

同濟大學

TONGJI UNIVERSITY



2016 年同济大学数学建模校内竞赛论文

选择题目：同济大学 2016 年数学建模竞赛 B 题游乐园客流疏导方案

姓名	学号	专业	联系方式
----	----	----	------

队员 1			
------	--	--	--

队员 2			
------	--	--	--

队员 3			
------	--	--	--

基于深度优先搜索和网络流的游乐园客流疏导方案

基于 spss 的皇冠假日酒店预定量分析

摘 要

游乐园客流疏导方案问题根据客流量不同分为两种情况讨论,客流量较少的情况下,建立“最优路线”模型,已知游客希望游览的项目点和目前所在位置,以路程或时间作为代价,规划出一条最小代价的游览路线,该模型数学上属于图论问题中的“旅行商问题”,难点在于时间作为代价时代价量的处理。由于项目点数较少,求解时可以用枚举法的思想,利用深度优先搜索的算法进行,这样一定会得出准确解。论文中给出了利用 Matlab 编写的程序和测试结果,规划出了指定项目点下的最优路线,并对结果进行了分析。

客流量较多的情况下,路程作为代价的模型显然不符合实际,时间作为代价的模型由于出现了等候轮次的问题,每个游客的最优线路会互相影响,难以用上述模型解决。因此从宏观的角度考虑,客流量较多时,游乐园更关心的问题是如何分配游客使游乐场各项目游客分布均匀。为此引入热量传递模型,将人抽象成热量,每个游乐项目根据其每场容纳游客数、每场持续时间不同有特定的比热容。游乐项目依据不同比热容容纳热量后产生相应温度,不难发现,最优的疏导方案应满足所有项目都达到相同的均衡温度。同时,热量在游乐项目间传递会产生代价,目标就是使每个游乐项目达到均衡温度的同时,产生的总代价最小。

于是最优人群疏导问题就转化成了最优热量传递问题,可以建立网络流模型并使用最小费用最大流算法求解出最优疏导方案。另外,考虑到游乐项目结束后的分配问题、人们偏好等问题,对模型进行了完善,使模型更加与现实情况吻合。最后,对应每种情况,使用不同测试数据对该模型进行了测试,分析比较测试结果并总结出该模型的优缺点。

皇冠假日预定量分析的已知条件仅有一份 2015 年全年预定数据,这道题的关键在于对已知数据的全面解读。在明确“预定量”的定义、将现有数据转化为每日预定量数值的前提下,对基础数据进行筛选:甄别数据质量,检查并剔除因数据残缺等问题导致的无效数据。

根据游乐场主要消费群体、消费行为特征,引入定性变量并赋值。使用 spss 软件将定性变量与因变量进行多元线性回归模型的拟合,在反复拟合的过程中,通过对各因变量的筛选、调整,提高各系数的显著性,最终达到尽可能合理的拟合结果。

面对定性变量带来的同一性质内的个性特征缺失等问题,根据现有数据的差异性(标准差等)对拟合结果进行离散处理,进一步完善该数学模型,使得 2016 年 1-3 月预定量预测更具备参考价值。

关键词: 深度优先搜索、旅行商问题、网络流、热量传递模型、spss、定性变量、离散处理。

一、问题重述

Youth 游乐园作为本市建有最多过山车的游乐园即将盛大开园，受到了青少年的热捧。预计届时园区将迎来每天 1 万的大客流。因此如何根据客流情况，及时分流人群，舒缓园内客流分布，为顾客提供游园线路引导，保障游客的游园体验是十分重要的。

同时皇冠假日酒店是游乐园内的酒店，目前已开业，为有需要住宿的游客提供了便利，鉴于园区内客流量较大，根据该酒店历史预订信息预测未来预定量也是保障酒店良好运营的重要手段之一。

为了解决上述两个实际问题，需要建立数学模型分析研究下面的问题：

(1) 已知 Youth 乐园的规划图，共计 A-J 十个项目点，游客可以按照图中路线往返下一个游乐项目，并且已知每个游乐项目的容纳游客量和时间安排表。要求保障每位游客体验游乐设施的前提下，建立对每个游乐项目的等候游客进行游览疏导和提醒的模型，进行分析求解后，使游园体验最优。

(2) 已知皇冠假日酒店 2015 年的全年预订数据，需要根据预定信息，综合考虑影响房间预定量的主要因素（例如季节，周末，节假日，暑期等）建立数学模型。并利用该数学模型预估 2016 年 1 月至 3 月每天的预定房间数。

二、问题分析

2.1 Youth 游乐园客流疏导方案分析

已知信息为游乐园中各项目间的路程，各项目所容纳的人数和持续时间，所要求的问题是如何根据客流量的不同对于每个项目的等候游客进行游览疏导和提醒。根据已知信息，首先有一点是值得确定的，客流量不同的时候对于游客的疏导方案是不同的，因此也需要规划不同的方案。

在客流量较少的时候，例如早晨刚刚开园或者晚间即将闭园的时候，各游乐项目是无需等待的，也就是说针对一个游客从一个游乐项目结束到另一个游乐项目开始的时间是可以唯一确定的，耗费时间不会因其他游客的加入而被延长。假设游客不会玩一个项目多遍，客流较少的时候主要需要解决的问题是对于一个游客来说如何规划最优路线，这条最优路线可以是所行走路程最短也可以是所耗费时间最短。

如上所述，对于客流量较少的时候，客流疏导问题被大大简化了。路程最短的问题，实际上就是“旅行商问题”，这里的代价是路程，问题即是如何以最小的代价遍历所有的游乐设施，有多种方法可解；时间最短的问题，是对于路程最短的问题的一个扩充，代价变成了时间，在到达一个游乐项目时通向其他游乐项目的时间代价发生了改变，需要重新更新后计算。鉴于项目点数较少，解决上述问题都可使用深度优先搜索的方法进行。

在客流量较大的时候，也是游乐园里比较通常的情况，各游乐项目是需要等待的，而且等待的时间可能会较长，这时解决的问题变得复杂起来，如果考虑一个游客应该走的最小代价路线，游客通常会考虑代价为时间，由于时间代价会受

到其他游客选择的影响，导致该问题十分复杂无法实现。

鉴于上述问题，客流量较大的情况时不能再以客流量较少的情况下规划最优路线来解决。宏观来看，客流量较大时游乐园人员调控决策者考虑的是整体园区的规划，他们的目的应该是使游乐园的游客相对均衡的分布，避免出现局部游客过多的现象。在这种目标下，只需要做的就是规划出一条代价最小的调控方案，使游客多的项目的人到游客少的项目去，达到游乐园游客分布均衡。求解该模型的方法可以使用网络流的算法。

2.2 皇冠假日酒店预定房间量分析

尽管提供数据不涉及客户个体信息，而与预定时间相关，因而影响预订房间量的因素主要取决于入住日期所具备的特性；但同时，游乐场主要消费人群的群体性偏好又对于入住日期的选择十分关键。因此，了解游乐场消费的主要群体，分析消费行为特征，是选取因变量的前提。

根据阿里旅行对 2015 年几大亚太主题乐园共同出行人数的统计，32% 是 25~29 岁的用户，32% 为来自 30~35 岁的用户。也就是说，已经经济独立的 80 后成了主题乐园产品的主力购买人群。这类人群主要由情侣、友人、年轻父母构成。情侣、友人多选择节假日出行，其中包括法定节假日以及非法定节假日，包括圣诞节、情人节、万圣节等，主题乐园往往也会借机举办相应主题活动，吸引更多的客流。年轻的父母要兼顾孩子的情况——有寒暑假，因此在选择的过程中需考虑季节因素、是否接近暑期等因素。

三、模型基本假设

3.1 Youth 游乐园客流疏导模型基本假设

3.1.1 客流量较少的情况

- ①假设游客一个项目只会游览一遍
- ②假设游客游览完所有项目后最后会回到出入口
- ③假设模拟开始时游客所在的位置一定是某个项目点或出入口处
- ④代价为路程的情况时，假设代价和路程呈正比关系，且比例系数为 1
- ⑤代价为时间的情况时，假设游客到一个项目后不考虑等候轮次，只需等待到项目开始即可
- ⑥代价为时间的情况时，假设游客的行走速度是匀速的

3.1.2 客流量较多的情况

- ①假设代价和路程呈正比关系，且比例系数为 1
- ②为了简化运算量，假设走在路上的人聚集在道路中点处，在此基础上进行疏导
- ③假设游客会到预计疏导到各项目点
- ④假设不考虑游客去重复的项目点

3.2 皇冠假日酒店预定房间量模型基本假设

假设酒店预订房间量与五个变量呈多元线性关系，分别为：

- ①入住日期在哪一季度
- ②入住日期是否为周末
- ③入住日期是否为法定节假日
- ④入住日期是否为暑期（7月、8月）
- ⑤入住日期是否为特定节日（非法定假日）

四、符号说明

4.1 Youth 游乐园客流疏导模型符号说明

4.1.1 客流量较少情况——最优路线模型

- Q_z ——园内总客流量
- Q_l ——园内临界客流量
- n_i ——项目*i*的每场容纳游客数
- n ——游玩项目和出入口个数的和，对于本题为常量 11
- G_{ij} ——*i*项目到*j*项目所耗费代价
- D_{minij} ——*i*项目到*j*项目的最短路程
- k ——代价和最短路程呈正比关系时的比例系数
- D_{ij} ——*i*项目到*j*项目的路程
- x_{ij} ——*i*项目到*j*项目的最短路径有无走过，走过为 1，没走过为 0
- v_m ——游客行走平均速度，不妨设为1.5m/s
- Δt_j ——游客在 *j* 项目等待项目开始花费时间
- ts_j ——*j* 项目运行时间

4.1.2 客流量最多情况——热量传导模型

部分符号定义已经包含在 4.1.1 中。

- Q_i ——项目 *i* 的当前热量，代表项目 *i* 的当前客流量
- c_i ——项目 *i* 的比热容，代表项目 *i* 的容纳能力
- c_z ——所有项目总比热容，代表整个游乐园总体容纳能力
- T_i ——每个项目经过容纳能力调整后的统一状态参量，即初始温度
- T^* ——均衡温度
- Q_i^* ——项目 *i* 的达到均衡温度时的热量
- dT_i ——项目 *i* 的偏好修正系数
- $T_i^{*'}$ ——考虑偏好的均衡温度
- G_z ——热量传递的总代价
- $V(f)$ ——网络流最大可行流

C ——网络流弧上容量集
 E ——网络流有向边集
 S ——网络流源点
 T ——网络流汇点

4.2 皇冠假日酒店预定房间量模型符号说明

Q ——预订房间量
 C ——回归模型常数项
 J ——季度变量赋值
 W ——周末变量赋值
 H ——法定假日变量赋值
 S ——暑期变量赋值
 F ——特定假日变量赋值

 B_J ——季度变量参数

 B_W ——周末变量参数
 B_h ——法定假日变量参数
 B_s ——暑期变量参数
 B_f ——特定假日变量参数

五、模型的建立与求解

5.1 Youth 游乐园客流疏导模型建立与求解

5.1.1 临界客流量 Q_l

如同问题分析和假设描述，在客流量较少的情况下，游客到了每个项目后不考虑等候轮次，只需等待到项目开始即可，因为园内客流不大，疏导的方案即是一个游客的最优游玩方案。客流量较大的情况下，则需要考虑相对复杂的因素，利用宏观调控的思想对于园内客流进行规划，减轻游乐园客流分布。

首先要确定的就是多大客流量时用哪种方案，即临界客流量 Q_l 是多少。因为客流量较少以时间为代价时建模方法中假定游客到一个项目无需排队，所以临界客流量 Q_l 即每个项目人数达到饱和，如果不考虑走在路上的人，一种最极端的情况就是在某一时刻所有的项目都在进行，且没有人等候，这时临界客流量 $Q_l = 2 \times \sum_{i=1}^n n_i = 1580$ 。考虑路上有一部分人，实际情况一定比上述结果多，取临界客流量 Q_l 为 2000 人，当园内总客流量 $Q_z < Q_l$ 时，采用 5.1.3 的建模和求解方法；当 $Q_z \geq Q_l$ 时，采用 5.1.4 的建模和求解方法。

5.1.2 代价 G_{ij} 的确定

关于代价的确定，5.1.3.1 和 5.1.4 中的代价以路程标定，当以路程作为代价时， $G_{ij} = f(D_{minij})$ ， D_{minij} 为 i 项目到 j 项目的最短路径，首先该函数应该是单调

递增函数。

①如果游客对于路程的敏感度是不变的,那么上述函数关系应该是正比关系, $f'(D_{minij}) = k$,不妨设比例系数为 1。

②如果游客对于路程的敏感度是递增的,也就是随着路程越长,敏感度越高, $f'(D_{minij})$ 应该是单调递增函数。

③如果具有游乐园历史道路客流数据,还可以统计出来各个道路的拥挤程度,这也可以反映在代价 G_{ij} 中,设置一个拥挤程度因子 k_y , $G_{ij} = f(D_{minij}, k_y)$,对于每个道路 k_y 都有不同的数值。

因为本题 G_{ij} 不是关键问题,为了简化考虑,以路程标定代价时全部以①情况考虑。另外,5.1.3.2 中的代价 G_{ij} 是以时间标定的,此代价的确定在 5.1.3.2 中还会有详细描述。

5.1.3 客流量较少情况模型建立和求解——基于深度优先搜索的最优路线模型

客流量较少的情况下,即 $Q_z < Q_l$,疏导和提醒方案为游客的最优游玩方案。值得注意的一点是,模型考虑了游客已经游览过了若干项目且目前在某个项目点处,在这个基础上再去希望浏览的项目;如果没有游览任何项目,就代表游客在出入口处,并已知其希望浏览的项目,规划出最优游玩方案。另外还应该注意的,模型中最后一个到达的点一定是出入口,这也是符合实际情况的,人们总是游览完毕后最后到达出口。

该模型的建立是基于“旅行商问题”^[1],“最优游玩方案”是指从图G的所有游览路线中取得代价最小的游览路线。对于代价 G_{ij} ,考虑为两种形式,一是行走的路程,二是花费的时间。

5.1.3.1 代价为路程的情况

路程作为代价时,代表着游客只关心自己走一条路程短的路线,在这种情况下,代价 G_{ij} 就是项目 i 到项目 j 的最短路程 D_{minij} , $G_{ij} = D_{minij}$,而且在任意项目到下一项目时,这个代价是不会变化且唯一确定的。

关于代价 G_{ij} 即最短路径 D_{minij} 的求解方法,采用 Floyd 算法^[2],又称插点法,可以寻找给定加权图G中多源点之间的最短路径。其时间复杂度是 $O(n^3)$,本题中将出入口算在内, $n = 11$,显然效率是可以接受的。

该算法的状态转移方程如下:

$$D_{minij} = \min(D_{ik} + D_{kj}, D_{ij})$$

如果 i 项目到 j 项目无法连通,则两者间路程 D_{ij} 为无穷;由于模型假设不重

复游玩，因此 D_{ii} 也设为无穷。经过动态规划后可以得到最短路径 $D_{\min ij}$ 的矩阵，在路程为代价的情况下即是代价 G_{ij} 。容易得知该矩阵为对称矩阵，下面是计算得出的代价矩阵 G_{ij} ，矩阵阶数为 11，第一点为出入口，从第二点开始为项目A, B …。

$$\begin{pmatrix} \infty & 300 & 400 & 700 & 1150 & 650 & 1650 & 1200 & 1600 & 1100 & 550 \\ - & \infty & 300 & 600 & 1050 & 350 & 1550 & 900 & 1300 & 600 & 250 \\ - & - & \infty & 300 & 750 & 350 & 1250 & 900 & 1300 & 800 & 550 \\ - & - & - & \infty & 450 & 500 & 950 & 1050 & 1450 & 950 & 850 \\ - & - & - & - & \infty & 950 & 500 & 1150 & 1550 & 1400 & 1300 \\ - & - & - & - & - & \infty & 1200 & 550 & 900 & 450 & 600 \\ - & - & - & - & - & - & \infty & 650 & 1050 & 1500 & 1800 \\ - & - & - & - & - & - & - & \infty & 400 & 850 & 1150 \\ - & - & - & - & - & - & - & - & \infty & 450 & 800 \\ - & - & - & - & - & - & - & - & - & \infty & 350 \\ - & - & - & - & - & - & - & - & - & - & \infty \end{pmatrix}$$

接下来需要找到一条代价最小的路径，使游客游览完所有希望去的项目，参照“旅行商问题”，这个问题实际上就是在赋权完全图中，寻找最小权的 Hamilton 圈。其数学表达如下：^{[3][4]}

$$\begin{aligned} \min & \sum_{i \neq j} G_{ij} x_{ij} \quad x_{ij} \in \{0,1\} \quad i, j = 1, 2, \dots, n \quad i \neq j \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = 1, i = 1, 2, \dots, n \quad (\text{每个点只有一条边出去}) \\ & \sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n \quad (\text{每个点只有一条边进去}) \end{aligned}$$

因为 G_{ij} 的阶数为 11，使用枚举思想解决此问题，考虑最不利情况，游客希望去所有的项目点。从入口点出发的游玩线路数目一共有 $(n-1)!$ ，即等于除初始结点外的 $(n-1)$ 个结点的排列数，如果通过枚举所有路线，其计算时间为 $O(n!)=39916800$ 。计算效率刚好在临界点，如果加以优化，使用枚举思想解决该问题是可行的。

基于枚举法的思想，使用深度优先搜索的算法来实现，其基本原理如下：^[5]

①访问项目点 x 。

②依次从 x 的未被访问的项目点出发，对图 G 进行深度优先遍历，直至图中和 x 有路径相通的顶点都被访问。

③若此时图中尚有顶点未被访问，则从一个未被访问的顶点出发，重新进行深度优先遍历，直到图中所有顶点均被访问过为止。

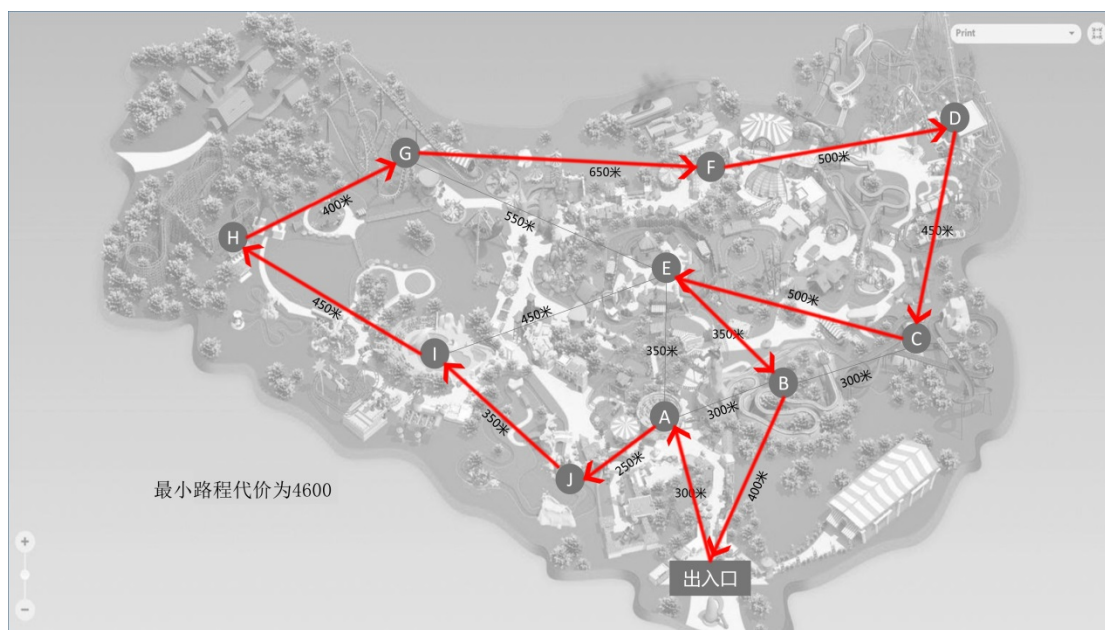
算法实现时，搜索时状态量为 x, pd, w, a, k 。 x 为当前访问项目， pd 存储所有项目点有无被访问过， w 为当前耗费代价， a 存储访问项目先后顺序， k 为已经访问项目的数目。这些状态量在搜索过程中实时被记录下来，采用深度优先的递归方法遍历图 G 中所有点，找到一条恰好访问要求项目点且代价最小的路线。

为了减少时间复杂度，在搜索过程中加以优化，如果当前耗费代价已经大于存储过的最小代价，那么这条路线直接放弃，实践检验这条优化大大减少了搜索量，从大概两小时的运算量减少到了小于一分钟。

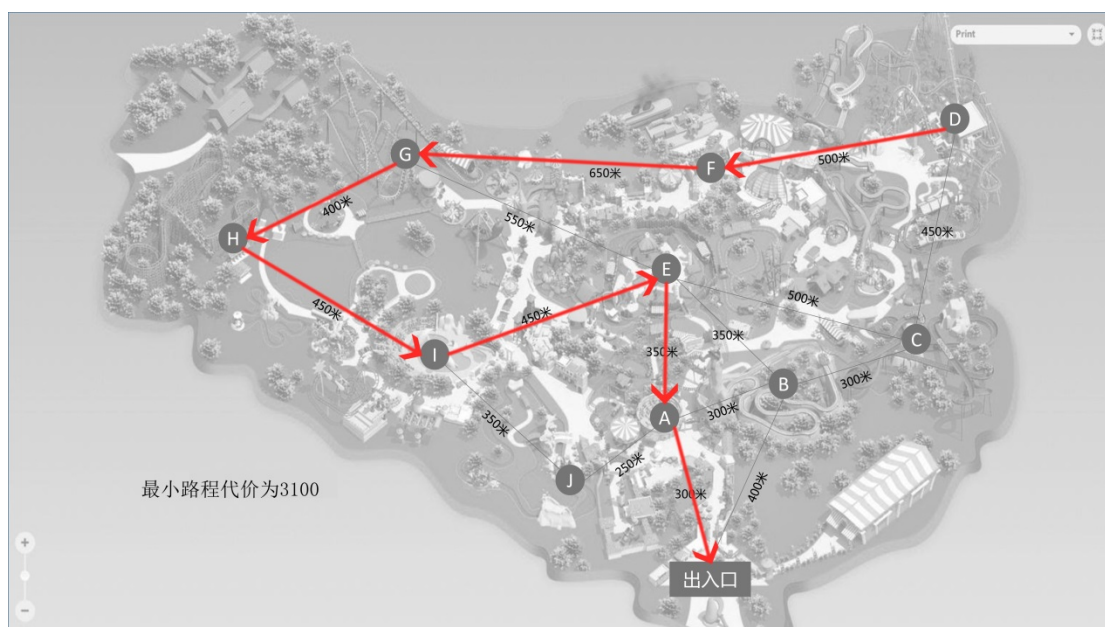
另外值得一提的是，算法设定最后一个遍历点是出入口，并且能够实现输入已经去过的项目点和希望未来去的项目点，并规划出最优路线。

代码详见附录，下面进行模型的测试。

①游客刚入园，希望得到一条游览所有项目的最小路程代价路线。



②游客已经游览完毕BCD项目，现在在D项目点，希望得到一条除去J项目的最小路程代价路线。



5.1.3.2 代价为时间的情况

代价为路程的模型中只是考虑了一个最短的路径长度，在实际情况中人们虽

然趋向于走最短路线,但是对于游乐场的游览方案人们更关心的是花了多少时间。有时走的是最短路线但是花费时间可能会很长,因为有可能很不巧刚刚走到游乐项目就开始了,如果还要遵循最短路线,那么需要等待游乐项目开始才可以。

为了更切合实际情况,需要考虑时间为代价,这时代价 G_{ij} 需要重新定义,

$G_{ij} = D_{minij}/v_m + \Delta t_j + ts_j$,即代价等于*i*到*j*的最短路径所耗费时间,在*j*项目等待开始所耗费时间,*j*项目游玩花费时间之和,其中不妨设 $v_m = 1.5m/s$ 。

值得注意的一点是, G_{ij} 是一直在变动的,因为受到 Δt_j 这一项的影响,到达后等候时间取决于何时到达,何时到达又和之前选择的路线有关。不过 Δt_j 一定可以在到达一点后唯一确定,该模型是在客流量较小的情况下建立的,假设中规定所有游乐项目到达后只需要等待开始,不考虑等待轮次的问题。

这样其实在 5.1.3.1 代价为路程的情况的模型的基础上加入时间状态量就可以实现,最后所求的是满足规定游览项目的最小时间花费路线。在每次搜索时需要重新求 G_{ij} ,本质思想还是枚举法,建模形式和路程为代价的情况相同,只不过考虑了 G_{ij} 的动态变化。

为了确定 Δt_j 这一项,需要给出所有项目的开始时间,不妨设该游乐园早晨 8 点将所有项目启动,每个项目以其运行时间 ts_j 为运转周期,周而复始的进行。这样可以得出每个项目开始时间的时刻表,在搜索过程中,就用该时刻表计算 Δt_j ,这张表的示例如下。

开始时间	第一次	第二次	第三次
A	8:00	8:33	9:06
B	8:00	8:01:15	8:02:30
C	8:00	8:02:30	8:05
.....	

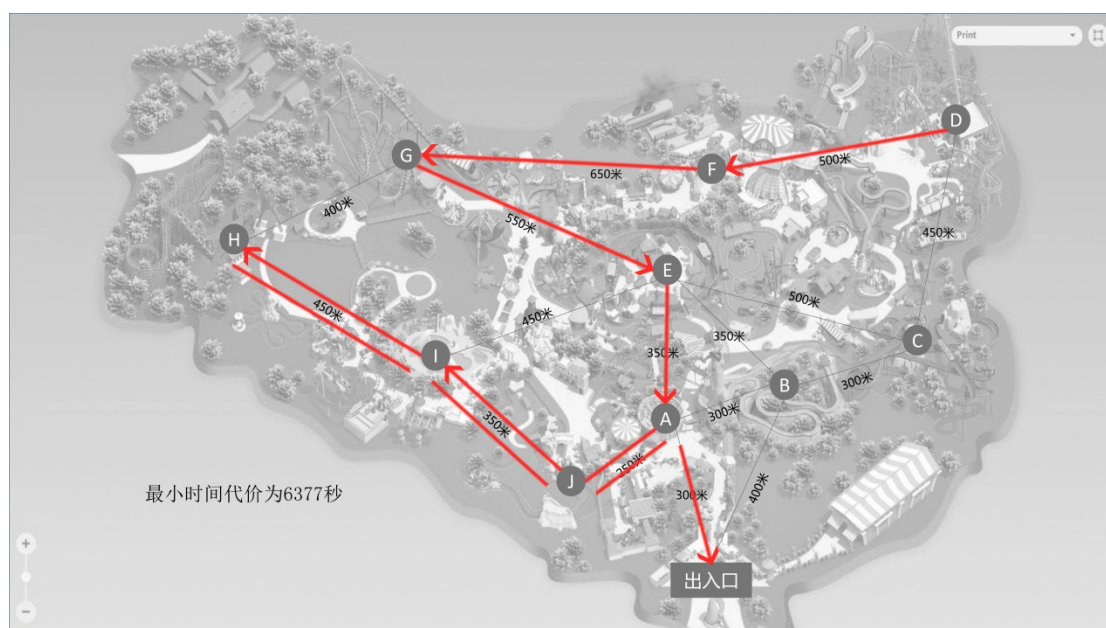
在深度优先搜索算法实现时,状态量变为了 x, pd, t, a, k, dt ,之前的 w 当前路程花费改为了 t 当前时刻,加入新的状态量 dt ,即是当前项目的等候时间,其他状态量不变。在程序输入需要输入当前时刻,算法原理和 5.1.3.1 中的相同。

代码详见附录,下面也给出测试数据。

①游客刚入园在出入口处,当前时刻为 8:00,希望得到一条游览所有项目的最小时间代价路线。



②游客已经游览完毕BCD项目，当前时刻为 9:00，现在在D项目点，希望得到一条除去J项目的最小路程代价路线。



③游客已经游览完毕BCD项目，当前时刻为 9:10，现在在D项目点，希望得到一条除去J项目的最小路程代价路线。



5.1.3.3 两种代价的测试结果对比

显而易见，代价为路程时的结果和代价为时间的结果，在同样的条件下如 5.1.3.1 和 5.1.3.2 的①结果，都是没有去过任何项目，当前在出入口，计划游览完毕所有项目。

代价为路程时：游览路线为出入口 $\rightarrow A \rightarrow J \rightarrow I \rightarrow H \rightarrow G \rightarrow F \rightarrow D \rightarrow C \rightarrow E \rightarrow B \rightarrow$ 出入口，最小路程为 4600m。

代价为时间时（入园时间为 8:00）：游览路线为出入口 $\rightarrow B \rightarrow C \rightarrow D \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow J \rightarrow A \rightarrow E \rightarrow$ 出入口，最小时间为 7033s。

可见因为所取的代价量不同，二者选择的路线是不同的，在实际情况中，可以根据个人的偏好来选择。

该模型还可以实现如果已经游玩了一些项目，还想再游玩特定的几个项目，按照最小路程或着最小时间来选择路线。例如②的结果。

游客已经游览完毕BCD项目，现在在D项目点，希望得到一条除去J项目的最小路程或时间代价路线。

代价为路程时：游览路线为 $D \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow E \rightarrow A \rightarrow$ 出入口，最小路程为 3100m。

代价为时间时（当前时间为 9:00）：游览路线为 $D \rightarrow F \rightarrow G \rightarrow E \rightarrow A \rightarrow I \rightarrow H \rightarrow$ 出入口，最小时间为 6377s。

同样也反映了代价选择不同，最优路线不同的问题。

另外，在不同时间时，所选取的时间最小路线也不同例如 5.1.3.2 中③的结果，同样是已经游览完毕BCD项目，现在在D项目点，希望得到一条除去J项目的最小时间代价路线，但是当前时间为 9:10，经过求解，此路线变为了 $D \rightarrow F \rightarrow E \rightarrow G \rightarrow H \rightarrow I \rightarrow A \rightarrow$ 出入口，最小时间为 5900s，确实反映了实际情况。

至此该模型解决了在客流量较少时游客的最优线路规划问题。

5.1.4 客流量较多的情况模型的建立和求解——基于网络流的热量传递模型

5.1.4.1 热量传递模型的建立

游园人数达到一定数量后,由于每个游乐项目有每场容纳游客数的限制,按照 5.1.3 给出的最佳游览路线进行游览,并不一定能达到最佳的游览体验,因为客流量较多时每个项目排队等候人较多,而 5.1.3 的模型中都不考虑排队等候轮次的人的影响。

如果在客流量较多的情况下还考虑一个人的最优路线，这是一个十分复杂的问题，因为其他游客的选择会互相影响，难以求出一个最优解。而且在客流量较多的情况下，宏观地考虑，游乐园管理者希望每个游乐项目根据其容纳能力不同，分配的游客尽量平均，目的主要是为了“分流”，使游乐园每个项目的客流在一个比较均衡的状态。

为此引入经典热力学模型, $Q = cm\Delta T$, 不妨 $m = 1$ 。设将每位游客抽象成1单位的热量, Q_i 代表项目*i*的当前人数。游客的到来会使游乐项目*i*得到热量 Q_i , 游客的离开会使游乐项目*i*失去热量 Q_i 。

定义每个项目的比热容为 c_i ，代表着每个项目的容纳能力。 c_i 与每场容纳游客数 n_i ，持续时间 ts_i 有关。其公式定义为 $c_i = \frac{n_i}{ts_i}$ 。与每场容纳游客数呈正比，与持续时间呈反比。

定义项目温度 $T_i = \frac{Q_i}{c_i}$ 。代表每个项目经过容纳能力调整后的统一状态参量，反映了该项目目前的“拥挤程度”。不难发现，游园体验达到最优状态时，每个游览项目都应达到均衡温度 T^* （下文会进行进一步讨论）。这时， $T^* = \frac{Q_Z}{c_Z}$ ，其中 $Q_Z = \sum Q_i$ ， $c_Z = \sum c_i$ 。

考虑到游客可能并不全部集中在游乐项目处, 每条路也应存在热量 Q_i , 为方便起见, 不妨假设其热量集中在道路中点。将路上的人建立新的节点的游乐场示意图如下:



游览提醒与疏导的过程可以抽象为热量传递的过程,每个游乐项目通过热量流动达到均衡温度 T^* 。热量流动会产生代价,单位热量 Q 从 i 点流动到 j 点产生的

代价使用 5.1.3.1 中路程作为代价的形式, $G_{ij} = D_{minij}$, 其中 D_{minij} 为 i 点到 j 点的最短距离。

于是, 大客流量游客疏导问题转化成了寻找最优热量传递方案的问题, 即在每个游览项目达到均衡温度 T^* 的前提下 (即满足游乐园内各项目客流均衡), 最小化总热量流动代价 (达到均衡的总代价最小)

$$G_z = \min \sum Q_{ij} \times G_{ij}$$

$$\text{s.t. } T_i = T^*$$

其中, Q_{ij} 为游乐项目 i 传递到游乐项目 j 的热量

G_{ij} 为单位热量 Q 从 i 点流动到 j 点产生的代价

5.1.4.2 热量传递模型的求解——基于网络流求解

采用最小费用最大流模型求解。

最小费用最大流算法是图论中的一种理论和方法, 研究网络上的一类最优化问题。所谓网络或容量网络指的是一个连通的赋权有向图 $G(V, E, C)$, 其中 V 是该图的顶点集, E 是有向边 (即弧) 集, C 是弧上的容量集。此外顶点集中包括唯一源点 S 和唯一汇点 T 。网络上的流就是由源点流向汇点的可行流, 这是定义在网络上的非负函数, 它一方面不能超过有向边容量的限制, 另一方面除去源点和汇点以外, 在所有中途点要求保持流入量和流出量平衡。 G_{ij} 表示通过弧 ij 单位流量的费用, 所谓最小费用最大流算法, 即在满足 S 到 T 的流量为最大可行流 $V(f)$ 的前提下, 总运输费用最小。

最小费用最大流问题可用如下的线性规划问题描述^[7]:

$$G_z = \min \sum_{(i,j) \in E} G_{ij} \times Q_{ij}$$

$$\text{s.t. } \sum_{j:(i,j) \in E} Q_{ij} - \sum_{j:(j,i) \in E} Q_{ji} = e_i$$

$$0 \leq Q_{ij} \leq C_{ij}, \quad \forall (i,j) \in E$$

$$\text{其中 } e_i = \begin{cases} v(f), & i = S \\ -v(f), & i = T \\ 0, & i \neq S, T \end{cases}$$

假设游乐项目的初始温度分别为 $T_A, T_B \dots T_J$, 初始热量分别为 $Q_A, Q_B \dots Q_J$, 显然, $T_i = \frac{Q_i}{C_i}$ 。每条路的初始热量为 $Q_a, Q_b \dots Q_n$, 疏导后达到最优状态后每个游乐项目均衡温度为 T^* , 均衡热量 $Q_A^*, Q_B^* \dots Q_J^*$, 显然, $T^* = \frac{Q_i^*}{C_i}$ 。

建立**热量传导模型** (1) 的步骤如下:

①构建二分图 $G(V_1, V_2, E)$;

②定义左集合 V_1 代表初始状态的所有游乐项目以及道路

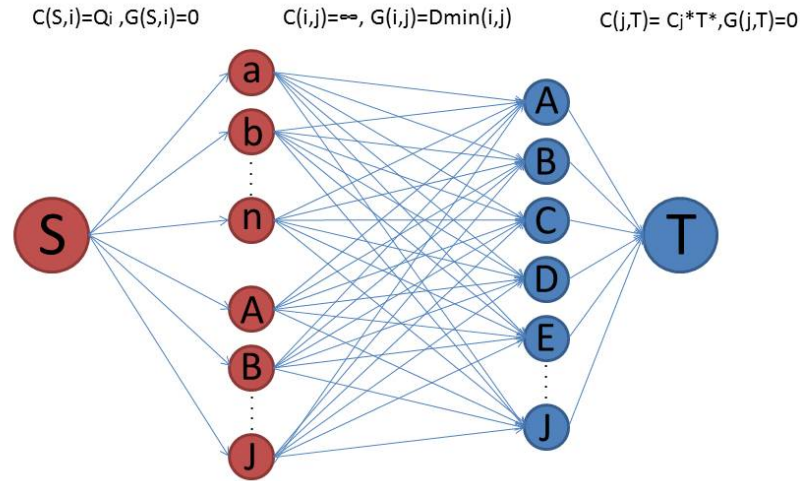
③定义右集合 V_2 代表目标状态的所有游乐项目

④构建连接 V_1, V_2 的有向边 (i, j) , 其中 $i \in V_1, j \in V_2$, 容量 $C_{ij} = \infty$,

$$G_{ij} = D_{minij}$$

⑤构建连接源 S 到 V_1 的有向边 (S, i) , 其中 $i \in V_1$, 容量 $C_{si} = Q_i, G_{si} = 0$

⑥构建连接源 V_2 到 T 的有向边 (j, T) , 其中 $j \in V_2$, 容量 $C_{jt} = Q_j^*, G_{jt} = 0$



采用迭代法计算最小费用最大流^[6]（具体代码见附录）。其主要步骤如下：

①求出从发点到收点的最小费用通路 Q_{ST} 。

②对该通路 Q_{ST} 分配最大可能的流量 f ：

$$f = \min_{(i,j) \in u(S,T)} \{Q_{ij}\}$$

并让通路上的所有边的容量相应减少 f 。这时，对于通路上的饱和边，其单位流费用相应改为 ∞ 。

③作该通路所有边 Q_{ST} 上所有边 (i, j) 的反向边 (j, i) 。令

$$Q_{ij} = f, C_{ij} = -C_{ji}$$

④在这样构成的新网络中，重复上述步骤①②③，直到从发点 S 到收点 T 的全部流量等于 $v(f)$ 为止（或者再也找不到从 S 到 T 的最小费用道路）。

最终，边的流量 Q_{ij} 其中 $i \in V_1, j \in V_2$ ，即为达到热量均衡的最优状态，且总代价最小的分配方式，即 i 点到 j 点所要疏导的热量（游客数）。

5.1.4.3 热量传递模型的修正

①疏导回自身的问题

考虑到游乐项目A的每场游客容纳数为400人，远比其他项目为多。因此，当游乐项目A结束时会使 T_A 骤降，如果按照5.1.4.2所示的建模方法采用最小费用最大流的算法求解，为达到均衡温度 T^* ，极有可能会出现从游乐项目A疏导到自身的情况。边的流量 Q_{ij} 其中 $i \in V_1, j \in V_2$ ，不能完全正确的反应人流疏导情况。

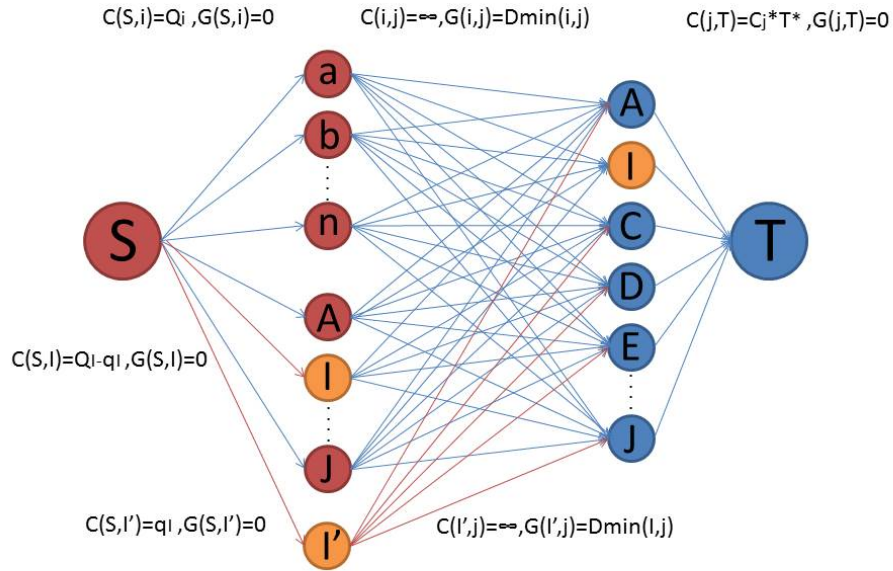
更一般地，考虑到每个项目结束时都会存在这样疏导回自身的问题（容纳人数过小时可以忽略），就此，提出模型的改进。

根据此问题建立**热量传导模型（2）**，假设当游乐项目I结束时

1. 在模型（1）的基础上引入点 I' ， $I' \in V_1$ 。
2. 构建连接源S到 I' 的有向边 (S, I') ，容量 $C_{SI'} = q_i$ ，单位流量费用 $G_{SI'} = 0$
3. 构建点 I' 到点j的有向边，其中 $j \in V_2$ 且 $I \neq j$ 。容量 $C_{I'j} = \infty$ ，单位流量费用

$$G_{I'j} = D_{minIj}。$$

4. 修正点S到点I的有向边 (S, I) 的容量 $C_{SI} = Q_I - q_I$ ，删除有向边 (I, I) ，其中 $I \in V_1, I \in V_2$ 。



采用与模型（1）相同的方法计算最小费用最大流，即可得到达到热量均衡的最优状态，且总代价最小的分配方式，边的流量 Q_{ij} ，其中 $i \in V_1, j \in V_2$ ，即游乐项目i到游乐项目j所要疏导的热量（游客数）。边的流量 $Q_{I'j}$ 其中 $j \in V_2$ ，即为项目I结束时，需要向项目j疏导的游客数，由于删除了构建点I到点I，其中 $I \in V_1, I \in V_2$ ，的有向边，解决了人流疏导回项目自身的问题。

②游客的偏好问题

考虑到现实生活中，每个游乐项目的吸引力可能不同，即每个游乐项目项目的均衡温度 T^* 可能不同。因此，可对每个游乐项目引入相对偏好修正系数 dT_i 。

$dT_i > 0$, 表示游乐项目较受欢迎。

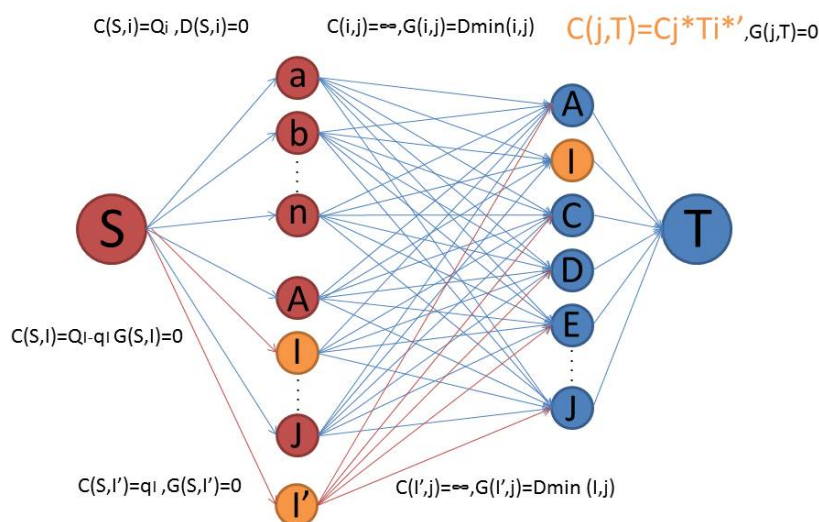
$dT_i < 0$, 表示此游乐项目较冷清。

每个游乐项目修正后的均衡温度 $T_i^{*'} = T^* + dT_i$, 总热量 Q_Z 应满足下列条件

$$Q_Z = \sum c_i \times T_i^{*'}$$

关于此问题建立**热量传导模型** (3)

修正点 j 到点 T 有向边的容量 $C_{jT} = c_j \times T_i^{*'}$, 其中 $j \in V_2$ 。



用与模型 (1)、模型 (2) 相同的方法计算该网络的最小费用最大流, 即可得出在考虑人们偏好情况下的最优疏导方案。

③ 游客的进出园问题

在现实生活中, 游乐园内的游客数不是一成不变的, 游客总数是实时变化的。

由于没有修正过的均衡温度 $T^* = \frac{Q_Z}{c_Z}$, 其中 $Q_Z = \sum Q_i$, $c_Z = \sum c_i$, 均衡温度 T^* 与游

乐园总人数 Q_Z 有一一对应的关系, 因此游乐园的总人数实际上反应在了未修正的均衡温度 T^* 上, 即模型本身即反应出了游客进出园的问题。

5.1.4.4 模型测试

所有测试中总客流量是 9976 人由此可推出园内均衡温度, 输入每个项目点或道路中点的初始热量, 并可由此求出初始温度。输出结果是在满足均衡温度的条件下的最优调整方案, 纵向为起点, 横向为终点, 输出结果中流向自身相当于这部分人待在该项目不走。

① 热量传导模型 (1) 的测试

输入数据:

项目点	A	B	C	D	E	F	G	H	I	J
初始热量 Q_i	1900	1500	700	300	1300	500	900	300	400	1200
初始温度 T_i	157	62.5	35	25	65	25	60	15	30	48
均衡热量 Q_i^*	665	1320	1100	660	1100	1100	825	1100	731	1375
均衡温度 $T^*=55$										

道路中点	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
初始热量 Q_i	46	35	30	80	60	100	80	60	83	95	100	60	77	30	40
初始温度 T_i															
均衡热量 Q_i^*															

输出数据:

起点/终点	A	B	C	D	E	F	G	H	I	J
A	665	0	0	215	0	0	0	0	571	449
B	0	1320	180	0	0	0	0	0	0	0
C	0	0	700	0	0	0	0	0	0	0
D	0	0	0	300	0	0	0	0	0	0
E	0	0	0	0	1100	200	0	0	0	0
F	0	0	0	0	0	500	0	0	0	0
G	0	0	0	0	0	75	825	0	0	0
H	0	0	0	0	0	0	0	300	0	0
I	0	0	0	0	0	0	0	400	0	0
J	0	0	0	0	0	0	0	293	0	907
a	0	0	10	17	0	0	0	0	0	19
b	0	0	0	35	0	0	0	0	0	0
c	0	0	30	0	0	0	0	0	0	0
d	0	0	80	0	0	0	0	0	0	0
e	0	0	0	33	0	27	0	0	0	0
f	0	0	100	0	0	0	0	0	0	0
g	0	0	0	0	0	80	0	0	0	0
h	0	0	0	60	0	0	0	0	0	0
i	0	0	0	0	0	83	0	0	0	0
j	0	0	0	0	0	95	0	0	0	0
k	0	0	0	0	0	0	0	0	100	0
l	0	0	0	0	0	0	0	0	60	0
m	0	0	0	0	0	0	0	77	0	0
n	0	0	0	0	0	0	0	30	0	0
o	0	0	0	0	0	40	0	0	0	0

结果分析: 此模型中没有考虑人们的偏好, 所有项目的均衡温度一致, 而且AB项目点热量大, 从结果来看, 确实有很多热量从AB项目点流到其他项目点, 符合实际。

一般来讲某项目点一定会有一部分人待着不动 (也就是流向自身), 但是有些极端的情况下如I (周围点温度很低), 其项目点有可能出现全部热量流走的现象, 不过大多数情况下不会出现, 此极端问题限于时间没有进一步讨论修正。

②热量传导模型 (2) 的测试

输入数据:

道路中点	A	B	C	D	E	F	G	H	I	J
初始热量 Q_i	460	1500	700	300	1300	500	1200	600	400	1640
初始温度 T_i	38	62.5	35	25	65	25	80	30	30	65.5
均衡热量 Q_i^*	665	1320	1100	660	1100	1100	825	1100	731	1375
均衡温度 $T^*=55$										

项目点	A'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
初始热量 Q_i	400	60	22	30	44	60	100	80	95	83	95	100	77	45	45	40
初始温度 T_i																
均衡热量 Q_i^*																

输出数据:

起点/终点	A	B	C	D	E	F	G	H	I	J
A	0	0	0	259	0	0	0	0	171	30
B	0	1320	174	6	0	0	0	0	0	0
C	0	0	700	0	0	0	0	0	0	0
D	0	0	0	300	0	0	0	0	0	0
E	193	0	0	0	1100	7	0	0	0	0
F	0	0	0	0	0	500	0	0	0	0
G	0	0	0	0	0	375	825	0	0	0
H	0	0	0	0	0	0	0	600	0	0
I	0	0	0	0	0	0	0	400	0	0
J	302	0	0	0	0	0	0	10	0	1328
a	60	0	0	0	0	0	0	0	0	0
b	0	0	22	0	0	0	0	0	0	0
c	30	0	0	0	0	0	0	0	0	0
d	0	0	44	0	0	0	0	0	0	0
e	0	0	60	0	0	0	0	0	0	0
f	0	0	100	0	0	0	0	0	0	0
g	80	0	0	0	0	0	0	0	0	0
h	0	0	0	95	0	0	0	0	0	0
i	0	0	0	0	0	83	0	0	0	0
j	0	0	0	0	0	95	0	0	0	0
k	0	0	0	0	0	0	0	0	100	0
l	0	0	0	0	0	0	0	0	77	0
m	0	0	0	0	0	0	0	45	0	0
n	0	0	0	0	0	0	0	45	0	0
o	0	0	0	0	0	40	0	0	0	0
A'	0	0	0	0	0	0	0	0	383	17

结果分析：此模型模拟了A项目结束后，A项目的400个人A'要去到别的项目游玩，也就是说这部分人一定不会回到自身，一定流向其他项目点，结果中也体现了这一点，结果符合实际。

③热量传导模型（3）的测试

输入数据：

项目点	A	B	C	D	E	F	G	H	I	J
初始热量 Q_i	1900	1500	700	300	1300	500	900	300	400	1200
初始温度 T_i	157	62.5	35	25	65	25	60	15	30	48
均衡热量 Q_i^*	999	1320	1034	633	887	1080	844	1100	849	1230
考虑偏好的均衡温度 T_i^*	83	55	51.5	53	44	54	56	55	64	49

道路中点	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
初始热量 Q_i	54	35	30	77	60	100	83	44	83	99	94	56	77	37	47
初始温度 T_i															
均衡热量 Q_i^*															
考虑偏好的均衡温度 T_i^*															

输出数据：

开始/结束	A	B	C	D	E	F	G	H	I	J
A	999	0	0	141	0	0	0	0	699	61
B	0	1320	157	23	0	0	0	0	0	0
C	0	0	700	0	0	0	0	0	0	0
D	0	0	0	300	0	0	0	0	0	0
E	0	0	0	0	887	212	0	201	0	0
F	0	0	0	0	0	500	0	0	0	0
G	0	0	0	0	0	56	844	0	0	0
H	0	0	0	0	0	0	0	300	0	0
I	0	0	0	0	0	0	0	400	0	0
J	0	0	0	0	0	0	0	85	0	1115
a	0	0	0	0	0	0	0	0	0	54
b	0	0	0	35	0	0	0	0	0	0
c	0	0	0	30	0	0	0	0	0	0
d	0	0	77	0	0	0	0	0	0	0
e	0	0	0	60	0	0	0	0	0	0
f	0	0	100	0	0	0	0	0	0	0
g	0	0	0	0	0	83	0	0	0	0
h	0	0	0	44	0	0	0	0	0	0
i	0	0	0	0	0	83	0	0	0	0
j	0	0	0	0	0	99	0	0	0	0
k	0	0	0	0	0	0	0	0	94	0
l	0	0	0	0	0	0	0	0	56	0
m	0	0	0	0	0	0	0	77	0	0
n	0	0	0	0	0	0	0	37	0	0
o	0	0	0	0	0	47	0	0	0	0

结果分析：该模型考虑了游客对于某项目的偏好程度，调整了临界温度，其余输入数据和①相同，结果确实和①中结果有所差异，反映了偏好程度对于分配的影响。

5.2 皇冠假日酒店预定房间量模型建立与求解

5.2.1 预订房间量定义界定

已知酒店 2015 年全年预定数据，其中包括每笔订单预定时间、到店时间、离店时间等信息。影响房间预定量的主要因素(比如季节, 工作日, 周末等)与当天入住量会有很大关联, 而时时预定量则存在很大的随机性: 首先, 在哪一天进行预订与天是否为特定日期并无关联; 其次, 选择在入住前多少天提前预定, 取决于不同用户的消费习惯, 同样与季节、周末、节假日、暑期等因素并无显著关联。因此, 将预订房间量定义为当天入住房间数。

5.2.2 模型建立方法

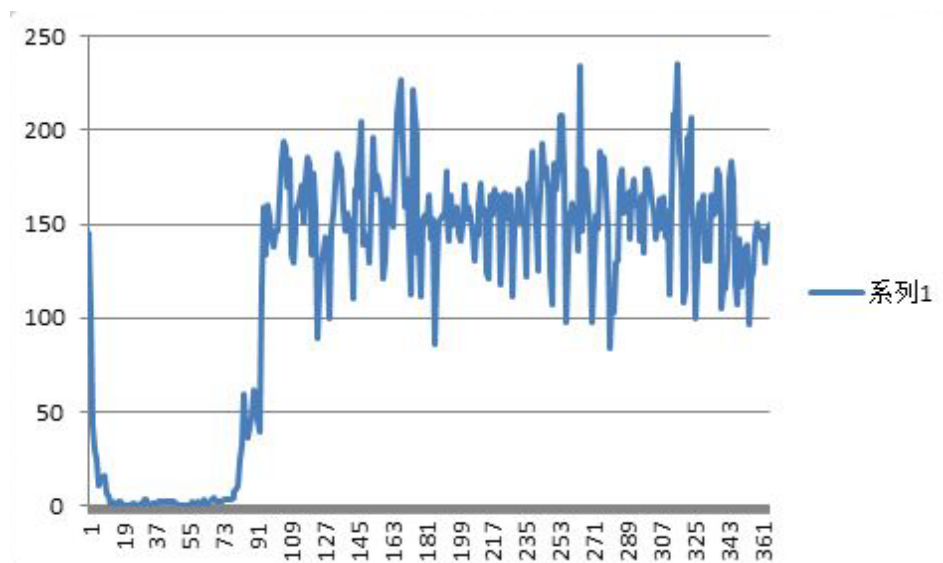
选取多个自变量来解释一个因变量, 考虑使用多元线性回归模型。

5.2.3 第一次筛选影响因素

初步观察提供数据, 如下图所示一月到三月预订房间量相比之后的月份显著较低。进一步分析发现, 预定时期与入住时期之间相隔一个月到四个月不等。因此第一季度的入住量相当部分来自于 2014 年内的预定, 但表格中却缺乏这一部分的数据, 因此表格所提供的第一季度预订房间量数据缺乏可靠性。

在考虑季度变化因素时, 如果顾客第一季度入住, 则不能对季度变量进行有效的赋值(如选取第一季度的平均数、中位数等)。这就意味着, 仅考虑后三个季度, 季度变化对于预订房间量是有意义的; 考虑全年, 仅凭后三个季度的数据无法模拟出第一季度的预订房间量。

综上所述, 入住日期在哪一季度将不再作为影响因素考虑。



5.2.4 特性变量处理赋值

按 5.2.3 论述，排除 2015 年第一季度的订房数据，以 4 到 12 月预订房间量为研究对象，考虑入住日期是否为周末、是否为法定节假日、是否为暑期（7 月、8 月）、是否为特定节日（非法定假日），对预订房间量的影响。

四组影响因素共产生八种情况，每种情况作为特性定量，要与最终的房间量产生多元线性关系，需要交特性变量处理赋值。这里采用平均数作为复制手段，比如“是周末入住”赋值为所有周末房间量的平均数。每种特性变量赋值情况见下表。

	W（周末）	H（假日）	S（暑期）	V（节日）
是	142.0385	148.3333	151.4677	150.5
否	158.5228	154.2335	154.5399	153.9736

以 1 月 3 号为例，2015 年 1 月 3 号是周末，是假日，不是暑期，不是特定节日，所以 $W=142.0385$, $H=148.3333$, $S=154.5399$, $V=153.9736$ 。

5.2.5 第一次模型拟合

以预定房间量为因变量，四个赋值处理后的特性变量为自变量，进行多元线性回归模型拟合，拟合结果如下。

ANOVA^b

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	15798.638	4	3949.660	5.400	.000 ^a
	Residual	197472.947	270	731.381		
	Total	213271.585	274			

a. Predictors: (Constant), 是否节日, 是否周末, 是否暑期, 是否法定假日

b. Dependent Variable: 入住房间数

上表为回归显著度检验，整个模型的回归显著度 $0.000 < 0.01$ ，显著性很高。

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-383.359	493.873		-.776	.438
	是否周末	.988	.222	.264	4.441	.000
	是否法定假日	.317	1.144	.017	.277	.782
	是否暑期	1.065	1.291	.049	.825	.410
	是否节日	1.122	2.522	.026	.445	.657

a. Dependent Variable: 入住房间数

上表为参数显著度检验，除了是否周末这一自变量参数 0.000,<0.01 显著性很高以外，其余参数包括常数项显著性都大于 0.05，显著性很低。

Model Summary^a

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.272 ^a	.074	.060	27.044

a. Predictors: (Constant), 是否节日, 是否周末, 是否暑期, 是否法定假日

b. Dependent Variable: 入住房间数

上表显示了决定系数 R^2 ——因变量的方差可以被自变量解释的比例，范围在 0 到 1 之间，越大越好，而该结果表明 $R^2=0.073$ ，显然不符合要求。

尽管最终模型的回归显著度较高，但由于个别自变量的参数显著度很低，导致了模型尚且存在决定系数很低等很大的问题。因此，还需要对自变量——影响因素进行第二次筛选。

5.2.6 第二次筛选影响因素

将各自变量与因变量进行 Pearson 相关性分析，分析结果如下表。

Correlations

		入住房间数	是否周末	季度	是否法定假日	是否暑期	是否节日
入住房间数	Pearson Correlation	1	.267**	.058	.052	.046	.023
	Sig. (2-tailed)		.000	.338	.387	.446	.700
	N	275	275	275	275	275	275
是否周末	Pearson Correlation	.267**	1	-.003	.160**	.008	.007
	Sig. (2-tailed)	.000		.966	.008	.895	.907
	N	275	275	275	275	275	275
季度	Pearson Correlation	.058	-.003	1	-.023	-.087	.062
	Sig. (2-tailed)	.338	.966		.704	.148	.307
	N	275	275	275	275	275	275
是否法定假日	Pearson Correlation	.052	.160**	-.023	1	-.143*	.027
	Sig. (2-tailed)	.387	.008	.704		.018	.654
	N	275	275	275	275	275	275
是否暑期	Pearson Correlation	.046	.008	-.087	-.143*	1	-.105
	Sig. (2-tailed)	.446	.895	.148	.018		.083
	N	275	275	275	275	275	275
是否节日	Pearson Correlation	.023	.007	.062	.027	-.105	1
	Sig. (2-tailed)	.700	.907	.307	.654	.083	
	N	275	275	275	275	275	275

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

相关性的判断依据见下表。

相关系数绝对值大小	一般判断
0.8 ~ 1	非常强的相关
0.6 ~ 0.8	强相关
0.4 ~ 0.6	中度相关
0.2 ~ 0.4	弱相关
0 ~ 0.2	非常弱的相关

除了是否周末这一影响因素，其余与影响因素与预订房间量的相关性非常弱。相比之下，是否特定节日（非法定节假日）与像因变量的相关性最为微小，考虑将该影响因素剔除，再次进行模型拟合。

5.2.7 第二次模型拟合

以预定房间量为因变量，四个赋值处理后的特性变量为自变量，进行多元线性回归模型拟合，拟合结果如下。

ANOVA^b

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	15654.031	3	5218.010	7.156	.000 ^a
	Residual	197617.555	271	729.216		
	Total	213271.585	274			

a. Predictors: (Constant), 是否暑期, 是否周末, 是否法定假日

b. Dependent Variable: 入住房间数

整个模型的显著性依然很高。

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-202.769	280.589		-.723	.471
	是否周末	.988	.222	.264	4.450	.000
	是否法定假日	.323	1.143	.017	.283	.778
	是否暑期	1.007	1.282	.046	.785	.433

a. Dependent Variable: 入住房间数

参数的显著性较第一次拟合有所提高。

Model Summary^b

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.271 ^a	.073	.063	27.004

a. Predictors: (Constant), 是否暑期, 是否周末, 是否法定假日

b. Dependent Variable: 入住房间数

决定系数依然很低，笔者分析，造成这一结果的原因在于，根据现有数据所

选取因变量，会造成拟合结果的大量重复：比如，同为非暑期，非周末，非法定假日的日期有很多，它们的拟合结果完全相等，因此，因变量的而方差很难被自变量解释。

该问题的解决方法是，按照现有数据的标准差，对相等数据进行按正态分布的离散处理。

5.2.8 预定房间量数学模型

$$C=-202.7685$$

$$B_w=0.9885$$

$$B_h=0.3229$$

$$B_s=1.0066$$

$$Q=-202.7685+0.9885*W+0.3229*H+1.0066*S$$

得出工作日、周末、节假日的房间预定量后，再根据三者的标准差(见下表)：

	工作日	周末	节假日
标准差	27.80996	24.5750598	33.08812

按正态分布离散得出每天的预订房间量。

5.2.9 2016 年第一季度预定房间数

按照 5.2.1 的数学模型，得出 2016 年第一季度预定房间数见下表：

日期	预订房间量	日期	预订房间量	日期	预订房间量	日期	预订房间量
2016/1/1	144	2016/1/24	137	2016/2/16	150	2016/3/10	188
2016/1/2	175	2016/1/25	190	2016/2/17	118	2016/3/11	191
2016/1/3	145	2016/1/26	135	2016/2/18	127	2016/3/12	120
2016/1/4	188	2016/1/27	169	2016/2/19	164	2016/3/13	105
2016/1/5	171	2016/1/28	164	2016/2/20	156	2016/3/14	177
2016/1/6	123	2016/1/29	164	2016/2/21	136	2016/3/15	170
2016/1/7	169	2016/1/30	138	2016/2/22	166	2016/3/16	184
2016/1/8	174	2016/1/31	155	2016/2/23	133	2016/3/17	176
2016/1/9	199	2016/2/1	179	2016/2/24	179	2016/3/18	119
2016/1/10	133	2016/2/2	160	2016/2/25	181	2016/3/19	170
2016/1/11	147	2016/2/3	141	2016/2/26	166	2016/3/20	127
2016/1/12	182	2016/2/4	176	2016/2/27	147	2016/3/21	165
2016/1/13	138	2016/2/5	176	2016/2/28	134	2016/3/22	160
2016/1/14	140	2016/2/6	99	2016/2/29	150	2016/3/23	163
2016/1/15	164	2016/2/7	140	2016/3/1	144	2016/3/24	174
2016/1/16	129	2016/2/8	102	2016/3/2	162	2016/3/25	160
2016/1/17	144	2016/2/9	185	2016/3/3	180	2016/3/26	143
2016/1/18	219	2016/2/10	145	2016/3/4	162	2016/3/27	133
2016/1/19	156	2016/2/11	111	2016/3/5	121	2016/3/28	155
2016/1/20	218	2016/2/12	140	2016/3/6	131	2016/3/29	193
2016/1/21	169	2016/2/13	129	2016/3/7	148	2016/3/30	162
2016/1/22	137	2016/2/14	185	2016/3/8	186	2016/3/31	142
2016/1/23	145	2016/2/15	139	2016/3/9	152		

黄色代表节假日，橙色代表周末，白色代表工作日。

六、模型评价

6.1 Youth 游乐园客流疏导模型评价

6.1.1 客流量较少情况的最优路线模型

此模型考虑客流量较少时,疏导和提醒的主要目的是给游客一个最优的游玩方案,即在满足达到希望游览点的条件下路程或时间的代价最小。模型的优点就是能够准确的结合实际情况给出最优路线,因为项目数较少,可以用枚举法的思想深度优先搜索的算法来得到准确解。

但是该模型本质还是属于“旅行商问题”,如果项目点数很多,难以得到精确解,只能得到近似解。而且一旦客流量变大,以路程为代价的模型不符合实际人们的需求,以时间为代价的模型因为出现排队轮次的问题不适用,每个人的决策都会互相影响。因此在客流量较大的情况下用热量传导模型解决。

6.2.2 客流量较多情况的热量传导模型

在大客流情况下将人流疏导问题转化为热量传递问题,可以直接给出在给定情况下的人流疏导方案,巧妙的使复杂的疏导问题简单化,且应用范围广。模型中所用的最小费用最大流算法时间复杂度为 $O(n^3)$,在点数以及边数不大的

($n \leq 30$)的情况下运行效率较高。考虑了游乐项目结束后的疏导问题、人们的偏好以及游乐园人数的动态变化,与现实情况吻合。

但是本模型由于假设在路上的人群集中在道路中点,建立的网络流模型可能会存在误差,必须用有限元分析才有可能得到精确解。由于假设人群要全都被疏导到游乐项目处,而在现实生活中,人群有可能并非全都集中在游乐项目处,即每个游乐项目的均衡温度 T^* 可能比实际情况要高。该模型在实际过程中有可能出现不合实际的情况,因为为了达到代价最小,有可能某项目点全部人都要去比较近的项目点补充,当然这也可以通过增加状态量修正,限于时间有限没有在本文中探讨。

6.2 皇冠假日酒店预订房间量模型评价

该模型显著体现处不同影响因素对结果的影响;通过对数据之间相关性等分析,选择出对拟合模型有显著相关的项目,剔除干扰项。

由于客观原因——数据的缺失(第一季度预订房间量),导致关键因素——“入住日期在哪一季度”不得已被剔除,整个模型的全面性还是有待改进。对于特性一致的变量,如何体现出预订房间量的标准差,按正态分布对数据进行离散的方法有待商榷。

七、参考文献

- [1] 百度百科, 最短路径, http://baike.baidu.com/link?url=G8dHMGU3JMipoXOYIYoYRgHpYp3fgegTotEnVRkDZMTPuEgm6SPJu6tpeC_D92Swdiuw3zVPjMuuJO01TTgYK, 2016-4-30
- [2] 百度百科, TSP 问题数学模型, <http://baike.baidu.com/link?url=3EIszMRleimzbpOxQELhbky0C-gVx-uR3GX4Xbyt6MSID1-VOCwmGuY82wzvkJXcBE9BJAM0hjoVogIhgT7KKK>, 2016-5-2。
- [3] 中国德州网, 图与网络, <http://www.tichinese.com>, 2016-5-1。
- [4] 王海英, 图论算法及其 MATLAB 实现, 北京: 北京航空航天大学出版社, 2010。
- [5] 百度百科, 深度优先搜索, <http://baike.baidu.com/view/288277.htm>,

2016-5-1。

[6] 百度文库，最小费用最大流的 Matlab 程序代码，
http://wenku.baidu.com/link?url=cD-tVZI6iJTLCTwVWyk8geLVxTJxyh4pCW3ou_2Jfc58GK7gYFIEBNvQboTbZQY1XYjXxf6iQTFcnHj3O1g1huCvoWYfaSOkbFmfFFYv0oa, 2016-4-30。

[7] 姜启源、谢金星、叶俊，数学模型（第四版），北京：高等教育出版，2011。

八、附件

附件①最优路线模型代价为路程时深度优先搜索代码

```
function main()
global wmin n ans G qg;
wmin=999999;
n=11;
[G]=xlsread('1_Gdata2');
for i=1:n
    for j=1:n
        for k=1:n
            if i~=j & G(i,k)+G(k,j)<G(i,j)
                G(i,j)=G(i,k)+G(k,j);
            end
        end
    end
end
end %处理最短路
for i=1:n
    pd(i)=0;
end %设置所有都没去过
m=5; qg=4;
%pd(2)=1;
pd(3)=1; pd(4)=1; pd(5)=1;
pd(11)=1;
w=0;
for i=1:n-qg+1
    a(i)=0;
end
a(1)=m;
k=1;
DFS(m,pd,w,a,k)
ans
wmin
end
%深度优先搜索
```

```

function DFS(x,pd,w,a,k)
global wmin n ans G qg;
if w>wmin
    return;
end
flag=0;
for i=1:n
    if pd(i)==0
        flag=1;
        break;
    end
end
if flag==0
    if (w<wmin) & (a(n-qg+1)==1) wmin=w;
    ans=a;
end
end
for i=1:n
    if i~=x & pd(i)==0
        w=w+G(x,i);
        pd(i)=1;
        k=k+1;
        a(k)=i;
        DFS(i,pd,w,a,k);
        pd(i)=0;
        w=w-G(x,i);
        a(k)=0;
        k=k-1;
    end
end
end
end

```

附件②最优路线模型代价为时间时深度优先搜索代码

```

function main()
global tmin n ans G qg ts ta;
tmin=999999;
n=11;
[G]=xlsread('1_Gdata');
for i=1:n
    for j=1:200000
        ts(i,j)=0;
    end
end %时间表初始化
ta=[1,1980,75,150,150,300,150,120,90,90,120];

```

```

for i=1:n
    for j=1:round(200000/ta(i))
        ts(i,ta(i)*j)=1;
    end
    ts(i+1,1)=1;
end
ta(1)=0;
for i=1:n
    for j=1:n
        for k=1:n
            if i~=j & G(i,k)+G(k,j)<G(i,j)
                G(i,j)=G(i,k)+G(k,j);
            end
        end
    end
end
end % 处理最短路
G=round(G/1.5);
for i=1:n
    pd(i)=0;
end % 设置所有都没去过
m=5; qg=4;
%pd(2)=1;
pd(3)=1; pd(4)=1; pd(5)=1;
pd(11)=1;
t0=4200;
for i=1:n-qg+1
    a(i)=0;
end
a(1)=m;
k=1;
DFS(m,pd,t0,a,k,0)
ans
tmin-t0
end
% 深度优先搜索
function DFS(x,pd,t,a,k,dt)
global tmin n ans G qg ts ta;
if t>tmin
    return;
end
flag=0;
for i=1:n
    if pd(i)==0
        flag=1;

```

```

        break;
    end
end
if flag==0
    if (t<tmin) & (a(n-qg+1)==1)
        tmin=t;
        ans=a;
    end
end
for i=1:n
    if i~=x & pd(i)==0
        pd(i)=1;
        k=k+1;
        a(k)=i;
        t=t+G(x,i);
        dt=0;
        while ts(i,t)==0
            t=t+1;
            dt=dt+1;
        end
        t=t+ta(i);
        DFS(i,pd,t,a,k,dt);
        t=t-G(x,i)-dt-ta(i);
        dt=0;
        pd(i)=0;
        a(k)=0;
        k=k-1;
    end
end
end
end

```

附件③热量传导模型网络流代码

```

n=38;
[C]=xlsread('c.xlsx');%弧容量
[b]=xlsread('b.xlsx');%弧上单位流量的费用
wf=0;
wf0=Inf; %wf 表示最大流量, wf0 表示预定的流量值
for(i=1:n)
    for(j=1:n)
        f(i,j)=0;
    end;
end %取初始可行流 f 为零流
while(1)
for(i=1:n)

```

```

        for(j=1:n)
            if(j~=i)
                a(i,j)=Inf;
            end;
        end;
    end;
end% 构造有向赋权图
for(i=1:n)
    for(j=1:n)
        if(C(i,j)>0&f(i,j)==0)
            a(i,j)=b(i,j);
        elseif(C(i,j)>0&f(i,j)==C(i,j))
            a(j,i)=-b(i,j);
        elseif(C(i,j)>0)a(i,j)=b(i,j);
            a(j,i)=-b(i,j);
        end;
    end;
end
for(i=2:n)
    p(i)=Inf;
    s(i)=i;
end %用 Ford 算法求最短路, 赋初值
for(k=1:n)
    pd=1; %求有向赋权图中 vs 到 vt 的最短路
    for(i=2:n)
        for(j=1:n)
            if(p(i)>p(j)+a(j,i))p(i)=p(j)+a(j,i);s(i)=j;pd=0;
            end;
        end;
    end
    if(pd)break
end;
end %求最短路的 Ford 算法结束
if(p(n)==Inf)break;
end %不存在 vs 到 vt 的最短路, 算法终止. 注意在求最小费用最大流时构造有向赋权图中不会含负权回路, 所以不会出现 k=n
dvt=Inf;
t=n; %进入调整过程, dvt 表示调整量
while(1) %计算调整量
    if(a(s(t),t)>0)|(a(s(t),t)==0 & s(t)<t)
        dvtt=C(s(t),t)-f(s(t),t); %前向弧调整量
    elseif(a(s(t),t)<0)|(a(s(t),t)==0 & s(t)>t)
        dvtt=f(t,s(t));
    end %后向弧调整量
    if(dvt>dvtt)

```

```

    dvt=dvtt;
end
if(s(t)==1)break;
end %当 t 的标号为 vs 时, 终止计算调整量
t=s(t);
end %继续调整前一段弧上的流 f
pd=0;if(wf+dvt>=wf0)
    dvt=wf0-wf;
    pd=1;
end%如果最大流量大于或等于预定的流量值
t=n;while(1) %调整过程
if(a(s(t),t)>0)|(a(s(t),t)==0 & s(t)<t)
    f(s(t),t)=f(s(t),t)+dvt; %前向弧调整
elseif(a(s(t),t)<0)|(a(s(t),t)==0 & s(t)>t)
    f(t,s(t))=f(t,s(t))-dvt;
end %后向弧调整
if(s(t)==1)
    break;
end %当 t 的标号为 vs 时, 终止调整过程
t=s(t);
end
if(pd)
    break;
end %如果最大流量达到预定的流量值
wf=0;
for(j=1:n)
    wf=wf+f(1,j);
end;
end %计算最大流量
zwf=0;
for(i=1:n)
    for(j=1:n)
        zwf=zwf+b(i,j)*f(i,j);
    end;
end %计算最小费用
f %显示最小费用最大流
wf %显示最小费用最大流量
zwf %显示最小费用, 程序结束

```