

假期国内最佳旅游路线规划

目录

摘要	3
关键词	3
Abstract.....	3
Keywords	4
引言	4
一、问题描述	4
1.1 问题详情.....	4
1.2 景点安排与花费.....	5
1.3 景点间费用及耗时.....	9
1.4 满意度.....	10
二、问题分析	11
2.1 方法选择.....	11
2.2 满意度处理.....	12
2.3 问题分解.....	14
三、主要方法	16
3.1 遗传算法.....	16
3.2 图论中的 Dijkstra 法.....	17
3.2.1 图论.....	17
3.2.2 最小生成树问题.....	17
3.2.3 Dijkstra 算法.....	18
3.2.4 Dijkstra 算法程序.....	20
四、问题求解与分析.....	21
4.1 情况 1.....	21
4.1.1 问题分析.....	21
4.1.2 基本思路.....	22
4.1.3 求解结果以及分析.....	23
4.2 情况 2.....	24
4.2.1 情况要求.....	24
4.2.2 基本思路.....	25
4.2.3 求解计算.....	25
4.3 情况 3.....	27
4.3.1 情况要求.....	27
4.3.2 基本思路.....	28
4.3.3 求解计算.....	28
4.4 情况 4.....	34

4.4.1 情况要求.....	34
4.4.2 基本思路.....	35
4.4.3 求解计算.....	35
4.5 情况 5.....	36
4.5.1 情况要求.....	36
4.5.2 基本思路.....	36
4.5.3 求解计算.....	36
五、结论与心得	37
六、团队风采	37
参考资料与参考文献.....	37
参考资料.....	37
参考文献.....	38
附录	38
附录 1:《假期国内最佳旅游路线规划》课题研究满意度调查表	38

摘要

为了解决包含主观满意度信息、价格与时间约束条件以及特殊情况的假期国内最佳旅游路线规划问题，采用遗传算法、Dijkstra 算法以及枚举法计算最小生成树，并利用约束条件的独立性将原问题分解为多个子问题，分情况讨论。计算结果表明，基于图论的遗传算法、Dijkstra 算法的求解效果较为理想，最终得到了综合满意度最高且符合其他独立约束条件的最佳旅游路线。本问题的求解过程为实际生活中的复杂条件路线规划问题以及其他最小生成树问题提供了一种新的思考。

关键词

旅行路线问题；图论；遗传算法；运筹学；Dijkstra 算法；最小生成树

Abstract

In order to solve the best itinerary problem with subjective satisfaction information, price and time constraint conditions and special conditions, which aims at traveling at home, the genetic algorithm, the Dijkstra algorithm and the enumeration method were used to calculate the Minimum Spanning Tree. Moreover, by using the independence of the constraint conditions, the original problem was broken into multiple sub problems and solved respectively. The results showed that the three algorithm perfectly solved the best itinerary problem step by step and finally reached the best tour route with highest comprehensive satisfaction, which also matched other independent constraints. The solving process of this problem provides a new way for route planning with complex conditions

and other minimum spanning tree problems during our daily life.

Keywords

itinerary; graph theory; genetic algorithm; operational research;
Dijkstra algorithm; minimum spanning tree

引言

随着中国经济的不断发展，人们的收入也不断提高，在能够满足衣、食、住、行等日常必须要求的同时，也有更多的钱花到精神生活的享受上来，而旅游就是其中之一。

为此我们选择了黄山、泰山、嵩山、三清山、三峡以及扬州六个景点，来完成在不同限制条件下具体旅游景点的确定以及最佳旅游路径的选择。

一、问题描述

1.1 问题详情

假设共有 6 个人，从杭州出发并按照各省的实际地理分布沿顺时针或逆时针方向旅行，最终回到杭州。同时由于时间限制，仅有 7 天的旅游时间，因此只能从黄山、泰山、嵩山、三清山、三峡以及扬州这六个景点中选择三个景点游览。现要求能够找到一条最佳的旅游线路，使得参加旅游的 6 个人对本次行程的满意度最高。同时也要满足以下几点要求：

- 1、考虑到游客的体力以及整体行程的合理性，不能连续两个景点都是爬山项目；

2、根据天气预报知，行程的第六天在扬州会突发高温，易中暑，考虑到游客的身体状况，扬州不能作为第三个旅游景点；

3、根据天气预报可知，行程的第四天在嵩山风景区附近会下大雨，登山项目存在较大的安全隐患，考虑到旅游的安全，嵩山不能作为第二个旅游景点；

4、为了丰富旅游内容，安排去泰山顶上看日出，结合天气情况和门票，第三天凌晨一定能在泰山看到日出。

1.2 景点安排与花费

为了方便计算，我们对每个景点的旅游时间进行了统一安排——每个景点安排两天的参观时间，其中第一天全部用于参观，第二天用于适当的参观、前往下一个景点以及休息。在此作出以下几点假设和要求：

1、旅游过程中不会发生意外情况（包括成员受伤生病、路上堵车等等），也就是旅游能按照我们的预计情况顺利进行；

2、除约束条件给出的特殊天气情况外，其余天气均适合旅游，不会影响正常的旅游项目；

3、忽略旅游行程中的公交车费用，旅游景点购买纪念品、零食饮料等消耗品的额外消费不计算在行程花费中，由每位旅客自己把握；

4、旅游中的餐饮费用每天都相同（例如 100 元每人每天），且各地的酒店费用都相同：100 元/人，其中旅游景点内部的酒店费用，如位于景区内山区的酒店价格另算；

5、旅游过程中的所有门票价格都按照成人票计算，也不考虑景点在某些特定时间段的优惠情况。

在此基础上可以得到黄山、泰山、嵩山、三清山、三峡以及扬州六个旅游景点的具体安排情况：

1、黄山



图1 黄山风景区照片

第一天从黄山市乘车到黄山风景区，徒步上山并参观沿途各风景点，晚上住在黄山上；第二天乘坐索道下山并乘车到黄山北站。该景区具体消费、时间信息如下：

门票：230 元/人；

索道：80 元/人；

风景区到黄山北站交通：20 元/人，一小时左右；

黄山市到风景区交通：20 元/人，一小时左右；

酒店：100 元/人；

黄山景区内酒店：180 元/人。

2、泰山



图2 泰山风景区照片

第一天从泰安市乘车到达中天门，徒步爬山至南天门，并参观沿途景观。晚上统一安排山上酒店住宿。第二天乘坐索道下山，并乘车到泰安站。该景区具体消费、时间信息如下：

门票：127 元/人；

索道：100 元/人；

风景区到泰安市或泰安站：5 元/人，30 分钟左右；

酒店：100 元/人；

位于泰山山上的酒店：150 元/人。

3、嵩山



图3 嵩山风景区照片

第一天从郑州乘车到嵩山风景区，徒步爬山并参观沿途的风景点，最后徒步下山并在附近的酒店住宿。第二天乘车到郑州站。该景区具体消费、时间信息如下：

门票：80 元/人；

嵩山到郑州：37 元/人，1 小时 30 分左右；

酒店：100 元/人。

4、三清山



图 4 三清山风景区照片

第一天从上饶乘车到三清山景区，并乘坐索道上山，参观山上各景点，最后再乘坐索道下山，在山下附近酒店中投宿。第二天乘车到上饶站。该景区具体消费、时间信息如下：

门票：150 元/人；

上、下索道：125 元/人；

三清山到上饶：15 元/人，1 小时 30 分左右；

酒店：100 元/人。

5、三峡



图 5 三峡风景区照片

第一天从宜昌市区上车到港口乘船，参观沿岸各景区，最后在葛洲坝散团回

宜昌市区。第二天直接从宜昌市区前往宜昌火车站，因此忽略费用以及时间。该景区具体消费、时间信息如下：

- 门票：170 元/人；
- 葛洲坝到宜昌市区：5 元/人；
- 酒店：100 元/人。

6、扬州



图 6 扬州风景区照片

第一天依次参观个园、东关街、何园、古运河，晚上住在扬州市区。第二天参观瘦西湖后在前往扬州站，由于扬州各景点都位于扬州市区内，因此忽略到扬州站的费用和时间。该景区具体消费、时间信息如下：

- 门票：个园 30 元，何园 30 元，瘦西湖 60 元，共 120 元；
- 酒店：100 元/人。

1.3 景点间费用及耗时

由于景点与景点间的距离较远，因此城市之间采用乘坐动车的交通方式。同时假设乘坐的动车都能按时出发并按时到达，不存在延误的情况，并忽略在动车站内的等待时间。根据铁路信息网站，得到如表 2 与表 3 所示数据信息。

表 2 动车票价与时长信息 1

	黄山北站	泰安站	郑州站	上饶站	宜昌站	扬州站
--	------	-----	-----	-----	-----	-----

黄山北站		4h40min 370.5 元	5h38min 442 元	49min 74.5 元	6h38min 331.5 元	4h23min 228 元
泰安站	4h30min 370.5 元		3h20min 269.5 元	5h18min 437.5 元	7h45min 435 元	5h20min 291.5 元
郑州站	5h35min 442 元	3h10min 269.5 元		6h30min 433 元	4h 328.5 元	6h11min 355 元
上饶站	49min 74.5 元	5h32min 437.5 元	6h14min 433 元		5h50min 313.5 元	6h13min 307 元
宜昌站	6h25min 331.5 元	6h59min 581 元	4h57min 328.5 元	7h12min 313.5		7h3min 278 元
扬州站	4h23min 228 元	5h3min 291.5 元	5h54min 355 元	4h56min 302.5 元	6h25min 278 元	

表 3 动车票价与时长信息 2

	黄山北站	泰安站	郑州站	上饶站	宜昌站	扬州站
杭州东站	3h26min 231 元	4h 359.5 元	4h47min 431 元	1h46min 156.5 元	7h9min 330.5 元	4h20min 155 元

1.4 满意度

考虑到每一个人都有各自的喜好，他们对不同的景点都有不同的满意程度，对于路上耗时的长短也都有各自的容忍程度，因此我们加入了满意度这一设定，用来表征每一个旅客对各个景点的喜欢程度以及对路上耗时长短的容忍程度。而包含这两种满意度信息的综合满意度最高也是旅游路线规划的重要依据。

对于满意度信息的征集，我们随机对六名旅客（分别为 A、B、C、D、E、F）进行了问卷调查，并得到以下的满意度表格。

表 4 景点满意度（值域 0—5）

	黄山	泰山	嵩山	三清山	三峡	扬州
A	3	5	1	1	4.5	5

B	4.5	4	4	4.5	3.5	5
C	3	5	4.5	1	1.5	5
D	4	3	5	2.5	4	2
E	4	4.5	3	2	1.5	5
F	4	2.5	4	4.5	5	3.5

表 5 两景点（包括起点杭州站）之间旅途时间（范围均为左开右闭区间）的满意度（值域-2—2）

	0-2h	2-4h	4-6h	6-8h	>8h
A	0	2	1	-2	-2
B	2	1.5	-0.5	-1.5	-2
C	2	2	2	0	-2
D	2	2	1	0.5	0
E	2	1	0	-1	-2
F	2	1.5	-0.5	-1.5	-2

二、问题分析

2.1 方法选择

旅游路径的规划看似是一个很简单的运筹学问题，但本课题的独特之处就在于约束条件的主观性较强，且突发情况、特殊情况对于整体旅游路线的时间、花费影响较大。由于本课题所研究的问题中存在“满意度”这一结合多角度约束条件的信息，单一的目标函数计算方法无法适用。在尝试了基于有限因子的多目标规划方法求解后，发现用该方法求解本题，起点杭州站到第一个景点的选择就有 6 种，要让最后的数学模型囊括题设的所有约束条件非常困难。

在整理数学模型时，最初设计的是 0-1 规划与多目标规划相结合的方法。但经过整理后发现，决策向量设为每个景点去（ $x_i = 1$ ）与不去（ $x_i = 0$ ）后，也很难将各景点的旅游顺序体现出来，这样一来就无法反应价格信息。因为价格信

息与景点的选择顺序有关，每两个景点之间的动车价格是确定的，但选定了某一景点之后，是否还有下一个景点、下一个景点是什么都是不确定的，再结合旅途时间信息，就更使得该方法下的数学模型复杂化，很难列出对应的数学公式来表示本课题中涉及的目标函数和约束条件之间的数学关系。

舍弃了目标规划和 0-1 规划的方法之后，考虑到路线的选择与图论中的旅行商问题、最小生成树问题较为接近，因此朝着图论问题求解的大方向来尝试着解决本题。

由于本题中条件较多且相互独立，因此选择分情况步步分解的方式，将问题中涉及的约束条件逐个分解，从而剖析得到最基础的问题，即不带连续爬山、泰山日出、只去三个景点等约束条件的 6 个景点的旅游路线规划，只保留满意度信息，即旅途时间与旅游花费两个约束条件。在解决了这个问题之后，再将其他条件一个个补充回来，在原先的基础上一步步求解出更加复杂的问题。

2.2 满意度处理

要用到图论的求解方法，首先必须对最原始的满意度进行处理。满意度的原始数据包括了 6 个接受调查的对象对 6 个景点的满意度和对两个景点之间的旅途时间的满意度，满意度越大，则最终的方案越好。

考虑到不同满意度项目的重要性和数值大小，定义“综合满意度”的概念，该概念包括了游客对景点的满意度、景点之间的旅途时间、两个城市之间的动车票价、景点住宿及内部项目费用等信息。相关的量及权重定义如下：

1、旅途时间 = 上一个景点到动车站时间 + 坐动车到下一个城市的时间；

2、平均满意度 = $\frac{1}{6}\Sigma$ （目的地景点满意度+旅途时间对应的满意度）；

3、综合满意度 = 平均满意度 — 两站之间的动车票价/100—（目标景点两晚住宿费 + 目标景点内门票等消费+出发景点到当地动车站的路费）/200。

上述数值中，为方便计算，“两站之间的动车票价/100”的结果根据四舍五入法保留一位小数，综合满意度的计算结果根据四舍五入法保留三位小数，各信息的权重根据其代表的实际量的重要性决定。

在上述方法的信息处理过程中，有如下两个前提：

1、为方便计算，忽略最后的返程时间、车票和某景点所在城市的动车站到景点的时间和路费；

2、因为各景点餐费相差不大，可通过预算“卡边”来设置用餐消费限额，因此不计入总花费的计算中。

由该处理过程得到如表 6 所示综合满意度数值表。

表 6 综合满意度

	黄山	扬州	泰山	嵩山	三峡	三清山
杭州	0.167	1.550	-0.318	-1.617	-2.733	0.275
黄山	/	0.750	-1.685	-3.233	-2.833	1.325
扬州	-1.000	/	-0.785	-2.333	-2.233	-2.142
泰山	-2.425	0.225	/	1.125	-4.833	-3.467
嵩山	-4.702	-2.052	-0.770	/	-1.502	-4.943
三峡	-3.442	-1.092	-5.127	-0.642	/	-3.583
三清山	1.692	-1.442	-3.777	-3.108	-2.608	/

由于 Dijkstra 算法的适用条件要求赋权图中的权值必须为非负数，且要求的最佳路径的权值之和最小。而表 6 中整理得到的“综合满意度”一值包含了负数，且根据其代表的实际意义，综合满意度越大，路径越优，与最小生成树的问题思想恰好相反。因此，采用加基础偏置保证各综合满意度值为正数，再求倒数以契合最小生成树问题特征的方法，得到起点和各景点之间的权值矩阵（邻接矩阵），其各元素数值如表 7 所示。

表 7 基础权值矩阵元素数值

	杭州	黄山	扬州	泰山	嵩山	三峡	三清山
杭州	0	1.76	1.42	1.93	2.58	3.61	1.73
黄山	∞	0	1.60	2.62	4.41	3.75	1.47
扬州	∞	2.22	0	2.12	3.16	3.06	2.98
泰山	∞	3.25	1.75	0	1.51	14.99	4.92
嵩山	∞	12.53	2.90	2.11	0	2.50	17.95
三峡	∞	4.86	2.27	26.81	2.06	0	5.22

表 7 中的每一个元素代表两地之间的满意度权，其计算公式为

$$\text{满意度权} = 10 \div (\text{综合满意度} + 5.5)。$$

该公式的计算结果根据四舍五入法保留两位小数。在表 7 所表示的两地之间的满意度中，若两地实际为同一地点，则满意度权取 0，若两地不可达，则取 ∞ 。由于本课题所涉及的旅游路线具有方向性，因此赋权图为有向图，其权值矩阵（邻接矩阵）不对称。

至此，权值矩阵中的元素满足 $W_{ij} \geq 0$ 的要求，可以使用 Dijkstra 算法等运筹学方法，且包含了满意度中的所有信息和景点花费、旅途时间等信息，满足本题约束条件的要求。因此，数据处理工作至此结束。

2.3 问题分解

为了方便问题的求解，将问题按照条件逐级分解，得到共计 5 个子问题，对应 5 种约束条件组合的情况。其中，情况 5 便是本课题所要解决的最终旅游路线规划问题。

● 情况 1

情况要求：

从杭州出发，旅游路径包括了所有可选的 6 个景点，且要求最后的路径中综合满意度最高（满意度权最小）。

● 情况 2

情况要求：

1、从杭州出发，只考虑满意度权，要求最后的路径中综合满意度最高；

2、按省份的地理分布顺时针或逆时针方向旅行，为了均匀旅游时间，黄山、扬州两处需二选一，泰山、嵩山两处需二选一，三峡、三清山两处需二选一，共计旅游 3 个景点，历时 7 天。

● 情况 3

情况要求：

- 1、从杭州出发，要求最后的路径中综合满意度最高；
- 2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；
- 3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目。

● 情况 4

情况要求：

- 1、从杭州出发，要求最后的路径中综合满意度最高；
- 2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；
- 3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目；
- 4、为丰富旅游项目，安排去泰山看日出。

● 情况 5

情况要求：

- 1、从杭州出发，要求最后的路径中综合满意度最高；
- 2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；
- 3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目；
- 4、为丰富旅游项目，安排去泰山看日出；
- 5、由于第六天扬州高温，考虑到游客舒适度的问题，不安排当天在扬州旅游，所以第三个景点不可以选择扬州；
- 6、由于第四天嵩山风景区附近降雨，存在安全隐患，不适合爬山，因此嵩山不能是第二个景点。

三、主要方法

本课题由于对于原始问题进行了分解处理，整个求解过程中除分情况讨论之外用到的主要求解思想为遗传算法、图论中的 Dijkstra 法和枚举法求解最小生成树。其中，遗传算法用于求解“情况 1”所述的子问题，Dijkstra 法用于求解“情况 2”、“情况 3”所述的子问题，枚举法用于求解“情况 4”、“情况 5”所述的子问题。

3.1 遗传算法

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是计算数学中用于解决最优化的搜索算法，是进化算法的一种。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。

遗传算法通常实现方式为一种计算机模拟。对于一个最优化问题，一定数量的候选解（称为个体）的抽象表示（称为染色体）的种群向更好的解进化。传统上，解用二进制表示（即 0 和 1 的串），但也可以用其他表示方法。进化从完全随机个体的种群开始，之后一代一代发生。在每一代中，整个种群的适应度被评价，从当前种群中随机地选择多个个体（基于它们的适应度），通过自然选择和突变产生新的生命种群，该种群在算法的下一代迭代中成为当前种群。

遗传算法的基本运算过程如下：

- 1、初始化：设置进化代数计数器 $t=0$ ，设置最大进化代数 T ，随机生成 M 个个体作为初始群体 $P(0)$ ；
- 2、个体评价：计算群体 $P(t)$ 中各个个体的适应度；
- 3、选择运算：将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的；
- 4、交叉运算：将交叉算子作用于群体。所谓交叉是指把两个父代个体的部

分结构加以替换重组而生成新个体的操作。遗传算法中起核心作用的就是交叉算子；

5、变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ ；

6、终止条件判断：若 $t=T$ ，则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

3.2 图论中的 Dijkstra 法

3.2.1 图论

图论 (graph theory) 是运筹学的一个重要分支，它是以图为研究对象的。“图”的概念解释为由若干给定的点及连接两点的线所构成的图形，这种图形通常用来描述某些事物之间的某种特定关系，用点代表事物，用连接两点的线表示相应的两个事物之间具有的这种特定关系。

图论其广阔的应用领域涵盖了人类学、计算机科学、化学、环境保护、流体动力学、心理学、社会学、交通管理、电信网络等领域。特别是在 20 世纪 50 年代以后，随着科学技术的发展和计算机的出现与广泛的应用，促使了运筹学的发展，图论的理论也得到了进一步的发展。特别是庞大的复杂工程系统和管理问题都可以转化为图的问题，从而可以解决很多工程设计和决策中的最优化问题。诸如像完成工程任务的时间最少、距离最短、费用最少、收益最大、成本最低等实际问题。

因此，图论在数学、工程技术及经济等各个领域都受到了越来越广泛的重视。

3.2.2 最小生成树问题

图论中包含了许多具体问题，诸如最短路问题、最大流问题、指派问题、匹配问题、最小生成树问题、旅行商问题等。本课题涉及的图论问题为最小生成树

问题。

在现实生活中，经常遇到这样一类问题：在一些城市之间修建高速公路的问题，在保证各城市连通的前提下，希望修建的高速公路总长度最短，这样既能节约费用，又能缩短工期。类似的问题还有：在多个村庄之间修建电网的问题，总是希望供电线路的长度最短。要解决这类问题，事实上就是寻求图的小生成树的问题。

对于不包含圈的图称为无圈图；连通的无圈图称为树，记为 T ；其度为 1 的顶点称为叶。显然有边的树至少有两个叶。

设 G' 是 G 的子图，如果子图 G' 还包含 G 的所有顶点，则子图 G' 称为 G 的生成子图。

若 T 是 G 的生成子图，且 T 是树，则称 T 是 G 的生成树。若图 $G(V, E)$ 是一个连通赋权图， T 是 G 的一棵生成树， T 的每条边所赋权数之和称为树 T 的权，记为 $W(T)$ 。图 G 中具有最小权的生成树称为 G 的最小生成树。

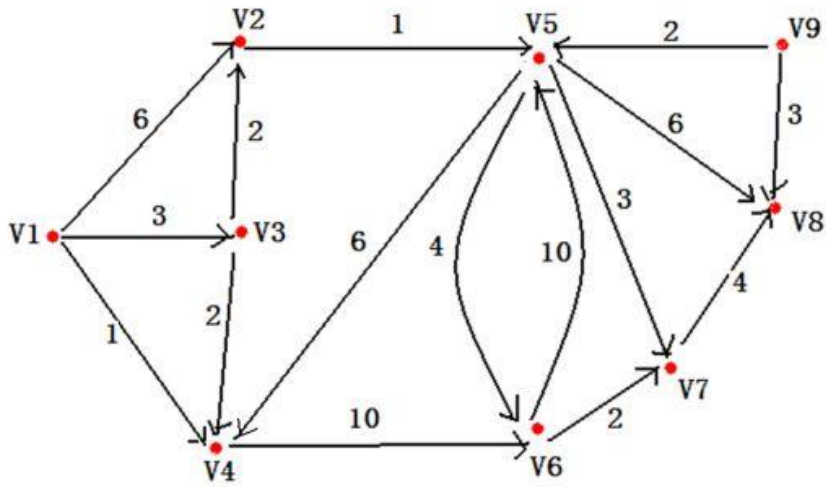


图 7 有向赋权图举例

3.2.3 Dijkstra 算法

计算赋权图的最小生成树有多种算法，例如克鲁斯卡尔（Kruskal）算法、普里姆（Prim）算法、warshall-floyd 算法等。本课题采用的最小生成树求解方法为 Dijkstra 算法。

Dijkstra 算法又叫标号算法，是由荷兰计算机科学家狄克斯特拉 1959 年提出的，因此命名为 Dijkstra 算法。该算法是从一个顶点到其余各顶点的最短路径算法，解决的是有向图中最短路径问题。其主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。

Dijkstra 算法的使用前提和结果适用性如下：

- 1、只适用于权 $W_{ij} \geq 0$ 的情况；
- 2、其结果不仅求出了从初始点到终点的最短路，实际上求出了从初始点到任何一点的最短路。

该算法的主要思想为通过在图上各点进行标号来寻求最短路。每一个点的标号有两种——临时表号 T 与永久标号 P。

T 标号表示从始点到该点的最短路的某一上界，根据到该点路线的不同，它可能有变化。

P 标号表示从始点到该点的最短路权，它的值不再改变。

标号过程共分为两步：

- 第一步：修改 T 标号

给定 V_i 是新产生的 P 标号点，观察以 V_i 为始点的所有弧段 $V_i \rightarrow V_j$ ，如果 V_j 是 P 标号点，则对 V_j 不再进行标号；如果 V_j 是点 T 标号点，则进行如下修改：

$$T(V_j) = \min[T(V_j), P(V_i) + W_{ij}]$$

其中，方括号内的 $T(V_j)$ 代表 V_j 点旧的 T 标号值， W_{ij} 为 $V_i \rightarrow V_j$ 弧段的权值。

- 第二步：产生新的 P 标号

原则如下：在所有现有的 T 标号点中将值最小者改为 P 标号，重复以上步骤，直到终点的 T 标号点变为 P 标号点为止。

Dijkstra 算法的流程图如图 8 所示。

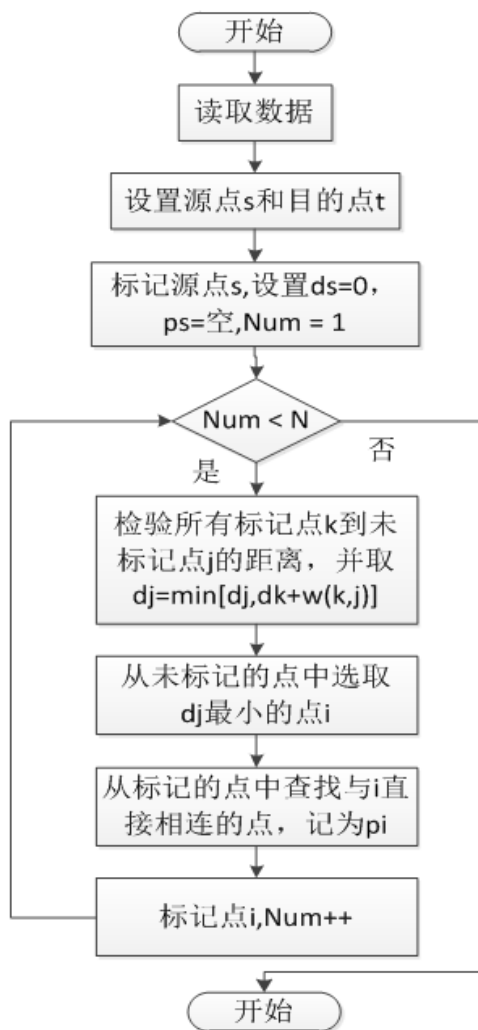


图8 Dijkstra 算法流程图

3.2.4 Dijkstra 算法程序

本课题中所用的 Dijkstra 算法由小组成员根据教学课件上的 Dijkstra 算法原理编写，MATLAB 程序如下：
略。

四、问题求解与分析

4.1 情况 1

4.1.1 问题分析

TSP 问题指的是从一个节点开始遍历其他所有节点并回到初始节点，构成一个哈密顿回路，节点与节点之间距离不同，目标是找到一条回路使得总路程最短，也即就是走最短的路遍历所有节点回到起点。

遗传算法模仿达尔文进化论中优胜劣汰的思想，从随机初始总群开始，不断进化最终选出接近最优解的一代，从而求解出近似最优解。

表 7 中的数据代表满意度权，数字越小满意度越高，故该问题可转化为最小路径问题。因为本课题不考虑回程问题，所以为了适应遗传算法，此情况将六个地点回杭州的满意度权值都设为 0，如表 8 所示。

表 8 情况 1 遗传算法权值矩阵

	杭州	黄山	扬州	泰山	嵩山	三峡	三清山
杭州	0	1.76	1.42	1.93	2.58	3.61	1.73
黄山	0	0	1.60	2.62	4.41	3.75	1.47
扬州	0	2.22	0	2.12	3.16	3.06	2.98
泰山	0	3.25	1.75	0	1.51	14.99	4.92
嵩山	0	12.53	2.90	2.11	0	2.50	17.95
三峡	0	4.86	2.27	26.81	2.06	0	5.22
三清山	0	1.39	2.46	5.80	4.18	3.46	0

上表可以看作不同城市之间的距离，城市到自身的距离为 0，现要求从杭州出发，找一条最短的旅游顺序。

4.1.2 基本思路

主要是问题的编码阶段，对于 TSP 问题在遗传算法中编码使用整数编码，使用整数来代表每一个城市，比如这里可以依次使用 1, 2, 3, ..., 7 表示这 7 个城市，1 代表杭州，2 为黄山，3 是扬州，4 是泰山，5 是嵩山，6 是三峡，7 是三清山。

```
Hangzhou    = [0    1.76    1.42    1.93    2.58    3.61    1.73]; %1
Huangshan   = [0  0    1.60    2.62    4.41    3.75    1.47]; %2
Yangzhou    = [0 2.22  0    2.12    3.16    3.06    2.98]; %3
Taishan     = [0  3.25 1.75    0    1.51    14.99    4.92]; %4
Songshan    = [0 12.53  2.90    2.11    0    2.50    17.95]; %5
Sanxia      = [0  4.86  2.27    26.81    2.06    0    5.22]; %6
Sanqingshan = [0  1.39    2.46    5.80    4.18    3.46    0]; %7

costM = [Hangzhou;Huangshan;Yangzhou;Taishan;Songshan;Sanxia;Sanqingshan];
```

图 9 权值矩阵输入

这 7 个数字的一个排列即是一种路径方案，但注意这条路径是一个环，收尾相接，因此起点是哪个城市是无所谓的，只要数字的相对位置确定，那 7 种(谁是起点)归并为同一种方案。因此所有可能的方案数为：7! / 7 (全排列除以 7)。

示例染色体：[1 2 3 4 5 6 7], 同[2 3 4 5 6 7 1]等属于同一种方案。

这种方案的路程代价为：

$$F = 1.76 + 1.60 + 2.12 + 1.51 + 2.50 + 5.22 = 14.71$$

每种方案的路程代价即个体的适应度，路程越短代价越小适应度越高，对应的实际意义就是我们题中所涉及的满意度最高。

4.1.3 求解结果以及分析

```
the lowest cost of every generation:
ans =
    12.9200
    12.9200
    12.9200
    12.0200
    12.0200
    12.0200
    11.8200
    11.8200
    11.8200
    11.8200
    11.8200
    11.8200
    11.8200
    11.8200
    10.8500
    10.8500
    10.8500
```

图 10 遗传算法运行结果 1

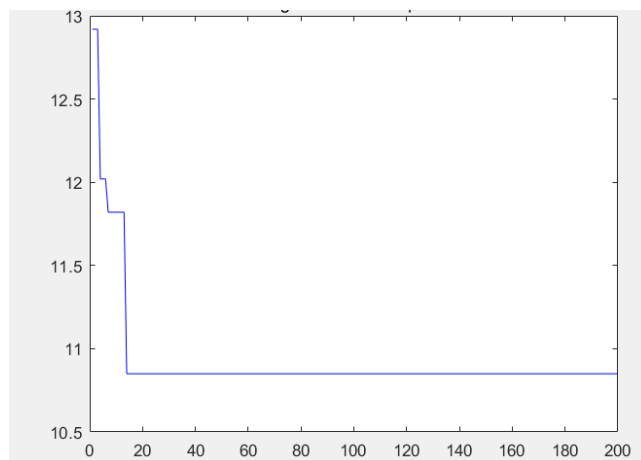


图 11 遗传算法运行结果 2

如图 10 和 11 所示，进行了 200 代左右进化，在 14 代以后收敛，得到近似最优方案的最小路程代价为 10.85。

最优染色体为：

the best parentpop:

parentpop =

3	4	5	6	1	7	2
3	4	5	6	1	7	2
3	4	5	6	1	7	2
3	4	5	6	1	7	2
3	4	5	6	1	7	2
3	4	5	6	1	7	2

图 12 最优染色体

[3 4 5 6 1 7 2]

等效于以杭州为起点的:

[1 7 2 3 4 5 6]

解码得到最佳方案为:

杭州→三清山→黄山→扬州→泰山→嵩山→三峡

即按照路线: 杭州→三清山→黄山→扬州→泰山→嵩山→三峡的顺序游玩游客的满意度最高, 该路线是最佳路线。

4.2 情况 2

4.2.1 情况要求

本情况的基本要求为:

- 1、从杭州出发, 只考虑满意度权, 要求最后的路径中综合满意度最高;
- 2、按省份的地理分布顺时针或逆时针方向旅行, 为了均匀旅游时间, 黄山、扬州两处需二选一, 泰山、嵩山两处需二选一, 三峡、三清山两处需二选一, 共计旅游 3 个景点, 历时 7 天。

4.2.2 基本思路

该情况下的子问题中，以综合满意度为权值信息，能够构成有向赋权图，但需要将该子问题进一步分解成顺时针旅行方向和逆时针方向进行分类讨论，分别产生各自符合该情况下子问题条件约束的最优路线。最后，再比较两者的综合满意度，即可得到整个子问题中的最优路线。

4.2.3 求解计算

● 顺时针

若遵循各景点地理分布上的顺时针旅游顺序，设杭州为点 v_1 ，黄山为点 v_2 ，扬州为点 v_3 ，泰山为点 v_4 ，嵩山为点 v_5 ，三峡为点 v_6 ，三清山为点 v_7 。其形成的图如图 12 所示。

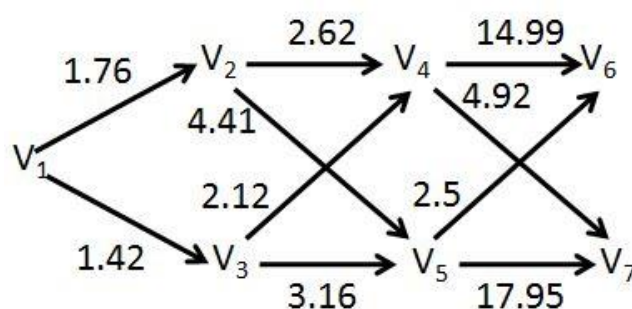


图 12 情况 2 顺时针

由图 12 得到的权值矩阵 W 为

$$W = \begin{bmatrix} 0 & 1.76 & 1.42 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & 2.62 & 4.41 & \infty & \infty \\ \infty & \infty & 0 & 2.12 & 3.16 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & 14.99 & 4.92 \\ \infty & \infty & \infty & \infty & 0 & 2.5 & 17.95 \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}。$$

该权值矩阵 W 中元素“0”表示某景点到本身的权值，元素“ ∞ ”表示该两点在情况 2 的前提条件下不可达。选用 Dijkstra 算法求解该情况下的最小生成树，使用 3.2.4 节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果：

```
>> Dijkstra
```

输入 $n \times n$ 权值矩阵 W :

```
[0 1.76 1.42 inf inf inf inf;inf 0 inf 2.62 4.41 inf inf;inf inf 0 2.12
3.16 inf inf;inf inf inf 0 inf 14.99 4.92;inf inf inf inf 0 2.5 17.95;inf
inf inf inf inf 0 inf;inf inf inf inf inf inf 0]
```

$P =$

```
0    1.7600    1.4200    3.5400    4.5800    7.0800
```

8.4600

$\lambda =$

```
0    1    1    3    3    5    4
```

因此,若终点为三清山,则综合满意度最高的路线为 杭州→扬州→泰山→三清山,其满意度权为 7.08;若终点为三峡,则综合满意度最高的路线为 杭州→扬州→嵩山→三峡,其满意度权为 8.46。相比之下,以三清山为终点的路线的满意度权最小,即综合满意度最高。

● 逆时针

若遵循各景点地理分布上的逆时针旅游顺序,设杭州为点 v_1 ,三清山为点 v_2 ,三峡为点 v_3 ,嵩山为点 v_4 ,泰山为点 v_5 ,扬州为点 v_6 ,黄山为点 v_7 。其形成的图如图 13 所示。

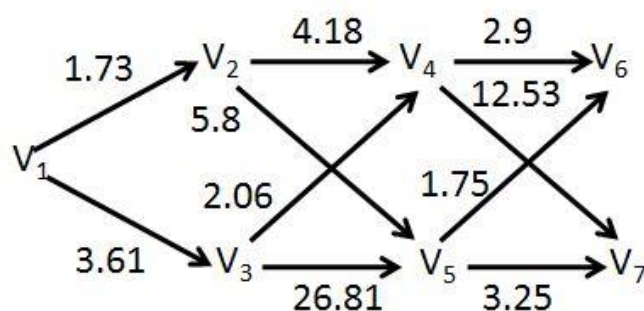


图 13 情况 2 逆时针

由图 13 得到的权值矩阵 W 为

$$W = \begin{bmatrix} 0 & 1.73 & 3.61 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & 4.18 & 5.8 & \infty & \infty \\ \infty & \infty & 0 & 2.06 & 26.81 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & 2.9 & 12.53 \\ \infty & \infty & \infty & \infty & 0 & 1.75 & 3.25 \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}。$$

同样选用 Dijkstra 算法求解该情况下的最小生成树，使用 3.2.4 节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果：

```
>> Dijkstra
```

输入 $n \times n$ 权值矩阵 W:

```
[0 1.73 3.61 inf inf inf inf;inf 0 inf 4.18 5.8 inf inf;inf inf 0 2.06
26.81 inf inf;inf inf inf 0 inf 2.9 12.53;inf inf inf inf 0 1.75 3.25;inf
inf inf inf inf 0 inf;inf inf inf inf inf inf 0]
```

```
P =
```

```
0    1.7300    3.6100    5.6700    7.5300    8.5700
10.7800
```

```
lambda =
```

```
0    1    1    3    2    4    5
```

因此，若终点为扬州，则综合满意度最高的路线为 杭州→三峡→嵩山→扬州，其满意度权为 8.57；若终点为黄山，则综合满意度最高的路线为 杭州→三清山→泰山→黄山，其满意度权为 10.78。相比之下，以扬州为终点的路线的满意度权最小，即综合满意度最高。

综合上述顺时针与逆时针两种路线，综合满意度最高的路线为 杭州→扬州→泰山→三清山，其满意度权为 7.08。

4.3 情况 3

4.3.1 情况要求

- 1、从杭州出发，要求最后的路径中综合满意度最高；

2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；

3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目。

4.3.2 基本思路

情况 3 由于约束条件的增加，为方便作图与计算，可再具体细分为第二个景点为爬山项目和第二个景点为非爬山项目两种情况进行讨论。如此分解的好处是减少了图的不确定性。

4.3.3 求解计算

- 第二个景点为爬山项目

由于不能连续两个景点都是爬山，所以第一个、第三个景点均不可为爬山项目，而供选择的景点中只有扬州、三峡两个景点不包含爬山项目，因此，这两个景点为必选项。

由此可简化问题，按省份顺时针方向旅行的路线选择只有如下两种，其方案用图表示则如图 14 所示，其中， v_1 表示杭州， v_2 为扬州， v_3 、 v_4 分别为泰山、嵩山， v_5 为三峡。

方案 1：杭州→扬州→泰山→三峡；

方案 2：杭州→扬州→嵩山→三峡。

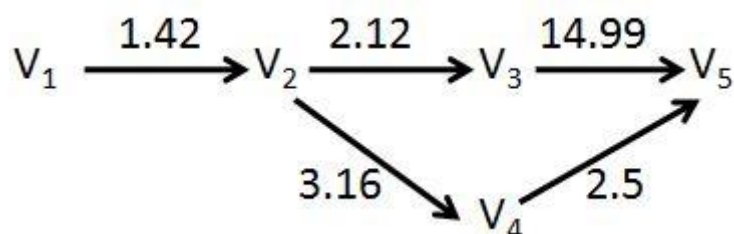


图 14 情况 3 景点二为爬山项目顺时针路线

其中存在区别的路线为方案 1 由扬州出发，经泰山到三峡，而方案 2 由扬州

出发后经嵩山到达三峡。因此为比较这两种方案的优劣，只需计算这部分权值的大小即可。但为了方便与逆时针方向旅行路线比较，仍将整条路线的满意度权计算出来。

方案 1： $1.42 + 2.12 + 14.99 = 18.53$ ；

方案 2： $1.42 + 3.16 + 2.5 = 7.08$ 。

由此可知，若必须选择顺时针方向旅行路线，则最佳路线为 杭州→扬州→嵩山→三峡。

按省份逆时针方向旅行的路线选择同样有如下两种，其方案用图表示则如图 15 所示，其中， v_1 表示杭州， v_2 为三峡， v_3 、 v_4 分别为泰山、嵩山， v_5 为扬州。

方案 3：杭州→三峡→嵩山→扬州；

方案 4：杭州→三峡→泰山→扬州。

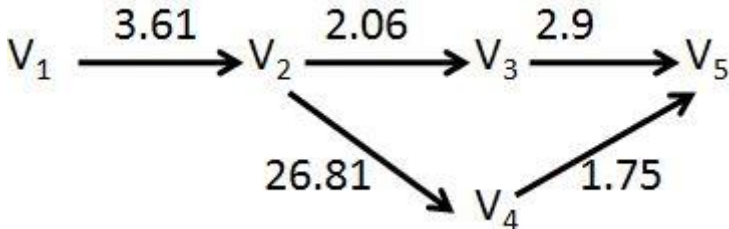


图 15 情况 3 景点二为爬山项目逆时针路线

其满意度权计算结果为

方案 3： $3.61 + 2.06 + 2.9 = 8.57$

方案 4： $3.61 + 26.81 + 1.75 = 32.17$

综上所述，当第二个景点为爬山项目时，最佳方案为 杭州→扬州→嵩山→三峡，该方案能在不存在连续两个景点都是爬山项目的前提下保证综合满意度最高。

● 第二个景点不包括爬山项目

由于可选景点中只有两个景点不包含爬山项目，第二个景点不是爬山项目的情况，又可以细分为两个类型，即第二个景点为扬州和第二个景点为三峡，分别进行讨论。

为了方便与“第二个景点为爬山”的情况进行比较，也需要计算整条路线的

满意度权。因此，选用 Dijkstra 方法求不同情况下对应图的最小生成树。由题意分析可知，此时可将情况细分为四种类型，分别为第二个景点为三峡的顺时针旅游路线、第二个景点为三峡的逆时针旅游路线、第二个景点为扬州的顺时针旅游路线、第二个景点为扬州的逆时针旅游路线。用图表示这四个类型，如图 16~图 19 所示。

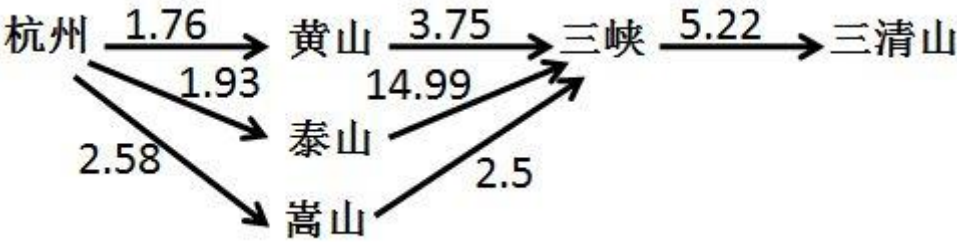


图 16 情况 3 景点二为三峡的顺时针路线

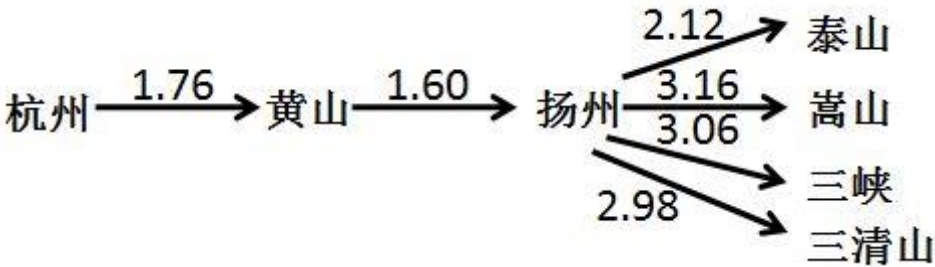


图 17 情况 3 景点二为扬州的顺时针路线

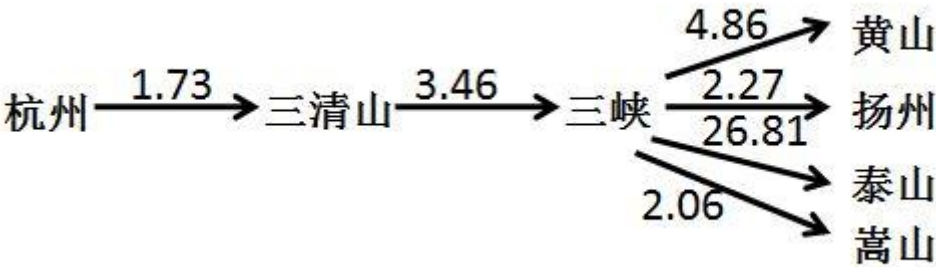


图 18 情况 3 景点二为三峡的逆时针路线

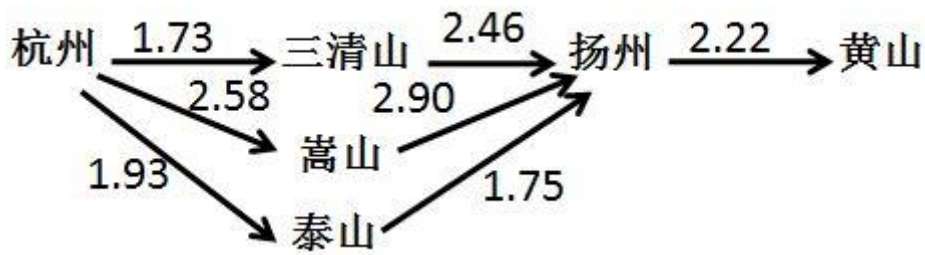


图 19 情况 3 景点二为扬州的逆时针路线

1、第二个景点为三峡、顺时针路线

此时点与弧构成的图如图 16 所示，权值矩阵整理如下，元素“0”表示某景点到本身的权值，元素“ ∞ ”表示该两点在情况 2 的前提条件下不可达。

$$W = \begin{bmatrix} 0 & 1.76 & 1.93 & 2.58 & \infty & \infty \\ \infty & 0 & \infty & \infty & 3.75 & \infty \\ \infty & \infty & 0 & \infty & 14.99 & \infty \\ \infty & \infty & \infty & 0 & 2.5 & \infty \\ \infty & \infty & \infty & \infty & 0 & 5.22 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

采用 Dijkstra 算法计算该权值矩阵代表的有向图的最小生成树，使用 3.2.4 节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果：

```
>> Dijkstra
```

输入 $n \times n$ 权值矩阵 W:

```
[0 1.76 1.93 2.58 inf inf;inf 0 inf inf 3.75 inf;inf inf 0 inf 14.99 inf;inf
inf inf 0 2.5 inf;inf inf inf inf 0 5.22;inf inf inf inf inf 0]
```

P =

```
0    1.7600    1.9300    2.5800    5.0800    10.3000
```

lambda =

```
0    1    1    1    4    5
```

根据运行结果，该情况下最优路线为 杭州→嵩山→三峡→三清山，该路线的满意度权为 10.3。

2、第二个景点为扬州、顺时针路线

此时点与弧构成的图如图 17 所示，权值矩阵整理如下，元素“0”表示某景点到本身的权值，元素“ ∞ ”表示该两点在情况 2 的前提条件下不可达。

$$W = \begin{bmatrix} 0 & 1.76 & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 1.6 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 2.12 & 3.16 & 3.06 & 2.98 \\ \infty & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

采用 Dijkstra 算法计算该权值矩阵代表的有向图的最小生成树，使用 3.2.4 节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果：

```
>> Dijkstra
```

```
输入 n×n 权值矩阵 W:
```

```
[0 1.76 inf inf inf inf inf;inf 0 1.6 inf inf inf inf;inf inf 0 2.12 3.16
3.06 2.98;inf inf inf 0 inf inf inf;inf inf inf inf 0 inf inf;inf inf inf
inf inf 0 inf;inf inf inf inf inf inf 0]
```

```
P =
```

```
0    1.7600    3.3600    5.4800    6.5200    6.4200
```

```
6.3400
```

```
lambda =
```

```
0    1    2    3    3    3    3
```

根据运行结果，该情况下最优路线为 杭州→黄山→扬州→泰山，该路线的满意度权为 5.48。

3、第二个景点为三峡、逆时针路线

此时点与弧构成的图如图 18 所示，权值矩阵整理如下，元素“0”表示某景点到本身的权值，元素“ ∞ ”表示该两点在情况 2 的前提条件下不可达。

$$W = \begin{bmatrix} 0 & \infty & \infty & \infty & \infty & \infty & 1.73 \\ \infty & 0 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & 4.86 & 2.27 & 26.81 & 2.06 & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3.46 & 0 \end{bmatrix}$$

采用 Dijkstra 算法计算该权值矩阵代表的有向图的最小生成树,使用 3.2.4 节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果:

```
>> Dijkstra
```

输入 $n \times n$ 权值矩阵 W:

```
[0 inf inf inf inf inf 1.73;inf 0 inf inf inf inf inf;inf inf 0 inf inf
inf inf;inf inf inf 0 inf inf inf;inf inf inf inf 0 inf inf;inf 4.86 2.27
26.81 2.06 0 inf;inf inf inf inf inf 3.46 0]
```

P =

```
0    10.0500    7.4600    32.0000    7.2500    5.1900
```

1.7300

lambda =

```
0    6    6    6    6    7    1
```

根据运行结果,该情况下最优路线为 杭州→三清山→三峡→嵩山,该路线的满意度权为 1.73。

4、第二景点为扬州、逆时针路线

此时点与弧构成的图如图 19 所示,权值矩阵整理如下,元素“0”表示某景点到本身的权值,元素“ ∞ ”表示该两点在情况 2 的前提条件下不可达。

$$W = \begin{bmatrix} 0 & \infty & \infty & 1.93 & 2.58 & 1.73 \\ \infty & 0 & \infty & \infty & \infty & \infty \\ \infty & 2.22 & 0 & \infty & \infty & \infty \\ \infty & \infty & 1.75 & 0 & \infty & \infty \\ \infty & \infty & 2.9 & \infty & 0 & \infty \\ \infty & \infty & 2.46 & \infty & \infty & 0 \end{bmatrix}$$

采用 Dijkstra 算法计算该权值矩阵代表的有向图的最小生成树,使用 3.2.4

节中的 Dijkstra 算法程序进行求解。程序运行后得到以下结果：

```
>> Dijkstra
```

输入 $n \times n$ 权值矩阵 W:

```
[0 inf inf 1.93 2.58 1.73;inf 0 inf inf inf inf;inf 2.22 0 inf inf inf;inf
inf 1.75 0 inf inf;inf inf 2.9 inf 0 inf;inf inf 2.46 inf inf 0]
```

P =

```
0    5.9000    3.6800    1.9300    2.5800    1.7300
```

lambda =

```
0    3    4    1    1    1
```

根据运行结果，该情况下最优路线为 杭州→泰山→扬州→黄山，该路线的满意度权为 1.93。

综合上述所有情况，综合满意度最高的路线为 杭州→三清山→三峡→嵩山，该路线的满意度权为 1.73，该方案能在不存在连续两个景点都是爬山项目的前提下保证综合满意度最高。

4.4 情况 4

4.4.1 情况要求

- 1、从杭州出发，要求最后的路径中综合满意度最高；
- 2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；
- 3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目；
- 4、为丰富旅游项目，安排去泰山看日出。

4.4.2 基本思路

该情况可以看做在情况 3 的基础上增加了去泰山看日出的前提条件。由题设信息可知，确保能在泰山看到日出的只有第三天凌晨，所以必须将泰山设置为旅游线路的第一个景点。

本情况的求解思路为在情况 3 的基础上对问题进行简化。

4.4.3 求解计算

由于情况 4 的所有约束条件中除看日出的要求之外与情况 3 完全一致，套用情况 3 的分析内容可知，只存在如下两种情况符合情况 4 的新增约束条件：

方案 A：杭州→泰山→三峡→三清山；

方案 B：杭州→泰山→扬州→黄山。

用图表示这两种方案，如图 20 所示。

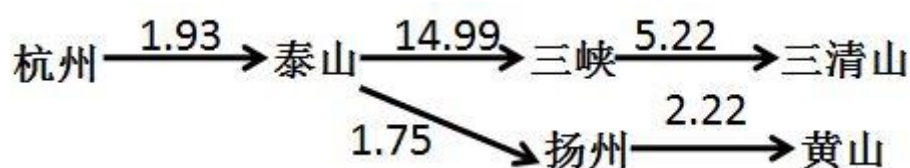


图 20 情况 4

因为可选择的方案较少，所以直接计算两条路线的满意度权：

方案 A： $1.93 + 14.99 + 5.22 = 22.14$

方案 B： $1.93 + 1.75 + 2.22 = 5.9$

显然，在该情况下，最佳路径为 杭州→泰山→扬州→黄山，该方案既保证了不出现连续的两个景点都是爬山项目，又保证了能够去泰山看日出。

4.5 情况 5

4.5.1 情况要求

- 1、从杭州出发，要求最后的路径中综合满意度最高；
- 2、按省份的地理分布顺时针或逆时针方向旅行，共计旅游 3 个景点，历时 7 天；
- 3、考虑到游客体力和兴趣的状态，旅游路线中，不存在连续两个景点都是爬山项目；
- 4、为丰富旅游项目，安排去泰山看日出；
- 5、由于第六天扬州高温，考虑到游客舒适度的问题，不安排当天在扬州旅游，所以第三个景点不可以选择扬州；
- 6、由于第四天嵩山风景区附近降雨，存在安全隐患，不适合爬山，因此嵩山不能是第二个景点。

4.5.2 基本思路

该情况可以看做在情况 4 的基础上增加了最后的路径满足扬州不是第三个景点、嵩山不是第二个景点这两个前提条件。因此，可以按照情况 4 的结果对该情况的子问题进行简化。

4.5.3 求解计算

情况 4 中选出的方案 B 满足本情况下除新增条件外的所有条件，且为最优情况。因此，直接判断方案 B 是否满足本情况的约束条件是最快的方法。

经验证，情况 4 的最优路线 杭州→泰山→扬州→黄山满足第二个景点不是嵩山且第三个景点不是扬州的约束条件，因此该方案也是本情况下的最优方案。

五、结论与心得

本课题通过多种运筹学求解问题的方法，对一个具体的旅游路线规划问题进行了求解。在求解过程中，由于问题中的约束条件结合了主观满意度、突发情况、时间与消费的限制等多角度的实际问题，单一的数学方法无法很好地解决本课题中涉及的实际旅游路线规划问题。

因此才有了将问题层层分解剖析，采用遗传算法、Dijkstra 算法和枚举法等多种方法结合的求解过程。根据最后的分析结果，该实际问题最终应选择 杭州→泰山→扬州→黄山这条路线，既能够保证 7 天的行程综合满意度最高，又能够在天气较差的日子避开对应的不适合旅游的景点，还能够满足去泰山看日出的要求。在本课题求解过程中，编程环境均使用 MATLAB，这也提高了我们对利用 MATLAB 进行运筹学领域问题编程求解的能力。

六、团队风采

图 21 成员合照

图 22 小组答辩人展示照片 1

图 23 小组答辩人展示照片 2

参考资料与参考文献

参考资料

梁军老师提供的《运筹学大作业：杭州国庆 5 天旅游路线规划设计》资料。

参考文献

[1]李俊纬. 物流配送 TSP 问题的研究[D]. 哈尔滨：哈尔滨工业大学控制科学与工程系, 2017 年

[2]张玲玉. 基于改进的遗传算法的物流配送系统研究与设计[D]. 邯郸：河北工业大学信息与电气工程学院, 2017 年

[3]王海英. 图论算法及其 MATLAB 实现[M]. 北京：北京航空航天大学出版社, 2010

[4] 刘啸. 基于运筹理论的旅游过程学讨论[J]. 首都师范大学学报（自然科学版）, 2012, 33(4) :77-82.

附录

附录 1：《假期国内最佳旅游路线规划》课题研究满意度调查表

景点满意度（值域 0—5）

	黄山	泰山	嵩山	三清山	三峡	扬州
A	3	5	1	1	4.5	5
B	4.5	4	4	4.5	3.5	5
C	3	5	4.5	1	1.5	5
D	4	3	5	2.5	4	2
E	4	4.5	3	2	1.5	5
F	4	2.5	4	4.5	5	3.5

两景点（包括起点杭州站）之间旅途时间（范围均为左开右闭区间）的满意度（值域-2—2）

	0-2h	2-4h	4-6h	6-8h	>8h
A	0	2	1	-2	-2

B	2	1.5	-0.5	-1.5	-2
C	2	2	2	0	-2
D	2	2	1	0.5	0
E	2	1	0	-1	-2
F	2	1.5	-0.5	-1.5	-2