单位代码：10293 密 级：

南京邮电大学

专业学位硕士论文

论文题目：基于遗传算法的复杂网络社区检测的应用研究

学 号 1213022635

姓 名 陈灵刚

导 师 周井泉教授

专业学位类别 工程硕士

类 型 全日制

专业（领域） 电子与通信工程

论文提交日期 二〇一六年三月

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生签名：陈灵刚 日期：2016.4.10

南京邮电大学学位论文使用授权声明

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

涉密学位论文在解密后适用本授权书。

研究生签名：陈灵刚 导师签名：周井泉 日期：2016.4.10

Application of Genetic Algorithm Based on Complex Network Community Detection

Thesis Submitted to Nanjing University of Posts and Telecommunications for the Degree of

Master of Engineering

By

Linggang Chen

Supervisor: Prof. Jingquan zhou

February 2016

摘要

近年来，随着复杂网络理论及其相关应用研究的兴起，人们开始尝试利用这些新的理论工具来研究现实社会中的各种大型复杂系统。因此，对复杂网络社区结构的检测，逐渐成为了研究的热点。社区结构是复杂网络最重要的拓扑结构属性之一，它揭示了复杂网络的隐藏规律和行为特征。社区结构的数学模型是指在一个复杂网络中内部连接紧密而外部连接稀疏的节点集。传统的方法需要预先设定权重参数来控制对目标函数的不同侧重，并且不能够自动识别社区个数，在寻优过程中会出现“早熟”和效率低下问题。

本文研究的多目标自适应快速遗传算法是传统遗传算法的演进算法，用于复杂网络社区结构的检测。首先，它将社区检测问题转化为多目标优化问题，构建社区分值和社区适应度两个目标函数。其次，引入外部精英基因库，用于存储适应度较高的非劣解，对于外部精英基因库已经存在的重复个体，不用再进行重复解码、计算个体适应度值等一系列过程。同时，执行自适应遗传算子，返回一组在两个目标函数之间折衷的非支配解。最后，选取一个模块度最高的 Pareto 最优解，解码生成一组独立的子网络，并用互信息度量和模块度去评价算法的性能。仿真表明，多目标自适应快速遗传算法大大地提高了复杂网络社区结构检测的精确度，并且能更好地发现复杂网络的层次结构。

关键词：复杂网络，遗传算法，自适应，多目标，精英基因库

Abstract

In recent years, with the rise of complex network theory and related applied research, people begin to try to apply these new theoretical tools to study a variety of complex systems of real world. Therefore, the community detection of complex networks gradually becomes a hot research. Community is one of the most important social network topology property, which reveals the hidden laws of social networks. Mathematical model of community structure refers to a set of nodes whose internal connection is tight and external connection is sparse. The traditional methods require pre-set weight parameters to control the objectives. But they are not able to automatically identify the number of communities in the optimization, and it processes the problem of "premature" and inefficiency.

In this thesis, the fast adaptive genetic algorithm is a evolution of genetic algorithm for detecting community structure in complex networks. First, this algorithm transforms the detecting problem into a multi-objective optimization problem. We build two objective functions, named community fitness and community score. Second, we construct a external elite gene pool for storing solutions which have high fitness value. For those individuals which already exist in external elite gene pool, we needn't to decode and calculate individual fitness value. Meanwhile, we perform adaptive genetic operator, which returns a set of non-dominated solutions of a pair of objective functions. Finally, we select a Pareto optimal solutions of the highest modularity, decodes and generates a set of independent sub-networks. We use *NMI* and modularity to evaluate performance of the algorithm. Simulation shows this algorithm greatly improves the accuracy of detection of complex networks and it can be better to find a hierarchy of complex networks.

Key words: complex network, genetic algorithm, adaptive, multi-objective, elite gene pool

目录

第一章 绪论

1.1 课题的研究目的和意义

随着互联网[1]的运用和普及，信息科学技术得到了前所未有的发展，人类社会在 21 世纪跨入了一个由各种网络构成的数字化时代。交通运输网、通信网、社交网络等现实网络与人们的生活学习密切相关。复杂网络[2], [3]就是各种网络体系[4]的一个抽象化的有效表示形式，同属于一个复杂网络相同社团的节点更有可能具有相似的性质或相近的功能。研究复杂网络中社团结构的发现算法显得尤为必要，通过这些算法可以找到真实网络中存在的社团，对复杂网络的研究提供了依据。社区检测[5]是利用网络拓扑结构中所包含的信息从复杂网络中解析出其模块化的社区结构。

复杂网络的功能是网络的各个社团之间综合作用的结果，知道网络的总体功能并不一定能知道每个社团各自的功能。因此，对复杂网络社区结构的深入研究有助于人们对整个网络的模块、功能及其演化的研究。

随着复杂网络理论研究的深入，人们不断对其物理意义和数学特性进行深入探索。用长远的眼光来看，其应用前景将会越来越好。网络的社区检测对网络的分层可视化有重要意义，网络的可视化是信息可视化的一个重要分支，所谓可视化就是把一个网络的拓扑表示(通常是矩阵表示)投影到二维平面上，一个好的可视化模型能够使节点的重叠和边的交叉越少越好。在生物信息学当中，生物网络的功能分析是生物信息学家非常关注的一个研究方向，在生物网络中结构决定其功能，因此分析网络的模块结构能够更深刻地理解生物网络的功能。复杂网络的社团结构研究对 WEB 的数据挖掘与个性化推荐及其用户行为分析大有帮助，WEB 页面之间的链接[6]就是一个具有社团结构的复杂网络，具有相似主题的网页之间存在着更稠密的链接，通过检测 WWW 网络[7], [8]的社团结构能够在分析用户行为的基础上，进行网页的个性化推荐。其次，在医药领域，复杂网络同样有其重要的应用价值。很多传染病的疫苗成本非常高，而且数量有限，为每个人都接种疫苗显然不太可能，因此如何充分利用数量有限的疫苗，做到既最大限度地节约成本，又能够有效地预防传染病的传播，对医学界来说是一个具有挑战性的研究课题。应用复杂网络的相关理论，我们可以采取措施直接或间接地针对集散节点(即那些与很多人联系紧密的人)接种疫苗，取得了很好的效果。可以说，复杂网络理论从医药的角度为传染病的防控提供了科学的依据。

另外，社团结构检测对电子商务[7]也有着重要的推介作用，而且它会对网络安全与管理

维护、社会行为与行政管理、疾病预防与传播控制、防灾减灾与应急处置、生态系统与环境保护等各专业学科的研究产生重要影响。

1.2 课题的研究历史和研究现状

目前，复杂网络社区发现的研究越来越深入，应用也越来越广泛，而且已经引起数学、生物、生命科学、物理、医药卫生、通信等各专业领域学者的关注和参与，这为复杂网络社区结构检测的研究搭建了更加广阔的平台，创造了更加有利的条件。而鲜为人知的是，复杂网络社区结构检测最早却是社会学的研究范畴，人们在研究社会学中的分级聚类[8]问题时，发现它与计算机科学中的图形分割理论有十分密切的关系，这种关系具有社区结构属性，从而衍生出复杂网络社区检测这一门新的学科。

复杂网络一般指节点众多、连接关系复杂的网络。由于其灵活普适的特性广泛应用于各科学领域，吸引了许多学者对复杂系统进行建模和分析。随着研究的深入，复杂网络有着许多显著的性质，如小世界性[9]、无尺度性[10]和高聚类性，而社区结构是复杂网络的另一个重要的属性，网络中由不同性质和类型的节点组成的关系密切的结构称为社区。整个复杂网络由若干个社区组成，社区外的连接相对稀疏，而社区内的连接相对稠密。

复杂网络经历了以下几个发展阶段。

规则图阶段：在 20 世纪初期，科学家一直认为复杂系统的各个对象之间可以用规则的结构来表示，每个对象只与周围少数几个对象存在联系，例如二维平面上的矩形网络。因此，在一个规模比较庞大的复杂系统内，每个对象通常具有近似相同的连接数量，而且对象之间需要通过很长的路径到达彼此[5]。

随机图[11]阶段：20 世纪 60 年代，欧洲数学家 Erdos 和 Reny 建立了随机图理论，奠定了系统性研究复杂网络的理论基础。他们认为在用图表示的一个复杂网络内，对于每一个 ER 随机图，给定一个任意的概率 P，他们都会同时具备某种性质或没有某种性质。在这一类网络中，网络中节点的度序列一般服从泊松概率分布[12]。与规则图不同的是，随机网络的对象之间通过很短的平均路径即可彼此互相到达，这一数学模型的提出，符合大型复杂系统内部对象之间的信息快速传递的需要。

复杂网络阶段：20 世纪 90 年代末，随着计算机信息网络技术的大力普及，研究者发现大量的真实网络既不是规则网络结构，也不是随机网络结构，而是具有与这两者不同的，具有微观数学统计特性的复杂网络结构。1998 年 6 月，美国 Cornell 大学 Strogatz 教授和其学生 Watts 博士在 Nature 杂志上发表了题为《collective dynamics of “Small world"networks》的

文章[13]。1999 年 10 月，美国 Notre Dame 大学物理系博士生 R.Albert 和其导师 A.L.Barabasi 教授在 science 杂志上发表了《Emergence of scaling in Random Networks》[14]。这两篇文章分别揭示了复杂网络的小世界特征和无标度特性，使复杂网络的相关性理论研究跨入了一个新的阶段。这两种网络特性的发现对复杂网络的研究具有划时代的意义，它们能够指导人们更好地研究网络的演化机制。

此后，在对网络演化机制的研究过程中，科学家们还发现真实网络中的节点通常具有局部聚类特性，即在某子图邻域内节点具有稠密的连接，而与子图外的节点具有稀疏的连接，这个称之为社团结构的特性为我们研究网络的结构和功能之间的关系提供了新的研究视角。

为了能准确有效地分析网络中的社区结构，人们提出了许多不同的社区结构检测方法。近些年来，研究者逐渐的倾向于利用人工智能技术优化模块度来找到理想的社区结构。智能优化算法是仿自然现象，对自然现象的长期观察、实践和深刻理解。如仿人类思维的智能优化算法、仿生物行为的智能优化算法和仿物理原理的智能优化算法。它们都是从随机的可行初始解出发，通过优胜劣汰策略，去逼近问题的最优解。虽然这些智能优化算法不能保证最终一定能求得问题的最优解，但是它们能够在计算复杂度和搜索精度之间达到某种平衡。到目前为止，广大研究者提出了很多的智能优化算法如：群智能算法[15]类的蚁群算法[16]、粒子群算法[17]、鱼群算法、蜂群算法等，进化算法类的遗传算法[18]、免疫算法[19]、差分进化算法[20], [21]等，本文提出的复杂网络社区检测算法是把复杂网络社区结构检测问题转化成多目标优化问题，通过一种自适应快速遗传算法[22], [23]的多个目标函数来得到全局最优解或一系列互补支配的解，这些解就对应于复杂网络的社区结构。。

1.3 本文结构安排

本文对智能算法在复杂网络社区检测的应用做了重点的研究，介绍了复杂网络社区结构和多目标的相关理论知识，并且简单地介绍了几种传统的复杂网络社区检测算法。在此基础上，对现有的遗传算法提出一种改进型遗传算法——多目标自适应快速遗传算法，通过模拟网络和真实网络进行实验仿真。本论文的结构如下：

第一章：绪论。主要介绍了复杂网络社区结构检测的研究意义和当前复杂网络社区结构划分的国内外研究现状。

第二章：复杂网络的相关理论。先介绍对复杂网络的概念和社区划分问题进行阐述，然后介绍复杂网络的研究历史，基本特性和复杂网络社区结构传统的检测方法。

第三章：人工智能优化算法。把社区划分问题作为一个多目标优化问题来进行阐述，并

且介绍了多种智能优化算法。

第四章：社区检测的多目标自适应快速遗传算法。首先介绍了多目标自适应快速遗传算法的基本思想，接着提出了自适应遗传算法和精英基因库，然后构造了社区分值和社区适应度两个目标函数，最后阐述了多目标自适应快速遗传算法的具体流程。

第五章：多目标自适应快速遗传算法的仿真。提出了算法的评价标准，将算法在模拟网络和真实网络中的实验结果与现有的优化算法进行对比。

第六章：总结与展望。对本文做了一个总结，指出了本文的研究成果和不足之处，并对未来的研究工作进行了展望。

第二章 复杂网络的相关理论

本章介绍研究工作所涉及到的理论基础内容，主要包括三部分：第一部分，复杂网络的基本概念；第二部分，复杂网络的常见特性；第三部分，复杂网络社区结构的搜索策略；第四部分，复杂网络社区结构的经典算法；最后一部分是本章小结。

2.1 复杂网络的基本概念

社团的网络形式首次出现在社会科学，它可以被建模成一个图 G =（V，E），其中 V 是一个对象的集合，称为节点，E 是一个连接的集合，用来连接节点，称为边。网络中的社区就是一个子图[7], [24]，社区也被称为社团。社区内部的边具有高密度特性，它们和外部相连的边具有低密度特性。社区的这个定义是比较抽象的，所以引进了基因位[25] k_i 的概念。$k_i=\sum_j A_{ij}$，i 和 j 为社区中的两个节点，G 的邻接矩阵表示为 A_{ij}，如果有从节点 i 到节点 j 的边，则 A_{ij} 为 1，反之则 A_{ij} 为 0。$S \subset G$，S 作为 G 的子图，子网络 S 中的节点 i 的度包含两个分量，即

$$k_i(S)=k_i^{in}(S)+k_i^{out}(S) \tag{2.1}$$

其中，

$$k_i^{in}(S)=\sum_{j\in s} A_{ij} \tag{2.2}$$

表示节点 i 连接子网络 s 中其他节点的边的条数。

$$k_i^{out}(S)=\sum_{j\notin s} A_{ij} \tag{2.3}$$

表示节点 i 连接子网络 S 外其他节点的边数。如果对于任意节点 i，子网络 S 满足

$$k_i^{in}(S)>k_i^{out}(S) \tag{2.4}$$

则称 S 为该网络的强社区结构[26]。

因此，在一个强社区结构中，社区内任意一个节点与这个社区内部其他点的连接，比它与该社区外部所有点的连接要紧密。在一个弱社区结构中，社区内部的边数之和大于社区边界上的边数之和。在本文中，我们将采取弱社区的概念，也就是说社区被定义为一组内部联

系多于不同群之间联系的节点。

2.1.1 度与度分布

（1）节点的度

节点v_i的度k_i定义为该节点连接的边数。通常情况下，一个节点的度越大，这个节点就越重要。网络中所有节点v_i的度k_i的平均值成为网络的平均度，记为$<k>$，即

$$<k>=\frac{1}{N}\sum_{i=1}^{N}k_i \tag{2.5}$$

无向无权图的节点v_i的度等于邻接矩阵的平方A^2，即

$$k_i=a_{ii}^2 \tag{2.6}$$

平均度等于A^2对角线之和除以节点数 N。式（2.7）中，$tr(A^2)$表示矩阵A^2的迹（对角线元素之和）。

$$<k>=tr(A^2)/N \tag{2.7}$$

（2）度分布

节点的分布可用分布函数$P(k)$来描述，$P(k)$表示网络中度为 k 的节点在所有节点的概率。规则对称的网络中的所有节点具有相同的度，它的度分布服从 Delta 分布，网络进行随机化，整个尖峰变宽，当网络完全随机化后，网络中的节点度分布会趋向于泊松随机分布。

研究表明，很多网络的度分布不趋向于泊松分布，而是服从幂律分布，即$P(k)\propto k^{-r}$。幂律分布也称为无标度分布[27]。因此，服从幂律分布的复杂网络也被称为无标度网络。如果对于一个任意常数a，存在常数b使得分布函数$P(k)$满足$P(ak)=bP(k)$，则必有

$$P(k)=P(1)x^{-r} \tag{2.8a}$$

$$r=-P'(1)/P(1) \tag{2.8b}$$

式（2.8a）和式（2.8b）中的$P(1)$是度为 1 的节点所占的比例。

符合幂律分布的概率分布函数同时也是唯一符合“无标度要求“的概率分布函数。在一个服从幂律分布的大规模网络中，除了少量高节点度的节点，大部分节点的度比较低，我们称之为非均匀网络，那些节点度很高的节点，我们称之为中心节点。

（3）累积度分布

除了度分布函数之外，另外一种表示度数据的方法就是累积度函数。

$$P(k)=\sum_{i=k}^{\infty}P(i) \tag{2.9}$$

它表示的是度大于等于 k 的节点的概率分布。

2.1.2 中心性

中心性[28]衡量复杂网络各节点的重要程度。在分析复杂网络的过程中，我们对节点中心性的表征有如下几种方法：度中心性、接近度中心性、介数中心性[4]等。

（1）度中心性

与度密切相关的是度中心性，它包含节点中心性和网络中心性。节点中心性是指一个节点在与其相邻节点当中的中心程度，网络中心性是指一个节点在整个复杂网络中的中心程度。节点 v_i 的度中心性 $C_D(v_i)$ 等于其度 k_i 除以 N-1，即

$$C_D(v_i)=k_i/(N-1) \tag{2.10}$$

对于含有 N 个节点的网络 G 的中心性 C_D 可以定义为

$$C_D=\frac{1}{H}\sum_{i=1}^{N}[C_D(v_{\max})-C_D(v_i)] \tag{2.11}$$

式（2.11）中 $v_{\max}$ 表示网络 G 中最大度中心性的节点，H 的值是指在所有含 N 个节点的网络中，使得式（2.12）达到的最大值。

$$H=MAX\sum_{i=1}^{N}[C_D(u_{\max})-C_D(u_i)] \tag{2.12}$$

（2）接近度中心性

接近度是衡量节点中心性的指标之一，指的是节点居于网络中心的程度。对于无向连通图，节点的接近度定义为

$$C_C(v_i)=\frac{(N-1)}{\sum_{\substack{j=1\\ j\neq i}}^{N}d_{ij}} \tag{2.13}$$

式（2.13）中 d_{ij} 为节点 v_i 到 v_j 的距离。节点的接近度越大，则节点和网络中心越近，它在网络的作用越重要。

网络 G 的接近度为

$$C_C = \frac{2N-3}{(N-1)(N-2)} \sum_{i=1}^{N} [C_C(v_{\max}) - C_C(v_i)] \tag{2.14}$$

（3）介数中心性

节点的介数是指网络中所有的最短路径中经过该节点的数量比例。节点 v_i 的介数中心性定义为节点 v_i 的归一化介数。设节点 v_i 的介数 B_i，该节点的介数中心性为

$$C_B(v_i) = \frac{2B_i}{(N-1)(N-2)} \tag{2.15}$$

网络 G 的介数中心性可表示为

$$C_B = \frac{1}{N-1} \sum_{i=1}^{N} [C_B(v_{\max}) - C_B(v_i)] \tag{2.16}$$

2.2 复杂网络的特性

复杂网络的结构具有以下几大特性：复杂性、小世界性[27]、无标度性和超家族性。

2.2.1 复杂性

（1）复杂网络的规模比较巨大。复杂网络的节点规模可以有成千上万，但超大规模的复杂网络具有一定的数学概率统计特性。

（2）节点和连接结构的复杂性。首先，各节点的复杂性表现为节点复杂的动力学特性，即各个节点本身可以是各种线性或者非线性系统[29]。其次，它的复杂性表现为节点的多样性，复杂网络的节点可以代表任何事物，并且一个复杂网络可能会出现各种不同类型的节点。网络连接结构既非完全随机也非完全有规则性，但其内部具有内在的自组织规律，网络结构可呈现多种多样的特性。

（3）多重因素融合在网络时空演化过程中的复杂性。复杂网络具有时间和空间的演化复杂性，可展示其比较多样的复杂变化行为，尤其是网络节点之间的不相关的运动。若多重复杂因素互相作用，则将产生更为难以预料的结果。比如，设计一个通信网络需要考虑此网络的更新过程，这在一定程度上决定了网络的拓扑结构。当两个节点之间进行数据传输时，它们的连接权重也会增大，需要通过不断的优化，才能够改善网络的性能。

（4）复杂网络连接的稀疏性。一个包含 N 个节点且具有全耦合结构的复杂网络的连接

数为 $O(N^2)$，而现实中的网络的连接数目为 $O(N)$。

2.2.2 小世界特性

小世界特性是指复杂网络虽然规模比较大，但是任意两点之间存在一条长度较短的路径。

（1）特征路径长度

在复杂网络中，d_{ij} 代表两个节点 i 和 j 之间的最短路径距离，具体指是指从 i 到 j 所经过的最少边数。网络中任意两个节点间的距离最大值称为网络的直径 D。网络的特征路径长度 L 表示为网络中所有节点对之间的最短路径的平均长度，即

$$L=\frac{1}{N(N-1)}\sum_{i\neq j}d_{ij} \tag{2.17}$$

复杂网络的特征路径长度 L 值衡量了网络的传输效率和性能。

（2）聚类系数

一个节点的聚类系数指的是与该节点相连的两个节点也存在连接的概率，用来描述网络中节点的聚集情况，即网络有多紧密。若 k_i 为第 i 个节点的度，则由节点 i 的 k_i 个相邻节点构成的子网中，实际存在的边数 $|H_i|$ 与全部 k_i 个节点完全连接的总边数 $(k_i\text{-}1)/2$ 的比值定义为节点 i 的聚类系数 B_i

$$B_i=\frac{2|H_i|}{k_i(k_i-1)} \tag{2.18}$$

整个网络的聚类系数等于所有节点的聚类系数的平均值，即

$$B=\frac{1}{N}\sum_{i=1}^{N}B_i \tag{2.19}$$

通常，我们用网络的特征路径长度 L 和聚类系数 B 来描述小世界网络的数字特征。小世界网络具有大的聚类系数和小的特征路径长度，将其合起来称为复杂网络的“小世界特性”。经科学研究表明，现实世界中的很多复杂网络都具有上述的“小世界特性”。

2.2.3 无标度特性

刻画一个节点的特性，就要用到度的概念。一个节点 i 的度 k 定义为与它相连接的边的数目，即与节点相连接的节点的个数。一个节点的度越大，那么它在网络中的重要性越高。节点度分布(Degree Distribution)可用一个分布函数 $P(k)$来刻画，表示一个随机选定的节点的度

为 k 的概率，或者等价地描述为网络中度为 k 的节点数占网络总节点数的比例。

人们发现一些复杂网络的节点的度分布具有幂指数函数的规律。因为幂指数函数在双对数坐标中是一条直线，这个分布与系统特征长度无关，所以该特性被称为无标度性质。无标度特性反映了网络中节点度的不均匀分布，其实只有少数的节点与周围节点有较多的连接，成为中心节点，大多数节点的度非常小。

2.2.4 超家族特性

有些不同类型的网络特性在一定的外界条件下具有相似性。尽管网络类型不同，但是只要组成网络的基本单元，即最小子图相同，它们拓扑性质的重大轮廓外形就可能具有一定的相似性，这种性质被称为复杂网络的超家族特性[3]。不同网络存在与某个类型网络的特性相似，归根到底在于它们具有相似的网络“基因”。

目前对于复杂网络的超家族特性的相关性工作上，不管在理论研究上，还是技术领域的发展上都有待进一步投入。

2.3 复杂网络的搜索策略

在复杂社区网络中，搜索有着重要的实际应用，在实际搜索中，我们可以使用一些搜索策略和搜索算法在复杂网络中寻找一个特定节点。一个节点能否寻找到它和网络中其他节点的最短路径在于这个节点掌握的网络结构信息、搜索策略和它所使用的搜索算法。本节主要介绍广度优先搜索、最大度搜索和随机游走搜索。

2.3.1 广度优先搜索

广度优先搜索[30]也称广播搜索。广度优先搜索的基本思想是从原节点 s 出发，逐步查找邻居节点，直到查到目标节点为止。由于访问邻接节点是逐步并行的，可设第 i 步访问的节点集合为V_i，已访问节点集合为 U，广度优先搜索的流程如下：

（1）若源节点是目标节点，则搜索停止，步数记为 0，否则$V_1=\{s\},U=\{s\}$；

（2）假定已访问了 i 步，得到第 i 步访问的节点集合为V_i，则对每个V_i中的节点求邻居节点，放入初始化为空的V_{i+1}'集合；

（3）对V_{i+1}'集合中的元素进行逐个分析，看是否在 U 中，如果集合中某个节点元素已经

被访问过，则集合V_{i+1}'移除这个元素，得到未访问节点集合V_{i+1}，再更新 U。

（4）查看集合V_{i+1}是否包含目标节点。若有，搜索停止；反之，则 i=i+1，返回步骤（2）。

广度优先搜索可以寻找网络中任意两点之间的最短路径，由于实现并行搜索；搜索范围以指数速度增大，因此，搜索速度非常快。但是这种搜索方式会消耗大量网络流量，导致网络拥塞。为了减小广度优先搜索带来的大量流量消耗，研究者提出了本地索引、迭代加深等改进方法。

2.3.2 最大度搜索

最大度搜索策略最早由 Adamic 等人提出[25]，在网络中使用最大度搜索策略的搜索过程如下：源节点 s 首先查询最大度的邻接节点，如果此邻接节点存储了目标文件，它将选择最大度的邻接节点将查询信息传递下去，直到找到目标节点为止。

在搜索过程中，节点有多个度相同的最大度邻接节点，则在这些最大度节点中随机选择一个。为了避免在查找过程中出现死循环，规定一个节点可以多次被访问，但是同一条边智能被访问一次。如果与节点相连的边都被遍历过，那么就返回至前个节点。

引入一个 $N\times N$ 矩阵 B，b_{ij} 为其中的元素，用于判定边是否被访问过。

$$b_{ij}=\begin{cases}0 & \text{节点}i\text{和}j\text{之间边没访问，}\\ 1 & \text{节点}i\text{和}j\text{之间的边已经访问，}\\ \infty & \text{节点}i\text{和}j\text{之间不存在边。}\end{cases} \tag{2.20}$$

2.3.3 随机游走搜索策略

随机游走策略搜索是复杂网络中一种比较常用的动态搜索策略。在规则网络和随机网络中，随机游走策略被广泛应用。近年来，无标度网络[10]和小世界网络的随机游走[31]也引起了广大的关注。

当一个节点只知道自己的邻接节点信息，则在网络中寻找目的节点时，可用随机游走搜索策略。随机游走搜索策略可以细分成三种不同的策略：

（1）无限制随机游走（Unrestricted Random Walk，URW）搜索策略：URW 是最常见的随机游走。每一步搜索中，当前节点不加任何限制地在其所有邻居中随机选择一个邻居节点将查询消息传递过去，直到搜索到目标节点的任一个邻居为止。

（2）不返回上一步节点的随机游走(No Retracing Random Walk，NRRW)搜索策略：将查

询传递过来的上一步节点之外，每一步中，当前节点在其余的所有邻居中随机选择一个邻居节点将查询消息传递过去，直到搜索到目标节点的任一个邻居为止。

（3）不重复访问节点的随机游走(Self-avoiding Random Walk，SARW)搜索策略：已经被查询过的节点不允许再被查询，且每一步中当前节点在其所有未被查询过的邻居中随机选择一个邻居节点将查询传递过去，直到搜索到目标节点的任一个邻居为止。

2.4 复杂网络社区检测的经典方法

复杂网络的一个非常重要问题就是社区检测算法的应用。一些研究人员在研究复杂网络的社区检测问题上，提出了比较富有启发性的算法。

在复杂网络社区检测算法中，一种早期的检测方法，就是网络分割方法。网络分割方法是起源于图论里的图形分割理论，它把网络分割成相对均匀独立的单元。它的基本准则是把复杂网络中的 M 个节点划分到 N 个分组中，每组之间边的数量越小越好，每组的节点个数大致一样。所以，当复杂网络的规模很大时，此方法的性能比较差。但是有很多演进型算法在很多情况下会得到比较好的结果。其中，基于 Laplace 图特征的谱平分法和 Kernighan-Lin 算法比较有名。这两种方法适合用于复杂网络的社区结构分析，但是很难对复杂网络进行一个全局的分析，它没法科学合理地去确定复杂网络分解的组数。

复杂网络社区检测的另外一种比较常见的方法，就是层次聚类法[32]（Hierarchical Clustering，UC）。它遵循同类相似原则即根据各个节点之间的相近性，把复杂网络划分成多个子模块。根据在网络中是移除边还是添加边，层次聚类法具体分为分裂算法（Divisive Method，DM）[32]和凝聚算法[33]（Agglomerative Method，AM）两类。一般情况下，层次聚类法能够对中心节点（和其他节点连接数较多且处在网络中处于中心枢纽位置的节点）进行准确的划分，但是对于外围节点（与中心节点相对应的节点）容易出现误分[34]。

分裂算法的基本思想是从关注的网络开始，尽力去寻找与自己相似性最低的相邻节点，然后删除和它相连的边。周而复始，就把整个网络分成很多个小型的子模块。在分裂过程中，可以在任何时候终止分裂活动，该状态下的网络就是若干个社团子网络的根网络。整个分裂算法的流程可以利用树状图表示，底部的圆代表各个节点，如图 2.1 所示。利用树状图可以直观地展现了复杂社团网络从一个大网络逐步分解成子网络的连续变化过程。

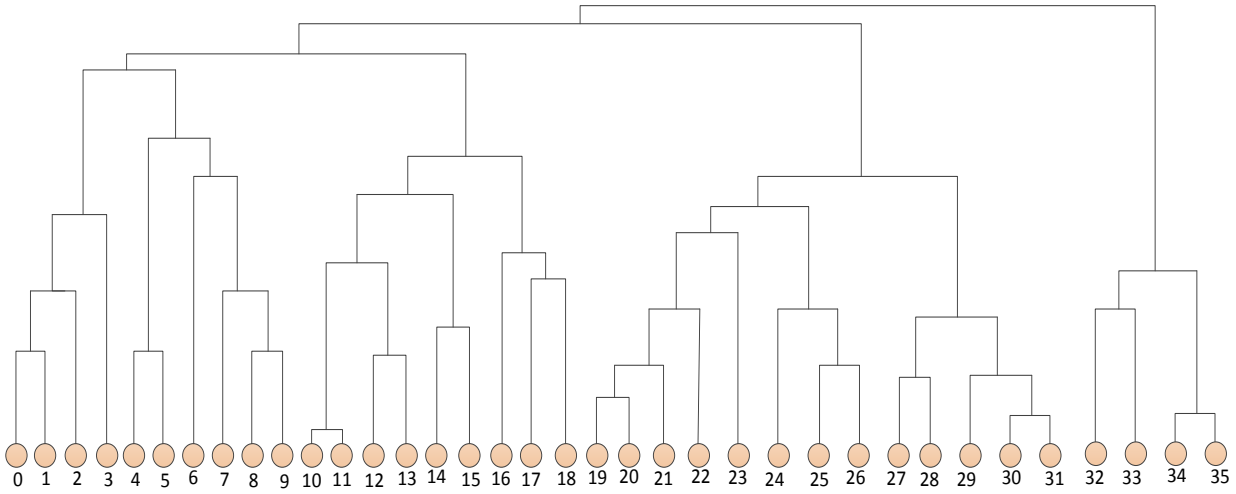


图 2.1 采用的树状图记录层次聚类算法的划分结果

相反地，在凝聚算法中，先计算出各节点对之间的相似性，然后从相似性最高的节点对开始，往一个原始网络中添加边。从空图凝聚到最终图的流程用树状图表示，如图 2.1 所示，当该树状图在任何位置停止凝聚时，就对应一种网络社团划分结果。

2.4.1 GN 算法

GN 算法[24]由 Girvan 和 Newman 在 2002 年提出的一种基于边介移除的分裂算法。它对复杂网络进行社区检测采用了反复识别和删除社区间连接的策略，根据网络中的边不属于社区的程度逐步把不属于任何社团的边移除，即把连接各个社区之间的边删除，一直到这类边全部移除为止。边介数：网络中经过该边的任意两点间最短路径的条数。通常，我们用它作为指标区分一个社区间的连通程度。

GN 算法的核心思想：通过不断地从网络中移除边介数最大的边，将目标复杂网络分解成各个社区。GN 算法的执行过程如下：

（1）计算复杂网络所有边的边介数。

（2）寻找边介数最高的边，并且把它从网络中删除，然后再重新计算网络中剩余边的边介数。

（3）重复步骤（2），直到网络中所有的边均被删除。

在算法的执行过程中，某条边被移除后一定要重新计算剩余边的边介数。因为当边介数最大的边被删除后，网络拓扑结构也会发生变化。原先计算出来的边介数并不能反映当前网络拓扑结构边介数的情况。对于 N 个节点和 M 条边的网络来说，采用广度优先方法来计算一个节点到其他节点的边界数最多耗时 $O(MN)$，每次删除后重新计算边介数，所以在最坏的情

况下，GN 算法总的时间复杂度 $O(M^2N)$。对于社区结构较强的复杂网络，该算法往往能够很快分出几个独立社区。

GN 算法的能够较好的识别出网络的社区，但是由于边介数的计算开销过大，它的计算速度慢，具有很高的时间复杂性，因此，只适合检测中小规模的社区网络。

2.4.2 Newman 快速算法

在上一小节已经说过，GN 算法虽然在复杂网络社区结构的检测中能够较好的进行检测，但是它的算法复杂度比较大，只能适用于检测中小规模的复杂网络。为了进一步满足大规模和超大规模复杂网络的研究需求，Newman 在 GN 算法的基础上提出了一种新的社区网络检测算法——Newman 快速算法[35]。

与分裂算法不同，Newman 快速算法是一种自底向上的凝聚算法，一个节点单独占据一个社区，然后计算节点的相似度，使得节点沿着模块性[3]增加最大的方向合并社团，把相似度最高的两个节点挑选出，然后连接这两个节点，就将这两个节点合成一个社团，逐渐合成一个大的社区网络，从而获得社区划分结果。

检测具有 N 个节点的复杂社区网络，Newman 快速算法的执行过程步骤如下：

Step 1：初始化网络为 N 个社区，每个节点是一个社区。

$$e_{ij}=\begin{cases}1/(2M), \text{如果节点i和j之间有边,}\\ 0, \qquad \text{其他，}\end{cases} \tag{2.21}$$

$$a_i=k_i/(2M) \tag{2.22}$$

式(2.21)，(2.22)中，k_i 为节点 i 的度，M 为复杂网络中的边的数目。

Step 2：依次合并相邻的社区，计算合并后的模块度[3]增量。使得每次合并沿着使模块度增大最多或减小最少的方向来进行。每次合并后，对 e_{ij} 进行更新。

Step 3：重复执行 Step2，不断合并社团，一直到整个复杂网络都合并成一个社团。这里最多需要 N-1 次合并。

Newman 算法总的算法复杂度 $O((M+N)N)$，对于稀疏网络则为 $O(N^2)$。这个算法完成后可以得到一个社团结构分解的树状图，可以通过在不同的位置断开连接得到不同的网络社团结构。

2.5 本章小结

本章首先介绍了复杂网络社区结构的基本概念，如复杂网络结构表示方法、度、度分布和中心性等基本概念，随后阐述了复杂社区网络的四大基本特性，接着分析了复杂社区网络中的三大搜索策略，最后叙述了复杂社区网络传统的经典算法。其中，本章重点介绍了 GN 分裂算法和 Newman 快速算法。

第三章 人工智能优化算法

在日常的工程实践领域中，我们定义只有一个目标函数的优化问题为单目标优化问题，存在多个目标函数的问题为多目标优化问题。在复杂网络社区检测算法的研究中，我们可以把它看做一个多目标优化问题，通过构造两个目标函数，然后对这两个目标函数进行优化折衷来划分社区结构。

3.1 多目标优化问题

关于多目标优化问题[36]的文字描述：D 个决策变量参数、N 个目标函数、$m+k$ 个约束条件组成一个优化问题，决策变量与目标函数、约束条件是函数关系。在非劣解集中决策者只能根据具体问题要求选择令其满意的一个非劣解作为最终解。多目标优化问题的数学形式可以描述为：

$$\text{Min} \quad y = f(x) = [f_1(x), f_2(x), \ldots, f_n(x)] \tag{3.1}$$

n=1，2，…，N

$$\text{s.t.} \quad h_i(x) \le 0 \qquad i=1，2，\cdots，m$$

$$g_j(x)=0 \quad j=1，2，\cdots，k$$

$$x = [x_1, x_2, \ldots, x_D]$$

$$x_{d\_\min} \le x_d \le x_{d\_\max} \qquad d=1，2，\cdots，D$$

在式（3.1）中，x 为 D 维决策向量，y 为目标向量，N 为优化目标总数；$h_i(x) \le 0$ 为第 i 个不等式约束，$g_j(x)=0$ 为第 j 个等式约束，$f_n(x)$ 为第 n 个目标函数；X 是决策向量形成的决定空间，Y 是目标向量形成的目标空间。$h_i(x) \le 0$ 和 $g_j(x)=0$ 确定了解的可行域，$x_{d\_\max}$ 和 $x_{d\_\min}$ 为每维向量搜索的上下限。关于多目标优化问题中最优解或非劣最优解定义如下：

定义 1 对任意的 $d \in [1，D]$ 满足 $x_d^* \le x_d$ 且存在 $d_0 \in [1，D]$ 有 $x_{d_0}^* < x_{d_0}$，则向量 $x^* = [x_1^*, x_2^*, \ldots, x_D^*]$ 支配向量 $x = [x_1, x_2, \ldots, x_D]$。

$f(x^*)$ 支配 $f(x)$ 必须满足以下两个条件：

(1) $\forall n, f_n(x^*) \leq f_n(x)$ n=1，2，…，N

(2) $\exists n_0, f_{n_0}(x^*) < f_{n_0}(x)$ $1 \leq n_0 \leq N$

$f(x)$的支配关系与 x 的支配关系是一致的。

定义 2 Pareto 最优解是不被可行解集中的任何解支配的解，若 x^* 是搜索空间中一点，说 x^* 为非劣最优解，当且仅当不存在 x（在搜索空间可行性域中）使得 $f_n(x) \leq f_n(x^*)$ 成立，n=1，2，…，N。

定义 3 给定一个多目标优化问题 $f(x)$，$f(x^*)$是全局最优解当且仅当对任意 x(在搜索空间中)，都有 $f(x^*) \leq f(x)$。

定义 4 由所有非劣最优解组成的集合称为多目标优化问题的最优解集，也称为可接受解集或有效解集，最优解集对应的目标函数值形成的区域称为 Pareto 前端。

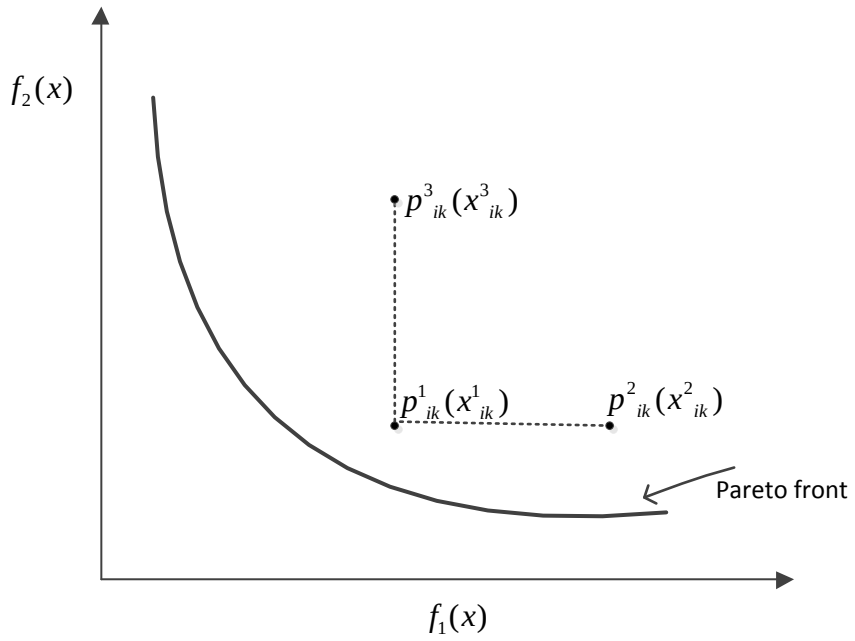


图 3.1 目标函数空间

根据支配解的定义以及目标函数空间图，如图 3.1 所示，可以得出 p^3_{ik} 和 p^2_{ik} 不存在支配关系，即 p^2_{ik} 不支配 p^3_{ik}，p^3_{ik} 也不支配 p^2_{ik}；根据图 3.1 可以得出结论，p^1_{ik} 支配 p^3_{ik} 和 p^2_{ik}，它比 p^3_{ik} 和 p^2_{ik} 更符合要求。因此，在这三个解中，p^1_{ik} 是最优解。

3.2 人工智能算法

人工智能算法，是一种人们受到自然界规律的启迪，根据规律的内在原理和外在表现形式，对其建模仿真，从而对问题进行求解的优化算法。人工智能算法包含了很多算法，如模拟退火算法、粒子群算法、遗传算法、蜂群算法等等。人工智能算法通常解决的是最优化问题，它是以数学为基础，用于实际工程中遇到的近似最优解求解问题。在实际工程领域存在的优化问题，很难找到准确的数学公式，只能寻找近似最优解。因此，人工智能优化算法已经成为目前的研究热点。

3.2.1 模拟退火算法

介绍模拟退火算法[37]前，先简单介绍爬山算法[38]。爬山算法是一种简单的贪心搜索算法[39]，该算法每次从当前解的临近解空间中选择一个最优解作为当前解，直到达到一个局部最优解。爬山法是一种典型的贪心法，每次都局限于选择一个当前最优解，因此只能搜索到局部的最优值。

模拟退火算法（Simulated Annealing，SA）来源于固体退火原理，它其实也是一种贪心算法，但它在搜索过程引入了随机因素。它是一种求解大规模组合优化问题的随机性方法，它以优化问题的求解与物理系统退火过程的相似性作为理论基础，利用 Metropolis 算法[40]并且适当地控制温度的下降过程实现模拟退火，用固体退火模拟目标优化问题，将内能 E 模拟为目标函数值 f，温度 T 演化成控制参数 t，即得到解组合优化问题的模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃“的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表控制，包括控制参数的初值 t 及其衰减因子 Δt、每个 t 值时的迭代次数 L 和停止条件 S。

模拟退火算法主要可以分解为初始解、目标函数、解空间三部分，图 3.2 是模拟退火算法流程图，它的具体描述如下：

（1）从可行解空间中任选一初始状态 x_0，计算其目标函数值 $f(x_0)$，并选择初始控制温度 T_0 和马尔可夫链的长度；

（2）在可行解空间中产生一个随机扰动，用状态产生函数产生一个新状态 x_1，计算其目

标函数值 $f(x_1)$；

（3）根据状态接受函数判断是否接受：如果 $f(x_1)<f(x_0)$，则接受新状态 x_1 为当前状态，否则按 Metropolis 准则判决是否接受 x_1，若接受，则令当前状态等于 x_1，若不接受，则令当前状态等于 x_0；

（4）根据某个收敛准则，判断抽样过程是否终止，是则转 5，否则转 2；

（5）按照某个温度冷却方案降低控制温度 T；

（6）根据某个收敛准则，判断退火过程是否终止，是则转 7，否则转 2；

（7）当前解作为最优解输出。

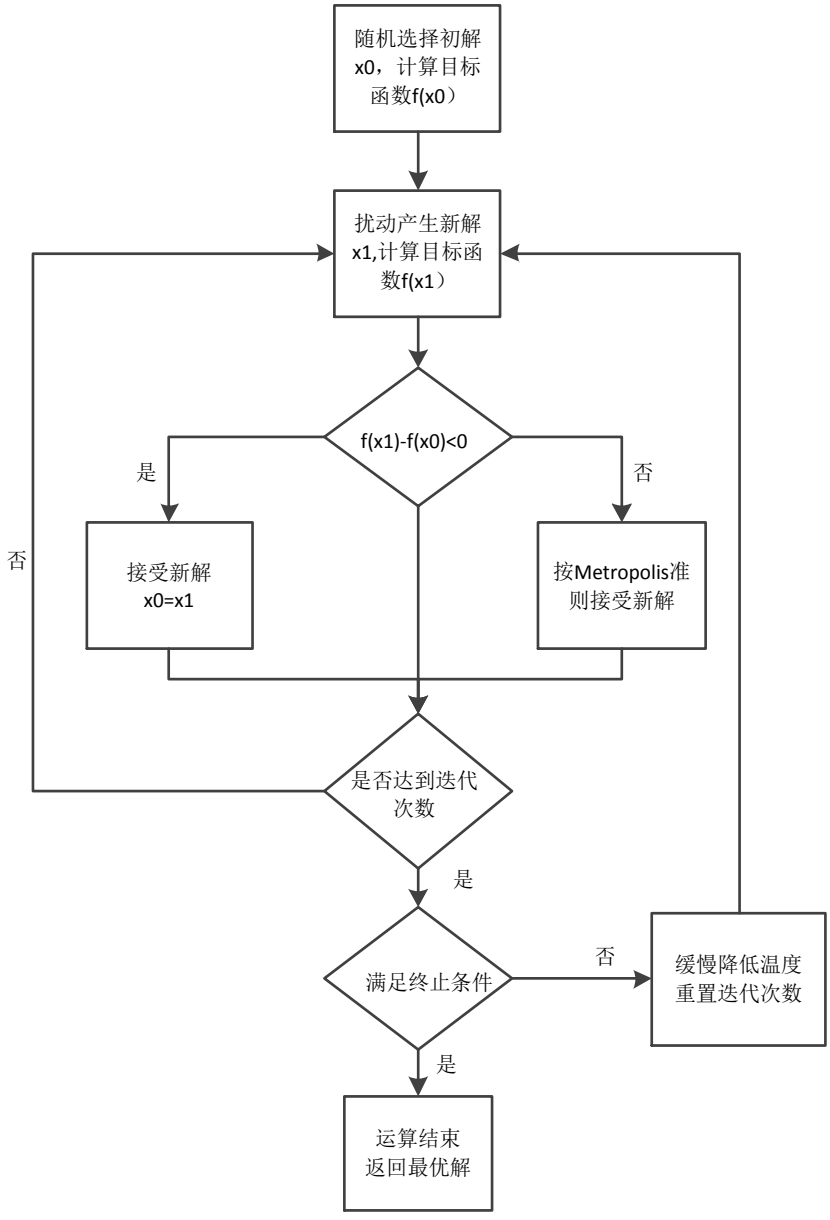


图 3.2 模拟退火算法流程图

模拟退火算法是一种随机算法，虽然不一定能找到全局的最优解，但可以比较快的找到问题的近似最优解。它适用于并行处理，可用于求解复杂的非线性优化问题。 如果参数设置得当，模拟退火算法搜索效率比穷举法要高。但是它存在收敛速度慢、参数敏感以及算法性能与初始值有关等缺点。

3.2.2 粒子群算法

(1) 算法原理和数学描述

Kennedy 和 Eberhart 在 1995 年提出了一种基于群体的演化算法 —— 粒子群算法（Particle Swarm Optimization，PSO）。它的思想来源于人工生命理论和演化计算科学理论，这种智能算法源于对鸟类捕食行为的研究而提出的。一群鸟在随机搜寻自己的食物，如果在这个区域里只有一块食物，那么最高效的方法策略就是搜寻当前离食物最近的鸟的周围区域。虽然鸟只是追踪它的邻居，但是最终的结果是整个鸟群的控制在一个中心中。PSO 优化算法就是从上述模型中得到的启示发展过来的。

在 PSO 算法中，每个优化问题的潜在解都是搜索空间中的一个点，称之为“粒子”。粒子在搜索空间以一定速度飞行，并且根据自己和同伴的飞行经验动态的调整飞行速度。所有的粒子都含有一个由目标函数决定的适应值，并且，每个粒子都知道目前群体发现的最好位置。然后各个粒子就按照当前的最好粒子在解空间中进行搜索。每个粒子按照以下信息调整自己的位置：①当前速度和当前位置；②自己最好位置和当前位置的距离；③群体最好位置和当前位置的距离。

粒子群算法需要随机初始化一个粒子群——随机解，然后通过迭代找到最优解，在每一次迭代中，粒子通过跟踪比较两个“极值点”来更新自己。每个粒子有位置 $x_i=(x_{i1},x_{i2},...,x_{ik})^T$ ，速度 $v_i=(v_{i1},v_{i2},...v_{ik})^T$ 。速度和位置的更新公式

$$v_{ik}^{t+1}=\omega\cdot v_{ik}^{t}+c_1 r\cdot(pbest_{ik}^{t}-x_{ik}^{t})+c_2 R\cdot(gbest^{t}-x_{ik}^{t}) \tag{3.2}$$

$$v_{ik}^{t+1}=x_{ik}^{t}+v_{ik}^{t+1} \tag{3.3}$$

其中 v_{ik}^{t} 是粒子 i 在第 t 次的迭代中第 k 维度的速度；r 和 R 是[0，1]之间的随机数；c_1 和 c_2 属于加速系数； $pbest_{ik}^{t}$ 是粒子 i 在第 t 次迭代时第 k 维的个体极值点的位置； $gbest^{t}$ 是第 t 次迭代时的全局极值点的位置。每个粒子每一维的速度都位于 $[-v_{\max},+v_{\max}]$ 中。

（2）算法流程

粒子群优化算法的基本流程如下：

Step1：对粒子群的速度和位置进行随机设定；

Step2：计算每个粒子的适应度值；

Step3：对每个粒子的适应度值和其飞过的最好位置 *pbest* 进行比较，若比 *pbest* 好，则取而代之；

Step4：将其适应度和 *gbest* 比较，若更好，则用它替代 *gbest*；

Step5：每个粒子按照公式（3.2）和公式（3.3）不断更新自己的位置和速度，每一代之后产生新的位置和速度；

Step6：如果没有达到条件，则返回 Step2，反之，则寻优过程结束。

PSO 算法流程图如图 3.3 所示。

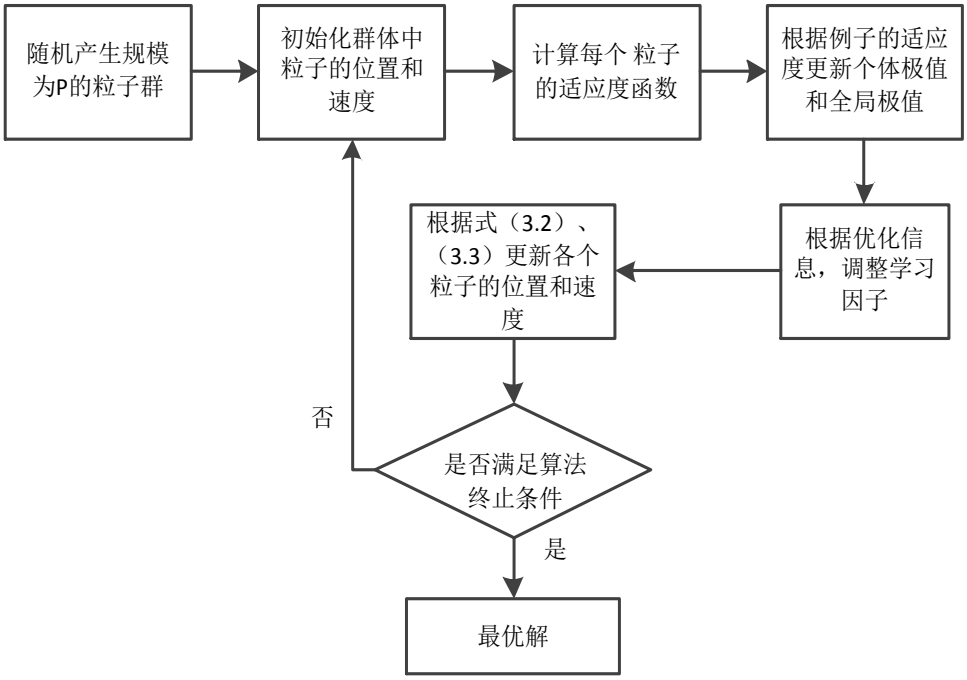


图 3.3 粒子群算法基本流程图

3.2.3 基本遗传算法

遗传算法[41], [42]（Genetic Algorithm，GA）是基于生物进化过程中的自然选择机制而解决实际问题中的一种过程搜索最优解算法。遗传算法在进化中不断的更新潜在解组成的种群。在算法迭代的每一步中，算法在当前种群中随机的选择一些优秀个体作为父代，然后让这些父代产生一些子代。经过连续的迭代运行，种群将向最优解的方向进化。我们可以使用遗传

算法解决目标函数是离散不可微的、非线性的随机问题。它不同于网络分割算法、层次聚类算法、搜索算法等传统优化算法，主要优点有：

（1）GA 按并行方式搜索一个种群数目的点，而不是单点；

（2）GA 不需要求导或其他辅助知识，只需要适应度函数值；

（3）GA 直接处理问题参数的适当编码而不是参数集本身 ；

（4）GA 在搜索过程中不易陷入局部最优，有较好的全局优化能力。

传统 GA 采用二进制编码方式，使用长度固定的二进制符号串来表示群体中的个体，个体的等位基因由二值符号集{0，1}组成。GA 算法中，个体的选择操作采用轮盘赌选择方法，交叉操作采用单点交叉，变异操作采用基本位变异，交叉概率和交叉概率都是一个固定值。

在基本遗传算法进化过程中，需要设定下面几个参数：

（1）N ：群体大小，种群规模，一般取为 20 ～ 100；

（2）$G_{\max}$ ：遗传算法的终止进化代数，一般取为 100 ～500；

（3） p_c ：交叉概率，一般取为 0.5 ~ 1；

（4） p_m ：变异概率，一般取为 0.001 ~ 0.1。

上述几个运行参数在一定程度上影响了 GA 算法的性能。GA 算法的实际应用过程中，需要多次尝试才能够确定这几个参数的取值大小。

算法流程描述:

Step1：确定需要求解问题的目标函数，并将此目标函数转化为适应度函数；

Step2：编码：将解空间的解数据表示为遗传空间的基因型；

Step3：种群初始化随机产生 N 个初始串，遗传算法以这 N 个串结构迭代起点；

设当前进化代数设为 s=0，最大进化代数为$G_{\max}$ ；

Step4：根据目标适应度函数，计算每个种群中每个个体的适应度值；

Step5：选择:使用轮盘赌选择法；

Step6：交叉：使用单点交叉算子；

Step7：变异：使用基本位变异算子；

父代种群经过上述运算得到子代种群；

Step8：是否达到迭代终止条件：如果 s ＜$G_{\max}$，则 s = s+1，转到 Step4；若 t ≥ $G_{\max}$，则输出最大适应度的个体作为最优解，停止运算。

算法的流程图如图 3.4 所示。

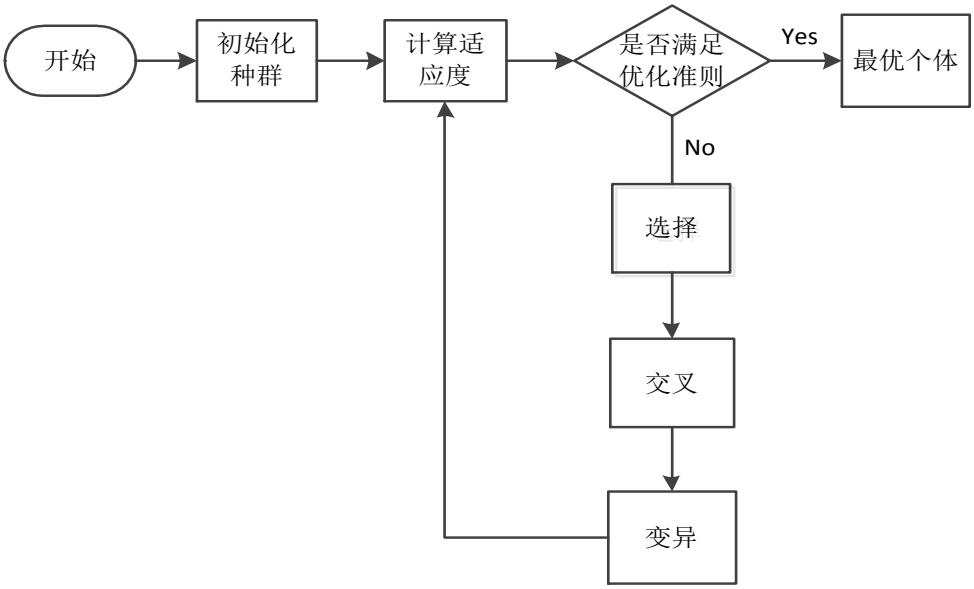


图 3.4 遗传算法流程图

3.3 本章小结

本章首先阐述了多目标问题及其基本概念，然后介绍了几种常见的智能优化算法：模拟退火算法、粒子群算法、基本遗传算法，为下一章节的多目标自适应快速遗传算法做了个铺垫。

第四章 社区检测的多目标自适应快速遗传算法

第三章介绍了传统基本遗传算法，但是它在应用方面是存在着一些不足。由于传统遗传算法采用策略参数固定的方法，无法满足在遗传进化中对这些策略参数动态与变化的要求，尤其是交叉概率和变异概率，所以其寻优效果不大理想。从生物进化的角度看，传统遗传算法虽然考虑到了种群对环境的适应能力的模拟，但却忽略了种群跟随环境进化时，遗传行为和个体生长随之变化的自适应特性，这是影响传统遗传算法性能和效率的根本原因。因此，本章针对传统遗传算法的不足，采用自适应交叉概率[43]和变异概率，并且在此基础上引入一个精英基因库，用于存储适应度较高的 Pareto 非劣解，提出了一种多目标自适应快速遗传算法(Multi-objective Adaptive Genetic Algorithms，A-MOGA）。

4.1 自适应快速遗传算法原理

在种群的进化早期，自适应快速遗传算法的遗传算子应该进行大规模搜索，以免过早收敛，出现“早熟”现象 。在种群进化后期，种群应该进行局部搜索，调整进化策略进行重点进化。本文的自适应快速遗传算法的改进主要包含以下两点：

（1）根据个体的适应度值和相似系数设计 A-MOGA 的自适应交叉率和变异率的调节公式，提高算法的寻优能力。

（2）引入精英基因库用于存放遗传算法进化过程中出现适应度较大的个体，当迭代过程中，对于精英基因库已经存在的重复个体， 可以不用再重复解码，计算个体的适应度值等一系列过程，提高了算法的效率。

4.1.1 自适应遗传算法

针对遗传算法的参数动态变化问题，Srinivas 等人提出了一种根据适应度值动态调整交叉概率和变异概率的自适应遗传算法。

（1）Logistic 曲线方程

Logistic 函数[44]是一种常见的 S 形函数，它是比利时科学家皮埃尔·弗朗索瓦·韦吕勒在 1844 年在研究它与人口增长的关系时命名的。广义 Logistic 曲线可以模仿一些情况如人口增长的 S 形曲线。起初阶段大致是指数增长，然后随着开始变得饱和，增加变慢，最后达到成熟时增加停止。

Logistic 曲线方程的微分形式为:

$$\frac{dN}{dt} = rN(1-\frac{N}{K}) \tag{4.1}$$

Logistic 曲线方程的积分形式为

$$N = \frac{K}{1+e^{a-rt}} \tag{4.2}$$

N 表示生长量(Growth)、生物量(Biomass)、或其他生物数量指标(如发病数等)；r 是常数，表示瞬时增长率(Instantaneous Growth Rate)或者自然增长率；t 表示时间(或温度等)值；K 也是常数，称为环境负载能力(carrying capacity)； a 为积分常数。式（4.2）即为 S 形 Logistic 累计分布曲线方程[44]。

在 Logistic 曲线积分方程的基础上，令 a=0，r =1，K =1， $N = P(t)$ ，t=±t 则得到两个演进后的 Logistic 函数方程:

$$P_1(t) = \frac{1}{1+e^{-t}} \tag{4.3}$$

$$P_2(t) = \frac{1}{1+e^{t}} \tag{4.4}$$

当一个物种迁入到一个新生态系统中后，其数量会发生变化。假设该物种的起始数量小于环境的最大容纳量，则数量会增长。该物种在此生态系统中有天敌、食物、空间等资源也不足（非理想环境），则增长函数满足式（4.3）的 Logistic 方程表达式，图像呈 S 形，如图 4.1 所示，此方程是描述在资源有限的条件下种群增长规律的一个最佳数学模型。

根据图 4.1 中式（4.3）中 $P_1(t)$ 的曲线图，可以更加直观地看出式（4.3）的函数变化趋势：随着 t 取值的增大，函数值越来越大，但是，当 $t \in (0,\infty)$，函数值增大的同时，函数的斜率越来越小，函数值是趋近于一个固定值 1。而根据图 4.1 中式（4.4）中 $P_2(t)$ 的曲线图，可以看出式（4.4）的函数变化趋势：随着 t 取值的增大，函数值是越来越小，但是，当 $t \in (0,\infty)$ 时，函数值继续减小的同时，函数的斜率也越来越小，函数值是趋近于一个固定值 0。

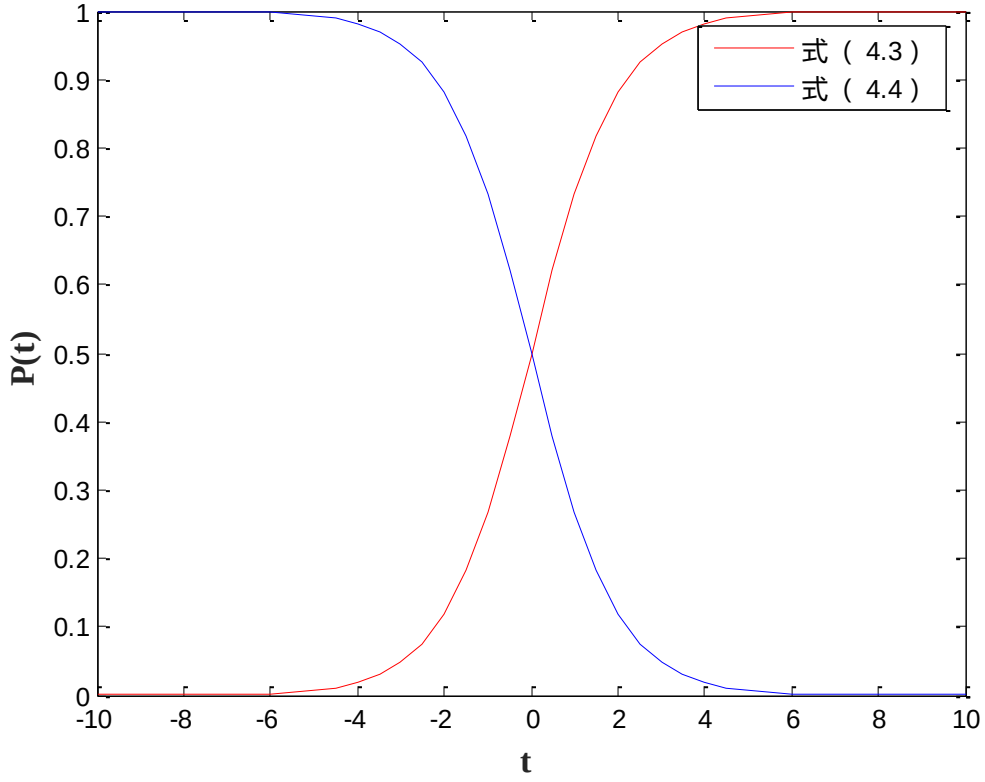


图 4.1 Logistic 函数方程式（4.3）、（4.4）曲线图

（2）自适应交叉概率和变异概率

自适应遗传算法中交叉概率和变异概率调节公式设计的依据是：对于适应度较差的种群个体应该给予较小的变异概率和较大的交叉概率，对于适应度较好的种群个体则依据其个体的优良等级和种群迭代状态给予此个体相应的交叉概率和变异概率，当迭代次数越接近最大迭代次数，个体交叉概率就越小而变异概率就越大，这样会使种群进化不会陷于一种停滞不前的状态。

在本文交叉概率和变异概率公式的设计过程中，体现了一种新的设计思路：在每次迭代过程中，当个体相似度较小，则个体之间的适应度值差异较大，则说明此种群的基因类型种类比较丰富，因此，给予较大的交叉概率和较小的变异概率；如果个体相似度较大，种群个体之间的适应度值差异很小，则说明种群的基因类型种类较少，因此，则给予较小的交叉概率和较大的变异概率。

按照式（4.3）、式（4.4）的函数曲线的特性，通过一些系数的调整，可以把该曲线的函数表达式改进为交叉概率 p_c 和变异概率 p_m 的调节公式，随着遗传算法进化代数的不断增加，种群中个体的适应度越来越大，重复个体也越来越多，种群的搜索空间的范围因此逐渐减少，最优解收敛半径越来越小，所以种群适应度的差异也越来越小。遗传算法交叉概率和变异在

实际应用中的准则主要有如下两点：

1、在 $p_c \in$（0.5，1），$p_m \in$（0，0.1）。

2、个体越趋向于收敛，交叉概率越来越小，而变异概率越来越大。

为了能够遵循上述的设计原则，解空间能够更好有效地进行搜索，文中引入了概率统计理论中的标准差[45]和相似参数概念，标准差是一组数据平均值分散程度的一种度量。一个较大的标准差，代表大部分数值和其平均值之间差异较大；一个较小的标准差，代表这些数值较接近平均值。相似参数，是反映当前种群个体的相似程度，当相似参数较大时，说明种群个体相似度高，算法趋于收敛，种群个体整体性能优良，反之，说明种群个体相似度比较低，种群整体性能较差。

$$g_{avg} = \frac{g_1 + g_2 + ... + g_N}{N} \tag{4.5}$$

$$\sigma = \sqrt{\frac{1}{N}(\sum_{i=1}^{N}(g_i - g_{avg})^2} \tag{4.6}$$

$$\Omega = \frac{g_{avg} + 1}{\delta} \tag{4.7}$$

其中，N 表示种群的个体数，g_1，g_2，$\ldots g_N$ 表示种群个体的适应度值，g_{avg} 是种群适应度的算术平均值，可以反映个体的平均适应度，σ 表示标准差，Ω 表示相似参数，随着遗传算法进化代数的增加，种群适应度的平均值越来越高，但标准差值越来越小，相似参数值则越来越大。

根据上述交叉概率和变异概率的设计准则，并且结合标准差的概念结合相关参数的定义，交叉概率和变异概率的动态调节公式可以设计成如下公式：

$$p_c = 0.5 \times \frac{1}{1 + e^{-\frac{k_1}{\Omega}}} + 0.4 \tag{4.8}$$

$$p_m = \frac{k_2}{5(1 + e^{\frac{1}{\Omega}})} \tag{4.9}$$

其中，k_1，k_2 为两个常数，$k_1 \in$（0，∞），$k_2 \in$（0，1）。

从交叉概率和变异概率的自适应调节公式中可以看出，$p_c \in$（0.65，0.9），$p_m \in$（0，0.1），交叉概率和变异概率的值在合理范围之内，随着平方差的值增大，交叉概率越来越小，而变异概率越来越大，所以符合交叉概率和变异概率的两个设计准则，自适应遗传算法流程如图 4.2 所示。

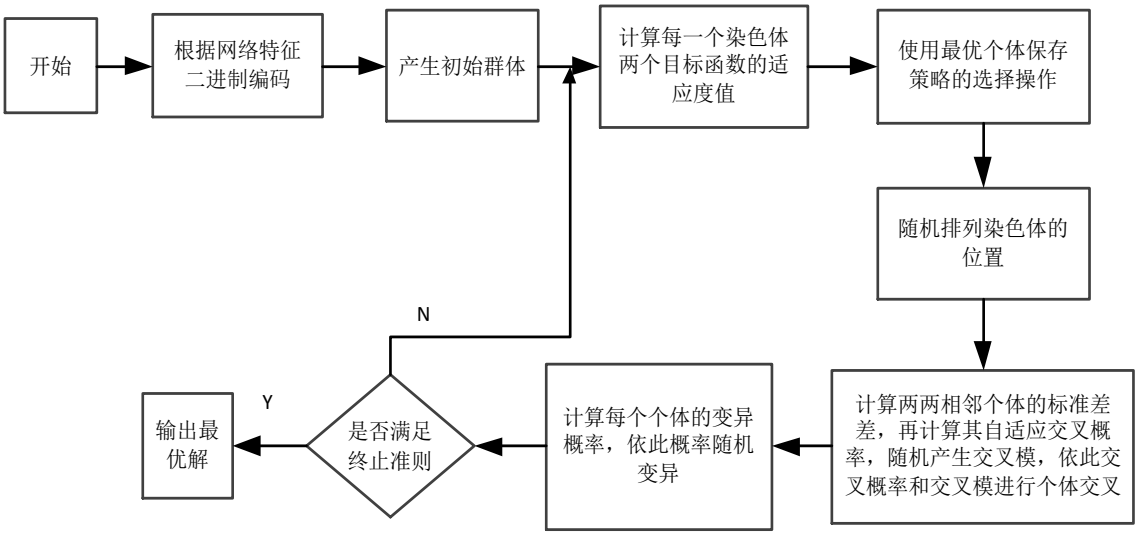


图 4.2 自适应遗传算法流程图

4.1.2 精英基因库

初步改进后的遗传算法在经过交叉、变异和选择后，在进化后期平方差值的日渐减小，个体趋向于收敛，基因编码相同的个体会逐步增多，所以文中引入精英基因库的概念，用于存储那些适应度较高的个体，对于精英基因库已经存在的重复个体，可以不用再重复解码，计算个体的适应度值等一系列过程。设计一个精英基因库，可以减小算法的复杂度，提高运行的效率，实用性。

精英基因库：用于存放遗传算法进化过程中后期出现适应度较大的个体以及对于的适应度值，个体按照适应度的大小进行排序，精英基因库的规模大小在 0.2~0.3 倍的个体种群的规模。

引入外部精英基因库[46]，根据当前种群中的染色体的适应度，构成初始外部精英基因库，将适应度较大的染色体的编码和对应的适应度值添加到精英基因库，并根据适应度值按照从大到小的规则进行排序。然后在子代的进化过程中，先计算个体适应度时，先在精英基因库中查找是否有相同编码的个体，如果存在相同个体，则直接以精英基因库中的对应个体的适应度值作为当前个体的适应度值；如果不存在则根据适应度函数进行计算，并根据计算得到的适应度值和精英基因库中个体的适应度进行比较，如果比精英基因库中的最小的适应度要大，则把此个体放入精英基因库，如果精英基因库中个体数量达到了规定大小，则舍弃精英基因库中适应度较小的个体，引入精英基因库后的 A-MOGA 算法流程如图 4.3 所示。

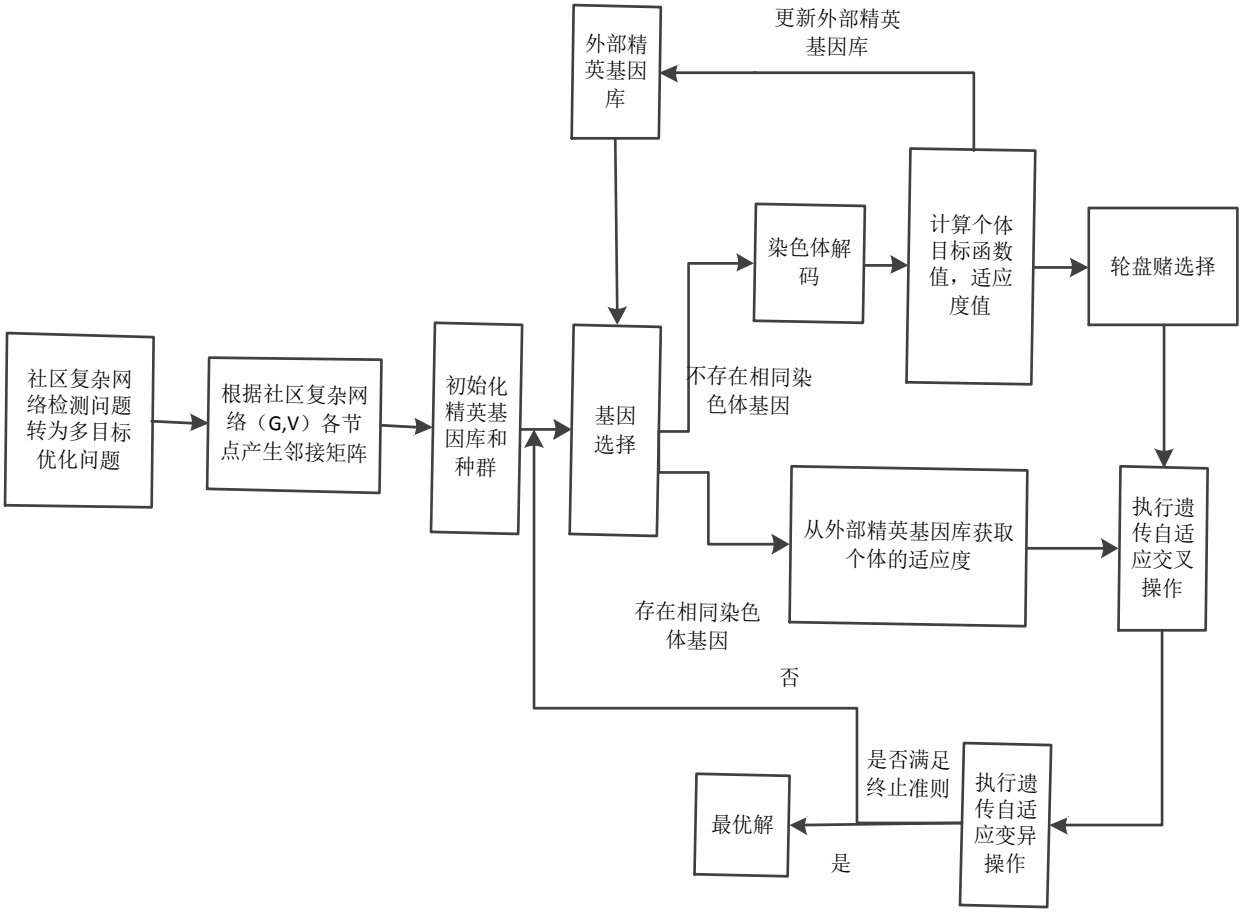


图 4.3 A-MOGA 算法流程图

由于遗传进化算法在进化前期，种群中个体的变化较大，所以个体重复率比较低，所以在算法进行到一定优化代数后，再开始进行精英基因库的创建，减少前期精英基因库的效率过低的问题。

虽然自适应快速遗传算法引入精英基因库在实际的求解中，会比传统的自适应遗传算法更加复杂一些，但是由于精英基因库会最大程度上会规避了个体的适应度重复计算问题，目标函数越复杂，越到了进化后期，重复个体的频繁出现，精英基因库的作用显得越来越明显。

4.2 多目标自适应快速遗传算法描述

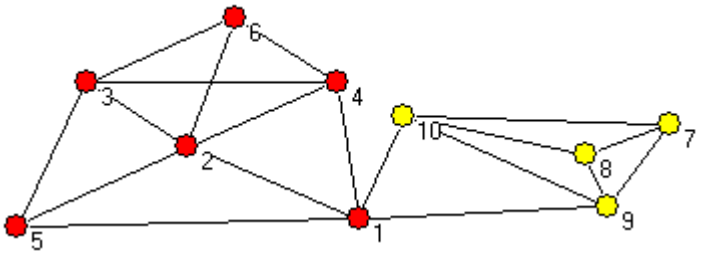
4.2.1 编码方式

多目标自适应快速遗传算法的采用基于基因近邻[18], [41], [47]的编码方式。在这种编码方式中，种群中的个体由 N 个基因 $g_1, g_2, \ldots g_N$ 组成，每个基因的等位基因值的取值范围为 $\{1, \ldots N\}$，基因和等位基因都表示图中 G=（V，E）的节点。例如第 i 个基因的等位基因值是 j，则可以理解为节点 i 和 j 之间有边连接。该编码方式需要通过一个解码步骤来识别各个

连通子网络，位于同一个连通向量的节点被划分在同一个社区网络当中。

这种编码方式的优点在于并不需要事先知道复杂社区的划分个数，在解码过程中可以自动计算得出社区的个数。图 4.4 给出了一个基于基因近邻编码方式的例子。其中图 4.4（a）表示了一个由 10 个节点组成的复杂网络，红色和黄色小圆点分别表示两个社区的网络节点，图 4.4（b）表示某个体的基因型，图 4.4（c）根据图 4.4（b）个体编码的社区划分解码结果，按照模块度 Q 的函数计算公式（在 4.2.5 节具体介绍），图 4.4 社区划分结果的 Q 值:

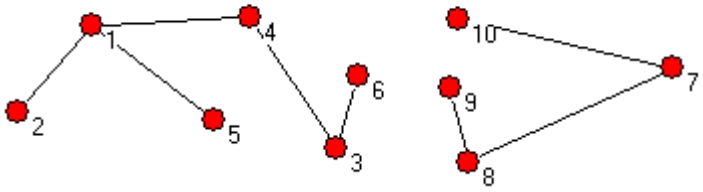
$$Q=\frac{11}{19}-(\frac{2}{19})^2+\frac{7}{19}-(\frac{2}{19})^2=0.8725$$



（a）复杂网络拓扑图

Position	1 2 3 4 5 6 7 8 9 10
Genotype	5 1 4 1 1 3 8 7 8 7

（b）某个基因型



（c）社区划分

图 4.4 复杂网络拓扑图、基因型和社区划分

4.2.2 种群初始化

首先随机产生一个种群，若干个随机产生的个体。在上一节叙述的基于基因近邻的编码方式中，如果基因 i 的等位基因是 j，则表示节点 i 和 j 之间有边连接，在解码后输出的社区划分结果中，节点 i 和 j 应该在相同社区中。因此，在本优化算法初始化过程中，我们按照连节点的连接关系进行合理分配，也就是说，对于每个个体的初始化，基因 i 的等位基因值 j 只能是 i 的相邻节点。如果第 i 个基因的等位基因值为 j，但边$(i，j)$实际上并不存在，则等位基因值 j 就会被 i 的相邻的一个点所取代。例如，在图 4.5(a)中，第 3 个和第 9 个基因的等位基因值分别是 9 和 6。但(3，9)和(9，6)的边并没有出现在图 4.4 (a)的网络拓扑结构中。因此，

第 3 个基因的等位基因值 9 可以被替换为 2，第 9 个基因的等位基因值 6 可以被替换为 8，替换后的基因如图 4.5(b)所示。这种初始化方式的有效的限制了解空间的大小，大大的减小了算法进化过程中的无效搜素，使算法的收敛速度得到了显著地提高。

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	9	1	1	3	8	7	6	7

(a)随机产生的基因

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	2	1	1	3	8	7	8	7

(b)改进后的基因

图4.5 基因初始化改进图

4.2.3 选择

A-MOGA 采用了轮盘赌选择法对当前种群个体实现“优胜劣汰”，它类似博彩游戏中的轮盘赌。轮盘赌选择过程中，每个个体进入到下一代的概率等于它的适应度值与整个种群中个体适应度值和的比例，个体适应度值越高，个体被选择的概率越高，被遗传到下一代的概率也越大。设种群大小为 M，个体 k 的适应度值为 f_k，则个体 k 被选中的概率为：

$$p_k = \frac{f_k}{\sum_{i=1}^{M} f_i}，\ k=1，2，\cdots，M \tag{4.10}$$

累积概率为：

$$q_k = \sum_{i=1}^{k} p_i，\ k=1，2，\cdots，M \tag{4.11}$$

选择过程就是旋转转轮 M 次，每次按照以下步骤选出一个个体加入到新的种群内：

Step1：在[0,1]区间内产生一个均匀分布的伪随机数 r；

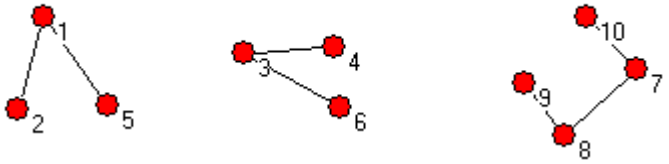
Step2：若 $r \le q_1$，则选择第一个个体；否则选择第 k 个（$2 \le k \le M$）使得 $q_{k-1} \le r \le q_k$ 成立；

重复上述步骤 Step1 和 Step2 进行 M 次。

4.2.4 交叉和变异

多目标自适应快速遗传算法采用了一种改进型的均匀交叉算子，来保证子代个体的有效性。均匀交叉对于种群个体的交叉概率为前面提到的自适应交叉概率 p_c，对父代个体染色体各个位置的基因以相同的概率实行交叉操作，首先，随机生成一个长度为 N（节点数量）的

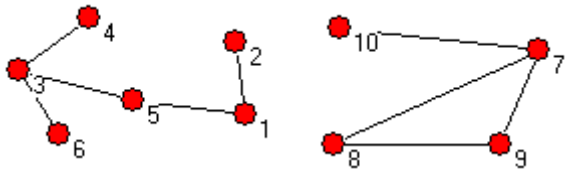
二进制交叉模。交叉模上的每一个值均为 0 或者 1。对于子代 C 的每个基因，如果交叉模上的这位为 1，则继承父代 B 中对应的等位基因值，如果交叉模上的这一个为 0，则继承父代 A 中对应的等位基因值，子代 D 则恰恰相反。由于在实际应用中，采用了上一节提到的有偏好的初始化，即在父代中第 i 个基因含有等位基因值 j，那么边$(i，j)$就会存在，采用均匀交叉，子代每个基因位的值都继承自父代，这样可以保证子代个体中网络各个节点的有效联系，图 4.6 是一个均匀交叉的例子。



Position　1 2 3 4 5 6 7 8 9 10

Genotype　5 1 4 3 1 3 8 7 8 7

（a）父个体 A 染色体和其图结构



Position　1 2 3 4 5 6 7 8 9 10

Genotype　5 1 5 3 1 3 9 7 8 7

（b）父个体 B 染色体和其图结构

父个体 A：　5 1 4 3 1 3 8 7 8 7

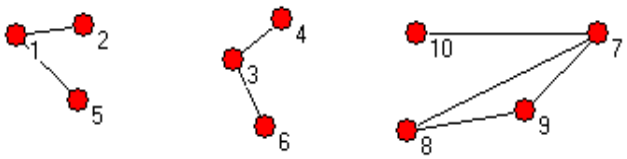
父个体 B：　5 1 5 3 1 3 9 7 8 7

交叉模 ：　0 0 0 1 1 0 1 1 0 1

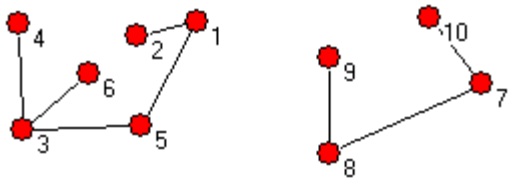
子代 C ：　5 1 4 3 1 3 9 7 8 7

子代 D：　5 1 5 3 1 3 8 7 8 7

（c）均匀交叉图



（d）子代 C 图结构



（e）子代 D 图结构

图 4.6 均匀交叉示例图

变异操作具体如下：对于待变异个体，以自适应变异概率 p_m 随机选择一个基因，将这个基因上的等位基因值改变为其对应的任意邻居节点，这种变异方式同样避免了无效解空间的搜索。因此，一个等位基因的可能值被限定于基因 i 的相邻基因。例如，在图 4.4(a)的网络拓扑图中，在第 3 个位置上的基因的等位基因值只能为 2，4，5，6。

在进化前期进行变异操作，进化过程中的变异概率设定为前面提到的自适应变异概率 p_m。如果变异算子随机改变基因 i 的等位基因值 j，它会导致对搜索空间的无效搜索，所以在实际变异过程中，这个基因上的等位基因值改变为其对应的任意邻居节点。这种变异保证了生成的变异后代中，每个节点只与复杂网络相邻的一个节点相连，提高了解空间的搜索效率。

4.2.5 目标函数

建立目标函数是优化设计中最重要的步骤，复杂网络划分工作中有密集的内部连接节点和稀疏的互连节点。因此，这种划分有两个相互竞争的目标：一个是社区之间联系的最小化，另一个是将同社区内节点之间的联系最大化。因此，社区检测的问题不能被表示为满足一个目标而忽视其他目标，所以它适合采用多目标优化方式。

在复杂网络社区结构划分的研究中，构造两个目标函数，即社区适应度函数和社区分值函数，第一个最小化目标函数社区适应度函数表述为：

$$fitness = \sum_{j=1}^{k} P(S_j) \tag{4.12}$$

其中，

$$P(S_j) = \sum_{i \in s} \frac{k_i^{in}(S)}{k_i^{in}(S) + k_i^{out}(S)} \tag{4.13}$$

而 k 为社区划分个数，$k_i^{in}(S)$ 表示节点 i 连接子网络 S 中其他节点的边的条数，$k_i^{out}(S)$ 表示

节点 i 连接子网络 S 外其他节点的边的条数。

第二个最大化目标函数，社区分值函数为：

$$CS=\sum_{i=1}^{k} score(S_i) \tag{4.14}$$

$$score(S)=M(S)\times V_s \tag{4.15}$$

$$M(S)=\frac{\sum_{i\in s}(u_i)^r}{|S|} \tag{4.16}$$

$$u_i=\frac{1}{|S|}k_i^{in}(S) \tag{4.17}$$

其中$|S|$表示社区 S 内部所有的节点数，V_s 表示的是社团 S 内部相连接的边总数，u_i 表示 S 内部与节点 i 相连的边占所有节点的比例，r 称为解析度参数，r 一般设为 2。是一个正实数，它用于控制网络社区的大小，r 一般设为 2。由于$0\le\mu_i\le1$，$r\ge1$，所以那些在社区 S 内部联系较多的节点的权重得到加强，那些在 S 内部联系比较少的节点的权重得到削弱。对于一个复杂社区网络，当$k_i^{out}(S)$=0 的时候，$P(S)$达到其最大值。

4.2.6 Pareto 解选择

A-MOGA 算法的每一个解都代表了两个目标间的不同权衡结果，这就造就了许多不同的网络社区检测方案。因此，我们需要确立一个标准参数来对 Pareto 解进行选择。多目标自适应快速遗传算法采用 Girvan 和 Newman 提出的模块度：

$$Q=\sum_{s=1}^{k}[\frac{l_s}{m}-(\frac{d_s}{2m})^2] \tag{4.18}$$

l_s 是连接模块 S 内部所有顶点的边的总数，d_s 是 S 中节点的度的总和，m 为网络中总的边数。该值越接近 1 表示越强的社区结构。实际网络中该值通常在 0.3~0.7 之间。

在多目标自适应快速遗传算法中，多目标划分形成的 Pareto 解集存储在精英基因库中。本文把模块度作为一个最优解选择标准，评定具有最大模块度的那个解所对应的网络划分为当前网络的最佳划分。

4.3 多目标自适应快速遗传算法流程

在多目标自适应快速遗传算法中，首先是种群的随机初始化，每个个体代表一个网络图

结构，它的每个组成部分都是 G 的一个相连子图。A-MOGA 计算每个个体的两个目标函数值，并根据 Pareto 解之间的支配关系对个体的两个目标函数值进行分类排序，然后执行前面描述的自适应交叉变异算子，产生一个新的群体。经过多次迭代，A-MOGA 算法最终返回一个模块度最高的 Pareto 最优解。

算法　4.1　A-MOGA 算法框架

1：输入：种群大小 Populaiton；最大迭代次数 Generation；

2：自适应参数：种群 P 的自适应交叉概率 p_c，自适应变异概率 p_m

3：$P \leftarrow$ Intialization(Populaiton);

4：While Termination(Generation);

5：$P_{parent} \leftarrow$ Select(P)

6：$p_c, p_m \leftarrow$ Adaptive ();

7：$P_{cross} \leftarrow$ Crossover(P_{parent}, p_c)

8：$P_{child} \leftarrow$ Mutation(P_{cross}, p_m);

9：$P \leftarrow$ Update(P_{child});

10：ElitePool();//精英基因库更新

11：End;

12：输出：将 ElitePool 适应度最大的非劣解转化网络划分输出。

给出一个网络 S 及它的模型图 $G = (V, E)$，A-MOGA 执行以下具体步骤：

Step1：把社区复杂网络检测问题转化为多目标问题，建立两个目标函数社区分值目标数和社区适应度目标函数；

Step2：根据社区复杂网络各节点连接关系产生邻接矩阵；

Step3：根据近邻原则初始化种群个体，同时初始化 0.2~0.3 倍种群规模的精英基因库；

Step4：进行基因选择，如果在精英基因库中存在相同染色体基因，则从基因基因库中获取个体的适应度，如果不存在相同染色体基因，则染色体解码，根据解码后的社区划分和原始复杂网络的邻接矩阵个体的两个适应度值，根据两个目标函数的支配关系，然后再按照 Q 值的大小再进行分类排序，更新精英基因库；

Step5：进行“优胜劣汰”，采用轮盘赌选择个体；

Step6：计算自适应交叉概率和变异概率，然后执行交叉变异操作，生成下一代种群；

Step7：判断是否达到最大迭代次数，如达到，返回精英基因库中中模块度最高的非劣解，作为最终输出的最优解。

在算法 4.1 框架中，函数 Initialization()用来初始化种群，函数 Select()是 A-MOGA 算法中的选择操作，函数 Adaptive ()根据式（4.8）和式（4.9）计算自适应交叉概率 p_c 和自适应变异概率 p_m；函数 Mutation()和 Crossover()分别为 A-MOGA 算法里的变异交叉操作和交叉操作，函数 Update()表示更新当前种群，即从种群 P 和 P_{child} 中选择适应度较大的个体，函数 Termination()表示循环语句的终止条件，设定最大迭代次数，函数 ElitePool()是精英基因库更新操作，对非 Pareto 非劣解按适应度值进行排序。

4.4 本章小结

本章首先在多目标优化策略和遗传算法基础上提出了多目标自适应快速遗传算法（A-MOGA），分析了该本算法中的自适应原理和建立精英基因库的理论基础，重点介绍了 A-MOGA 算法中的自适应交叉概率和变异概率的调节公式。然后，详细介绍了 A-MOGA 的具体算法流程，例如编码方式、种群初始化、两个目标函数和 Pareto 解的选择。最后，简单的描述了其算法中的函数框架。

第五章 多目标自适应快速遗传算法的仿真

由于我们提出的算法要用于复杂社区网络的检测，则算法的性能是一个不能回避的问题。我们需要用已知特性的社团网络去检测算法的性能。当前，用于检测算法性能的网络分两类：模拟网络和真实网络。在这一章中，我们将用一个人工模拟网络和四个真实网络来检测A-MOGA 算法的有效性，然后将其性能指标与 GN 算法[24]、GA-net 算法以及 MOPSO[48]算法划分的真实网络所得出的性能指标进行比较。为了测试 A-MOGA 算法的性能，本算法基于 Matlab 平台和 Pajek 软件进行实验仿真。仿真表明，A-MOGA 算法能够有效划分网络社区，并且和其他几种算法相比具有较大竞争力。

5.1 评价标准

为了评估算法得到的最优解，识别好的社区划分，需要引入评价指标。 一方面，社区检测算法将第四章 Newman 和 Girvan 提出的模块度[49]函数作为衡量社区划分的度量标准。另一方面，基于信息论的知识，采用标准化的互信息度量(Normalised Mutual Informationn Measure，*NMI*)[49]，来评估真实分区与所检测出来的社区的相似性。*NMI* 指标主要是用来度量两个分割的相关性的，它不只考虑每行最大的元素与行元素之和的比较，而是考虑了每行的所有元素的非零值与该行元素之和的比较，其中 $0 \le NMI \le 1$。

关于 *NMI* 这个评价指标的，Danon 等人在实验中已经证明了它比较适合网络分区[47]。因此，我们把 *NMI* 作为本文仿真实验中的评价指标之一，来评估网络社区划分的优劣性。

NMI 的表达式为

$$NMI(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{ij}\log(C_{ij}N/C_iC_j)}{\sum_{i=1}^{C_A}C_i\log(C_i/N)+\sum_{j=1}^{C_B}C_j\log(C_j/N)} \tag{5.1}$$

假设 A 和 B 是一个网络的两种划分，C 为混合矩阵，其元素 C_{ij} 表示划分 A 中的社区 i 里面的顶点在划分 B 中的社区 j 里面也出现的个数。其中 C_A 和 C_B 分别表示划分 A 和 B 中社区的个数，C_i 表示 C 中第 i 行元素之和，C_j 表示 C 中第 j 列元素之和，N 是节点的数量。

如果检测得到的社团结构与真实的社团完全相同，*NMI* 取得最大值 1；相反，当检测得到的社团结构与真实的社团结构完全无关时，如果整个社区复杂网络被检测为一个大的社团

时，*NMI* 取得最小值 0。

5.2 模拟网络的仿真

在复杂网络研究分析中，为了检验算法检测复杂网络社区结构的性能优劣性，通常需要把算法应用到拓扑结构已知的复杂网络中，即这类网络有着明确的社团结构，这些网络往往依据一定的算法通过计算机随机产生。最后，我们把它的性能评价指标与现在流行的算法进行比较。

一个好的复杂网络社区结构检测算法应当能够很好地发现网络中的社区结构。在本研究的模拟网络的仿真中，我们采用文献[50]提出的 Benchmark 网络[50]来检测算法的可行性和有效性。该网络包含 128 个节点，4 个社区，每个社区有 32 个节点，节点的平均度为 16，混合参数μ控制了节点外度所占的比例，μ越小，节点和其社区外节点的连接比例越小，社区结构越清晰。实验中调节μ的值，生成μ从 0 到 0.5 变化（间隔为 0.05）的 11 个网络，并且使用 *NMI* 衡量真实网络的划分和社区划分结果之间的相似性。针对整个网络，计算 20 次独立运行结果中 *NMI* 各次最大值的平均值和 *NMI* 的全局最大值。当μ 取 0.5 时，每个节点平均有一半与其相连接的节点在社区外，此时社区结构比较模糊。当$\mu < 0.5$时，节点外度所占比例小于内度所占比例。当μ取 0 时，表明外度所占比例为 0，节点仅与自身社区内的节点相连接，此时社区结构最明显。

图 5.1 是 A-MOGA 算法的迭代次数设为 100 次，种群规模设为不同，在不同混合参数μ的 Benchmark 模拟网络下，算法运行 20 次的 *NMI* 最大值的统计数据图，从图 5.1 中可以看出，当$\mu \le 0.25$时，种群规模分别为 100，150，200 的 A-MOGA 算法的 *NMI* 最大值均为 1，说明社区划分效果非常好；当$0.25 \le \mu \le 0.3$时，种群规模为 100 的算法 NMI 开始下降，其他规模的 *NMI* 保持在 1；当$\mu \ge 0.3$，种群规模为 150 的 *NMI* 开始下降，种群规模为 200 的 *NMI* 依旧保持在 1，种群规模为 100 的 *NMI* 下降速度不变；当混合参数$0.35 \le \mu \le 0.4$时，种群规模为 100 和 150 的 A-MOGA 算法的 *NMI* 下降速度比较快，而种群规模为 200 的 *NMI* 最高值依旧为 1； 在$\mu > 0.45$，种群规模为 200 的 A-MOGA 算法 *NMI* 急剧下降，而种群规模为 100，150 的 *NMI* 值缓慢下降，当μ下降至 0.5 时，种群规模为 100，150 和 200 值的 *NMI* 差不多。

图 5.2 是 A-MOGA 算法的种群规模设为 150，最大迭代次数设为不同，在不同混合参数μ的 Benchmark 模拟网络下，算法运行 20 次的 *NMI* 最大值的统计数据图，我们从图 5.2 中可

以看出，当$\mu \le 0.25$时，迭代次数 50，100，150 时的 A-MOGA 算法的 *NMI* 最大值为 1，说明社区划分效果非常好；当$\mu = 0.25$迭代次数为 50 的 *NMI* 开始下降，迭代次数 100 和 150 的 *NMI* 保持在 1；当$\mu = 0.3$，迭代次数为 100 的 *NMI* 开始下降，迭代次数为 50 的下降速度几乎保持不变，而迭代次数为 150 的 *NMI* 还保持在 1；当混合参数$\mu > 0.35$时，迭代次数 150 时的 A-MOGA 算法的 *NMI* 最大值开始急剧下降；当$\mu \ge 0.45$，不同迭代次数的 *NMI* 最大值下降速度相差不大。

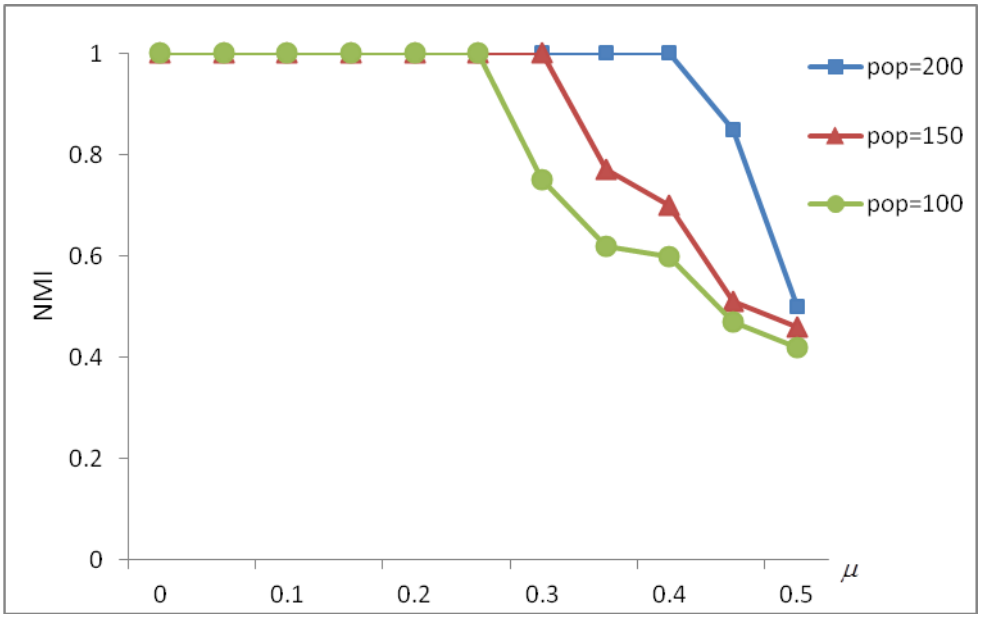


图 5.1　A-MOGA 算法在不同种群规模下对 Benchmark 网络运行 20 次的 *NMI* 最高值

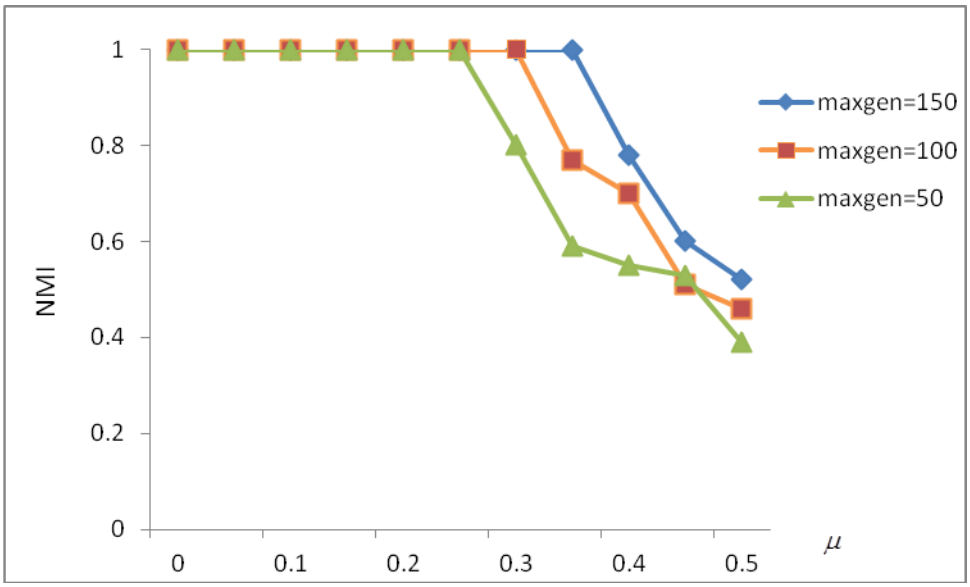


图 5.2　A-MOGA 算法在不同迭代次数下对 Benchmark 网络运行 20 次的 *NMI* 最高值

图 5.3 是 A-MOGA 算法的种群规模设为 200，迭代次数设定在 150 次，在不同混合参数μ

的 Benchmark 模拟网络下，A-MOGA，MOPSO，GN，GA-net 算法运行 20 次的 *NMI* 最大值的统计数据图，我们从图 5.3 中可以看出，当 $\mu \le 0.15$ 时，A-MOGA，MOPSO，GN，GA-net 算法的 *NMI* 最大值都为 1，说明它们能够很好地发现 Benchmark 模拟网络中的社区结构；$\mu = 0.15$，GA-net 的 *NMI* 最大值开始下降，其他几种算法的 *NMI* 最高值还是保持在 1；当 $0.15 \le \mu \le 0.35$ 时， A-MOGA，MOPSO，GN 算法的 *NMI* 最大值依然为 1，说明能准确地划分 Benchmark 网络，而 GA-net 的 *NMI* 快速下降到一个较小值，说明它无法准确检测 Benchmark 网络社区结构；在混合参数 $\mu > 0.35$ 时， GN 算法的 *NMI* 值开始下降，A-MOGA，MOPSO 算法的 *NMI* 值依然保持在 1；在混合参数 $\mu > 0.4$ 时 A-MOGA，MOPSO 算法的 *NMI* 值都开始下降，但 A-MOGA 下降速度比 MOPSO 慢；当 $\mu = 0.5$，A-MOGA，MOPSO，GN 算法的 *NMI* 值下降至 0.45 左右，GA-net 的 *NMI* 下降至一个很小的值。

随着 μ 增大，Benchmark 模拟网络被各种算法准确识别的难度越来越大。总而言之，当混合参数 $\mu > 0.35$ 时，A-MOGA 智能优化算法比其他优化算法在检测模拟人工合成网络社区结构中的性能更加优越。

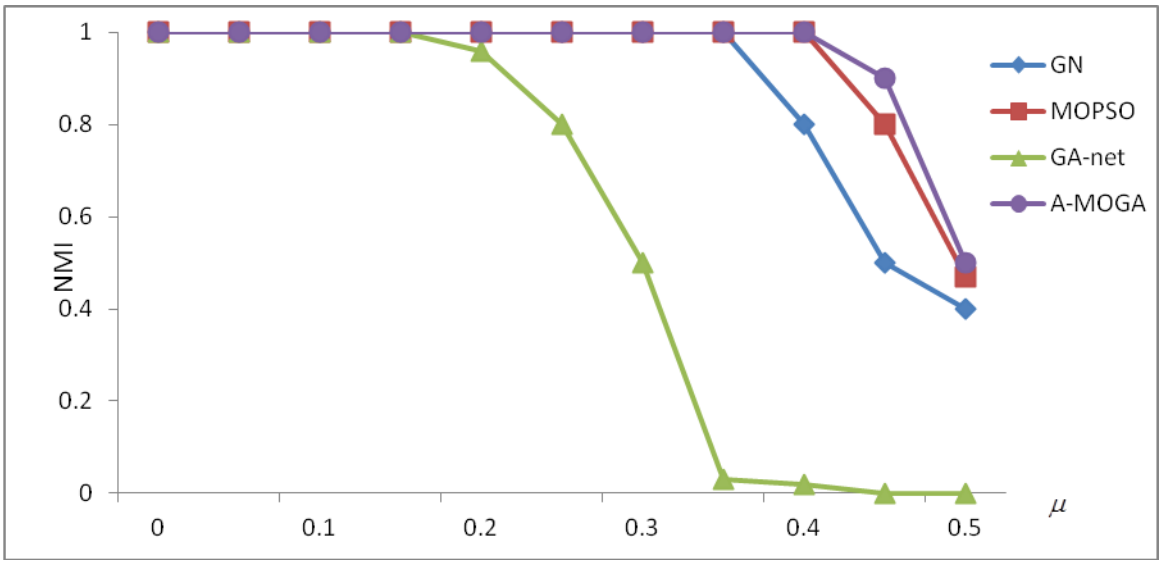


图 5.3　A-MOGA 和其他优化算法在对 Benchmark 网络运行 20 次的 *NMI* 最高值

5.3 真实网络的仿真

将 A-MOGA 算法应用到四个真实的网络上，分别是 Bottlenose Dolphins[51]，Zachary's Karate Club [52]和 American Coll. Football[24]以及 Krebs' Book [53]。表 5.1 描述了这四个真实网络的边数、节点数和真实网络分簇的个数。

表 5.1 四个真实网络的特性

网络	边数	节点数	真实分簇个数
Karate	78	34	2
Dolphin	159	62	2
football	613	115	12
Krebs' Books	441	105	3

Zachary Karate Club 网络是 Zachary 在两年时间内通过观察一个 34 个成员的空手道俱乐部得到的。图 5.4 记录 Karate Club 真实的社区划分结果，图 5.5 表明了 A-MOGA 算法在 Karate Club 上的社区检测结果。从图 5.5(a)可以清楚地看到 A-MOGA 算法能完全正确的检测 Karate Club 的社区划分结果(对应的 *NMI*=1)，同时图 5.5(b)也显示了最高的 *Q* 值(对应的 *Q*=0.4195)的社区划分结构，很明显图 5.5(b)中为图 5.5(a)的子图。

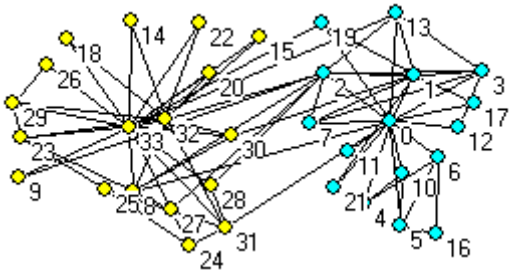
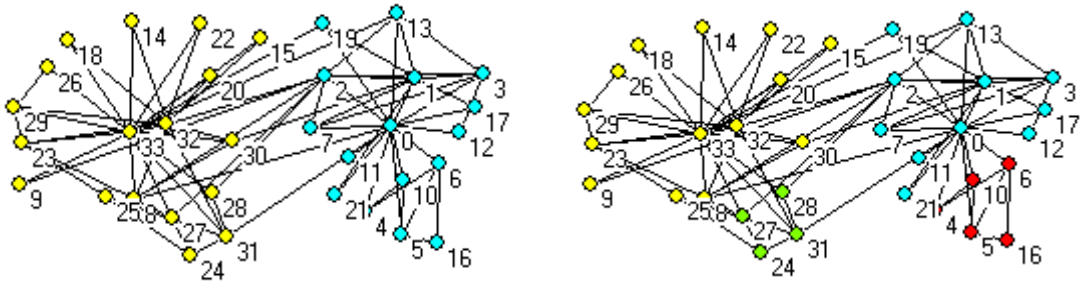


图 5.4 Karate Club 真实的划分结果



（a）NMI=1　　（b）Q= 0.4195

图 5.5 A-MOGA 算法在 Karate Club 的社区检测结果

American Coll.Football 是指美国各个大学足球联赛的比赛网络，由 Newman 等人提出。网络中的节点代表球队，节点之间的边表示球队与球队之间有比赛。图 5.6 表明 Football 的真实网络划分，图 5.7（a）表明 Football 在 *NMI* 最高值（对应 *NMI*=0.9268）时的网络划分结果图，图 5.7（b）表明 Football 在 *Q* 最高值（对应 *Q*=0.6046）时的网络划分结果图。根据图 5.6 和图 5.7 的比较结果，A-MOGA 算法在 Football 网络中错分了几个节点，例如在 *NMI* 最高值，节点 28，59，63，80，97 被错分，在 *Q* 最高值时，也出现了几个重叠模糊节点（节 28，63，90，97）被错误划分的情况。对于像 Football 这种结构比较复杂，且有若干个模糊节点出现的真实复杂网络中，A-MOGA 也不能 100%准确地划分出社区结构，而且在多次运行中，被错误划分的节点集中在那几个模糊节点。虽然如此，但根据表 5.2 和表 5.3，在 Football 网络社区结构检测中，A-MOGA 算法的 *NMI* 和 *Q* 值相比其他几种算法，也还是具有一定的优势。

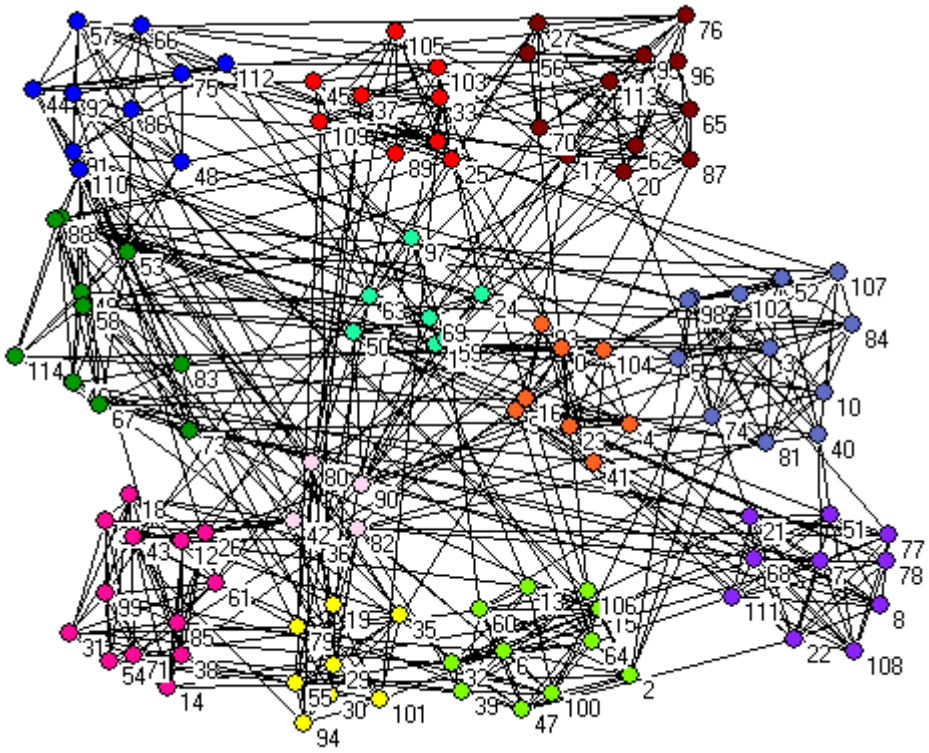
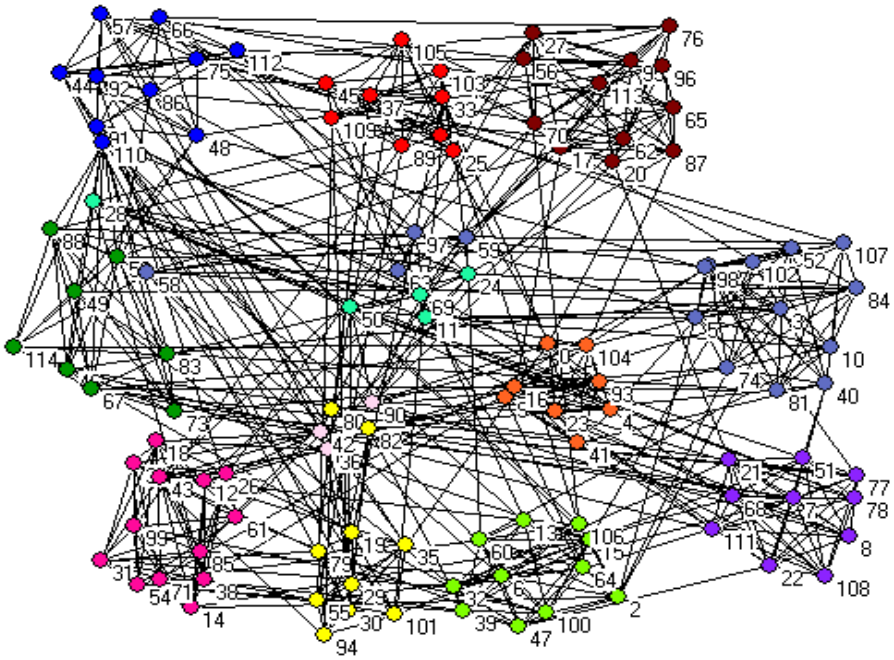
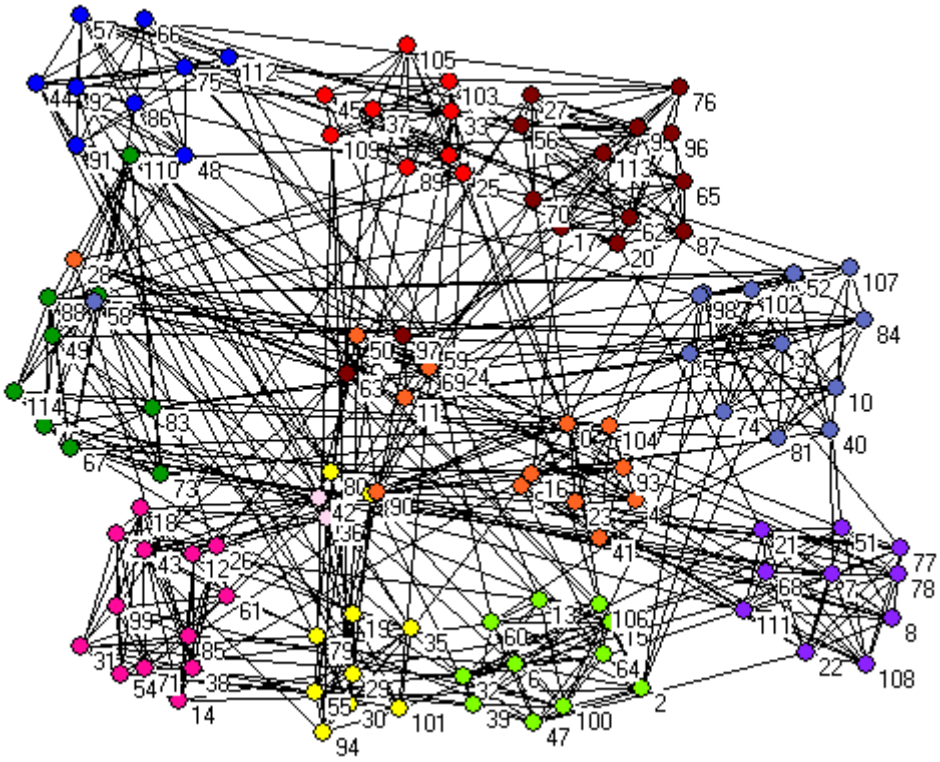


图 5.6 Football 的真实网络划分

Krebs' Books :这个网络是 2004 年美国总统大选时出版并在网络书店 Amazon.com 销售的有关美国政治的书籍。书籍之间的连接表示同一个买家通常同时购买这两本书[28]。Mark Newman 根据书籍的描述和 Amazon 上的评论对这些书籍做了分类。



（a）*NMI*=0.9286



（b）*Q*=0.6046

图 5.7　A-MOGA 算法在 Football 网络的社区检测结果

图 5.8 记录了 Krebs' Books 网络的真实网络划分，图 5.9（a）记录了 Krebs' Books 网络在 *NMI* 最高值（对应 *NMI*=0.5971）时的网络划分结果图，图 5.9（b）记录了 Krebs' Books 网络在 *Q* 最高值（对应 *Q*=0.5258）时的网络划分结果图。根据图 5.9 的 Book 两个网络划分结果和图 5.8 真实网络划分的比较得出，当 *NMI* 最高值时，Books 网络被划分出 *A*、*B*、*C* 三个子

网络，但是节点 64、102、103 节点没有被准确地划分出，而当 Q 为最大值时，Books 网络网络划分个数由 3 个增加到 4 个，图 5.9（b）中的 B 网络和 D 网络实际上是图 5.9（a）中网络 B 的两个子网络，而被错分的几个节点依然是节点 64、102、103。这种不同性能标准产生出网络分层结构现象，我们在下一小节会详细介绍。

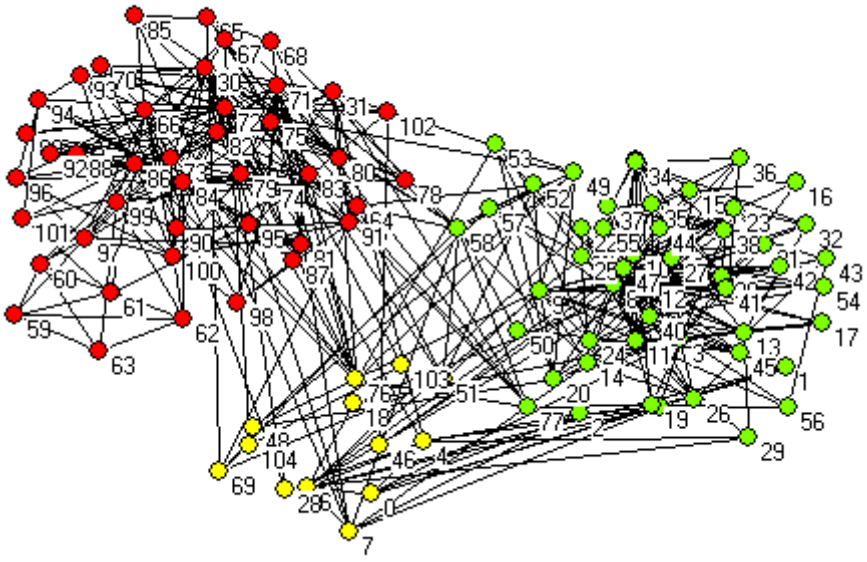
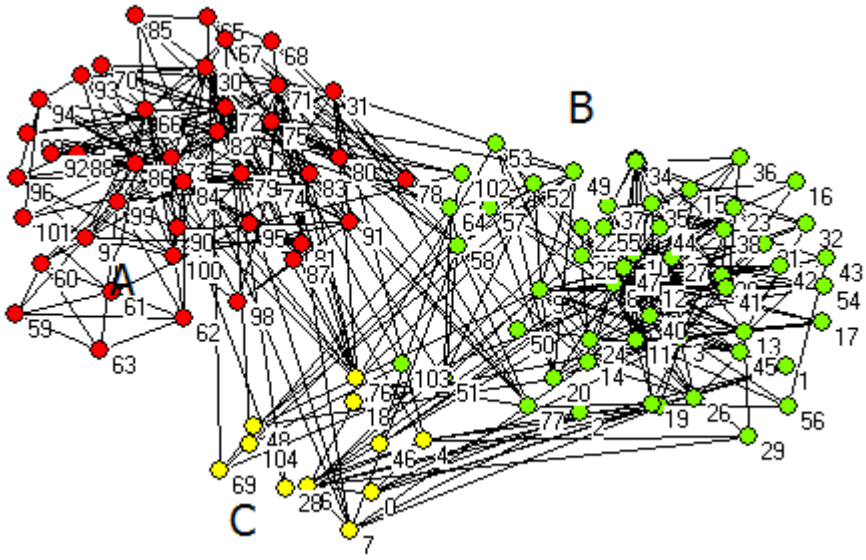
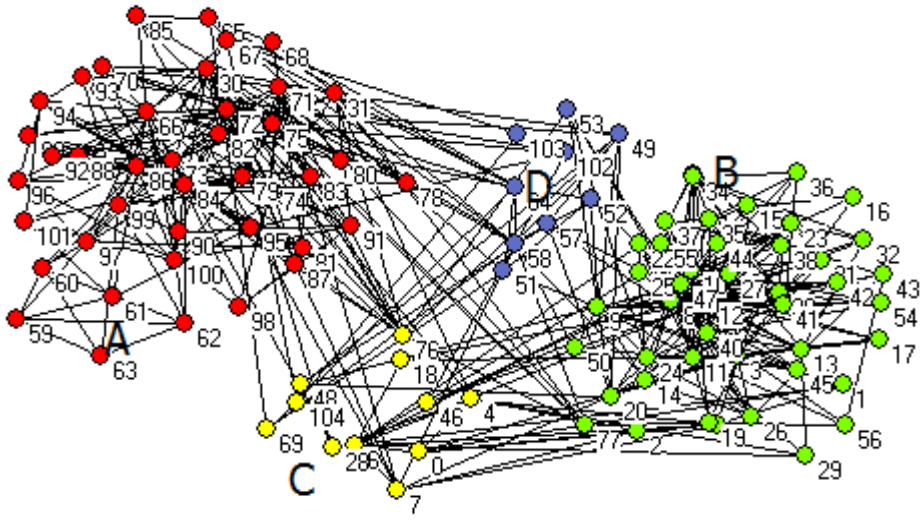


图 5.8 Krebs' Books 网络的真实网络划分



（a）*NMI*=0.5971



（b）Q=0.5258

图 5.9 A-MOGA 算法在 Krebs' Books 的社区检测结果

表 5.2　A-MOGA 算法 和其他四种算法四个真实复杂网络独立运行 20 次的 *NMI* 的对比结果

Methods / Networks	**A**-MOGA		GA-net		MOPSO		GN	
	NMI_{max}	NMI_{avg}	NMI_{max}	NMI_{avg}	NMI_{max}	NMI_{avg}	NMI_{max}	NMI_{avg}
karate	1	1	0.6369	0.6369	1	0.9566	0.8630	0.8630
dolphin	1	0.9453	0.4304	0.4148	1	0.9442	0.5540	0.5540
Football	0.9286	0.8955	0.9159	0.8984	0.9046	0.8963	0.8210	0.8210
Krebs' Books	0.5971	0.5202	0.5371	0.5147	0.5310	0.5214	0.5341	0.5174

表 5.3　A-MOGA 算法和其他四种算法在四个真实复杂网络独立运行 20 次的 Q 值的对比结果

Methods / Networks	**A**-MOGA-		GA-net		MOPSO		GN	
	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}	Q_{max}	Q_{avg}
karate	0.4195	0.4162	0.4059	0.4059	0.4198	0.4160	0.2330	0.2330
dolphin	0.5267	0.5223	0.5013	0.4946	0.5258	0.5215	0.4060	0.4060
Football	0.6046	0.6013	0.5941	0.5830	0.6046	0.6012	0.5350	0.5350
Krebs' Books	0.5258	0.4984	0.5151	0.4896	0.5231	0.5065	0.5061	0.4957

表 5.2 中数据统计显示：当 A-MOGA 算法和其他三个算法在四个真实网络各运行 20 次，在 karate 网络和 dolphin 网络（该网络在 5.4 节中具体介绍）中，A-MOGA 算法和 MOPSO 算法的 *NMI* 最高值等于 1，GA-net 算法的 *NMI* 值最小；在 Football 和 Books 网络中，A-MOGA

算法的 *NMI* 的最高值明显比其他几种算法高，它的 *NMI* 平均值比 GA-net 和 GN 高，但是比 MOPSO 略低。表 5.3 数据统计显示：在 karate 网络运行 20 次时，A-MOGA 算法的 *Q* 最大值比 MOPSO 算法略低，而平均值略高，比 GA-net 算法的 Q 最大值和平均值略高，但是比 GN 的 *Q* 值高近一倍；在 Football 网络中，A-MOGA 算法的最大 *Q* 值和平均 *Q* 值比 GA-net 和 GN 略高，和 MOPSO 差不多；在 Books 网络中，A-MOGA 的最大 *Q* 值比其他三个算法高一点，但是它的 Q 平均值比 MOPSO 略低。

综述，由 A-MOGA 在四个真实复杂网络[53]中运行的最大 *NMI* 和最大 *Q* 值的分析得出，它比常见的几种传统算法在寻优性能上有所提升。从仿真运行的平均 *NMI* 和平均 *Q* 值上分析得出，其算法稳定性能虽然在 Books 网络中比 MOPSO 稍逊，但整体而言，它的稳定性还是一定程度地提高了。

5.4 快速性仿真验证

MOGA 算法引入了精英基因库，前一章从精英基因库的设计原理上对算法的快速性做了一个理论分析。本小节针对引入的基因精英库在 Benchmark、Karate、Football 和 Books 网络上进行快速性仿真验证。

仿真过程中，算法的种群规模设为 200，迭代次数为 100。

表 5.4 A-MOGA 仿真时间复杂度

Networks / Methods	Benchmark	Karate	Football	Books
多目标自适应遗传算法	49.081s	10.587s	324.024s	383.455s
多目标自适应快速遗传算法	35.325s	10.012s	261.511s	270.324s

在表 5.4 中，多目标自适应快速遗传算法在 Benchmark、Karate、Football 和 Books 网络中比一般的多目标自适应遗传算法消耗的时间短。可以得出，当复杂网络的节点数增多，引入基因精英库比没有基因精英库的算法节省了一定的优化时间，显示了它在实际应用中具有一定的快速性。

5.5 Pareto 解的网络层次结构

模体是网络中有少量节点按照一定拓扑结构构成，并且相对于随机网络在网络中复制出现的小规模模式。模体实际上就是网络中大量出现的具有相同结构的子图，子图从局部层次刻画了网络内部相互连接偶的特定模式。模体是复杂网络的基本模块[54]，每个复杂网络通常都由其自身一组特定的模体构建，而检测出这些模体有利于识别网络的局部连接模式。模体可以帮助我们对网络简化，让我们对复杂网络有一个全局的认识。

层次模块性：模块是指一组物理上或功能上连在一起，共同完成一个相对独立功能的节点组。很多系统都包含模块，高模块化是一个大型复杂系统的基本设计要求。人们基于网络拓扑结构识别出网络的模块，通过分析这些模块与功能之间的关系可说明模块划分的有效性。在高度连接的社区子网络中，度很小的节点具有高的聚类系数；相反，度很高的中心具有较低的聚类系数，其作用只是把不同的子网络连起来。在具有层次模块性的网络中，很多内部关联密集的小规模节点之间比较松散，从而形成更大规模的拓扑模块。这种拓扑结构按照层次排列，由模块迭代方式生成的网络称为层次网络[55]。层次网络同时存在局部聚类特性、模块性和无标度拓扑特性。

Bottlenose Dolphins：这个网络是由 Lusseau 等通过对 62 只海豚的行为进行长时间的观察得到的，两只海豚之间有边相连接则说明它们间经常接触，此网络有 159 条边，形成了两个社团。在 Lusseau 的整个研究中，首先，整个 Dolphins 网络自然地分为 A 和 B 两组网络，分别对应于雌性 Dolphins 和雄性 Dolphins。通过进一步的研究，Lusseau 发现雄性 Dolphins 网络进一步分成 3 个社区分组，并猜测这是由于这三个网络分组属于三个不同的社区母系族谱 [54]。

图 5.10 表示 Dolphins 网络运行 20 次某次精英基因库里的 8 个 Pareto 前沿解。横坐标和纵坐标分别表示两个目标函数社区适应度和社区分值的函数值，红点方框里的两个数值记录了该解对应的 *NMI* 和 *Q* 值。图 5.10 显示，*NMI* 最高值为 1，*Q* 值最高为 0.5267，这两个解对应的社区网络划分结果如图 5.12(a)和 5.12(b)所示，图 5.12(c)是当 *NMI*=0.8326 的网络划分结果，根据图 5.11，*NMI*=1 的划分结果和真实社区划分结果一样，把 Dolphins 网络分成 *A* 和 *B* 两个子网络，图 5.12(b)是把图 5.11 中的 *B* 网络再分成两个子网络，而 5.12(c)是把图 5.12(b)中的 *C* 网络继续划分成 *C* 和 *D* 两个更紧凑的子网络。

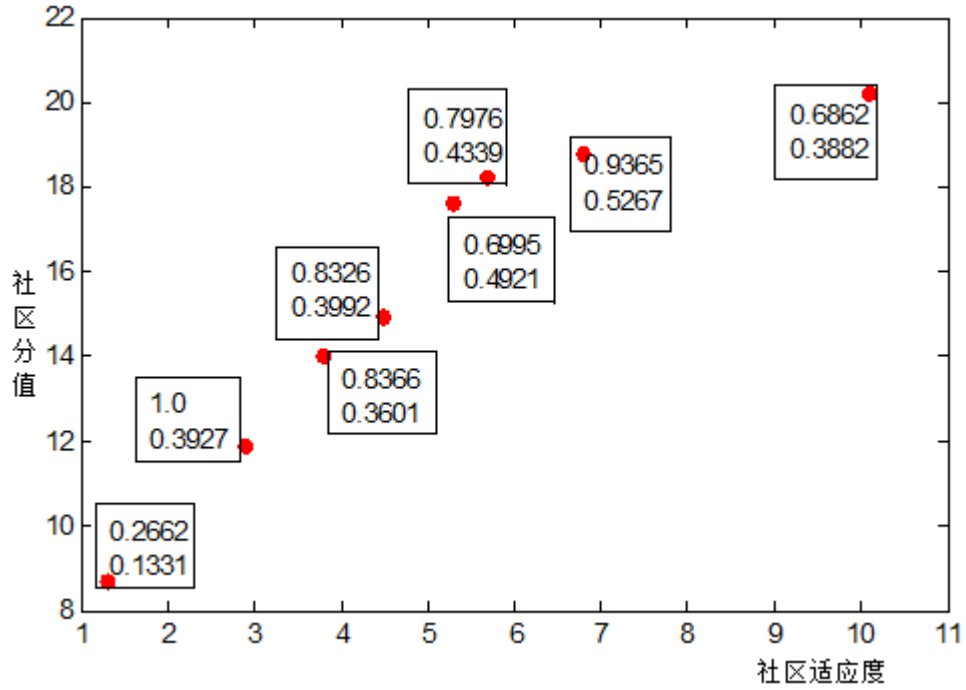


图 5.10 Dolphins 网络某次运行的精英基因库里的 Pareto 解

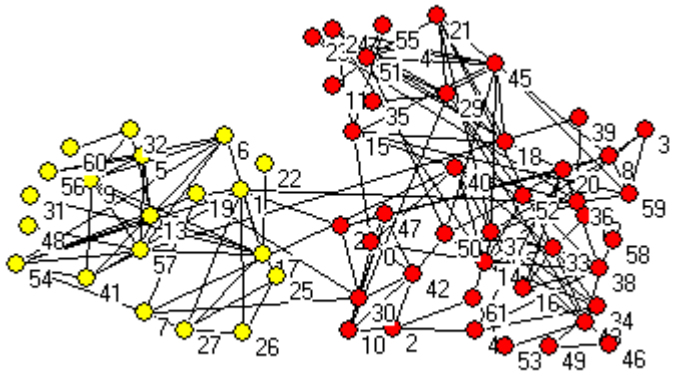
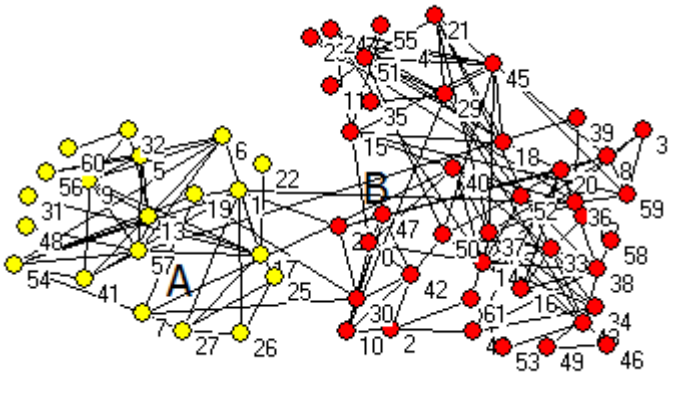
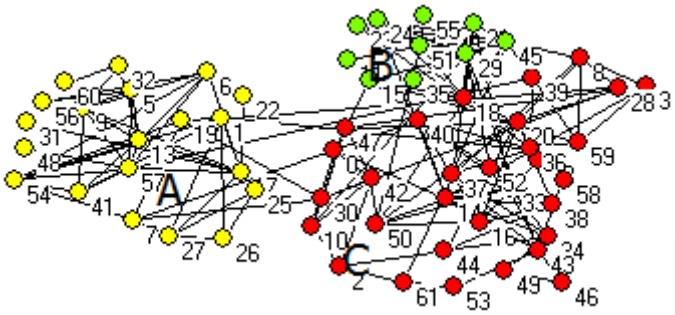


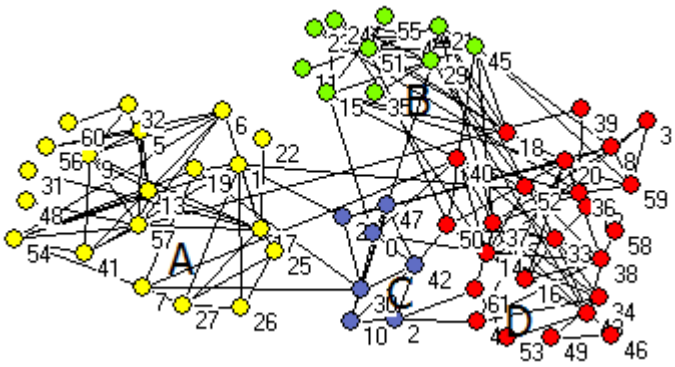
图 5.11 Dolphins 网络的真实社区划分



（a）*NMI*=1，*Q*=0.3927



（b）*NMI*=0.9365，*Q*=0.5267



（c）*NMI*=0.8326，*Q*=0.3992

图 5.12 A-MOGA 算法在 Dolphins 的社区检测结果

复杂网络的层次结构是对复杂网络中不同粒度，不同层次的社区进行整合。在复杂网络中，有的复杂网络的子网络还可以在低层次上细分为多个紧凑的子社区。根据上述实验结果分析，A-MOGA 在网络社区划分中它为研究者提供了一整套的 Pareto 前沿解集，发现网络中的分层结构，从而让我们更加了解掌握复杂网络的内部结构，有利于对复杂网络进行开发利用。在这一方面，它比纯粹的单目标优化方法更有优势。

5.6 本章小结

本章首先介绍了本算法性能的两个衡量标准，即模块度 *Q* 和标准化的互信息度量 *NMI*，然后使用 Benchmark 模拟网络和真实网络对 A-MOGA 进行仿真验证，最后将实验仿真结果与几种传统智能优化算法进行对比。分析得出，根据问题自身特性构造出的自适应遗传算子和基因精英库，有利于提高算法在复杂网络社区检测上的寻优性和稳定性；基于 Pareto 解的 A-MOGA 有利于发现复杂网络的分层结构；A-MOGA 的内在并行机制及其全局优化的特点适合多目标优化问题的解决。

第六章 总结与展望

随着复杂网络研究的兴起，社区结构检测的应用性研究逐渐成为这几年的研究热点。因为传统社区检测算法存在着效率低、优化解单一的缺点，本文研究分析了一种多目标自适应快速遗传算法。

6.1 工作总结

A-MOGA 算法实现了串集搜索，可以在事先并不知道社区准确数量的情况下对网络进行划分，它包含三个核心部分：

（1）引入两个目标函数，与单目标方法相比，多目标方法的优势在于，它可以同时对多重标准进行优化，并且提供一套解集而不是单个解，每个解都对应不同数量的社区，从而找到最佳的均衡。由于复杂网络社区结构往往是分层的，网络中的小社区会聚集形成更大的社区。因此，在各种层次上对网络性质进行研究就显得特别重要，Pareto 前沿中的非支配解集可以对社区结构在不同的层次上进行分析。

（2）引入外部精英基因库，用于存储适应度较高的非劣解，对于外部精英基因库已经存在的重复个体， 可以不用再重复解码，计算个体的适应度值等一系列过程。当 A-MOGA 执行遗传算子，它返回一组两个目标函数之间进行折衷的非支配解，经过解码把那些解生成图的邻接矩阵，从而把一个复杂网络分成多个独立的子网络。由于解集的趋向收敛性，精英基因库的引入在最大程度上减小了重复的计算量。

（3）引入 Logisitic 自适应变异概率和交叉概率，根据种群的适应度和种群个体之间的相关特性来改变变异概率和交叉概率，对遗传算法进化过程的寻优效率和准确性能有显著提高。

实验仿真表明，根据模块度 Q 和 NMI 这两个复杂社区网络划分的衡量标准，A-MOGA 比现有的几种方法更加有效。

6.2 展望未来

对于复杂网络社区结构的检测问题，本文提出了一种新的自适应快速遗传优化算法，虽然取得了一定的成果，但是这只是一个探索性研究。在实际工程应用中，可能还会出现 A-MOGA 的算法参数不佳、目标函数不合适的问题。因此，我们还需要在以下方面进一步研究：

（1）需要专门对一些重叠节点比较多的已知网络进行仿真研究，调整算法的某些参数，使它在该类网络中的性能得到进一步提升。

（2）在算法中的多目标处理问题上，对于两个目标函数，我们在接下来的工作中，可以进一步优化研究，争取找到更加合适的目标函数。

（3）由于目前我们对 A-MOGA 的研究还只停留在几个已知社区网络的仿真研究中，接下来，我们努力把它推广到跟我们实际生活密切相关的网络中。

参考文献

[1] 谢高岗, 张玉军, 李振宇等. 未来互联网体系结构研究综述[J]. 计算机学报, 2012, 35(6): 1109-1119.
[2] 张洋洋. 复杂网络中社团结构发现算法的研究与实现[D]. 南京理工大学, 2014. [3] 赖大荣. 复杂网络社团结构分析方法研究[D]. 上海交通大学, 2011.
[4] 黄韬, 刘江, 霍如等. 未来网络体系架构研究综述[J]. 通信学报, 2014, 35(8): 184-197.
[5] Clauset A. Finding local community structure in networks[J]. Physical review E, 2005, 72(2): 026132.
[6] Newman M E J, Forrest S, Balthrop J. Email networks and the spread of computer viruses[J]. Physical Review E, 2002, 66(3): 035101.
[7] Li Z, Zhang S, Wang R S, et al. Quantitative function for community detection[J]. Physical review E, 2008, 77(3): 036109.
[8] Newman M E J. The structure of scientific collaboration networks[J]. Proceedings of the National Academy of Sciences, 2001, 98(2): 404-409.
[9] Kleinberg J M. Navigation in a small world[J]. Nature, 2000, 406(6798): 845-845.
[10] Scala A, Amaral L A N, Barthélémy M. Small-world networks and the conformation space of a short lattice polymer chain[J]. EPL (Europhysics Letters), 2001, 55(4): 594.
[11] Krapivsky P L, Redner S. Organization of growing random networks[J]. Physical Review E, 2001, 63(6): 066123.
[12] 于洋. 浅析二项分布, 泊松分布和正态分布之间的关系[J].企业科技与发展:下半月,2008 (10): 108-110.
[13] Watts D J, Strogatz S H. Collective dynamics of 'small-world'networks[J]. nature, 1998, 393(6684): 440-442.
[14] Barabási A L, Albert R. Emergence of scaling in random networks[J]. science, 1999, 286(5439): 509-512.
[15] 寇晓丽. 群智能算法及其应用研究[D]. 西安电子科技大学, 2009.
[16] Dorigo M, Birattari M, Stützle T. Ant colony optimization[J]. Computational Intelligence Magazine, IEEE, 2006, 1(4): 28-39.
[17] Encyclopedia of machine learning[M]. Springer Science & Business Media, 2011.
[18] Pizzuti C. Ga-net: A genetic algorithm for community detection in social networks[M]//Parallel Problem Solving from Nature–PPSN X. Springer Berlin Heidelberg, 2008: 1081-1090.
[19] Chun J S, Jung H K, Hahn S Y. A study on comparison of optimization performances between immune algorithm and other heuristic algorithms[J]. Magnetics, IEEE Transactions on, 1998, 34(5): 2972-2975..
[20] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. Evolutionary Computation, IEEE Transactions on, 2009, 13(2): 398-417.
[21] Gong M, Ma L, Zhang Q, et al. Community detection in networks by using multiobjective evolutionary algorithm with decomposition[J]. Physica A: Statistical Mechanics and its Applications, 2012, 391(15): 4050-4060.
[22] 初雪宁. 自适应记忆遗传算法研究及在 TSP 问题中的应用[D]. 东北大学, 2012. [23] 陈超. 自适应遗传算法的改进研究及其应用[D]. 广州: 华南理工大学, 2011..
[24] Girvan M, Newman M E J. Community structure in social and biological networks[J]. Proceedings of the national academy of sciences, 2002, 99(12): 7821-7826.
[25] Adamic L A, Lukose R M, Puniyani A R, et al. Search in power-law networks[J]. Physical review E, 2001, 64(4): 046135.
[26] 陈艺璇.基于多目标遗传算法的复杂网络社区划分[D]. 2012.
[27] Holme P, Kim B J. Growing scale-free networks with tunable clustering[J]. Physical review E, 2002, 65(2): 026107.
[28] 周漩, 张凤鸣, 李克武等.利用重要度评价矩阵确定复杂网络关键节点[J]. 物理学报, 2012, 61(5): 1-5.
[29] 赵猛. 约束非线性系统的预测控制方法研究[D]. 重庆大学, 2014.

[30] 陈东明, 徐晓伟. 一种基于广度优先搜索的社区发现方法[J]. 东北大学学报: 自然科学版, 2010, 31(3): 346-349.
[31] Tadic B. Exploring complex graphs by random walks[J]. arXiv preprint cond-mat/0310014, 2003.
[32] Ramos M C. Divisive and hierarchical clustering techniques to analyse variability of rainfall distribution patterns in a Mediterranean region[J]. Atmospheric Research, 2001, 57(2): 123-138.
[33] Orloci L. An agglomerative method for classification of plant communities[J]. The Journal of Ecology, 1967: 193-206.
[34] 王玓. 复杂网络社团检测算法及其应用研究[D]. 西安电子科技大学, 2014.
[35] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Physical review E, 2004, 69(6): 066133..
[36] 肖晓伟, 肖迪, 林锦国等. 多目标优化问题的研究概述倡[J]. 计算机应用研究, 2011, 28(3): 805-809.
[37] 庞峰. 模拟退火算法的原理及算法在优化问题上的应用[D]. 长春: 吉林大学, 2006.
[38] 郭凯熠. 布尔函数设计中爬山算法的研究[D]. 西安电子科技大学, 2010.
[39] 常友渠, 肖贵元, 曾敏. 贪心算法的探讨与研究[J]. 重庆电力高等专科学校学报, 2008, 13(3): 40-42.
[40] Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines[J]. The journal of chemical physics, 1953, 21(6): 1087-1092.
[41]Pizzuti C. A multiobjective genetic algorithm to find communities in complex networks[J]. Evolutionary Computation, IEEE Transactions on, 2012, 16(3): 418-430.
[42] Mukhopadhyay A, Maulik U, Bandyopadhyay S. Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes[J]. Evolutionary Computation, IEEE Transactions on, 2009, 13(5): 991-1005.
[43] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. Systems, Man and Cybernetics, IEEE Transactions on, 1994, 24(4): 656-667.
[44] 殷祚云. Logistic 曲线拟合方法研究[J]. 数理统计与管理, 2002, 21(1): 41-46.
[45] 王志衡, 吴福朝. 均值-标准差描述子与直线匹配[J]. 模式识别与人工智能, 2009, 22(1): 32.
[46] 赵越, 徐鑫, 赵焱等. 自适应记忆遗传算法研究[J]. 计算机技术与发展, 2014, 24(2): 63-66.
[47] Carvalho R, Saldanha R R, Gomes B N, et al. A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas[J]. Magnetics, IEEE Transactions on, 2012, 48(2): 803-806.
[48] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)[C].Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE. IEEE, 2003: 26-33.
[49] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Physical review E, 2004, 69(2): 026113.
[50] Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities[J]. Physical Review E, 2009, 80(1): 016118.
[51] Lusseau D, Schneider K, Boisseau O J, et al. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations[J]. Behavioral Ecology and Sociobiology, 2003, 54(4): 396-405.
[52] Zachary W W. An information flow model for conflict and fission in small groups[J]. Journal of anthropological research, 1977: 452-473.
[53] Schuetz P, Caflisch A. Multistep greedy algorithm identifies community structure in real-world and computer-generated networks[J]. Physical Review E, 2008, 78(2): 026112.
[54] Arenas A, Diaz-Guilera A. Synchronization and modularity in complex networks[J]. The European Physical Journal Special Topics, 2007, 143(1): 19-25.
[55] 陈静. 基于自然计算的复杂网络社区检测[D]. 西安电子科技大学, 2013.

附录 1 攻读硕士学位期间申请的专利

（1）周井泉，陈灵刚，周春霞，姚莹. 社区网络检测的多目标快速遗传算法[P]. 江苏：201610042196.X, 2016.1.22.

附录 2 攻读硕士学位期间参加的科研项目

（1）博士后基金，中国博士后科学基金资助项目（2015M571790）

致谢

我衷心地感激我的导师——周井泉教授，本论文是在周老师手把手的指导下才得以完成，周老师理论深厚，治学严谨，同时为人和善可亲，不仅在学术科研上给予我指导，同时在生活中给学生无微不至的关心和爱护。在周老师手下做课题项目，虽然要求严格，但是这是真正的为学生好，让学生形成严谨的科研态度。我感激周老师传授给我知识与文化，感激周老师给我犯错后的忠告，感激周老师对我的包容。回首看，正是由于周老师严格的要求和认真的教学，我才能像如今这样顺利自信地毕业。另外，由于论文课题原因，经常去请教秦立庆和徐琼师姐，秦师兄和徐师姐不厌其烦地答疑解惑，对我的课题给予了莫大的帮助。同时，我感谢学六 640 宿舍的舍友：屠亮亮、吴子杰、温炜、吉峰和张亚运，感谢有机会能跟他们在一起生活学习。今后，我将更加努力不懈地学习与工作来回报他们。

我要感激父母在这 26 年来给我无微不至的疼爱，在我跌倒时，鼓励、开导和帮助我，让我感受到真挚的亲情。同时，我要感谢同师门的周春霞、王野同学对我的包容、关心和帮助。在王野同学的帮助下，自己才顺利开题。在周春霞同学不厌其烦地指导，自己熟练地掌握了 Matlab 和 Pajek 软件。

我要感谢教研室里面的姜斐、常瑞云、王建同学，谢谢他们在教研室提供了温馨而轻松舒适的学习环境。感谢电子与通信工程 130226 班的徐强，葛斌、谢晶晶、黄辉辉等同学，在找工作期间，给了我技术上的支持，使我感受到了家一样的温暖。最后，我衷心地感激各位评审专家，感激你们在百忙之中腾出时间来细心评阅我的论文。