评委一评分，签名及备注	队号： 10300	评委三评分，签名及备注
评委二评分，签名及备注	选题： C	评委四评分，签名及备注
题目：	最佳旅游路线设计与对比	

摘要

本文主要研究最佳旅游路线的设计问题。在满足相关约束条件的情况下，花最少的钱游览尽可能多的景点。首先运用主成分分析法对景点品质进行分析，选取主成分，并把主成分得分按照方差的贡献率进行加权求和，得出每一个景点的综合评价指数，进而对各景点进行排序，基于对此的研究，建立数学模型，设计出最佳的旅游路线。

1. 给定时间约束，要求为家庭设计合适的旅游路线。我们建立了一个最优规划模型，在给定游览景点个数的情况下以人均总费用最小为目标。再引入 0—1 变量表示是否游览某个景点，从而推出交通费用和景点花费的函数表达式，给出相应的约束条件，使用 lingo 编程对模型求解。旅游方案：重庆→合川→万州→江津→丰都→重庆，人均费用为 1080 元（不考虑旅游人数对游览费用的响）。

2. 放松时间约束，要求代表们游遍所有的景点，该问题也就成了典型的 TSP 问题。同样使用第一问的模型，改变时间约束，使用 lingo 编程得到最佳旅游路线为：重庆→合川→达州→都江堰→成都→峨眉山→南充→泸州→绵阳→重庆

3. 要求在第一问的基础上充分考虑代表们的旅游意向，建立模型求解。通过对附件一数据的观察，我们使用综合评判的方法，巧妙地将代表们的意愿转化为对相应旅游景点的权重，再对第一问的模型稍加修改，编程求出对应不同景点数的最佳路线。推荐路线：重庆→合川→都江堰→成都→九寨沟→重庆。

4.建立新的约束条件和目标函数的线性规划模型：

$$MinT=\sum_{i=1}^{11}\sum_{j=1}^{11}t_{ij}x_{ij}+\frac{1}{2}\sum_{i=1}^{11}\sum_{j=1}^{11}x_{ij}\left(t_i+t_j\right)+\frac{1}{2}\sum_{i=1}^{11}\sum_{j=1}^{11}x_{ij}\left(e_i+e_j\right),$$

5.考虑车费、景点费、车次衔接、旅游路线最短等因素，使用最优化方法和线性规划法，建立总费用最小的最优路线目标函数：

$$MinA=\sum_{i=1}^{11}\sum_{j=1}^{11}c_{ij}x_{ij}+\frac{1}{2}\sum_{i=1}^{11}\sum_{j=1}^{11}x_{ij}\left(b_i+b_j\right)+\frac{1}{2}\sum_{i=1}^{11}\sum_{j=1}^{11}x_{ij}\left(d_i+d_j\right),$$

本文思路清晰，模型恰当，结果合理. 利用 Excel 排序，SPSS 预测，这样给处理数据带来了不少的方便。本文成功地对 0—1 变量进行了使用和约束，简化了模型建立难度，并且可方便地利用数学软件进行求解。此外，本文建立的模型具有很强普适性，便于推广。

关键词： 主成分分析法 TSP 问题 最佳路线 线性规划 最小费用

最佳旅游路线设计与对比

1 背景资料

随着暑假的来临，越来越多的人会带着家人孩子一起外出旅游，不同的家庭消费、时间、人员都各不相同，我们就以我们所在的地区—重庆，为具体的研究对象，讨论出不同情况的家庭对旅游的相关需求，假设设立重庆、四川是个景点作为旅游景点，作为暑假旅游的全部景点。

重庆市地图如下所示：

图 1 重庆市地图

四川省地图如图所示：

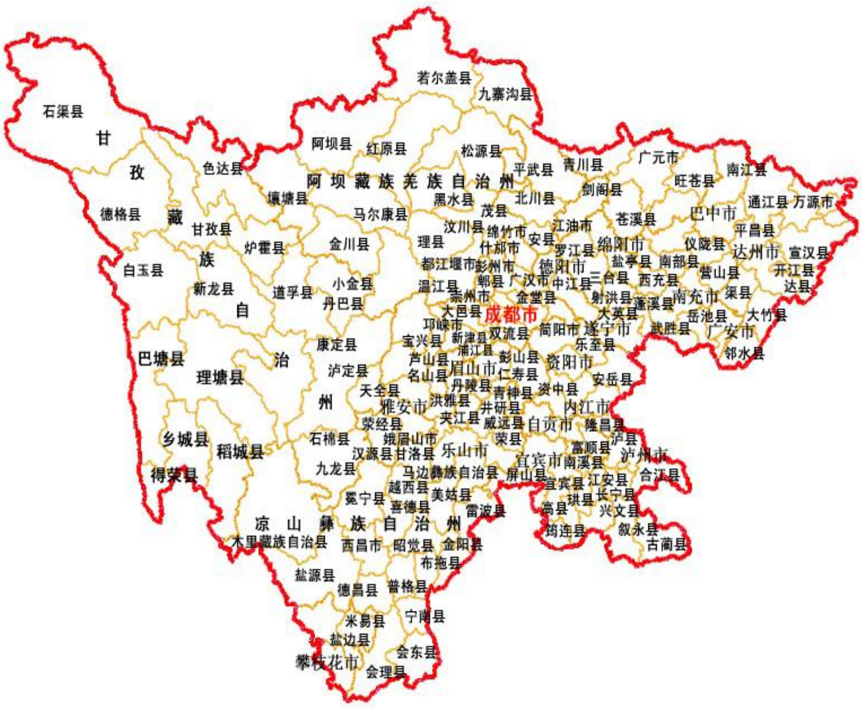


图 2 四川省地图

从重庆市出发选择合适的路线旅游每一个城市一次，使路费最少，其本质是一个 TSP 商旅问题。我们可以对已有的 TSP 商旅模型进行修改，通过编程将所有路线所需费用列举出来，找出最经济的路线。

关于 TSP 旅行商问题旅行商问题（Traveling Saleman Problem TSP）是 VRP 的特例，由于 Gaery[1]已证明 TSP 问题是 NP 难题，因此，VRP 也属于 NP 难题。旅行商问题（TSP）又译为旅行推销员问题、货郎担问题，简称为 TSP 问题，是最基本的路线问题，该问题是在寻求单一旅行者由起点出发，通过所有给定的需求点之后，最后再回到原点的最小路径成本。最早的旅行商问题的数学规划是由 Dantzig（1959）等人提出。TSP 问题在物流中的描述是对应一个物流配送公司，欲将 n 个客户的订货沿最短路线全部送到。如何确定最短路线。TSP 问题最简单的求解方法是枚举法。它的解是多维的、多局部极值的、趋于无穷大的复杂解的空间，搜索空间是 n 个点的所有排列的集合，大小为（n-1)。可以形象地把解空间看成是一个无穷大的丘陵地带，各山峰或山谷的高度即是问题的极值。求解 TSP，则是在此不能穷尽的丘陵地带中攀登以达到山顶或谷底的过程。

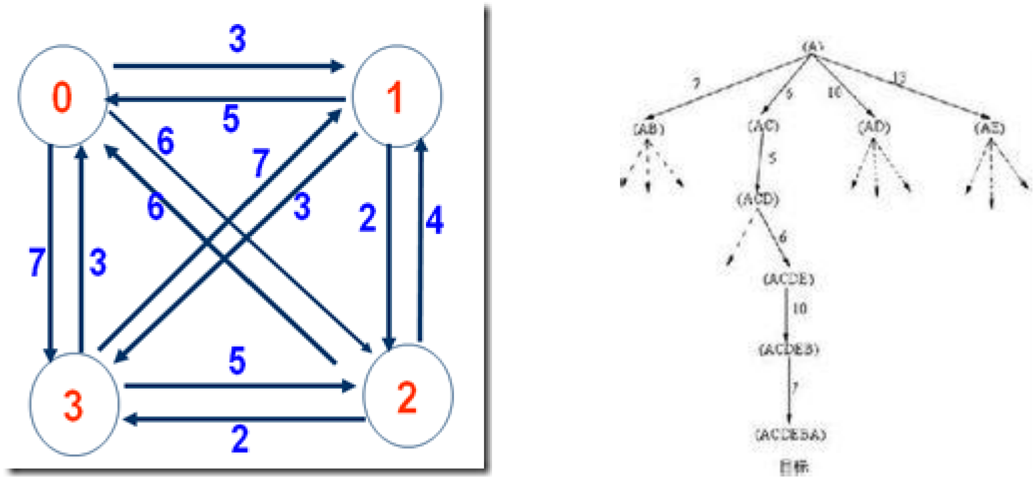


图 3　TSP 问题模型图

TSP 旅行商问题常见算法：枚举法，蚁群算法，模拟退火柴法，TSP 问题是一个组合优化问题。该问题可以被证明具有 NP 计算复杂性。因此，任何能使该问题的求解得以简化的方法，都将受到高度的评价和关注。旅行推销员问题是数图论中最著名的问题之一，即“已给一个 n 个点的完全图，每条边都有一个长度，求总长度最短的经过每个顶点正好一次的封闭回路”。Edmonds，Cook 和 Karp 等人发现，这批难题有一个值得注意的性质，对其中一个问题存在有效算法时，每个问题都会有有效算法。迄今为止，这类问题中没有一个找到有效算法。倾向于接受 NP 完全问题（NP-Complet 或 NPC）和 NP 难题（NP-Hard 或 NPH）不存在有效算法这一猜想，认为这类问题的大型实例不能用精确算法求解，必须寻求这类问题的有效的近似算法。此类问题中，经典的还有 子集和问题；Hamilton 回路问题；最大团问题。

解决方法：

1、途程建构法（Tour Construction Procedures）

从距离矩阵中产生一个近似最佳解的途径，有以下几种解法：

2）节省法（Clark and Wright Saving）：以服务每一个节点为起始解，根据三角不等式两边之和大于第三边之性质，其起始状况为每服务一个顾客后便回场站，而后计算路线间合并节省量，将节省量以降序排序而依次合并路线，直到最后。

3）插入法（Insertion procedures）：如插入法、最省插入法、随意插入法、最远插入法、最大角度插入法等。

2、途程改善法（Tour Improvement Procedure）

先给定一个可行途程，然后进行改善，一直到不能改善为止。有以下几种解法：

1）K-Opt(2/3 Opt)：把尚未加入路径的 K 条节线暂时取代路径中 K 条节线，并计算其成本（或距离），如果成本降低（距离减少），则取代之，直到无法改善为止，K 通常为 2 或 3。

2）Or-Opt：在相同路径上相邻的需求点，将之和本身或其它路径交换且仍保持路径方向性。

回溯法思想：

为了应用回溯法，所要求的解必须能表示成一个 n- 元组(x1,…,Xn)，其中 x1 是取自某个有穷集 Si。通常，所求解的问题需要求取一个使某一规范函数 P(x1,…,Xn)取极大值（或取极小值或满足该规范函数条件）的向量。

假定集合 Si 的大小是 mi，于是就有 m=m1m2…Mn 个 n-元组可能满足函数 P。所谓硬性处理是构造这 m 个 n-元组并逐一测试它们是否满足 P,从而找出该问题的所有最优解。而回溯法的基本思想是，不断地用修改过的函数 Pi(x1,…Xi)（即限界函数）去测试正在构造中的 n-元组的部分向量(x1,…,Xi)，看其是否可能导致最优解。如果判定(x1,…,Xi)不可能导致最优解，那么就可能要测试的后 n-i 个元素组成的向量一概略去。因此回溯法作的次数比硬性处理作的测试次数（m 次）要少得多。用回溯法求解的旅行商问题，即在枚举法的基础上多了一个约束条件，约束条件可以分为两种类型：显示约束和隐式约束。分支限界法思想：本题采用 FIFO 分支限界法。如前所述，分支限界法是在生成当前 E-结点全部儿子之后再生成其它活结点的儿子，且用限界函数帮助避免生成不包含答案结点子树的状态空间的检索方法。在总的原则下，根据对状态控件树中结点检索的次序的不同又将分支限界设计策路分为数种不同的检索方法。在求解旅行商问题时，程序中采用 FIFO 检索（First In First Out），它的活结点表采用一张先进先出表（即队列）。可以看出，分支限界法在两个方面加速了算法的搜索速度，一是选择要扩展的节点时，总是选择选择一个最小成本的结点，尽可能早的进入最有可能成为最优解的分支；二是扩展节点的过程中，舍弃导致不可行解或导致非最优解的子结点。

贪心法是一种改进了的分级处理方法。它首先对旅行商问题进行描述，选取一种度量标准。然后按这种度量标准对 n 个输入城市排序，并按序一次输入一个城市。如果这个输入和当前已构成在这种量度意义下的部分最优解加在一起不能产生一个可行解，则不把这个城市加入到这部分解中。这种能够得到某种量度意义下的最优解的分级处理方法成为贪心方法。

获得最优路径的贪心法应一条边一条边地构造这棵树。根据某种量度来选择将要计入的下一条边。最简单的量度标准是选择使得迄今为止计入的那些边的成本的和有最小增量的那条边。

2 问题重述

随着暑假的来临，很多家长会选择这个时间带孩子去某城市旅游，旅游已成为提高人们生活质量的重要活动。但不同的家庭有不同的需求(人数，费用限制，时间限制等)，我们选择自己所在的城市—重庆，为暑期旅游路线第一站，

一号线：重庆→九寨沟、南充；

二号线：重庆→绵阳、峨嵋；

三号线：重庆→合川、万州；

四号线：重庆→都江堰、峨眉山；

五号线：重庆→海螺沟、绵阳；

每条线路中的景点可以全部参观，也可以参观其中之一。不仅如此，一起参观景点的人数越多，每个家庭承担的费用也会越小。

结合上述要求，请你回答下列问题：

1. 请你们为他们设计合适的旅游路线，使他们在今年暑假一个月的时间里花最少的钱游尽可能多的地方，并估算除吃饭之外的费用。

2. 如果他们打算今年暑假完成对重庆的旅游，请你们为他们设计合适的旅游路线，使在重庆境内的交通费用尽量地节省。

3. 请你们为不同家庭设计合适的旅游路线，使家庭的 15 天时间内花最少的钱游尽可能多的地方。

4. 如果有一些家庭的时间非常充裕（比如一个月），他们打算将上述旅游景点全部参观完毕后才离开四川，请你们为他们设计合适的旅游路线，使在四川境内的交通费用尽量地节省。

5. 假设景点的开放时间为 8:00 至 18:00。

3 问题分析

2.1 问题背景的理解：

根据对题目的理解我们可以知道，旅游的总费用包括交通费用和在景点游览时的费用，而在确定了要游览的景点的个数后，所以我们的目标就是在满足所有约束条件的情况下，求出成本的最小值。

2.2 问题一和问题二的分析：

问题一要求我们为主办方设计合适的旅游路线，使不同家庭在一个月时间内花最少的钱游尽可能多的地方。在这里我们的做法是在满足相应的约束条件下，先确定游览的景点数，然后计算出在这种情况下的最小花费。这样最终会得出几种最佳方案，而组织方可以根据自己的实际情况进行选择。

问题二实质上是在问题一的基础上改变了时间约束，即代表们要游览所有的景点，我们完全可以使用与问题一同样的方法进行求解。

2.3 问题三的分析：

问题三要求我们在问题一的基础上充分考虑代表们对各个景点的意愿来设计最佳旅游路线，而代表们的意愿由附件 1 给出。对于意愿，我们的做法是将其转化为相应的权重，然后乘以相应的旅游景点的花费，再利用问题一的模型得出几种最佳方案供主办方选择。

2.4 问题四的分析：

这是一个综合比对问题，我们可以根据空间距离、运行路线、时间 安排、线路总体相似度、总体差异度、特色饱和度等准则建立一个层次分析模型 ， 对每一条线路进行综合比较，最后得出各条路线的综合打分，根据综合打分来判 断各条路线的相似程度。

4 模型假设

1. 所给的 5 条路线每条路线中的景点可以全部参观，也可以参观其一；
2. 参观景点的人数越多，每人承担的费用越少；
3. 其中部分家庭往返于各个旅游景点，其交通费用、在景点的花费、在景点的逗留时间参照当地客运公司及旅行社的数据；
4. 假设在旅游过程当中不走回头路，并且游玩了一个旅游城市后必须回到市里再去下一个旅游城市；
5. 假设旅游只在白天的十二个小时内进行，而且在一个城市游玩的时间不能够超过三天；
6. 一个景点直接到达另外一个景点是指，途中经过的其他景点只是一个转站地，而并不进行游览；
7. 在限定的时间内，代表们最终要返回重庆，并且假设成都是代表们肯定要去的一个旅游景点；
8. 假设参观景点的人数每增加一人，每个代表在景点的费用就减少原价的 1‰；
9. 家庭成员在途中和游览景点的时间为 12 小时，而另外 12 小时为休息、用餐及其他琐事时间。
10. 将城市和路径的关系转化为图论问题。

5 符号说明

i，j——第 i 个或者第 j 个景点， i， j=1，2，……，12；

分别表示重庆、九寨沟、合川、绵阳、峨嵋、江津、万州、都江堰、达州、成都、南充、泸州；

c——每个家庭代的旅游总花费；

t_i——每个家庭在第 i 个景点的逗留时间；

c_i——每个家庭在 i 个景点的总消费；

t_{ij}——从第 i 个景点到第 j 个景点路途中所需时间；

c_{ij}——从第 i 个景点到第 j 个景点所需的交通费用；

$$r_{ij}=\begin{cases}1 & \text{代表们直接从第}i\text{个景点到达第}j\text{个景点}\\0 & \text{其他}\end{cases}$$

Z——各旅游城市的评价值

6 模型建立及求解

6.1 主成分分析

主成分分析(Principal Components Analysis，PCA)也称为主分量分析，是 一种通过降维来简化数据结构的方法：如何把多个变量（指标）化为少数几个综 合变量（综合指标），而这几个综合变量可以反映原来多个变量的大部分信息。
为了使这些综合变量所含的信息互不重叠，应要求它们之间互不相关。主成份分析的目的是从原始的多个变量取若干线性组合，能尽可能多地保留 原始变量中的信息。从原始变量到新变量是一个正交变换（坐标变换）。

主成份分析的目的是从原始的多个变量取若干线性组合，能尽可能多地保留原始变量中的信息。从原始变量到新变量是一个正交变换（坐标变换）。设有 是一个 维随机变量且有二阶矩，记 $X=(X_1,X_2,\cdots X_p)'$ 是一个 维随机变量且有二阶矩，记 $\mu=E(X)$，

$\sum=Var(X)$，考虑它的线性变换。

$$\begin{cases} Y_1 = L_1{}'X = l_{11}X_1 + \cdots + l_{p1}X_p \\ \vdots \\ Y_p = L_p{}'X = l_{1p}X_1 + \cdots + l_{pp}X_p \end{cases} \tag{1}$$

易见

$$\begin{aligned} Var(Y_i) &= L_i{}'\sum L_i \\ Cov(Y_i,Y_j) &= L_i{}'\sum L_j \end{aligned} \qquad i,j=1,2,\cdots,p \tag{2}$$

得到：

$$\begin{aligned} Var(Y_i) &= L_i{}'\sum L_i \\ Cov(Y_i,Y_j) &= L_i{}'\sum L_j \end{aligned} \qquad i,j=1,2,\cdots,p \tag{2}$$

6.2 目标函数的确立：

经过对题目分析，我们可以知道本题所要实现的目标是，使家庭在一个月时间内花最少的钱游览尽可能多的地方。显然，花费最少和游览的景点尽量多是该问题的两个目标。因此，我们的做法是在满足相应的约束条件下，先确定游览的景点数，然后计算出在这种情况下的最小花费。这样最终会得出几种旅游路线，而组织方可以根据自己的实际情况进行选择。

游览的总费用由 2 部分组成，分别为交通总费用和在旅游景点的花费。我们定义：

m ——每个家庭的旅游总花费；

m_1 ——每个家庭的交通总费用；

m_2 ——每个家庭的旅游景点的花费；

从而得到目标函数： Min　$m = m_1 + m_2$

（1）交通总花费

因为 c_{ij} 表示从第 i 个景点到第 j 个景点所需的交通费用，而 r_{ij} 是判断家庭是否从第 i 个景点直接到第 j 个景点的 0—1 变量，因此我们可以很容易的得到交通总费用为：

$$m_1 = \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times c_{ij}$$

（2）旅游景点的花费

因为 c_i 表示家庭成员们在 i 个景点的总消费，r_{ij} 也可以表示出家庭成员是否到达过第 i 个和第 j 个景点，而整个旅游路线又是一个环形，因此 $\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times (c_i + c_j)$ 实际上将成员在所到景点的花费计算了两遍，从而我们可得旅游景点的花费为：

$$m_2 = \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times (c_i + c_j)$$

从而我们可以得到目标函数为：

$$\begin{aligned}\text{Min}\quad m &= m_1 + m_2 \\ &= \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times c_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times (c_i + c_j)\end{aligned}$$

6.3 约束条件：

①时间约束

由题目可知，家庭在重庆的旅游时间应该不多于 10 天(120 小时)，而这些时间包括在路途中的时间和在旅游景点逗留的时间。因为t_{ij}表示从第i个景点到第j个景点路途中所需时间，所以路途中所需总时间为$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times t_{ij}$；$t_i$表示成员们在第$i$个景点的逗留时间，故家庭们在旅游景点的总逗留时间为$\frac{1}{2}\times\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times(t_i+t_j)$。因此，总的时间约束为：

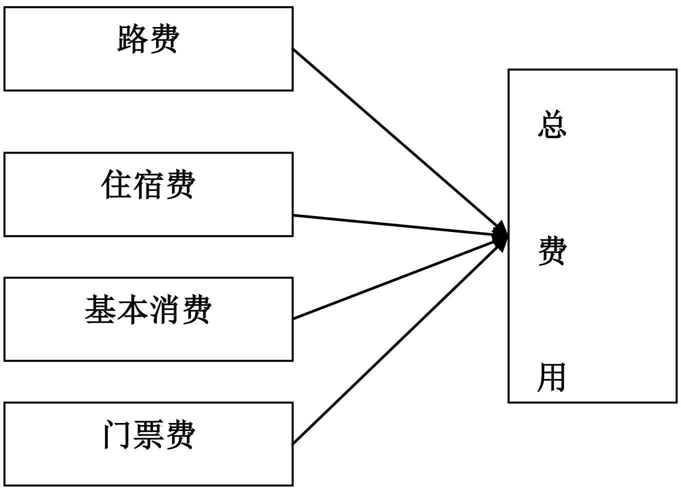
$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times t_{ij}+\frac{1}{2}\times\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times(t_i+t_j)\le 120$$

②旅游景点数约束

根据假设，整个旅游路线是环形，即最终代表们要回到重庆，因此$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}$即表示代表们旅游的景点数，这里我们假定要旅游的景点数为n（n=2，3，……，11）。因此旅游景点数约束为：

$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}=n \quad (n=2, 3, \cdots\cdots, 11)$$

③家庭消费的约束：



④0——1 变量约束

我们可以把所有的景点连成一个圈，而把每一个景点看做圈上一个点。对于每个点来说，只允许最多一条边进入，同样只允许最多一条边出来，并且只要有一条边进入就要有一条边出去。因此可得约束：

$$\sum_i r_{ij} = \sum_j r_{ij} \le 1 \qquad （i，j=1，2，……，11）$$

当 $i=1$ 时，因为重庆市出发点，所以 $\sum_{i=1} r_{ij} = 1$；

$j=1$ 时，因为家庭成员最终要回到重庆，所以 $\sum_{j=1} r_{ij} = 1$。

综合以上可知，

$$\sum_i r_{ij} = \sum_j r_{ij} \le 1 \qquad （i，j=1，2，……，11）$$

$$\sum_{i=1} r_{ij} = 1 \qquad \sum_{j=1} r_{ij} = 1$$

同样，当 i，$j \ge 2$ 时，根据题意不可能出现 $r_{ij} = r_{ji} = 1$，即不可能出现游客在两地间往返旅游，因为这样显然不满足游览景点尽量多的原则。因此我们可得约束：

$$r_{ij} \times r_{ji} = 0 \qquad （i，j=2，3，……，11）$$

6.4 模型建立：

6.4.1 建立图论数学模型

将各个旅游景点之间的关系转化为图论问题，并做以下分析：建立有向图 $G=(V,A)$。其中 $V=\{V_1,V_2,......,V_n\}$ 称为图 G 的顶点集，V 中的每一个元素 $V_i(i=1,2,\dots n$ 称为该图的一个顶点，在该题中表示 n 城市；$A=\{a_1,a_2,......a_n\}$ 称为图 G 的弧集，A 中的每个元素 $a_k=(V_i,V_j)$ 称为该图的一条从 V_i 到 V_j 的弧，在此题中表示各个城市两两连线的集合。[1]

设城市个数为 n，d_{ij} 表示两个城市 i 与 j 之间的距离，$x_{ij}=0$ 或 1（1 表示走过城市 i 到城市 j 的路，0 表示没有选择走这条路）。本题可以向 TSP 问题进行转化，则 TSP 问题的数学模型为：

$$\min \sum_{i \ne j} d_{ij} x_{ij}$$

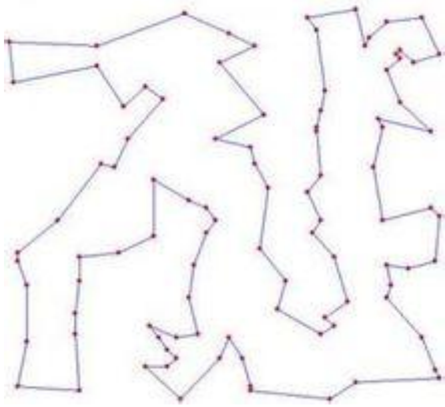


图 4　TSP 模型图

对于任意一条给定的旅游路线 L，都可以算出这条路给游客带来的满意度，具体的满意度的目标函数：

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ik}(t)\right]^{\beta}}{\sum\limits_{s\not\subset tabu_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]\beta} & j\notin tabu_k\\ 0 & j\in tabu_k\end{cases}$$

时间的目标函数构造。对于 N 中路径 L，旅游时间的目标函数

T（L）$=\sum\limits_{E_{i,j}\in L}W_{i,j}^{1}+\sum\limits_{V_i\in L}Y_i^{1}$，单位为小时。

（3）花费的目标函数构造。对于 N 中路径 L，花费的目标函数

C（L）$=\sum\limits_{E_{i,j}\in L}W_{i,j}^{2}+\sum\limits_{V_i\in L}Y_i^{2}+r\times\left\lceil\dfrac{T(L)}{15}\right\rceil$， 其中 r 为一个家庭每天固定的住宿费用，本题中我们认为 r=150 元，$\left\lceil\dfrac{T(L)}{15}\right\rceil$表示对$\dfrac{T(L)}{15}$下取整，注意，分母之所以取 15，是因为我们已作假设每天的旅游支配时间只有 15 小时。

综上所述，我们可以得到总的模型为：

Min　$m=m_1+m_2$

$$=\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times c_{ij}+\frac{1}{2}\times\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times\left(c_i+c_j\right)$$

6.4.2 模型求解与结果分析：

将 L 看成变量，这是一个典型的双目标决策问题，我们的决策目标函数有两个：P(L)和 C(L)。对于“P(L)尽量大，C(L)尽量小”的双目标要求，我们很难给出一个完全贴合实际的界定函数 F，使得 F（P(L)，C(L)）取最大（或最小）时的（P(L)，C(L)）就是最优解。也即如果通过构造新目标函数，硬生生的将问题转化为“单目标决策问题”，虽然会使问题的求解变的简单可行，但模型会因此而失真，从而大打折扣。因此，我们放弃上述方法，采用下面“不断优化可行解集”的方法，将会得到非常好的结果。

其基本思路是通过构造恰当的优化条件（用不等式形式表达），不断从可行解的集合中除去不满足优化条件的点，直到可行解的集合只有有限的几个或更少为止；最后从剩下的为数不多的可行解中，通过人脑进行主观判断，得到最贴合实际的最优可行解，因为具体什么才算最优是一个模糊的概念，并且还是一对矛盾。所以最后一步需要人来判断。

为实现上述算法，并抽象成严格数学模型，先做对后面需要用的名词给出定义。

1.可行解：满足 T(L)≤30 天的解均称为可行解。

2.劣解：L0 称为劣解，如果存在可行解 L1，且不等于 L0，满足 P(L0)<P(L1)，C(L0) >C(L1)，即 L0 得到的满意度比 L1 小，花费却比 L1 大。

3.非劣解（有效解）：不是劣解的可行解称为非劣解或有效解。

4.目标空间：所有非劣解对应的目标函数集（P(L) C(L)）组成的集合称为目标空间。

5.优化后的目标空间：在目标空间中，除出不满足优化条件的元素，剩下的元素所组成的集合称为优化后的目标空间。

6.优化条件：在本题目中，我们要求“P(L)尽量大，C(L)尽量小”，因而可根据实际情况，分别给 P(L)，C(L)加一个范围限制，P(L)>P min，C(L)<C max;

我们的算法实质分四步：

1.求解所有可行解。

2.从所有可行解中得到非劣解，并组成非劣解集合。

3.由非劣解集合生成目标空间，优化目标空间，并得到优化后的目标空间。

4.找到优化后的目标空间中元素所对应的非劣解，并人为主观的从中判断出最优解。

第 2、3、4 步均是简单的比较和判定，完全可以找到多项式时间级的算法。现在问题的关键是解决“第 1 步的问题”，即求解所有可行解。

如采用穷举法，将图 N 中的所有路径一一列出，再验证每条路是否满足 T(L)≤30 天，从而可以得到所有可行解。但穷举路径法的时间复杂度为 O( )，其中 n 为图 N 的顶点数，在本题目中 n=34，即使用计算机也是永远穷举不完的。故此，我们给出多项式时间的近似算法。

在这里我们引入以下符号：

d_{ij}——第 i 个景点和第 j 个景点之间的路程；

v——成员们所乘坐的旅游大巴的平均时速，v=50km/h；

m——成员们所乘坐的旅游大巴的平均费用，h=0.3 元/h；

我们可以得到 d_{ij} 的具体值，根据公式 $t_{ij}=d_{ij}/v$ 可得到相应的 t_{ij}，同样根据公式 $c_{ij}=d_{ij}\times m$ 可以得到相应的 c_{ij}（i，j=1，2，……，11）。（d_{ij}、t_{ij} 和 c_{ij} 的具体数值见附录）

同样，通过对四川的一些旅行社进行咨询，我们得出家庭成员们们在第 i 个景点的最佳逗留时间和他们在第 i 个景点总消费：

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
7	24	18	12	36	30	12	9	15	24	17

（单位：小时）

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}
120	423	300	135	378	390	175	90	148	303	241

（单位：元）

用 *Matlab* 做模拟图得到：

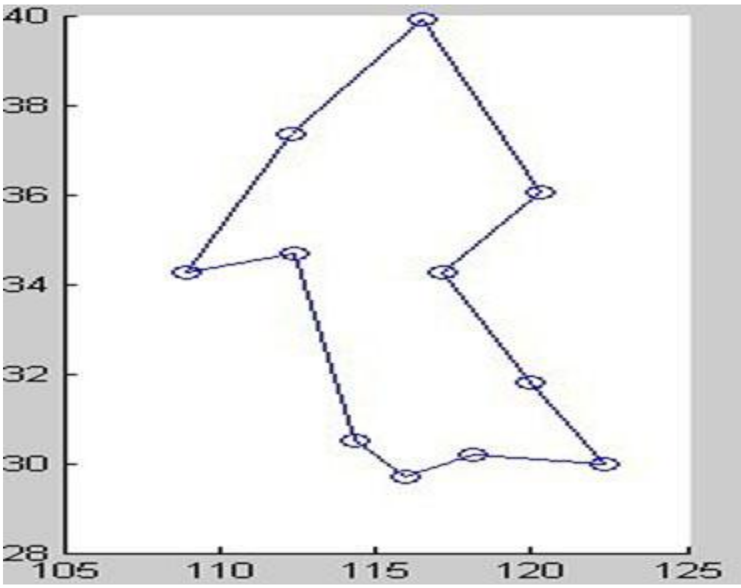


图 5　*Matlab* 模拟图

6.5 问题二

目标函数的确立：

此问与第一问大同小异，不同的是家庭要完成所有景点的旅游，而目标函数是求最少的交通费。由第一问结论可知，交通费用为：$m_1=\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times c_{ij}$

因此，该问题的目标函数为：

$$\text{Min}\quad m_1=\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times c_{ij}$$

编号	路线	满意度	价钱（元）	时间（天）
I	4*-9*-10*-11*-12-13*-12*-11-14*-11-10-9-7*-6*-7-8*-7-22*-23*-22-21*-20*-19*-34*-18-17*-18*-31*-32*	112	9079	29.98
II	23*-22*-21*-20*-19*-34*-18-17*-18*-31*-18-14*-11-12*-13*-12-11*-10*-9*-7-6*-7*-9-4*-1*-2*	109	8980	29.44
III	5*-4*-9*-10*-11*-12-13*-12*-11-14*-11-10-9-7*-6*-7-8*-7-22*-23*-22-21*-20*-19*-34*-18*-31*	104.5	8690.5	28.42
IV	14*-11*-12*-13*-12-11-10*-9*-4-2-3*-2*-1*-4*-5*-4-9-7*-6*-7-8*-7-22*-23*-22-21*-20*-19*	106	8869	28.66
V	3*-2*-1*-4*-9*-10*-11*-12-13*-12*-11-14*-11-10-9-7*-6*-7-8*-7-22*-23*-22-21*-20*-19*-34*	102.5	8448	27.20
VI	23*-22*-21*-20*-19*-34*-18-17*-18*-31*-18-14*-11-12*-13*-12-11*-10*-9*-7-6*-7*-9-4*-1*	100	8435	27.75

6.6 模型建立：

综上所述，我们可以得到总的模型为：

$$\text{Min}\quad m_1=\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times c_{ij}$$

约束条件：

$$\begin{cases}\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times t_{ij}+\frac{1}{2}\times\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}\times\left(t_i+t_j\right)\le 360 \\ \sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}=11 \qquad （i，\ j=1，2，\cdots\cdots，11） \\ \sum_{i}r_{ij}=1 \qquad \sum_{j}r_{ij}=1 \qquad （i，\ j=1，2，\cdots\cdots，11） \\ r_{ij}\times r_{ji}=0 \qquad （i，\ j=2，3，\cdots\cdots，11）\end{cases}$$

6.6.1 模型求解与结果分析：

根据模型，使用 Lingo 编程，得出结果为：

旅游景点数 n	12
每人总花费 m（单位：元）	3243
路线	重庆→成都→峨眉→绵阳→九寨沟→达州→都江堰→江津→万州→丰都→泸州→重庆

6.7 问题三

（1）问题的再次分析

此问在第一问的基础上增加了代表们意愿这一条件，通过对附件一的观察，我们发现家庭人员的意愿分为“去”、“不去”和“无所谓”三种。怎样将这些文字转换到公式中来表达代表们的意愿就成为了解决该问的关键。在这里我们采用加权重的方式，将他们的的意愿理解为对该线路上两个景点的权重，又因为我们最终的目标是使旅游的费用最少，因此越热门的景点相应的权重也应该越低（这是因为权重越低，其与该景点的费用相乘后也越低，从而增加了对该景点游览的可能性）。

（2）数据处理

将所有的“去”替换为 0，所有的“不去”替换为 1，所有的“无所谓”替换为 0.5，从而得到一个 100×5 的矩阵 $\left(A_{ks}\right)_{100\times5}$（见附录）。

我们定义：

λ_i——第 i 个旅游景点的权重。

由假设可知成都是成员肯定要游览的一个景点，因此$\lambda_1 = 0$。

对其他权重进行标准化处理可得：

$$\lambda_2 = \lambda_3 = \frac{\sum_{k=1}^{100} A_{k1}}{\sum_{k=1}^{100}\sum_{s=1}^{5} A_{ks}} = 0.185 \qquad \lambda_4 = \lambda_5 = \frac{\sum_{k=1}^{100} A_{k2}}{\sum_{k=1}^{100}\sum_{s=1}^{5} A_{ks}} = 0.217$$

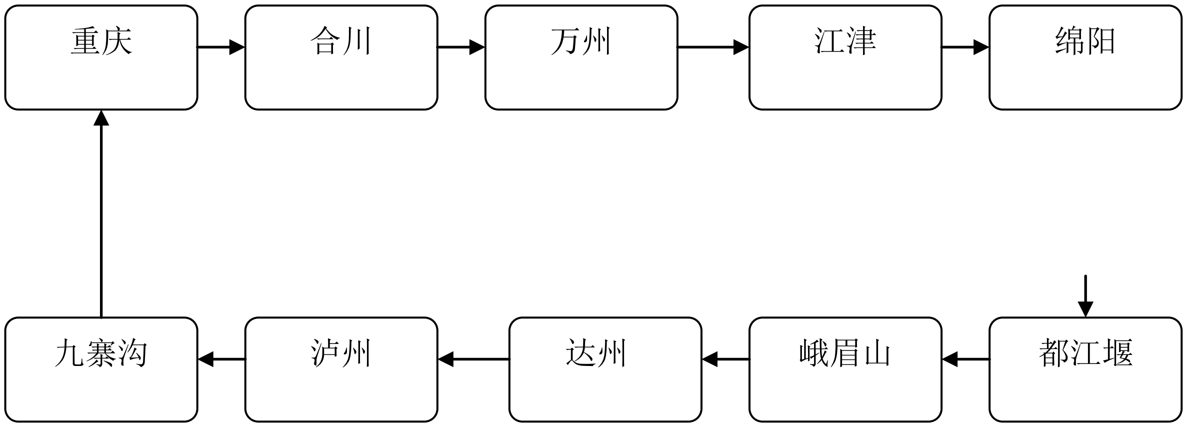
$$\lambda_6 = \lambda_7 = \frac{\sum_{k=1}^{100} A_{k3}}{\sum_{k=1}^{100}\sum_{s=1}^{5} A_{ks}} = 0.196 \qquad \lambda_8 = \lambda_9 = \frac{\sum_{k=1}^{100} A_{k4}}{\sum_{k=1}^{100}\sum_{s=1}^{5} A_{ks}} = 0.206$$

$$\lambda_{10} = \lambda_{11} = \frac{\sum_{k=1}^{100} A_{k1}}{\sum_{k=1}^{100}\sum_{s=1}^{5} A_{ks}} 0.196$$

我们把各景点转化为纯数学形式的点线集合，利用图论方面的知识求解。已经分析该题是已知费用的最大限度求最佳路径问题，因此要选择到下一景点总的最低费用作为去下一景点的前提条件且剩余费用要足够回到徐州。如前面模型，我们把两景点的最省路费作为赋权值$w(e)$，在一定程度上，各景点间的距离与两点间的单程最省路费是成正比的，因此把两景点的最短路费作为权值$w(e)$是可行的。

根据建立的模型，我们利用 LINGO 软件编程得到全局最优解为 7.7 天（其中在景点停留的时间按最短时间计算）。据分析，该游客在无车离开时可以在景点停留更多的时间，因此此结果偏小。可以通过开往下一景点的车次时刻表，计算出该游客在景点区或车站（包括机场）停留的时间。所以通过车次（航班）时刻表查询可得出的最短天数为 9.4 天。

最佳的旅游路线如下：



（3）确定目标函数

本文我们的做法同样是在满足相应的约束条件下，先确定游览的景点数，然后计算出在这种情况下的最小花费。这样最终会得出几种最佳方案，而组织方可以根据自己的实际情况进行选择。

游览的总费用由 2 部分组成，分别为交通总费用和在旅游景点的花费。又根据假设，参观景点的人数每增加一人，在景点的总费用就减少原价的 1‰，假设共有 100 名成员，这就相当于每人在旅游景点的花费打了“九折”，因此得目标函数为：

$$\text{Min}\quad m_\lambda = 100\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i\times r_{ij}\times c_{ij} + \frac{1}{2}\times 90\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i\times r_{ij}\times\left(c_i+c_j\right)$$

而所得结果所对应的每个代表的总花费为：

$$m = \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times c_{ij} + \frac{1}{2}\times\frac{90}{100}\times\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}\times\left(c_i+c_j\right)$$

（4）运用算法确定

①算法一

对于一个 k 阶完全图 G，

1. 置 S 为可变集，S=空集。对给定起点 V_{i_0}， $S=S\bigcup\{V_{i_0}\}$，在 G-S 中找到 V_{i_1}，使得 V_{i_1} 到 V_{i_0} 的权最小， $S=S\bigcup\{V_{i_1}\}$ 。

2. 若 $G-S\neq\Phi$，在 $G-S$ 中找到 V_{i_t}，使得 V_{i_t} 到 $V_{i_{t-1}}$ 的边权最小，$S=S\bigcup\{V_{i_t}\}$，转 2°。

3. 从 S 中依次输出一条近似最优 Hamilton 路 $V_{i_0}V_{i_1}\cdots V_{i_k}$

在此基础上，可以求出旅游时间趋近于给定值 t，且以给定点 V_{i_0} 为起点的最优 Hamilton 路，这要在图 N 的生成完全图 N*中实现，具体的，我们给出如下算法：

②算法二

1. 置 S 为可变顶点集， $S=\Phi$； T 为可变实参数（表示旅游时间），T=0。对给定起点 V_{i_0}， $S=S\bigcup\{V_{i_0}\}$，T=T+$Y_{i_0}^1$，在 N^*-S 中找到 V_{i_1}，使得 V_{i_1} 到 V_{i_0} 的时间权（ W_{i_1,i_0}^1 ）最小，T=T+$W_{i_1,i_0}^1+Y_{i_1}^1$， $S=S\bigcup\{V_{i_1}\}$ 。

2. 若 $N^*-S\neq\Phi$， 在 N^*-S 中找到 V_{i_t}， 使得 V_{i_t} 到 $V_{i_{t-1}}$ 的边权最小，T=T+$W_{i_t,i_{t-1}}^1+Y_{i_t}^1$ 如果 $T>t$ 转 3°，否则 $S=S\bigcup\{V_{i_t}\}$，转 2°。

3. 从 S 中依次输出一条近似最优 Hamilton 路 $V_{i_0}V_{i_1}\cdots V_{i_k}$，并输出其旅游时间

T 。

假设$C=v_1v_2\cdots v_pv_1$是图 G 的一条 Hamilton 路,若存在i,j适合$0<i<j<p$,并且$\omega(v_iv_j)+\omega(v_{i+1}v_{j+1})<\omega(v_iv_{i+1})+\omega(v_jv_{j+1})$则 Hamilton 路 $C_{i,j}=v_1v_2\cdots v_iv_jv_{j-1}\cdots v_{i+1}v_{j+1}\cdots v_pv_1$(原图删除两条边$v_iv_{i+1}$和$v_jv_{j+1}$用两条新边$v_iv_j$和$v_{i+1}v_{j+1}$代换)的权和将会变小。

证明：$W(C_{i,j})$—$W(C)=-\omega(v_iv_{i+1})-\omega(v_jv_{j+1})+\omega(v_iv_j)+\omega(v_{i+1}v_{j+1})<0$，故知定理成立。

基于定理 4，我们给出检验最优解的算法。

③算法三

设$C=v_1v_2\cdots v_p$是顶点集$\{v_1,v_2\cdots v_p\}$的生成完全图的一条近似 Hamilton 路

1° 置$k=0$

2° $k=k+1$

3° 若$k>p-3$，转 9°

4° 置$j=i+1$

5° $j=j+1$

6° 若$j>n-1$，转 2°

7° 若$\omega(v_iv_j)+\omega(v_{i+1}v_{j+1})\geq\omega(v_iv_{i+1})+\omega(v_jv_{j+1})$，转 5°

8° 将 Hamilton 路修改为$v_1v_2\cdots v_iv_jv_{j-1}\cdots v_{i+1}v_{j+1}\cdots v_p$，转 1°

9° 结束

6.8 约束条件

1. 旅游景点数约束

根据假设，整个旅游路线是环形，即最终成员要回到重庆，因此$\sum_{i=1}^{11}\sum_{j=1}^{11}r_{ij}$即表示成员旅游的景点数，这里我们假定要旅游的景点数为n（n=2，3，……，11)。因此旅游景点数约束为:

$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} = n \quad （n=2，3，……，11）$$

2. 0——1 变量约束

我们可以把所有的景点连成一个圈，而把每一个景点看做圈上一个点。对于每个点来说，只允许最多一条边进入，同样只允许最多一条边出来，并且只要有一条边进入就要有一条边出去。因此可得约束：

$$\sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \quad （i，j=1，2，……，11）$$

当$i=1$时，因为重庆是出发点，所以$\sum_{i=1} r_{ij} = 1$；

当$j=1$时，因为代表们最终要回到重庆，所以$\sum_{j=1} r_{ij} = 1$。

综合以上可知，

$$\sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \quad （i，j=1，2，……，11）$$

$$\sum_{i=1} r_{ij} = 1 \qquad \sum_{j=1} r_{ij} = 1$$

同样，当i，$j \ge 2$时，根据题意不可能出现$r_{ij} = r_{ji} = 1$，即不可能出现游客在两地间往返旅游，因为这样显然不满足游览景点尽量多的原则。因此我们可得约束：

$$r_{ij} \times r_{ji} = 0 \quad （i，j=2，3，……，11）$$

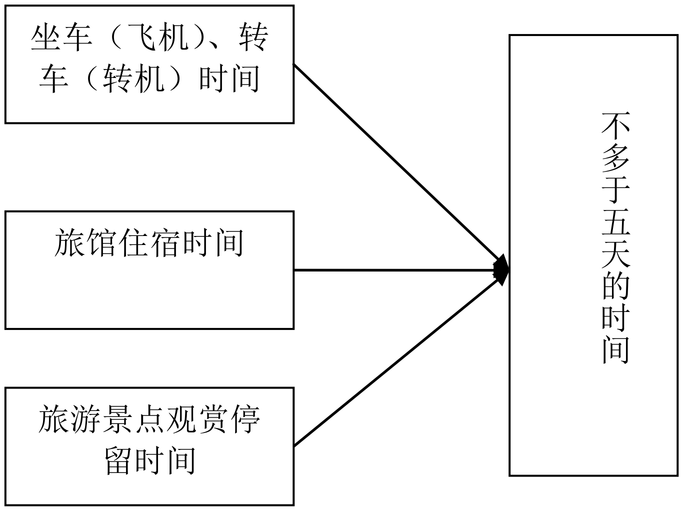
6.9 模型建立：

综上所述，我们可以得到总的模型为：

$$\text{Min} \quad m_{\lambda} = 100 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i \times r_{ij} \times c_{ij} + \frac{1}{2} \times 90 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i \times r_{ij} \times (c_i + c_j)$$

约束条件：

$$\begin{cases} \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times (t_i + t_j) \le 120 \\ \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} = n \quad （n=2，3，……11） \\ \sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \quad （i，j=1，2，……，11） \\ \sum_{i=1} r_{ij} = 1 \quad \sum_{j=1} r_{ij} = 1 \quad （i，j=2，3，……，11） \\ r_{ij} \times r_{ji} = 0 \quad （i，j=2，3，……，11） \end{cases}$$



6.9.1 数据的处理

类似上一问，我们定义：

$\lambda_i^{'}$——第 i 个旅游景点对于第一组家庭的权重；

$\lambda_i^{''}$——第 i 个旅游景点对于第二组家庭的权重。

运用与第一问同样的方法，我们可以得到：

$\lambda_1^{'}=0$	$\lambda_2^{'}=\lambda_3^{'}=0.2$	$\lambda_4^{'}=\lambda_5^{'}=0.212$
$\lambda_6^{'}=\lambda_7^{'}=0.188$	$\lambda_8^{'}=\lambda_9^{'}=0.212$	$\lambda_{10}^{'}=\lambda_{11}^{'}=0.188$
$\lambda_1^{''}=0$	$\lambda_2^{''}=\lambda_3^{''}=0.169$	$\lambda_4^{''}=\lambda_5^{''}=0.221$
$\lambda_6^{''}=\lambda_7^{''}=0.205$	$\lambda_8^{''}=\lambda_9^{''}=0.201$	$\lambda_{10}^{''}=\lambda_{11}^{''}=0.204$

6.9.2 目标函数的确立：

此问中，我们引入以下符号：

m——旅游总花费；

$m_1^{'}$——第一组每个成员的交通总费用；

$m_1^{''}$——第二组每个成员的交通总费用；

$m_2^{'}$——第一组每个成员的旅游景点的花费；

$m_2^{''}$——第二组每个成员的旅游景点的花费。

（上述四个量是假设两个组分别旅游的费用）

m_3——两个组同时在一景点旅游比分别旅游节约的费用。

由以上的假设和符号，我们可以很容易的得到总的目标函数为：

$$\text{Min} \quad m = m_1^{'} + m_1^{''} + m_2^{'} + m_2^{''} - m_3$$

而所得结果所对应的每个代表的总花费为：

$$m = \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times c_{ij} + \frac{1}{2} \times \frac{90}{100} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij} \times \left(c_i + c_j\right)$$

定义：

$$r_{ij}^{'} = \begin{cases} 1 & \text{第一组直接从第}i\text{个景点到达第}j\text{个景点} \\ 0 & \text{其他} \end{cases}$$

$$r_{ij}^{''} = \begin{cases} 1 & \text{第二组直接从第}i\text{个景点到达第}j\text{个景点} \\ 0 & \text{其他} \end{cases}$$

从而可以推得：

$$\begin{cases} m_1^{'} = 50 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i^{'} \times r_{ij}^{'} \times c_{ij} \\ m_1^{''} = 50 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i^{''} \times r_{ij}^{''} \times c_{ij} \end{cases}$$

又因为假设参观景点的人数每增加一人，每个代表在景点的费用就减少原价的 1‰，因此可得：

$$\begin{cases} m_2^{'} = \frac{1}{2} \times 50 \times 0.95 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i^{'} \times r_{ij}^{'} \times \left(c_i + c_j\right) \\ m_2^{''} = \frac{1}{2} \times 50 \times 0.95 \times \sum_{i=1}^{11}\sum_{j=1}^{11} \lambda_i^{''} \times r_{ij}^{''} \times \left(c_i + c_j\right) \end{cases}$$

6.10 约束条件

①时间约束

由题目可知，家庭成员在川的旅游时间应该不多于 15 天（120 小时），而这些时间包括在路途中的时间和在旅游景点逗留的时间。因为 t_{ij} 表示从第 i 个景点到第 j 个景点路途中所需时间，所以两组家庭在路途中所需总时间分别为 $\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{'} \times t_{ij}$ 和 $\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{''} \times t_{ij}$；$t_i$ 表示家庭成员们在第 i 个景点的逗留时间，故两组代表们在旅游景点的总逗留时间分别为 $\frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{'} \times (t_i + t_j)$ 和 $\frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{''} \times (t_i + t_j)$。因此，总的时间约束为：

$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{'} \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{'} \times (t_i + t_j) \le 120$$

$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{''} \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{''} \times (t_i + t_j) \le 120$$

②旅游景点数约束

根据假设，整个旅游路线是环形，即最终代表们要回到重庆，因此 $\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}$ 即表示成员们旅游的景点数，这里我们假定两组家庭要旅游的景点数均为 n（n=2，3，……，11）。因此旅游景点数约束为：

$$\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{'} = \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{''} = n$$

③0——1 变量约束

我们可以把所有的景点连成一个圈，而把每一个景点看做圈上一个点。对于每个点来说，只允许最多一条边进入，同样只允许最多一条边出来，并且只要有一条边进入就要有一条边出去。因此可得约束：

$$\sum_{i} r_{ij}{'} = \sum_{j} r_{ij}{'} \le 1$$

$$\sum_{i} r_{ij}{''} = \sum_{j} r_{ij}{''} \le 1 \qquad （i，\ j=2，……，11）$$

当 $i=1$ 时，因为成都是出发点，所以 $\sum_{i=1} r_{ij}{'} = 1$ 并且 $\sum_{i=1} r_{ij}{''} = 1$；

当 $j=1$时，因为代表们最终要回到成都，所以$\sum_{j=1} r_{ij}{}^{'}=1$并且$\sum_{j=1} r_{ij}{}^{''}=1$。

综合以上可知，

$$\sum_{i} r_{ij}{}^{'}=\sum_{j} r_{ij}{}^{'}\leq 1$$

$$\sum_{i} r_{ij}{}^{''}=\sum_{j} r_{ij}{}^{''}\leq 1 \qquad (i，\ j=2，\cdots\cdots，11)$$

$$\sum_{i=1} r_{ij}{}^{'}=1 \qquad \sum_{i=1} r_{ij}{}^{''}=1$$

$$\sum_{j=1} r_{ij}{}^{'}=1 \qquad \sum_{j=1} r_{ij}{}^{''}=1$$

同样，当i，$j\geq 2$时，根据题意不可能出现$r_{ij}{}^{'}=r_{ji}{}^{'}=1$和$r_{ij}{}^{''}=r_{ji}{}^{''}=1$，即不可能出现游客在两地见往返旅游，因为这样显然不满足游览景点尽量多的原则。因此我们可得约束：

$$r_{ij}{}^{'}\times r_{ji}{}^{'}=0$$

$$r_{ij}{}^{''}\times r_{ji}{}^{''}=0 \qquad (i，\ j=2，3，\cdots\cdots，11)$$

6.10.1 模型建立：

综上所述，我们可以得到总的模型为：

$$\text{Min} \qquad m_{\lambda}=m_1{}^{'}+m_1{}^{''}+m_2{}^{'}+m_2{}^{''}-m_3$$

其中：

$$\begin{cases} m_1{}^{'}=50\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i{}^{'}\times r_{ij}{}^{'}\times c_{ij} \\ m_1{}^{''}=50\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i{}^{''}\times r_{ij}{}^{''}\times c_{ij} \\ m_2{}^{'}=\frac{1}{2}\times 50\times 0.95\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i{}^{'}\times r_{ij}{}^{'}\times\left(c_i+c_j\right) \\ m_2{}^{''}=\frac{1}{2}\times 50\times 0.95\times\sum_{i=1}^{11}\sum_{j=1}^{11}\lambda_i{}^{''}\times r_{ij}{}^{''}\times\left(c_i+c_j\right) \\ m_3=100\times 0.05\times\frac{1}{2}\times\sum_{j=1}^{11}\sum_{i=1}^{11}\gamma_{ij}\times\left(\alpha_i\times\lambda_i\times c_i+\alpha_j\times\lambda_j\times c_j\right) \end{cases}$$

约束条件：

$$
\begin{cases}
\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}' \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}' \times \left(t_i + t_j\right) \le 120 \\
\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}'' \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}'' \times \left(t_i + t_j\right) \le 120 \\
\sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}' = \sum_{i=1}^{11}\sum_{j=1}^{11} r_{ij}{}'' = n \qquad （n=2，3，\cdots\cdots，11） \\
\sum_{i} r_{ij}{}' = \sum_{j} r_{ij}{}' \le 1 \qquad \sum_{i} r_{ij}{}'' = \sum_{j} r_{ij}{}'' \le 1 \qquad （i，j=1，2，\cdots\cdots，11） \\
\sum_{i=1} r_{ij}{}' = 1 \qquad \sum_{i=1} r_{ij}{}'' = 1 \\
\sum_{j=1} r_{ij}{}' = 1 \qquad \sum_{j=1} r_{ij}{}'' = 1 \\
r_{ij}{}' \times r_{ji}{}' = 0 \qquad r_{ij}{}'' \times r_{ji}{}'' = 0 \qquad （i，j=2，3，\cdots\cdots，11）
\end{cases}
$$

问题五：

在问题三的基础上我们引入以下符号：

l——阴雨天气带来的旅游损失；

$c(n)_{\min}$——家庭成员旅游 n 个景点需要的最小的花费；

$c(n)_{\max}$——家庭成员旅游 n 个景点需要的最大的花费；

$l(n)_{\min}$——家庭成员旅游 n 个景点阴雨天气所带大的最小损失；

$l(n)_{\max}$——家庭成员旅游 n 个景点阴雨天气所带大的最大损失。

考虑无向图 G0 (V0,E0,W0)，每一个旅游景点对应于图 G0 内的一个顶点 V_i^0，对任意 i ，j，V_i^0 V_j^0 之间如果有路直接相连，就在 G0 中生成一条边 $E_{i,j}^0$，且边 $E_{i,j}^0$ 上的赋上权 $W_{i,j}^0$，其中 $W_{i,j}^0$ 为这两个景点之间的交通费用。

重新来观察一下问题，问题要求今年暑假完成对重庆，四川的旅游，且花费最小。

先作定义

二覆盖路：图 G0 (V0,E0,W0)的两条路径 L_1, L_2 称为二覆盖路，如果 L_1 中的顶点与 L_2 中的顶点之并组成顶点全集 V0。

则该问题的理想数学模型可叙述为：

求解图 G0 (V0,E0,W0)的权和最小的二覆盖路。

考虑图 G0 的生成完全图 G0 ∗(生成方法见前面)，由定理 1 及定理 2（该处考虑费用权而非时间权，但道理如前)，可知：

图 G0 (V0,E0,W0)的权和最小的二覆盖路就是图 G0 ∗ (V0,E0∗,W0∗)的权和最小的二覆盖路。

于是模型等价于：

求解图 G0 ∗ (V0,E0∗,W0∗)的权和最小的二覆盖路。

现在首先在图 G0∗(V0,E0∗,W0∗)上增加两个点 V1,V2，构造一个新完全图 G(V，E，W)，令

$V = V_0 \bigcup \{V_1, V_2\}$

$E = E_0 * \bigcup \{E_{1,i}, E_{2,i}, i \geq 3\}$ 也就是原图中任意一个顶点均与 V_1, V_2 相关联；

$W_{i,j} = W_{i,j}^0 *$， $i \geq 3; j \geq 3$， 也就是原来各边权值不变；

$W_{1,i} = W_{2,i} = M$ 其中 $M \geq \sum_{i,j=3}^{n+2} W_{i,j}$，也就是让与 V1 ，V2 相连的边的权值大于原来所有边的权值和；

$W_{1,2} = +\infty$

6.10.2 目标函数的确立

六个评价指标的相关系数矩阵为:

$$R = \begin{pmatrix} 1 & -0.088146 & -0.23058 & 0.11421 & -0.21947 & 0.10483 \\ -0.088146 & 1 & 0.20762 & 0.32925 & 0.50615 & 0.37672 \\ -0.23058 & 0.20762 & 1 & 0.057933 & 0.26334 & -0.033186 \\ 0.11421 & 0.32925 & 0.057933 & 1 & 0.36964 & 0.014597 \\ -0.21947 & 0.50615 & 0.26334 & 0.36964 & 1 & -0.0025187 \\ 0.10483 & 0.37672 & -0.033186 & 0.014597 & -0.0025187 & 1 \end{pmatrix}$$

根据题目的要求，要求每一条旅游路线上至少应包含 3 至 6 个高品质的景点，因此我们可从表 5 中选出重庆、重庆、万州，合川、绵阳、都江堰和九寨沟作为旅游干线的可选对象。干线城市选择的结构示意图如下：

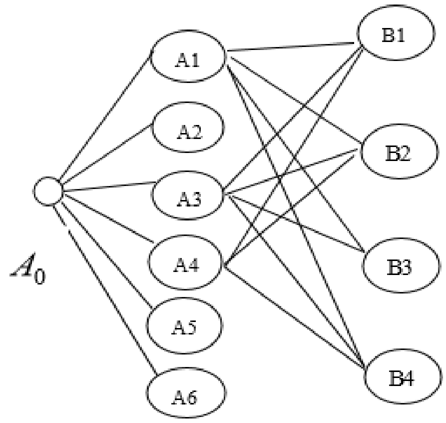


图 6　干线设计结构图

对于该问，沿用上几问的思想，我们的做法是在满足相应的约束条件下，先确定游览的景点数，然后分别表示出相应的旅游总费用和阴雨天气带来的旅游损失，归一化处理后加权求最小值。这样最终会得出几种最佳方案，而组织方可以根据自己的实际情况进行选择。由此得到最终的目标函数：

$$\text{Min}\quad \mathrm{Q}=\gamma_1\times C+\gamma_2\times L$$

(其中 C,L 如上所述，γ_1 γ_2 为权重且 $\gamma_1+\gamma_2=1$)

6.11 模型建立：

综上所述，我们可以得到总的模型为：

$$\text{Min}\quad \mathrm{Q}=\gamma_1\times C+\gamma_2\times L$$

在旅游路线干线确定好后，只要在各个旅游城市里设计好旅游路线就能解决问题。由于不考虑时间的约束，即旅游者游完十日之后就结束，因此我们只要考虑旅游路线最短和旅游花费最少两个目标即可。旅游线路越短，在线路上花费的时间也越少，因此可以保证在有限的时间内游玩更多的景点；对于旅游花费最主要包括住宿、餐饮、购物及娱乐等方面，但是在本文我们只研究景点的门票费和路费。旅游花费越低，旅游者的积极性就越高，也能为更多的旅游者所接受。

考虑到本问题是一个双目标规划问题，但是如果直接对旅游路线和旅游花费进行双目标规划求解，因为很难找到约束条件，而且计算的复杂性，所以很难求出 问题的准确解。通过进一步分析目标函数，发现路径最短问题可以直接转化为旅游路费最少问题，由于在前文已经假设了在海南省内旅游只坐空调旅游车，而做 空调旅游车的车费按约 1 元/（公里*人）计算，因此只要把路程 乘以空调旅游 S 车单位距离的车费就可得到路费。综上所述，我们可以把去某个景点的要花的路 费和景点的门票价加起来作为该景点的旅游花费。然后用蒙特卡罗改进的模拟退 火算法对问题进行求解。

模拟退火的基本思想：模拟退火算法来源于固体退火原理，将固体加温至

充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小，如对构成新解的全部或部分元素进行置换、互换等，注意到产生新解的变换方法 决定了当前新解的邻域结构，因而对冷却进度表的选取有一定的影响。 计算与新解所对应的目标函数差。因为目标函数差仅由变换部分产生，所以 目标函数差的计算最好按增量计算。事实表明，对大多数应用而言，这是计算目 标函数差的最快方法。当新解被确定接受时，用新解代替当前解，这只需将当前解中对应于产生新 解时的变换部分予以实现，同时修正目标函数值即可。此时，当前解实现了一次 迭代。可在此基础上开始下一轮试验。而当新解被判定为舍弃时，则在原当前解 的基础上继续下一轮试验。 模拟退火算法与初始值无关，算法求得的解与初始解状态 (是算法迭代的 S 起点)无关；模拟退火算法具有渐近收敛性，已在理论上被证明是一种以概率 l 收敛于全局最优解的全局优化算法；设某个旅游城市下有 个景点，用数码 代表。在进行路线设计时，n n …,2,1。每一个景点都有可能会游览其他 景点。因此，所有的旅游路线结构如下图所示：

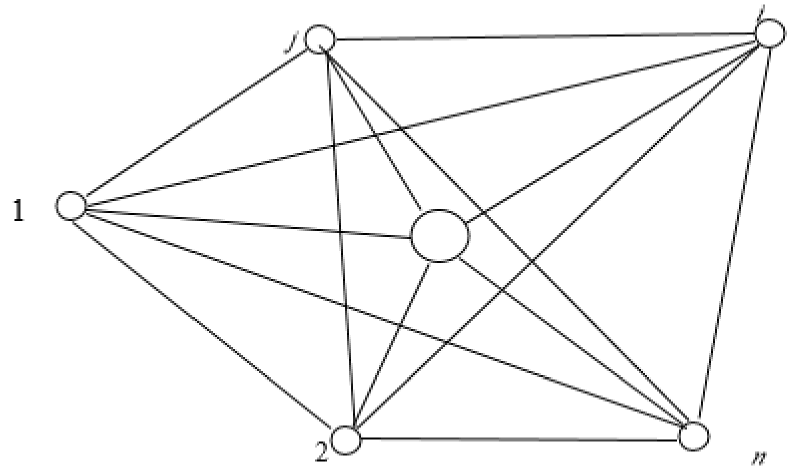


图 7　景点结构的的拓扑图

7 模型的评价、改进及推广

7.1. 模型的评价

1. 本文思路清晰，模型恰当，得出的方案合理；

2. 本文成功的使用了 0—1 变量，使模型的建立和编程得以顺利进行；

3. 在第二问中采用了 TCP 算法，简化了模型的求解难度；

4. 问题五由于数据庞大，对程序的要求很高，尽管经过了检验，但结果依然比较粗糙，有待进行进一步的改进。

5. 在建立费用最少模型时，我们把两景点间的交通费用、住宿费用、每天的基本费用和门票费用作为边 i 到 j 的赋权值；在建立时间最短模型时我们把两景点间的旅行费用作为边 i 到 j 的赋权值，分别构成了有向赋权图，巧妙地将原问题转化为图论的旅行商问题。

6. 本题所建立的数学模型是在 LINGO 环境下进行的。LINGO 软件是一个利用线性和非线性最优化方法将复杂的大型规划问题转化为简明公式的工具，具有简单实用的特点。在本题中，通过 LINGO 建立并求解了最优化模型，从而在可行解中得到了最佳结果。

7.2．模型的改进与推广：

1. 实际情况中，两景点之间可能还有出公路外其他交通方式，如航班、铁路，增加这些考虑后，结果会更加合理。
2. 因数据资料搜集的不完整，准确性也有待商榷，而且没有对最终方案进行更为细致的讨论研究，这些方面有待改进。

3. 旅游线路的优化设计，不仅在省时、省钱方面做到最优考虑，而且为在时间，金钱方面有制约的设计旅游线路最优化和最大化。真正是站在旅行者角度思考问题，为旅行者设计最合适最经济最省时的旅游线路。

8 参考文献

[1]姜启源 谢金星 叶俊，《数学模型（第三版）》，北京：高等教育出版社，2003。

[2]谢金星 薛毅，《优化建模与 LINDO/LINGO 软件》，北京：清华大学出版社，2005。

[3]周仁郁，《SPSS13.0 统计软件》，成都，西南交通大学出版社，2005。

[4]李庆扬 王能超 易大义，《数值分析》，北京：清华大学出版社 施普林格出版社，2001。

[5]曾建军 李世航等，《MATLAB 语言与数学建模》，合肥，安徽大学出版社，2005。

[6]甘应爱 田丰等，《运筹学》，北京，清华大学出版社，2005.6。

[7]阮晓青 周义仓等，《数学建模引论》，北京，高等教育出版社，2005.7。

9 附录

$\left[t_{ij}\right]_{11\times 11}=$

0	8.54	4.74	2.82	3.44	5.08	8.4	1.32	1.54	6.14	6.6
8.54	0	1.22	11.52	12.14	10.9	13.1	8.84	8.98	14.84	15.54
4.74	1.22	0	11.22	11.82	9.38	11.58	7.66	7.46	13.44	13.9
2.82	11.52	11.22	0	0.88	7.78	8.08	4.02	4.24	5.84	6.3
3.44	12.14	11.82	0.88	0	8.42	8.24	4.66	4.88	6	6.46
5.08	10.9	9.38	7.78	8.42	0	2.18	4.24	4.04	5.98	6.74
8.4	13.1	11.58	8.08	8.24	2.18	0	6.08	6.22	3.86	2.86
1.32	8.84	7.66	4.02	4.66	4.24	6.08	0	0.3	6.28	6.74
1.54	8.98	7.46	4.24	4.88	4.04	6.22	0.3	0	6.08	6.54
6.14	14.84	13.44	5.84	6	5.98	3.86	6.28	6.08	0	2.08
6.6	15.54	13.9	6.3	6.46	6.74	2.86	6.74	6.54	2.08	0

$\left[c_{ij}\right]_{11\times 11}=$

0	128	71	42	52	76	126	20	23	92	99
128	0	18	173	182	164	197	133	135	223	233
71	18	0	168	177	141	174	115	112	202	209
42	173	168	0	13	117	121	60	64	88	95
52	182	177	13	0	126	124	70	73	90	97
76	164	141	117	126	0	33	64	61	90	101
126	197	174	121	124	33	0	91	93	58	43
20	133	115	60	70	64	91	0	5	94	101
23	135	112	64	73	61	93	5	0	91	98
92	223	202	88	90	90	58	94	91	0	31
99	233	209	95	97	101	43	101	98	31	0

$\left[P_{is}\right]_{11\times 5}=$

0.15	0.1	0.3	0.8	0.7	0.5	0.6	0.3	0.1	0
1	0.8	0.8	0.5	0.5	0.4	0.5	0.6	0.3	0.2
0.5	1	0.9	1	0.3	0	0.2	0	0.4	0.4
0.1	0	0.1	0.5	0.5	0.7	0.3	0.3	0.1	0.1
0.3	0.4	0.4	0.6	0.3	0.3	0.2	0.4	0.6	0.6
0.6	0.6	0.6	0.5	0.8	0.3	0.1	0.1	0.1	1
0.3	0.2	0.2	0.1	0	0.1	0.1	0.2	0.2	0.3
0.4	0.3	0.3	0.2	0.4	0.9	0.9	0.9	0.8	0.8
0.5	0.3	0.3	0.4	0.3	0.8	1	0.9	0.9	0.9
0.2	0.6	0.6	0.4	0.1	0	0.8	0.7	0.6	0.4
0.1	0.1	0.3	0.3	0.5	0.6	0.8	0.7	0.63	0.4

```
model:
sets:
cities/1..11/:level; !level(i)=the level of city;
link(cities,cities):money,x;
endsets
data:!money matrix,it need not be symmetirc;
money=0 62 130 154 179 122 159 158 159 118 176
62 0 150 140 180 125 73 361 165 120 193
130 150 0 116 120 245 360 373 363 371 422
154 140 116 0 94 106 182 210 210 225 493
179 180 120 94 0 184 248 379 109 337 569
122 125 245 106 184 0 287 87 33 125 546
159 73 360 182 248 287 0 205 206 120 200
158 361 373 210 379 87 205 0 267 177 674
159 165 363 210 109 33 206 267 0 207 313
118 120 371 225 337 125 120 177 207 0 169
176 193 422 493 569 546 200 674 313 169 0;

enddata
n=@size(cities); !the model size;
p=160+ 70+ 50+ 40+ 80+ 150+ 50+ 90+ 180+ 160;
g=4+4+30+20+4+26+4+12+24+28;
T=60*10;
min=@sum(link(i,j)|i#ne#j:money(i,j)*x(i,j))+p+T+g;
@for(cities(i):
    @sum(cities(j)|j#ne#i:x(j,i))=1;
    @sum(cities(j)|j#ne#i:x(i,j))=1;
    @for(cities(j)|j#gt#1#and#j#ne#i:
    level(j)>=level(i)+x(i,j)-(n-2)*(1-x(i,j))+(n-3)*x(j,i);
);
);
@for(link:@bin(x));
@for(cities(i)|i#gt#1:
        level(i)<=n-1-(n-2)*x(1,i);
        level(i)>=1+(n-2)*x(i,1);
);
enddata
n=@size(cities); !the model size;
m=@sum(link(i,j)|i#ne#j:money(i,j)*x(i,j))+a+f;
a=@sum(link(i,j)|i#ne#j:p(i,j)*x(i,j))/2;

f=c*60;
```

```
c=@sum(link(i,j)|i#ne#j:x(i,j));
max=c/m;
m=2000;
@for(cities(i):
     @sum(cities(j)|j#ne#i:x(j,i))<=1;
     @sum(cities(j)|j#ne#i:x(i,j))<=1;
     @for(cities(j)|j#gt#1#and#j#ne#i:
     level(j)>=level(i)+x(i,j)-(n-2)*(1-x(i,j))+(n-3)*x(j,i);
);
);
@for(link:@bin(x));
@for(cities(i)|i#gt#1:
        level(i)<=n-1-(n-2)*x(1,i);
        level(i)>=1+(n-2)*x(i,1);
```