

# 数据结构大作业报告

## 图数据结构与路径规划的应用

Rev 1.0

姚皓天 (2013011515)

2015 年 1 月

初步文档，稍后完善。

# 目录

<b>1</b>	<b>基本原理</b>	<b>3</b>
1.1	概述 . . . . .	3
1.2	原理 . . . . .	3
1.3	界面 . . . . .	3
1.4	操作说明 . . . . .	6
1.4.1	系统设置 . . . . .	6
1.4.2	缩放 . . . . .	6
1.4.3	添加节点 . . . . .	6
1.4.4	修改、删除节点 . . . . .	6
1.4.5	修改城市 . . . . .	6
1.4.6	最短路径 . . . . .	6
1.4.7	路线推荐 . . . . .	6
<b>2</b>	<b>程序设计</b>	<b>7</b>
2.1	总体框架 . . . . .	7
2.2	图数据结构的实现 . . . . .	7
2.3	最短路径算法 . . . . .	7
2.4	路径推荐算法 . . . . .	8
2.4.1	并行文件查找 . . . . .	8
2.4.2	蚁群算法 . . . . .	8
<b>3</b>	<b>设计心得</b>	<b>8</b>
3.1	收获 . . . . .	8
3.2	特色 . . . . .	8
<b>4</b>	<b>文件清单</b>	<b>9</b>

# 1 基本原理

## 1.1 概述

本程序是由 Microsoft Visual Studio 2012 创建,目标框架为.NET Framework 4.5 , WPF 应用程序。程序第一部分实现了图结构的存储,遍历,多元最短路的规划,以及相应的可视化操作;第二部分使用蚁群算法(和并行文件查找遍历)的路线推荐功能。

## 1.2 原理

命名空间 TravelAgency.Graph 下实现了城市类和图类的数据结构封装,实现了图的存储,修改,遍历等基本操作,以及对 Floyd 算法。

命名空间 TravelAgency.ACO 下实现了蚁群算法。

命名空间 TravelAgency 下实现了基本的界面交互逻辑以及文件操作。

## 1.3 界面

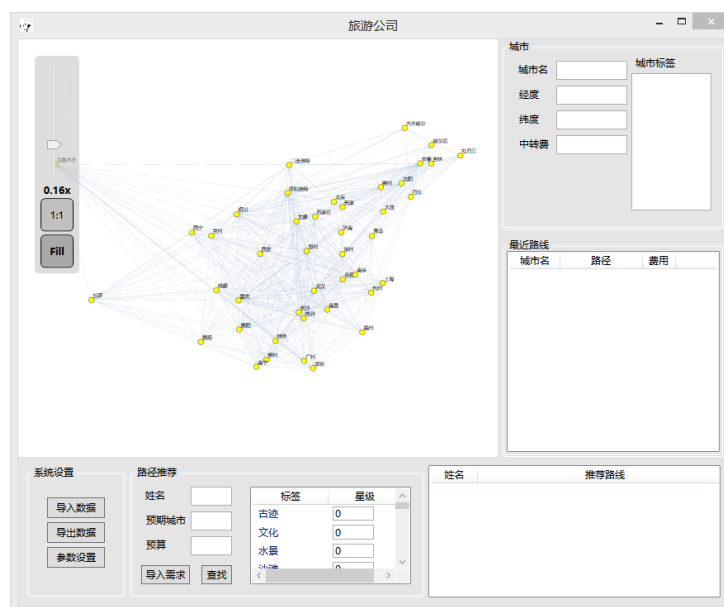


Figure 1: 主界面

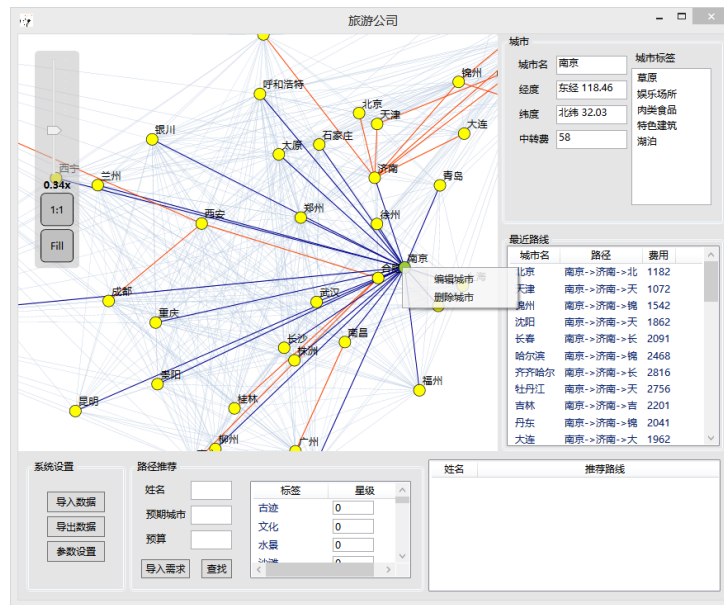


Figure 2: 修改删除节点

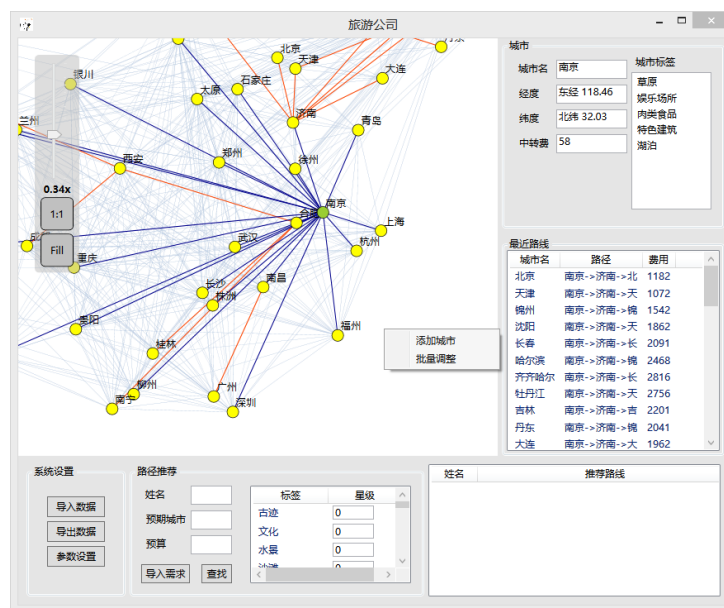


Figure 3: 添加节点



Figure 4: 编辑节点

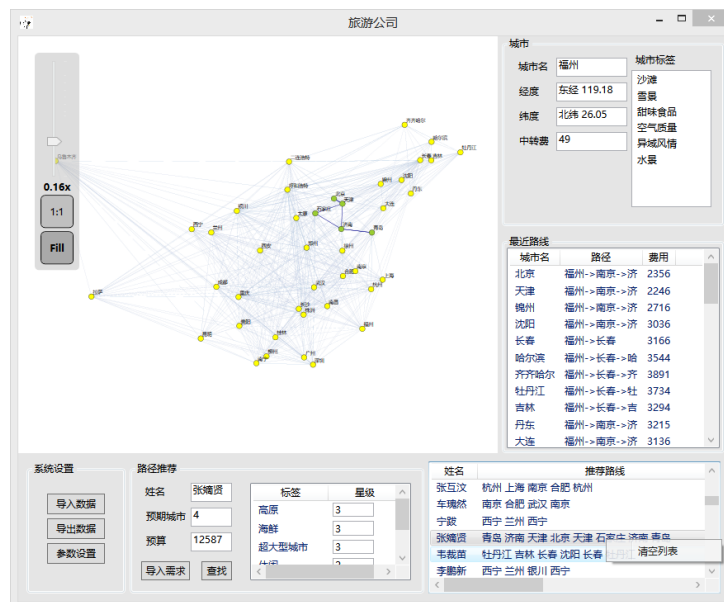


Figure 5: 路径推荐

## 1.4 操作说明

### 1.4.1 系统设置

导入按钮，可以从 Excel 文件或者地图二进制文件（.map 文件）中导入地图信息。导出按钮，可以将地图数据导出到二进制文件中。参数设置可以调整推荐算法的参数。

### 1.4.2 缩放

地图区域提供缩放控件，可以使用缩放功能实现对地图的调整。

### 1.4.3 添加节点

在空白处点击右键可以打开上下文菜单，将弹出添加节点的对话框。对话框初始经纬度就是鼠标点击位置对应的地理坐标。

### 1.4.4 修改、删除节点

右键点击城市，上下文菜单提供修改和删除的选项。

### 1.4.5 修改城市

修改城市信息将在弹出的对话框中完成，对话框中可以添加路径，维护城市标签。

### 1.4.6 最短路径

点击城市，将显示出到达其他所有城市的最优路径，并且直达和非直达路径使用不同的颜色标出。点击城市的同时，右侧将列出城市的详细信息。在右侧路径列表中双击路径，对应的路径将在地图中高亮。

### 1.4.7 路线推荐

在下方填入需求，将会给出推荐路径，双击路径，对应的路径将在地图中高亮。同时在路径列表中右键可以将列表清空。

## 2 程序设计

### 2.1 总体框架

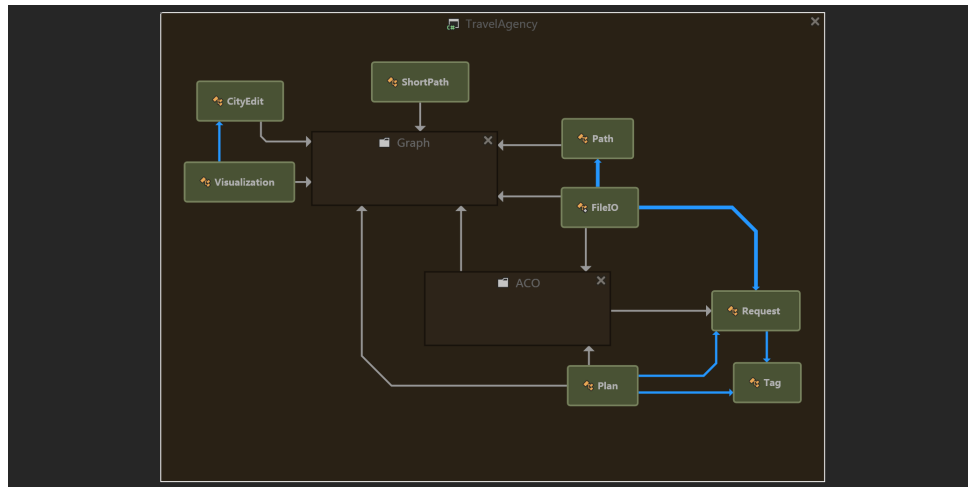


Figure 6: 总体框架

### 2.2 图数据结构的实现

各类的依赖关系如下图：

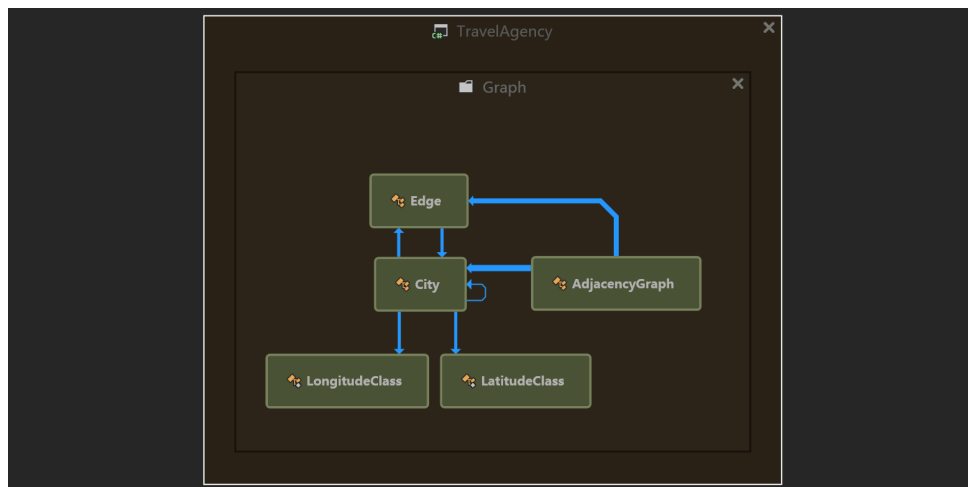


Figure 7: 图类依赖关系

### 2.3 最短路径算法

城市间最短路径的计算采用了 Floyd 算法。



## 2.4 路径推荐算法

### 2.4.1 并行文件查找

在路径推荐过程中，我首先使用了遍历的方法，遍历所有可能的路径，并将每条路径，具有的标签和总费用保存在文件中。文件按照一定的大小分成区块。得到的一系列路径文件保存在以出发城市名的文件夹中。

当用户输入需求的时候，将采用并行的方式，对每个区块的进行处理。计算评估函数的值，最后进行归并，找出最优值。

但是由于城市数量较多，在路径长度大于 5 的情况下，将会有大量的文件需要处理，运算时间过长。

### 2.4.2 蚁群算法

参考文章[http://blog.sina.com.cn/s/blog\\_6a409d8701011wr8.html](http://blog.sina.com.cn/s/blog_6a409d8701011wr8.html)

实现了蚁群算法，可以较快的给出较优的解。

但是由于算法的固有特点，每次所得路径可能会不一致，并且不能保证取到最优解。各类的依赖关系如下图：

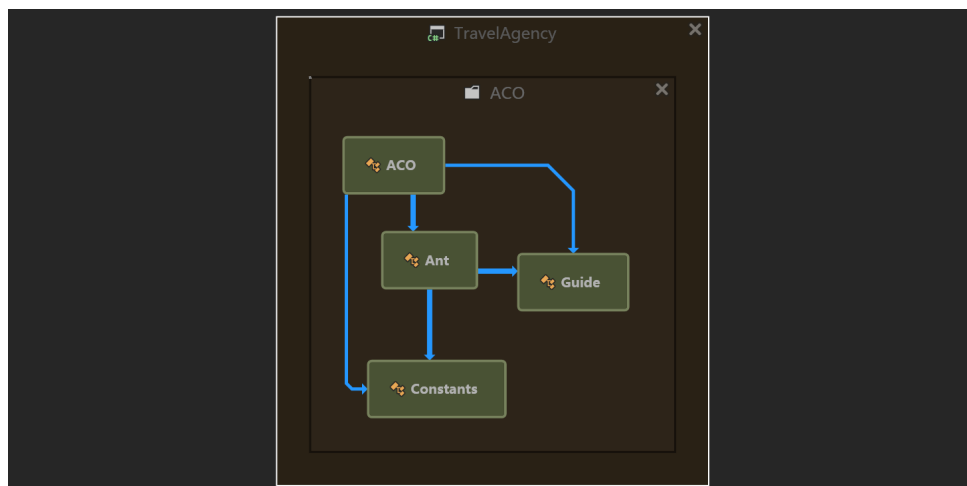


Figure 8: 蚁群算法类依赖关系

## 3 设计心得

### 3.1 收获

这次熟悉了 WPF 框架下的界面开发，体验了一个较复杂程序的开发。

## 4 文件清单

- src\ 工程文件
- bin\ 可执行文件

- doc\ 文档