

评委一评分，签名及备注	队号： 10317	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：家庭暑期旅游套餐的设计

摘要

一次舒适经济的家庭旅行人们生活中孜孜不倦的追求。人类只是希望在相同的时间和空间条件下获取最大的价值和享受，旅游套餐的设计尤其是这样。

本文选取成都作为我们的旅游城市，并择优选取了 11 个热门景点作为选择。

根据不同家庭的需求，本文建立了三个独立的模型。

第一，玩遍所有景点的最经济方案。我们建立了改良圈算法模型，通过无向图的 Hamilton 回路建模，使用 Dijkstra 的改进算法将问题转化为单目标的非线性规划函数并得出了单一旅行商的问题。

第二，在一定预算下的景点最多方案。这个模型我们逆向思维，借鉴第一个模型中的算法，通过改变景点个数算出最经济方案，逆向给出一定预算下的最优方案。

第三，在时间限制下，用最少的钱游览最多的景点方案。这是一个典型的多目标规划模型，我们给出了时间，交通花费，景点个数以及 0-1 变量约束，实现了不同时间下的最优家庭旅行方案

三个方案几乎全面的满足不同家庭的需求。

在进行我们模型改进的思考时，我们提出了两个不同的改进模型——考虑景点意愿的综合模型和考虑天气风险的综合模型。进一步完善了家庭旅行套餐的设计。

最后，我们设计了 9 钟满足不同家庭需求的家庭旅行套餐。

序号	旅游名称	在川停留时间	最佳路线	人均花费
1	成都周边休闲周末游	2 天	成都→青城山→都江堰→成都	406
2	成都周边黄金三日游	3 天	成都→乐山→都江堰→青城山→成都	623
3	川内精品四日游	4 天	成都→都江堰→青城山→丹巴→乐山→成都	949
4	川西经典五日游	5 天	成都→都江堰→青城山→丹巴→康定→乐山→成都	1206
5	川西豪华六日游	6 天	成都→乐山→都江堰→青城山→丹巴→康定→黄龙→成都	1533
6	蜀地全景八日游	8 天	成都→乐山→海螺沟→康定→丹巴→四姑娘山→青城山→都江堰→成都	1923
7	四川深度九日游	9 天	成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→九寨沟→黄龙→成都	2389
8	四川深度十日游	10 天	成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→峨眉山→九寨沟→黄龙→成都	2783
9	四川自由全景游	12 天	成都→乐山→峨眉山→海螺沟→康定→丹巴→四姑娘山→青城山→都江堰→九寨沟→黄龙→成都	3242

关键字：改良圈 TCP 逆向思维 多目标线性规划 旅行套餐

目录

1. 问题重述..... - 2 -

2. 问题分析..... - 2 -

3. 模型假设..... - 2 -

4. 符号说明..... - 3 -

5. 模型建立与求解..... - 3 -

 模型一：玩遍景点所需费用最少方案..... - 3 -

 模型二、在一定预算下景点最多方案..... - 5 -

 模型三、在一定时间内游览尽可能多的景点并且花费最少 - 6 -

6. 模型的改进与推广 - 16 -

 6.1 模型改进方向..... - 16 -

 6.2 改进模型一、考虑家庭成员意愿度的改进模型 - 16 -

 6.3 改进模型二、考虑天气因素的综合模型。 - 17 -

7. 家庭暑期旅游线路推荐..... - 22 -

参考文献..... - 23 -

附录..... - 24 -

家庭暑期旅游套餐的设计

1. 问题重述

暑假是家庭出游最好的时间之一，在孩子放暑假的时候，很多家长会选择这个时间带孩子出门旅游，开拓孩子的视野，培养家人的感情。然而每个家庭都有自己对旅行不同的要求，比如家庭人口多少，家庭经济所能承受的旅行费用，家长假期长短所以对旅行时间的限制等。选取一个旅游城市，通过考虑旅行路线、费用、时间等其他重要因素为由不同要求的家庭设计不同的家庭暑期旅游套餐。

2. 问题分析

此问题是在一定约束条件下的离散型目标优化问题。我们的首要任务是确定家庭旅游过程中所考虑的不同因素，之后，我们将分别对这些因素进行有重点的考虑，通过数学模型进行旅行路线的优化。

通过日常生活中的观察以及网络上的调查，我们发现，大部分家庭在选择出行计划时关注的重点集中在旅行是否需要过多休假时间，是否具有高的性价比，以及旅行是否足够深度等三个问题上。由于不同的家庭的实际情况往往不同，我们分别用旅游时间、旅游费用、以及旅游景点数目这三个可量化的变量来阐述家庭所关注的三方面问题，并分别以之为优化重点建立数学模型给出优化的旅游线路。

在问题解决过程中，我们选取成都及其周围名胜景点作为我们的研究对象，也就是我们设计的家庭旅游计划的游览地。我们将建立三个模型分别给出需求不同的家庭最适合的游览方案。

三个模型分别针对将旅游景点数目、旅行费用、旅游时间作为首要考虑因素的家庭制定旅游计划，分别给出游览全部景点所需费用最少的方案，在给定旅行预算下所能游览最多景点的方案以及在给定旅行时间下最经济游览最多景点的方案。

3. 模型假设

1. 家庭成员我们视为相同的自然人，不存在景点游玩上的分歧，每人所用费用均摊，并且不考虑人数的差异；
2. 使用旅游大巴安排代表们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；
3. 所乘坐的旅游大巴平均时速为 50km/h，平均费用为 0.3 元/km；
4. 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；
5. 在限定的时间内，家庭成员从成都出发，最后并返回成都；
6. 游客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。

7. 整个旅行途中不存在任何意外情况的发生，比如天气突变，交通事故，景点关闭等等。我们模型中的旅行一切顺利。

8. 在综合考虑了成都周边的热门旅游景点后之后，我们从国家 5A 级景点以及部分 4A 级景点中选择了以下 11 个热门景点：成都、九寨沟、黄龙、乐山、峨眉山、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定，作为所有家庭旅游的候选目的地。

4. 符号说明

i, j ——第 i 个或者第 j 个景点， $i, j=1, 2, \dots, 11$;

分别表示成都、九寨沟、黄龙、乐山、峨眉山、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定；

c ——每个家庭成员的旅游总花费；

t_i ——每个家庭成员在第 i 个景点的逗留时间；

c_i ——每个家庭成员在 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；

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

5. 模型建立与求解

模型一：玩遍景点所需费用最少方案

1. 模型简介

在此模型中，我们将讨论旅游家庭在游遍所有 11 个景点的情况下花费最少的形成。在 11 个景点确定了的情况下，可以将 11 个景区看做 11 个点，该问题就抽象成了典型的货郎担（TSP）问题。

2. 模型建立

我们可以将家庭的旅行问题抽象成以下问题：

某商人欲往 n 个城市推销货物，从某个城市出发，沿途经过各个城市一次后

返回出发城市,要确定一条行走的路线,使得总路径最短。即为旅行商问题(TSP)。用图论的术语说,就是在一个赋权完全图中,找出一个有最小权的 Hamilton 圈 C 。称这种圈为最优圈。与最短路问题及连线问题相反,尽管目前还没有求解旅行商问题的有效算法。但是却有一个可行的办法是求一个 Hamilton 圈,然后适当修改以得到具有较小权的另一个 Hamilton 圈。修改的方法叫做改良圈算法。

设初始圈 $C = v_1 v_2 \cdots v_n v_1$ 。

(1) 对于 $1 < i+1 < j < n$, 构造新的 Hamilton 圈:

$$C_{ij} = v_1 v_2 \cdots v_i v_j v_{j-1} v_{j-2} \cdots v_{i+1} v_{j+1} v_{j+2} \cdots v_n v_1,$$

它是由 C 中删去的边 $v_i v_{i+1}$ 和 $v_j v_{j+1}$, 添加边 $v_i v_j$ 和 $v_{i+1} v_{j+1}$ 而得到的。若 $w(v_i v_j) + w(v_{i+1} v_{j+1}) < w(v_i v_{i+1}) + w(v_j v_{j+1})$, 则以 C_{ij} 代替 C , C_{ij} 叫做 C 的改良圈。

不断由 (2) 转到 (1), 直至无法改进, 停止。

用改良圈算法得到的结果几乎可以肯定不是最优的。为了得到更高的精确度, 在不给定起始位置的前提下, 可以选择不同的初始圈, 重复进行 n 次算法, 以求得精确的结果。

由此, 我们用如下数学表达对问题抽象:

设城市的个数为 n , d_{ij} 是两个景点 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)。则有

$$\begin{aligned} \min & \sum_{i \neq j} d_{ij} x_{ij} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, (\text{每个点只有一条边出去}) \\ & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, (\text{每个点只有一条边出去}) \\ & \sum_{i, j \in s} x_{ij} \leq |s| - 1, \quad 2 \leq |s| \leq n-1, \quad s \subset \{1, 2, \dots, n\} \\ & (\text{各起点和终点外, 各边不构成圈}) \end{aligned}$$

3. 模型求解

在我们的模型一中, 我们将旅行商模型中的权值“最短距离”换为“最少花费”。利用下面的分段函数求出花费最小的矩阵:

$$F(i, j) = 0.3 \times t_{ij}$$

我们规定了假设中以成都为起点玩遍 11 个景点。因此利用 $F(i, j)_{11 \times 11}$ 为

$w_2(i, j)_{11 \times 11}$ 构造无向图 UG_2 , 再利用 Matlab 软件进行 1 次改良圈算法 (算法见附录), 从而有游览 11 个景点的最佳路线如下:

最佳路线: 成都 \rightarrow 乐山 \rightarrow 峨眉山 \rightarrow 海螺沟 \rightarrow 康定 \rightarrow 丹巴 \rightarrow 四姑娘山 \rightarrow 青城山 \rightarrow 都江堰 \rightarrow 九寨沟 \rightarrow 黄龙 \rightarrow 成都

此时, 花费为 3242.4 元



模型二、在一定预算下景点最多方案

1. 模型简介

在这一阶段的模型中，我们将把家庭的经济状况作为制定旅行计划的首要考虑因素。通过这一模型，我们将给出在家庭预算一定的情况下，某一家庭所能游览的最多景点数。

2. 模型建立

在模型实现过程中，我们将逆向思考，先分别求解游览 $n(n=2,3,...,11)$ 个景点所需要的最小花费，之后通过该结果分别设定对应的家庭预算，并最终给出给定预算下所可能的游览景点数。

为了实现该模型，我们依旧可以运用模型一中的货郎担问题的解决方法，分别更改上述模型所要到达的城市数，计算不同城市数下所需要的最少费用。因此，该模型实现过程中的算法与模型一大致相同。

3. 模型实现

通过计算有，游览 n 个景点的最小花费计算结果如下：

景点数	3	4	5	6	7	8	9	10	11
花费	405.4	623.2	949.1	1206.3	1533.6	1923.6	2389.8	2783.4	3242.4

由此，我们对旅游预算与旅游路线的推荐对应关系如下：

当旅游预算在 800 元/人时，我们推荐景点在 3~4 个的简易游览线路，包括：

成都->青城山->都江堰->成都

成都->乐山->青城山->都江堰->成都

当旅游预算在 1500 元/人时，我们推荐景点在 5~6 个的轻松游览线路，包括：

成都->都江堰->青城山->丹巴->乐山->成都

成都->都江堰->青城山->丹巴->康定->乐山->成都

当旅游预算在 2000 元/人时，我们推荐景点在 7~8 个的经典游览线路，包括：

成都->乐山->海螺沟->康定->丹巴->青城山->都江堰->成都

成都->乐山->海螺沟->康定->丹巴->四姑娘山->青城山->都江堰->成都

当旅游预算在 2500 元/人时，我们推荐景点在 9~10 个的深度游览线路，包括：

成都->黄龙->青城山->都江堰->四姑娘山->丹巴->康定->海螺沟->乐山->成都

成都->乐山->峨眉山->海螺沟->康定->丹巴->四姑娘山->青城山->都江堰

->黄龙->成都

当旅游预算在 3500 元/人时，我们推荐景点为 11 个的全面游览线路，即为：

成都->乐山->峨眉山->海螺沟->康定->丹巴->四姑娘山->青城山->都江堰
->九寨沟->黄龙->成都

模型三、在一定时间内游览尽可能多的景点并且花费最少。

1. 模型简介

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最小花费。这样最终会得出几种旅游路线，而具体家庭可以根据自己的实际情况进行选择。

2. 模型建立

i, j ——第 i 个或者第 j 个景点， $i, j=1, 2, \dots, 11$;

分别表示成都、九寨沟、黄龙、乐山、峨眉山、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定；

c ——每个家庭成员的旅游平均总花费；

t_i ——每个家庭成员在第 i 个景点的逗留时间；

c_i ——每个家庭成员在 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；

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

我们规定：旅游的总费用由 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 ——每个家庭成员的旅游平均总花费；

m_1 ——每个家庭成员的交通平均总费用；

m_2 ——每个家庭成员在旅游景点的平均花费；

从而得到目标函数： $\text{Min } m = m_1 + m_2$

(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} 是判断代

表们是否从第*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，因此 $\sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times (c_i + c_j)$ 实际上将家庭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：

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

很明显，旅游的花费与所用的时间以及所去景点的数目有关，这就需要对我们的目标函数进行一定的约束：

1) 时间约束

我们假设家庭计划在四川旅游的总时间是 T。根据模型假设， t_{ij} 表示从第*i*个景点到第*j*个景点路途所需时间，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 表示家庭在第*i*个景点的游览时间，故家庭在旅游景点的总

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：

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

2) 旅游景点数约束

根据假设，整个旅游路线是环形，即最终家庭要回到成都，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

($n=2, 3, \dots, 11$)。因此旅游景点数约束为：

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

3) 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。所有景点构成的旅游路线图就是一个有向图。所有的景点必须要形成一个欧拉回路，因此可得约束：

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11)$$

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

$j=1$ 时，因为最终要回到成都，所以 $\sum_{j=1} r_{ij} = 1$ 。

综合以上可知，

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11)$$

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

同样，当 $i, j \geq 2$ 时，根据假设三不可能出现 $r_{ij} = r_{ji} = 1$ 。

因此我们可得约束：

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

综上所述，我们建立的数学模型为：

$$\text{Min } m = m_1 + m_2$$

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

约束条件：

$$\left\{ \begin{array}{l} \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times (t_i + t_j) \leq T \\ \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} = n \quad (n=2, 3, \dots, 11) \\ \sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11) \\ \sum_{i=1} r_{ij} = 1 \quad \sum_{j=1} r_{ij} = 1 \end{array} \right.$$

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

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

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

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

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)

3. 模型求解

利用 LINGO 分别对不同旅行时间限制下的旅行方案求解，结果如下：

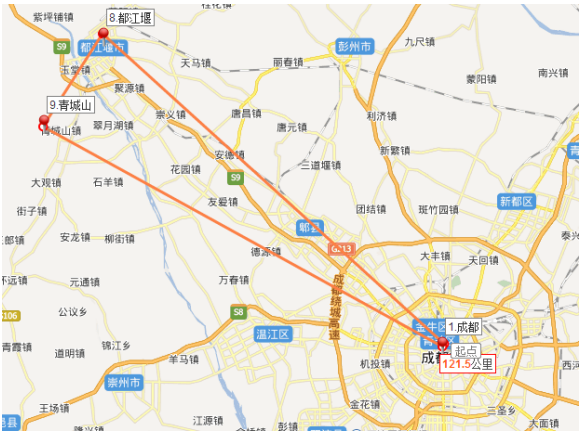
游览时间是40小时

游览3个景点

Global optimal solution found.

Objective value:		405.4000
R(1, 9)	1.000000	157.1000
R(8, 1)	1.000000	124.8000
R(9, 8)	1.000000	123.5000

最佳路线：成都->青城山->都江堰->成都



游览时间是 60 小时

游览 4 个景点

Global optimal solution found.

Objective value:		623.2000
R(1, 4)	1.000000	169.8000
R(4, 9)	1.000000	205.1000
R(8, 1)	1.000000	124.8000
R(9, 8)	1.000000	123.5000

最佳路线：成都->乐山->青城山->都江堰->成都



游览时间是 80 小时

游览5个景点

Objective value:		949.1000
Variable	Value	Reduced Cost
R(1, 8)	1.000000	124.8000

R(4, 1)	1.000000	169.8000
R(7, 4)	1.000000	276.2000
R(8, 9)	1.000000	123.5000
R(9, 7)	1.000000	254.8000

最佳路线：成都—>都江堰—>青城山—>丹巴—>乐山—>成都



游览时间是 100 小时
游览6个景点

Global optimal solution found.

Objective value: 1206.300

R(1, 8)	1.000000	124.8000
R(4, 1)	1.000000	169.8000
R(7, 11)	1.000000	250.9000
R(8, 9)	1.000000	123.5000
R(9, 7)	1.000000	254.8000
R(11, 4)	1.000000	282.5000

最佳路线：成都—>都江堰—>青城山—>丹巴—>康定—>乐山—>成都



游览时间是 120 小时
游览7个景点

Objective value:		1533.600
R(1, 4)	1.000000	169.8000
R(4, 10)	1.000000	306.6000
R(7, 9)	1.000000	254.8000
R(8, 1)	1.000000	124.8000
R(9, 8)	1.000000	123.5000
R(10, 11)	1.000000	303.2000
R(11, 7)	1.000000	250.9000

最佳路线：成都—>乐山—>海螺沟—>康定—>丹巴—>青城山—>都江堰—>成都



游览时间是 160 小时
游览8个景点

Objective value:		1923.600
R(1, 4)	1.000000	169.8000
R(4, 10)	1.000000	306.6000
R(6, 9)	1.000000	329.6000
R(7, 6)	1.000000	315.2000
R(8, 1)	1.000000	124.8000
R(9, 8)	1.000000	123.5000
R(10, 11)	1.000000	303.2000
R(11, 7)	1.000000	250.9000

最佳路线：成都—>乐山—>海螺沟—>康定—>丹巴—>四姑娘山—>青城山—>都江堰—>成都



游览时间是 180 小时

游览9个景点

Objective value:

2389.800

R(1, 3)	1.000000	281.1000
R(3, 9)	1.000000	335.9000
R(4, 1)	1.000000	169.8000
R(6, 7)	1.000000	315.2000
R(7, 11)	1.000000	250.9000
R(8, 6)	1.000000	303.6000
R(9, 8)	1.000000	123.5000
R(10, 4)	1.000000	306.6000
R(11, 10)	1.000000	303.2000

最佳路线：成都—>黄龙—>青城山—>都江堰—>四姑娘山—>丹巴—>康定—>海螺沟—>乐山—>成都



游览时间是 220 小时
游览10个景点

Objective value:		2783.400
R(1, 4)	1.000000	169.8000
R(3, 1)	1.000000	281.1000
R(4, 5)	1.000000	269.7000
R(5, 10)	1.000000	430.5000
R(6, 9)	1.000000	329.6000
R(7, 6)	1.000000	315.2000
R(8, 3)	1.000000	309.9000
R(9, 8)	1.000000	123.5000
R(10, 11)	1.000000	303.2000
R(11, 7)	1.000000	250.9000

最佳路线：成都—>乐山—>峨眉山—>海螺沟—>康定—>丹巴—>四姑娘山—>青城山—>都江堰—>黄龙—>成都



游览时间是 240 小时
游览11个景点

Objective value:		3242.400
R(1, 4)	1.000000	169.8000
R(2, 3)	1.000000	379.8000

R(3, 1)	1.000000	281.1000
R(4, 5)	1.000000	269.7000
R(5, 10)	1.000000	430.5000
R(6, 9)	1.000000	329.6000
R(7, 6)	1.000000	315.2000
R(8, 2)	1.000000	389.1000
R(9, 8)	1.000000	123.5000
R(10, 11)	1.000000	303.2000
R(11, 7)	1.000000	250.9000

最佳路线：成都→乐山→峨眉山→海螺沟→康定→丹巴→四姑娘山→青城山→都江堰→九寨沟→黄龙→成都



6. 模型的改进与推广

6.1 模型改进方向

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. 家庭成员中可能会有小孩和老人，本文模型对于这方面并没有考虑各年龄段对于景点的要求。
 3. 模型对于家庭旅行的不同需求可能还有遗漏。在现实选择旅行计划时，会根据家庭意愿进行景点，费用以及时间的选择
 4. 在实际旅行中，天气因素并未考虑。而天气是影响旅行行程一个很重要的客观因素。我们可以加入天气因素来改进我们的模型。
- 下面我们提出我们的改进综合模型。

6.2 改进模型一、考虑家庭成员意愿度的改进模型

6.2.1 模型简介

在此改进模型中，我们将会根据家庭成员对不同旅游景点的需求以及不同旅游景点的热门程度确定各景点的评级函数，并由此确认更令人满意的旅行路线。

6.2.2 模型建立

Step 1: 评级函数:

成都(4)、九寨沟(5)、黄龙(4)、乐山(3)、峨嵋(4)、四姑娘山(4)、丹巴(3)、都江堰(3)、青城山(3)、海螺沟(4)、康定(4)

根据旅行社以及国家景区评定的相关标准，得到1~11景区的评级矩阵

$$A = [4, 5, 4, 3, 4, 4, 3, 3, 3, 4, 4]$$

再根据公式: $W_i = \frac{\text{sum}(A)}{A_i}$ 算出对应景点的评分（分数越低表示该景点越热门），可以得到11个景点的评分矩阵w:

$$w = [10.25, 8.2, 10.25, 13.67, 10.25, 10.25, 13.67, 13.67, 13.67, 10.25, 10.25]$$

Step 2: 目标函数:

根据前面模型的结果，我们可以得到目标函数:

$$\text{Min } m = m_1 + m_2$$

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

Step 3: 约束条件:

此部分与之前模型约束条件部分相类似，在此不再赘述。

6.2.3 模型求解

对于上述结果，我们的推荐为:

游览5个景点的路线为:

成都→乐山→都江堰→青城山→丹巴→成都

游览6个景点的路线为:

成都→乐山→都江堰→青城山→丹巴→康定→成都

游览7个景点的推荐路线为:

成都→乐山→都江堰→青城山→丹巴→康定→黄龙→成都

游览8个景点的路线为:

成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→黄龙→成都

游览9个景点的路线为:

成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→九寨沟→黄龙
→成都

游览10个景点的路线为:

成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→峨眉山→九寨
沟→黄龙→成都

6.3 改进模型二、考虑天气因素的综合模型。

6.3.1 模型准备

6.3.1.1 符号引入:

l ——阴雨天气带来的旅游损失;

$c(n)_{\min}$ ——家庭成员们旅游 n 个景点需要的最小的花费;

$c(n)_{\max}$ ——家庭成员们旅游 n 个景点需要的最大的花费;

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

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

6.3.1.2 问题的进一步分析

本问在模型三的基础上考虑了天气的因素,相应的也就增加了一个目标即:使因阴雨天气而带来的旅游损失降到最低。对于旅游损失,我们定义为家庭成员们在景点逗留时所对应的阴雨天气概率的总和。

6.3.1.3 数据处理

(1) 通过查找相关数据,我们得出 11 个景点在暑假 10 天内降水概率

$[P_{is}]_{11 \times 10} =$									
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

(2) 数据的归一化处理

通过观察数据，我们发现旅游总花费和阴雨天气带来的旅游损失的数值差距较大，在利用二者综合确立目标时，为了避免其的影响，采用数据常用处理方法——极差变化法，将数据做归一化处理。即：

$$C = \frac{c - c(n)_{\min}}{c(n)_{\max} - c(n)_{\min}} ; \quad L = \frac{l - l(n)_{\min}}{l(n)_{\max} - l(n)_{\min}}$$

(3) 确定目标函数

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：

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

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

I. 对于 C ：

$$c = 100 \times \sum_{i=1}^{11} \sum_{j=1}^{11} \lambda_i \times r_{ij} \times c_{ij} + \frac{1}{2} \times 100 \times \sum_{i=1}^{11} \sum_{j=1}^{11} \lambda_i \times r_{ij} \times (c_i + c_j), \text{ 而相应的}$$

$c(n)_{\min}$ 与 $c(n)_{\max}$ 也可以根据前几问的模型计算出（具体值见附录表格），因此我们用可以已知的结果来表达 C 。

II. 对于 L ：

在 L 的表达式中，关键是表示出 l ，表示出 l 后 $l(n)_{\min}$ 和 $l(n)_{\max}$ 可以根据相应的模型通过编程很容易的计算出来。因为 l 表示阴雨天气带来的旅游损失，而我们定义该损失为家庭成员们在景点逗留时所对应的阴雨天气概率的总和。这里我们设：

P_{is} ——第*i*个景点在第*s*天阴雨的的概率；（ $s=1, 2, \dots, 10$ ）

T_i ——家庭成员们到达第*i*个景点的时间；

V ——家庭成员们游览的景点的集合，如 $V=\{1, 8, 5, 7\}$ 表示家庭成员们游览了第 1、8、5、7 个景点。

因此家庭成员们会从第 $[T_i]$ 天到第 $[T_i+t_i]$ 天都留在第*i*个景点（ $[]$ 表示取

整）。综上， l 的可以表示为 $\sum_{i \in V} \sum_{s=[T_i]}^{[T_i+t_i]} P_{is}$ ，这样我们也就可以表达出 L 。

6.3.2 约束条件

①时间约束

为简化模型，我们假定在川的旅游时间不多于 10 天，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

景点路途所需时间，所以路途所需总时间为 $\sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times t_{ij}$ ； t_i 表示会议家

庭成员们在第*i*个景点的逗留时间，故家庭成员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：

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

②旅游景点数约束

根据假设，整个旅游路线是环形，即最终家庭成员们要回到成都，因

此 $\sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij}$ 即表示家庭成员们旅游的景点数，这里我们假定要旅游的景点数

为 n （ $n=2, 3, \dots, 11$ ）。因此旅游景点数约束为：

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

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：

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11)$$

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

当 $j=1$ 时，因为家庭成员们最终要回到成都，所以 $\sum_{j=1} r_{ij} = 1$ 。

综合以上可知，

$$\sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11)$$

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

同样，当 $i, j \geq 2$ 时，根据题意不可能出现 $r_{ij} = r_{ji} = 1$ ，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：

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

④对 T_i 的约束

因为 T_i 表示家庭成员们到达第 i 个景点的时间，因此 T_i 应该等于到达第 $i-1$ 个景点的时间与第 $i-1$ 个景点到第 i 个景点途中时间的和。从而可得：

$$T_i = T_{i-1} + t_{i-1,i} \quad (i=2, 3, \dots, 11)$$

6.3.3 模型建立：

综上所述，我们可以得到总的模型为：

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

约束条件：

$$\left\{ \begin{array}{l} \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times t_{ij} + \frac{1}{2} \times \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} \times (t_i + t_j) \leq 120 \\ \sum_{i=1}^{11} \sum_{j=1}^{11} r_{ij} = k \\ \sum_i r_{ij} = \sum_j r_{ij} \leq 1 \quad (i, j=1, 2, \dots, 11) \\ r_{ij} \times r_{ji} = 0 \quad (i, j \geq 2) \\ T_i = T_{i-1} + t_{i-1,i} \quad (i=2, 3, \dots, 11) \end{array} \right.$$

6.3.4 模型求解与结果分析：

由于结果数据庞大，我们在这里只列出 $n=5$ 的情形。

当 $\gamma_1 = 0.6, \gamma_2 = 0.4$ 时

人均总费用 C (单位: 元)	损失
965	1.7
路线: $1 \rightarrow 11 \rightarrow 7 \rightarrow 4 \rightarrow 8 \rightarrow 1$	

当 $\gamma_1 = 0.4, \gamma_2 = 0.6$ 时

人均总费用 C (单位: 元)	损失
1028	1.6
路线: $1 \rightarrow 7 \rightarrow 9 \rightarrow 8 \rightarrow 4 \rightarrow 1$	

当 $\gamma_1 = 0.7, \gamma_2 = 0.3$ 时

人均总费用 C (单位: 元)	损失
946	1.8
路线: $1 \rightarrow 10 \rightarrow 11 \rightarrow 7 \rightarrow 4 \rightarrow 1$	

当 $\gamma_1 = 0.3, \gamma_2 = 0.7$ 时

人均总费用 C (单位: 元)	损失
1206	0.9
路线: $1 \rightarrow 11 \rightarrow 9 \rightarrow 8 \rightarrow 4 \rightarrow 1$	

当 $\gamma_1 = 0.5, \gamma_2 = 0.5$ 时

人均总费用 C (单位: 元)	损失
987	1.6
路线: $1 \rightarrow 11 \rightarrow 9 \rightarrow 8 \rightarrow 4 \rightarrow 1$	

(其中数字 1—11 分别表示成都、九寨沟、黄龙、乐山、峨嵋、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定)

由以上结果, 我们推荐路线为: 成都→康定→青城山→都江堰→乐山→成都, 相应人均消费 987 元, 阴雨天气带来的损失为 1.6。



7. 家庭暑期旅游线路推荐

在此部分中，我们综合之前所有模型的求解，为家庭暑期旅游线路提供如下备选方案。

序号	旅游名称	在川停留时间	最佳路线	人均花费
1	成都周边休闲周末游	2 天	成都→青城山→都江堰→成都	406
2	成都周边黄金三日游	3 天	成都→乐山→都江堰→青城山→成都	623
3	川内精品四日游	4 天	成都→都江堰→青城山→丹巴→乐山→成都	949
4	川西经典五日游	5 天	成都→都江堰→青城山→丹巴→康定→乐山→成都	1206
5	川西豪华六日游	6 天	成都→乐山→都江堰→青城山→丹巴→康定→黄龙→成都	1533
6	蜀地全景八日游	8 天	成都→乐山→海螺沟→康定→丹巴→四姑娘山→青城山→都江堰→成都	1923
7	四川深度九日游	9 天	成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→九寨沟→黄龙→成都	2389
8	四川深度十日游	10 天	成都→乐山→都江堰→青城山→丹巴→康定→海螺沟→峨眉山→九寨沟→黄龙→成都	2783
9	四川自由全景游	12 天	成都→乐山→峨眉山→海螺沟→康定→丹巴→四姑娘山→青城山→都江堰→九寨沟→黄龙→成都	3242

参考文献

- [1]王振龙, 时间序列分析, 北京: 中国统计出版社, 2002 年。
- [2]韩中庚, 长江水质综合评价与预测的数学模型, 工程数学学报第 22 卷第 7 期:
P47-P52, 2005 年
- [3] 张超, 杨奕, 李嵩, 关于灾情巡视路线的模型: 1~3 页, 1998。
- [4] 耿素云, 集合论与图论, 北京: 北京大学出版社, 1998。
- [5] 边馥萍, 侯文化, 梁冯珍, 北京: 高等教育出版社, 2005。
- [6] 赵静 但琦, 数学建模与数学实验, 高等教育出版社, 2001;
- [7]栗雪娟, 崔尚森, 张柯, 最佳旅游路线选择的神经网络方法, 交通与计算机
2006 年第 5 期 第 24 卷 总 132 期
- [8][日]田村坦之, 系统工程, 北京: 科学出版社, 2001
- [9]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。
- [10]谢金星 薛毅, 《优化建模与 LINDO/LINGO 软件》, 北京: 清华大学出版社,
2005。
- [11]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

附录

模型一程序:

```
clc,clear
%主函数:
function main
load mydata
global d
d=zeros(11);
C=zeros(11,13);
for i=1:11
for j=1:11
d(i,j)=0.3*t(i,j);
end
end
[i,j,y]=find(d);
a=sparse(i,j,y);
a=tril(a);
L=size(d,1)
for i=1:11
c1=[i 1:11 11];
[ circle, long]=modifycircle(c1,L);
c2=[i 11 1:11];%改变初始圈,该算法的最后一个顶点不动
[ circle2, long2]=modifycircle(c2,L);
if long2<long
long=long2;
circle=circle2;
end
changdu(i)=long;
C(i,:)=circle;
circle, long
end
[zuiyou,I]=min(changdu);
circle0=C(I,:)
x=[x0(circle0(1:end-1)) x0(51)];
y=[y0(circle0(1:end-1)) y0(51)];
scatter(x,y,'*')
%被调用函数:
function [circle, long]=modifycircle(c1,L);
global d
flag=1;
while flag>0
flag=0;
```

```

for m=1:L-3
for n=m+2:L-1
if d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<...
d(c1(m),c1(m+1))+d(c1(n),c1(n+1))
flag=1;
c1(m+1:n)=c1(n:-1:m+1);
end
end
end
end
long=d(c1(1),c1(L));
for i=1:L-1
long=long+d(c1(i),c1(i+1));
end
circle=c1;

```

模型二程序:

sets:

jingdian/1..11/:c,t,l;

!其中: 1,2,...,11 分别代表成都、九寨沟、黄龙、乐山、峨嵋、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定; c, t 分别表示旅行团在各景点的吃住消费和逗留时间;; l 是为了控制不出现两个以上环形回路而设的一个变量;

links(jingdian,jingdian):r,cc,tt;

!其中: r 为 0-1 变量 (0 表示两景点不相连, 1 表示两景点相通); cc 为两景点之间的交通费用; tt 为两景点之间的交通时间;

endsets

data:

t=7 24 18 12 36 30 12 9 15 24 17;

c=120 423 300 135 378 390 175 90 148 303 241;

tt=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;

!其中: 主对角线为零,表示各景点到自身交通费用为零;

cc=0128 71 42 52 77 126 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;

```

!其中：主对角线为零,表示各景点到自身的交通时间为零;

enddata

min=@sum(jingdian(j):@sum(jingdian(i):r(i,j)*(cc(i,j)+0.5*(c(i)+c(j)))));

!目标函数：表示计划游玩的景点数目为 n 时的最小费用;

@for(jingdian(i):r(i,i)=0);

!约束条件：表示各景点到自身没有路线相连的约束条件;

@for(jingdian(i)|i#ge#2:@for(jingdian(j)|j#ge#2:r(i,j)+r(j,i)<1));

!约束条件：表示除起点（成都）外，若旅行团从景点 i 到景点 j 去游玩（即 r(i,j)=1），则不会从景点 j 到景点 i 去游玩（即 r(j,i)=0），也就是说除起点外每个景点只游玩一次;

a=@sum(jingdian(j):@sum(jingdian(i):r(i,j)*(tt(i,j)+0.5*(t(i)+t(j)))));

@sum(jingdian(j):@sum(jingdian(i):r(i,j)*(tt(i,j)+0.5*(t(i)+t(j))))<360;

!约束条件:表示总的旅行时间（交通时间和景点逗留时间）不超过给定时间 30 天 360 小时;

@for(jingdian(i):@sum(jingdian(j):r(i,j))=@sum(jingdian(j):r(j,i)));

@for(jingdian(i)|i#eq#1:@sum(jingdian(j):r(i,j))=1);

@for(jingdian(i)|i#ne#1:@sum(jingdian(j):r(i,j))<1);

!这三个约束条件：表示起点（成都）有且仅有一条路线出去和一条路线进来，其它景点要么有且仅有一条路线出去和一条路线进来，要么既没有路线出去也没有路线进来;

@for(links:@bin(r));

!约束条件:表示 0-1 变量约束;

@sum(jingdian(j):@sum(jingdian(i):r(i,j)))=11;

!约束条件:表示旅游景点的数目为 n 的约束;

@for(jingdian(i):@for(jingdian(j)|j#gt#1#and#j#ne#i:l(j))>=l(i)+r(i,j)-(n-2)*(1-r(i,j))+(n-3)*r(j,i));

@for(jingdian(i)|i#gt#1:l(i)<n-1-(n-2)*r(1,i);l(i)>1+(n-2)*r(i,1));

!这两个约束条件：为了控制不出现两个以上环形回路，保证有且仅有一条环形路线;

模型三程序:

sets:

jingdian/1..11/:c,t,l;

!其中：1,2,...,13 分别代表成都、九寨沟、黄龙、乐山、峨嵋、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定；c, t 分别表示旅行团在各景点的吃住消费和逗留时间；w 表示各景点选择性权重；1 是为了控制不出现两个以上环形回路而设的一个变量;

links(time,time):r,feijiang,shijian;

!其中：r 为 0-1 变量（0 表示两景点不相连，1 表示两景点相通）；feiyong 为两景点之间的交通费用；shijian 为两景点之间的交通时间;

```

endsets
data:
t=7 24 18 12 36 30 12 9 15 24 17;
c=120 423 300 135 378 390 175 90 148 303 241;
shijian=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;
!主对角线为零,表示各景点到自身交通费用为零;
feiyong=0 128.1 71.1 42.3 51.6 76.2 126 19.8 23.1 92.1 99
128.1 0 18.3 172.8 182.1 163.5 196.5 132.6 134.7 222.6 233.1
71.1 18.3 0 168.3 177.3 140.7 173.7 114.9 111.9 201.6 208.5
42.3 172.8 168.3 0 13.2 116.7 121.2 60.3 63.6 87.6 94.5
51.6 182.1 177.3 13.2 0 126.3 123.6 69.9 73.2 90 96.9
76.2 163.5 140.7 116.7 126.3 0 32.7 63.6 60.6 89.7 101.1
126 196.5 173.7 121.2 123.6 32.7 0 91.2 93.3 57.9 42.9
19.8 132.6 114.9 60.3 69.9 63.6 91.2 0 4.5 94.2 101.1
23.1 134.7 111.9 63.6 73.2 60.6 93.3 4.5 0 91.2 98.1
92.1 222.6 201.6 87.6 90 89.7 57.9 94.2 91.2 0 31.2
99 233.1 208.5 94.5 96.9 101.1 42.9 101.1 98.1 31.2 0;
!主对角线为零,表示各景点到自身的交通时间为零;
n=?;
T=?;
!其中: n 表示计划游玩的景点数目;
!T 表示家庭游玩的时间
enddata
min=@sum(time(j):@sum(time(i):r(i,j)*(cc(i,j)+0.5*(c(i)+c(j)))));
!表示计划游玩的景点数目为 n 时的最小费用;
@for(time(i):r(i,i)=0);
!约束条件: 表示各景点到自身没有路线相连的约束条件;
@for(time(i)|i#ge#2:@for(time(j)|j#ge#2:r(i,j)+r(j,i)<1));
!约束条件: 表示除起点(成都)外,若旅行团从景点 i 到景点 j 去游玩(即 r(i,j)=1),则不会
再从景点 j 到景点 i 去游玩(即 r(j,i)=0),也就是说除起点外每个景点只游玩一次;
a=@sum(time(j):@sum(time(i):r(i,j)*(time(i,j)+0.5*(t(i)+t(j)))));
@sum(v(j):@sum(v(i):r(i,j)*(time(i,j)+0.5*(t(i)+t(j))))<T;
!表示总的旅行时间(交通时间和景点逗留时间)不超过给定时间 T;
@for(time(i):@sum(time(j):r(i,j))=@sum(time(j):r(j,i)));
@for(time(i)|i#eq#1:@sum(time(j):r(i,j))=1);

```

```

@for(time(i)|i#ne#1:@sum(time(j):r(i,j))<1);
!表示起点（成都）有且仅有一条路线出去和一条路线进来，其它景点要么有且仅有一条路线出去和一条路线进来，要么既没有路线出去也没有路线进来;
@for(links:@bin(r));
!约束条件:表示 0-1 变量约束;
@sum(time(j):@sum(time(i):r(i,j)))=n;
!约束条件:表示旅游景点的数目为 n 的约束;
@for(time(i):@for(time(j)|j#gt#1#and#j#ne#i:l(j)>=l(i)+r(i,j)-(n-2)*(1-r(i,j))+(n-3)*r(j,i)));
@for(time(i)|j#gt#1:l(i)<n-1-(n-2)*r(1,i);l(i)>1+(n-2)*r(i,1));
!为了控制不出现两个以上环形回路，保证有且仅有一条环形路线;

```

改进模型一程序:

sets:

jingdian/1..11/:c,t,w,l;

!其中: 1, 2, ..., 11 分别代表成都、九寨沟、黄龙、乐山、峨嵋、四姑娘山、丹巴、都江堰、青城山、海螺沟、康定; c, t 分别表示旅行团在各景点的吃住消费和逗留时间; w 表示各景点选择性权重; l 是为了控制不出现两个以上环形回路而设的一个变量;

links(jingdian,jingdian):r,cc,tt;

!其中: r 为 0-1 变量 (0 表示两景点不相连, 1 表示两景点相通); cc 为两景点之间的交通费用; tt 为两景点之间的交通时间;

endsets

data:

t=7 24 18 12 36 30 12 9 15 24 17; !其中: t(1)=0, 表示在成都的逗留时间为 0, 因为相对旅行团而言, 由于成都既是起点又是终点, 并未在成都游玩;

c=120 423 300 135 378 390 175 90 148 303 241; !其中: c(1)=0, 表示在成都的吃住费用算做, 理由同上;

w=10.25 8.2 10.25 13.67 10.25 10.25 13.67 13.67 13.67 10.25 10.25;

tt=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;

!其中: 主对角线为零, 表示各景点到自身交通费用为零;

cc=0	128	71	42	52	77	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;

```

!其中：主对角线为零,表示各景点到自身的交通时间为零;

n=?;!其中：n表示计划游玩的景点数目;

enddata

min=@sum(jingdian(j):@sum(jingdian(i):100*w(j)*r(i,j)*(cc(i,j)+0.5*100*(c(i)+c(j)))));!目标函数：表示计划游玩的景点数目为n时的

最小费用;

b=@sum(jingdian(j):@sum(jingdian(i):r(i,j)*(cc(i,j)+0.45*(c(i)+c(j)))));

@for(jingdian(i):r(i,i)=0);!约束条件：表示各景点到自身没有路线相连的约束条件;

@for(jingdian(i)|i#ge#2:@for(jingdian(j)|j#ge#2:r(i,j)+r(j,i)<1));!约束条件：表示除起点(成都)外,若旅行团从景点i到景点j去游玩(即r(i,j)=1),则不会再从景点j到景点i去游玩(即r(j,i)=0),也就是说除起点外每个景点只游玩一次;

@sum(jingdian(j):@sum(jingdian(i):r(i,j)*(tt(i,j)+0.5*(t(i)+t(j)))))<240;!约束条件:表示总的旅行时间(交通时间和景点逗留时间)

不超过给定时间10天;

@for(jingdian(i):@sum(jingdian(j):r(i,j))=@sum(jingdian(j):r(j,i)));

@for(jingdian(i)|i#eq#1:@sum(jingdian(j):r(i,j))=1);

@for(jingdian(i)|i#ne#1:@sum(jingdian(j):r(i,j))<1);!这三个约束条件:表示起点(成都)有且仅有一条路线出去和一条路线进来,其它景点要么有且仅有一条路线出去和一条路线进来,要么既没有路线出去也没有路线进来;

@for(links:@bin(r));!约束条件:表示0-1变量约束;

@sum(jingdian(j):@sum(jingdian(i):r(i,j)))=n;!约束条件:表示旅游景点的数目为n的约束;

@for(jingdian(i):@for(jingdian(j)|j#gt#1#and#j#ne#i:l(j)>=l(i)+r(i,j)-(n-2)*(1-r(i,j))+(n-3)*r(j,i)));

@for(jingdian(i)|i#gt#1:l(i)<n-1-(n-2)*r(1,i);l(i)>1+(n-2)*r(i,1));!这两个约束条件：为了控制不出现两个以上环形回路,保证有且仅有一条环形路线;