**2011年第八届苏北数学建模联赛**

****

题 目 B题 旅游线路的优化设计

**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、花最短的时间或者游览尽可能多的景点是我们追求的三大目标。基于对此的研究，建立数学模型，设计出最佳的旅游路线。

第一问没有时间约束，要求游客游遍所有的景点，该问题也就成了典型的货郎担（TSP）问题。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旅游总费用最小为目标。再引入两个0—1变量分别表示是否游览某个景点以及是否需要在该景点住宿，从而推出交通总费用、住宿费以及吃饭等其它费用的函数表达式，给出相应的约束条件，使用lingo编程对模型求解得到最佳旅游路线为：徐州→恐龙园→黄山→普陀山→庐山→黄鹤楼→龙门石窟→秦始皇兵马俑→乔家大院→长城→崂山→徐州，至少需要的旅游费用为2842元。

第二问没有费用约束，要求游客游遍所有的景点，花最少的时间。同样使用第一问的约束条件，将目标函数改成时间函数，使用lingo编程得到最佳旅游路线为：徐州→崂山→长城→乔家大院→秦始皇兵马俑→龙门石窟→黄鹤楼→庐山→黄山→普陀山→恐龙园→徐州，又由附件一的给火车时刻表及逗留时间等得出，至少需要的旅游时间为 8天。

第三问给定费用约束，我们建立了一个最优规划模型，以景点个数最大为目标。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。得出最多可旅游7个景点的结论，且旅游路线为：徐州→恐龙园→庐山→黄鹤楼→龙门石窟→秦始皇兵马俑→乔家大院→长城→崂山→徐州。

对于第四问，与第三问大同小异，只是将约束条件由费用改成了时间，目标函数还是以景点个数最大化。还是使用lingo编程对模型求解，得到最多可旅游8个景点的结论，且旅游路线为：徐州→恐龙园→庐山→黄鹤楼→秦始皇兵马俑→长城→乔家大院→龙门石窟→徐州。

第五问，其实就是第三问与第四问的综合，在目标函数不变的基础上，将两问的约束条件合并即可。最后得出最多可旅游的景点数为6个，其路线为：徐州→恐龙园→庐山→乔家大院→秦始皇兵马俑→龙门石窟→黄鹤楼→徐州。

本文思路清晰，模型恰当，结果合理.由于所需利用的数据较繁杂，给数据的整理带来了很多麻烦，故我们利用Excel排序，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：最佳路线 TCP问题 最短时间 景点个数 最小费用

**1 问题重述**

随着人们的生活不断提高，旅游已成为提高人们生活质量的重要活动。江苏徐州有一位旅游爱好者打算现在的今年的五月一日早上8点之后出发，到全国一些著名景点旅游，最后回到徐州。由于跟团旅游会受到若干限制，他(她)打算自己作为背包客出游。他预选了十个省市旅游景点，如表1所示。

表1. 预选的十个省市旅游景点

|  |  |  |
| --- | --- | --- |
| 省市 | 景点名称 | 在景点的最短停留时间 |
| 江苏 | 常州市恐龙园 | 4小时 |
| 山东 | 青岛市崂山 | 6小时 |
| 北京 | 八达岭长城 | 3小时 |
| 山西 | 祁县乔家大院 | 3小时 |
| 河南 | 洛阳市龙门石窟 | 3小时 |
| 安徽 | 黄山市黄山 | 7小时 |
| 湖北 | 武汉市黄鹤楼 | 2小时 |
| 陕西 | 西安市秦始皇兵马俑 | 2小时 |
| 江西 | 九江市庐山 | 7小时 |
| 浙江 | 舟山市普陀山 | 6小时 |

根据以上要求，针对如下的几种情况，为该旅游爱好者设计详细的行程表，该行程表应包括具体的交通信息(车次、航班号、起止时间、票价等)、宾馆地点和名称，门票费用，在景点的停留时间等信息。

(1) 如果时间不限，游客将十个景点全游览完，至少需要多少旅游费用？请建立相关数学模型并设计旅游行程表。

(2) 如果旅游费用不限，游客将十个景点全游览完，至少需要多少时间？请建立相关数学模型并设计旅游行程表。

(3) 如果这位游客准备2000元旅游费用，想尽可能多游览景点，请建立相关数学模型并设计旅游行程表。

(4) 如果这位游客只有5天的时间，想尽可能多游览景点，请建立相关数学模型并设计旅游行程表。

(5) 如果这位游客只有5天的时间和2000元的旅游费用，想尽可能多游览景点，请建立相关数学模型并设计旅游行程表。

**2 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，游览的总费用由4部分组成，分别为交通总费用、住宿费、景点门票以及吃饭等其它费用。而在确定了相应的目标函数后，所以我们的目标就是在满足所有约束条件的情况下，求出目标函数的最优值。

2.2问题一和问题二的分析：

问题一要求我们为游客设计合适的旅游路线，使游客在无时间限制的情况下游览完全部的景点。在这里我们的做法是在满足相应的约束条件下，计算出在这种情况下的最小花费。这样最终会得出一个唯一的旅游路线图。

问题二实质上是在问题一的基础上，将目标函数由旅游费用改为了旅游时间，即游客要游览所有的景点花最少的时间，我们完全可以使用与问题一同样的方法进行求解。

2.3问题三和问题四的分析：

问题三和问题四均要求我们以旅游景点的个数为目标函数，并在问题一与问题二的基础上多加一个约束条件。不同之处在于，问题四的约束条件为时间约束，问题三的为旅游费用约束，其解题方法与问题一、二大同小异。

2.4问题五的分析：

问题五的求解只需将问题三与问题四的约束条件结合即可参照问题三或四的方法进行求解。

**3 模型假设**

(1) 城际交通出行可以乘火车(含高铁)、长途汽车或飞机（不允许包车或包机），并且车票或机票可预订到。

(2) 市内交通出行可乘公交车(含专线大巴、小巴)、地铁或出租车。

(3) 旅游费用以网上公布为准，具体包括交通费、住宿费、景点门票(第一门票)。晚上20：00至次日早晨7：00之间，如果在某地停留超过6小时，必须住宿，住宿费用不超过200元/天。吃饭等其它费用60元/天。

(4) 假设景点的开放时间为8:00至18:00。

**4 符号说明**

,——第个或者第个景点， ，=1，2，……，11；

分别表示徐州、恐龙园、崂山、八达岭长城、乔家大院、龙门石窟、黄山、黄鹤楼、秦始皇兵马俑、庐山、普陀山；

**——旅游总花费，元；

**——交通总费用，元；

**——住宿总费用，元；

**——景点门票总费用，元；

**——吃饭等其它费用，元；

——游客在第个景点的停留时间，小时；

——游客在个景点的住宿费，元；

*pi*——第*i*个景点的第一门票价格，元

——从第个景点到第个景点路途中所需时间，小时；

——从第个景点到第个景点所需的交通费用，元；

 

 

**5 模型建立及求解**

5.1 问题一：

5.1.1 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，使这位游客将十个景点全游览完花最少的钱。显然，花费最少是该问题的目标。因此，我们的做法是在满足相应的约束条件下，计算出在这种情况下的最小花费。这样最终会得出一条最优旅游路线。

游览的总费用由4部分组成，分别为交通总费用、住宿费、景点门票以及吃饭等其它费用。由于十个景点全游览，故景点门票费用是确定的，

从而得到目标函数：Min *m*0＝**+**+*m*4

(其中，*m*0表示除去景点门票后的旅游花费)

（1）交通总花费

因为表示从第个景点到第个景点所需的交通费用，而是判断这位游客是否从第个景点直接到第个景点的0—1变量，因此我们可以很容易的得到交通总费用为：



（2）住宿总费用

因为表示在第*i*个景点的平均住宿费，而是判断这位游客是否在晚上20：00至次日早晨7：00之间，在某地停留超过6小时的0—1变量，则易知住宿总费用为：



（3）吃饭等其它费用

因为表示从第个景点到第个景点路途中所需时间，所以路途中所需总时间为；表示游客在第个景点的停留时间，故在旅游景点的总停留时间为。故旅游花费的总时间为：

+

因此，吃饭等其它费用为：

=60×{[+]÷24}

从而我们可以得到目标函数为：

Min *m*0＝**+**+*m*4

＝++60×{[+]÷24}

5.1.2 约束条件：

（1）旅游景点数约束

为游览的景点总数，由题目要求可知，因为时间不限，因此游客要游览完全部10个景点，徐州也假设为一个景点。因此该约束为：

 (，=1，2，……，11)

（2）0——1变量约束

根据假设，整个旅游路线是环形，即最终代表们要回到徐州，因此我们可以把整个路线看做一个Hamilton圈，这样该问题就归结为货郎担（TSP）问题，当然前提是我们已经知道了要旅游所有的景点。因此，对于Hamilton圈中的每个点来说，只允许有一条边进入，同样，也只允许有一条边出去。用公式表示即为：

  （，=1，2，……，11）

同样，当，时，根据题意不可能出现，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：

 （，=2，3，……，11）

5.1.3模型建立：

综上所述，我们可以得到总的模型为：

Min *m*0＝

++60×{[+]÷24}

约束条件：

 （，=1，2，……，11）

  （，=1，2，……，11）

 （，=2，3，……，11）

5.1.4 模型求解与结果分析

通过上网查询资料及原题所提供资料，我们可以得到旅客在第个景点的最佳逗留时间和他们在第个景点总消费：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 | t11 |
| 0 | 4 | 6 | 3 | 3 | 3 | 7 | 2 | 2 | 7 | 6 |

(单位：小时)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地点 | 到达景区时间/小时 | 工具 | 交通费用/元 | 景区门票/元 |
| 常州 | 2 | 公交旅游车 | 10\*2 | 160 |
| 青岛 | 1 | 公交 | 2\*2 | 90 |
| 北京 | 3 | 公交 | 5\*2 | 45 |
| 祁县 | 2 | 公交 | 5\*2 | 40 |
| 洛阳 | 1 | 公交 | 1.5\*2 | 120 |
| 黄山 | 2 | 专车 | 15\*2 | 230 |
| 武汉 | 2 | 专车 | 4\*2 | 80 |
| 西安 | 1 | 专车 | 7\*2 | 90 |
| 九江 | 0.5 | 专车 | 15\*2 | 180 |
| 宁波 | 3 | 汽车、轮渡 | 38\*2 | 200 |

此外，综合考虑只会在宁波住宿，住宿费为90元。

从而根据模型，使用Lingo编程，得出最省钱的旅游路线为：

徐州→恐龙园→黄山→普陀山→庐山→黄鹤楼→龙门石窟→秦始皇兵马俑→乔家大院→长城→崂山→徐州，至少需要的旅游费用为2842元。

5.2 问题二

5.2.1 目标函数的确立：

此问与第一问大同小异，不同的是要求游客游遍所有的景点，花最少的时间。同样使用第一问的约束条件，将目标函数改成时间函数。由第一问结论可知，旅游花费的总时间为：

+

因此，该问题的目标函数为：

Min T=+

(其中，T为旅游花费的总时间)

5.2.2 约束条件：

(1)旅游景点数约束

由题目要求可知，因为费用不限，因此游客打算游览完全部十个景点。因此该约束为：

 (，=1，2，……，11)

（2）0——1变量约束

根据假设，整个旅游路线是环形，即最终代表们要回到徐州，这样同问题一相同，用公式表示即为：

  （，=1，2，……，11）

同样，当，时，根据题意不可能出现，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：

 （，=2，3，……，11）

5.2.3模型建立：

综上所述，我们可以得到总的模型为：

Min T=+

约束条件：

 （，=1，2，……，11）

  （，=1，2，……，11）

 （，=2，3，……，11）

5.2.4 模型求解与结果分析：

根据模型，使用Lingo编程，得到最佳旅游路线为：

徐州→崂山→长城→乔家大院→秦始皇兵马俑→龙门石窟→黄鹤楼→庐山→黄山→普陀山→恐龙园→徐州

又由附件一的给火车时刻表及逗留时间等得出，至少需要的旅游时间为 8天。

5.3 问题三

5.3.1目标函数的确立

本文我们的做法同样是在满足相应的约束条件下，以景点个数最大为目标。根据假设，整个旅游路线是环形，即最终游客要回到徐州，因此即表示游客旅游的景点数。因此得目标函数为：

Max N =

(其中，N表示游客所游览的景点个数)

5.3.2 约束条件

（1）总费用约束

游览的总费用由4部分组成，分别为交通总费用、住宿费、景点门票以及吃饭等其它费用。又因为*pi*表示第*i*个景点的第一门票价格，也可以表示出代表们是否到达过第个和第个景点，而整个旅游路线又是一个环形，因此实际上将游客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：



则旅游总花费为:

*m*＝**+**+*m*3+ *m*4

=+++60×{[+]÷24}

因此，费用约束条件为：

*m*=+++60×{[+]÷24}≦2000

（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：

 （，=1，2，……，11）

当时，因为成都是出发点，所以；

当时，因为代表们最终要回到徐州，所以。

综合以上可知，

 （，=1，2，……，11）

 

同样，当，时，根据题意不可能出现，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：

 （，=2，3，……，11）

5.3.3模型建立：

综上所述，我们可以得到总的模型为：

Max N =

约束条件：

+++60×{[+]÷24}≦2000

 （，=1，2，……，11）

  （，=2，3，……，11）

 （，=2，3，……，11）

5.3.4 模型求解与结果分析：

使用lingo编程对模型求解，且求解的程序见附件二所示，得出最多可旅游7个景点的结论，且旅游路线为：徐州→恐龙园→庐山→黄鹤楼→龙门石窟→秦始皇兵马俑→乔家大院→长城→崂山→徐州。

5.4 问题四

5.4.1目标函数的确立

此问的目标函数与问题三一样，均为实现最大景点数。因此得目标函数为：

Max N =

(其中，N表示游客所游览的景点个数)

5.4.2 约束条件

（1）时间约束

因为表示从第个景点到第个景点路途中所需时间，所以路途中所需总时间为；表示游客在第个景点的停留时间，故在旅游景点的总停留时间为。故旅游花费的总时间为：

+

而该游客只有5天的时间，即120个小时。因此，时间约束为：

+≦120

（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：

 （，=1，2，……，11）

当时，因为成都是出发点，所以；

当时，因为代表们最终要回到徐州，所以。

综合以上可知，

 （，=1，2，……，11）

 

同样，当，时，根据题意不可能出现，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：

 （，=2，3，……，11）

5.4.3模型建立：

综上所述，我们可以得到总的模型为：

Max N =

约束条件：

+≦120

 （，=1，2，……，11）

  （，=2，3，……，11）

 （，=2，3，……，11）

5.4.4 模型求解与结果分析：

使用lingo编程对模型求解，且求解的程序见附件二所示，得出最多可旅游8个景点的结论，且旅游路线为：徐州→恐龙园→→庐山→黄鹤楼→龙门石窟→秦始皇兵马俑→乔家大院→长城→崂山→徐州。

5.5 问题五

5.5.1目标函数的确立

此问的目标函数与问题三和问题四一样，均为实现最大景点数。因此得目标函数为：

Max N =

(其中，N表示游客所游览的景点个数)

5.5.2 约束条件

（1）时间约束

因为表示从第个景点到第个景点路途中所需时间，所以路途中所需总时间为；表示游客在第个景点的停留时间，故在旅游景点的总停留时间为。故旅游花费的总时间为：

+

而该游客只有5天的时间，即120个小时。因此，时间约束为：

+≦120

（2）总费用约束

游览的总费用由4部分组成，分别为交通总费用、住宿费、景点门票以及吃饭等其它费用。又因为*pi*表示第*i*个景点的第一门票价格，也可以表示出代表们是否到达过第个和第个景点，而整个旅游路线又是一个环形，因此实际上将游客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：



则旅游总花费为:

*m*＝**+**+*m*3+ *m*4

=+++60×{[+]÷24}

因此，费用约束条件为：

*m*=+++60×{[+]÷24}≦2000

（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。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：

 （，=1，2，……，11）

当时，因为成都是出发点，所以；

当时，因为代表们最终要回到徐州，所以。

综合以上可知，

 （，=1，2，……，11）

 

同样，当，时，根据题意不可能出现，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：

 （，=2，3，……，11）

5.5.3模型建立：

综上所述，我们可以得到总的模型为：

Max N =

约束条件：

+++60×{[+]÷24}≦2000

+≦120

 （，=1，2，……，11）

  （，=2，3，……，11）

 （，=2，3，……，11）

5.5.4 模型求解与结果分析：

使用lingo编程对模型求解，且求解的程序见附件二所示，最后得出最多可旅游的景点数为6个，其路线为：徐州→恐龙园→庐山→乔家大院→秦始皇兵马俑→龙门石窟→黄鹤楼→徐州。

**6 模型的评价、改进及推广**

6.1．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.由于需要考虑的因素众多，在对相应资料整理时，工作量大且比较繁琐。在进行路线设计时，必须充分考虑好各个因素，包括城际之间乘那种交通工具、在城内使用哪种交通工具、以及是否需要住宿等诸多因素都需考虑周全。

5.本模型的最大不足之处在于只给出符合条件的最优路线，但却未给出详细的旅游方案行程表。

6.2．模型的改进与推广：

1.实际情况中，还会有天气状况、交通事故等各种不定因素的影响，若充分考虑这些因素，并将其并入到模型中，这样会使所得结果更准确、更符合实际情况。

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。

**7 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]未知，最佳旅游线路数学建模，<http://www.doc88.com/p-78640836252.html，2011>

年4月30日。

**8 附录**

附录清单：附录1为搜集的一些数据

附录2为相关程序及运行结果

附录1：

网上查询到的一些数据及相应的计算出的数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地点 | 到达景区时间 | 工具 | 钱 | 景区门票 |
| 常州 | 2 | 公交旅游车 | 10\*2 | 160 |
| 青岛 | 1 | 公交 | 2\*2 | 90 |
| 北京 | 3 | 公交 | 5\*2 | 45 |
| 祁县 | 2 | 公交 | 5\*2 | 40 |
| 洛阳 | 1 | 公交 | 1.5\*2 | 120 |
| 黄山 | 2 | 专车 | 15\*2 | 230 |
| 武汉 | 2 | 专车 | 4\*2 | 80 |
| 西安 | 1 | 专车 | 7\*2 | 90 |
| 九江 | 0.5 | 专车 | 15\*2 | 180 |
| 宁波 | 3 | 汽车、轮渡 | 38\*2 | 200 |

附录2：程序及运行结果：

模型一：

model:

sets:

jingdian/1..11/:l;

links(jingdian,jingdian):r,cc;

endsets

data:

cc=0,70,100,106,110,70,100,116,113,87,130

70,0,150,158,165,125,73,105,165,87,75

100,150,0,125,127,125,182,186,165,175,220

106,158,125,0,94,106,182,154,150,165,190

110,165,127,94,0,53,135,157,41,200,240

70,125,125,106,53,0,170,90,55,125,186

100,73,182,182,135,170,0,100,200,93,92

116,105,186,154,157,90,100,0,137,45,81

113,165,165,150,41,55,200,137,0,143,194

87,87,175,165,200,125,93,45,143,0,94

130,75,220,190,240,186,92,81,194,94,0;

enddata

min=@sum(links(i,j):r(i,j)\*cc(i,j));

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

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

@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);

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

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

@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));

end

结果：

Local optimal solution found.

Objective value: 879.0000

Objective bound: 879.0000

Infeasibilities: 0.4262146E-05

Extended solver steps: 119

Total solver iterations: 281810

Variable Value Reduced Cost

R( 1, 2) 1.000000 0.000000

R( 2, 7) 1.000000 0.000000

R( 3, 1) 1.000000 0.000000

R( 4, 3) 1.000000 0.000000

R( 5, 4) 1.000000 0.000000

R( 6, 9) 1.000000 0.000000

R( 7, 11) 1.000000 0.000000

R( 8, 6) 1.000000 30.00000

R( 9, 5) 1.000000 0.000000

R( 10, 8) 1.000000 0.000000

R( 11, 10) 1.000000 2.000000

路线一：1→2→7→11→10→8→6→9→5→4→3→1

模型二：

model:

sets:

jingdian/1..11/:l;

links(jingdian,jingdian):r,tt;

endsets

data:

tt=0,6,10,11,10,7,11,11,13,10,15

6,0,16,17,15,14,10,10,19,8,8

10,16,0,13,14,16,14,15,16,19,20

11,17,13,0,10,10,20,16,15,16,22

10,15,14,10,0,7,18,14,11,16,30

7,14,16,10,7,0,16,9,6,12,18

11,10,14,20,18,16,0,12,18,8.75,9

11,10,15,16,14,9,12,0,14,4.5,15

13,19,16,15,11,6,18,14,0,17,26

10,8,19,16,16,12,8.75,4.5,17,0,14

15,8,20,22,30,18,9,15,26,14,0;

enddata

min=@sum(links(i,j):r(i,j)\*tt(i,j));

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

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

@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);

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

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

@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));

end

结果：

Local optimal solution found.

Objective value: 95.25000

Objective bound: 95.25000

Infeasibilities: 0.5440239E-06

Extended solver steps: 0

Total solver iterations: 1835

Variable Value Reduced Cost

R( 1, 3) 1.000000 0.000000

R( 2, 1) 1.000000 0.000000

R( 3, 4) 1.000000 0.000000

R( 4, 5) 1.000000 0.000000

R( 5, 9) 1.000000 0.000000

R( 6, 8) 1.000000 0.000000

R( 7, 11) 1.000000 0.000000

R( 8, 10) 1.000000 0.000000

R( 9, 6) 1.000000 0.000000

R( 10, 7) 1.000000 0.000000

R( 11, 2) 1.000000 0.000000

可得路线为：1→3→4→5→9→6→8→10→7→11→2→1

模型3

model:

sets:

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

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

endsets

data:

c=0,180,100,55,50,124,260,88,104,210,272;

t=0,8,8,9,7,5,11,6,4,8,12;

cc=0,70,100,106,110,70,100,116,113,87,130

70,0,150,158,165,125,73,105,165,87,75

100,150,0,125,127,125,182,186,165,175,220

106,158,125,0,94,106,182,154,150,165,190

110,165,127,94,0,53,135,157,41,200,240

70,125,125,106,53,0,170,90,55,125,186

100,73,182,182,135,170,0,100,200,93,92

116,105,186,154,157,90,100,0,137,45,81

113,165,165,150,41,55,200,137,0,143,194

87,87,175,165,200,125,93,45,143,0,94

130,75,220,190,240,186,92,81,194,94,0;

tt=0,6,10,11,10,7,11,11,13,10,15

6,0,16,17,15,14,10,10,19,8,8

10,16,0,13,14,16,14,15,16,19,20

11,17,13,0,10,10,20,16,15,16,22

10,15,14,10,0,7,18,14,11,16,30

7,14,16,10,7,0,16,9,6,12,18

11,10,14,20,18,16,0,12,18,8.75,9

11,10,15,16,14,9,12,0,14,4.5,15

13,19,16,15,11,6,18,14,0,17,26

10,8,19,16,16,12,8.75,4.5,17,0,14

15,8,20,22,30,18,9,15,26,14,0;

enddata

max=@sum(links(i,j):r(i,j));

@sum(links(i,j):r(i,j)\*(cc(i,j)+0.5\*(c(i)+c(j))))+60\*@sum(links(i,j):r(i,j)\*(tt(i,j)+0.5\*(t(i)+t(j))))/24<=2000;

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

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

@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);

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

@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));

end

结果：Local optimal solution found.

Objective value: 9.000000

Objective bound: 9.000000

Infeasibilities: 0.9280937E-05

Extended solver steps: 0

Total solver iterations: 1744

Variable Value

R( 1, 2) 1.000000

R( 2, 10) 1.000000

R( 3, 1) 1.000000

R( 4, 3) 1.000000

R( 5, 4) 1.000000

R( 6, 9) 1.000000

R( 8, 6) 1.000000

R( 9, 5) 1.000000

R( 10, 8) 1.000000

可得路线为：1→2→10→8→6→9→5→4→3→1，共去8个景点

模型4：

model:

sets:

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

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

endsets

data:

k=0,4,6,3,3,3,7,2,2,7,6;

t=0,8,8,9,7,5,11,6,4,8,12;

tt=0,6,10,11,10,7,11,11,13,10,15

6,0,16,17,15,14,10,10,19,8,8

10,16,0,13,14,16,14,15,16,19,20

11,17,13,0,10,10,20,16,15,16,22

10,15,14,10,0,7,18,14,11,16,30

7,14,16,10,7,0,16,9,6,12,18

11,10,14,20,18,16,0,12,18,8.75,9

11,10,15,16,14,9,12,0,14,4.5,15

13,19,16,15,11,6,18,14,0,17,26

10,8,19,16,16,12,8.75,4.5,17,0,14

15,8,20,22,30,18,9,15,26,14,0;

enddata

max=@sum(links(i,j):r(i,j));

@sum(links(i,j):r(i,j)\*(tt(i,j)+0.5\*(t(i)+t(j))))<=120;

@sum(links(i,j):r(i,j)\*(0.5\*(k(i)+k(j))))<=50;

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

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

@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);

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

@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));

end

结果：Local optimal solution found.

Objective value: 8.000000

Objective bound: 8.000000

Infeasibilities: 0.000000

Extended solver steps: 66

Total solver iterations: 132610

Variable Value

R( 1, 2) 1.000000

R( 2, 10) 1.000000

R( 4, 5) 1.000000

R( 5, 6) 1.000000

R( 6, 1) 1.000000

R( 8, 9) 1.000000

R( 9, 4) 1.000000

可得行程：1→2→10→8→9→4→5→6→1 共7个景点

R( 10, 8) 1.000000

模型5：

model:

sets:

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

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

endsets

data:

c=0,180,100,55,50,124,260,88,104,210,272;

k=0,4,6,3,3,3,7,2,2,7,6;

t=0,8,8,9,7,5,11,6,4,8,12;

cc=0,70,100,106,110,70,100,116,113,87,130

70,0,150,158,165,125,73,105,165,87,75

100,150,0,125,127,125,182,186,165,175,220

106,158,125,0,94,106,182,154,150,165,190

110,165,127,94,0,53,135,157,41,200,240

70,125,125,106,53,0,170,90,55,125,186

100,73,182,182,135,170,0,100,200,93,92

116,105,186,154,157,90,100,0,137,45,81

113,165,165,150,41,55,200,137,0,143,194

87,87,175,165,200,125,93,45,143,0,94

130,75,220,190,240,186,92,81,194,94,0;

tt=0,6,10,11,10,7,11,11,13,10,15

6,0,16,17,15,14,10,10,19,8,8

10,16,0,13,14,16,14,15,16,19,20

11,17,13,0,10,10,20,16,15,16,22

10,15,14,10,0,7,18,14,11,16,30

7,14,16,10,7,0,16,9,6,12,18

11,10,14,20,18,16,0,12,18,8.75,9

11,10,15,16,14,9,12,0,14,4.5,15

13,19,16,15,11,6,18,14,0,17,26

10,8,19,16,16,12,8.75,4.5,17,0,14

15,8,20,22,30,18,9,15,26,14,0;

enddata

max=@sum(links(i,j):r(i,j));

@sum(links(i,j):r(i,j)\*(tt(i,j)+0.5\*(t(i)+t(j))))<=120;

@sum(links(i,j):r(i,j)\*(0.5\*(k(i)+k(j))))<=50;

@sum(links(i,j):r(i,j)\*(cc(i,j)+0.5\*(c(i)+c(j))))+60\*@sum(links(i,j):r(i,j)\*(tt(i,j)+0.5\*(t(i)+t(j))))/24<=2000;

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

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

@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);

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

@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));

end

结果：

Local optimal solution found.

Objective value: 7.000000

Objective bound: 7.000000

Infeasibilities: 0.1425375E-09

Extended solver steps: 41

Total solver iterations: 102023

Variable Value Reduced Cost

R( 1, 2) 1.000000 1.000000

R( 2, 10) 1.000000 1.000000

R( 5, 9) 1.000000 1.000000

R( 6, 8) 1.000000 1.000000

R( 8, 1) 1.000000 1.000000

R( 9, 6) 1.000000 1.000000

R( 10, 5) 1.000000 1.000000

可得线路为：1→2→10→5→9→6→8→1 共6个景点