进行探讨。它们是移动电话通讯网络，这个实际问题并没有给定的真实解。

4.5.1 移动电话通讯网络

该数据集来自于[30]，是位于西班牙的 Isla Del Sueño 地区在 2006 年 6 月这一时段中，选取的 400 个移动电话以及它们互相之间产生的所有通讯记录。在网络中，每个节点代表一个移动电话终端，节点之间由带权重的边关联起来，表示节点在这段时间之内通话的频率，总共有 400 个节点。在实际中，总共约 30 个社区。

4.5.2 实验结果

第一个实验是移动电话数据的实验，选取 T=3 时刻，整个种群进化的情况。通过 TMOGA 与 DYNMOGA 进行对比。

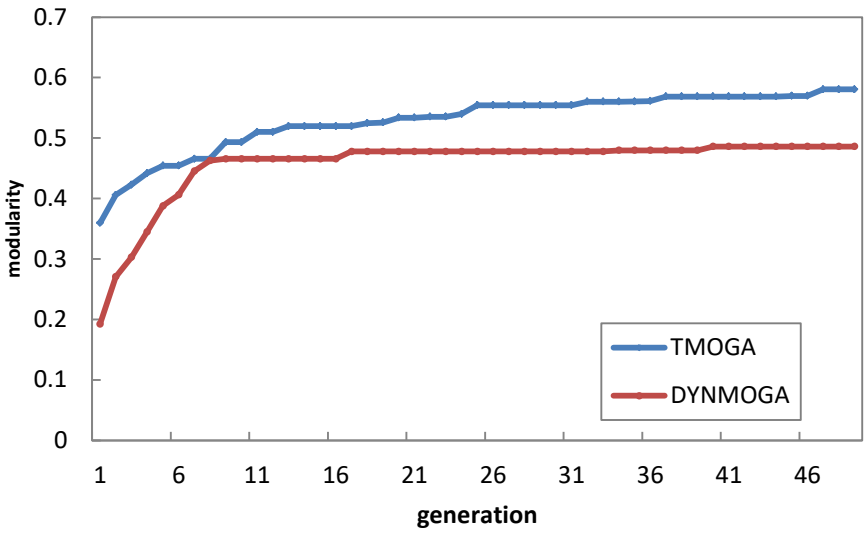


图 4.6- 1

（移动电话问题，T=5，最大代数=50，横轴代表当前进化代数，纵轴代表当前种群中最优解的模块度）

如图 4.6- 1 所示，是移动电话问题在 T=5 时刻整个进化过程中种群，可以看出 TMOGA 算法开始时刻的模块度领先于 DYNMOGA 算法，因为 TMOGA 算法中的种群并不是像 TMOGA 中一样是随机生成的，而是保留了一部分前一时刻特征的初始种群。实验结果证明了我们的算法在初期的几代中优势明显，并且在后期可以将节省下来的计算量用在搜索更多可能的解上，使得最终结果能够获得更好的收敛性。因此 TMOGA 算法最终获得的模块度仍然大于 DYNMOGA 所获得的。

下面部分研究整个算法中的一些参数对于实验效果的影响。参与研究的参数包括变异率(Mr)，交叉率(Cr)，参考解数量(Rn)，筛选通过率(Tr)。

G	Mr	Cr	Rn	Tr	Error
1	0.8	0.2	10	0.5	139.99
2	0.8	0.5	10	0.5	141.12
3	0.8	0.8	10	0.5	132.72
5	0.5	0.2	10	0.5	202.25
7	0.8	0.2	50	0.5	129.61
8	0.8	0.2	100	0.5	141.66
10	0.8	0.2	10	0.5	316.07
11	0.8	0.2	10	0.8	334.79

根据实验结果，变异率的大小设定对于整个实验效果影响并不大。而交叉率过低会导致最后结果变差。对于参考解的设定，并不是越多越能够有更好的效果，参考解个数超过一定数量之后，最后的结果并没有明显变化。但是参考解过少会使得最后的结果变差。最后，参考解的通过率，设定太高就会减少特征的数量，但是会保留强特征；设定过低会导致无用的特征保留下来。这两种情况下都会导致整个算法效果减弱。

4.6 实验四：与单目标对比

本节实验中研究同时优化多目标是否能够优于单独优化一个目标。我们将两个目标值进行修正，使得第二个目标值（NMI）始终等于 0，从而退化成只考虑模块度 Q 的单目标算法（TMOGA-single）。同时把 DYNMOGA 算法也做同样的调整成为 DYNMOGA-single。使用 4.5 节中的测试集（Merge and Split）。

4.6.1 实验结果

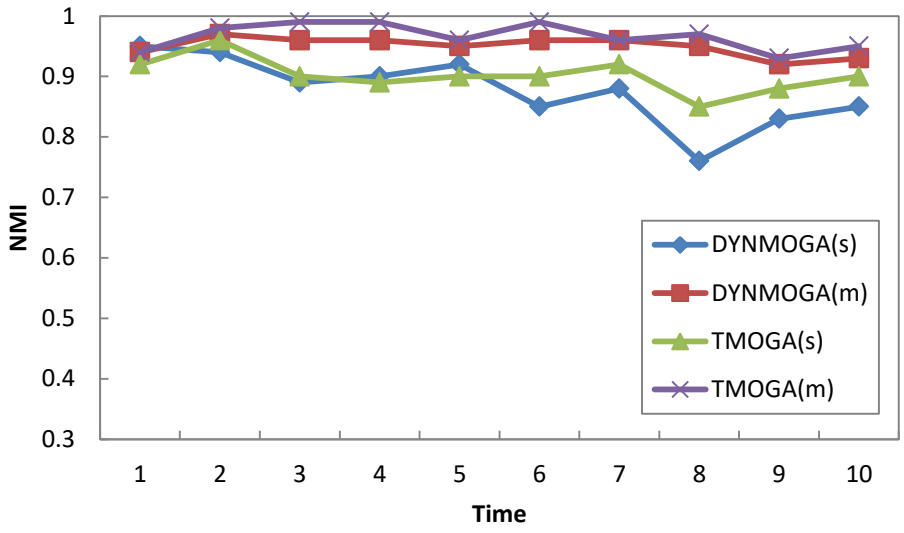


图 4- 15 Merge and Split

如图所示，其中的评价指标 NMI 是指算法的解与真实结果的 NMI 值。从实验结果可以看出，同样的算法，单目标所获得的最终结果始终都要比同时优化两个目标的实验结果要差。这个实验表明了单独优化一个目标能够取得更好的效果。

第五章 总结与展望

5.1 总结

动态网络社区检测问题是网络研究的一个重要问题，也是目前学术界广泛研究的经典网络优化问题。目前已经有大量学者提出了一系列的算法来解决这类问题，同时也提出了很多经典测试集来验证算法的效果。在本文中，针对动态网络结构中的动态特性，我们提出了一个基于特征迁移的多目标进化算法（TMOGA），改进自传统的多目标算法 NSGA-II，将迁移学习中的特征迁移思想引入到了解决这个问题当中，取得了非常好的效果。

本文首先介绍了多目标优化问题、动态社区网络检测问题的定义。然后介绍了该领域相关的研究工作。接下来，提出了我们的基于特征迁移的多目标进化算法，解释了如何在动态网络中提取并且迁移特征，同时给出算法的具体实现过程以及伪代码。最后，本文设计了一系列对照实验，通过 Lin 与 Greene 提出的社区演化模型产生了参与实验的测试集。通过这些测试集验证了我们的 TMOGA 算法在各种规模的问题上运行的效果。实验结果表明，在小规模测试集上，我们的算法几乎总能够完全求得标准解。在中大规模问题上，我们的算法相对于其他算法，具有更好的收敛速度。

5.2 展望

本文提出的算法在各种规模的网络中均取得良好的效果，但是仍然有不足，我们相信这个领域还有更多的课题值得学者们研究。未来的研究可以从以下几个方面开展：

1) 我们的算法仅仅只参与到进化算法的种群初始化阶段，未来的研究可以将迁移学习的思想进一步引入到进化过程中，从每一个代的变化中找寻不变的特征并且保留。

2) 可以尝试对于特征进行另外的定义，或者采用更好的特征筛选，特征迁移方法来代替我们提出的方法。或者将迁移学习中的其他迁移方式，比如样本迁移，模型迁移等方式引入动态问题中。

3) 可以增加更多的目标值，或者对于原来有的两个目标值进行调整。从而取得更好的效果。

4) 考虑更大规模的问题，对于算法进行并行计算优化，可以改进成为分布式进化算法，转移到 Spark 等平台，能够大大提高算法的运行效率。

5) 研究更多的测试集，从而更加全面评估算法对于各类问题的适应度。

参考文献

[1] Girvan, M. and M.E. Newman, Community structure in social and biological networks. Proceedings of the national academy of sciences, 2002. 99(12): p. 7821-7826.

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

[3] Newman, M.E., Fast algorithm for detecting community structure in networks. Physical review E, 2004. 69(6): p. 066133.

[4] Dai, W., et al. Boosting for transfer learning. in Proceedings of the 24th international conference on Machine learning. 2007: ACM.

[5] Deb, K., et al., A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 2002. 6(2): p. 182-197.

[6] Miettinen, K., Nonlinear Multiobjective Optimization, volume 12 of International Series in Operations Research and Management Science. 1999, Kluwer Academic Publishers, Dordrecht.

[7]Ruzika, S. and M.M. Wiecek, Approximation methods in multiobjective programming. Journal of optimization theory and applications, 2005. 126(3): p. 473-501.

[8] Wiecek, M.M., W. Chen, and J. Zhang, Piecewise quadratic approximation of the non‐dominated set for bi‐criteria programs. Journal of Multi‐Criteria Decision Analysis, 2001. 10(1): p. 35-47.

[9] Srinivas, N. and K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation, 1994. 2(3): p. 221-248.

[10]Newman, M.E. and M. Girvan, Finding and evaluating community structure in networks. Physical review E, 2004. 69(2): p. 026113.

[11]Zitzler, E., M. Laumanns, and L. Thiele, SPEA2: Improving the strength Pareto

evolutionary algorithm. 2001, Tik-report.

[12]Corne, D., J. Knowles, and M. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. in Parallel problem solving from nature PPSN VI. 2000: Springer.

[13]Bader, J. and E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary computation, 2011. 19(1): p. 45-76.

[14]Zhang, Q. and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE transactions on evolutionary computation, 2007. 11(6): p. 712-731.

[15]Yuan, Y., H. Xu, and B. Wang. An improved NSGA-III procedure for evolutionary many-objective optimization. in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. 2014: ACM.

[16]Benedetti, A., M. Farina, and M. Gobbi, Evolutionary multiobjective industrial design: The case of a racing car tire-suspension system. IEEE transactions on evolutionary computation, 2006. 10(3): p. 230-244.

[17]Deng, G., et al. MOEA/D for Energy-Aware Scheduling on Heterogeneous Computing Systems. in Bio-Inspired Computing-Theories and Applications. 2015: Springer.

[18]Raghavan, U.N., R. Albert, and S. Kumara, Near linear time algorithm to detect community structures in large-scale networks. Physical review E, 2007. 76(3): p. 036106.

[19]Lancichinetti, A., S. Fortunato, and J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics, 2009. 11(3): p. 033015.

[20]Lin, Y.-R., et al., Analyzing communities and their evolutions in dynamic social networks. ACM Transactions on Knowledge Discovery from Data (TKDD), 2009. 3(2): p. 8.

[21]Tasgin, M., A. Herdagdelen, and H. Bingol, Community detection in complex networks using genetic algorithms. arXiv preprint arXiv:0711.0491, 2007.

[22]Gong, M., et al., Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A: Statistical Mechanics and its Applications, 2012. 391(15): p. 4050-4060.

[23]Wang, Z., et al., Deployment Optimization of Near Space Airships Based on MOEA/D with Local Search. 2014 Ieee Congress on Evolutionary Computation (Cec), 2014: p. 2345-2352.

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

[25]Ma, X. and D. Dong, Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. IEEE Transactions on Knowledge and Data Engineering, 2017.

[26]Pan, S.J. and Q. Yang, A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010. 22(10): p. 1345-1359.

[27]Park, Y. and M. Song. A genetic algorithm for clustering problems. in Proceedings of the third annual conference on genetic programming. 1998.

[28]Wen, X., et al., A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE transactions on evolutionary computation, 2016.

[29]Greene, D., D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. in Advances in social networks analysis and mining (ASONAM), 2010 international conference on. 2010: IEEE.

[30]Mini Challenge 3: Cell Phone Calls 2008; Available from: http://www.cs.umd.edu/hcil/VASTchallenge08/.