学 号

密 级

哈尔滨工程大学学士学位论文

人员社会关系图谱可视化加速算法设计与实现

院（系）名　称：计算机科学与技术学院

专　业　名　称：计算机科学与技术

学　生　姓　名：

指　导　教　师：

哈尔滨工程大学

2018年6月

**人员社会关系图谱可视化加速算法设计与实现 哈尔滨工程大学**

学 号

密 级

人员社会关系图谱可视化加速算法设计与实现

Visualization Accelerated Algorithm Design and Implementation of Human Society Relationship Graph

**学生姓名：**

**所在学院：**计算机科学与技术学院

**所在专业：**计算机科学与技术

**指导教师：**

**职称：**

**所在单位：**哈尔滨工程大学

**论文提交日期：**2018年6月

**论文答辩日期：**2018年6月

**学位授予单位：**哈尔滨工程大学

**摘 要**

人员社会关系图谱是根据社会中人与人之间的联系，利用节点和连线表示人与人之间关系的工具。最初，社会关系图谱主要由亲戚、朋友以及它们之间的关系组成，后来随着互联网的普及，人与人之间的联系通过论坛、微博等各种互联网社交工具的普遍使用急剧增加，人员社会关系图谱的节点、节点之间的关系数量越来越多。当图谱中节点、节点之间的关系数量达到千量级以上时，图谱可视化存在的节点和边重叠、边大量交叉、布局缓慢、混乱等布局问题，以及展示耗时、效率低等问题会更加突出，现有的图谱可视化布局策略存在布局消耗时间长、布局不够合理以及可视化速度慢等不足。

本课题旨在设计并实现人员社会关系图谱可视化加速算法，解决千量级以上的人员社会关系图谱布局加速和快速展示问题。人员社会关系图谱布局加速是在高维嵌入的基础上，通过社团发现算法对人员社会关系图谱进行社区划分，然后使用两级布局算法对人员社会关系图谱进行加速布局，从而实现将千量级的图谱可视化时节点和边重叠少、边交叉少、布局和展示快速的目的；人员社会关系图谱展示加速是基于多线程的加速展示策略，从而将千量级的图谱快速的展示出来。

本课题研究工作结合高维嵌入和社区划分算法，为解决人员社会关系图谱的可视化加速算法提供了新的技术手段。大量的实验结果证明了本文所提供方法的实用性和有效性。

**关键字：**人员社会关系图谱；可视化加速；高维嵌入；布局加速；聚类

**ABSTRACT**

The map of people's social relations is a tool that uses nodes and connections to represent people-to-people relationships based on the connections between people in society. Initially, the map of social relations was mainly composed of relatives, friends, and the relationships between them. Later, with the popularity of the Internet, people-to-people contacts increased dramatically through the widespread use of various social networking tools such as forums and Weibo. There are more and more nodes and nodes in the social relationship map. When the number of nodes and nodes in the map reaches thousands or more, the nodes and edges that exist in the map visualization are overlapped, and the edges are overlapped. Slow, chaotic and other layout problems, as well as time-consuming and inefficient display problems will become more prominent. Existing map visualization layout strategies still have problems such as long layout time, unreasonable layout, and slow visualization.

The purpose of this project is to design found and implement a visual acceleration algorithm for the mapping of human social relationships, and to solve the problem of accelerated and rapid display of the layout of human social relationship maps of thousands or more levels. Based on high-dimensional embedding, the layout of human social relationship maps is based on the high-dimensional embedding. The Society found algorithm is used to divide the community maps of people, thereby achieving thousands of maps. Visualization has the advantages of less overlap of nodes and edges, less edge crossing, rapid layout and display; and the acceleration of human social relationship map display is based on multi-threaded accelerated display strategy, so that thousands of maps can be quickly displayed.

The research work of this project combines high-dimensional embedding and community partitioning algorithms, and provides a new technical means for solving the visual acceleration algorithm of human social relationship maps. A large number of experimental results prove the effectiveness of the method provided in this paper.

**Key words:** Human Society Relationship Graph;Visualization Accelerated; High-dimensional embedding ;layout algorithm; clustering

目 录

[第1章 绪论 1](#_Toc517267133)

[1.1课题背景 1](#_Toc517267134)

[1.2课题目的和意义 1](#_Toc517267135)

[1.3 国内外研究现状 2](#_Toc517267136)

[1.4 主要研究内容 4](#_Toc517267137)

[1.5 论文结构安排 4](#_Toc517267138)

[第2章 人员社会关系图谱可视化加速算法设计 6](#_Toc517267139)

[2.1 可视化加速算法需求分析 6](#_Toc517267140)

[2.2 可视化加速算法总体设计 6](#_Toc517267141)

[2.3基于KK图谱模型的人员社会关系图谱布局加速算法设计 8](#_Toc517267142)

[2.4 基于多线程的人员社会关系图谱展示加速算法设计 9](#_Toc517267143)

[2.5 本章小结 10](#_Toc517267144)

[第3章 基于KK图谱模型的人员社会关系图谱布局加速算法实现 11](#_Toc517267145)

[3.1 总体实现 11](#_Toc517267146)

[3.2 人员社会关系图谱节点和关系解析 12](#_Toc517267147)

[3.3 人员社会关系图谱节点分层 13](#_Toc517267148)

[3.3.1 基于社团发现算法的图谱聚类 13](#_Toc517267149)

[3.3.2 构造抽象图谱 15](#_Toc517267150)

[3.4人员社会关系图谱节点和关系布局 18](#_Toc517267151)

[3.4.1 KK图谱能量模型 18](#_Toc517267152)

[3.4.2 基于KK图谱算法抽象图谱布局 20](#_Toc517267153)

[3.4.3 基于KK图谱算法子图谱布局 21](#_Toc517267154)

[3.5 本章小结 23](#_Toc517267155)

[第4章 基于多线程的人员社会关系图谱展示加速算法实现 24](#_Toc517267156)

[4.1 总体实现 24](#_Toc517267157)

[4.2展示策略 25](#_Toc517267158)

[4.3 XML解析 26](#_Toc517267159)

[4.4 节点展示和关系展示 28](#_Toc517267160)

[4.5 本章小结 29](#_Toc517267161)

[第5章 实验结果与分析 31](#_Toc517267162)

[5.1 总体设计 31](#_Toc517267163)

[5.2 基于KK图谱算法的人员社会关系图谱布局加速算法实验 31](#_Toc517267164)

[5.2.1 实验设计 31](#_Toc517267165)

[5.2.2 实验结果 32](#_Toc517267166)

[5.2.3 实验分析 34](#_Toc517267167)

[5.3 基于多线程的人员社会关系图谱展示加速算法实验 35](#_Toc517267168)

[5.3.1 实验设计 35](#_Toc517267169)

[5.3.2 实验结果 35](#_Toc517267170)

[5.3.3 实验分析 36](#_Toc517267171)

[5.4 人员社会关系图谱可视化加速算法应用 36](#_Toc517267172)

[5.4.1 用户广告精准推荐 36](#_Toc517267173)

[5.4.2 社交娱乐 37](#_Toc517267174)

[5.5 本章小结 38](#_Toc517267175)

[结论 40](#_Toc517267176)

[参考文献 41](#_Toc517267177)

[攻读学士学位期间发表的论文和取得的科研成果 43](#_Toc517267178)

[致谢 44](#_Toc517267179)

# 第1章 绪论

## 1.1课题背景

随着人与人之间的联系日益紧密，人与人之间的交流日益频繁，人们之间的关系也变得越来越复杂，如果用节点表示人，用节点之间的边表示人与人之间的关系，那么这样的图谱就是人员社会关系图谱，则人员社会关系图谱中节点之间的关系越来越复杂，全球人口数量庞大，节点的数量也非常庞大。现有的算法又不适合处理大数据量的图谱，所以在人员社会关系图谱的可视化面临布局缓慢、不理想，展示耗时等一系列问题。在实际生活中，人们通过清晰可见的图谱结构，展示社会人员之间的关系，从复杂人员社会关系图谱中发掘出隐藏的重要信息。研究图谱可视化的方法一直被人们所重视和研究，但由于图谱中节点的数量越来越多，结构也越来越复杂，快速找出人员社会图谱中的重要信息和将人员社会关系图谱快速展示的问题变得越来越尖锐。在国内外，尽管有许多人提出的许多行之有效的方法去解决节点布局和快速展示的问题，随着时代的发展，慢慢的这些方法已经很难将大量节点的图谱进行快速、合理的布局，几乎无法从这样的图谱中发现任何有用的信息，并且国内对大量节点图谱的处理技术更是落后于一些诸如英美日德等发达国家。

## 1.2课题目的和意义

由于随着互联网社交的迅速繁荣，人与人之间的联系日益紧密，人与人之间的关系也日趋复杂，接入互联网的人口数量也越来越庞大，因此人员社会关系图谱节点之间的联系增多，节点之间的关系更加复杂，节点数量越来越庞大，这导致人员社会关系图谱在布局的过程更加容易出现节点重叠，边大量交叉、重复等布局混乱，以及布局和展示耗时等不合理现象。从布局缓慢、节点重叠、边大量交叉和展示耗时的人员社会关系图谱中快速挖掘人员社会图谱中的重要信息，变得越来越困难。在国内外，尽管有许多人提出的许多行之有效的方法去解决节点布局和快速展示的问题，但是随着人员社会关系图谱的复杂度的大幅度提高和数量的快速增大，这些方法已经慢慢的很难处理这些大量节点的图谱。所以当前急需一套方法和技术去适应在互联网社交蓬勃发展下，人员社会关系图谱急剧增大所带来布局缓慢、混乱以及图谱展示耗时的问题。

本课题旨在提出一个人员社会关系图谱可视化加速算法，解决千量级人员社会关系图谱的布局加速和快速展示问题。在千量级的人员社会关系图谱中，由于存在大量节点重叠、布局耗时和展示耗时等问题，所以需要对图谱进行布局以及布局加速和展示加速。

在实际生活中，人们通过清晰可见的图谱结构，展示社会人员之间的关系，探索隐藏在复杂的人员社会关系图谱中重要信息，发掘有用人员社会关系信息，研究网络图形化的方法也一直被人们所重视和研究。首先，从复杂的人员社会关系图谱中找出关键节点具有重要实际意义，当检索一个网站时，在没有拼写全部网址的情况下，系统就已经推荐出几个重要网址，大多数情况下，其中一个就是所要拼写的网址；其次，购物软件往往会根据用户最近的关注点去推荐一些合适的商品；最后，社交网址经常会推荐一些可能认识的好友，这一切都是基于对人员关系图谱的布局所做出的推荐

## 1.3 国内外研究现状

在图谱的可视化研究方面，主要从图谱布局、可视化加速和聚类等方面进行研究，下面分别对这三个方面进行论述。

（1）图谱布局

第一，在图谱的可视化方面，第一个里程碑式的方法是Eades P[1]在二十世纪八十年代第一次提出的力导向模型，后来Kama-da T和Kawai S[2]改进了Eades P提出的力导向模型，提出了KK图谱[3]模型，并且在KK图谱模型中加入了理想距离的概念，理想距离是节点之间的边数与图谱半径的比值。并且通过理想距离，KK图谱模型定义了系统的总能量的计算公式，由此得到最优布局的过程就是不断减少系统总能量的过程，即当所有的节点能量和最少时，图谱就达到平衡状态，也是所求的最优布局状态。KK图谱模型最大的贡献是将图谱的最优布局问题转换为减小弹簧能量系统总能量的问题，实现了数学转化，提供了一条新的切实可行的方法以及思路，通过KK图谱模型对图谱布局可以达到一个比较合理的布局，但是KK图谱模型有个最大的缺陷，它虽然实现了问题的数学转化，但是KK图谱模型也存在缺陷，首先，KK图谱模型只能解决局部的布局问题，并不能解决图谱的全局布局；其次，KK图谱模型计算复杂度较高；最后，KK图谱模型不能很好的观察图谱的内在结构。第二，吴鹏在2011年提出了子群分析算法[4]，它是基于力导引模型和子群划分的用于图谱可视化的一种算法。子群分析算法最大的贡献是给图谱中的节点赋予现实意义，首先通过该算法计算图谱中所有节点的关键身份值和特殊属性值，然后根据计算得到的关键身份值和特殊属性值对源图谱进行子群划分，最后基于力导引模型对子群划分之后得到的图谱可视化布局。子群分析算法的主要贡献是通过给图谱中每个节点加入实际意义，计算节点关键身份值和特殊属性值来分析节点之间的相似程度，使得图谱的最终可视化布局具有明显的子群特性，具有子群特性的可视化图谱更容易清楚的挖掘图谱中隐藏的信息，观察图谱中子群图谱的内在结构，并且图谱的显示也更加合理和具有条理性。子群分析算法也有明显的缺陷，子群分析算法主要关注图谱中内在结构的清晰可见，不太注重算法的时间复杂度，分析千量级节点的图谱时需要消耗过多的时间。第三，圆形布局是将图谱布局在一个圆环上展示，图谱布局对称、清晰，具有很高的美学特性，但是其计算复杂度相对较高，并且也无法观察图谱的内在结构。

（2）可视化加速

Fruchterman T M J和Reigold E M[5] 提出了FR模型，该模型是在KK图谱模型的基础上对布局算法做了两个方面的重大优化。首先是在KK图谱模型中加入了电子斥力，电子斥力反应的是节点之间的斥力与距离的关系，节点之间的距离越短斥力越大；其次是加快了算法的收敛速度。为了提高算法的运行效率，引入了“冷却函数”的概念，冷却函数与退火过程相似，慢慢减少每次减少的最大能量值，即节点的位置变化变得缓慢，从而加速了算法的执行效率。该模型不但能达到合理布局社交图谱的目的，相对其他算法，该模型的时间复杂度也相对低很多，但是FR模型的计算复杂度仍旧比较高；并且FR模型不能很好的观察图谱的内在结构。后来人们又提出了LL模型[6]、Koren Y[7]提出了双压力模型（Binary Stress Model）和MDS模型算法等，但是都存在FR模型同样的缺陷。

（3）聚类

聚类[7]就是将各个相似的对象组织成多个集合的过程。目前已经存在许多优秀的聚类算法，K-Means[9]算法是一种基于划分的聚类算法，其的思想就是找出图谱中的m个中心，将其他的节点划分到这m个中心表示的集合里边，并且使得每个集合中数据集中的点与该集合的中心的距离平方和最小，但是该算法无法处理单个点的图谱。FCM[9]是一种基于模糊的聚类算法，这两个算法可以将图谱聚类，但是时间复杂度还是偏高。社团发现[11]算法是基于边介数[12]的聚类算法，该算法的思想是就是按照边介数的大小不断地去掉图谱中的边，核心是分裂思想，该算法可以快速的将图谱聚类。

综上所述，力导引算法、圆形布局和矩形布局可视化具有较高的美学特性，但是计算复杂度高，不能观察子群内部结构；子群分析算法可视化可以清楚地观察、分析图谱中子群的内在结构，但是计算复杂度高；FR模型具有较低的时间复杂度，但是不能观察子群的内在结构。在人员社会关系图谱日趋复杂和庞大的今天，这些算法显得有些力不从心，所以可以通过聚类思想加速图谱可视化。

## 1.4 主要研究内容

本课题旨在完成人员社会关系图谱可视化加速算法的设计并实现，解决千量级以上的人员社会关系图谱布局加速和快速展示问题，可视化加速算法是一种将图谱加速展示的算法，主要用于加速展示大量节点和边的图谱。本课题主要包括图谱布局加速算法、图谱展示加速算法和实验与应用。

图谱布局加速算法是一种将图谱合理布局的算法，它主要实现将千量级的图谱以简洁、合理、清晰的形式快速的布局。图谱布局加速算法是基于KK图谱模型和分层思想的一种力导引布局算法，首先利用图谱的的社区性质将图谱划分为若干个小图谱；然后通过高维嵌入使用三维的KK图谱模型对每个小图谱和整个图谱进行布局。

图谱展示加速算法基于多线程的思想，研究一种加速展示千量级图谱的展示加速算法，图谱展示加速算法主要实现将千量级的图谱快速的展示出来的目标，它主要通过调用多线程来加速节点和关系的展示，从而解决千量级的图谱快速展示的问题。

实验与应用是对图谱布局加速算法图谱布局和图谱展示加速算法图谱展示有效性的验证。分别使用不同区间的数量的数据集来分别验证图谱布局加速算法和图谱展示加速算法，通过应用体现其实际价值。

## 1.5 论文结构安排

本篇论文主要分为5章，从人员社会关系图谱可视化加速算法的背景、研究的意义以及研究现状，到图谱布局加速算法和图谱展示加速算法的设计与实现，以及人员社会关系图谱可视化加速算法验证与应用等方面进行论述。

第1章主要论述本课题的背景，研究的意义目的、国内外的对本课题的研究现状以及本课题的主要内容方面做大体的说明。

第2章主要论述本课题研究的内容、方法和算法的总体设计，并且对算法做相应的需求分析，简要介绍本课题的研究思路。

第3章是重点论述基于KK图谱算法的人员社会关系图谱布局加速算法设计与实现，先介绍KK图谱模型，然后论述将图谱分层，并且使用KK图谱算法实现图谱的加速布局。

第4章重点论述基于多线程的人员社会关系图谱展示加速算法设计与实现，先论述选择合适的展示策略，然后将节点和边展示出来。

第5章是实验与应用，验证图谱布局加速算法和图谱展示加速算法，讨论可视化加速算法在实际生活中的应用。

最后是结论、致谢和参考文献。

# 第2章 人员社会关系图谱可视化加速算法设计

## 2.1 可视化加速算法需求分析

将千量级图谱可视化的过程中，存在节点和边重叠、边大量交叉、布局缓慢、混乱等布局问题，以及展示耗时、效率低等展示问题。首先，如果仅仅依赖人的计算能力去实现对千量级图谱的布局计算和规划是一件很困难的事情，并且布局效率低、质量差；其次，千量级图谱展示时需要消耗大量的时间，人们对图谱展示的实时性、有效性以及准确性要求越来越高。在已有的研究中，力导引算法可视化虽然具有较高的美学特性，子群分析算法虽然可以清楚地观察、分析图谱中子群的内在结构，但是它们都有具有较高的时间复杂度，在人员社会关系图谱日趋复杂和庞大的今天，这些算法显得有些力不从心。

本课题是基于KK图谱模型进行的研究，克服KK图谱模型计算复杂度高的缺点，优化了KK图谱模型易于陷入局部优化的缺陷，提出了基于KK图谱算法的二级KK图谱算法的力导引布局策略来解决千量级节点的人员社会关系图谱的布局加速问题，基于多线程的策略来解决千量级节点的人员社会关系图谱的展示加速问题。

## 2.2 可视化加速算法总体设计

本课题是对人员社会关系图谱的可视化加速进行研究，对于千量级人员社会关系图谱可视化，需要先设计图元的布局，然后基于这个布局进行展示加速算法设计。因此，本课题可视化加速算法主要包含两个子算法：图谱布局加速算法和图谱展示加速算法。

可视化加速算法总体设计包括图谱布局加速算法、图谱展示加速算法和图谱的优化与验证，具体人员社会关系图谱可视化加速算法功能结构如图2.1 所示。



图 2.1 人员社会关系图谱可视化加速算法功能结构图

（1）图谱布局加速算法

图谱布局是将图谱中的节点、节点之间的边，根据节点之间的关系改变节点、节点之间边的位置，从而达到图谱展示合理的目标。由于图谱节点和边的信息存储在XML文件中，所以在进行图谱布局时，首先需要节点解析和关系解析；由于分层可以可以解决千量级图谱可视化的过程中布局缓慢、耗时的问题，，所以接下来将节点进行分层；最后进行节点布局和关系布局。因此，该算法主要包括：节点和关系解析、节点分层、节点和关系布局。

* 节点和关系解析：节点解析和关系解析就是从存储图谱的XML文件中解析出节点的横坐标、纵坐标以及节点的源节点Id和目标节点Id。
* 节点分层：节点分层就是将大量的节点通过基于边介数的社团发现算法进行聚类，得到具有明显社区性质的子图谱，将子图谱通过构造抽象图谱[13]算法构造成抽象图谱。
* 节点和关系布局：节点布局和关系布局就是使用KK力导引布局[14]算法计算抽象图谱节点坐标以及各个子图谱内部节点坐标，相应的改变节点之间边的坐标，最终实现了节点、节点之间边的合理布局。总之，该算法是基于KK图谱模型和分层思想的一种二级的力导引布局算法，致力于解决千量级节点图谱布局加速问题。

（2）图谱展示加速算法

图谱展示是将图谱中节点、节点之间边按照它们各自的坐标，将其图形化的过程。首先，由于图谱节点坐标和关系的源节点和目标节点都存储在XML文件中，所以需要选择合适的策略来解析XML文件；然后，将节点、节点之间的边展示出来，因此该算法主要包含三大部分：节点和节点之间关系的展示策略、节点和关系展示。

* 节点和关系的展示策略：节点和关系的展示策略就是根据节点、节点之间关系的数目来确定具体多线程数。
* 节点和关系展示：节点展示和关系展示就是将图谱中节点、节点之间的关系按照一定的展示策略将其可视化。该算法使用多线程来加速节点和关系的展示，然后使用多线程将节点和节点之间关系展示出来。总之，该算法基于多线程的思想，研究一种加速展示千量级图谱的展示加速算法。

（3）图谱的验证与应用部分

图谱的验证与应用部分主要检验可视化加速算法的有效性和加速效果，分析实验结果，论述可视化加速算法的实际应用，图谱的验证与应用主要包括两部分：可视化加速算法的验证和可视化加速算法的应用。

* 可视化加速算法验证：首先通过构造合适的数据集来验证算法有效性和加速效果；第二个数据集验证图谱布局加速算法的加速效果，所以该数据集可以不具实际意义，可以通过程序随机构造，但是可视化加速算法是千量级节点的加速布局和快速展示，所以数据集必须达到千量级；然后根据实验结果在布局效果和时间复杂度等方面对算法的有效性和优劣进行分析，基于不同节点数目的图谱所做的可视化加速实验结果，从中分析出规律或者总结出实验结论。
* 可视化加速算法的应用：该部分主要从广告精准推荐、人物搜索和社交娱乐三个方面说明其实际应用。

## 2.3基于KK图谱模型的人员社会关系图谱布局加速算法设计

在“人员社会关系图谱”项目中，图谱节点坐标和关系的源节点和目标节点都存储在XML文件中，所以需要从XML文件中解析出节点和边的信息。节点解析是从XML图谱文件中解析出社交图谱中所有节点的横坐标和纵坐标，节点XML标签如下。

<people>

<id>5</id>

<name>老师</name>

<photo>img/John.png</photo>

<height>70</height>

<width>140</width>

<x>651.00</x>

<y>204.00</y>

</people>

在该节点XML标签中，“id”是该节点的编号，“height”是节点矩形的高度，“width”是节点矩形的宽度，“x”和“y”分别是该节点的横坐标和纵坐标。

关系解析是从XML图谱文件中解析出社交图谱中所有关系的源节点和目标节点，关系的标签如下。

<connect>

<sourceid>10</sourceid>

<targetid>6</targetid>

</connect>

在该边XML标签中，“sourceid”是该边的源节点，“targetid”是该边的目标节点。

在解析节点时，先从XML文件中获取标签people，然后获取该标签下的子标签id、height、width、X和Y，最后获取子标签各自的内容。

在解析关系时，先从XML文件中获取标签connect，然后获取该标签下的子标签sourceid和targetid，最后获取子标签各自的内容。

节点分层可以解决千量级图谱可视化的过程中布局缓慢、耗时的问题，节点分层就是将千量级图谱的节点通过社团发现算法进行聚类，得到具有明显社区性质的子图谱，将子图谱通过构造抽象图谱算法构造成抽象图谱，此时将整个图谱划分为了两层图谱。

节点布局和关系布局就是改变节点的坐标，从而解决图谱可视化过程中遇到的节点重叠、大量边交叉的问题，KK图谱布局算法可以解决该问题。另外，KK图谱布局算法本身具有时间复杂度高和容易陷入局部最优的缺陷，但与分层思想结合之后不仅提高了算法的时间复杂度，还消弱了KK图谱模型容易导致局部最优的缺点，从而完美的解决了布局问题。在节点布局和关系布局中，首先需要计算图谱中所有节点的平铺面积使用KK力导引布局算法计算抽象图谱节点坐标以及各个子图谱内部节点坐标，相应的改变节点之间边的坐标，最终实现了节点、节点之间边的合理布局。

总之，该算法是基于KK图谱模型和分层思想的一种二级的力导引布局算法，致力于解决千量级节点图谱布局加速问题。

## 2.4 基于多线程的人员社会关系图谱展示加速算法设计

人员社会关系图谱展示是从存储图谱信息的XML文件中解析出节点的坐标以及节点之间边的源节点和目标节点，并且按照节点的坐标和边的目标节点和源节点将其图形化的过程。千量级图谱在展示时，存在耗时的问题，并且主要的时间消耗在解析XML文件和展示节点和节点之间的边上，多线程可以加速解析XML文件和加速展示节点和节点之间的边，为了最大限度的加速算法，需要合适的展示策略、节点展示和关系展示。

节点和关系的展示策略是基于多线程实现的，由于线程的管理需要消耗额为的CPU和内存资源，以及线程过多时会出现抖动现象，所以要根据节点、节点之间的关系的数目运行合适的线程数，让CPU最大化的使用，并且做的都是有效功。因此，多线程的数目以及节点和边的展示顺序需要根据具体的节点和关系的数量来制定一定的策略。对于千量级节点的图谱，首先运行若干个线程，让每个线程平均去解析一定数量的people标签和connect标签；比如，总共有1000个节点和1000个边，总共使用四个线程去解析XML文件，第一个线程去解析第0到第250的people标签和第0到第250的connect标签，第二个线程去解析第251到第500的people标签和251到第500的connect标签，以此类推，最后一个线程解析剩下的所有节点标签和边标签。

节点展示和关系展示就是根据节点坐标、边的源节点和目标节点将节点和边显示出来。

## 2.5 本章小结

本章论述可视化加速算法的总体设计，可视化加速算法的总体设计包括三大部分：图谱布局加速算法设计、图谱展示加速算法设计和实验与应用。图谱布局加速算法主要完成五部分内容：节点解析、关系解析、节点分层、节点布局和关系布局；图谱展示加速算法主要完成三部分内容：节点和关系的展示策略、节点展示和关系展示；实验与应用是对图谱布局加速算法和图谱展示加速算法进行实验检验，并且介绍可视化加速算法在实际生活中的应用。总之，研究的思路主要是人员社会关系图谱布局加速算法采用二级的KK图谱算法来进行优化，人员社会关系图谱展示加速算法采用多线程来优化算法，构造不同的数据集进行验证。

# 第3章 基于KK图谱模型的人员社会关系图谱布局加速算法实现

## 3.1 总体实现

基于KK图谱模型的人员社会关系图谱布局加速算法是将图谱快速合理的布局，其实现思路是：从XML文件中解析出节点坐标、关系源节点Id和目标节点Id，通过高维嵌入将二维节点坐标扩展到三维，使用基于社团发现算法划分子图谱，将子图谱构造为抽象图谱，使用三维KK图谱算法布局子图谱和抽象图谱，最终得到布局之后的三维节点坐标和关系源节点Id和目标节点Id，使用低维嵌入将三维节点坐标转化为二维坐标，输出二维节点左边和关系源节点Id和目标节点Id。该算法主要包括五大部分：节点解析、关系解析、节点分层、节点布局和关系布局，基于KK图谱模型人员社会关系图谱布局加速算法流程图如图3.1所示。



图 3.1基于KK图谱模型人员社会关系图谱布局加速算法流程图

第一，由于图谱的信息都存储在XML文件中，所以节点解析和关系解析就是从XML文件中解析出图谱的中节点的坐标和边的源节点和目标节点，并将其分别存储在Vector中。第二，节点分层就是先将整个图谱通过社团发现算法（Edge betweenness）划分子图谱，最终得到子图谱...；然后通过抽象图谱算法将聚类子图谱构造成抽象图谱；初始化抽象图谱的一些基本参数，抽象图谱的节点和边，抽象图谱之间的理想距离，抽象图谱中所有节点的聚集系数，抽象图谱的特征向量，抽象图谱中所有节点的PR值之和；第三，节点布局和关系布局就是使用新的理想距离优化KK图谱算法，然后对抽象图谱进行布局，基于KK图谱模型的子图谱布局是在子图谱中心点处调用KK图谱模型对子图谱布局。如果KK图谱模型时间复杂度为，那么在最优化的情况下分层的KK图谱模型的时间复杂度可以达到。

## 3.2 人员社会关系图谱节点和关系解析

人员社会关系图谱的信息都存储在XML文件中，节点解析就是从XML文件中解析出图谱中所有节点坐标的X值和Y值，关系解析就是从XML文件中解析出图谱中所有节点之间边的源节点Id和目标节点Id，从XML文件中解析出节点和关系信息的图谱节点和关系解析流程图如图 3.2所示。



图 3.2 图谱节点和关系解析流程图

解析XML图谱文件主流方法主要有DOM、SAX、JDOM和DOM4J等方法，由于DOM4J解析方法相比其他三种方法，首先不仅可以对XML图谱文件进行增、删、改、查，还具有更大的灵活性和简单性；其次，DOM4J解析XML图谱文件时不受内存大小的限制，可以解析比内存大的XML图谱文件；最后不得不提的是DOM4J解析方法具有更快的速度和效率；所以XML图谱文件采用DOM4J方法解析，如下所示。

Step1：输入XML文件的路径，使用read()函数获取XML文件的Document；

Step2：使用getRootElement()函数和elementIterator()函数获取根标签map的Iterator；

Step3：使用getName()函数和elementIterator()函数获取map下的子标签people和connect的Iterator；

Step4：使用getName()函数和elementIterator()函数获取people下的子标签width、height、X和Y各自的Iterator，以及connect下的子标签sourceid和targetid各自的Iterator；

Step5：使用getStringValue()函数获取people子标签width、height、X和Y的内容，将其分别对应存储在Vector变量widNode、heiNode、nodesx、nodesy中，子标签所属的节点编号i就是Vector变量的第i-1个元素；使用getStringValue()函数获取connect的子标签sourceid和targetid的内容，分别存储在Vector变量sourceNode和targetNode中。

Step6：输出存储节点width、height、X和Y内容的Vector变量widNode、heiNode、nodesx、nodesy，以及存储边sourceid和targetid内容的Vector变量sourceNode和targetNode。

节点解析使用DOM4J方法解析，其实质就是一层一层的遍历XML标签，首先需要获取根标签map，然后获取map下的子标签connect，在获取map的子标签sourceid和targetid，最后获取子标签sourceid和targetid的内容，将其各自存储在Vector变量中。

## 3.3 人员社会关系图谱节点分层

在实现人员社会关系图谱节点分层时，需要先对图谱使用社团发现算法进行图谱聚类得到子图谱，然后将子图谱抽象为一个抽象图谱。

### 3.3.1 基于社团发现算法的图谱聚类

图谱的节点分层是通过基于社团发现算法的图谱聚类实现的，图谱聚类就是将图谱按照一定的规则分割为若干个子图谱，图谱布局的目的就是消除图谱中节点重叠，减少节点之间的边交叉数量，让布局更加合理，图谱的内在结构更加清晰，社团发现算法用于发现图谱中具有社区性质的子图谱，将相似的节点归为同一个子图谱，所以使用社团发现算法对图谱进行聚类。

社团发现算法属于社区发现算法，是一种边介数算法，本质是分层聚类思想。边介数表示图谱中节点之间的边在整个图谱中的控制力强度，是一个重要的社区发现变量，具有很高的参考价值。边介数的大小与该边连接的聚类图谱的聚集程度高低成正比关系，将此条边删除后，就可以得到两个社区性质更加明显的图谱，如果一直这样划分，就可以将整个图谱划分为具有社区性质的图谱。

社团发现算法就是按照边介数的大小不断地去掉图谱中的边，核心是聚类中的分裂思想，以边介数作为是否去除该边的尺度，其愈大，被删除的可能性也就愈大；由于该算法不断删除那些连接两个具有明显社区性质的子图谱，所以该算法收敛的速度相当快。

社团发现为了衡量所划分图谱的优劣，后来在社团发现函数中引入了Q函数，模块性Q[16]函数的值与所划分子图谱的社区性质成正比，其值愈大，划分子图谱的社区性质愈好。无权图谱和有权图谱的模块性函数具有不同的计算方法，在无权图谱中，Q函数表述见公式（3-1）。

（3-1）

公式（3-1）中，表示存储该图谱的邻接矩阵的迹，表示存储图谱的邻接矩阵中每一行或每一列的值，即第K个社团中的节点相连的边在所有边中所占的比例，在加权图谱中，Q函数见公式（3-2）。

（3-2）

公式（3-2）中为图谱邻接矩阵的元素，如果n和m 两节点之间存在边，则为n和m 两节点之间边的权重，否则等于 0，另外，图谱聚类后节点之间是否存在边的概率为 。M为图谱中所有边的权重之和，见公式（3-3）。

（3-3）

在正常的社交图谱中，模块性值越大，越能说明图谱具有更加明显的的社区结构，基于社团发现算法的图谱聚类流程图如图 3.4所示。



图 3.4 基于社团发现算法的图谱聚类流程图

本课题中的图谱是有权图谱，根据上面对于基于社团发现算法的图谱聚类的论述，下面直接给出基于社团发现算法的图谱聚类的伪代码。

|  |  |
| --- | --- |
| 输入： | 存储图谱的邻接矩阵 |
| 输出： | 删除边之后的图谱邻接矩阵 |
| （1） | 初始化图谱邻接矩阵 |
| （2） | 广度优先遍历初始化无权图谱任意两点之间的最短路径 |
| （3） | 初始化所有连接边的边介数PR |
| （4） | 初始化所有边的边权比PQV |
| （5） | 删除边权比最大的所有边CON，根据公式（3-2）计算图谱的Q值 |
| （6） | 重复步骤（1）（2）（3）（4）（5），直到图谱中所有的边均被移除 |
| （7） | 计算出最大的Q值 |
| （8） | 输出Q值最大时的图谱邻接矩阵 |
| （9） | 算法结束 |

社团发现算法在实现过程中，第一，需要求出图谱中任意两点之间的最短路径[17]，并将其存储在邻接矩阵中，此处的最短路径指的是节点与节点之间边的条数，在求解图谱的最短路径时，可以使用广度优先遍历也可以使用Dijkstra算法或者弗洛伊德[18]算法；第二，计算图谱中所有边的边介数，边介数是指该边在图谱中所有最短路径中出现的次数；第三，计算图谱中所有边的边权比，边权比是指该边的边介数与它的权重的比值；比较图谱中所有边的边权比，将边权比最大的边在图谱中删除，当边权比最高的边又多条时，将它们全部删除，依据公式（3-2）计算模块性Q函数，直到所有的边都被移除，此时求出最大的Q值，并且将图谱恢复到此时的图谱状态，此时的图谱状态就是社区发现社团发现算法最终计算得到的图谱状态，它具有最好的模块性质。

通过基于社团发现算法的图谱聚类之后，就得到了存储Q值最大时图谱的邻接矩阵，该邻接矩阵存储的是若干个不连通的子图谱。

### 3.3.2 构造抽象图谱

构造抽象图谱就是将各个孤立的子图谱抽象为一个完整的抽象图谱，本部分构造抽象图谱就是将基于社团发现算法图谱聚类之后得到的若干个子图谱抽象为一个抽象图谱。构造抽象图谱的目的是将其进行KK图谱算法布局，所以要初始化一系列参数，这些参数包括理想距离、子图谱中所有节点的聚集系数、子图谱的特征向量和子图谱中所有节点的PR[19]值之和。

抽象图谱与单个节点不同，在KK图谱算法中要用到一些非常重要的参数，其中一个就是理想距离。在原始的图谱中，理想距离是指节点与节点之间的边数，但是在抽象图谱中这么定义就显得不合理，所以需要重新定义理想距离。在给出抽象图谱理想距离的距离之前，需要先对子图谱的特征向量T和子图谱的聚集系数进行论述。

节点聚集系数表示为，它表示图谱中节点的聚集性，换句话说，在子图谱中与节点相邻的两个节点仍然相邻的可能性是多少，的具体表示见公式（3-4）。

（3-4）

公式（3-4）中表示与该子图谱中节点相连接的节点个数，表示在该子图谱中与节点相邻的节点之间实际存在的边个数，子图谱的特征向量具体表示见公式（3-5）。

（3-5）

公式（3-5）中是子图谱中所有节点的聚集系数之和，表示子图谱中所有节点的PR值之和，PR值反映该子图谱的影响力程度，PR值越大该子图谱影响程度越大。基于上面对子图谱的特征向量*T*和聚集系数C的详细描述下面给出抽象图谱理想距离具体公式表示，见公式（3-6）。

（3-6）

公式（3-6）就是通过子图谱和子图谱之间的余弦相似度来得到的两个抽象图谱之间的理想距离，其中表示该三维空间的直径长度。

在图谱中，当一个节点指向节点时，说明节点认为节点比较重要，从而将自己的一部分重要性给节点，将的出链数N看做该节点拥有的重要程度。每个节点开始时的重要程度设置为PR/N，PR值的计算方法见公式（3-7）。

（3-7）

公式（3-7）中d表示与节点v属性相关连的随机概率，大多数情况默认d=0.85，N表示图谱中的节点总数。很明显PR(u)表示节点u影响力的大小，反映节点u在图谱中的重要程度。

从公式（3-7）可以清楚的看出，要想求出节点u的PR(u)值需要求出节点v的PR(v)值。根据马尔科夫[20]收敛定理可以看出的计算符合马尔可夫收敛定理，所以根据公式（3-7）和马尔可夫收敛定理可以求出PR值。

在初始状态下，首先需要初始化存储图谱的邻接矩阵；然后将所有节点的默认PR值设置为1/N，N图谱中的节点总个数，即将其的重要程度设置为一个定值；由于节点的PR值的计算符合马尔可夫收敛定理，所以使用公式（3-7）计算节点u的PR值，不断更新所有节点的PR值，直到达到一定的迭代次数。

上面论述了构造抽象图谱时所需的参数如何计算，下面给出抽象图谱构造算法流程图，如图 3.5所示。



图 3.5 抽象图谱构造算法流程图

基于上面对抽象图谱算法的理论思想详细论述，下面给出抽象图谱算法的伪代码实现。

|  |  |
| --- | --- |
| 输入： | 存储聚类图谱的邻接矩阵 |
| 输出： | 存储抽象图谱的邻接矩阵和抽象图谱节点间的理想距离矩阵 |
| （1） | 初始化聚类图谱邻接矩阵 |
| （2） | 广度优先遍历图谱，每次遍历完一个连通分量就单独存起来，并且初始化该连通分量的中心点 |
| （3） | 构建子图谱的特征向量 |
| （4） | 通过特征向量计算整个抽象图谱节点间的理想距离 |
| （5） | 输出存储抽象图谱的邻接矩阵和抽象图谱节点间的理想距离矩阵 |
| （6） | 算法结束 |

通过社团发现算法，可以得到存储聚类图谱的邻接矩阵，广度优先遍历存储聚类图谱的邻接矩阵，将单独的连通分量的节点Id保存在Vector数据结构中。根据公式（3-4）、公式（3-7）和公式（3-5）计算各个连通分量（子图谱）的特征向量，然后根据求得的特征向量和公式（3-6）计算子图谱理想距离，并且构造抽象图谱的理想距离矩阵。

## 3.4人员社会关系图谱节点和关系布局

人员社会关系图谱布局是基于KK图谱能量模型实现的，所以先对KK图谱能量模型进行论述，然后使用KK图谱能量模型对抽象图谱和子图谱布局进行布局。

### 3.4.1 KK图谱能量模型

KK图谱模型是Kamada 和Kawai[21]提出的一个能量模型，其思想就是减少系统总能量来改变图谱中节点的坐标。KK图谱能量模型遵循“胡可定律”。图谱的总能量表示为：

（3-8）

是图谱中节点与节点之间力的强度，是常数K与最短距离的平方的比值。公式表示为：

（3-9）

在KK图谱能量模型中第一次引入了“理想距离”[22]的概念。理想距离就是：图谱中任意两个顶点之间的理想距离与图谱中任意两个顶点之间的最短路径成正比。公式表示为：

（3-10）

公式（3-8）中E是图谱的中能量；n为图谱中节点总数，公式表示为；是节点对应的位置向量；表示社交图谱中节点与节点之间最短距离，需要指出的是，此处的最短距离是从节点到节点经过的边数。公式（3-9）中常数K表示节点之间影响力强度的大小。公式（3-10）中是图谱中节点与节点之间关系的理想距离长度；图谱中两点之间关系的理想距离长度一般有两种定义方式，的第一种定义方式公式表示为：

（3-11）

的第二种定义方式公式表示为：

（3-12）

公式（3-11）中L0是显示方块区域中一条边的长度。公式（3-12）中W是显示区域的宽，H是显示区域的高，是图谱中节点的总个数。

基于KK图谱能量模型的KK图谱算法在求解图谱最优布局的过程就是不断较少系统总能量。在求出系统总能量的基础上对各个节点求偏导数，如公式（3-13）。

（3-13）

公式（3-13）中E是系统总能量，反映节点的偏移量大小。求出最大的节点m，然后根据公式（3-14）和公式（3-15）求出节点m的偏移量和。

（3-14）

（3-15）

KK图谱模型算法的核心就是公式（3-14）和公式（3-15），先求出偏移量最大的节点，然后根据关于系统总能量E的偏微分方程来迭代减少系统总能量E，直到满足一定的条件，算法才终止，此条件也是影响算法计算复杂度的一个重要因素，后文算法优化部分将详细阐述。在求得节点m的偏移量之后需要更新节点m的横坐标和纵坐标，根据公式（3-16）更新节点m的横坐标，根据公式（3-17）更新节点m的纵坐标。

（3-16）

（3-17）

综上所述，KK图谱模型算法实质就是迭代减小系统总能量，让图谱中节点的能量值达到最小。接下来，对KK图谱算法的整个流程做一个简单描述。在此，需要做几点声明：为方便描述，暂且默认图谱使用邻接矩阵存储，节点之间的最短距离以及其他系数使用二维数组存储。

上面对KK图谱模型做了详细描述，KK图谱模型是二维空间的，这儿需要将二维KK图谱模型扩展到三维空间。与二维空间相比，三维空间多了一个z坐标轴。首先在三维空间中，系统总能量E计算公式不变，仍然可以采用公式（3-8）计算。系数矩阵的计算方法不变。但是，反映节点的偏移量大小的需要加入第三维坐标轴，见公式（3-18）。

（3-18）

其次，偏微分方程也需要加入第三维坐标轴，见公式（3-19）、公式（3-20）和公式（3-21）。

（3-19）

（3-20）

（3-21）

最后，在更新节点m时需要加入对第三维坐标轴的更新，见公式（3-22）。

（3-22）

至此，已经将二维的KK图谱模型添加第三维坐标轴，扩展到了三维KK图谱模型，今后便可以写出基于三维的KK图谱算法。

### 3.4.2 基于KK图谱算法抽象图谱布局

基于KK图谱算法抽象图谱布局就是将抽象得到的抽象图谱使用KK图谱算法改变抽象图谱节点的坐标，实现了图谱的整体布局，基于KK图谱算法抽象图谱布局流程图如图3.6所示。



图3.6 基于KK图谱算法抽象图谱布局流程图

基于KK图谱模型的抽象图谱布局应用KK图谱能量模型改变抽象图谱节点的坐标，实现了对抽象图谱的合理布局。

下面详细阐述基于KK图谱模型的抽象图谱布局算法的流程：

Step1：从存储图谱节点坐标和边源节点和目标节点的Vector中，获取节点坐标以及边的源节点和目标节点，初始化抽象图谱的邻接矩阵；

Step2：通过Dijkstra算法求解抽象图谱的邻接矩阵的最短距离矩阵；

Step3：根据公式（3-16）初始化抽象图谱中节点之间的理想距离矩阵，并存储在二维数组中，以及表示节点之间的强度系数，存储在二维数组中，根据公式（3-8）计算抽象图谱的总能量E；

Step4：根据公式（3-18）计算图谱中所有节点的能量，存储在二维数组中，并且求解出能量最大的节点；

Step5：依据公式（3-19）、公式（3-20）以及公式（3-21）表示的偏微分方程，求解出节点关于横坐标的位移和关于纵坐标的位移以及关于第三维坐标轴的位移；

Step6：由公式（3-16）、公式（3-17）和公式（3-22）更新存储在Vector中节点的位置；

Step7：重新计算图谱的总能量E以及能量值最大的节点。

Step8：迭代减少节点的最大能量值，直到能量值最大的节点的能量小于设定的阈值。

根据上面对KK图谱能量模型KK图谱能量模型的描述以及流程图的详细解释，下面直接给出实现基于KK图谱模型的抽象图谱布局算法的伪代码。

|  |  |
| --- | --- |
| 输入： | 存储抽象图谱的邻接矩阵 |
| 输出： | 存储布局之后的抽象图谱邻接矩阵 |
| （1） | 初始化抽象图谱邻接矩阵 |
| （2） | 计算抽象图谱最短距离矩阵、总偏移量E、系数矩阵 和 |
| （3） | 计算抽象图谱中偏移量最大的节点*pm’* |
| （4） | 计算节点*pm’*位移向量*(δx,δy,δz)* |
| （5） | 若节点*pm’*的偏移量大于阈值，则执行下一步，否则执行步骤（2）（3）（4） |
| （6） | 更新节点*pm’*的位置 |
| （7） | 抽象图谱的总偏移量E |
| （8） | 抽象图谱中偏移量最大的节点*pm’* |
| （9） | 若节点*pm’*的偏移量大于阈值VALUE，则执行下一步，否则执行步骤（2）到（8） |
| （10） | 输出抽象图谱邻接矩阵 |
| （11） | 算法结束 |

### 3.4.3 基于KK图谱算法子图谱布局

基于KK图谱算法子图谱布局就是将子图谱使用KK图谱算法改变图谱节点的坐标，基于KK图谱算法子图谱布局流程图如图3.7所示。



图 3.7 基于KK图谱算法子图谱布局流程图

基于KK图谱模型的子图谱布局应用KK图谱能量模型改变子图谱节点的坐标，实现了对子图谱的合理布局。前面已经详细介绍了KK图谱模型的原理以及三维空间下KK图谱算法的理论与实现，

聚类子图谱就是使用KK图谱算法布局，根据的KK图谱模型就可以写出KK图谱布局算法，KK图谱能量模型与KK图谱能量模型相似，都遵守胡克定律，最优模型布局就是系统总能量最少，所以，求解KK图谱模型的最优布局就是迭代减少系统总能量。

KK图谱算法思想与基于二维KK图谱模型算法也相似。KK图谱算法实在三维的空间环境条件下，每次计算偏移量最大的节点,移动节点的坐标，然后有计算偏移量最大的节点，接着移动节点的坐标，如此不断迭代移动偏移量最大的节点，直至节点的最大偏移量小于设定的阈值。

下面是实现基于KK图谱算法子图谱布局算法的伪代码。

|  |  |
| --- | --- |
| 输入： | 存储子图谱的邻接矩阵 |
| 输出： | 存储布局之后的子图谱邻接矩阵 |
| （1） | 初始化子图谱邻接矩阵 |
| （2） | 计算子图谱最短距离矩阵、总偏移量E、系数矩阵和 |
| （3） | 计算子图谱中偏移量最大的节点*pm* |
| （4） | 计算节点*pm*位移向量(*δx,δy,δz*) |
| （5） | 若节点*pm*的偏移量大于阈值，则执行下一步，否则执行步骤（2）（3）（4） |
| （6） | 更新节点*pm*的位置 |
| （7） | 子图谱的总偏移量E |
| （8） | 子图谱中偏移量最大的节点*pm* |
| （9） | 若节点*pm*的偏移量大于阈值，则执行下一步，否则执行步骤（2）到（8） |
| （10） | 输出子图谱邻接矩阵 |
| （11） | 算法结束 |

## 3.5 本章小结

人员社会关系图谱布局加速算法是本课题的核心内容，主要包括节点解析、关系解析、节点分层、节点布局和关系布局五大部分。节点解析和关系解析使用DOM4J解析方法从XML文件中解析出图谱信息；节点分层通过社团发现算法对源图谱进行聚类，得到聚类子图谱，然后构造抽象图谱将源图谱分为两层；节点布局和关系布局使用KK图谱算法进行布局，首先对抽象图谱进行布局，实现对图谱的整体布局，然后对子图谱布局，最终完成了对图谱的布局加速。

# 第4章 基于多线程的人员社会关系图谱展示加速算法实现

## 4.1 总体实现

图谱展示加速算法是通过多线程解析XML文件、展示图谱节点和边来实现图谱加速展示的算法，图谱展示加速算法思想是：首先获取XML文件，然后运行若干个线程，使用这几个线程分段解析XML文件，将读取节点和边的信息传递给可视化工具提供的接口，从而将图谱展示出来。该算法主要包括三个方面：展示策略、节点展示以及关系展示。基于多线程展示加速算法流程图如图 4.1所示。



图 4.1 基于多线程展示加速算法流程图

（1）展示策略

展示策略是通过多线程来解析XML文件和展示节点和边，一般情况下，都是采用单线程来展示图谱中节点和关系，单线程是该进程中只有一个线程，在程序执行过程中，程序是线性的，程序从开始到结束完全按照一个方向来执行，前面的执行完，后面的才能接着执行。多线程是该进程中有多个线程，可以有多个线程同时执行，实现了多个线程之间并发执行[23]的技术。所以，如果通过多线程来展示图谱的节点和关系，便能提高展示图谱的效率。另外，每个线程需要消耗一定的CUP资源来管理，最主要的是进程的上下文切换需要消耗资源，如果频繁的上下文切换，就会出现抖动现象，CUP的有效利用机会很低，因此针对不同的节点和关系数目，需要设置不同数目的多线程，以最大幅度的提高图谱的展示速度。在人员社会关系图谱的展示过程中，主要的耗时在图谱文件XML文件的解析和图谱中节点和关系的展示上，所以主要对这两点进行优化。多线程便可实现较大程度的优化，用多线程解析XML文件，用多线程展示图谱的节点和关系。

（2）节点展示

节点展示就是从存储图谱的XML文件中解析出所有节点id值、横坐标X值、纵坐标Y值、节点的宽度widNode和节点的高度heiNode，然后将其在画布上展示出来，本课题使用的可视化工具只需要将有节点id值、横坐标X值、纵坐标Y值、节点的宽度widNode和节点的高度heiNode分别存储在Vector类型的变量中，将这些变量的地址存放在Vector类型的变量people中，最后将Vector变量people存储在map的数据结构中，然后传给该工具就可将图谱展示出来。

（3）关系展示

节点展示就是从存储图谱的XML文件中解析出节点之间所有边的源节点Id与目标节点Id，然后将其在画布上展示出来，本课题使用的可视化工具只需要将节点信息存储在map的数据结构中，然后传给该工具就可将图谱展示出来。

## 4.2展示策略

展示策略就是根据节点的数量选择合适的多线程数去解析XML，然后展示节点和节点之间的关系，所以多线程的数目和节点展示以及关系展示方法需要制定合适的策略。

（1）多线程的数目

多线程的数目会影响图谱可视化的速度，多线程可以同时运行多个程序，多线程可以提高程序运行效率的本质是当一个线程不需要CPU时可以快速的将CPU分配给其他其他线程，从而提高了CPU的利用率。但是多线程也存在许多缺点：对多线程资源分配和管理需要消耗额外的CPU和内存，更加重要的是线程之间的上下文切换需要消耗额外的CPU和内存，当这种消耗大于多线程本身的优点时，多线程就变得不在优化；更加重要的是，如果频繁的切换线程，会出现抖动现象，此时CPU的利用率虽然很高，但是实际做的有用功却特别低。所以要防止出现抖动现象。具体方法就是开合适的线程数，让CPU最大化的使用，并且做的都是有效功。

（2）节点展示和关系展示的方法

节点展示和关系展示的方法也会影响图谱可视化的速度，本课题中节点和关系展示是调用一个图谱可视化的工具，而该工具是将节点的横纵坐标以及关系的源节点id和目标节点id封装到一个数据结构中，然后展示出来，所以节点展示和关系展示实质就是构造这个包含所有节点的横纵坐标以及关系的源节点id和目标节点id的数据结构，然后调用图谱可视化工具接口将其展示出来。

综上所述，多线程的数目以及节点和边的展示方法需要根据具体的节点和关系的数量来制定一定的策略。所以，对于多线程数目的设定考虑得到本课题研究的内容时千量级的图谱，因此暂且使用少量的线程数来解析XML文件；对于节点和边的展示顺序，本课题采用的方法是直接调用可视化工具提供的接口，将节点横坐标和纵坐标以及关系的源节点id和目标节点id传递给可视化工具。

## 4.3 XML解析

XML解析就是使用合适的线程数从XML图谱文件中解析出所有节点的横纵坐标以及关系的源节点id和目标节点id。其思想是：先获取XML文件，运行相应的线程数，每个线程平均解析所有的节点和边，最后一个线程解析剩下的所有节点和边。通过多线程解析XML图谱文件以及展示图谱的节点和关系是此算法的核心思想，多线程解析XML文件并展示图谱流程图如图 4.2所示。

  
图 4.2 多线程解析XML文件并展示图谱流程图

鉴于DOM4J优越的性能，该算法在解析XML文件时使用DOM4J方法，下面对多线程解析XML文件并展示图谱的流程进行详细论述：

Step1：输入XML文件的文件路径；

Step2：使用read()函数获取XML文件的Document；

Step3：使用start()函数运行N个线程；

Step4：计算节点总数numNode和边总数numCon；

Step5：第i个线程解析第（i-1）\*numNode/N ~ i\*numNode/N的节点；

Step6：第i个线程解析第（i-1）\* numCon /N ~ i\* numCon /N的边；

Step7：最后一个线程解析剩下的所有节点和边。

Step8：使用getRootElement()函数和elementIterator()函数获取根标签map的Iterator；

Step9：使用getName()函数和elementIterator()函数获取map下的子标签people和connect的Iterator；

Step10：使用getName()函数和elementIterator()函数获取people下的子标签width、height、X和Y各自的Iterator，以及connect下的子标签sourceid和targetid各自的Iterator；

Step11：将 Vector变量widNode、heiNode、nodesx、nodesy、sourceNode和targetNode存储在Map变量map中；

Step12：输出存储节点节点的横坐标和纵坐标以及关系的源节点id和目标节点id的变量map。

Step13：算法结束

下面给出基于多线程的人员社会关系图谱展示加速算法的伪代码。

|  |  |
| --- | --- |
| 输入： | 存储图谱信息的XML文件 |
| 输出： | 可视化图形 |
| （1） | 获取XML文件的Document |
| （2） | Start N个新的线程 |
| （3） | 让第i个线程每次解析SUMNode/N个节点坐标 |
| （4） | 让第i个线程每次解析SUMSide/N条边的源节点以及目标节点 |
| （5） | 最后一个线程解析剩下的所有节点和边 |
| （6） | 展示解析到的节点和边 |
| （7） | 算法结束 |

本课题研究的是千量级的图谱，通过实验测试发现当线程数目为4时，并且在千量级的图谱情况下，都能取得比较合理的结果，所以在千量级图谱的展示策略上选取四个线程。

首先运行4个线程，让每个线程平均去解析一定数量的people标签和connect标签，并且将解析到的节点坐标、边的源节点和目标节点存储到一个map的数据结构中；比如，总共有1000个节点和1000个边，第一个线程去解析第0到第250的people标签和第0到第250的connect标签，第二个线程去解析第251到第500的people标签和251到第500的connect标签，以此类推，最后一个线程解析剩下的所有节点标签和边标签。XML文件解析结束后将得到一个存储节点信息和边信息的map，最后将传给可视化工具，从而将图谱显示出来。

## 4.4 节点展示和关系展示

节点展示和关系展示是调用一个图谱可视化的工具，而该工具是将节点的横纵坐标以及关系的源节点id和目标节点id封装到一个map的数据结构中，然后展示出来，所以节点展示和关系展示实质就是构造这个包含所有节点的横坐标和纵坐标以及关系源节点id和目标节点id的数据结构，然后调用图谱可视化工具接口将其展示出来。

存储节点和边信息的类图如图4.3所示。



图4.3 节点和边信息的类图

该Map类型的变量map数据结构中，存储节点信息和关系信息。

从XML文件中解析到的节点的信息存储在Vector变量的people中，peopleList中存储节点的宽度、高度、横坐标和纵坐标，存储的变量名称分别为widNode、heiNode、nodesx和nodesy；节点的横坐标存储在Vector变量nodesx中，纵坐标存储在Vector变量nodesy中，节点的宽度存储在Vector变量widNode中，节点的宽度存储在Vector变量heiNode中，关键代码如下所示。

if (nodeChild.getName() == "id"){

// 获取id

id = Integer.valueOf(nodeChild.getStringValue());

}

if (nodeChild.getName() == "x") {

// 获取x值

dx = Double.valueOf(nodeChild.getStringValue());// 获取x值

nodesx.addElement(dx);

}

if (nodeChild.getName() == "y") {

// 获取y值

dy = Double.valueOf(nodeChild.getStringValue());

nodesy.addElement(dy);

}

从XML文件中解析到的关系的信息存储在Vector变量的connection中，connection中存储关系的源节点和目标节点，存储的变量名称分别是sourceNode和targetNode；关系的源节点id存储在Vector变量sourceNode中，关系的目标节点存储在Vector变量targetNode中，关键代码如下所示。

if (nodeChild.getName() == "sourceid") {

// 获取源节点值

sourceid = Integer.valueOf(nodeChild.getStringValue());

sourceNode.addElement(sourceid);

}

if (nodeChild.getName() == "targetid") {

// 获取目标节点值

targetid = Integer.valueOf(nodeChild.getStringValue());

targetNode.addElement(targetid);

}

## 4.5 本章小结

基于多线程的人员社会关系图谱展示加速算法是本课题的第二个重要算法，主要包括展示策略、节点展示以及关系展示三大部分。展示策略的选择就是通过运行合适的多线程来解析存储节点横坐标和纵坐标以及关系的源节点id和目标节点id的XML图谱文件，图谱的节点展示和关系展示就是将读取的节点横坐标和纵坐标以及关系的源节点id和目标节点id传递该可视化工具，通过多线程可以从XML图谱文件中快速的解析出节点和边的信息，通过多线程可以快速的展示节点和边。

# 第5章 实验结果与分析

## 5.1 总体设计

可视化加速算法实验目的是检验该算法的有效性，本实验是在Windows平台上使用CPU为Intel i5-6500 3.20 GHz、内存为8.00GB的配置eclipse环境的机器上运行的，程序使用java语言编写。图谱可视化加速算法主要分为两个子算法，所以其实验包括图谱布局加速算法实验和图谱展示加速算法实验两部分。

（1）图谱布局加速算法实验

为了验证图谱布局加速算法在图谱布局时的有效性，本实验使用两个数据集来验证，社会人员关系图谱数据集和不同量级的图谱数据集。社会人员关系图谱数据集是具有实际意义的数据集，不同量级的图谱数据集是不具实际意义的，通过程序自动生成的，节点数量在[0,500]、[501,1000]、[1001,1500]、[1501,2000]、[2001,2500]…[9500,10000]等不同区间的数据集，使用这两种数据集分别对基于KK图谱算法的人员社会关系图谱布局加速算法做了测试。根据第一个数据集的实验结果从节点重叠程度、边交叉数目的多少、图谱的整体布局效果等方面分析可视化加速算法的布局效果；根据第二个数据集的实验结果从节点数目和布局时间分析图谱布局加速算法的时间复杂度，最后分析该算法的有效性。

（2）图谱展示加速算法实验

为了验证图谱展示加速算法在图谱展示时的有效性，该实验使用节点数量在[0,500]、[501,1000]、[1001,1500]、[1501,2000]、[2001,2500]…[9500,10000]等不同区间的数据集。从节点数目和展示耗时分析图谱展示加速算法的时间复杂度，并且绘制出相应的可视化时间曲线，分析该算法的有效性。

## 5.2 基于KK图谱算法的人员社会关系图谱布局加速算法实验

### 5.2.1 实验设计

基于KK图谱算法人员社会关系图谱布局加速算法实验验证该算法的有效性以及加速效果，首先验证布局加速算法是否能够合理布局，是否对原有的算法进行了优化；其次验证展示加速算法是否可以有效展示图谱，是否相对于单线程进行了优化。实验的设计主要包括实验数据的构造、获取以及实验结果和实验分析三个方面。

第一，实验测试的数据集是验证算法的有效性以及优劣的重要因素。本实验的数据集主要来自两个途径，第一个是社会人员关系图谱数据集[24]。这个数据集构造了一个包括62个节点和159条边的社交图谱，节点表示人，节点之间的边表示人之间具有紧密的交流与对话。如果两个人之间具有频繁的交流现象，那么在这个构造社会图谱中人对应的节点之间就会有一条边。第二个数据集是不具实际意义的，属于程序随机构造的数据集，本课题所研究的图谱布局加速算法是千量级的节点加速布局，所以数据集必须达到千量级，这个数据集通过程序自动生成，并且具有一定节点和边的。每500个节点的间隔，从500个节点一直到10000个节点，分别对基于KK图谱算法的人员社会关系图谱布局加速算法做了测试。第一个社会人员关系图谱数据集主要用于验证图谱布局加速算法的有效性，第二个数据集主要验证图谱布局加速算法的加速效果。主要从时间复杂度上进行分析。第二，使用基于KK图谱算法的人员社会关系图谱布局加速算法实完成实验，获取实验结果。第三，在布局效果和时间复杂度方面对算法的有效性和优劣进行分析，基于不同节点数目的图谱，所做的布局加速实验结果，从中分析出规律或者总结出实验结论。

### 5.2.2 实验结果

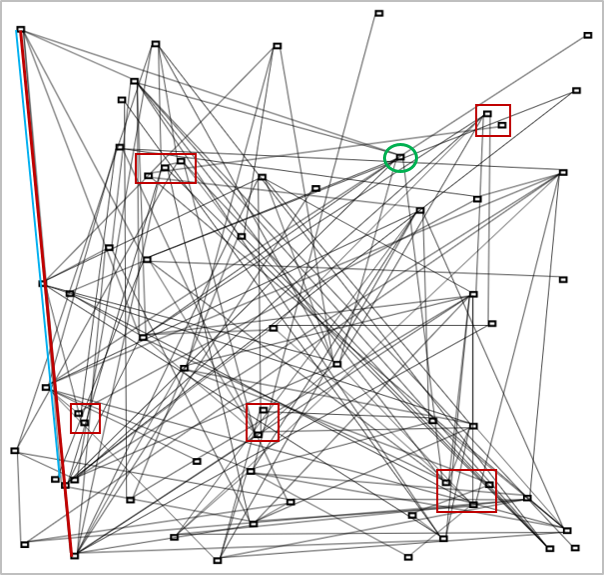
社会人员关系图谱数据集是一个包括62个节点和159条边的社会图谱。通过这个社交图谱可以验证图谱布局加速算法的有效性。在未对其使用KK图谱算法布局前，社会人员关系图谱数据集未布局可视化图如图 5.1所示。

图 5.1 社会人员关系图谱数据集未布局可视化图

从图 5.1可以看出蓝色边和红色边完全重合，红色矩形框中的节点距离太近，导致该图整体十分杂乱的，很难辨清哪些节点之间有关系，哪些节点之间没有关系；椭圆形绿色中的节点本来连接了许多节点，该节点显得非常重要，但是在改图中完全显示不出来，完全无法看出哪些节点是重要节点，具有连通其他节点的作用。下面给出使用KK图谱算法布局之后的图谱，社会人员关系图谱数据集KK图谱算法布局可视化图如图 5.2所示。

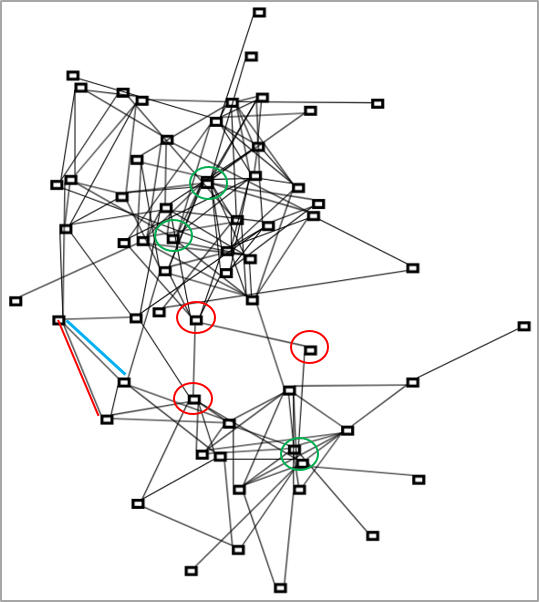


图 5.2 社会人员关系图谱数据集KK图谱算法布局可视化图

观察图5.2，可以发现图中虽然边仍然存在交叉，但是节点之间的联系可以清楚的表达出来，红色线条和蓝色线条显示的十分清楚；绿色椭圆圈起来的节点是与大部分节点有关系的节点，在次图中也可以清楚的辨认；并且红色椭圆圈起来的三节点起到连通两个社区的作用，在整个图谱中具有举足轻重的作用，在本图中也显示的特别清楚。该图整体上可以清晰的辨清图中节点之间的关系，并且能够轻而易举的辨别出哪些节点与多个节点连通，哪些节点与很少的节点连接。

以上是社会人员关系图谱数据集验证KK图谱算法的布局效果；接下来通过测试范围从500个节点到10000个节点的图谱，间隔为500个节点，记录运行时间如图 5.3所示。

图5.3 图谱可视化布局加速算法时间

从图5.3可以清楚地看出当500个节点时消耗时间0.8分钟，1000个节点时消耗时间7.1分钟，2000个节点消耗时间25.2分钟，3000个节点消耗194.6分钟，4000个节点消耗456.3分钟，5000个节点消耗837.8分钟，6000个节点消耗1400.8分钟，7000个节点消耗1900.6分钟，8000个节点消耗2567.7分钟，9000个节点消耗3408.6分钟，10000个节点消耗4429.5分钟。从图5.3结合数据可以看出，随着节点数目的增加，布局耗时也在慢慢增加，并且增加的幅度也在慢慢变大，从500个节点到1500节点时间消耗增长了22.1分钟，从1500个节点到2500个节点时间消耗增长了35分钟，从2500个节点到3500个节点时间消耗增长了234.6分钟，从3500个节点到4500个节点时间消耗增长了333.7分钟，从4500个节点到5500个节点时间消耗增长了489.1分钟，从5500个节点到6500个节点时间消耗增长了479.5分钟，从6500个节点到7500个节点时间消耗增长了654.8分钟，从7500个节点到8500个节点时间消耗增长了783.5分钟，从8500个节点到9500个节点时间消耗增长了870.5分钟。

总之，图谱的节点数目越多消耗时间越长，时间消耗的增加也在随着节点数目的增加而增加。

### 5.2.3 实验分析

对于基于KK图谱算法的人员社会关系图谱布局加速算法分别用了社会人员关系图谱数据集验证了KK图谱算法布局的有效性，使用随意生成的从500个节点到10000个节点的图谱测试了KK图谱算法的时间复杂度，及其随着节点数目变化时间的变化曲线。

实验证明，KK图谱算法的布局效果完全达到了预期，节点和边显示合理，节点和节点之间的关系清晰明了，边交叉也比较少。在不超过1000个节点时算法的时间复杂度保持在一个较小的范围里边，当800个节点时，时间消耗不超过一分钟。

## 5.3 基于多线程的人员社会关系图谱展示加速算法实验

### 5.3.1 实验设计

本实验是对基于多线程的人员社会关系图谱展示加速算法的验证，在图谱能够展示的基础上进行加速。

本实验通过比较多线程展示策略和单线程展示策略花费的时间来体现加速。本次实验测试展示加速算法使用的数据集通过程序自动生成具有一定节点和边的千量级图谱。节点区间为[0,500]、[501,1000]、[1001,1500]、[1501,2000]、[2001,2500]…[9500,10000]的图谱使用单线程方法和多线程方法分别展示，然后比较各自所消耗的时间。

最后，基于不同节点数目的图谱，所做的展示加速实验结果，从中分析出规律或者总结出实验结论。

### 5.3.2 实验结果

实验测试的社会图谱数据包括500、1000、1500…10000个节点的图谱，分别使用单线程展示策略和多线程展示策略，如图 5.4所示。

图 5.4 图谱展示加速算法运行时间

图 5.4是不同数量节点的图谱使用单线程策略展示和多线程策略展示时，所消耗的不同时间。当图谱具有500个节点时单线程耗时127毫秒多线程耗时112毫秒，1500个节点时单线程耗时210毫秒多线程耗时210毫秒，2500个节点时单线程耗时253毫秒多线程耗时235毫秒，3500个节点时单线程耗时290毫秒多线程耗时270毫秒，4500个节点时单线程耗时336毫秒多线程耗时321毫秒；5500个节点时单线程耗时370毫秒多线程耗时356毫秒；6500个节点时单线程耗时430毫秒多线程耗时410毫秒；7500个节点时单线程耗时510毫秒多线程耗时479毫秒；8500个节点时单线程耗时546毫秒多线程耗时518毫秒；9500个节点时单线程耗时590毫秒多线程耗时518毫秒；10000个节点时单线程耗时610毫秒多线程耗时566毫秒；

总体上，多线程策略展示要优与单线程策略展示。在500个节点时多线程展示策略消耗112ms的时间，单线程展示策略需要127ms的时间，它们之间没有太大的区别。虽然多线程展示策略和单线程展示策略之间的差别不是特别明显，但是还是可以看出随着节点数目的增多，多线程的优势越来越明显，它们之间消耗的时间差别越来越大。

### 5.3.3 实验分析

随着节点数目的增多，多线程展示策略与单线程展示策略之间的时间差越来越大，也就是多线程展示策略越来越有优势，这是因为随着节点数目的增多，多线程额外消耗的上下文切换以及线程资源管理所消耗的时间变得更加微不足道，所以多线程的优势更加明显。随着节点的增多，这种优势会更加凸显。

## 5.4 人员社会关系图谱可视化加速算法应用

可视化加速算法只能在实际应用中体现其实际价值，该部分主要从广告精准推荐和社交娱乐三个方面说明其实际应用。

### 5.4.1 用户广告精准推荐

广告推荐需要将用户的兴趣爱好以及社交行为等数据收集起来，从中挖掘出隐藏的其他用户，将广告推荐给这类用户，某人的社会关系图谱如图5.5所示。

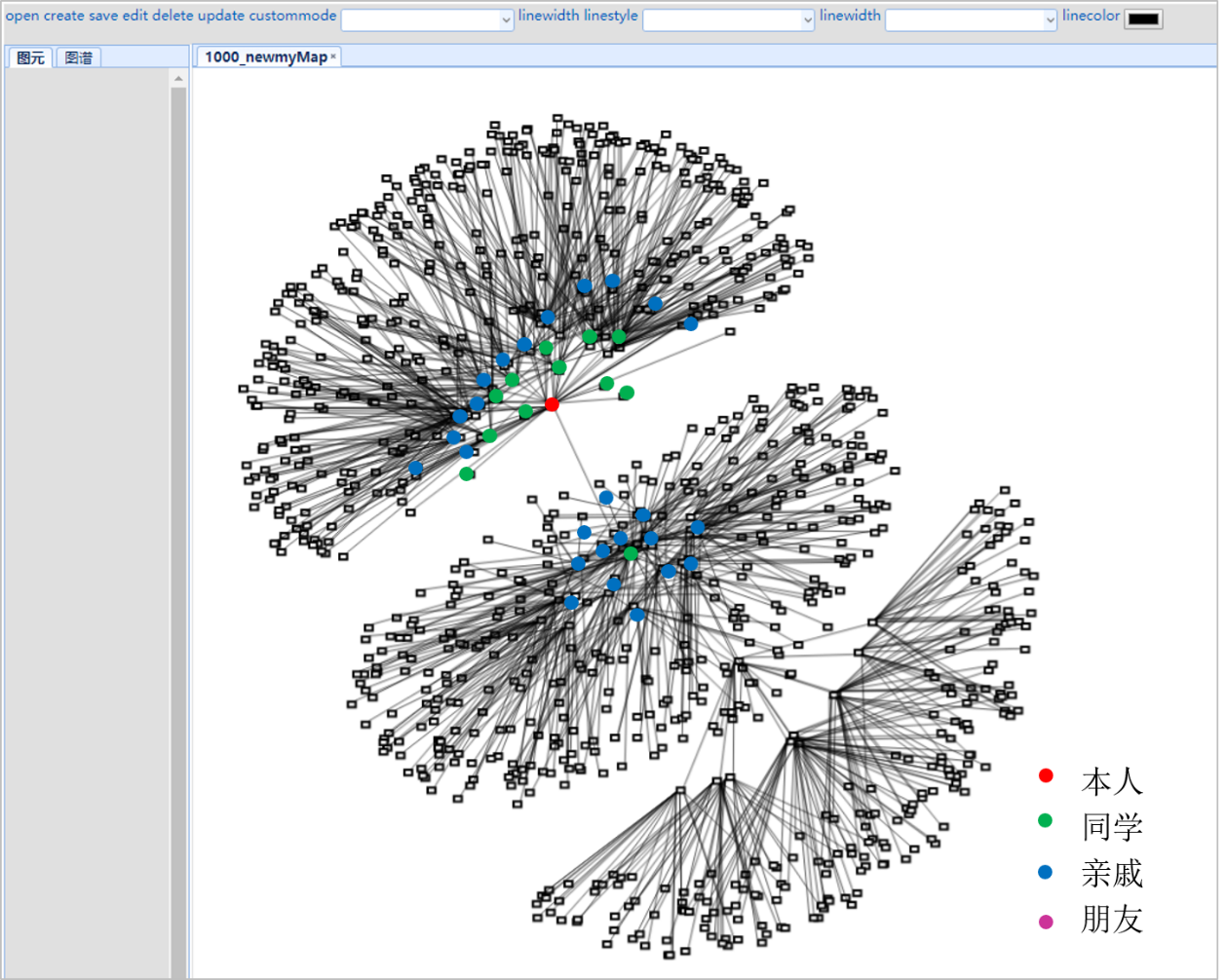
图 5.5 红色人的社会关系图谱

图5.5中红色人喜欢电子产品，则与其紧密联系的同学、同事、QQ好友和亲人（即图中的绿色人）也可能喜欢电子产品，与这些人紧密联系的人群（即图中的蓝色人）也就有很大可能喜欢电子产品，可以再次扩展人员社会关系图谱，当图谱中节点数目过多时图谱可视化就会消耗过多的时间，通过此算法优化了图谱可视化的时间，图5.5使用可视化加速算法时耗时为37ms。

### 5.4.2 社交娱乐

在社交娱乐方面社会关系图谱一直起着举足轻重的作用。谈到社交娱乐，在21世纪不得不说网络游戏，或者称之为社交游戏。网络游戏利用社会关系图谱将密切相关的人聚集起来，在游戏中促进了人与人之间的互动、交流，促进了文化的传播。比如，2016年大火的游戏“王者荣耀”，该款游戏，通过腾讯的其他软件的巨大流量，迅速的在青少年以及中年人群中快速传播。该游戏中根据用户的游戏等级，可以自动匹配与之相应的玩家，将各行各业的各个年龄段的人都聚集到一起，提供了一个互动交流平台；并且该游戏可以于用户的QQ好友、微信好友、同事、朋友和亲戚匹配，这也是基于人员社会关系图谱，如图5.6所示。

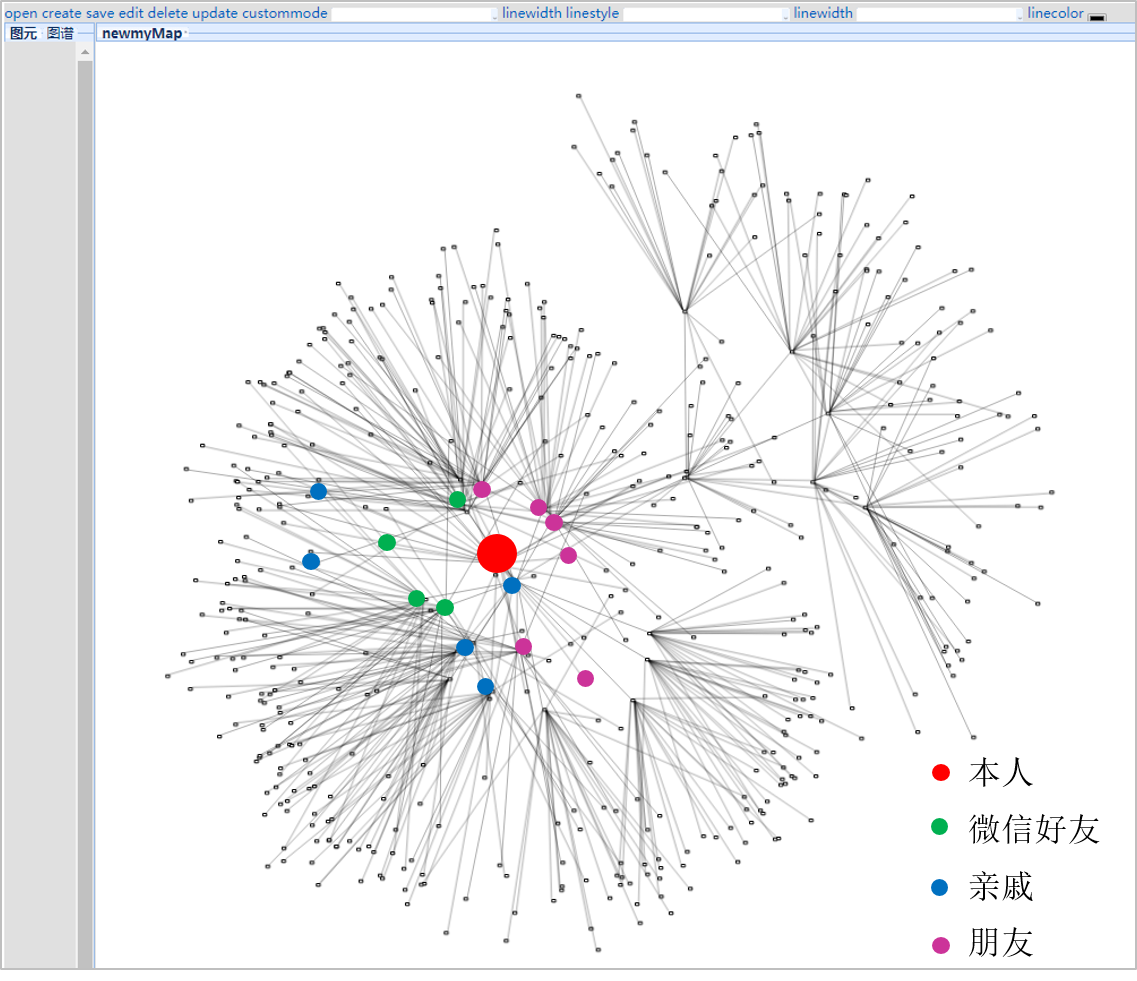


图5.6 游戏人员关系图

图5.6中红色人与绿色表示微信好友蓝色表示的亲戚以及紫色表示的朋友匹配玩游戏，随着这个关系图谱的扩展，图谱可视化消耗更多的时间，使用人员社会关系图谱可视化加速算法可以优化图谱的可视化。

## 5.5 本章小结

本章是关于基于KK图谱算法的人员社会关系图谱布局加速算法和基于多线程的人员社会关系图谱展示加速算法的验证实验,以及可视化加速算法的应用。

图谱布局加速算法的验证实验主要检验该算法的有效性和优化效果。社会人员关系图谱数据集完美的测试了图谱布局加速算法的有效性以及显示效果，500到10000个节点不同规模的图谱测试了算法的时间消耗，总体而言，图谱布局加速算法已经有较低的时间复杂度，测试证明也消耗了相对较少的时间，特别是在布局方面，具有比较好的布局效果。

图谱展示加速算法的验证实验主要是检验算法的优化效果，这个优化是与单线程展示策略相比较体现出来的。测试发现在节点数目比较少时（500个节点），单线程展示策略与多线程展示策略所消耗的时间差不多，多线程展示并没有太多优势，但是随着节点数目（7000个节点）增多时，多线程展示策略的优势开始显现，所以多线程展示策略更加适合于千量级节点图谱甚至更大规模的节点数目的图谱。

人员社会关系图谱可视化加速算法的应用主要依据是人员社会关系图谱加速算法的特征、价值以及一些重要优势，其典型应用主要分为四大部分。本章主要对社会化推荐和社会化娱乐的应用方面做了详细介绍。广告精准推荐属于社会化推荐，社交娱乐属于社会化娱乐范畴。随着互联网的发展，人员社会关系图谱可视化加速算法还会在更多领域应用。

# 结论

社交图谱可视化一直是人们关心和研究的问题，随着社交图谱图谱的增大，对社交图谱可视化的要求越来越高。为了能够将社交图谱中节点与节点之间的关系更加简单、直观、清楚的体现出来，就必须提出有种有效的方法来对人员社会关系图谱进行布局加速和展示加速，本课题致力于解决千量级图谱可视化布局加速和展示加速，通过对人员社会关系图谱可视化算法的设计与实现，取得的成果有以下几个方面：

（1）在图谱布局加速算法中，将二维空间下的KK图谱能量模型扩展到三维空间，然后基于KK图谱算法从而实现了人员社会关系图谱的布局。紧跟着又使用分层思想对布局算法进行优化，根据图谱的社区性质将图谱分为两层，第一层是抽象图谱层，该层是将通过社团发现算法将整个图谱划分为若干个子图谱，将各个子图谱构建成抽象图谱；第二层是各个子图谱层。分层之后分别使用图谱布局算法对抽象图谱层和子图谱层进行布局，从而达到加速布局的目的。另外，KK图谱布局算法本身具有时间复杂度高和容易陷入局部最优的缺陷，但是它又具有可以将图谱布局的非常合理，通过图谱节点分层可以很大程度将提高算法的执行效率，但是图谱节点分层无法将图谱布局，将KK图谱算法和分层思想结合之后不仅提高了算法的时间复杂度，还消弱了KK图谱模型容易导致局部最优的缺点，并且解决了千量级图谱中节点重叠、节点之间的边交叉、布局效率低等一些列布局问题，通过两级布局的策略很好的优化了算法。

（2）在图谱的展示加速展示算法中，主要面临图谱展示耗时的问题，使用多线程解析XML文件，并且将节点和边展示出来，这与单线程策略相比，在展示时间实现了较大幅度的提升。

# 参考文献

###### Peter Eades.A heuristic for graph drawing.[J].Congr. Numer., 1984,Vol.42,pp.149-160.

###### Tomihisa Kamada,Satoru Kawai.An algorithm for drawing general undirected graphs.[J].Information Processing Letters,1989,31(1):7-15.

###### 吴渝,林茂,雷大江.一种3D空间中的两级力导引可视化算法[J].重庆邮电大学学报,2015,27(5):668-671.

###### 吴鹏,李思昆.适于社会网络结构分析与可视化的布局算法[J].软件学报,2011,22(10):2467-2475.

###### Kozo Sugiyama,Kazuo Misue.Graph Drawing by the Magnetic Spring Model.[J].Journal of Visual Languages and Computing,1995,Vol.6(3):217-231.

###### Andreas Noack.An energy model for visual graph clustering.[J].Liotta, Giuseppe(ed.), Graph drawing.11th international symposium, GD 2003, Perugia, Italy, September 21-24.

###### Yehuda Koren,Ali Çivril.The Binary Stress Model for Graph Drawing[B]. Lecture Notes in Computer Science2009 Springer.

###### Xiaoxin Yin,awei Han,Philip S. Yu.CrossClus: user-guided multi-relational clustering[J]. Data Mining and Knowledge Discovery,2007,15:321-348.

###### Jianguo Jin.PPOS SYSTEM: A System of Partitioning Polygonal Objects[C].Information Science and Engineering,2007,42:43-65.

###### Haojun Sun,Shengrui Wang,Qingshan Jiang.FCM-Based Model Selection Algorithms for Determining the Number of Clusters[J].Pattern Recognition,2004(37):2027-2037.

###### 杨立文.基于改进的社团发现算法的社区发现技术[D].吉林.吉林大学,2012:23-25.

###### 金超.基于图聚类的社会网络数据挖掘算法研究[D].山东,山东理工大学,2017:28-31.

###### 井靖,蒋烈辉,刘铁铭,司彬彬,曾韵,朱晓清.软件逆向分析过程中的多维图谱抽取方法.[J].计算机应用与软件,2016,33(4):1-4.

###### 水超,陈涛,李慧,陈国升.基于力导向模型的网络图自动布局算法综述[J]. 计算机工程与科学期刊,2015,37(3):457-460.

###### 赵俊岚. XML编程中的DOM与SAX技术[J].计算机工程,2004,30(24):70-72.

###### 李海威,韦天瀚.基于Q函数优化的加权有向复杂网络模糊聚类算法设计研究[J].广东科技,2016,10(4):54-55.

###### 曹梦琦,李德敏,张光林,郭畅.一种时间最短的交通网络路径求解方法[J]. 计算机工程与应用,2018:2-6.

###### 王荣,江东,韩惠.基于Floyd方法的最短路径算法优化算法[J].甘肃科学学报,2012,24(4):1110-114.

###### 王顶,徐军,段存玉,吴玥瑶,孙静.基于PageRank的用户影响力评价改进算法[J].哈尔滨工业大学学报,2018,50(5):61-63.

###### 郭华东.基于线性回归与马尔科夫链相结合的云资源监控预测算法研究与实现[D]. 浙江,浙江大学,2017:17-19.

###### Tomihisa Kamada,Satoru Kawai.An algorithm for drawing general undirected graphs.[J].Information Processing Letters,1989,Vol.31(1):7-15.

###### 吴海林.基于社区划分和改进PageRank的影响力最大化算法[J].移动通信,2017,10:82-84.

###### MOORE B CJ.An introduction to the psychology of hearing[M].London:Academic Press,2003.

###### 崔俊明,李勇,李跃新.基于非加权图的大型社会网络检测算法研究[J].电子技术应用,2018,44(2):80-83.

###### Gail-Joon Ahn,Ravi Sandhu.Role-based authorization constraints specification.Acm Transactions on Information and System Security,Vol.3,No.4,2000.

# 攻读学士学位期间发表的论文和取得的科研成果

# 致谢