学校代码： 10246
学　　号： 072021113

硕 士 学 位 论 文

图对称理论在社会网络分析若干重要问题中的应用

院　　系：　计算机科学技术学院
专　　业：　计算机软件与理论
姓　　名：　吴文涛
指 导 教 师：　汪卫　教授
完 成 日 期：　2010 年 5 月 20 日

指导小组成员名单

施伯乐　　教　授

汪　卫　　教　授

何震瀛　　讲　师

肖仰华　　讲　师

目　　录

摘　要

社会网络是对社会中的实体及其之间的关系进行建模的有力工具。许多实际应用问题，如信息处理、分布式搜索、消息传播等，都可以基于社会网络模型来进行研究。因此，近年来，社会网络分析吸引了越来越多研究人员的关注。社会网络分析的兴起也使得图(网络)数据管理成为了当前数据库和数据挖掘领域的一个研究热点。

本文基于图对称理论，对社会网络分析中的若干重要问题进行了研究。图对称理论刻画了网络中的对象(顶点和边)之间在拓扑结构上的等价关系。这种等价关系揭示了蕴含于网络结构中的一种内在的不变性。复杂网络领域最新的研究结果表明，真实网络(包括社会网络)是十分对称的。本文的主要贡献在于首次将网络对称性这一真实网络中普遍存在的重要性质应用到了实际的社会网络分析(尤其是社会网络数据管理) 问题的解决中。

具体而言，本文主要针对社会网络中的实体匿名问题，最短路径查询问题，以及社团挖掘问题进行了深入研究。对于实体匿名问题，本文基于图对称理论提出了可以抵御基于任意结构知识的攻击的k-对称模型，并详细研究了匿名后网络的可用性问题；对于最短路径查询问题，本文基于图对称理论提出了对最短路径索引进行无损压缩的有效算法，大大降低了存储索引所需的空间代价；对于社团挖掘问题，本文基于图对称理论，利用社会网络中的弱邻接现象，提出了一种基于个体之间联系强弱程度的社团挖掘方法，其能够比传统方法发现更多规模较小但语义更丰富的社团。与此同时，本文还简单讨论了图对称理论在社会网络分析其他一些重要问题中的应用，包括顶点重要性刻画，网络健壮性，网络演化，以及子结构模式枚举等问题。

关键词： 社会网络分析, 图对称, 自映射, 实体匿名, 最短路径查询, 社团挖掘
中图分类号：TP311

Abstract

Social network can be used to model entities and their relationships in the society. Recently, social network analysis has attracted more and more researchers since many real application problems could be solved based on social network modeling, such as information processing, distributed search, and diffusion of message. The surge of social network analysis also makes graph (network) data management become one of the hottest research topic within the database and data mining community.

This dissertation gives a study to several important problems in social network analysis, based on graph symmetry theory. Graph symmetry theory describes the equivalence relationship on topological structure between graph objects (vertices and edges). This equivalence relationship characterizes the inherent invariance existing in network structure. Latest results in complex network research shows that, real networks (including social networks) are very symmetric. The major contribution of this dissertation is that it first applies network symmetry, which is a ubiquitous property of real networks, to solve real social network analysis (especially social network data management) problems.

Specifically, this dissertation mainly focuses on the problems of identity anonymization, shortest path query, and community detection in social networks. For identity anonymization problem, we proposed k-symmetry model, which could resist attacks based on *any* structural knowledge, and we also carefully examined the utility of the anonymized network; For shortest path query problem, we proposed lossless algorithms to compress the index structure of shortest paths, which dramatically reduces the storage cost; And for community detection problem, we proposed a novel algorithm by utilizing the tie strength information, based on the inherent relationship between local symmetry and weak-tie phenomenon within social networks. This dissertation also briefly discusses the application of graph symmetry theory on other important topics in social network analysis, including characterization of vertex importance, network robustness, network evolution, and substructure pattern enumeration.

Keywords: social network analysis, graph symmetry, automorphism, identity anonymization, shortest path query, community detection

第一章　绪论

社会网络是一个社会学概念。德国社会学家G. Simmel在1908年发表的《社会学：关于社会交往形式的探讨》中首先提出“社会为相互交织的社会关系”的社会网络思想。到20世纪20-30年代，英国人类学家A. R. Radeliffe-Brown 在社区发展研究中首次使用了”社会网络”(social network)这一术语。1954年，J. A. Barnes正式提出了社会网络的概念，将其定义为由关系所连接的社会实体的网络。对社会网络进行有效分析，能够为证明某些社会学假说提供严格的理论依据，进而为许多实际社会问题的宏观分析奠定技术基础。经过半个多世纪的发展，社会网络分析已经成为现代社会学研究的一种基本手段，并且在人类学、生物学、经济学、地理学、信息科学、组织行为学、社会心理学等领域都有着广泛而深入的应用。

随着时间的推移，社会网络数据的数量和规模都在不断地增长。目前，包含数以万计的社会实体的社会网络已经十分常见。显然，依靠人工手段去处理大规模社会网络数据是不现实的。因此，最近几年来，利用计算机技术分析和研究社会网络数据逐渐成为计算机科学(特别是数据库和数据挖掘)领域的一个研究热点。与其他领域将社会网络作为一种对特定问题进行分析和建模的工具有所不同的是，计算机科学研究领域更注重对实际的社会网络分析和应用提供技术层面上的支持。

首先，社会网络分析能够进行的前提条件是能够获得所需的社会网络数据。随着互联网的迅速发展，目前不少社会网络数据已经可以从互联网上公开获得。然而，许多社会网络数据是带有敏感信息的。例如，一个用于艾滋病传播途径研究的社会网络就真实地记录了网络中个体之间的性接触史。如果这些信息被泄露出去，可能导致数据发布者承担严重的法律后果，最终会导致数据发布者不愿意再发布网络数据，从而影响到社会网络研究的继续深入。传统的数据库隐私保护技术在保护关系型数据的敏感信息泄露方面已经相对成熟，但是社会网络数据是一种图数据模型，与关系数据模型存在着很大的差异。如何在社会网络数据上有效地防止个体隐私泄露目前是一个具有很大挑战性的开放问题，本文第三章将对此进行深入研究。

其次，社会网络分析需要对网络数据有效地进行存储和管理。目前社会网络分析的常用软件工具，如Pajek[1]等，都假设网络数据能够存储在内存中，因此一般只能处理顶点数在10^4数量级以内的网络。然而，实际的社会网络数据规模常常十分庞大。例如MSN好友关系网络的顶点数接近2个亿。因此研究基

于磁盘的网络数据存储和管理方法是十分必要的，这为图[1] 数据管理研究领域提出了新的挑战。尽管图数据库技术的研究已有接近10年的历史，其研究的对象却主要是包含大量小规模图(顶点数一般不超过几百)的数据库。虽然数据库中图的数量很多，但是由于任意的单张图可以载入内存进行处理，因此研究的核心内容是如何根据查询减少需要载入内存的图的数量，一般并不需要对单张图的处理算法进行修改。而在基于磁盘存储的社会网络数据环境下，问题则完全不同。例如，最短路径查询是社会网络应用中最常见的查询之一。查询的基本形式是，给定网络中任意的两个顶点，要求给出网络中连接这两个顶点的一条长度最短的路径。当网络规模较小时，利用宽度优先搜索(无权网络)或者Dijkstra算法(带权网络)[2]即可高效处理这一查询。但是当网络规模很大甚至无法完全载入内存时，实时采用上述算法进行查询处理所需的时间往往无法满足响应时间上的要求。这时就需要有效的数据管理手段(尤其是索引技术)作为支撑，本文第四章将详细研究这一问题。

最后，社会网络分析的大部分任务需要高效的算法。在大规模网络数据上，算法的运行时间和效果是同等重要的。例如，社团挖掘是社会网络分析中的一项非常重要的任务，其研究历史已接近40年。早期研究人员所提出的社团挖掘算法都需要事先指定网络中社团的数量。这在当时的情况下是可行的，因为网络数据的规模都比较小，可以比较容易地通过人工手段鉴别出网络中的社团。因此，这些算法能够在小规模网络数据上工作地很好，却无法应用到现在的大规模社会网络数据上。直到2004年左右，Newman等才逐渐提出一系列可以应用到大规模社会网络数据上的社团挖掘算法，但是其效率和效果并不总是令人满意的。本文第五章将对大规模社会网络数据上的社团挖掘问题进行仔细研究并提出一种启发式的挖掘算法，实验表明该算法在大规模数据上具有不错的效率和效果。

在对上述问题的研究中，我们发现，网络自身结构中蕴含的对称性信息在解决这些问题的过程中扮演着重要角色。所谓对称性，非正式地说，是指网络中存在不同的部分，其在拓扑结构上是完全等价的。如何有效地利用这一等价性信息解决实际问题，成为了本文研究工作的一条主线。例如，本文第四章在处理基于磁盘存储的社会网络数据上的最短路径查询问题时，利用网络自身结构上的等价性信息提出了一种对最短路径索引进行压缩的方法，其思想就是在索引中对于网络中所有结构等价的最短路径，只保留其一个拷贝，从而大大减少了存储索引所需要的空间代价。

事实上，在代数组合学领域，图对称理论的研究已经有大约40多年的历史。概括而言，图对称理论主要利用代数手段研究图的拓扑结构性质，其成果在于

[1] 由于通常用图来对网络(包括社会网络)进行建模，在后文中将不加区别地使用“图”和“网络”。

建立了一套描述图的内部对象(顶点和边)之间的结构等价性关系的理论和方法。然而，在过去的几十年中，图对称理论主要用于帮助化学家研究化学分子的拓扑结构，在数据管理领域并未受到关注。直到最近，在复杂网络(社会网络可以视为复杂网络的一种)研究领域，有研究者发现，真实网络的结构是非常对称的[3]。这一事实与经典的理论结果相悖，因为根据传统理论，几乎所有网络都是非对称的(*Almost all the networks are asymmetric*) [4]。真实网络高度对称性的发现，为利用图对称理论解决社会网络分析中的许多重要问题奠定了基础。本文的研究工作正是在这方面进行了一些初步的探索和尝试。

本文其余部分的组织结构如下：第二章介绍图对称理论的基本内容；第三章至第五章分别针对社会网络分析和应用中的三个重要问题，即实体匿名问题、最短路径查询问题、以及社团挖掘问题，介绍基于图对称理论的解决方案；第六章给出全文的总结，并对图对称理论在其他社会网络分析问题中的应用做进一步的讨论。

第二章　图对称理论概述

本章将介绍图对称理论的基本内容。第一部分将简要回顾图论中的一些基本概念。第二部分介绍图的自映射群理论，这是整个图对称理论的核心内容。第三部分简要介绍基于图对称理论的图的自映射划分的计算方法。第四部分介绍真实网络中对称性研究的最新成果，这是图对称理论能够应用于解决实际问题的关键所在。

2.1　图的基本概念

2.1.1　顶点和边

图*(graph)*可以用一个二元组$G = G(V, E)$来表示，其中V表示图G的顶点集，$E \subseteq V \times V$表示图G的边集[1]。若二元组$(u, v)(u, v \in V)$是有序的，即$(u, v) \neq (v, u)$，则称图G是有向图*(directed graph)*；否则称图G是无向图$(undirected\ graph)$[2]。若$(u, v) \in E$，则称顶点u与顶点v邻接$(adjacent)$，顶点v是顶点u的一个邻居$(neighbor)$。顶点u的所有的邻居的集合记作$N(u)$，$|N(u)|$称为顶点u的度$(degree)$。

2.1.2　路径和环

图G中的一条路径$(path)p$是一个顶点序列$< v_1, v_2, \ldots, v_k >$，其中$v_i \in V(G)\ (1 \leq i \leq k)$且$(v_j, v_{j+1}) \in E(G)\ (1 \leq j < k)$。通常用$v_i \in p$表示顶点$v_i$出现在路径$p$中，用$(v_j, v_{j+1}) \in p$表示边$(v_j, v_{j+1})$出现在路径$p$中。称顶点$v_1$和$v_k$通过路径$p$相连接$(linked)$，并称它们为路径$p$的两个端点$(end)$。路径$p$的长度$(length)|p|$定义为路径$p$中边的数量，亦即$k - 1$。若路径$p$中的所有顶点都不重复，则称路径$p$是简单$(simple)$路径[3]。若$v_1 = v_k$，则称$p$是一个环$(cycle)$。在图$G$中，如果存在一条从顶点$u$到顶点$v$的路径，则称顶点$u$到$v$是可达$(reachable)$的。

在图G中，如果对于任意两个顶点$u, v \in V(G)(u \neq v)$，都存在一条连接顶点u和v的路径，那么称图G是连通的$(connected)$。若图G中不存在自环$(self\ loop)$，即$\forall v \in V(G), (v, v) \notin E(G)$，则称$G$是简单图$(simple\ graph)$[4]。

[1]在后文中，有时也用$V(G)$和$E(G)$分别表示图G的顶点集和边集。
[2]如无特别说明，后文用到术语“图”时，指的都是无向图。
[3]在后文的论述中，若不特别指出，讨论路径时指的都是简单路径。
[4]在后文中，默认情况下讨论图时，都是指连通的简单图。

2.1.3 树

若图G中不存在任何环，则称G为无环图$(acyclic)$。一个连通的无环的无向图称作自由树$(free\ tree)$。可以证明，一个树中的任意两个顶点之间存在唯一路径。如果一棵自由树中的某个顶点被识别出来作为这棵树的根，那么这棵树被称为有根树$(rooted\ tree)$。设有根树T的根为r，给出T中的顶点$v(v \neq r)$；设p是连接v和r的唯一路径，对于任意$u \in p$，$u \neq v$，称u是v的祖先$(ancestor)$，并称v是u的后代$(descendant)$；特别地，若$(u,v) \in E$，则称u是v的父亲$(parent)$，称v是u的孩子$(child)$。如果有根树的每个节点的孩子节点都是有序的，又称这棵树为有序树$(ordered\ tree)$。

2.1.4 子图

给出图H和G, 若$V(H) \subseteq V(G)$且$E(H) \subseteq E(G)$，则称H是G的子图$(subgraph)$，称G是H的父图$(supergraph)$。若图H是图G的子图，且$E(H) = (V(H) \times V(H)) \cap E(G)$，则称$H$是$G$的一个导出子图。换言之，导出子图是保留了其顶点集在父图中的所有边的子图。

2.1.5 图同构

给出图G_1和G_2，若存在一个双射$(bijection)\phi : V(G_1) \to V(G_2)$，使得$(u,v) \in E(G_1)$当且仅当$(\phi(u), \phi(v)) \in E(G_2)$，则称图$G_1$和图$G_2$是同构$(isomorphic)$的，记作$G_1 \cong G_2$。称$\phi$为图$G_1$到$G_2$的同构映射$(isomorphism)$。

2.2 图的自映射群理论

2.2.1 划分和等价关系

集合[5]V的一个划分$(partition)$ 是V的子集的集合$\{V_1, V_2, ..., V_m\}$，并且满足：$(1) V_i \cap V_j = \emptyset$，对于任意$1 \leq i, j \leq m, i \neq j$；$(2) \cup_{1 \leq i \leq m} V_i = V$。每个$V_i$被称作是一个单元$(cell)$。若$V_i$仅包含一个元素，那么称之为平凡单元$(trivial\ cell)$。若划分中的每个单元都是平凡单元，则称该划分为离散划分$(discrete\ partition)$。若划分只包含一个单元(即所有元素都在该单元内)，则称该划分为单位划分$(unit\ partition)$。

集合V上的划分唯一对应于V上的一个等价关系。给定集合V上的任意一个等价关系，都可以得到该等价关系下的划分，通常把这种划分称作等价关系导出$(induced)$的划分。反之亦然。对于集合V上的两个不同的划分P 和Q，如

[5]如无特别说明，本文所指的集合都是有限的。

果P的每个单元都是Q的某个单元的子集，那么称P比Q细致($finer$)，称Q比P粗糙($coarser$)。

2.2.2　置换和置换群

从集合V到其自身的双射称作是集合V上的置换($permutation$)。V上有一个特殊置换，它将V中的每个元素都映射到其自身，这个置换称为单位置换($identity\ permutation$)，通常记作e。集合V上所有的置换的集合记作$S(V)$。给出$g \in S(V)$，对于$v \in V$，称$g(v)$为v在置换g下的像($image$)(通常记为v^g)。

进一步，设$\mathcal{V} = \{V_1, V_2, ..., V_m\}$是顶点集$V$上的一个划分。定义$V_i$在置换$g$下的像为$V_i^g = \{x^g | x \in V_i\}$，并定义$\mathcal{V}$在置换$g$下的像为$\mathcal{V}^g = \{V_1^g, V_2^g, ..., V_m^g\}$。记$\mathcal{V}^g = \mathcal{V}$当且仅当$V_i^g = V_i (1 \leq i \leq m)$。

定义$S(V)$上的(二元)乘积操作($product$)$\circ$为：对于$f, g \in S(V)$，$h = f \circ g$ (通常简记为fg)是满足$x^h = (x^f)^g$的置换$h : V \to V$。对于定义了二元操作的集合，比如$(U, *)$，如果满足下面三个性质，集合U在操作$*$下是一个群($group$)。

1. (结合律)对于任意$x, y, z \in U$，$x * (y * z) = (x * y) * z$；
2. 存在唯一的元素$e \in U$，使得对于任意$x \in V$，$e * x = x * e = x$，e称作幺元($neural\ element$)；
3. 对于任一元素$x \in U$，都存在逆元($inverse\ element$)$x^{-1} \in U$使得$x * x^{-1} = x^{-1} * x = e$。

如果群中的元素都是置换，则称该群为置换群。容易验证，$(S(V), \circ)$是一个置换群，称为对称群($symmetric\ group$)。通常，若$|V| = n$，则将群$(S(V), \circ)$简记为S_n。显然，$|S_n| = n!$。

2.2.2.1　支持集

对于置换$g \in S(V)$，定义其支持集($support\ set$)为$supp(g) = \{v \in V | v^g \neq v\}$，亦即所有被$g$所移动的顶点。

2.2.2.2　可分解性

$S(V)$中的任意置换g都可以表示为一个环[6] 或者若干不相交的环的乘积。反过来，每个环都可以表达$S(V)$中的某个置换[7]。所以任意置换g都可

[6]所谓环是指将$\{1, 2, ..., n\}$的子集$\{i_1, i_2, ..., i_m\}$进行这样的变换：将i_1映射到i_2，i_2映射到i_3,...,i_{m-1}映射到i_m, i_m映射到i_1，通常记为$\binom{i_1, i_2, ..., i_m}{i_2, i_3, ..., i_1}$，或者简记为$(i_1, i_2, ..., i_m)$，见例2.1。

[7]长度为1的环可以表示单位置换e。

以分解为$g = g_1g_2, ..., g_sg_{s+1}, ..., g_t$，这种置换的分解称为完全分解(*complete factorization*)。在不考虑分解的因子间的顺序的前提下，这种分解是唯一的[5, 6]。例2.1给出了对置换进行分解的例子。

例 2.1. 设置换g表示为：

$$g = \begin{pmatrix} 1,2,3,4,5 \\ 1,3,2,5,4 \end{pmatrix}$$

，其中每一列$\binom{i}{j}$表示g将i映射到j。则g的完全分解为

$$g = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2,3 \\ 3,2 \end{pmatrix} \begin{pmatrix} 4,5 \\ 5,4 \end{pmatrix}$$

，简记为$g = (1)(2,3)(4,5)$[8]。若

$$g = \begin{pmatrix} 1,2,3,4,5 \\ 3,5,1,2,4 \end{pmatrix}$$

，则g的完全分解为

$$g = \begin{pmatrix} 1,3 \\ 3,1 \end{pmatrix} \begin{pmatrix} 2,5,4 \\ 5,4,2 \end{pmatrix}$$

，简记为$g = (1,3)(2,5,4)$。

2.2.3 自映射和自映射群

给定图$G(V, E)$，在顶点集V上的某个置换$g \in S(V)$作用下，可以定义置换g对于E的导出作用(*induced action*)为$E^g = \{(u^g, v^g)|(u, v) \in E\}$。若$E^g = E$，则称$g$是图$G$的一个自映射(*automorphism*)。

容易验证，图G的所有自映射的集合$Aut(G) = \{g \in S(V)|E^g = E\}$在置换乘积操作下构成一个群，称为$G$的自映射群(*automorphism group*)。例2.2给出了自映射的一些例子。

例 2.2. 如图2.1所示，置换$g_1 = (1,3)$，$g_2 = (1,3)(2)$，$g_3 = (4,5)(6,8)$，$g_4 = (1,3)(4,5)(6,8)$都是例图G的自映射。

如果自映射$f \in Aut(G)(f \neq e)$能够表示为$f = f_1f_2(f_1, f_2 \in Aut(G))$且$supp(f_1) \cap supp(f_2) = \emptyset \Rightarrow f_1 = e$或者$f_2 = e$，那么称$f$是不可分解的(*indecomposable*)；否则称f是可分解的(*decomposable*)。容易验证，在例2.2中，g_1和g_3都是不可分解的，g_4可以分解为g_1g_3。

[8]通常省略长度为1的环，因此g可以进一步简记为$g = (2,3)(4,5)$。

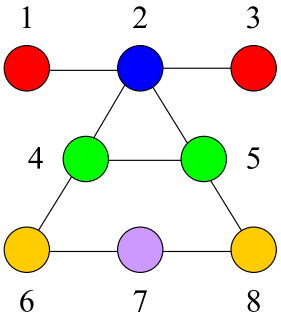


图 2.1: 例图G

2.2.4　自映射划分和轨道

给定图$G(V,E)$的两个顶点u,v，如果存在某个$g \in Aut(G)$使得$u^g = v$，那么称u和v是自映射等价$(automorphically\ equivalent)$的，记作$u \sim v$。容易验证，自映射等价关系是顶点集$V$上一个等价关系，因此对应了顶点集$V$上的一个划分，称为自映射划分$(automorphism\ partition)$，一般记为$Orb(G)$。自映射划分中的每个单元，称为$Aut(G)$的轨道$(orbit)$。容易验证，例2.2所示的例图$G$的自映射划分为$Orb(G) = \{\{1,3\},\{2\},\{4,5\},\{6,8\},\{7\}\}$。

进一步，对于任意的$g \in Aut(G)$，有$Orb(G)^g = Orb(G)$；并且$Orb(G)$是满足这一性质的最为粗糙的划分，即不存在V的划分$\mathcal{V} \neq Orb(G)$，$\mathcal{V}$比$Orb(G)$粗糙，且对任意的$g \in Aut(G)$，有$\mathcal{V}^g = \mathcal{V}$。

2.3　自映射划分的计算方法

可以证明，计算自映射划分的问题复杂度和图同构判定的复杂度是多项式等价的[7]。由于图同构判定问题(GI)的复杂度仍然是一个未决问题[8][9]，因此在最坏情况下其时间复杂度是指数级的。实践中最为广泛采用的是nauty算法 [10]，其运行效率是非常高的。在不要求完全精确地计算出$Orb(G)$的情况下，图G的完全度划分$(total\ degree\ partition)$$TDP(G)$提供了对$Orb(G)$的非常好的近似，而计算$TDP(G)$的算法的时间复杂度是多项式级的[11][10]。

2.4　图对称及真实网络中的对称性

在代数图论领域，对于图对称有如下定义[4]：

定义 2.1 (图对称). 如果图G中存在一个自映射$g \neq e$，则称G是对称的$(symmetric)$，否则称G是不对称的 $(asymmetric)$。

[9]具体而言，容易知道GI是NP的，但至今未能找到GI的多项式时间算法，也未能证明其为NP完全的[9]。

[10]更为通用的计算近似自映射划分的方法见[12]。

当图的规模足够大时，有如下结论成立[4]：

定理 2.1. *“几乎所有的图都是非对称的(almost all graphs are asymmetric)”*。

定理2.1表明，大规模的随机网络都是趋于不对称的。换句话说，当网络足够大时，找到对称网络的概率是相当低的。然而，复杂网络领域最新的研究结果[3, 13]表明，绝大多数真实网络却是非常对称的。这表明真实网络的演化规律和传统的随机网络模型(ER模型[14, 15])的生成机制是完全不同的，并且真实网络中的对称也不同于著名的BA模型[16]所生成的网络中的所谓的“树形对称”。为了进一步研究真实网络中对称性的产生机制，文献[17] 提出了一种基于相似链接模式(*similar linkage pattern*)的网络生成模型，它可以产生与真实网络中的对称性非常接近的模拟网络。“真实网络(包括社会网络) 是非常对称的”这一事实的发现，为基于图对称理论解决实际的社会网络分析问题奠定了基础。

第三章　社会网络中的实体匿名问题与k-对称模型

社会网络数据中往往包含着社会实体的个人敏感信息。例如，一个用于研究传染病传播路径的社会网络记录了网络中个体之间的接触关系。随着越来越多的社会网络数据的发布，如何防止社会网络中的个人隐私泄漏成为了网络数据发布者非常关心的问题。实体匿名问题是目前社会网络隐私保护的主要研究方向之一，它考虑的是如何防止攻击者基于掌握的结构信息从网络中唯一识别出社会实体。之前的研究工作通常只考虑对基于某种特定的结构信息的攻击进行防范的匿名模型，本章提出的基于图对称理论的k-对称模型则能够防范基于任意结构信息的攻击。

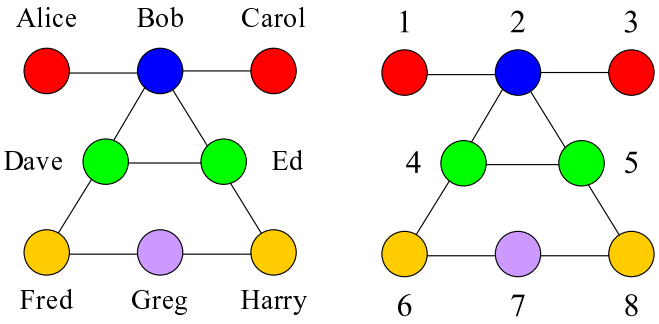
3.1　研究背景

一种最简单的实体匿名方法是将网络中个体的身份识别信息隐去，或者代之以随机产生的数字。这种匿名方法被称为*幼稚匿名*(*naive anonymization*)，它使得攻击者无法直接通过身份识别信息来鉴别出网络中的实体。然而，Backtrom等 [18]在2007年的WWW会议上指出，幼稚匿名无法抵御基于*结构知识* (*structural knowledge*)的攻击(结构知识是指攻击者所掌握的针对网络中某个个体在网络中的拓扑结构信息，比如个体所对应的顶点的度等信息)，见例3.1。

例 3.1 (基于结构知识的攻击). 如图3.1所示，3.1(a)是一个社会网络，3.1(b)是其经过幼稚匿名后的版本。假设攻击者针对Bob进行攻击(即识别出哪个顶点是Bob)，并且具有结构知识P_1：*Bob至少在网络中有3个邻居*，则Bob只可能是顶点2，4，5中的一个。如果攻击者具有结构知识P_2：Bob有两个度为1的邻居，那么Bob只可能是顶点2。在这种情况下，攻击者就成功地从幼稚匿名后的网络中重新识别(re-identify)出了Bob。

一些研究者随后针对经过幼稚匿名后的网络如何抵御基于结构知识的攻击的问题展开了研究，并提出了相应的匿名模型，如k-度匿名 [19]以及k-邻域匿名 [20]。但是这些模型都是针对特定形式的结构知识提出的。由于社会网络发布者很难在发布网络之前知道攻击者掌握了何种形式的结构知识，因此上述模型都只能抵御他们所假定的攻击方式。

于是产生了一个很自然的问题：是否能够找到一种独立于结构知识的匿名模型，使得任意基于结构知识的攻击失效？基于图对称理论的知识我们不难发现如下的重要事实：*自映射等价的顶点在结构上是无法相互区别的*。换句话说，*任意基于结构知识的攻击无法区分两个自映射等价的顶点*。k-对称模型正是基



(a) G (b) G_a

图 3.1: 社会网络G及其经过幼稚匿名后的版本G_a

于这一事实来对上述问题作出肯定的回答的，其基本的思路为修改原始网络(加入新的点和边)，使得得到的新网络其自映射划分的每个轨道至少包含k个顶点。由于位于同一个轨道内的顶点是自映射等价的，攻击者无法区分出同一个轨道中的任意两个顶点，无论他掌握了何种结构知识。所以攻击者识别出新网络中任何一个个体的概率不会超过$\frac{1}{k}$。下面详细介绍k-对称模型。

3.2 k-对称模型

首先给出k-对称的严格定义。

定义 3.1 (k-对称匿名). 给出图G和正整数k，设G的自映射划分为$Orb(G)$，如果对任意的轨道$\Delta \in Orb(G)$都有$|\Delta| \geq k$，则称G是k-对称的，或者说G是满足k-对称匿名要求的。

在上述定义下，问题于是可以描述为：给出图G和正整数k，如何修改G使得得到的图G'是k-对称的。

3.2.1 轨道拷贝操作

由于位于同一个轨道中的顶点已经相互之间是自映射等价的，一个基本想法是对$Orb(G)$中的每个轨道进行拷贝，直到每个轨道连同其拷贝的并集合至少包含k个顶点。然而，这样的拷贝并非如此直接了当，因为需要保证在这个并集合中的所有顶点之间仍然是自映射等价的。下面介绍的轨道拷贝操作(*Orbit Copying Operation*)能够达到这一要求。

在引入轨道拷贝操作的概念之前，需要先对自映射划分的概念进行进一步泛化，为此引入子自映射划分(*sub-automorphism partition*)的概念。它是后文对轨道拷贝操作进行理论分析的基础。

定义 3.2 (子自映射划分). 给出图G，设$\mathcal{V}$是G的顶点集$V(G)$的划分，$Aut(G)$为G的自映射群，满足$\forall O \in \mathcal{V}$以及$\forall u, v \in O$，$\exists g \in Aut(G)$满足$u^g = v$且$\mathcal{V}^g = \mathcal{V}$，则称$\mathcal{V}$是$G$的一个子自映射划分。

显然，若$\mathcal{V}$是G的一个子自映射划分，那么$\mathcal{V}$比$Orb(G)$细致，即，对于任意的$V_i \in \mathcal{V}$, 存在某个$\Delta_j \in Orb(G)$，使得$V_i \subseteq \Delta_j$。注意，$Orb(G)$显然也是G的一个子自映射划分，所以可以把子自映射划分看成是自映射划分的一种泛化。但是需要指出的是，并非所有比$Orb(G)$细致的划分都是子自映射划分，见例3.2。

例 3.2 (子自映射划分). 考虑一个4个顶点的环，即顶点集为$\{1, 2, 3, 4\}$, 边集为$\{(1, 2)(2, 3)(3, 4)(1, 4)\}$。其自映射划分是单位划分$\{\{1, 2, 3, 4\}\}$。容易验证，$\{\{1, 2\}, \{3, 4\}\}$是一个子自映射划分。但是$\{\{1, 2, 3\}, \{4\}\}$则不是一个子自映射划分，因为无法找到将顶点2映射到顶点3且同时能够使得该划分保持不变的自映射。

下面给出轨道拷贝操作的定义：

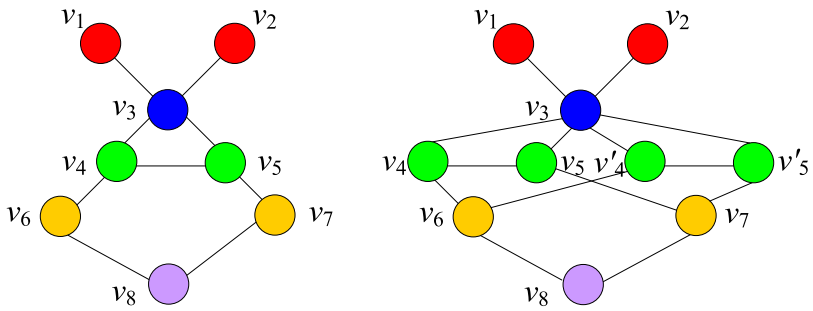
定义 3.3 (轨道拷贝操作). 给出图G以及G的一个子自映射划分$\mathcal{V}$，设$V \in \mathcal{V}$，定义轨道拷贝操作$Ocp(G, \mathcal{V}, V)$如下：
对于每个顶点$v \in V$，在G中插入一个新的顶点v'，并且
(1)若$(u, v) \in E(G)$，$u \in U$，$U \in \mathcal{V}$且$U \neq V$，则在G中插入一条新的边(u, v')；
(2)若$(u, v) \in E(G)$，$u \in V$，则在G中插入一条新的边(u', v')。

例3.3给出了轨道拷贝操作的一个示例。

例 3.3 (轨道拷贝操作). 如图3.2所示，图3.2(a)为一个幼稚匿名后的社会网络，其自映射划分为$Orb(G) = \{V_1, V_2, V_3, V_4, V_5\}$，其中$V_1 = \{v_1, v_2\}$, $V_2 = \{v_3\}$, $V_3 = \{v_4, v_5\}$, $V_4 = \{v_6, v_7\}$，$V_5 = \{v_8\}$。图3.2(b)展示了V_3被拷贝之后得到的图。

轨道拷贝操作的核心思想在于被拷贝的轨道的每个拷贝会保留住被拷贝的轨道与其他轨道之间的邻接关系。例如，在例3.3中，V_3' 与V_2和V_4仍然是邻接的。下面我们证明，被拷贝的轨道及其拷贝的并集合中的顶点是自映射等价的(引理3.1)。

引理 3.1. *给定图G和它的一个子自映射划分$\mathcal{V} = \{V_1, V_2, ..., V_m\}$。经过轨道拷贝操作$Ocp(G, \mathcal{V}, V_i)(1 \leq i \leq m)$后的划分$\mathcal{V}^{(1)} = \{V_1, V_2, ..., V_i^{(1)}, ..., V_m\}$是得到的图$G^{(1)}$的子自映射划分。这里$V_i^{(1)}$是$V_i$及其拷贝$V_i$'的并集合。*



(a) 图G (b) V_3被拷贝后的图G'

图 3.2: 轨道拷贝操作示例

证明. 我们需要证明，对于每个$V \in \mathcal{V}^{(1)}$，对$\forall u, v \in V$，存在$h \in Aut(G^{(1)})$，满足$u^h = v$且$(\mathcal{V}^{(1)})^h = \mathcal{V}^{(1)}$。我们首先考虑$V = V_i^{(1)}$的情况。

设$V_i = \{v_1, v_2, ..., v_s\}$，则$V_i^{(1)}$中的顶点可以被表示为$V_i^{(1)} = \{v_1, v_2, ..., v_s, v'_1, v'_2, ..., v'_s\}$。为了记号上的方便起见，我们用$W_i$表示集合$\{v_1, v_2, ..., v_s\}(W_i = V_i)$，用$W'_i$表示集合$\{v'_1, v'_2, ..., v'_s\}(W'_i = V'_i)$，于是有$V_i^{(1)} = W_i \cup W'_i$，$W_i \cap W'_i = \emptyset$。进一步，有$V(G^{(1)}) = V(G) \cup W'_i$。定义$f : W_i \to W'_i$，满足对于$1 \le j \le s$，有$f(v_j) = v'_j$。易知$f$为双射。用$f^{-1}$表示$f$的逆映射。

令u和v是$V_i^{(1)}$中的任意两个顶点。不失一般性，有以下两种情况需要考虑：u和v都在W_i中；u和v分别在W_i和W'_i中。下面分别对这两种情况进行考虑。证明的基本思路是，无论何种情况，我们构造$G^{(1)}$的自映射$h \in Aut(G^{(1)})$，满足$u^h = v$且$(\mathcal{V}^{(1)})^h = \mathcal{V}^{(1)}$。

(1) *u和v都在W_i中*[1]。

由于$W_i = V_i$，因此存在$g \in Aut(G)$，满足$u^g = v$ and $\mathcal{V}^g = \mathcal{V}$(因为$\mathcal{V}$ 是G的子自映射划分)。现构造$V(G^{(1)})$上的置换h，h在$V(G)$上的作用与g在$V(G)$上的作用相同，并且对于$1 \le j \le s$，满足$(v'_j)^h = f(v_j^g)$(显然$v_j^g \in W_i = V_i$，且$u^h = u^g = v$)。下面证明$h \in Aut(G^{(1)})$。

设$e = (a, b) \in E(G^{(1)})$为$G^{(1)}$的任意一条边，需要证明$e^h = (a^h, b^h) \in E(G^{(1)})$。分三种情况讨论如下：

(1.1) $e \in E(G)$。则有$a \in V(G)$且$b \in V(G)$，并且$e^h = (a^h, b^h) = (a^g, b^g) = e^g \in E(G) \subseteq E(G^{(1)})$。

(1.2) $e \notin E(G)$，但是$a \in V(G)$且$b \in W'_i$ ($b \in V(G)$且$a \in W'_i$的情况是相同的)。则有$e^h = (a^h, b^h) = (a^g, f((f^{-1}(b))^g))$。因为$b \in W'_i$, $f^{-1}(b) \in W_i = V_i$，因此$(a, f^{-1}(b)) \in E(G)$(否则$(a, b) \notin E(G^{(1)})$)。所以，我们有$(a^g, (f^{-1}(b))^g) \in E(G)$。进一步，因为$\mathcal{V}^g = \mathcal{V}$，有$(f^{-1}(b))^g \in V_i$。于是，$(a^g, f((f^{-1}(b))^g)) \in$

[1] u和v都在W'中的情况可以用同样的方法进行证明。

$E(G^{(1)})$。

(1.3) $e \notin E(G)$, 且a和b都在W_i'中。则有$e^h = (a^h, b^h) = (f((f^{-1}(a))^g), f((f^{-1}(b))^g))$。因为$f^{-1}(a) \in V_i$，$f^{-1}(b) \in V_i$，我们有$(f^{-1}(a), f^{-1}(b)) \in E(G)$，否则$(a,b) \notin E(G^{(1)})$。因此，$((f^{-1}(a))^g, (f^{-1}(b))^g) \in E(G)$，从而我们有$(f((f^{-1}(a))^g), f((f^{-1}(b))^g)) \in E(G^{(1)})$(因为$\mathcal{V}^g = \mathcal{V}$，所以$(f^{-1}(a))^g$ 和$(f^{-1}(b))^g$都在V_i中)。

接着，我们证明$(\mathcal{V}^{(1)})^h = \mathcal{V}^{(1)}$。由于$h$在$V(G)$上的作用和$g$相同，并且$\mathcal{V}^g = \mathcal{V}$，我们只需要证明$(V_i^{(1)})^h = V_i^{(1)}$。根据$h$的定义，我们有$(V_i^{(1)})^h = \{v_1^h, ..., v_s^h, (v_1')^h, ..., (v_s')^h\} = \{v_1^g, ..., v_s^g, f(v_1^g), ..., f(v_s^g)\}$。因此$(V_i^{(1)})^h = W_i \cup W_i' = V_i^{(1)}$，情况(1)证毕。

(2) u和v分别在W_i和W_i'中[2]。

因为$v \in W_i'$，我们有$f^{-1}(v) \in W_i$。于是根据情况(1)的证明过程，存在自映射$h_1 \in Aut(G^{(1)})$，满足$u^{h_1} = f^{-1}(v)$且$(\mathcal{V}^{(1)})^{h_1} = \mathcal{V}^{(1)}$。现构造$V(G^{(1)})$上的置换$h_2$，满足对于$1 \le j \le s$，$v_j^{h_2} = v_j'$且$(v_j')^{h_2} = v_j$；对于任意$v \in (V(G^{(1)}) \setminus V_i^{(1)})$，$v^{h_2} = v$。令$h = h_1 \cdot h_2$，我们有$u^h = u^{h_1 h_2} = (u^{h_1})^{h_2} = (f^{-1}(v))^{h_2} = f(f^{-1}(v)) = v$。下面证明$h \in Aut(G^{(1)})$，且$(\mathcal{V}^{(1)})^h = \mathcal{V}^{(1)}$。由于$h_1 \in Aut(G^{(1)})$，且$(\mathcal{V}^{(1)})^{h_2} = \mathcal{V}^{(1)}$，因此只需要证明$h_2 \in Aut(G^{(1)})$以及$(\mathcal{V}^{(1)})^{h_2} = \mathcal{V}^{(1)}$[3]。为了证明这一点，我们只需要证明，对于每条边$e = (a,b) \in E(G^{(1)})$，有$e^{h_2} = (a,b)^{h_2} \in E(G^{(1)})$成立。事实上，由于$h_2$只是互换$W_i$和$W_i'$中相应的顶点，我们只需要考虑那些至少有一个端点在$W_i$或者$W_i'$中的顶点即可。于是有两种情况：

(2.1) a和b都在W_i中[4]。则$e^{h_2} = (a^{h_2}, b^{h_2}) = (f(a), f(b)) \in E(G^{(1)})$。

(2.2) $a \in W_i$，$b \in (V(G^{(1)}) \setminus V_i^{(1)})$[5]。则$e^{h_2} = (a^{h_2}, b^{h_2}) = (f(a), b) \in E(G^{(1)})$。

注意，根据轨道拷贝操作的定义，$a \in W_i$且$b \in W_i'$ (或者$a \in W_i'$且$b \in W_i$)的情况是不可能的。因此，我们证明了$h_2 \in Aut(G^{(1)})$，从而$h \in Aut(G^{(1)})$。由于h_2只在$V_i^{(1)}$上进行置换，为了证明$(\mathcal{V}^{(1)})^{h_2} = \mathcal{V}^{(1)}$，我们只需要证明$(V_i^{(1)})^{h_2} = V_i^{(1)}$。事实上，我们有$(V_i^{(1)})^{h_2} = \{v_1^{h_2}, ..., v_s^{h_2}, (v_1')^{h_2}, ..., (v_s')^{h_2}\} = \{v_1', ..., v_s', v_1, ..., v_s\} = V_i^{(1)}$，从而完成了对情况(2)的证明。

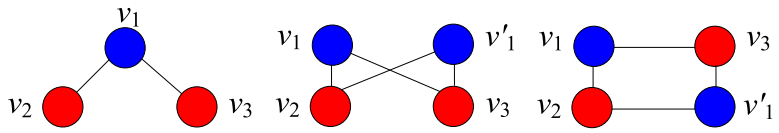
最后，我们考虑$V \neq V_i^{(1)}$的情况。则此时，$\forall u, v \in V$，$\exists g \in Aut(G)$，满足$u^g = v$且$\mathcal{V}^g = \mathcal{V}$。构造$V(G^{(1)})$上的置换$h$，$h$在$V(G)$上的作用和$g$相同，并且$(v_j')^h = f(v_j^g)$。根据上面对情况(1)的证明过程可知，这样构造的$h$是$G^{(1)}$的一

[2] $u \in W_i'$且$v \in W_i$ 的情况可以用同样的方法证明

[3] 因为$Aut(G^{(1)})$是群，所以$h = h_1 \cdot h_2$必然也是自映射，并且有$(\mathcal{V}^{(1)})^h = ((\mathcal{V}^{(1)})^{h_1})^{h_2} = (\mathcal{V}^{(1)})^{h_2}$

[4] a和b都在W_i'中的情况同此证明

[5] $a \in W_i'$ and $b \in (V(G^{(1)}) \setminus V_i^{(1)})$的情况同此证明



(a) Graph G (b) G' (c) G''

图 3.3: 一个$\mathcal{V}' \neq Orb(G')$的例子

个自映射，且满足$(\mathcal{V}^{(1)})^h = \mathcal{V}^{(1)}$。显然有$u^h = u^g = v$。至此我们完成了对引理3.1 的证明。 □

这里需要指出是，引理3.1表明，在对$Orb(G)$的某个轨道V进行一次轨道拷贝操作$Ocp(G, \mathcal{V}, V)$得到的图G'中，一定存在某个$Orb(G')$的轨道Δ，满足$V \cup V' \subseteq \Delta$，这里$V'$是$V$的拷贝。然而引理3.1没有说明$V \cup V'$一定是$Orb(G')$的某个轨道，事实上这是不一定成立的。例3.4给出了一个反例。而在图3.2所示的例子中，$V' = Orb(G')$是成立的。

例 3.4 ($\mathcal{V}'$和$Orb(G')$). 如图3.3所示，这里$\mathcal{V} = Orb(G) = \{V_1, V_2\}$($V_1 = \{v_1\}$, $V_2 = \{v_2, v_3\}$)。拷贝V_1后，得到G'和划分$\mathcal{V}' = \{\{v_1, v'_1\}, \{v_2, v_3\}\}$。如果将$G'$重绘为$G''$，容易发现$G'$事实上是一个4个顶点的环，$Orb(G')$是单位划分。因此，$\mathcal{V}' \neq Orb(G')$。

由于目标是要使得修改后的网络的每个轨道至少包含k个顶点，一次轨道拷贝操作并不足以做到这一点。引理3.2是对引理3.1的扩展，它说明，对某一轨道重复进行轨道拷贝操作后得到的划分仍然是修改后的图的一个子自映射划分。

引理 3.2. *给定图G和它的一个子自映射划分$\mathcal{V} = \{V_1, V_2, ..., V_m\}$。设对同一单元$V_i (1 \leq i \leq m)$进行$N \geq 0$次轨道拷贝操作$Ocp(G, \mathcal{V}, V_i)$后的划分为$\mathcal{V}^{(N)} = \{V_1, V_2, ..., V_i^{(N)}, ..., V_m\}$，相应的图为$G^{(N)}$(这里$V_i^{(N)}$是$V_i$及其所有拷贝的并集合，特别地，$V_i^{(0)} = V_i$)，则$\mathcal{V}^{(N)}$是$G^{(N)}$的一个子自映射划分。*

证明. 若$N = 0$，则$G_i^{(0)} = G$且$\mathcal{V}_i^{(0)} = \mathcal{V}$，引理显然成立。现在设$N \geq 1$，令$V_i = \{v_1, v_2, ..., v_s\}$，$V_i^{(N)} = \{v_1, ..., v_s, v_1^{(1)}, ..., v_s^{(1)}, ..., v_1^{(N)}, ..., v_s^{(N)}\}$，这里$v_1^{(n)}, ..., v_s^{(n)}$是在第$n$次操作中拷贝$v_1, ..., v_s$得到的相应的顶点$(1 \leq n \leq N)$。和引理3.1的证明类似，我们用记号$W_i^{(0)} = V_i$表示集合$\{v_1, ..., v_s\}$，用$W_i^{(n)}$表示集合$\{v_1^{(n)}, ..., v_s^{(n)}\}$。同样，对于每个$1 \leq n \leq N$，我们定义映射$f_n : W_i^{(0)} \to W_i^{(n)}$，满足$f_n(v_j) = v_j^{(n)} (1 \leq j \leq s)$。为了简便起见，我们定义$f_0 : W_i^{(0)} \to W_i^{(0)}$为$W_i^{(0)}$上的恒等映射，即$f_0(v_j) = v_j (1 \leq j \leq s)$。显然每个$f_n$均为双射。为了证明引理，我们需要证明，对每个单元$V \in \mathcal{V}_i^{(N)}$，$\forall u, v \in V$，存在$h \in Aut(G_i^{(N)})$满

足$u^h = v$且$(\mathcal{V}_i^{(N)})^h = \mathcal{V}_i^{(N)}$。由于证明过程和引理3.1 非常相似，这里我们只给出大致的步骤，不再介绍具体的细节。

我们首先考虑$V = V_i^{(N)}$的情况，显然同样有两种情况，即是否u和v处于同一个$W_i^{(j)}$中$(0 \leq j \leq N)$。在第一种情况下，不失一般性，设$u, v \in W_i^{(0)}$。于是存在$g \in Aut(G)$，满足$u^g = v$且$\mathcal{V}^g = \mathcal{V}$。然后我们构造$V(G_i^{(N)})$上的自映射$h$，$h$在$V(G)$上的作用和$g$相同，并且$(v_j^{(n)})^h = f_n(v_j^g)$，对于$1 \leq j \leq s$，$1 \leq n \leq N$。可以证明$h \in Aut(G_i^{(N)})$且$(\mathcal{V}_i^{(N)})^h = \mathcal{V}_i^{(N)}$。在第二种情况下，不失一般性，设$u \in W_i^{(p)}$，$v \in W_i^{(q)}$，$p \neq q$。令$u_0 = f_p^{-1}(u)$，$v_0 = f_q^{-1}(v)$，则$u_0 \in W_i^{(0)}$，$v_0 \in W_i^{(0)}$。于是存在$g \in Aut(G)$，满足$u_0^g = v_0$且$\mathcal{V}^g = \mathcal{V}$。用和在第一种情况下相同的方法构造$h_1 \in Aut(G_i^{(N)})$。构造$V(G_i^{(N)})$上的置换$h_2$，$h_2$仅交换$W_i^{(p)}$和$W_i^{(q)}$相应的顶点，而保持$V(G_i^{(N)})$中其他的顶点不变。正式地，$(v_j^{(p)})^{h_2} = v_j^{(q)}$，$(v_j^{(q)})^{h_2} = v_j^{(p)}$，且$v^{h_2} = v$，$\forall v \in V(G_i^{(N)}) \setminus V_i^{(N)}$。然后可以证明$h_2 \in Aut(G_i^{(N)})$且$(\mathcal{V}_i^{(N)})^{h_2} = \mathcal{V}_i^{(N)}$。令$h = h_1 \cdot h_2$，则$h \in Aut(G_i^{(N)})$，$(\mathcal{V}_i^{(N)})^h = \mathcal{V}_i^{(N)}$，且$u^h = u^{h_1 h_2} = (u^{h_1})^{h_2} = (f_p(v_0))^{h_2} = f_q(v_0) = v$。

最后考虑$V \neq V_i^{(N)}$的情况，此时$\forall u, v \in V$，存在$g \in Aut(G)$，满足$u^g = v$且$\mathcal{V}^g = \mathcal{V}$。我们可以用和证明上面第一种情况$(V = V_i^{(N)})$时相同的方法构造$h \in Aut(G_i^{(N)})$，于是有$u^h = u^g = v$且$(\mathcal{V}_i^{(N)})^h = \mathcal{V}_i^{(N)}$，从而完成了对引理3.2的证明。□

引理3.2仍然只涉及到单个单元的轨道拷贝操作，但是为了达到k-对称匿名的要求，可能需要对初始划分中的多个单元都进行轨道拷贝操作。因此一个重要的问题便是：交换轨道拷贝操作的顺序是否会产生不同的结果图和划分？正式地，设$\mathbf{O} = O_1 ... O_N$是任意的一个作用于G的长度为N的轨道拷贝操作的序列，这里$O_n = Ocp(G, \mathcal{V}, V_{i_n})(1 \leq n \leq N)$，并设$\mathbf{O}$产生的结果图为$G_{\mathbf{O}}$。令$\pi$为集合$\{1, 2, ..., N\}$上的任意一个置换，并设$\mathbf{O}$在置换$\pi$下的轨道拷贝操作序列为$\pi(\mathbf{O}) = O_{\pi(1)} ... O_{\pi(N)}$，这里$O_{\pi(n)} = Ocp_{\pi(n)}(G, \mathcal{V}, V_{i_{\pi(n)}})(1 \leq n \leq N)$。引理3.4表明，轨道拷贝操作是顺序无关的。为此，我们首先证明一个更为基本的结果(见引理3.3)。

引理 3.3. *给定图G和它的一个子自映射划分$\mathcal{V} = \{V_1, V_2, ..., V_m\}$。设$Ocp_1(G, \mathcal{V}, V_{i_1}), ..., Ocp_N(G, \mathcal{V}, V_{i_N})$为任意作用于$G$上的轨道拷贝操作序列，这里$N \geq 1$，$i_n \in \{1, 2, ..., m\}$，$1 \leq n \leq N$，并设得到的图为$G_N$。若互换任意两个相继的轨道拷贝操作的次序(称为操作对换(*operation transposition*))，不失一般性，设为$Ocp_j(G, \mathcal{V}, V_{i_j})$和$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})(1 \leq j \leq N-1)$，并设这样得到的图为$G'_N$，则$G_N$和$G'_N$是同构的$(G_N \cong G'_N)$。*

证明. 设两种操作次序下，经过第n次操作后的图分别为G_n和G'_n，并设相应的顶点集上的划分为$\mathcal{V}^{(n)}$和$(\mathcal{V}^{(n)})'(1 \leq n \leq N)$。为了证明引理，我们只需要证明$G_{j+1} \cong G'_{j+1}$。下面分两种情况讨论：

(1) $i_j = i_{j+1}$。则$Ocp_j(G, \mathcal{V}, V_{i_j})$和$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$事实上作用于同一个单元。因此显然$G_{j+1}$和$G'_{j+1}$是同构的。

(2) $i_j \neq i_{j+1}$。令$V_{i_j} = \{v_1, ..., v_s\}$，$V_{i_{j+1}} = \{u_1, ..., u_t\}$，并且令$\mathcal{V}^{(j-1)} = (\mathcal{V}^{(j-1)})' = \{V_1^{(j-1)}, ..., V_m^{(j-1)}\}$。我们设$V_{i_j}^{(j-1)} = \{v_1, ..., v_s, v_1^{(1)}, ..., v_s^{(1)}, ..., v_1^{(p-1)}, ..., v_s^{(p-1)}\}$，且$V_{i_{j+1}}^{(j-1)} = \{u_1, ..., u_t, u_1^{(1)}, ..., u_t^{(1)}, ..., u_1^{(q-1)}, ..., u_s^{(q-1)}\}$，这里$p, q \geq 1$为正整数。当两个操作以$Ocp_j(G, \mathcal{V}, V_{i_j})$和$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$的次序进行后，我们设$V_{i_j}^{(j+1)} = \{v_1, ..., v_s, v_1^{(1)}, ..., v_s^{(1)}, ..., v_1^{(p)}, ..., v_s^{(p)}\}$，以及$V_{i_{j+1}}^{(j+1)} = \{u_1, ..., u_t, u_1^{(1)}, ..., u_t^{(1)}, ..., u_1^{(q)}, ..., u_t^{(q)}\}$。若互换次序，则我们设$V_{i_j}^{(j+1)} = \{v_1, ..., v_s, v_1^{(1)}, ..., v_s^{(1)}, ..., (v_1^{(p)})', ..., (v_s^{(p)})'\}$，以及$V_{i_{j+1}}^{(j+1)} = \{u_1, ..., u_t, u_1^{(1)}, ..., u_t^{(1)}, ..., (u_1^{(q)})', ..., (u_t^{(q)})'\}$。现构造从$V(G_{j+1})$到$V(G'_{j+1})$的映射$f$，满足$f(v_k^{(p)}) = (v_k^{(p)})'$，$f(u_l^{(q)}) = (u_l^{(q)})'(1 \leq k \leq s, 1 \leq l \leq t)$，且$f(v) = v$，对任意$v \in V(G_{j-1}) = V(G'_{j-1})$。下面我们证明$f$是$G_{j+1}$和$G'_{j+1}$之间的同构映射。

首先，显然f是双射。其次，为了证明f是一个同构映射，我们需要证明，对于每个$e = (u, v) \in E(G_{j+1})$，$(f(u), f(v)) \in E(G'_{j+1})$。分别对$e$的四种不同的情况进行讨论：

(2.1) u和v都在$V(G_{j-1})$中。则$(f(u), f(v)) = (u, v) \in E(G_{j-1})$，因此有$(f(u), f(v)) \in E(G'_{j+1})$(因为轨道拷贝操作不会删除图中的任何边)。

(2.2) $u \in V(G_{j-1})$且$v \in \{v_1^{(p)}, ..., v_s^{(p)}\}$(或者$u \in V(G_{j-1})$且$v \in \{u_1^{(q)}, ..., u_t^{(q)}\}$)。则$v$是通过轨道拷贝操作$Ocp_j(G, \mathcal{V}, V_{i_j})$加入的。不失一般性，设$v = v_k^{(p)}$，则$(f(u), f(v)) = (u, (v_k^{(p)})')$。由于当两次操作的次序为$Ocp_j(G, \mathcal{V}, V_{i_j})$，$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$时，我们通过$Ocp_j(G, \mathcal{V}, V_{i_j})$加入$(u, v)$是因为$(u, v_k) \in E(G_{j-1})$，因此当我们互换两次操作的次序后，$(u, (v_k^{(p)})')$必定也会在作用$Ocp_j(G, \mathcal{V}, V_{i_j})$时被加入。

(2.3) u和v都在$\{v_1^{(p)}, ..., v_s^{(p)}\}$中(或者$u$和$v$都在$\{u_1^{(q)}, ..., u_t^{(q)}\}$中)。令$u = v_l^{(p)}$，$v = v_k^{(p)}$，则$(f(u), f(v)) = ((v_l^{(p)})', (v_k^{(p)})')$。由于当两次操作的次序为$Ocp_j(G, \mathcal{V}, V_{i_j})$，$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$时，我们通过$Ocp_j(G, \mathcal{V}, V_{i_j})$加入$(u, v)$是因为$(v_l, v_k) \in E(G_{j-1})$(更确切地说，$(v_l, v_k) \in E(G)$)，因此当我们互换两次操作的次序后，$((v_l^{(p)})', (v_k^{(p)})')$必定也会在作用$Ocp_j(G, \mathcal{V}, V_{i_j})$时被加入。

(2.4) $u \in \{v_1^{(p)}, ..., v_s^{(p)}\}$且$v \in \{u_1^{(q)}, ..., u_t^{(q)}\}$ (或者$u \in \{u_1^{(q)}, ..., u_t^{(q)}\}$且$v \in \{v_1^{(p)}, ..., v_s^{(p)}\}$)。令$u = v_l^{(p)}$，$v = u_k^{(q)}$，则$(f(u), f(v)) = ((v_l^{(p)})', (u_k^{(q)})')$。由于当两次操作的次序为$Ocp_j(G, \mathcal{V}, V_{i_j})$，$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$时，$u$是通过$Ocp_j(G, \mathcal{V}, V_{i_j})$加入的，$v$是通过$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$加入的，$(u, v)$被加入必定是因为$(v_l, u_k) \in$

$E(G_{j-1})$(更确切地说，$(v_l, u_k) \in E(G)$)，因此当互换两次操作的次序后，我们会首先在进行$Ocp_{j+1}(G, \mathcal{V}, V_{i_{j+1}})$时加入$(u_k^{(q)})'$，边$((u_k^{(q)})', v_l)$(也包括边$((u_k^{(q)})', v_1^{(1)})$, ...,$((u_k^{(q)})', v_1^{(p-1)})$)会同时被加入，因为$(u_k^{(q)})'$是从$u_k$拷贝而来的。接着，在进行$Ocp_j(G, \mathcal{V}, V_{i_j})$时，$(v_l^{(p)})'$会被加入，边$((v_l^{(p)})', (u_k^{(q)})')$也同时被加入，因为$(v_l^{(p)})'$是从$v_l$拷贝而来的，并且现在边$((u_k^{(q)})', v_l)$已经在图中存在了。

至此我们完成了对引理3.3的证明。

□

有了引理3.3之后，引理3.4的证明就直截了当了。

引理 3.4. *给定图G和它的一个子自映射划分$\mathcal{V} = \{V_1, V_2, ..., V_m\}$。设$\mathbf{O}$是任意一个作用于$G$的长度为$N$的轨道拷贝操作的序列。设$\alpha$和$\beta$为集合$\{1, 2, ..., N\}$上的任意两个置换，则$G_{\alpha(\mathbf{O})}$和$G_{\beta(\mathbf{O})}$是同构的。*

证明. 置换群论的一个经典结论是，任意一个置换一定能够表示为一系列对换的乘积[5, 6]。因此$\beta(\mathbf{O})$可以通过对$\alpha(\mathbf{O})$施加一系列操作对换得到，反之亦然。根据引理3.3，任何操作对换不会影响产生的结果图。因此，$\alpha(\mathbf{O})$和$\beta(\mathbf{O})$产生的结果图是相同的，即$G_{\alpha(\mathbf{O})} \cong G_{\beta(\mathbf{O})}$。 □

下面给出定理3.1，它表明一个作用于初始子自映射划分的任意轨道拷贝操作序列将产生结果图的一个子自映射划分。这是下一节将要介绍的匿名算法的基础。

定理 3.1. *给定图G和它的一个子自映射划分$\mathcal{V} = \{V_1, V_2, ..., V_m\}$。设$\mathbf{O}$是任意一个作用于$G$的长度为$N$的轨道拷贝操作序列，并设$\mathcal{V}^{(N)}$和$G^{(N)}$为相应的结果划分和结果图($\mathcal{V}^{(N)}$中的每个单元为初始单元及其拷贝的并集合)。则$\mathcal{V}^{(N)}$是$G^{(N)}$的一个子自映射划分。*

证明. 设$\mathbf{O} = Ocp_1(G, \mathcal{V}, V_{i_1}), ..., Ocp_N(G, \mathcal{V}, V_{i_N})$，$N \geq 1$，且$i_n \in \{1, 2, ..., m\} (1 \leq n \leq N)$。设在$i_1, i_2, ..., i_N$中，1出现$k_1$次，2出现$k_2$次，...，$m$出现$k_m$次，这里$k_j \geq 0$，对于$1 \leq j \leq m$。根据引理3.4，如果我们首先进行$k_1$次$Ocp(G, \mathcal{V}, V_1)$，接着进行$k_2$次$Ocp(G, \mathcal{V}, V_2)$，这样下去直到最后进行$k_m$次$Ocp(G, \mathcal{V}, V_m)$，则最后得到的结果图仍然为$G_N$。当我们进行$k_1$次$Ocp(G, \mathcal{V}, V_1)$后，根据引理3.2，得到的划分$\mathcal{V}^{(k_1)}$是$G_{k_1}$的子自映射划分。注意，现在引理3.2的假设在$G_{k_1}$和$\mathcal{V}^{(k_1)} = \{V_1^{(k_1)}, V_2^{(k_1)}, ..., V_m^{(k_1)}\} = \{V_1^{(k_1)}, V_2, ..., V_m\}$上成立。因此当我们进行$k_2$次$Ocp(G_{k_1}, \mathcal{V}^{(k_1)}, V_2)$后，根据引理3.2，得到的划分$\mathcal{V}^{(k_1+k_2)}$同样是$G_{k_1+k_2}$的子自映射划分。这一过程可以继续重复，直到我们进行k_m次$Ocp(G, \mathcal{V}, V_m)$后，得到的划分$\mathcal{V}^{(\sum_{j=1}^m k_j)} = \mathcal{V}^{(N)}$是$G_{\sum_{j=1}^m k_j} = G_N$的子自映射划分。证毕。 □

3.2.2 匿名算法

基于上一节定义的轨道拷贝操作，下面介绍将给定的图G修改为k-对称的图G'的匿名算法(见算法1)。其基本思想是对每个$V_i \in Orb(G)(|V_i| \leq k)$重复进行轨道拷贝操作，直到$V_i$及其拷贝的并集合的大小至少为$k$。

算法 1: k-对称匿名算法

Input: 图G及其自映射划分$Orb(G) = \{V_1, V_2, ..., V_m\}$，正整数$k$

Output: k-对称的图G'(对应于G和$Orb(G)$)

```
for 1 ≤ i ≤ m do
    if |V_i| ≥ k then
        Continue;
    end
    else
        Let V_i' = V_i;
        while |V_i'| < k do
            Ocp(G, Orb(G), V_i);
            V_i' = V_i' ∪ V_i;
        end
    end
end
```

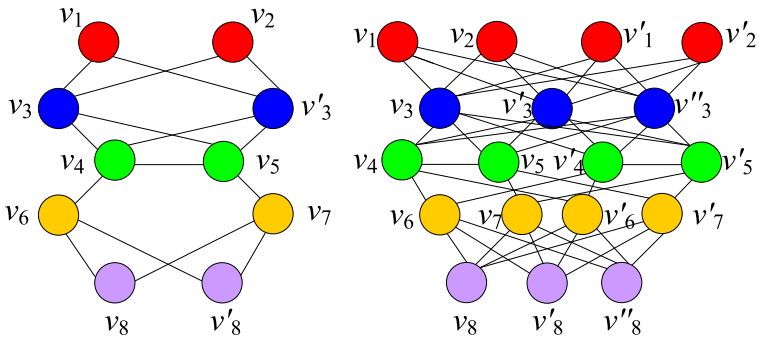
容易证明，算法1产生的结果图是k-对称的(见定理3.2)。

定理 3.2. *算法1的匿名过程产生的结果图G'是k-对称的。*

证明. 匿名过程事实上可以被视作一个轨道拷贝操作序列。设得到的划分为$\mathcal{V}'$，则根据定理3.1，$\mathcal{V}'$是G'的子自映射划分，因此$\mathcal{V}'$比$Orb(G')$细致。由于$\forall V \in \mathcal{V}'$，$|V| \geq k$，于是$\forall U \in Orb(G')$，$|U| \geq k$。因此$G'$是$k$-对称的。 □

下面通过一个例子来说明算法1的匿名过程(见例3.5)。

例 3.5 (匿名过程). 考虑图3.2(a)中的图G。设现在$k = 2$，则需要拷贝V_2和V_5才能使得产生的图是2-对称的。图3.4(a)展示了经过匿名过程后的图G'。匿名后得到$V(G')$的顶点划分为$\mathcal{V}' = \{V_1, V_2, V_3, V_4, V_5\}$，其中$V_2 = \{v_3, v_3'\}$，$V_5 = \{v_8, v_8'\}$，其他的单元和$Orb(G)$中是一样的。容易验证，$\mathcal{V}'$是$G'$的一个子自映射划分，且每个单元至少包含两个自映射等价的顶点。图3.4(b)展示了当$k = 3$时匿名化后得到的图G'。这里，由于$Orb(G)$的所有轨道都不满足3-对称的约束，因此它们都需要进行拷贝。



(a) 当$k=2$时产生的图G'　　(b) 当$k=3$时产生的图G'

图 3.4: 匿名过程示例

由于存在不同的自映射划分的计算方法，因此对于给定的图G，无法保证所有方法得到的自映射划分$Orb(G)$中的轨道的顺序都相同。但是根据3.4，易知匿名过程的结果与轨道拷贝的顺序无关，即无论$Orb(G)$中的轨道的顺序如何，得到的结果k-对称图G'总是相同的。

下面分析一下匿名算法的时间复杂度。注意到算法1的时间复杂度和新加入的点和边的数量呈正比例。设初始情况下$\mathcal{V}$中有N个单元包含少于k 个顶点，记为$V_{i_1},...,V_{i_N}$。令$k_1,...,k_N$分别为对$V_{i_k}(1\le k\le N)$进行轨道拷贝的次数，则新加入的顶点数量为$\sum_{j=1}^{N}k_j|V_{i_j}|\le(k-1)|V(G)|$，因为每个$k_j\le k-1$并且$\sum_{j=1}^{N}|V_{i_j}|\le|V(G)|$。新加入的边的数量则少于$\sum_{j=1}^{N}k_j|V_{i_j}|(k|V(G)|)\le k(k-1)|V(G)|^2$。通常情况下$k$远小于$|V(G)|$，因此可以被视作常量。所以，匿名过程在最差情况下的运行时间为$O(|V(G)|^2)$。

3.3　可用性

由于在匿名过程中加入了新的顶点和边，原网络的拓扑结构发生了改变，因此需要考虑匿名后网络的可用性问题，即是否能够在匿名后的网络中还原出原网络的大部分宏观统计性质。本节将对这一问题进行深入探讨。

3.3.1　B骨架

复杂网络领域的最新研究结果表明，网络商[6]能够保留原网络G的许多重要宏观统计性质，如最短路径的平均长度等[21]。因此保留住原网络的骨架就成为了从匿名后的网络中还原出原网络的重要统计性质的关键所在。注意到轨道拷贝操作事实上保留了原网络中轨道之间的链接模式，因此有理由认为原网络和

[6]给出网络G，设G的自映射划分为$Orb(G)=\{\Delta_1,\Delta_2,...,\Delta_s\}$。$G$的商$(quotient)$，(或称骨架)，定义为网络$Q$，其中$Q$的顶点集为$V(Q)=\{\delta_1,\delta_2,...,\delta_s\}$，边$(\delta_i,\delta_j)\in E(Q)$当且仅当$\exists u\in\Delta_i,v\in\Delta_j,(u,v)\in E(G)$(对任意的$i,j\in1,2,...,s$)。

匿名后的网络具有相似的骨架。本节将对此进行详细分析。

设二元组$(H,\mathcal{V}_H)$表示图H和它的一个子自映射划分$\mathcal{V}_H$，则对$(H,\mathcal{V}_H)$作用一组轨道拷贝操作后可以得到一个唯一的二元组$(G,\mathcal{V}_G)$，其中G为结果图，$\mathcal{V}_G$为相应于G的结果划分。称$(G,\mathcal{V}_G)$是$(H,\mathcal{V}_H)$的泛化(*generalization*)，同时称$(H,\mathcal{V}_H)$是$(G,\mathcal{V}_G)$的约简(*reduction*)[7]。用$(H,\mathcal{V}_H)\leq(G,\mathcal{V}_G)$表示两者之间的约简-泛化关系。特别地，$G$同时是其自身的泛化和约简。显然，算法1匿名过程产生的结果图G'是原图G的泛化(相应于$Orb(G)$)。

下面分析上述约简-泛化关系的重要性质。首先，对于给定的子自映射划分$\mathcal{V}_G$，可能存在G的多个约简。用$\mathcal{R}(G,\mathcal{V}_G)$表示$(G,\mathcal{V}_G)$的所有约简组成的集合，即$\mathcal{R}(G,\mathcal{V}_G)=\{(H,\mathcal{V}_H)|(H,\mathcal{V}_H)\leq(G,\mathcal{V}_G)\}$。由于$(G,\mathcal{V}_G)\in\mathcal{R}(G,\mathcal{V}_G)$，所以$\mathcal{R}(G,\mathcal{V}_G)$一定是非空的。容易验证，$\leq$关系在$\mathcal{R}(G,\mathcal{V}_G)$上是自反的，反对称的，和传递的。因此$\leq$是定义在$\mathcal{R}(G,\mathcal{V}_G)$ 上的偏序关系。

在给出关于$\mathcal{R}(G,\mathcal{V}_G)$的重要定理(定理3.3)之前，先介绍几条重要的引理。首先给出引理3.5。

引理 3.5. *设A为有限集，$|A|=n$，R为A上定义的等价关系。令$p|n$，$q|n$，$p\neq q$且$n=p\cdot s=q\cdot t$。现假设下面两个条件成立：*
(1) A中的元素可以被划分为s个互不相交的子集A_i，满足$\forall a\in A_i,b\in A_i$，$(a,b)\in R$，并且对于每个$1\leq i\leq s$成立$|A_i|=p$；
(2) A中的元素可以被划分为t个互不相交的子集B_j，满足$\forall a\in B_j,b\in B_j$，$(a,b)\in R$，并且对于每个$1\leq j\leq t$成立$|B_j|=q$；
则A中的元素一定可以被划分为$gcd(s,t)$个互不相交的子集C_k，满足$\forall a\in C_k,b\in C_k$，$(a,b)\in R$，并且对于每个$1\leq k\leq gcd(s,t)$成立$|C_k|=lcm(p,q)$。这里$gcd(x,y)$和$lcm(x,y)$分别表示$x$和$y$的最大公约数和最小公倍数。

证明. 设A/R为R在A上的商集。令$E\in A/R$，则有$|E|=m\cdot lcm(p,q)$，这里$m\geq 1$为正整数。换句话说，A/R中的等价类的大小一定是$lcm(p,q)$的一个倍数。下面用反证法说明这一点。

若结论不成立，则存在$F\in A/R$，$|F|=M$且$lcm(p,q)\nmid M$。因此有$p\nmid M$或者$q\nmid M$。不失一般性，设$p\nmid M$，于是有$M=k\cdot p+r$，这里$k\geq 1$是某个非负整数且有$1\leq r\leq p-1$。取$a_1\in F$。根据条件(1)，$\{A_i\}$是A的划分，因此存在某个i_1，$1\leq i_1\leq s$，满足$a_1\in A_{i_1}$。因为$(a_1,y)\in R$，所以每个A_{i_1}中的y必然也在F中，于是我们有$A_{i_1}\subset F$。令$F_1=F\setminus A_{i_1}$。取$a_2\in F_1$，同理存在某个i_2，$i_2\neq i_1$，$a_2\in A_{i_2}$，于是有$A_{i_2}\subset F_1$。令$F_2=F_1\setminus A_{i_2}=F\setminus(A_{i_1}\cup A_{i_2})$。类似地可以

[7]为了简便起见，下文中，在上下文非常清楚的情况下，也称H是G的约简，或者G是H的泛化，省略了相应的划分

取$a_3 \in F_2$并重复这一过程。由于$M = k \cdot p + r$，因此上述过程可以重复k次直至我们有$|F_k| = r$，这里$F_k = F \setminus (\cup_{j=1}^{k} A_{i_j})$。但是若我们现在取$a_{i_{k+1}} \in F_k$，那么根据上面的论证过程，存在$i_{k+1}$，$i_{k+1} \neq i_j$(对于所有$1 \leq j \leq k$)，满足$a_{i_{k+1}} \in A_{i_{k+1}}$。于是有$A_{i_{k+1}} \subset F_k$，因此$|F_k| \geq |A_{i_{k+1}}| = p > r$，这与$|F_k| = r$相矛盾。$q \nmid M$的情况可以完全类似地用反证法进行证明。因此$M$一定是$lcm(p,q)$的一个倍数，即$lcm(p,q)|M$。由于$F$的选择是任意的，因此我们可以得出结论：每个等价类$E \in A/R$的大小一定是$lcm(p,q)$的一个倍数。

进一步，由于A/R也是A的一个划分，于是划分$\{C_k\}$可以通过将每个$E \in A/R$ 划分为互不相交且大小为$lcm(p,q)$的子集来得到。由于$n = p \cdot s = q \cdot t$，$n$是$p$和$q$的一个公倍数，因此有$lcm(p,q)|n$。令$n = lcm(p,q) \cdot l$，则$l$是划分$\{C_k\}$中等价类的数目。下面证明$l = gcd(s,t)$。

设$lcm(p,q) = l_1 \cdot p = l_2 \cdot q$，这里$gcd(l_1, l_2) = 1$。因为$n = p \cdot s = lcm(p,q) \cdot l$，我们有$p \cdot s = l_1 \cdot p \cdot l$，即$s = l_1 \cdot l$。同理，因为$n = q \cdot t = lcm(p,q) \cdot l$，我们有$q \cdot t = l_2 \cdot q \cdot l$，即$t = l_2 \cdot l$。因此，$l|s$且$l|t$，于是有$l|gcd(s,t)$。令$l_3$为$s$和$t$的任意公约数，即$l_3|s$且$l_3|t$。设$s = k_s \cdot l_3$，$t = k_t \cdot l_3$。因为$p \cdot s = lcm(p,q) \cdot l$，我们有$p \cdot l_3 \cdot k_s = p \cdot l_1 \cdot l$，即$l_3 \cdot k_s = l_1 \cdot l$；又因为$q \cdot t = lcm(p,q) \cdot l$，我们有$q \cdot l_3 \cdot k_t = q \cdot l_2 \cdot l$，即$l_3 \cdot k_t = l_2 \cdot l$。因此$l_3 \cdot k_s \cdot l_2 = l_3 \cdot k_t \cdot l_1$，即$k_s \cdot l_2 = k_t \cdot l_1$。由于$gcd(l_1, l_2) = 1$，因此必有$l_1|k_s$，$l_2|k_t$。又因为$l_3 \cdot k_s = l_1 \cdot l$且$l_3 \cdot k_t = l_2 \cdot l$，所以必有$l_3|l$。由于$l_3$ 的选择是任意的，若令$l_3 = gcd(s,t)$，则有$gcd(s,t)|l$。因此，$l = gcd(s,t)$，证毕。 □

在进一步介绍下面两条引理之前，我们需要先引入一些必要的记号。在下面的表述中，我们用V_G^i表示$\mathcal{V}_G$中的第i个单元。若$(H, \mathcal{V}_H) \leq (G, \mathcal{V}_G)$，则存在向量$\mathbf{k} = (k_1, k_2, ..., k_m)$，满足$k_i \geq 1$且$(k_i - 1)$是$V_H^i$被拷贝的次数。我们称$\mathbf{k}$为轨道拷贝频次向量 (*orbit copying frequency vector*)，或简称为*ocf*向量(*ocf-vector*)。由于每次轨道拷贝后产生的图是唯一的，并且重新排列轨道拷贝操作的次序不会影响最后的结果，因此当G，$\mathcal{V}_G$，H，$\mathcal{V}_H$给定时，$\mathbf{k}$是唯一确定的。我们用记号$\mathbf{k}^{H,\mathcal{V}_H,G,\mathcal{V}_G}$来强调这种唯一性。当$\mathcal{V}_G$和$\mathcal{V}_H$在上下文中不会引起歧义时，为了简便起见，我们使用记号$\mathbf{k}^{H,G}$。进一步，我们用记号$(H, \mathcal{V}_H, \mathbf{k}^{H,G}) \rightarrow (G, \mathcal{V}_G)$来表示$(G, \mathcal{V}_G)$可以通过对$(H, \mathcal{V}_H)$的单元$V_H^i$拷贝($\mathbf{k}_i^{H,G}$-1)次来得到，这里$\mathbf{k}_i^{H,G}$是向量$\mathbf{k}^{H,G}$的第$i$个分量。

下面的引理(引理3.6)介绍了偏序集$\mathcal{R}(G, \mathcal{V}_G)$的一个重要性质。

引理 3.6. *设$(H_1, \mathcal{V}_{H_1})$和$(H_2, \mathcal{V}_{H_2})$是$\mathcal{R}(G, \mathcal{V}_G)$中的任意两个元素，则存在元素$(H, \mathcal{V}_H) \in \mathcal{R}(G, \mathcal{V}_G)$，满足$(H, \mathcal{V}_H) \leq (H_1, \mathcal{V}_{H_1})$，$(H, \mathcal{V}_H) \leq (H_2, \mathcal{V}_{H_2})$，并且对于任意其它满足$(H', \mathcal{V}_{H'}) \leq (H_1, \mathcal{V}_{H_1})$和$(H', \mathcal{V}_{H'}) \leq (H_2, \mathcal{V}_{H_2})$的元素$(H', \mathcal{V}_{H'}) \in$*

$\mathcal{R}(G,\mathcal{V}_G)$，有$(H',\mathcal{V}_{H'}) \le (H,\mathcal{V}_H)$成立。记$H = H_1 \wedge H_2$。换句话说，$\mathcal{R}(G,\mathcal{V}_G)$中的任意两个元素在$\mathcal{R}(G,\mathcal{V}_G)$中具有最大的下界。

证明. 设$\mathbf{k}^{H_1,G}$和$\mathbf{k}^{H_2,G}$分别为满足$(H_1,\mathcal{V}_{H_1},\mathbf{k}^{H_1,G}) \to (G,\mathcal{V}_G)$和$(H_2,\mathcal{V}_{H_2},\mathbf{k}^{H_2,G}) \to (G,\mathcal{V}_G)$的ocf向量。定义每个$V_i \in \mathcal{V}$上的二元关系$R_i$为：$\forall u,v \in V_i$, $(u,v) \in R_i$当且仅当$u = v$或者u和v互为拷贝。易知R_i为等价关系。由于V_G^i能够通过对$V_{H_1}^i$拷贝($k_i^{H_1,G}$-1)次得到，V_G^i中的顶点于是可以被划分为$|V_{H_1}^i|$个互不相交的子集，每个子集包含$k_i^{H_1,G}$个在R_i下相互等价的顶点。另一方面，由于V_G^i同时可以通过对$V_{H_2}^i$拷贝($k_i^{H_2,G}$-1)次得到，于是V_G^i中的顶点又可以被划分为$|V_{H_2}^i|$个互不相交的子集，每个子集包含$k_i^{H_2,G}$个在R_i下相互等价的顶点。因此，根据引理3.6，V_G^i中的顶点能够被划分为$gcd(|V_{H_1}^i|,|V_{H_2}^i|)$个互不相交的子集，每个子集包含$lcm(k_i^{H_1,G},k_i^{H_2,G})$个在$R_i$下相互等价的顶点。这说明存在$V_H^i \subseteq V_G^i$，$|V_H^i| = gcd(|V_{H_1}^i|,|V_{H_2}^i|)$，且$V_G^i$可以通过对拷贝$V_H^i$拷贝($|V_G^i|/|V_H^i| - 1 = lcm(k_i^{H_1,G},k_i^{H_2,G}) - 1$)次得到。令$H$为由所有$\cup V_H^i$中的顶点导出的子图，$\mathcal{V}_H$为由$V_H^i$为单元构成的$V(H)$的划分。可以证明$\mathcal{V}_H$为$H$的子自映射划分[8]。

下面我们定义向量$\mathbf{k}^{H,H_1}$和$\mathbf{k}^{H,H_2}$，分别满足$k_i^{H,H_1} = |V_{H_1}^i|/gcd(|V_{H_1}^i|,|V_{H_2}^i|)$，$k_i^{H,H_2} = |V_{H_2}^i|/gcd(|V_{H_1}^i|,|V_{H_2}^i|)$，则有$(H,\mathcal{V}_H,\mathbf{k}^{H,H_1}) \to (H_1,\mathcal{V}_{H_1})$，以及$(H,\mathcal{V}_H,\mathbf{k}^{H,H_2}) \to (H_2,\mathcal{V}_{H_2})$。因此，$(H,\mathcal{V}_H) \le (H_1,\mathcal{V}_{H_1})$，$(H,\mathcal{V}_H) \le (H_2,\mathcal{V}_{H_2})$。进一步，定义向量$\mathbf{k}^{H,G}$为$k_i^{H,G} = k_i^{H,H_1} \cdot k_i^{H_1,G} = k_i^{H,H_2} \cdot k_i^{H_2,G} = |V_G^i|/gcd(|V_{H_1}^i|,|V_{H_2}^i|)$，则有$(H,\mathcal{V}_H,\mathbf{k}^{H,G}) \to (G,\mathcal{V}_G)$。因此$(H,\mathcal{V}_H) \in \mathcal{R}(G,\mathcal{V}_G)$。现在设$(H',\mathcal{V}_H) \in \mathcal{R}(G,\mathcal{V}_G)$为任意满足$(H',\mathcal{V}_{H'}) \le (H_1,\mathcal{V}_{H_1})$和$(H',\mathcal{V}_{H'}) \le (H_2,\mathcal{V}_{H_2})$的元素，则类似地我们有$(H',\mathcal{V}_{H'},\mathbf{k}^{H',H_1}) \to (H_1,\mathcal{V}_{H_1})$，$(H',\mathcal{V}_{H'},\mathbf{k}^{H',H_2}) \to (H_2,\mathcal{V}_{H_2})$。由于$H' \in \mathcal{R}(G,\mathcal{V}_G)$，我们同样有$(H',\mathcal{V}_{H'},\mathbf{k}^{H',G}) \to (G,\mathcal{V}_G)$。因此，$k_i^{H',G} = k_i^{H',H_1} \cdot k_i^{H_1,G} = k_i^{H',H_2} \cdot k_i^{H_2,G} = |V_G^i|/|V_{H'}^i|$。因为$k_i^{H',H_1} = |V_{H_1}^i|/|V_{H'}^i|$，$k_i^{H',H_2} = |V_{H_2}^i|/|V_{H'}^i|$，我们有$(|V_{H'}^i|) \mid gcd(|V_{H_1}^i|,|V_{H_2}^i|)$。又因为$|V_G^i| = k_i^{H,G} \cdot gcd(|V_{H_1}^i|,|V_{H_2}^i|) = k_i^{H',G} \cdot |V_{H'}^i|$，所以$k_i^{H,G} \mid k_i^{H',G}$。若我们定义向量$\mathbf{k}^{H',H}$为$k_i^{H',H} = k_i^{H',G}/k_i^{H,G}$，则有$(H',\mathcal{V}_{H'},\mathbf{k}^{H',H}) \to (H,\mathcal{V}_H)$。因此，$(H',\mathcal{V}_{H'}) \le (H,\mathcal{V}_H)$，证毕。 □

[8]对于任意V_H^i中的任意两个顶点u和v，注意到同时有$u \in V_G^i$，$v \in V_G^i$，因此存在G的一个自映射g，满足$u^g = v$，且$\mathcal{V}_G^g = \mathcal{V}_G$。现固定$i$，$u$和$v$，并设$g$为满足条件的自映射。对于任意$V_H^j$中的顶点$x$，若$x^g \notin V_H^j$，则$g$必将$x$映射到$V_H^j$的某个拷贝中的顶点$y$，因此$g$将$V_H^j$的导出子图中$x$所在的连通分量映射到该拷贝中对应的连通分量。设$g'$是$V(G)$上的这样一个置换，它在$x$所在的连通分量和对应的拷贝中的连通分量之间的映射作用与拷贝操作的对应关系相同，在这两个连通分量之外不置换任何一个顶点。为方便起见，我们记f为g'的逆映射。容易知道$g'(f)$也是G的自映射。现在令$h = gf$，则h也是G的自映射并且有$x^h = (x^g)^f = y^f \in V_H^j$。事实上$h$将$x$所在的连通分量置换到了其自身。由于$i$，$u$，$v$，$j$和$x$的选择是任意的，这说明对于$\mathcal{V}_H$的任意一个单元中任意两个顶点$u$和$v$，我们都能找到$G$的一个自映射$h$，满足$u^h = v$且$\mathcal{V}_H^h = \mathcal{V}_H$。由于事实上$h$只是在$V(H)$上进行置换，因此若$h$是$G$的自映射，那么$h$必然也是$H$的自映射。因此$\mathcal{V}_H$为$H$的子自映射划分。

下一条引理(引理3.7)陈述了偏序集$(\mathcal{R}(G,\mathcal{V}_G);\leq)$的一条与引理3.6中陈述的性质相对偶的性质。

引理 3.7. 设$(H_1,\mathcal{V}_{H_1})$和$(H_2,\mathcal{V}_{H_2})$为$\mathcal{R}(G,\mathcal{V}_G)$中的任意两个元素，则存在元素$(H,\mathcal{V}_H)\in\mathcal{R}(G,\mathcal{V}_G)$，满足$(H_1,\mathcal{V}_{H_1})\leq(H,\mathcal{V}_H)$，$(H_2,\mathcal{V}_{H_2})\leq(H,\mathcal{V}_H)$，并且对于任意其它满足$(H_1,\mathcal{V}_{H_1})\leq(H',\mathcal{V}_{H'})$和$(H_2,\mathcal{V}_{H_2})\leq(H',\mathcal{V}_{H'})$的元素$(H',\mathcal{V}_{H'})\in\mathcal{R}(G,\mathcal{V}_G)$，有$(H,\mathcal{V}_H)\leq(H',\mathcal{V}_{H'})$成立。记$H=H_1\vee H_2$。换句话说，$\mathcal{R}(G,\mathcal{V}_G)$中的任意两个元素在$\mathcal{R}(G,\mathcal{V}_G)$中具有最小的上界。

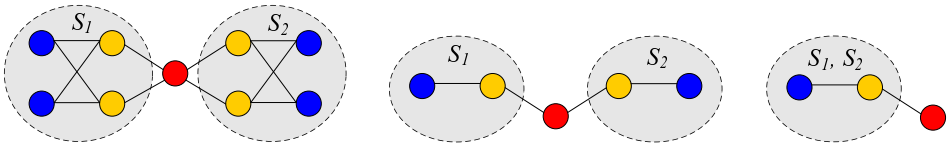
证明. 设$\mathbf{k}^{H_1,G}$和$\mathbf{k}^{H_2,G}$为相应的ocf向量。定义向量$\mathbf{k}^{H,G}$满足$k_i^{H,G}=gcd(k_i^{H_1,G},k_i^{H_2,G})$，并且定义向量$\mathbf{k}^{H_1,H}$和$\mathbf{k}^{H_2,H}$分别满足$k_i^{H_1,H}=k_i^{H_1,G}/k_i^{H,G}$，$k_i^{H_2,H}=k_i^{H_2,G}/k_i^{H,G}$。令基于$\mathbf{k}^{H_1,H}$对$\mathcal{V}_{H_1}$进行轨道拷贝操作后的图和子自映射划分分别为$G_1$和$\mathcal{V}_{G_1}$，基于$\mathbf{k}^{H_2,H}$对$\mathcal{V}_{H_2}$进行轨道拷贝操作后的图和子映射划分分别为$G_2$和$\mathcal{V}_{G_2}$。由于$(H_1,\mathcal{V}_{H_1},\mathbf{k}^{H_1,G})\to(G,\mathcal{V}_G)$，因此有$(H_1,\mathcal{V}_{H_1},\mathbf{k}^{H_1,H})\to(G_1,\mathcal{V}_{G_1})$，$(G_1,\mathcal{V}_{G_1},\mathbf{k}^{H,G})\to(G,\mathcal{V}_G)$。类似地，由于$(H_2,\mathcal{V}_{H_2},\mathbf{k}^{H_2,G})\to(G,\mathcal{V}_G)$，因此有$(H_2,\mathcal{V}_{H_2},\mathbf{k}^{H_2,H})\to(G_2,\mathcal{V}_{G_2})$，$(G_2,\mathcal{V}_{G_2},\mathbf{k}^{H,G})\to(G,\mathcal{V}_G)$。所以，一定有$G_1=G_2$，以及$\mathcal{V}_{G_1}=\mathcal{V}_{G_2}$。

我们用H表示这一公共的图，并用$\mathcal{V}_H$表示相应的子自映射划分。显然我们有$(H_1,\mathcal{V}_{H_1})\leq(H,\mathcal{V}_H)$，$(H_2,\mathcal{V}_{H_2})\leq(H,\mathcal{V}_H)$，因为$(H_1,\mathcal{V}_{H_1},\mathbf{k}^{H_1,H})\to(H,\mathcal{V}_H)$，$(H_2,\mathcal{V}_{H_2},\mathbf{k}^{H_2,H})\to(H,\mathcal{V}_H)$。又因为$(H,\mathcal{V}_H,\mathbf{k}^{H,G})\to(G,\mathcal{V}_G)$，所以有$H\in\mathcal{R}(G,\mathcal{V}_G)$。现在设$(H',\mathcal{V}_{H'})\in\mathcal{R}(G,\mathcal{V}_G)$为任意满足$(H_1,\mathcal{V}_{H_1})\leq(H',\mathcal{V}_{H'})$和$(H_2,\mathcal{V}_{H_2})\leq(H',\mathcal{V}_{H'})$的元素，则类似地我们有$(H_1,\mathcal{V}_{H_1},\mathbf{k}^{H_1,H'})\to(H',\mathcal{V}_{H'})$，$(H',\mathcal{V}_{H'},\mathbf{k}^{H',G})\to(G,\mathcal{V}_G)$以及$(H_2,\mathcal{V}_{H_2},\mathbf{k}^{H_2,H'})\to(H',\mathcal{V}_{H'})$，$(H',\mathcal{V}_{H'},\mathbf{k}^{H',G})\to(G,\mathcal{V}_G)$。因此，必有$k_i^{H',G}|k_i^{H_1,G}$，$k_i^{H',G}|k_i^{H_2,G}$。于是有$k_i^{H',G}|gcd(k_i^{H_1,G},k_i^{H_2,G})$，即$k_i^{H',G}|k_i^{H,G}$。因为$k_i^{H_1,G}=k_i^{H_1,H}\cdot k_i^{H,G}=k_i^{H_1,H'}\cdot k_i^{H',G}$，$k_i^{H_2,G}=k_i^{H_2,H}\cdot k_i^{H,G}=k_i^{H_2,H'}\cdot k_i^{H',G}$，我们有$k_i^{H_1,H'}/k_i^{H_1,H}=k_i^{H_2,H'}/k_i^{H_2,H}=k_i^{H,G}/k_i^{H',G}$。因此，若我们定义向量$\mathbf{k}^{H,H'}$为$k_i^{H,H'}=k_i^{H,G}/k_i^{H',G}$，则有$(H,\mathcal{V}_H,\mathbf{k}^{H,H'})\to(H',\mathcal{V}_{H'})$，所以$(H,\mathcal{V}_H)\leq(H',\mathcal{V}_{H'})$，证毕。□

在上述引理3.6和3.7的基础上，可以很自然地得到下面的重要定理(定理3.3)。$(\mathcal{R}(G,\mathcal{V}_G);\leq)$的有界性可以由$(\mathcal{R}(G,\mathcal{V}_G);\leq)$是有限集这一事实直接得到。

定理 3.3. 偏序集$(\mathcal{R}(G,\mathcal{V}_G);\leq)$是一个有界格*(bounded latice)*。

在定理3.3的基础上，我们定义图G(相应于$\mathcal{V}_G$)的B骨架为$(\mathcal{R}(G,\mathcal{V}_G);\leq)$的极小元(见定义3.4)。



(a) 图G (b) 图G的B骨架 (c) 图G的商

图 3.5: 图的B骨架示例

定义 3.4 (B骨架). 给定图G和它的一个子自映射划分$\mathcal{V}_G$，称有界格$(\mathcal{R}(G,\mathcal{V}_G);\leq)$的极小元为$(G,\mathcal{V}_G)$的B骨架。

由于$(\mathcal{R}(G,\mathcal{V}_G);\leq)$是有界格，所以它具有唯一的极小元。因此$(G,\mathcal{V}_G)$的B骨架是唯一确定的。我们用记号$B_{G,\mathcal{V}_G}$来表示这一唯一的B骨架。下面给出B骨架的一个例子。

例 3.6 (B骨架). 图3.5给出了一个图G和它的B骨架$B_{G,\mathcal{V}_G}$。图G中用同一颜色标记的顶点处于相应的子自映射划分$\mathcal{V}_G$的同一个单元中。

这里我们对B骨架再进行一点说明。从B骨架的定义不难看出，$(G,\mathcal{V}_G)$的B骨架$B_{G,\mathcal{V}_G}$事实上是能够通过轨道拷贝操作得到$(G,\mathcal{V}_G)$的最小的图。因此，如果我们定义一种轨道拷贝操作的逆操作的话，$B_{G,\mathcal{V}_G}$即是对$(G,\mathcal{V}_G)$不断作用这一逆操作的最终结果。为了描述方便，我们称这一逆操作为约简操作(*reduction operation*)。约简操作只对同一单元内部的顶点进行约简，这与得到网络商的操作(合并每个轨道为单个顶点)相似。但是，与得到网络商的操作的不同点在于，约简操作不会约简掉自映射等价但是跨越多个单元的两个子结构。例如，在图3.5中，自映射等价的两个子结构S_1和S_2在G的B骨架(图3.5(b))中会被保留，而在G的商(图3.5(c))中则会被约简。约简操作的这一约束是有意义的，因为相互同构但是不同的子结构(它们是自映射等价的)通常被视为网络中的不同模块(module)。在图3.5(a)中，很显然S_1和S_2是G的两个不同的模块，因此在约简后的图中保留这一模块信息是更为合理的。从这个意义上来说，B骨架是比商更为细致的一种网络骨架。因此既然网络商能够保留住原网络的许多重要的统计性质，那么可以预期B骨架能够更好地保留住原网络的这些统计性质。

B骨架的另一个重要性质是:轨道拷贝操作能够保留住B骨架。换言之，经过任意一系列轨道拷贝操作后得到的网络，能够约简得到和原网络相同的B骨架(见定理3.4)。

定理 3.4. 设$(H,\mathcal{V}_H)\leq(G,\mathcal{V}_G)$，即$(H,\mathcal{V}_H)$是$(G,\mathcal{V}_G)$的约简，则有：$B_{G,\mathcal{V}_G}=B_{H,\mathcal{V}_H}$。

证明. 因为$(H,\mathcal{V}_H)\leq(G,\mathcal{V}_G)$，所以$(H,\mathcal{V}_H)\in\mathcal{R}(G,\mathcal{V}_G)$，从而$B_{G,\mathcal{V}_G}\leq(H,\mathcal{V}_H)$，于是$B_{H,\mathcal{V}_H}\leq B_{G,\mathcal{V}_G}$，因此有$(B_{H,\mathcal{V}_H},\mathcal{V}^{B_{H,\mathcal{V}_H}},\mathbf{k}^{B_{H,\mathcal{V}_H},B_{G,\mathcal{V}_G}})\rightarrow(B_{G,\mathcal{V}_G},\mathcal{V}^{B_{G,\mathcal{V}_G}})$。定义向量$\mathbf{k}^{B_{H,\mathcal{V}_H},G}$为$k_i^{B_{H,\mathcal{V}_H},G}=k_i^{B_{H,\mathcal{V}_H},B_{G,\mathcal{V}_G}}\cdot k_i^{B_{G,\mathcal{V}_G},G}$，则有$(B_{H,\mathcal{V}_H},\mathcal{V}^{B_{H,\mathcal{V}_H}},\mathbf{k}^{B_{H,\mathcal{V}_H},G})\rightarrow(G,\mathcal{V}_G)$。因此$B_{H,\mathcal{V}_H}\in\mathcal{R}(G,\mathcal{V}_G)$，于是我们有$B_{G,\mathcal{V}_G}\leq B_{H,\mathcal{V}_H}$。所以$B_{G,\mathcal{V}_G}=B_{H,\mathcal{V}_H}$。□

3.3.2　基于B骨架的采样方法

根据定理3.4，算法1所示的匿名过程所产生的图G'和原图G具有相同的B骨架(相应于给定的G的子自映射划分$\mathcal{V}_G$)。既然B骨架能够保留住原网络的基本结构特性，有理由认为具有相同B骨架和相似规模的网络也应该具有相似的统计性质。因此如何从匿名后的k-对称网络中得到它的B骨架就成为了得到原网络的统计性质的关键所在。本节将提出两种基于B骨架的采样方法，它们能够从匿名后的网络中抽取出原网络的一个近似版本。我们首先给出基于B骨架进行采样的基本框架，然后分别介绍基于精确计算的B骨架和近似计算的B骨架的采样方法。下一节的实验部分我们将看到采样得到的这些原网络的近似版本的宏观统计性质和原网络是非常接近的，虽然它们在微观局部上已经完全不同于原网络。

3.3.2.1　总体框架

设$(G',\mathcal{V}')$为$(G,\mathcal{V}_G)$匿名后的结果。设$\mathcal{P}$是关于G的一组结构知识，例如G的顶点数或边数。我们可以视任意结构知识$P\in\mathcal{P}$为关于G的断言(*assertion*)。默认地，我们设$P(G)=true$(即$P(G)$为真)。于是，样本空间(*sample space*)可以定义为$SS(G',\mathcal{V}',\mathcal{P})=\{(H,\mathcal{V}_H)|B_{G',\mathcal{V}'}=B_{H,\mathcal{V}_H}\wedge(P(H)=true,\forall P\in\mathcal{P})\}$。一次基于B骨架的采样即是从样本空间中完全随机地(uniformly)抽取一个图H。

显然，采用不同的$\mathcal{P}$将得到不同的样本空间。一方面，由于原始网络G也在样本空间中，因此我们希望样本空间越大越好；但是另一方面，越大的样本空间通常意味着对采样得到的图在结构上更少的约束，因此有可能导致样本图的可用性下降。因此，在上述框架下，网络的发布者需要选择一组合理的$\mathcal{P}$来达到样本空间的大小和样本图的可用性之间的一个较好的平衡(trade-off)。然而事实上，仅仅采用原图的顶点数$|V(G)|$来作为$\mathcal{P}$就已经给出了这样一个平衡。这里我们进行一个简单的分析。设相应于$B_{G',\mathcal{V}'}$的子自映射划分为$\mathcal{B}=\{B_1,B_2,...,B_m\}$。样本空间的大小于是等于方程$\sum_{i=1}^m k_i|B_i|=|V(G)|$的所有可行解(feasible solution)$(k_1,...,k_m)$的数目(每个k_i为正整数)。因此，在一般情况下，样本空间的大小和$|V(G)|$之间呈指数关系。所以即便是中等规模的网络，其样本空间也会非常庞大。但是，在下一节的实验部分我们将看到，在绝

大多数情况下，样本图的可用性是非常好的。因此，在下面讨论基于B骨架的采样方法的具体实现时，我们仅考虑发布原网络的顶点数的情况。

3.3.2.2 基于精确计算的B骨架的采样方法

这一节介绍基于精确计算的B骨架的采样方法。我们首先给出从网络中得到B骨架的算法(见算法2)。

算法 2: 从网络中得到B骨架

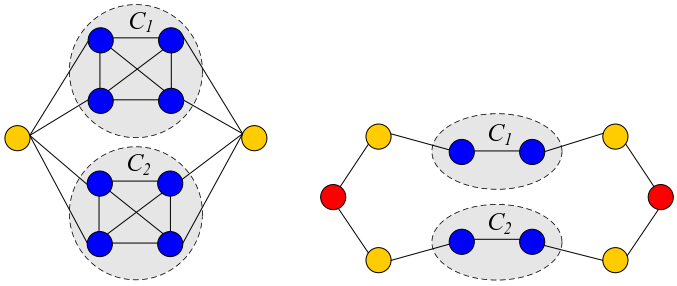
Input: $G, \mathcal{V}$

Output: $B_{G,\mathcal{V}}$

```
1  foreach V ∈ 𝒱 do
2      foreach v ∈ V do
3          计算L(v);
4          foreach u ∈ L(v) do
5              ℒ(V) = ℒ(V) ∪ (v, u);
6          end
7      end
8      foreach C ∈ 𝒞(G_V) do
9          if ∃C′ ∈ 𝒞(G_V), G′_C ≅_{ℒ(V)} G_C then
10             从G中删除C′;
11         end
12     end
13 end
14 return 结果图(即B_{G,𝒱});
```

算法2的基本想法是不断地删除网络中可以通过轨道拷贝操作得到的子图(行8至12)。然而在此之前，我们首先需要知道网络的哪些子图是可以通过轨道拷贝操作得到的(行2至7)。给定图G和它的一个子自映射划分$\mathcal{V}$，由每个单元$V \in \mathcal{V}$导出的子图(记为$G[V]$)是由一组连通分量构成的。因此根据轨道拷贝操作的定义，如果V可以被约简的话，那么其中一些连通分量可以被视作另一些分量的拷贝。令$\mathcal{C}(G[V])$为$G[V]$的所有连通分量构成的集合，若$C_1 \in \mathcal{C}(G[V])$是$C_2 \in \mathcal{C}(G[V])$的拷贝，则$C_1$一定与$C_2$同构。例如，在图3.6的两张图中，连通分量$C_1$与连通分量$C_2$是同构的($C_1$和$C_2$都属于由标记为蓝色的单元导出的子图)。

然而，在一些情况下，连通分量之间的同构关系并不足以刻画两者之间互为拷贝这一关系。例如，在图3.6(b)中，连通分量C_1 (C_2)和C_2 (C_1)之间



(a) G_1　　(b) G_2

图 3.6: 图及其B骨架的实例

并不构成互为拷贝关系，尽管它们是同构的，这是因为C_1中不存在和C_2中的任何一个顶点具有相同邻居的顶点。但是，在图3.6(a)中，对于每个顶点$u \in C_1$，可以找到对应的顶点$v \in C_2$，u和v在蓝色标记的单元之外具有相同的邻居。正式地，设$\mathcal{L}(V)$为V中所有顶点对(v,u)，v和u在V外具有相同的邻居，即$N(v) \cap (V(G) - V) = N(u) \cap (V(G) - V)$，并且$v \in C_i$，$u \in C_j$，$C_i \in \mathcal{C}(G[V])$，$C_j \in \mathcal{C}(G[V])$，$i \neq j$。(在算法2中，对于$V$中的每个顶点$v$，维护一个表$L(v)$。$L(v)$包含了所有满足这一条件的顶点$u$。见行2至7。)则仅当我们能够找到一个$C_i$到$C_j$的同构映射$\theta$，满足$\forall v \in C_i, (v, \theta(v)) \in \mathcal{L}(V)$时(这一关系记为$C_i \cong_{\mathcal{L}(V)} C_j$)，$C_i$和$C_j$之间才互为拷贝关系，因此它们中的一个可以从网络中删去(行9到11)。当我们从网络中删去一个连通分量时，除了删掉属于该连通分量的顶点外，同时删除这些顶点所关联的边。

上述分析表明，被删去的每个连通分量都可以通过其他某个连通分量拷贝而来。因为算法2删除了网络中所有可能存在的拷贝，因此可以确定算法2最后得到的图即为$B_{G,\mathcal{V}}$。在算法2得到的B骨架的基础上，我们提出一种采样方法来得到原网络的一个近似版本(见算法3)。

算法3的输入为原网络G的k-对称版本$(G', \mathcal{V}')$，以及G的顶点数。这里还可以指定$p[i]$作为输入，它是$\mathcal{V}'$中单元V_i'的*采样概率*(*sampling probability*)。一般而言，$p[i]$可以为任意概率分布。但是真实社会网络通常具有右倾斜(right-skewed)的度分布，意味着度为k的顶点数量和k是反相关的(一般有$p(k) \sim k^{-\lambda}$, $\lambda \in (2,3)$[16])。因此，在原图的划分$\mathcal{V}$中，单元中所包含的顶点数量也应该和单元中顶点的度是反相关的。所以，通常设$p[i] = d_i^{-1} / \sum_{j=1}^{|\mathcal{V}'|} d_j^{-1}$(这里$d_i$是$V_i'$中顶点的度)。

算法3的基本思想是将$N = n - |V(B_{G',\mathcal{V}'})|$个顶点根据概率$p[i]$分配到$\mathcal{B}$的不同单元中去。在实际实现时，我们首先计算$CPN[i]$(初始化为0) 以记录根据$p[i]$，$\mathcal{B}$的单元$B_i$需要被拷贝的次数(行3至7)。然后，我们对$\mathcal{B}$的每个单元$B_i$进行$CPN[i]$次轨道拷贝操作。

算法 3: 基于精确计算的B骨架的采样方法

Input: G',$\mathcal{V}'$, $n = |V(G)|$, $p[1...|\mathcal{V}'|]$

Output: G'的一个连通子图G_s，满足$|V(G_s)| \approx n$

```
计算B_{G',V'};
N = n − |V(B_{G',V'})|;
while N > 0 do
    基于概率p[i]随机选择i，满足(CPN[i] + 1) · |B_i| ≤ |V'_i|，这里1 ≤ i ≤ |V'|, B_i ∈ B并且V'_i ∈ V';
    CPN[i] = CPN[i] + 1;
    N = N − |B_i|;
end
for 1 ≤ i ≤ |B| do
    重复CPN[i]次Ocp(B_{G',V'}, B, B_i);
end
return 得到的结果图作为G_s;
```

注意，最后得到的结果图可能会包含比$|V(G)|$稍多一些的顶点数目，但是超出部分的顶点数目不会超过while循环中最后一次选择的单元的大小。由于绝大多数自映射划分中的单元包含的顶点数远远小于网络中总的顶点数，因此超出部分的顶点数目通常可以忽略不计。

上述采样方法的一个主要问题在于它的效率。注意，在算法2中计算B骨架时，事实上我们需要进行一系列图同构检测操作，如前文所述，其复杂度至今仍是一个未决问题[8]。因此在最坏情况下，可能并不存在比穷举所有可能的匹配更为有效的方法。

3.3.2.3 基于近似计算的B骨架的采样方法

为了降低计算复杂度，这一节中，我们提出另一种基于近似计算的B骨架的采样方法。该方法在最坏情况下具有线性时间复杂度，因此是非常高效的。

算法4详细描述了所提出的采样方法，其输入和算法3相同。在算法4中，$S[i]$(初始化为1)记录了根据$p[i]$从单元$V_i \in \mathcal{V}'$中需要采样的顶点数目；$Visited[i]$和$Selected[i]$是两个布尔数组，分别表示是否顶点v_i在算法5所示的深度优先搜索(DFS，Depth First Search)过程中被访问和被选择；其它记号的含义和算法3相同。

算法4的主要思想是通过对图G'进行一次深度优先搜索从$\mathcal{V}'$ 中采样n个顶点，并将这n个顶点导出的子图作为原网络的近似版本返回。对于$\mathcal{V}'$ 中的每个

算法 4: 基于近似计算的B骨架的采样方法

Input: G', $\mathcal{V}'$, $n = |V(G)|$, $p[1...|\mathcal{V}'|]$

Output: G'的一个连通子图G_s，满足$|V(G_s)| = n$

1 $N = n - |\mathcal{V}'|$;
2 **while** $N > 0$ **do**
3 　根据概率$p[i]$随机取i，满足$S[i] < |V_i'|$，这里$1 \le i \le |\mathcal{V}'|$并且$V_i' \in \mathcal{V}'$;
4 　$S[i] = S[i] + 1$;
5 　$N = N - 1$;
6 **end**
7 随机取顶点$r \in V(G')$，并设r in V_j';
8 $Visited[r] = true$;
9 $Selected[r] = true$;
10 $S[j] = S[j] - 1$;
11 $n = n - 1$;
12 $DFS(r, n, Visited, Selected, S, \mathcal{V}')$;
13 **return** 由$V = \{v | v \in V(G') \wedge Selected[v] = true\}$导出的子图;

单元，首先需要根据$p[i]$计算出该单元中所需采样的顶点数量(存在$S[i]$中)。然后，从G'的顶点中随机选择一个顶点作为DFS树的根(算法4，行7至11)。接着，用$S[1, ..., |\mathcal{V}'|]$引导DFS过程：对于每个单元$V_i$，最多$S[i]$个顶点会被采样到(算法5，行8至13)。使用DFS的一个目的是为了保证得到的结果图是连通的，这是原网络所具备的一个要求。

算法4试图但是并不保证能够完整地提取出G'的B骨架，因此产生的样本图并不一定来自于样本空间$SS(G', \mathcal{V}', \mathcal{P})$。然而，在下一节的实验部分我们会看到，基于启发式DFS过程的采样方法能够非常近似地还原出原网络的许多重要的统计性质。算法4的另一优势在于它的效率。因为它事实上只是对G' 的一次DFS过程(加上一些预处理过程)，因此它在最坏情况下的运行时间为$O(|V(G')| + |E(G')|)$，是线性的。

3.3.3　实验结果

这一节中我们将通过实验说明所提出的基于B骨架的采样方法的效果。我们使用了3个真实的社会网络数据集**Hepth**，**Enron**和**Net_trace**。这3个数据集同时也在[22] 中使用。表4.1给出了所用数据集的一些统计量的信息。出于采样的目的，我们设发布的信息包括匿名后的网络G'，相应的子自映射划分$\mathcal{V}'$，以及G的顶点数$|V(G)|$。

算法 5: $DFS(v, n, Visited, Selected, S, \mathcal{V}')$

Input: v, n, $Visited$, $Selected$, S, $\mathcal{V}'$

```
foreach u ∈ N(v) do
    if n < 1 then
        return;
    end
    if !Visited[u] then
        Visited[u] = true;
        //设u ∈ V_t'.
        if S(t) > 0 then
            Selected[u] = true;
            S[t] = S[t] − 1;
            n = n − 1;
            DFS(u, n, Visited, Selected, S, V');
        end
    end
end
```

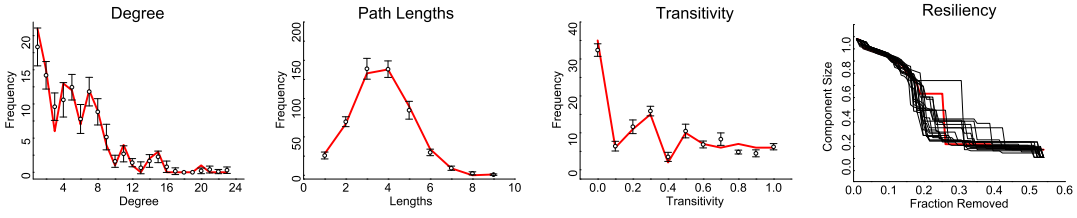
和[22]类似，我们只关注在原图统计性质上的可用性。具体来说，我们设想网络分析人员通过采样的方法从匿名后的网络G'上得到一些样本图，测量每个样本得到所需的统计性质的数值，再对这些数值进行聚集(aggregate，比如取平均值)来作为对原网络相应的统计性质的一个估计。我们检验了在网络数据上分析人员经常需要测算的4种统计性质，分别是：(1)度分布(degree distribution)；(2)路径长度分布(path length distribution)，是指500个随机产生的顶点对之间在网络中的最短路径长度的分布；(3)聚集系数分布(clustering coefficient distribution)，一个顶点聚集系数是指其所有邻居顶点对中相互邻接的邻居顶点对的比例[23]；(4)网络弹性(network resilience)，是指按照顶点度从高到低删除网络中顶点的过程中，网络中最大的连通分量包含的顶点数占整个网络的顶点数的比例[24]。

我们在$k = 5$和$k = 10$的情况下分别对原网络G进行了匿名。然后，我们首先为原网络G 以及分别用算法3和算法4 得到的20个样本图计算了上述4种统计性质。我们发现两种采样方法所产生的结果其可用性几乎是完全一样的。在数据集**Hepth**和**Net_trace**上，近似方法(算法4)甚至比精确方法表现得更好一点。事实上，出现这一结果并不奇怪，因为在算法3中，当一个被采样到的顶点具有很大的度时，在拷贝它的过程中需要拷贝它所关联的所有边。因此如果这个顶

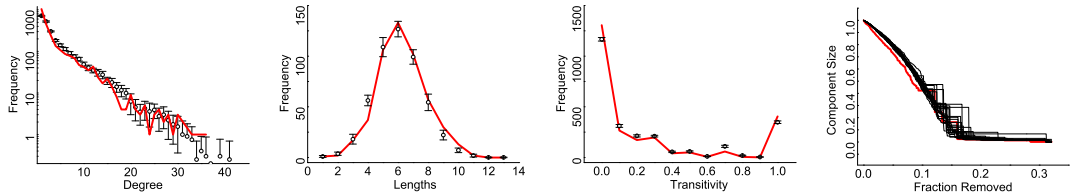
表 3.1: 所用数据集的统计量

统计量	网络数据集		
	Hep-Th	Enron	Net-trace
顶点数	2510	111	4213
边数	4737	287	5507
顶点度最小值	1	1	1
顶点度最大值	36	20	1656
顶点度中位数	2	5	1
顶点度平均值	3.77	5.17	2.61

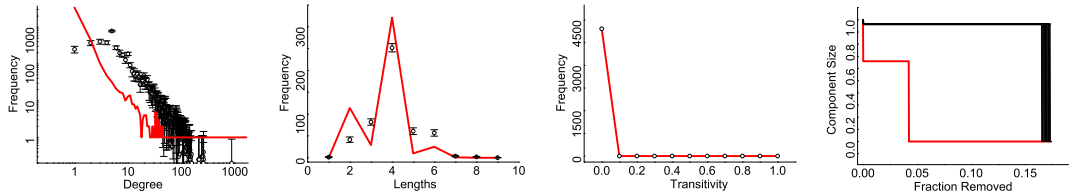
点在原网络中没有自映射等价的顶点的话，那么样本图的近似效果就会大打折扣。由于两种采样方法得到的结果如此接近，在图3.7中，我们只显示了近似方法(算法4)产生的结果。上述实验同样在$k = 10$的设置下进行了重复，产生了相似的结果，因此出于节约空间的考虑这里没有重复显示。从图3.7中我们可以看到对于大部分的可用性度量，基于B骨架采样的方法能够达到较好的可用性质量。



(a)Enron

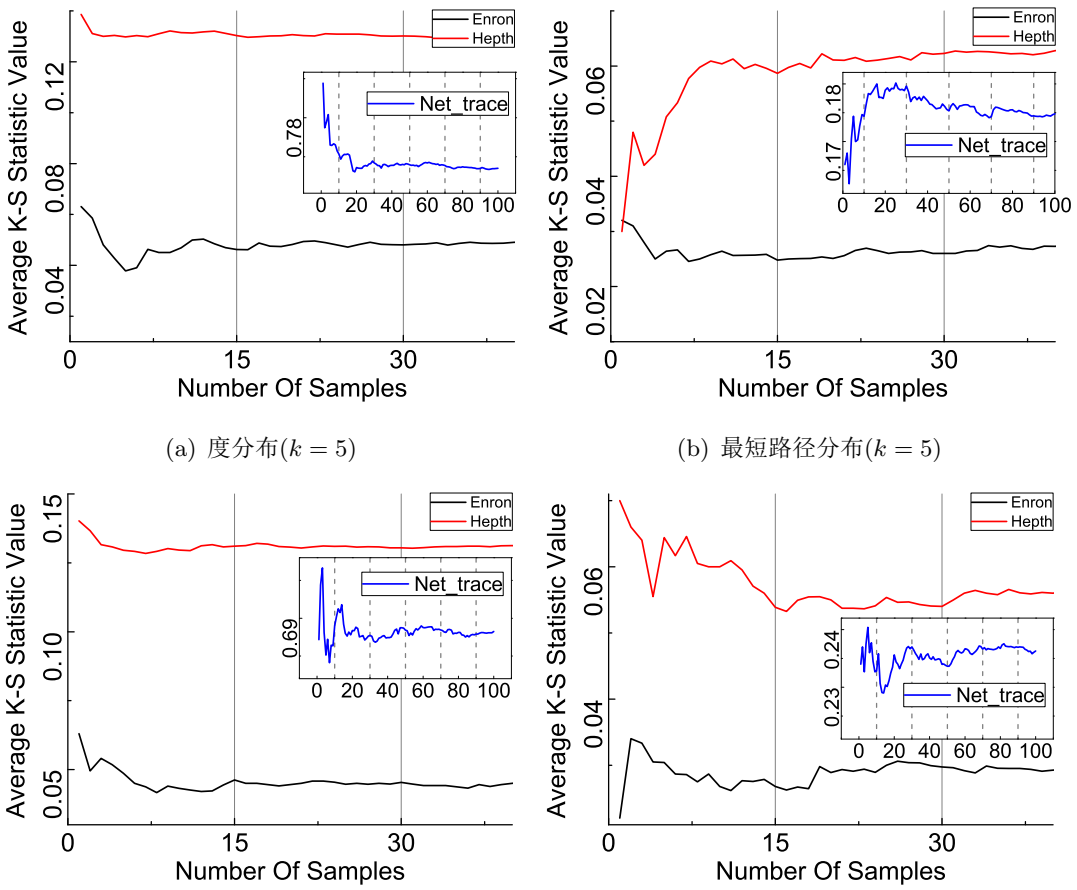


(b)Hepth



(c)Net_trace

图 3.7: 匿名后的网络的可用性实验结果。图中比较了用基于近似计算的B骨架的采样方法得到的样本图(黑色)和原网络(红色)计算统计性质所得到的结果。



(a) 度分布($k=5$)　(b) 最短路径分布($k=5$)

(c) 度分布($k=10$)　(d) 最短路径分布($k=10$)

图 3.8: 当样本数量增长时可用性质量的收敛情况

由于基于B骨架的采样方法是一种随机化算法，因此一个重要的问题是其收敛到稳定状态的速度如何。为此我们需要研究随着样本数量的增加，所测量的统计值的变化趋势。这里，对于度分布和最短路径长度分布，我们计算了原始图和样本图之间的Kolmogorov-Smirnov统计值[9]的平均值。该值越小说明样本图和原始图在所比较的统计性质上匹配得越好。我们计算了将样本数量从1增加到100的过程中，该统计值的平均值在所考虑的两个分布(度分布和最短路径长度分布)上的变化情况。图3.8给出了相应的结果。

如图3.8所示，在所有测试的情况中，可用性度量的值(即K-S的平均值) 会很快地收敛到一个稳定状态；并且在许多情况下，5-10个样本图就已经足够达到一个相对较好的可用性结果了。这说明我们的采样方式是有效并且是可靠的，我们能够通过只从匿名后的网络中采样很少一部分的样本图就达到一个对原网络统计性质的较好的近似。

[9]Kolmogorov-Smirnov统计值(Kolmogorov-Smirnov statistic，简记为K-S)用来计算两个累积分布(cumulative distribution)之间的最大垂直距离。

3.4　模型的进一步改进

在前面几节已经看到，尽管匿名过程的复杂度为$O(|V(G)|^2)$，我们仍然能够通过采样的方法很好地得到匿名后网络的可用性。但是在有些情况下希望对网络所作的修改越少越好，因此需要尽可能地减少新加入的顶点和边的数量。本节将讨论这一问题。

3.4.1　最小化新加入的顶点数量

观察图3.2(a)不难发现，我们确实能够进一步减少新引入的顶点的数量。例如，在图3.2(a)中，当$k=3$时，轨道$V_1=\{v_1,v_2\}$需要被拷贝一次。拷贝后的单元$V_1'=\{v_1,v_2,v_1',v_2'\}$包含4个顶点，对于3-对称的要求而言是冗余的(见图3.4(b))。事实上，我们只需要引入一个新的顶点v_1'就能够使得得到的单元$V_1''=\{v_1,v_2,v_1'\}$满足3-对称的要求。

因此一个有意思的问题便是：如何引入最少数量的顶点使网络达到k-对称？注意到，在图3.2(a)中，v_1和v_2已经构成互为拷贝的关系，所以才会导致拷贝V_1后的k-对称网络中存在冗余。在前面的小节已经提到，网络的B骨架是能够通过轨道拷贝操作得到网络的最小的图，因此它是不包含上述冗余的(即B骨架中不存在任何两个互为拷贝的子结构)。所以，如果我们将匿名过程(算法1)作用于网络的B骨架(即$B_{G,Orb(G)}$)而不是网络G本身的话，那么我们就能够使得新引入的顶点的数量达到最小。

3.4.2　不保护少数Hub顶点

事实上，网络中的少数hub顶点(即度很大的顶点)主导了对网络进行k-对称匿名所需的代价。大部分真实网络(包括社会网络在内)的度分布是极其异构的(heterogeneous)，意味着大部分顶点的度很小，而少数顶点的度很大。文献[21]表明，hub顶点一般位于网络的平凡单元中，因此一般而言hub顶点在网络中不存在与之自映射等价的顶点。因此，为了构建$k-1$个与某个hub 结点v等价的顶点，我们不得不引入$k-1$个顶点以及$(k-1)deg(v)$条边($deg(v)$表示v的度)。

但是，我们真的有必要保护这些hub顶点吗？事实上是不必要的。首先，社会网络中的hub顶点通常代表了众所周知的个体。比如在一个公司内部的电子邮件通信网络中，度最大的顶点很可能就是CEO 的邮件地址。文献[22]表明，hub顶点的这一特殊性使得要对它们进行实体匿名变得非常困难。因此我们认为对少数hub顶点进行保护并非必要。其次，即使攻击者识别出了hub顶点，他仍然不能识别出与hub顶点相连的其他非hub顶点，因此并不会导致网络中其他个体的隐私泄露。

3.4.2.1 f-对称模型

基于上面的论述，我们研究了在允许不保护hub顶点的前提下网络匿名代价的变化情况。为此，我们首先提出f-对称模型的概念(见定义3.5)。

定义 3.5 (f-对称模型). 给定图G和函数$f : Orb(G) \to N$，若$\forall \Delta_i \in Orb(G) = \{\Delta_1, \Delta_2, ..., \Delta_m\}$，有$|\Delta_i| \geq f(\Delta_i)$，则$G$是$f$-对称的，或者说$G$是满足$f$-对称匿名要求的。

显然，k-对称模型是f对称模型的特殊情况(f被定义为将$Orb(G)$中的每个轨道都映射到常整数k)。由于真实网络度分布的异构性，通常而言f应被定义为关于轨道Δ_i中的顶点度d_i的非增函数，即若$d_i \geq d_j$，则$f(\Delta_i) \leq f(\Delta_j)$。$f$-对称模型为网络发布者提供了更大的灵活性。我们设定一个阈值δ，令f将所有顶点度超过δ的轨道映射到1，将剩余的轨道映射到k。在这一定义下的f-对称模型起到了不保护hub顶点的作用。

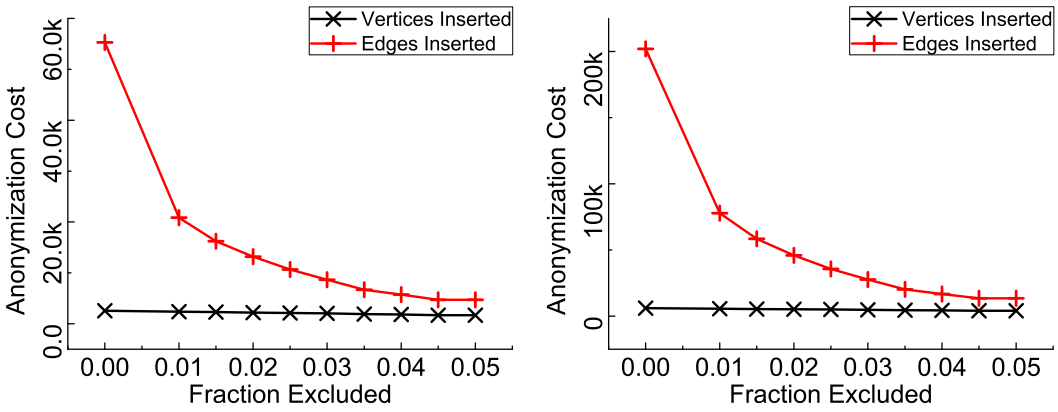
3.4.2.2 实验结果

这一小节给出在**Net_trace**数据集(其度分布极其异构)上不保护hub顶点后匿名过程的代价及匿名后网络的可用性的变化情况。

我们首先研究了匿名代价(用新加入的顶点和边的数量进行度量)和不保护的顶点的比例之间的关系。如图3.9所示，当不保护的顶点(按顶点度从大到小的顺序)数量发生轻微增加时，匿名代价呈现出显著下降的趋势。例如，当$k = 10$时，若不保护度最大的5%的顶点，新加入的边的数量从201913下降至13444，减少了将近94%。即使只不保护度最大的1%的顶点，新加入的边的数量也从201913下降至77749，减少了61.5%。从图3.9还可以看到，通常新加入的边的数量远远超过新加入的点的数量，因此不保护少数hub顶点能够使得网络的匿名代价大幅度下降。

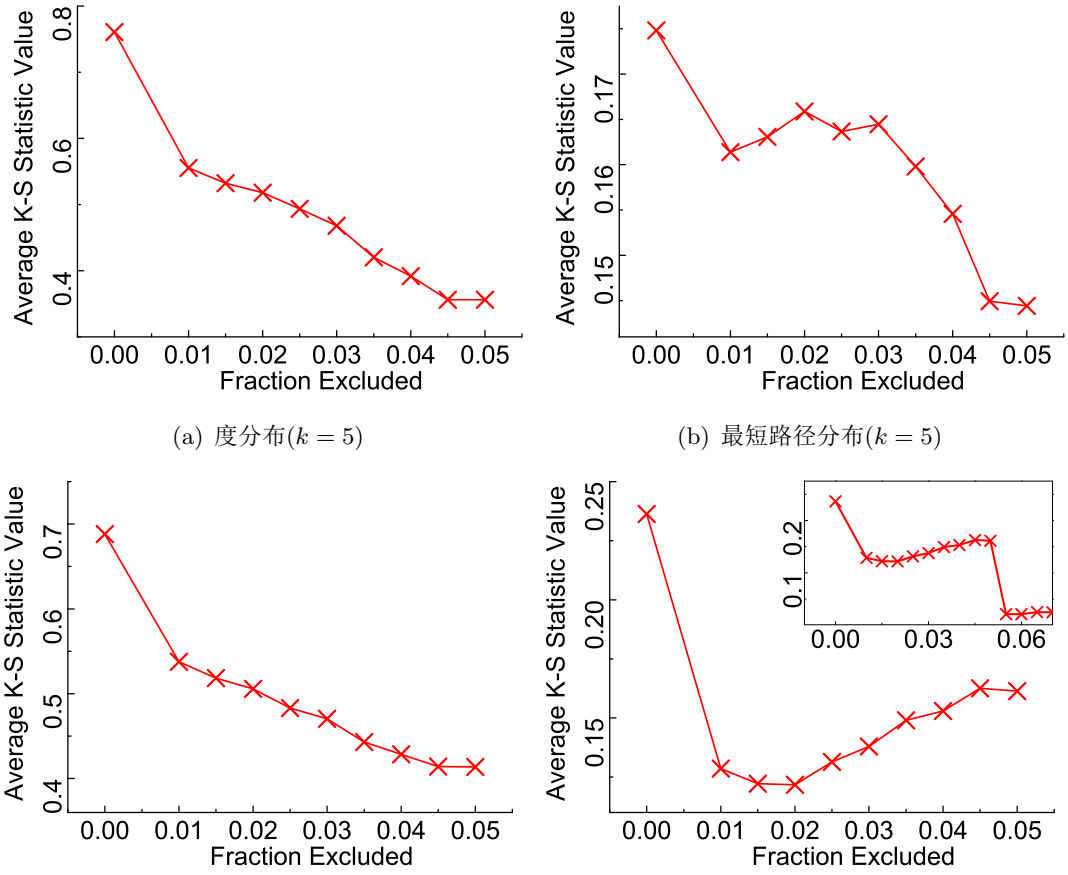
我们进一步研究了不保护少数hub顶点后产生的匿名网络的可用性变化情况。直觉上，由于不保护少数hub顶点后减少了加入到网络中的顶点和边的数量，在匿名后的网络上进行采样能够产生和原网络更为接近的样本图。图3.10证实了这一猜想。这里，我们仍然使用K-S的平均值作为可用性质量的度量。我们同样分别计算了当$k = 5$和$k = 10$时度分布和最短路径分布上的K-S平均值。由于已经验证过采样方法的收敛性情况，这里在每种情况下我们取了100个样本图。

由此我们可以得出结论，当不保护极少数(与网络中总的顶点数相比)hub顶点时，匿名方法的有效性并不会受到多大影响，因为网络中除掉这些顶点之外的顶点仍然满足k- 对称的要求；同时，匿名代价会显著下降，并且匿名后网络



(a) $k = 5$　(b) $k = 10$

图 3.9: 当不保护少数hub顶点时匿名代价的变化情况



(a) 度分布($k = 5$)　(b) 最短路径分布($k = 5$)

(c) 度分布($k = 10$)　(d) 最短路径分布($k = 10$)

图 3.10: 当不保护少数hub顶点时匿名后网络的可用性变化情况

的可用性会有显著地提高。所以，不保护少数hub顶点是合理的，并且是值得的。

3.5 相关工作

社会网络中的实体匿名问题在[18]中首次被提出。文献[18]指出，幼稚匿名的网络对于*主动攻击*(active attack)和*被动攻击*(passive attack)而言都是不充分的。虽然主动攻击的防护难度很大，但是实施主动攻击的难度也很大。与主动攻击不同，被动攻击(也就是这里的基于结构知识的攻击)的实施难度则要小的多，因此引起了研究者们的广泛关注。一些研究者专注于匿名性度量问题(如[25, 26])，另一些则专注于各种匿名技术。文献[27]提出了一种基于随机插入和删除边的技术，可以抵御某些形式的攻击，但是匿名后网络的可用性比较差，因此他们在后续工作[22]中提出了另一种基于网络概化的匿名技术。文献[28]对边随机化技术进行了进一步探索，其目的在于保持住网络的谱特征(即网络的所有特征值)。这种技术虽然大大提高了匿名后网络的可用性，但无法对网络的匿名效果进行有效的度量。其他的匿名技术主要基于传统表结构数据隐私保护邻域的k-匿名模型进行([29, 30, 31])。文献[20]介绍了k-邻域(即邻居节点的导出子图)匿名模型，其基本思想是不断往网络中插入新的边，直到对于网络中的每个顶点，至少存在$k-1$个与其具有同构的邻域的顶点为止。文献[19]介绍了k-度匿名模型，其思想也是不断地往网络中插入新的边，直到对于网络中的每个顶点，至少存在$k-1$个与其度相同的顶点为止。最近，文献[32]又提出了k-自映射匿名模型，其目的与k-对称模型相似，也是要抵御基于任意结构知识的攻击，但是其要求要比k-对称模型高得多，在很多情况下很难也没有必要达到k-自映射匿名模型的要求。事实上，k-自映射模型可以看成是k-对称模型的一种特殊情况。

3.6 本章小结

本章对社会网络中的实体匿名问题进行了深入研究。基于图对称理论，我们提出了k-对称模型，它可以抵御基于任意结构知识的攻击。我们进一步给出了k-对称模型的实现方法，并详细研究了匿名后的网络的可用性问题。基于图对称理论，我们提出了一种新的网络骨架(即B骨架)，并且提出了两种基于B骨架的采样方法。实验证明，基于B骨架的采样方法能够很好地还原出原网络的大部分宏观统计性质。

第四章　社会网络中的最短路径查询问题与基于图对称的索引压缩技术

最短路径查询是社会网络分析中的另一个重要的应用问题：给出网络中的两个顶点，要求快速找出连接两个顶点之间的最短路径。例如，在一个大型的通信网络中，网络中顶点之间的最短路径信息对于整个网络的资源管理有着重要影响 [33, 34]。当网络的规模很大时，在线(online)计算最短路径的时间往往无法忍受，因此必须首先通过离线(offline)预计算来物化网络中的最短路径信息，然后通过有效的索引技术来支持实时的最短路径查询。然而，存储一个N个顶点的网络中所有的顶点之间的最短路径需要$\Theta(N^2)$ 的存储空间。本章提出一种基于图对称理论的索引压缩技术，能够显著地降低存储空间的规模。

4.1　基于宽度优先搜索树的最短路径索引

给定图G和G中的任意两个顶点u，$v(u \neq v)$，设$P(u,v)$是所有G中连接顶点u和v的路径的集合。令$p \in P(u,v)$，若对于任意$p' \in P(u,v)$，$p' \neq p$，满足$|p| \leq |p'|$，则称p是连接顶点u和v的最短路径(*shortest path*)[1]。

经典算法理论[2]指出，给出图G中的顶点u，若从u出发进行宽度优先搜索(BFS, Breadth First Search)，则可以得到一棵以顶点u为根的宽度优先搜索树(*BFS Tree*，简记为BFS树)T_u，T_u中从u到G中其他所有顶点$v \in V(G)-\{u\}$的路径就是u和v之间在G中的一条最短路径。

因此，一种直接的索引策略是为图G的每个顶点u都计算出相应的T_u并进行存储。考察这一索引方式的存储代价：若图G的顶点数为N，则由于需要为每个顶点u存储T_u(空间代价为$\Theta(N)$)，因此空间代价为$\Theta(N^2)$。容易发现，这样的存储方式存在着大量的冗余信息(见例4.1)，这为下面利用图对称理论来进行索引压缩创造了条件。

例 4.1. 在图4.1所示的图G中，属于$Orb(G)$中不同轨道的顶点用不同颜色进行了标记。图4.2进一步给出了图G中以v_1和v_2为根的BFS树T_{v_1}和T_{v_2}。对于任意顶点$v \in V(G)-\{v_1,v_2\}$而言，容易看出v到v_1和v_2的最短路径只在边(v_1,v_3)和(v_2,v_3)上有区别。由于v_1和v_2属于同一轨道中，因此存在至少一个将v_1映射到v_2 的自映射(比如图4.2中所示的自映射g_1)。所以如果我们事先知道并记录下v_1和v_2之间的映射关系，并在查询v到v_2 的最短路径时将v_2映射到v_1，

[1] 显然，u和v之间的最短路径可能不止一条，在本章中，如果没有特别说明，仅考虑其中的一条最短路径。

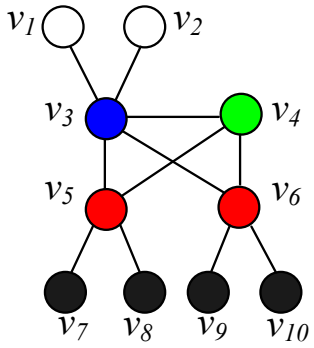


图 4.1: 本章所使用的例图G

就可以通过先查询v到v_1的最短路径p_1，再将p_1 映射回v到v_2的最短路径p_2的方式来实现查询回答。这样我们就能够只存储T_{v_1}而不存储T_{v_2}。

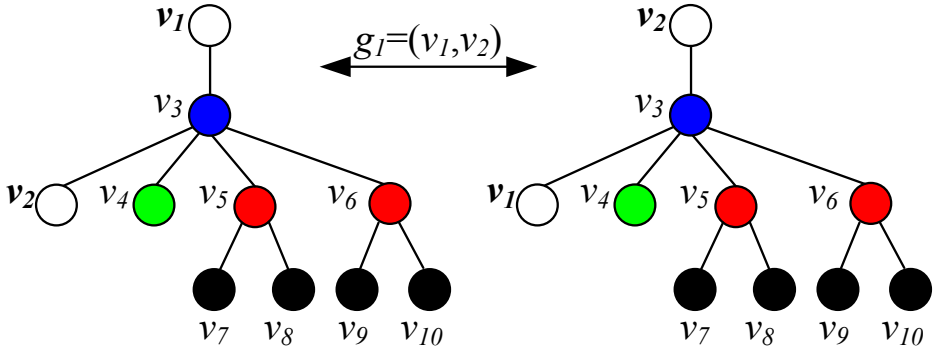


图 4.2: 例图G中的BFS树T_{v_1}和T_{v_2}

4.2 基于轨道的压缩技术

通过例4.1我们已经注意到，利用顶点之间的自映射等价关系，可以对基于BFS树的索引进行压缩。一般地，我们可以为每个轨道，而不是每个顶点 生成BFS树。本节将对这一想法进行详细阐述。

4.2.1 自映射作用下的最短路径

令$g \in S(V)$，设$H = (V(H), E(H))$ 为图$G = (V(G), E(G))$的一个子图，定义子图H在置换g作用下的结果为$H^g = (V(H)^g, E(H)^g)$。一般来说$E(H)^g \subseteq E(G)$不一定成立，但是当g是G的自映射时，$E(H)^g \subseteq E(G)$也成立(引理4.1)，此时H^g也是图G的子图。

引理 4.1. 设$H = (V(H), E(H))$ 为图$G = (V(G), E(G))$的子图，则对于任意自映射$g \in Aut(G)$，$H^g = (V(H)^g, E(H)^g)$ 是图G的子图并且$H^g \cong H$。

证明. 由于$E(H) \subseteq E(G)$，$E(H)^g \subseteq E(G)^g$。既然g是图G的自映射，那么$E(G)^g = E(G)$成立。因此$E(H)^g \subseteq E(G)$，从而H^g是图G的子图。

由于自映射g是$V(G)$到其自身的双射，因此可以构造从$V(H)$到$V(H)^g$的双射f：将每个$v \in V(H)$映射到v^g。显然，对于任意$(x,y) \in E(H)$，$(x^f, y^f) \in E(H)^g$，反之亦然。因此，f是H和H^g之间的同构映射，从而H^g同构于H。□

接着，我们介绍引理4.2，该引理告诉我们，网络中任意两个顶点之间的最短路径距离在自映射作用下能够保持不变。

引理 4.2. *[35]* 给定图G中的两个顶点u和v，令$d(u,v)$为u和v之间的最短路径的距离，则对于任意自映射$g \in Aut(G)$，有$d(u^g, v^g) = d(u,v)$成立。

现在我们可以给出关于自映射作用下的最短路径一条重要定理(定理4.1)，它指出在任意自映射g的作用下，顶点u和v之间的最短路径将被映射到顶点u^g和v^g之间的最短路径。

定理 4.1. 给定图G中的两个顶点u和v，令p为u和v之间的一条最短路径，则对于任意自映射$g \in Aut(G)$，成立p^g是顶点u^g和v^g之间的一条最短路径，这里$p^g = \{(x^g, y^g) | (x,y) \in p\}$。

证明. 由引理4.1可知，p^g一定是u^g和v^g之间的一条路径，并且显然有$|p^g| = |p|$成立。

假设p^g不是u^g和v^g之间的最短路径，那么必定存在连接u^g和v^g最短路径p'，满足$|p'| < |p^g|$。同样根据引理4.1，$p'^{g^{-1}}$是连接u和v的一条路径，并且根据引理4.2有$|p'^{g^{-1}}| = |p'| = d(u^g, v^g) = d(u,v)$。因此$p'^{g^{-1}}$是连接$u$和$v$的一条最短路径，且$|p'^{g^{-1}}| = |p'| < |p^g| = |p|$。这与已知$p$为$u$和$v$之间的一条最短路径相矛盾。因此假设不能成立，$p^g$必定是$u^g$和$v^g$之间的最短路径。□

4.2.2　自映射作用下的BFS树

定理 4.2. 给定图G，令T_v是以G中顶点$v \in V(G)$为根的*BFS*树。若顶点u与v处于同一轨道中，$u \neq v$，则存在一个自映射$g \in Aut(G)$，满足$v^g = u$并且T_v^g是一棵以$v^g = u$为根的*BFS*树，这里$T_v^g = \{(x^g, y^g) | (x,y) \in T_v\}$。

证明. 由于u和v处于同一轨道中，根据轨道的定义，存在至少一个自映射$g \in Aut(G)$使得$u = v^g$，因此只需要证明T_v^g是以顶点u为根的BFS树。

根据引理4.1，T_v^g同构于T_v。设T_v^g和T_v之间的同构映射为f。又由于$V(T_v) = V(G)$，f事实上是$V(G)$上的双射。由引理4.1 的证明过程可知f可以定义为：

将$v \in V(G)$映射到v^g。因此我们可以设$f = g$(从而f是G上的自映射)。所以如果在生成T_v的BFS过程的每一步中将访问的顶点w通过f映射到w^f，将所访问的边(x, y)通过f映射到(x^f, y^f)，则整个过程即相当于从$v^f = v^g = u$出发在网络$G^f = G^g = G$上进行了一次BFS，最后生成的BFS树即为T_v^g。因此T_v^g是以顶点u为根的BFS树。 □

定理4.1和定理4.2证实了例4.1 所描述的索引压缩方法的合理性。下面正式给出这一算法的框架(见算法6)。

算法 6: 算法框架

Input: 图G

Output: **T**(压缩后的BFS树索引)

```
1  计算Orb(G);
2  foreach Δ ∈ Orb(G) do
3      if |Δ| > 1 then
4          任意选择一顶点u ∈ Δ作为该轨道的基本顶点;
5          foreach u' ∈ Δ, u' ≠ u do
6              计算将u映射到u'的自映射g_{u,u'} ;
7          end
8      end
9  end
10 foreach Δ ∈ Orb(G) do
11     设u是Δ的基本顶点，构造u的BFS树;
12 end
```

在算法6中，对于每个非基本顶点u'，需要为其选择从基本顶点u映射到u'的自映射$g_{u,u'}$。但是一般而言，可能存在多个将u映射到u'的自映射。那么，应该选择哪个自映射呢？注意到所选择的自映射是需要作为索引的一部分进行存储的。所以这里在选择自映射时我们主要考虑存储代价的问题。

给定自映射g，为了减少存储空间，可以只存储那些在自映射下发生了变化的点对，即$x^g = y$且$x \neq y$的点对(x, y)。在此存储方式下某个自映射g 的存储代价可以表示为$\Theta(|supp(g)|)$。所以，理论上应该在多个可行的自映射中选择支持集最小的那个进行存储，然而寻找具有最小支持集的自映射一般而言需要枚举出所有可能的自映射，其计算代价是相当大的。因此，为了有效解决这一问题，在具体实现时我们使用了`nauty` [10]算法所生成的网络自映射群的生成集**Gens**

表 4.1: 一些真实网络G的φ_G值

Graph	N	$\|Orb(G)\|$	$r_G\%$	M	$\varphi_G\%_0$	Avg	Max
PPI	1458	1019	69.89	1948	4.11	1.43	46
Yeast	2284	1852	81.09	6646	2.63	1.23	34
Homo	7020	6066	86.41	19811	0.57	1.15	44
P-fei1738	1738	1176	67.66	1876	5.75	1.48	10
Geom	3621	2803	77.41	9461	1.66	1.29	13
Erdos02	6927	2365	34.14	11850	3.46	2.93	142
DutchElite	3621	1907	52.67	4310	2.21	1.90	49
Eva	4475	898	20.07	4652	4.47	4.98	545
California	5925	4009	67.66	15770	1.01	1.48	46
Epa	4253	2212	52.01	8897	0.94	1.92	115
InternetAs	22442	11392	50.76	45550	0.27	1.97	343

[2]。根据文献[10]，**Gens**中的每个自映射都是不可分解的。而根据我们的实验，对于真实社会网络数据，上述自映射生成集中的自映射的存储代价是相当低的。因此，对于基本顶点u和非基本顶点u'，若存在**Gens**中的某个自映射能够将u 映射到u'，我们就选择该自映射作为将u转换成u'的自映射。当不存在这样直接的自映射时我们再枚举这些自映射的组合。实践证明，这样的策略在选择自映射时是高效的。

我们分析一下上述索引方式的存储代价。对于每个基本顶点，需要存储一棵BFS树，所以存储基本顶点所需的空间为$\Theta(|Orb(G)||V(G)|)$。对于非基本顶点，其存储代价由与之相对应的支持集的大小决定。若以$\bar{p}$表示支持集的平均大小，则存储非基本顶点所需的空间为$O((|V(G)| - |Orb(G)|)\bar{p})$。那么$\bar{p}$是多大呢？为了研究这个问题，我们定义了度量$\varphi_G$：

$$\varphi_G = \frac{\max_{g \in ID(G)}\{|supp(g)|\}}{|V(G)|}$$

，这里$ID(G)$表示图G中所有不可分解的自映射的集合。于是，我们有$\bar{p} = \Theta(|V(G)|\varphi_G)$。因此，通过计算$\varphi_G$，我们就可以估计出$\bar{p}$的大小。

表4.1给出了一些真实网络的φ_G值[3]。从表中的数据不难看出，对于这里测

[2]在本章后面的论述中，如无特别说明，**Gens**均指**nauty**算法所生成的自映射群的生成集。(自映射)群$Aut(G)$的生成集(*generator*)是指：令$g_1, g_2, ..., g_m \in Aut(G)$，如果$\forall g \in Aut(G)$，成立$g = g_{i_1} g_{i_2} ... g_{i_k}$ $(i_1, i_2, ..., i_k \in \{1, 2, ..., m\})$，则称集合$\{g_1, g_2, ..., g_m\}$为$Aut(G)$的一个生成集。

[3]每个真实网络的具体信息请参考文献[21]。为了后文的需要，除了φ_G值外，表4.1中同时给出了每个网络的其他一些统计指标，分别为：N，网络的顶点数；M，网络的边数；$|Orb(G)|$，网络中的轨道数；r_G：定义为$r_G = \frac{|Orb(G)|}{N}$；Avg：网络中轨道的平均规模；Max：网络中轨道的最大规模。

试的所有真实网络，其φ_G都在10^{-3}甚至10^{-4}的数量级上。在定义了φ_G后，存储非基本顶点的总代价可以进一步表示为$O((|V(G)| - |Orb(G)|)|V(G)|\varphi_G)$，近似为$O(|V(G)|)$，是非常小的。

我们将上述分析结果表示为下面的定理(定理4.3):

定理 4.3. *对于图G，算法6构建的BFS树索引的存储代价为$O(|Orb(G)||V(G)| + \delta)$，这里$\delta = (|V(G)| - |Orb(G)|)|V(G)|\varphi_G \approx O(|V(G)|)$。*

事实上，可以利用轨道邻接性与可达性理论对基于基本顶点构建的BFS树进行进一步压缩，从而得到压缩的BFS树索引。由于这一理论的细节较为复杂，这里不再做进一步展开，有兴趣的读者可以参看文献[36]。其主要结论如下：设u是某个基本顶点，T_u是以u为根构建的BFS树。令v是T_u中的任意顶点，$v \neq u$。当v所在的轨道$Orb(v)$非弱可达于根轨道$Orb(u)$(即u所在的轨道)时，$Orb(v)$中的任意顶点x与$Orb(u)$中的任意顶点y之间的最短路径长度为常量。这时在T_u中只需保存$Orb(v)$中的任意一个顶点即可，从而对T_u进行了压缩。需要指出的是，基于轨道邻接性和可达性理论所进行的压缩并没有改变定理4.3所陈述的理论上界，但是在实践中，对于对称程度较高(轨道数较少)的网络，可以达到较为明显的压缩效果。

4.3 实验结果

如上文所述，本章所提的索引方法的空间代价主要包括两部分：存储基本顶点的BFS树的代价，记为TS；存储基本顶点到非基本顶点的自映射的代价，记为PS。创建索引的时间代价也包含两部分：寻找非基本顶点和基本顶点之间的自映射的时间代价，记作t_1；为所有的基本顶点生成压缩的BFS树的时间代价，记作t_2。我们以为每个顶点生成BFS树的直接方法作为比较的基准[4]。

4.3.1 在真实网络数据上的实验结果

实验中使用的真实网络数据的基本统计指标见表4.1。

我们首先通过实验比较了基于压缩的BFS树索引进行最短路径查询与实时进行宽度优先搜索两种方法的查询响应时间。实时BFS查询时我们将每个网络存储于内存中，而在基于索引进行查询时索引项则从磁盘上读取。对于每个真实网络，我们随机采样$10,000$个顶点对，对每个顶点对进行最短路径查询并计算响应时间，然后取平均值。表4.2显示了对于不同网络两种方法的响应时间情

[4]为一个一般的网络(非平面网络) 有效地索引最短路径仍然是一件具有挑战性的开放问题。据我们所知，目前为止还没有有效的方法。

表 4.2: 基于压缩的BFS树的最短路径查询

网络	Geom	Epa	Eva	California	Erdos02	PPI	Yeast
有索引(ms)	0.0219	0.0250	0.0438	0.0235	0.0265	0.0235	0.0203
无索引(ms)	0.9	0.9906	0.8844	1.3937	1.5297	0.3125	0.5203
加速比	41.10	39.62	20.20	59.31	57.72	13.30	25.63

表 4.3: 真实网络数据上的压缩比率和索引构建时间

	索引大小(M)					索引创建时间(s)				
网络	原始	TS	PS(K)	压缩后	r_c	原始	t_1	t_2	压缩后	r_t
PPI	12.2	5.9	9.9	6.0	48.9%	1.0	0.5	1.1	1.5	64%
Yeast	29.9	19.4	7.5	19.4	65.1%	2.6	3.8	3.8	7.6	35%
Homo	282.0	210.6	18.2	210.6	74.7%	27.0	1.5	52.5	54.0	50%
Geom	75.0	45.0	8.8	45.0	60.0%	6.6	0.5	14.3	14.9	44%
Eva	114.6	4.6	909	5.5	4.8%	7.7	287.4	2.1	289.5	3%
Epa	103.5	28.0	189.8	28.2	27.2%	9.3	31.2	6.0	37.1	25%

况，并计算了基于索引时回答查询的加速程度。可以看到，基于压缩的BFS树索引的查询回答要显著快于实时的BFS查询回答，约快一个数量级。

表4.3进一步比较了原始的BFS树索引的规模(不作压缩)和压缩的BFS树索引的规模。在表中，压缩比r_c定义为$r_c = \frac{\text{压缩的BFS树索引规模}}{\text{原始的BFS树索引规模}}$，时间比$r_t$定义为$r_t = \frac{\text{创建原始的BFS树索引所需的时间}}{\text{创建压缩的BFS树索引所需的时间}}$。容易看出，压缩的BFS树索引的规模要显著小于原始的BFS树索引的规模。表4.3同时证实了定理4.3 的分析，在绝多大数测试的网络中，为非基本顶点存储自映射的代价至少两个数量级地小于为基本顶点存储压缩的BFS树的代价。表4.3还给出了索引的创建时间结果。显然，创建压缩的BFS树索引所需的时间更长。但是，由于索引创建是离线完成的，以所测试的网络规模考虑，表中所示的时间代价是可以接受的。

前文已经提到，对于不同对称程度的网络，算法的压缩效果会有显著不同。为此，我们绘制了图4.3。这里我们用$r_G = \frac{|Orb(G)|}{|V(G)|}$表达图$G$对称的程度。$r_G$越小，网络越对称。从图4.3 可以看出，网络越对称，基于压缩的BFS树索引所需的存储空间就越小(压缩比例越低)。

4.3.2　在模拟数据集上的实验结果

进一步，我们验证了本章所提出的索引方法的扩展性(scalability)，即随着网络规模的增大，算法性能的变化情况。我们根据BA模型 [16]生成了模拟网络

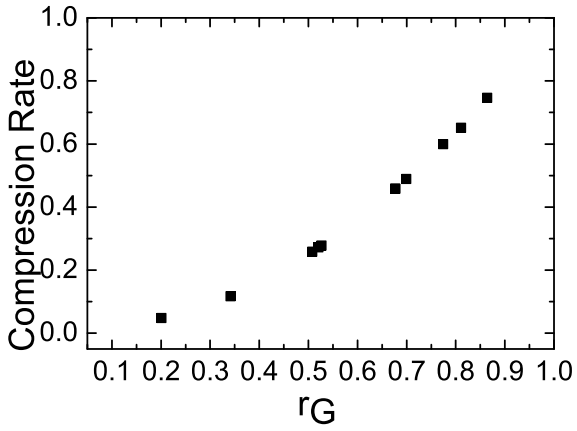


图 4.3: 真实网络的对称性和索引压缩比之间的关系

数据[5]。

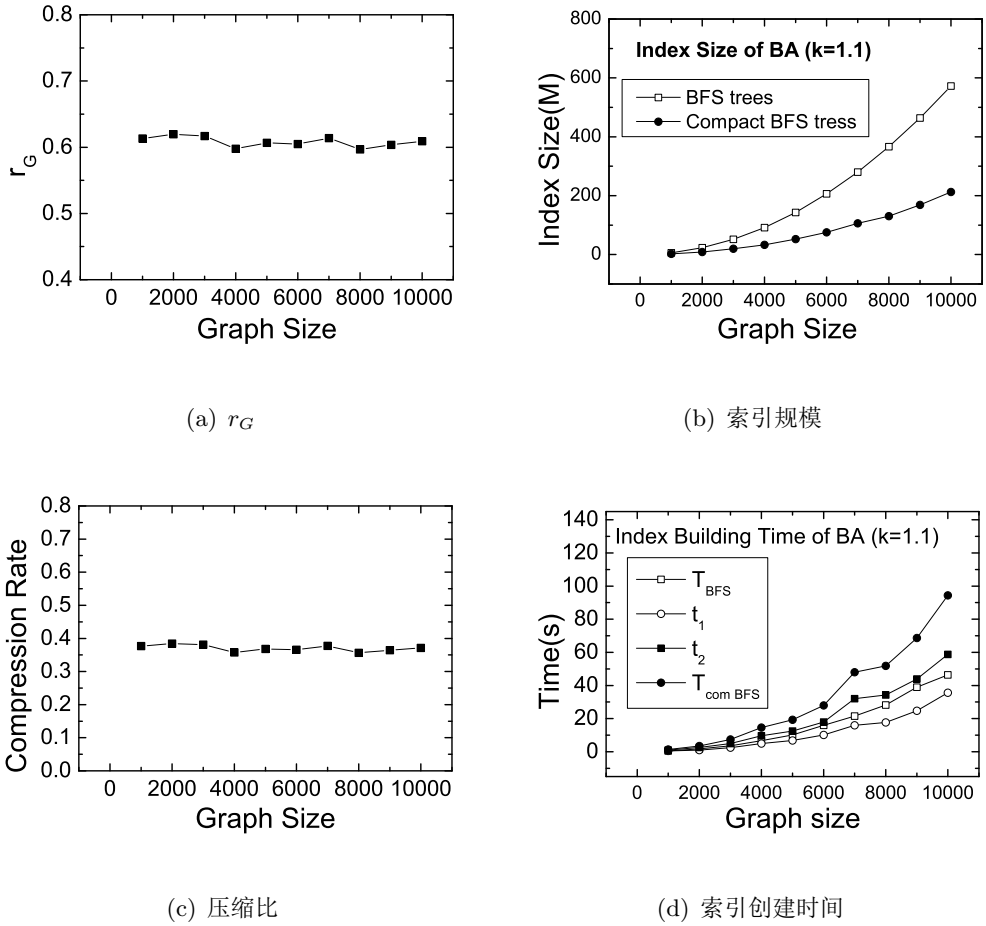
在实验(如图4.4所示)中，取$k = 1.1$(BA模型中每个新加入的顶点被连接到k个已经存在于网络中的顶点)，以1,000为步长，依次生成顶点数从1,000到10,000规模不等的模拟网络。图4.4(a)给出了生成的模拟网络的r_G值。由于这些模拟网络都有着相近的r_G值，因此可以近似认为它们具有相近程度的对称性。

在图4.4(b)中，对于生成的模拟网络，和在真实网络数据上进行的实验相似，我们首先对其原始的BFS树索引的规模和压缩后的BFS树索引的规模进行了比较。可以看到，原始索引与压缩之后的索引规模的绝对差异随着网络规模的增长而增长(图4.4(b))，但是当网络规模增大时，我们发现压缩比却能基本保持常量(图4.4(c))。进一步，在图4.4(d)中，我们给出了创建索引所需的时间代价的情况。可以看到，时间代价基本是以接近线性的方式平滑增长的。上述实验表明，本章所提的索引方法具有良好的扩展性。

4.4 相关工作

最为著名的最短路径算法是Dijkstra算法[37]和Bellman-Ford算法[38, 39]。一般来说，这些算法都假设图可以完全载入内存进行运算。但在现实社会网络应用中，这一假设是不成立的。为了解决不能完全放在内存中的大图上的最短路径查询问题，文献[40]最早提出了对图进行划分和物化的思路。较近的文献[41]针对物化数据能够存于内存的情况，提出了一个更为有效的算法。但是，这些工作的思路都是对原始图，而不是最短路径索引，进行物化处理。其出发点

[5]BA模型是一个广泛应用于模拟真实网络结构的模型：当一个新顶点v加入到网络中时，v被随机地连接到k个已经存在于网络中的顶点。其核心是，新加入网络的顶点v与网络中的某个已经存在的顶点u之间的连接概率正比于顶点u的度数。



(a) r_G　(b) 索引规模

(c) 压缩比　(d) 索引创建时间

图 4.4: 索引方法的扩展性

在于尽可能地减少读取磁盘的次数，但是对于载入到内存中的图，仍然采用实时查询的方法进行处理。因此当图的规模非常大时，必然会在响应时间上存在瓶颈。

在直接物化最短路径索引的研究方向上，一直缺乏相关的工作。直到最近,文献[42]针对道路网络，基于预先计算所有最短路径的思路，提出了在道路网络上高效回答k-近邻顶点查询的相关方法(最短路径查询问题可以视为k-近邻顶点查询问题当$k = 1$时的特殊情况)。但是文献[42]所提出基于四叉树(quad-tree)的索引方式严重依赖于道路网络的一个特定性质–平面性。文献[42]并未给出针对一般的非平面网络的最短路径索引方法。

在现有的图查询研究工作中，可达性查询(reachability query)与最短路径查询关系最为密切，因此这里也介绍一下相关的工作。与最短路径查询不同，可达性查询只需回答给定的点对之间是否存在一条路径(但不需要给出具体路径，对路径长度也没有要求)，所以一般应用于有向图。最简单的回答可达查询问题的方法是对图进行DFS或BFS [2]。但是和最短路径查询的情况相似，用这种方

法对于无法存放在内存中的大图进行实时回答会出现响应时间上的问题，所以现有的大网络上的可达性查询基本采用预先计算的策略。最直接的方法是计算并存储网络的传递闭包[43, 44]，其计算的时间代价为$O(n^3)$，空间代价为$O(n^2)$。当网络规模较大时这一时间和空间代价即使是在离线情况下也是无法接受的。之后的研究者提出了许多其他的用于可达性查询的索引结构 [45, 46, 47, 48, 49]，这方面最新的研究工作已经能够做到在时间和空间代价上都接近线性。

4.5 本章小结

本章对社会网络中的最短路径查询问题进行了深入研究。基于图对称理论，我们提出了对BFS树索引进行无损压缩的有效算法。具体而言，我们只为每个轨道的基本顶点，而不是每个顶点存储一棵BFS树。对于非基本顶点，我们只存储将基本顶点映射到非基本顶点的自映射。进一步，我们可以基于轨道邻接性和可达性理论，对基本顶点的BFS树继续进行压缩。我们通过一系列实验证明了所提出的索引压缩算法的有效性和扩展性。

第五章　社会网络中的社团挖掘问题与基于图对称理论的社团分析

社团结构是一种普遍存在于各类真实网络中的结构特性。挖掘网络的社团结构对于理解网络的功能与行为有着重要作用。图对称理论告诉我们，真实网络是十分对称的，并且这种对称是一种局部对称。真实网络中局部对称性的揭示，为理解社会网络中社团的形成方式提供了新的视角，也为社团分析提供了新的手段。在基于局部对称性的网络演化模型的启示下，本章基于社会学研究中早已发现的弱邻接现象，提出了一种基于权重信息的社团分析方法。与传统方法相比，它可以挖掘出许多更小的但是语义信息更为丰富的社团，并且其效率很高，可以运行于非常大规模的社会网络上。

5.1　研究背景

社团挖掘一直是社会网络分析领域的重要任务之一。目前，还缺乏对于社团的严格统一的定义。在不同的应用场景中，社团可以理解为模块、类、群、组等。例如，WWW可以看成是由大量网站社团组成，同一社团内部的各个网站有着共同的兴趣或话题[50, 51]。揭示网络中的社团结构，对于理解网络结构与分析网络特性来说都是十分重要。

到目前为止，绝大部分已有的社团挖掘算法只依赖于网络的结构特性。从拓扑结构上来看，直觉上，社团内部的点之间的连接应该相对稠密，而社团之间的连接应该相对稀疏。早期的社团挖掘算法[52, 53, 54, 55, 56]都需要预先指定社团的数目，因此无法应用于无法指定社团数目的应用场景。为了解决这一问题，Newman等人[57]引入了一个衡量网络划分质量的度量，称为模块性(*modularity*)。简单地说，模块性度量了模块(社团)内部的边的数量偏离在具有相同顶点度序列的随机网络下的平均值的程度。因此，如果以模块性作为划分质量的标准，那么社团挖掘问题就转化为了对模块性进行优化(即寻找使得模块性达到最大的网络划分)的问题。但是，要精确地找到模块性的最大值及其对应的网络划分方式，其计算代价是非常巨大的。为此，Newman提出了一种基于贪心策略的搜索算法[58]。该算法的一个高效实现[59]可以在稀疏的网络上达到接近线性($O(n\log 2n)$)的时间复杂度(n为网络的顶点个数)，因此是目前唯一能够应用于超大规模(几十万顶点以上)网络的方法。

然而，最近的文献[60]指出，如果把模块性作为网络划分质量的标准，那么为了使模块性的值增大，任何优化算法都会倾向于将规模较小的社团合并为较大的社团。这一情况的发生只取决于模块内部的边的数量，与整个网络的拓扑

结构无关。因此，采用基于模块性的算法得到的社团往往包含了许多更小的社团。文献[60]进一步通过实验验证了这一结果。社团挖掘的目的是为了对挖掘出的社团进行进一步分析。显然，规模庞大的社团对于分析算法的效率、社团的可视化等而言都是一个挑战。从另一方面来看，大的社团从语义层面也不如小的社团更有意义。事实上，在文献[59]中，Newman 等用所提的算法对一个409, 687个顶点，2, 464, 630条的网络进行了社团挖掘，结果得到的前4个社团规模分别为114538, 92276, 78661, 54582，占了所有顶点的83%。很显然，这样的社区结构粒度较粗，每个社团都还有进一步细分的必要和可能。

5.2 真实社会网络中的局部对称性与弱邻接现象

前文已经提到，文献[3]的开创性工作表明，真实网络是十分对称的。在其后续工作[13]中，作者进一步指出，真实网络中的对称是一种局部对称性。所谓局部对称性，是与全局对称性相对而言的。对于给定的图G，若存在一个不可分解的自映射$g \in Aut(G)$，使得$supp(g) = V(G)$，则称G是全局对称的，否则称G是局部对称的。真实网络的局部对称性表明，很难找到一个自映射，它只交换网络中相距很远的点对，而保持其它顶点不动。文献[17]进一步提出了基于相似链接模式的网络生成模型，揭示了局部对称性产生的内在机理。在相似链接模式下，网络中将存在大量的局部对称子团(完全子图或完全二分子图)，这与真实网络中的实证研究结果相吻合[13, 17]。这为社团分析提供了新的视角。如果我们把这些对称子团想象为小的社区，相似链接模式即表明，网络中的个体倾向于与社团内部的个体进行交互。表现在宏观上，社团内部的个体之间往往联系较强，而社团之间的个体之间则联系较弱，这就是社会学中所谓的弱邻接现象[61]。事实上，在许多社会网络中，顶点之间的边是带有权重的信息的。例如，在文献合作者网络中，边的权重一般表示作者之间的合作次数；在通讯网络中，边的权重一般表示通讯的时间长短，等等。在这些情况下，权重信息体现了两个个体之间联系的强弱程度。但是在目前的社团挖掘算法中，这一信息却被忽视了。下一节我们提出一种基于个体之间联系强弱程度的启发式社团挖掘方法。我们将看到，新的社团挖掘方法能够得到许多规模较小，语义信息更为丰富的社团。

5.3 启发式社团挖掘算法

给定图G，设$w : E(G) \to R^+$为边集所带的联系强弱程度的信息[1]。我们的基本想法是设定一个阈值，然后比较边的权重与该阈值来确定该边是应该属于

[1]为方便起见，后文简称权重信息或权重。

社团内部还是社团外部，最后取所有在社团内部的边的导出子图，以它的每一个连通分量作为一个社团。但是这个想法在实践时存在两个必须克服的困难：
(1)阈值的选取对于划分结果有很大影响，过低的阈值会导致社团数量少而社团的规模大，过高的阈值的作用正好相反；
(2)阈值的选取对权重分布的依赖性很大，当权重的分布较广时，容易通过调整阈值来选取一个比较合理的划分，但当权重的分布很窄(比如只有有限的几个值)时，不论如何调整得到的结果都可能不理想。

因此，在考虑利用权重信息进行社团划分时，必须使得算法与权重的分布情况独立。为此，我们提出如算法7所示的启发式算法。

算法的基本思想是自底向上对顶点进行聚类。算法首先对图中的边按照其权重从高到低进行排序(行1)。然后依次扫描每一条边e。若e的两个端点u和v目前都不属于任何一个已经找到的社团，则创建一个新的社团H，将u和v加入H中，并对其进行扩展(行4至15，扩展过程稍后讨论)。这里我们用$H \cup U$表示将顶点集U中的顶点及其与图H中的顶点的连边加入到H中。由于扫描时要求e的两个端点都未被划分到任何社团，因此在扫描结束时可能仍有顶点未被归类，算法需要对这些顶点进行进一步处理(行16至27)。对于任何一个未被归类的顶点x，考察它的邻居顶点所在的社团。这里，x的任何邻居y必定已经被归入到某个社团中，否则边(x,y)的两个顶点均未被归类，那么(x,y)必定会在之前的扫描过程中被归入一个新创建的社团，这与x未被归类是矛盾的。设x的邻居所属的社团为$H_1, H_2, ..., H_m$，算法计算x与各H_i $(1 \geq i \geq m)$ 的连边的平均权重W_i(行17至23), 即

$$W_i = \frac{\sum_{y \in V(H_i) \cap N(x)} w(x,y)}{|V(H_i) \cap N(x)|}$$

。算法然后将x归入W_i值最大的社团中(行24至26，这里$\mathcal{C}[mCID]$表示在$\mathcal{C}$中取ID为$mCID$的社团)。

在创建一个新的社团后，算法将对其进行扩展。扩展过程的伪代码描述如算法8所示。这里，算法首先计算u和v的公共邻居的集合$N(u) \cap N(v)$，并将这些公共的邻居加入到H中(行1至6)。然后算法以当前的H为基础，继续向外进行扩展(行7至17)。具体而言，对于与当前H有连边且未被归类的任何顶点z，算法计算z与H内部的顶点之间连边权重的平均值$avg(z,H)$，并将其与当前H内部的顶点之间连边权重的平均值$avg(H)$进行比较，若$avg(z,H) > avg(H)$，则将z归入H。

5.3.1　算法分析

这里分析一下算法的时间复杂度。对$E(G)$进行排序的代价为$O(|E(G)| \log |E(G)|)$；

算法 7: 算法框架

Input: 图G, w

Output: 找到的社团集合$\mathcal{C} = \{H|H$是G的子图$\}$

将$E(G)$中的边e按$w(e)$从高到低进行排序;
$C = \emptyset$;
$curCID = 0$;
foreach $e = (u, v) \in E(G)$ **do**
    if $u.cid == -1 \wedge v.cid == -1$ **then**
        $curCID = curCID + 1$;
        创建一个新的社团H;
        $H.cid = curCID$;
        $H = H \cup \{u\} \cup \{v\}$;
        //对社团H进行扩展;
        $EnlargeCommunity(H, w, e, \mathcal{C})$;
        $\mathcal{C} = \mathcal{C} \cup \{H\}$;
    end
    $E(G) = E(G) - \{e\}$
end
foreach $x \in V(G)$ **do**
    if $x.cid == -1$ **then**
        设WM和CM分别为维护权重和计数信息的数组;
        foreach $y \in N(x)$ **do**
            $WM[y.cid] = WM[y.cid] + w(x, y)$;
            $CM[y.cid] = CM[y.cid] + 1$;
        end
    end
    $mCID = argmax_{cid}\{WM[cid]/CM[cid]\}$;
    $x.cid = mCID$;
    $\mathcal{C}[mCID] = \mathcal{C}[mCID] \cup \{x\}$;
end

对$E(G)$进行扫描的代价为$O(|E(G)|)$；在扫描过程中，对于每个社团只会调用一次$EnlargeCommunity$过程，而在每次调用中，第一个for循环的代价为$O(|N(u)|)$，计算$avg(H)$的代价为$O(|E(H)|)$，第二个for循环的代价为$O(\sum_{v \in V(H)} |N(v)|)$，因此总的时间代价为$O(\sum_{H \in \mathcal{C}} |N(u)| + \sum_{H \in \mathcal{C}} |E(H)| +$

算法 8: 社团扩展过程$EnlargeCommunity(H, w, e, \mathcal{C})$

Input: 社团H, w, 边e, 社团集合$\mathcal{C}$

Output: 找到的社团集合$\mathcal{C} = \{H|H是G的子图\}$

```
1  foreach x ∈ N(u) do
2      if x.cid == −1 ∧ x ∈ N(v) then
3          H = H ∪ {x};
4          x.cid = H.cid;
5      end
6  end
7  计算H内部的顶点的联系强度的平均值avg(H);
8  U = ∅;
9  foreach y ∈ V(H) do
10     foreach z ∈ N(y) do
11         if z.cid == −1 then
12             计算z与H内部顶点的联系强度的平均值avg(z, H);
13             if avg(z, H) > arg(H) then
14                 U = U ∪ {z};
15             end
16         end
17     end
18 end
19 H = H ∪ U;
```

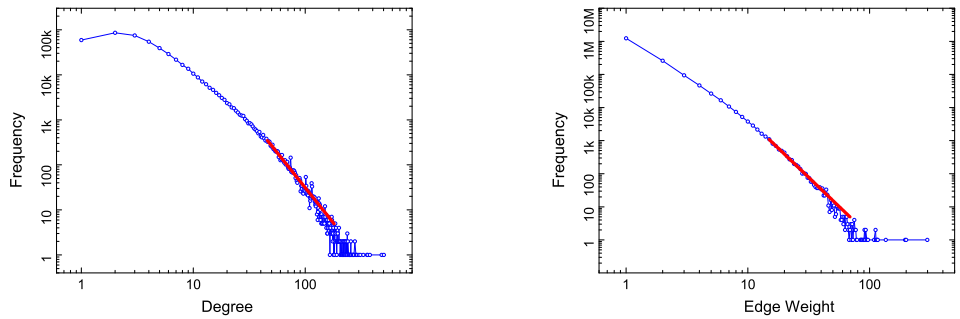
$\sum_{v \in V(H)} |N(v)|) = O(|E(H)|)$；对扫描过程中未归类的顶点进行归类的代价$O(|E(G)|)$。所以算法的时间复杂度为$O(|E(G)| \log |E(G)|)$。在稀疏的网络中(绝大部分真实社会网络都是稀疏网络)，$|E(G)| = O(|V(G)|)$，因此时间复杂度可以减少到$O(|V(G)| \log |V(G)|)$。因此，我们的算法是接近线性复杂性的，能够处理较大的有权网络。

5.4　实验结果

我们通过实验对提出的启发式社团挖掘算法的有效性进行了验证。

5.4.1　实验环境与数据集

所有实验均在配置为Intel Core Duo 2.1GHz的CPU和2G内存的PC机上进



(a) 顶点度数分布 (b) 边权重分布

图 5.1: 实验数据集中顶点度数和边权重的分布情况

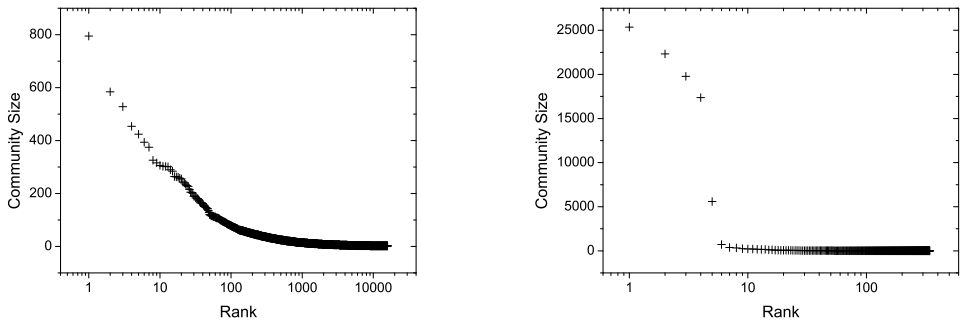
行。我们用Java 1.6实现了所提出的启发式算法。Newman算法的C++实现从[62]获得。实验数据集取自最近的DBLP数据库。我们抽取了它的文献合作者网络(coauthorship network)，包含574, 578个顶点，1, 824,798条边。我们进一步抽取了网络的规模最大的连通分量，最终得到的带权图包含481, 433个顶点，1, 719, 320条边，边的权重表示合作者之间的合作次数。

图5.1显示了网络中顶点度数和边权重的分布情况。从图中可以看出，顶点度和边的权重分布满足幂律分布(幂指数分别为$\gamma_k = 3.1$和$\gamma_w = 3.5$)，与前人结论相吻合[63]。权重满足幂律分布进一步证明社会网络中个体之间的交互是局部的。

5.4.2 社团挖掘的宏观分析

我们首先用Newman等提出的基于模块性优化的算法[59]对数据集进行了社团挖掘，找到了4769个社团。其中规模最大的社团包含96733个顶点，规模最小的社团包含3个顶点，社团的平均规模为100.951。规模在10000以上的社团有6个，其规模分别为96733，79981，57004，42878，36703，12751，包含了全部顶点的65.6%。这说明，单纯依赖结构信息，绝大多数顶点在模块性最大时，被分配到一些非常巨大的社团。

我们用所提出的启发式算法对规模最大的社团(包含96733个顶点)进行了进一步的挖掘，得到了15869个更小的社团，用时仅需3.2秒。规模最小的社团包含2个顶点，规模最大的社团包含795个顶点，社团的平均规模为10.8。作为对比，我们参照文献[60]的做法，在最大社团的内部用Newman算法重新进行了挖掘，用时需3662秒，且只得到了343个更小的社团。最小的社团包含2个顶点，最大的社团仍然包含25362个顶点，社团的平均规模为282.0。可见，从性能上来看，本章提出的方法明显优于Newman算法。



(a) 本章所提的算法　　(b) Newman的算法

图 5.2: 两种方法挖掘出的最大社团内部的小社团的rank-size分布图

为了对比基于权重和基于拓扑结构的方法对于隐含社团的识别能力，我们为上述两种方法绘制了相应的rank-size分布图(如图5.2所示)。其中横坐标是不同社团挖掘算法所得社团的规模排名，纵坐标是相应排名社团的规模。可以明显看出，Newman方法所得到的社团规模分布范围明显宽于我们的方法。进一步，可以看出，对于前几名排名，Newman方法所对应的社团都显著大于我们的方法。从图5.2(b)可以看出，排名前5的大社团占据了整个原社团规模的绝大多数，达到93.5%。而相比较而言，利用我们的方法得到的社团，排名前5的大社团只占原社团规模的2.9%，且直到排名12718，在其之前的社团累积规模才能占到总规模的93.5%。因此，本章所提的算法可以进一步发现更多的规模较小的社团。

5.4.3　社团挖掘的微观分析

我们将进一步通过分析说明，利用我们方法得到的小社团是有意义的，利用本章的方法得到的较细粒度的网络划分是能够体现蕴涵于社会网络中的语义内涵的。为此，我们分别选择了数据库(DB)和数据挖掘(DM)领域两位著名的学者M. Stonebraker和Jiawei Han所在的社团进行观察。

图5.3显示了M. Stonebraker所在的社团[2]，该社团包含216位学者。图5.4显示了Jiawei Han所在的社团，该社团包含795位学者。可以看出，Stonebraker社团包含了绝大多数DB领域的重要学者。可以明显地看到，许多数据库领域的著名学者如J. Gray, J. D. Ullman, D. J. Dewitt等都在这一社团中。而Jiawei Han社团包含了绝大多数DM领域的知名学者。而这两个社团，在Newman算法下，将会隐藏于一个多达25362多个点的大社团中。相对于大社团的规模，这两个社团的总规模不到其1/20。因此如果以大社团为单位进行分析，是很难捕捉这些丰

[2]图5.3和图5.4使用免费软件Pajek[1]绘制。

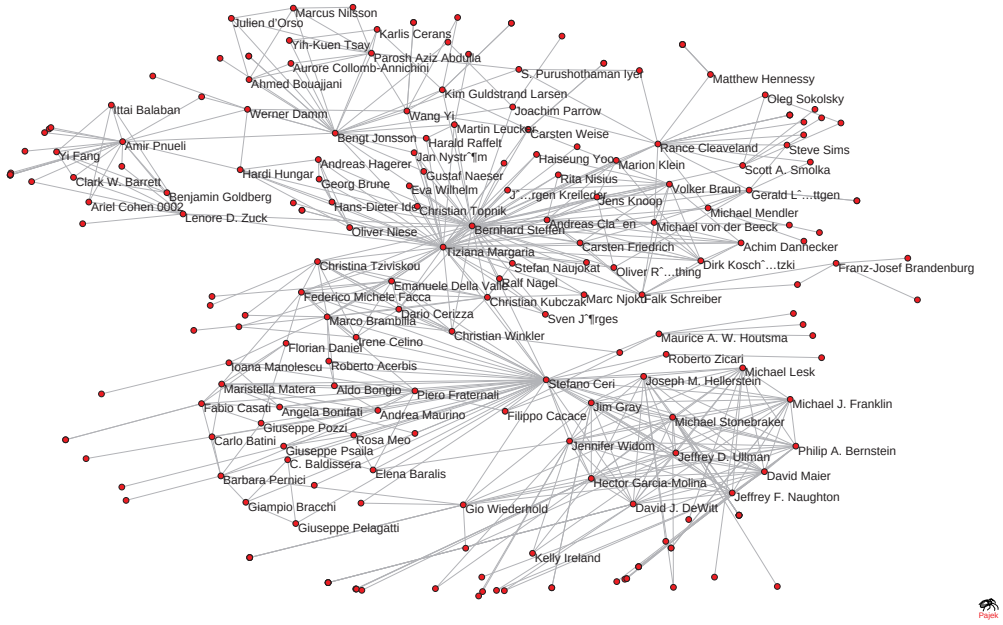


图 5.3: M. Stonebraker所在的社团(仅显示了顶点度数超过3的作者的姓名)

富的语义信息的。另一个有意义的事实是，虽然我们在算法中仅仅使用了合作次数这一信息，而从未使用任何学者的其它背景信息，比如其研究领域等。却依然可以很自然地将网络划分为了不同研究主题的社区。这一事实充分说明，基于局部对称性的网络演化机制是符合真实社会网络的演化规律的。

进一步，我们对上述社团内部作了细致分析。通过对图5.3的观察，我们发现在DB领域，学者间的合作具有明显的地域性。比如欧洲和美国的学者之间的合作主要通过Stefano Ceri教授在进行。需要指出的是得出这一事实并不简单。在原始的大社团中，Stefano Ceri的度数为172，远大于其在上述Stonebraker社团中的度数47；而Stonebraker社团中与Stefano Ceri合作的作者的平均度数为11，远小于他们在原来大社团中的平均度数17.1。如果直接以未细分的大社团为分析对象，这些事实都会对得到“Stefano Ceri连接欧洲和美国两个DB研究团体”的结论产生干扰。

从图5.4显示的Jiawei Han所在的社团中，可以看到，该社团大多为华人学者，且欧美学者与华人学者之间的合作较少。就地域性而言，不如图5.3的社团来得明显。中国大陆、北美、新加坡、澳洲的华人学者之间都有着大量的合作。整个Jiawei Han社团在原大社团中，与其他社团有7145条合作关系，是整个Jiawei Han社团内部1706条边的4.2倍。这是一个不可忽视的比率，足以干扰我们观察出上述信息。因此，与上述Stonebraker社团分析类似，如果以Jiawei Han社团所在的大社团为单位进行分析，由于受到大社团中其它社团的影响, 这些信息也将隐藏于大社团中，难以发现。

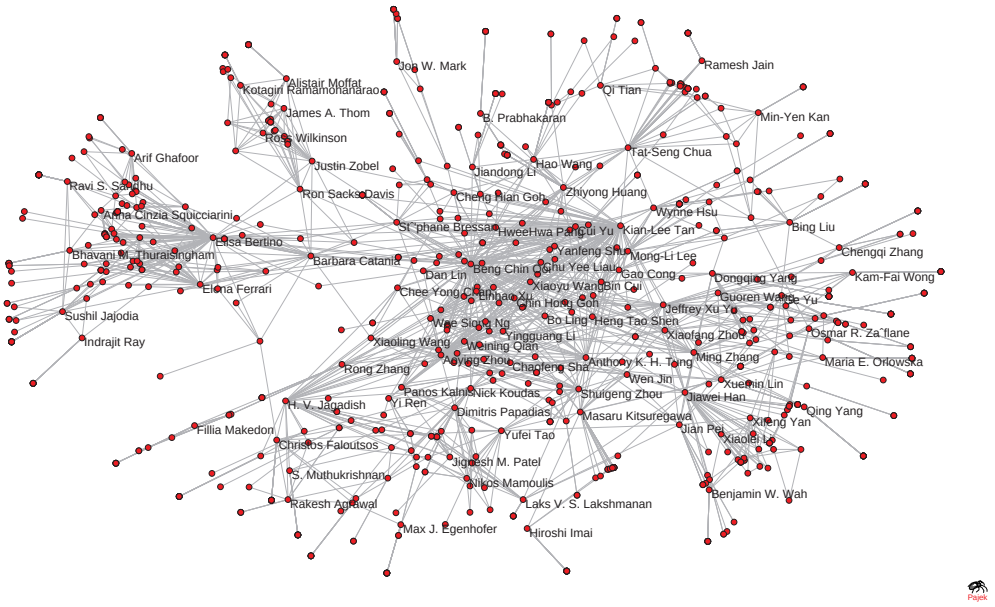


图 5.4: Jiawei Han所在的社团(仅显示了顶点度数超过10的作者的姓名

5.5　相关工作

在社团挖掘方面，基于网络拓扑结构的算法已经有了许多研究。早期比较有名的算法包括Kernighan-Lin算法[52]和谱平分法[53, 54, 55, 56]，但是这些算法都需要预先知道网络中社团的个数。2004年至今被广泛接受的是Newman等人基于模块性度量提出的一系列算法，包括基于介数(*betweenness*)的自顶向下对网络进行分解的GN算法[57]，以及基于模块性优化的自底向上对网络中的顶点进行合并的Newman快速算法[57]及其改进版本[58]。

最近，权重信息开始受到社会网络研究者的重视。文献[63]首先对大规模移动通讯网络的权重统计性质进行了研究，包括权重的分布，基于权重的传播(diffusion)过程，基于权重的网络健壮性(resilience)分析等。文献[64]提出了一种基于权重的产生式模型来生成具有社团结构的网络。但就我们所知，目前尚没有基于权重信息的在大规模社会网络中进行社团挖掘的有效算法。

5.6　本章小结

本章对如何利用社会网络中的权重信息进行社团挖掘的问题进行了初步探索。通过分析网络的局部对称性与网络中的弱邻接现象的内在联系，本章提出了一种利用个体之间的联系强弱程度的启发式算法来对大的社团进行进一步分解，从而发现了其中隐含的规模更小，但是语义更丰富的社团。在DBLP文献合作者网络上进行的实验验证了算法的有效性。

第六章　总结与展望

在前面的章节中，我们已经看到图对称理论在社会网络分析的若干重要问题中都有着有效的应用，基于图对称理论的解决方案往往比已有的解决方案更为普适和高效。本章将对全文进行总结，并对应用图对称理论解决实际的社会网络问题的进一步研究工作进行评述和展望。我们认为，图对称理论在社会网络分析中有着广阔的应用前景，本文的工作只能算是这方面研究工作的一个起步。我们希望本文的介绍能对未来的研究者有所启示和帮助。

6.1　总结

本文对图对称理论在社会网络分析若干重要问题中的应用进行了深入研究。针对社会网络中的实体匿名问题，本文基于图对称理论提出了能够抵御任何基于结构知识的攻击的k-对称模型，并详细研究了匿名后网络的可用性问题；针对社会网络中的最短路径查询问题，本文基于图对称理论提出了一种对BFS树索引进行无损压缩的技术，大大降低了存储索引所需的空间代价；针对社会网络中的社团挖掘问题，本文基于图对称理论，通过分析社会网络的局部对称性与弱邻接现象的内在联系，提出了利用网络中个体之间的联系强弱信息进行社团分析的算法，其能够找到比传统方法规模更小但是语义信息更为丰富的社团。本文的主要贡献在于首次将网络对称性这一真实网络中普遍存在的重要性质应用到了实际的社会网络分析(尤其是社会网络数据管理)问题的解决中。

6.2　进一步研究工作

这里，作为未来工作，我们简单讨论图对称理论在其它社会网络分析重要问题中的潜在应用。

6.2.1　基于对称的顶点重要性刻画

在一个社会网络中，顶点的重要性显然是不同的。直觉上来看，度大的顶点应该更为重要，因为其关联了更多的其他顶点。例如，在通信网络中，度大的顶点往往代表了通信枢纽。然而，由于真实社会网络度分度通常满足幂律分布[16]，网络中度大的顶点往往只是少数。因此，对于网络中剩余的顶点，如何度量其重要性，是一个很有意义的问题。在这方面已经有大量的研究工作，包括顶点中心性[65]、PageRank[66]、HITS[67]等方法。这些方法都在一定程度上

考虑了网络的拓扑结构信息，因此比单纯依靠顶点的度信息进行重要性度量的方法更进了一步。

图对称理论对于顶点重要性度量给出了新的角度。事实上，在很多场景下，如果一个顶点在网络中不存在与它结构完全等价的顶点，那么破坏这一顶点就必然会对网络整体的功能造成某种影响。但是这些顶点在其他度量方式下可能无法得到较高的重要性分值。例如，位于亚洲和北美洲之间的海底通信节点，由于地理位置的限制，导致其度和中心性都很小，因此其重要性在基于度和中心性的度量下会被忽视掉。如何基于对称信息综合考虑顶点在网络整体拓扑结构中扮演的角色，并进一步给出合理的重要性分值，是一个值得研究的课题。

6.2.2 基于对称的网络健壮性研究

社会网络的健壮性分析是另一个很有意义的问题，有着很强的理论和实践价值。一方面，对于某些社会网络，如通信网络，我们希望其健壮性越高越好。但是，另一方面，对于另外一些网络，如传染病传播网络，我们则希望能够尽可能地降低其健壮性。因此，这就需要有一套合理的网络健壮性的评价体系，并在此基础上派生出行之有效的提高或者降低网络健壮性的方法。在文献[24]中，作者分析了基于ER模型[14]和BA模型[16]产生的网络在随机攻击(每次任意从网络中去掉一个顶点)和特定攻击(按度从大到小从网络中去掉顶点)模式下网络的健壮程度(以网络中最大的连通分量包含的顶点数量的比例作为量化指标)。但是文献[24] 并没有给出任何能够提高或者降低网络健壮性的实际方法。

网络健壮性从图对称理论的角度可以重新进行考察。从图对称角度而言，网络脆弱的一个重要的原因在于网络中存在着许多结构不可替换的顶点。对这些顶点进行攻击就有可能导致网络的行为和功能发生显著变化。因此提高网络健壮性的关键就在于如何降低网络中顶点的这种结构独特性，或者说，如何提高网络的对称性；反之，降低网络健壮性的关键，显然就是要降低网络的对称性。

6.2.3 基于对称的网络演化研究

社会网络的演化问题已经吸引了越来越多研究者的关注，并且已经从很多角度被研究过，比如网络的规模，平均度，平均最短路径等指标[15, 68, 23, 69]。然而，这些研究工作都只从网络拓扑结构的某一特性进行了考察，却忽视了网络整体拓扑结构的宏观特性。若从网络对称性的角度来考察社会网络的演化机制，有可能得到更为全面和客观的结果。事实上，在物理学中，复杂系统的演化通常被刻画为对称破缺的过程[70, 71, 72, 73]。社会网络作为复杂系统的一种，其对称破缺的机制究竟如何，这需要进一步的研究工作来予以解答。

6.2.4　图对称理论在子结构模式枚举中的应用

当社会网络数据规模足够大时，显然需要基于磁盘的有效存储方式来对网络数据进行管理，这在本文第四章的研究工作中已经予以证实。但是社会网络数据上查询分析任务并不止最短路径查询这一种。对于不同形式的查询，需要不同的数据索引机制和查询应答策略。因此，研究针对大规模社会网络数据的图数据库是非常必要的。已有的图数据库研究工作包括了频繁子图模式挖掘[74, 75]、子图模式查询处理[76, 77]、以及图索引的创建与维护[78, 79]等。事实上，在社会网络分析中，这方面的工作可以进行平行的迁移，因为这些任务是许多后续的社会网络分析任务的基础。但是应该看到，将基于小图的图数据库技术改造为适用于大型社会网络数据管理的技术手段，必然会遇到许多新的问题和挑战。

在上述图数据库上的基本操作中，子结构模式枚举居于核心地位。合理利用真实社会网络的对称信息，子结构模式枚举问题可能可以通过新的途径来解决。事实上通过观察我们发现，从自映射等价的节点出发所作的模式枚举是同构的。这一发现启示我们只需针对自映射划分中每个自映射等价的单元，选择其中一个顶点展开枚举即可，从而可以大大加速模式枚举过程。显然，可以预先计算对称信息进行存储而不必在模式枚举过程中每次都去实时地进行计算。这又涉及到如何有效地索引对称信息从而加快模式枚举过程中查询对称信息所需的代价的问题。这些都将作为进一步研究的方向。

参考文献

[1] V. Batagelj and A. Mrvar, *Pajek - program for large network analysis*, Connections, **21**(2):47 – 57, 1998.

[2] T. H. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*, MIT Press, Cambridge, MA, USA, 2001.

[3] B. D. MacArthur and J. W. Anderson, *Symmetry and self-organization in Complex Networks*, arXiv:cond-mat/0609274, 2006.

[4] C. Godsil and G. Royle, *Algebraic Graph Theory*, Graduate Texts in Mathematics, Vol.207, Springer, 2001.

[5] J. J. Rotman, *An Introduction to the Theory of Groups*, Fourth Edition, Springer, 1999.

[6] M.Klin, C.Rücker, G.Rücker and G. Tinhofer, *Algebraic Combinatorics in Mathematical Chemistry. Methods and Algorithms. I. Permutation Groups and Coherent (Cellular) Algebras*, Technical Report, TUM-M9510, Technische Universitat Munchen, 1995.

[7] R. Mathon, *A note on the graph isomorphism counting problem*, Information Processing Letters, **8**:131-132, 1979.

[8] S. Fortin, *The graph isomorphism problem*, Technical Report TR 96-20, Dept. of Computer Science, University of Alberta, Canada, 1996.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, New York, 1990.

[10] B. D. McKay, *Practical graph isomorphism*, Congressus Numerantium, **30**: 45-87, 1981.

[11] G. Tinhofer and M. Klin, *Algebraic combinatorics in mathematical chemistry. Methods and algorithms. III. Graph Invariants and Stabilization Methods (Preliminary Version)*, Technical Report, TUM-M9902, Technische Universitat Munchen, 1999.

[12] L. Babel, I. V. Chuvaeva, and M. Klin, *Algebraic combinatorics in mathematical chemistry II. Program implementation of theWeisfeiler-Leman algorithm*, Preprint TUM-M9701, TU-Munchen, 1997.

[13] B. D. MacArthur, R. J. Sánchez-García and J. W. Anderson, *Symmetry in Complex Networks*, Discrete Applied Mathematics, **156**(18): 3525-3531, 2008.

[14] P.Erdös and A.Rényi, *On random graphs*, Publicationes Mathematicae:290-297, 1959.

[15] P. Erdos and A. Renyi, *On the evolution of random graphs*, Publ. Math. Inst. Hung. Acad. Sci, Vol.5:17-61, 1960.

[16] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, Science **286**(5439):509-512, 1999.

[17] Y. Xiao, M. Xiong, W. Wang, and H. Wang, *Emergence of symmetry in complex networks*, Physical Review E, **77**:066108, 2008.

[18] L. Backtrom, C. Dwork, and J. Kleinberg, *Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography*, In Proceedings of the 16th International Conference on World Wide Web(WWW'07):181-190, 2007.

[19] K. Liu and E. Terzi, *Towards identity anonymization on graphs.*, In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data(SIGMOD'08):93-106, 2008.

[20] B. Zhou and J. Pei, *Preserving privacy in social networks against neighborhood attacks*, In Proceedings of the 24th International Conference on Data Engineering(ICDE' 08):506-515, 2008.

[21] Y. Xiao, B. D.MacArthur, H. Wang, M. Xiong, and W. Wang, *Network quotients: Structural skeletons of complex systems*, Physical Review E, **78**:046102, 2008.

[22] M. Hay, G. Miklau, D. Jensen, D.Towsley and P. Weis, *Resisting structural re-identification in anonymized social networks*, PVLDB, Vol.**1**(1):102-114, 2008.

[23] M. E. J. Newman, *The structure and function of complex networks*, SIAM Rev., **45**(2):167-256, 2003.

[24] R. Albert, H. Jeong and A. -L. Barabási, *Error and attack tolerance of complex networks*, Nature, **406**(378), 2000.

[25] L. Singh and J. Zhan, *Measuring topological anonymity in social networks*, Intl. Conf. on Granular Computing, 2007.

[26] D. -W. Wang, C. -J. Liau and T. -S. Hsu, *Privacy protection in social network data disclosure based on granular computing*, International Conference on Fuzzy System, 2006.

[27] M. Hay, G. Miklau, D. Jensen, P. Weis and S. Srivastava, *Anonymizing social networks*, Technical report 07-19, UMass Amherst, 2007.

[28] X. Ying and X. Wu, *Randomizing social networks: a spectrum preserving approach*, SIAM Conf. on Data Mining, 2007.

[29] P. Samarati and L. Sweeney, *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*, Technical report, SRI International, 1998.

[30] P. Samarati, *Protecting respondent's privacy in microdata release*, IEEE Transactions on Knowledge and Data Engineering, 2001.

[31] L. Sweeney, *k-anonymity: a model for protecting privacy*, Journ. of Uncertainty, Fuzziness, and KB Systems, 2002.

[32] L. Zou, L. Chen, and M. Tamer Özsu, *k-Automorphism: A General Framework For Privacy Preserving Network Publication*, PVLDB, Vol.**2**(1):946-957, 2009.

[33] S. Boccalettia, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Complex networks: Structure and dynamics*, Physics Reports, **424**, 2006.

[34] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach*, Cambridge University Press, New York, NY, USA, 2004.

[35] J. Lauri and R. Scapellato, *Topics in graph automorphisms and reconstruction*, London Mathematical Society Student Texts, **54**, Cambridge University Press, Cambridge, 2003.

[36] Y. Xiao, W. Wu, J. Pei, W. Wang, and Z. He, *Efficiently indexing shortest paths by exploiting symmetry in graphs*, In Proceedings of the 12th International Conference on Extending Database Technology (EDBT'09): 493-504, 2009.

[37] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik, 1959.

[38] R. Bellman, *On a routing problem*, Quarterly of Applied Mathematics, 16(1): 87 – 90, 1958.

[39] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[40] R. Agrawal and H. V. Jagadish, *Algorithms for searching massive graphs*, IEEE Trans. on Knowl. and Data Eng., **6**(2), 1994.

[41] E. P. F. Chan and N. Zhang, *Finding shortest paths in large network systems*, In Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems(GIS'01), 2001.

[42] H. Samet, J. Sankaranarayanan, and H. Alborzi, *Scalable network distance browsing in spatial databases*, In Proceedings of the 28th ACM SIGMOD International Conference on Management of Data(SIGMOD'08):43-54, 2008.

[43] R. Agrawal and H. V. Jagadish, *Direct algorithms for computing the transitive closure of database relations*, In Proceedings of the 13th International Conference on Very Large Data Bases(VLDB'87):255-266, 1987.

[44] H. Lu, *New strategies for computing the transitive closure of a database relation*, In Proceedings of the 13th International Conference on Very Large Data Bases(VLDB'87):267-274, 1987.

[45] L. Chen, A. Gupta, and M. E. Kurul, *Stack-based algorithms for pattern matching on dags*, In Proceedings of the 31st international conference on Very Large Data Bases(VLDB'05), 2005.

[46] R. Schenkel, A. Theobald, and G. Weikum, *Efficient creation and incremental maintenance of the hopi index for complex xml document collections*, In Proceedings of the 21st International Conference on Data Engineering(ICDE'05), 2005.

[47] H. Wang, H. He, J. Yang, P. S. Yu, and J. X. Yu, *Dual labeling: Answering graph reachability queries in constant time*, In Proceedings of the 22nd International Conference on Data Engineering(ICDE'06), 2006.

[48] S. Trißl and U. Leser, *Fast and practical indexing and querying of very large graphs*, In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data(SIGMOD'07), 2007.

[49] H. He, H. Wang, J. Yang, and P. S. Yu, *Compact reachability labeling for graph-structured data*, In Proceedings of the 14th ACM International Conference on Information and Knowledge Management(CIKM'05), 2005.

[50] D. Gibson, J. Kleinberg , and P. Raghavan, *Inferring web communities from link topology*, Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, 225-234, 1998.

[51] G. W. Flake, S. Lawrence, C. L. Giles and F. M. Coetzee, *Self-organization and identification of web communities*, IEEE Computer, 35(3): 66-71, 2002.

[52] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, Bell System Technical Journal, 49(2): 291-307, 1970.

[53] M. Fiedler, *Algebraic connectivity of graphs*, CzechMath J, 23(98): 298-305, 1973.

[54] A. Pothen, H. Simon, and K-P Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J Matrix Anal App, 11(3): 430-452, 1990.

[55] F. Wu and B. A. Huberman, *Finding communities in linear time: a physics approach*, Eur Phys J B, 38(2): 331 - 338, 2004.

[56] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, *Detecting communities in large networks*, Physica A, 352(2-4): 669-676, 2005.

[57] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Phys. Rev. E, 69(2):026113, 2004.

[58] M. E. J. Newman, *Fast algorithm for detecting community structure in networks*, Phys. Rev. E, 69(6):066133, 2004.

[59] A. Clauset, M. E. J. Newman, and C. Moore, *Finding community structure in very large networks*, Phys. Rev. E, 70(6):066111, 2004.

[60] S. Fortunato and M. Barthelemy, *Resolution limit in community detection*, Proc Natl Acad Sci, vol. 104, no. 1, pp.36-41, 2007.

[61] M. Granovetter, *The Strength of Weak Ties*, Am J Sociol 78:1360-1380, 1973.

[62] *http://cs.unm.edu/ aaron/research/fastmodularity.htm.*

[63] J. -P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A. -L. Barabasi, *Structure and tie strengths in mobile communication networks*, Proc Natl Acad Sci, vol. 104, no. 18, pp. 7332-7336, 2007.

[64] J. M. Kumpula, J. -P. Onnela, J. Saramaki, K. Kaski, and J. Kertesz, *Emergence of communities in weighted Networks*, Phys. Rev. Lett, 99:228701, 2007.

[65] Freeman, L.C. *A set of measures of centrality based on betweenness.* Sociometry, **40**:35 – 43, 1977.

[66] L. Page, S. Brin, R. Motwani and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report 1999-66, Stanford InfoLab, 1999.

[67] J. Kleinberg, *Authoritative Sources in a Hyperlinked Environment*, Journal of the ACM **46**(5): 604 – 632, 1999.

[68] R. Albert and A.-L. Barabási, *Statistical mechanics of complex networks*, Rev. Modern Phys., **74**(1):47-97, 2002.

[69] B. Bollobás, *Random graphs*, Second Edition, Cambridge Studies in Advanced Mathematics, Vol.73, Cambridge University Press, Cambridge, 2001.

[70] K. Mainzer, *Symmetry and Complexity. The Spirit and Beauty of Nonlinear Science*, World Scientific Publ., Singapore, 2005.

[71] M. Quack, *Molecular spectra, reaction dynamics, symmetries and life*, Chimia, **57**(4):147-160, 2003.

[72] G. Felder, J. Garcia-Bellido, P. B. Greene, L. Kofman, A. Linde, and I. Tkachev, *Dynamics of Symmetry Breaking and Tachyonic Preheating*, Physical Review Letters **87**:011601, 2001.

[73] F. Heylighen and D. Aerts, *The Growth of Structural and Functional Complexity during Evolution*, in The Evolution of Complexity, Kluwer Academic Publishers, 1996.

[74] A. Inokuchi, T. Washio, and H. Motoda, *An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data*. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery(PKDD'00), 2000.

[75] X. Yan and J. Han, *gSpan: Graph-Based Substructure Pattern Mining*, In Proceedings of the 2002 IEEE International Conference on Data Mining(ICDM'02), 2002.

[76] D. W. Williams, J. Huan and W. Wang, *Graph Database Indexing Using Structured Graph Decomposition*, In Proceedings of IEEE 23rd International Conference on Data Engineering(ICDE'07):976-985, 2007.

[77] H. Jiang, H. Wang, P. S. Yu, S. Zhou, *Gstring: a novel approach for efficient search in Graph Databases*, In Proceedings of IEEE 23rd International Conference on Data Engineering(ICDE'07):566-575, 2007.

[78] D. Shasha, J. T-L Wang, and R. Giugno, *Algorithmics and applications of tree and graph searching*, In Proceedings of the 21st ACM Symposium on Principles of Database Systems(PODS'02):39-52, 2002.

[79] X. Yan, P. Yu, and J. Han, *Graph Index: A Frequent Substructure-based Approach*, In Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data(SIGMOD'04):335-346, 2004.

发表文章目录

[1] **Wentao Wu**, Yanghua Xiao, Wei Wang, Zhenying He, Zhihui Wang, *k-symmetry model for identity anonymization in social networks*, In Proceedings of the 13th International Conference on Extending Database Technology (**EDBT'10**): 111-122, 2010.

[2] Jidong Chen, Hang Guo, **Wentao Wu**, Chunxin Xie, *Search your memory! - an associative memory based desktop search system*, In Proceedings of the 35th SIGMOD International Conference on Management of Data (**SIGMOD'09**): 1099-1102, 2009.

[3] Yanghua Xiao, **Wentao Wu**, Jian Pei, Wei Wang, Zhenying He, *Efficiently indexing shortest paths by exploiting symmetry in graphs*, In Proceedings of the 12th International Conference on Extending Database Technology (**EDBT'09**): 493-504, 2009.

[4] Jidong Chen, Hang Guo, **Wentao Wu**, Wei Wang, *iMecho: an associative memory based desktop search system*, In Proceedings of the 18th ACM Conference on Information and Knowledge Management (**CIKM'09**): 731-740, 2009.

[5] Hang Guo, Jidong Chen, **Wentao Wu**, Wei Wang, *Personalization as a service: the architecture and a case study*, In Proceedings of the 1st International CIKM Workshop on Cloud Data Management (**CloudDb'09**): 1-8, 2009.

[6] **吴文涛**，肖仰华，何震瀛，汪卫，余韬，基于权重信息挖掘社会网络中的隐含社团，**计算机研究与发展**，第46卷(增刊)：540-546，2009。

[7] Yanghua Xiao, Hua Dong, **Wentao Wu**, Momiao Xiong, Wei Wang, Baile Shi, *Structure-based graph distance measures of high degree of precision*, **Pattern Recognition**, Vol. 41(12): 3547-3561, 2008.

[8] Yanghua Xiao, **Wentao Wu**, Wei Wang, Zhenying He, *Efficient algorithms for node disjoint subgraph homeomorphism determination*, In Proceedings of 13th International Conference on Database Systems for Advanced Applications (**DASFAA'08**): 452-460, 2008.

[9] Yanghua Xiao, **Wentao Wu**, Hui Wang, Momiao Xiong, Wei Wang, *Symmetry-based structure entropy of complex networks*, **Physica A**, Vol. 387(11): 2611-2619, 2008.

[10] Yanghua Xiao, Wei Wang, **Wentao Wu**, *Mining conserved topological structures from large protein-protein interaction networks*, In Proceedings of 18th IEICE data engineering workshop / 5th DBSJ annual meeting (**DEWS'07**), 2007.

致　谢

值此论文完成之际，谨在此向多年来给予我关心和帮助的老师、同学、朋友和家人表示衷心的感谢。

首先，感谢我的导师汪卫教授在过去的几年中对我研究工作的辛勤指导和对我学习生活的无微不至的关心。我从大三起进入汪老师的课题组从事科研工作。汪老师不仅学术造诣高，而且为人师表，言传身教，一步一步地引领我从对科学研究一无所知到今天能够基本独立地从事科学研究工作。

其次，感谢课题组的肖仰华老师和何震瀛老师，感谢已经毕业的林琛师姐，感谢解春欣同学，以及学弟学妹们余韬，陈垚亮，丁白露，朱翔，鲁轶奇等。这几年的硕士生活绝大部分时间是在与他们的共同讨论、争论中度过的。

感谢EMC中国研究中心的毛文波博士、陈继东博士、郭杭博士，以及微软亚洲研究院的王海勋博士，硕士期间与他们在研究工作上的交流与合作极大地开阔了我的视野，提高了独立从事科研工作的能力。

感谢上海国际数据库中心的所有老师和同学们，过去几年在这里的研究生活是充实而愉快的。感谢07级硕士班的两位辅导员施兰珍老师和赵进老师，感谢07 级硕士班所有的同学们，虽然3年的硕士生活转瞬即逝，但我会永远铭记这段难忘的岁月。

感谢复旦大学为我们创造了优美的校园环境，建造了宽敞明亮的图书馆，营造了开放自由的学术氛围，创造了跨学科的交流平台。在复旦大学从本科到硕士的7年是我人生中最为重要的阶段，复旦的学习经历给予了我足够的自信和能力去迎接未来的机遇和挑战。

最后感谢我的父母、家人和朋友，是他们一直以来的支持和鼓励使得我可以安心从事研究工作，并取得今天的成绩。

论文独创性声明

本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。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名：吴文涛　　日期：2010.6.4

论文使用授权声明

本人完全了解复旦大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名：吴文涛　导师签名：汪卫　日期：2010.6.4