# 杭州国庆5天旅游路线规划设计

**摘要：**本文主要研究在杭州市内的最佳旅游路线的设计问题。在满足相关约束条件的情况下，花钱最少且游览尽可能多的景点是我们追求的目标。

好不容易等来了国庆7天的长假，这也是大家外出旅游的好时机，但是大家都盼望着这7天的长假,所以选择长途旅行不是一个很好的选择,而且刚来杭州，杭州也是一个美丽幸福的城市，所以我们可以选择拿出7天中5天的时间（2天在旅游中休息）来了解一下杭州，参观一些杭州著名的景点，体味到杭州独特的魅力。

最终问题是我们要设计出一条在杭州市内的最佳旅游路线，基于此进行研究，建立了相应的数学模型，并通过lingo软件编程，计算出模型的结果，并结合模型进行分析,从而寻找出最佳的旅游路线,尽可能多的游玩杭州著名的景点，并且花费最少的钱。

在问题分析中，我们忽略在旅途中天气、交通等意外因素影响，并假设票价都是成人票，每天游玩一个景点，不然会太累。

**关键词：**最佳路线　运筹学　最小费用　模型求解

# 一、问题重述

杭州是我国的旅游城市，拥有丰富的旅游资源，吸引了大批的省外游客，越来越多的人选择到杭州旅游，旅行社也推出了各种不同类型的旅行路线，使得公众的面临多条线路的选择问题。

在这里我们选择了 7 个著名的景点来做我们的模型分析。他们分别是：杭州西湖区，六和塔，西溪国家湿地公园，龙井问茶，黄龙洞，杭州乐园，宋城。

以下是一些景区的图片：



## 二、问题分析

在旅游中我们要考虑的因素很多，比如天气，交通，费用等。在这里我们做最基础的假设：不考虑天气的影响，天气都是适合旅游的天气；在交通方面，我们假设不堵车，交通都是通畅的，交通的费用与两地之间交通所用的时间成正比。

每天旅游一个或两个景点，因此我们要把 7 个著名的景点都要游玩；在住宿和饮食方面的费用我们假设每天都是一样的价钱，所以就不考虑了；每天游览的时间我们假定是从早上 8 点到晚上 8 点。我们要追求的目标是在规定的时间内花费最少，所以我们在进行分析前对问题要做假设，假设不考虑旅游中发生的一切意外，即旅游按照所设计的路程顺利进行，在旅途中不考虑阴雨天气给旅游时间滞留带来的麻烦，并且把因为去旅游人员的多少带来门票打折的因素忽略了，在此基础上，我们通过约束条件规划方法，分析并写出了目标函数，并对路线进行简单分析，最终得到最优的旅游线路。

假设条件：

1、假设中途不发生任何堵车等交通事故以及天气等影响行程；
2、在途中和游览景点的时间为 12 小时，从早上 8 点到晚上 8 点。
3、在限定的时间内，最终要返回玉泉校区。
4、两地的交通花费与两地的交通时间成正比。
5、7 个著名的景点都要游玩。
6、所有的票价都是成人票价。
7、不考虑住宿和饮食的费用，因为在这里我们假设每天在这两项的费用是一样的。

## 三、符号说明

$i$，$j$——第 $i$ 个或者第 $j$ 个景点，　　$i$，$j$=1，2，…，8；

分别表示玉泉校区、杭州西湖区、六和塔、宋城、杭州乐园、黄龙洞、龙井问茶、西溪国家湿地公园。

$c$——每个游客的旅游总花费；

$t_i$——每个游客在第 $i$ 个景点的逗留时间；

$c_i$——每个游客在$i$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；

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

## 四、模型构成

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，使旅游者在最短时间内花最少的钱游览 7 个地方。

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：

$m$ ——每个游客的旅游总花费；

$m_1$ ——每个游客的交通总费用；

$m_2$ ——每个游客的旅游景点的花费；

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$r_{ij}$是判断游客们是否从第$i$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：

$$m_1 = \sum_{i=1}^{8}\sum_{j=1}^{8} r_{ij} \times c_{ij}$$

（2）旅游景点的花费

因为$c_i$表示游客在$i$个景点的总消费， 而这 7 个景点都要被游玩到。故我们先设景点上花费的总费用为$m_2$。

从而我们可以得到目标函数为：

Min $m = m_1 + m_2$

2、约束条件：

①时间约束

假设旅游时间应该不多于 5 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$j$个景点路途中所需时间，所以路途中所需总时间为$\sum_{i=1}^{8}\sum_{j=1}^{8} r_{ij} \times t_{ij}$；我们假定路上花费的交通费用与时间成正比。

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昆明，因此$\sum_{i=1}^{8}\sum_{j=1}^{8} r_{ij}$即表示旅游者的景点数，这里我们假定要旅游的景点数为 7，加上起点为 8。因此旅游景点数约束为：

$$\sum_{i=1}^{8}\sum_{j=1}^{8} r_{ij} = 8$$

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：

$$\sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \quad (i, j = 1,2,\cdots,8)$$

当$i=1$时，因为成都是出发点，所以$\sum_{i=1} r_{ij} = 1$；$j=1$时，因为代表们最终要回到成都，所以$\sum_{j=1} r_{ij} = 1$。

综合以上可知，

$$\begin{cases} \sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \\ \sum_{i=1} r_{ij} = 1 \\ \sum_{j=1} r_{ij} = 1 \end{cases} \quad (i, j = 1,2,\cdots,8)$$

同样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：

$r_{ij} \times r_{ji} = 0$　　（$i, j$=1，3，……，8）

综上所述，我们可以得到总的模型为：

Min　$m = m_1 + m_2$

约束条件：

$$\begin{cases} \sum_{i=1}^{8}\sum_{j=1}^{8} r_{ij} = 8 \\ \sum_{i} r_{ij} = \sum_{j} r_{ij} \le 1 \\ \sum_{i=1} r_{ij} = 1 \\ \sum_{j=1} r_{ij} = 1 \\ r_{ij} \times r_{ji} = 0 \end{cases} \quad \begin{matrix} \\ (i, j = 1,2,3\cdots,8) \\ \\ \\ (i, j = 1,3\cdots,8) \end{matrix}$$

## 五、模型求解与结果分析：

在这里，我们主要求解一种方法如何使在交通上花费的时间最少，也就是在交通上花费的费用最少，这里我们假设交通花费的费用与交通花费的时间成正比。

首先我们根据百度地图查询出来了一张各地地点到达的距离表。


距离表如下：单位是分钟。

| | 玉泉校区 | 杭州西湖区 | 六和塔 | 宋城 | 杭州乐园 | 黄龙洞 | 龙井问茶 | 西溪国家湿地公园 |
|---|---|---|---|---|---|---|---|---|
| 玉泉校区 | 0 | 20 | 50 | 60 | 84 | 20 | 50 | 80 |
| 杭州西湖区 | 20 | 0 | 40 | 60 | 120 | 20 | 40 | 80 |
| 六和塔 | 50 | 40 | 0 | 20 | 100 | 40 | 40 | 90 |
| 宋城 | 70 | 60 | 30 | 0 | 120 | 60 | 60 | 110 |
| 杭州乐园 | 130 | 130 | 100 | 110 | 0 | 130 | 120 | 180 |
| 黄龙洞 | 20 | 20 | 40 | 50 | 120 | 0 | 40 | 80 |
| 龙井问茶 | 50 | 40 | 40 | 60 | 120 | 40 | 0 | 100 |
| 西溪国家湿地公园 | 80 | 70 | 110 | 120 | 190 | 80 | 110 | 0 |

在这里，我们可以把如上的问题转化成一个典型的 TSP 问题，即旅行商问题，即 TSP 问题（Travelling Salesman Problem）又译为旅行推销员问题、货郎担问题，是数学领域中著名问题之一。假设有一个旅行商人要拜访 n 个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

求解 TSP 问题的常规算法是分治动态规划，贪心法，回溯法，分枝限界法，近似算法，概率算法，现代启发算法（包括遗传算法，蚁群算法，模拟退火算法等）。在这里，我们主要介绍 LINGO 软件自带的一个算法和遗传算法，蚁群算法。

1. LINGO 软件编程求解：

   代码如下：

   即从 1->5->4->3->7->8->2->6->1。

   即玉泉校区-杭州乐园-宋城-六和塔-龙井问茶-西溪国家湿地公园-杭州西湖区-黄龙洞-玉泉校区。

2. 介绍遗传算法求解。

   遗传算法是模仿自然界生物进化机制发展起来的随机全局搜索和优

化方法，它借鉴了达尔文的进化论和孟德尔的遗传学说。其本质是一种高效、并行、全局搜索的方法，它既能在搜索中自动获取和积累有关空间知识，并自适应地控制搜索过程以求得最优解遗传算法操作使用适者生存的原则，在潜在的解决方案种群中逐次产生一个近视最优方案。在遗传算法的每一代中，根据个体在问题域中的适应度值和从自然遗传学中借鉴来的再造方法进行个体选择，产生一个新的近视解。这个过程导致种群中个体的进化，得到的新个体比原个体更适应环境，就像自然界中的改造一样。

## 算法流程

编码：解空间中的解数据 x，作为作为遗传算法的表现型形式。从表现型到基本型的映射称为编码。遗传算法在进行搜索之前先将解空间的解数据表示成遗传空间的基本型串结构数据，这些串结构数据的不同的组合就构成了不同的点。

初始种群的形成：随机产生 N 个初始串数据，每个串数据称为一个个体，N 个串数据构成了一个群体。遗传算法以这 N 个串结构作为初始点开始迭代。设置进化代数计数器 t 0;设置最大进行代数 T；随机生成 M 个个体作为初始群体 P(0)。

适应度检测:适应度就是借鉴生物个体对环境的适应程度,适应度函数就是对问题中的个体对象所设计的表征其优劣的一种测度。根据具体问题计算 P(t)的适应度。

选择：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。

交叉：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。

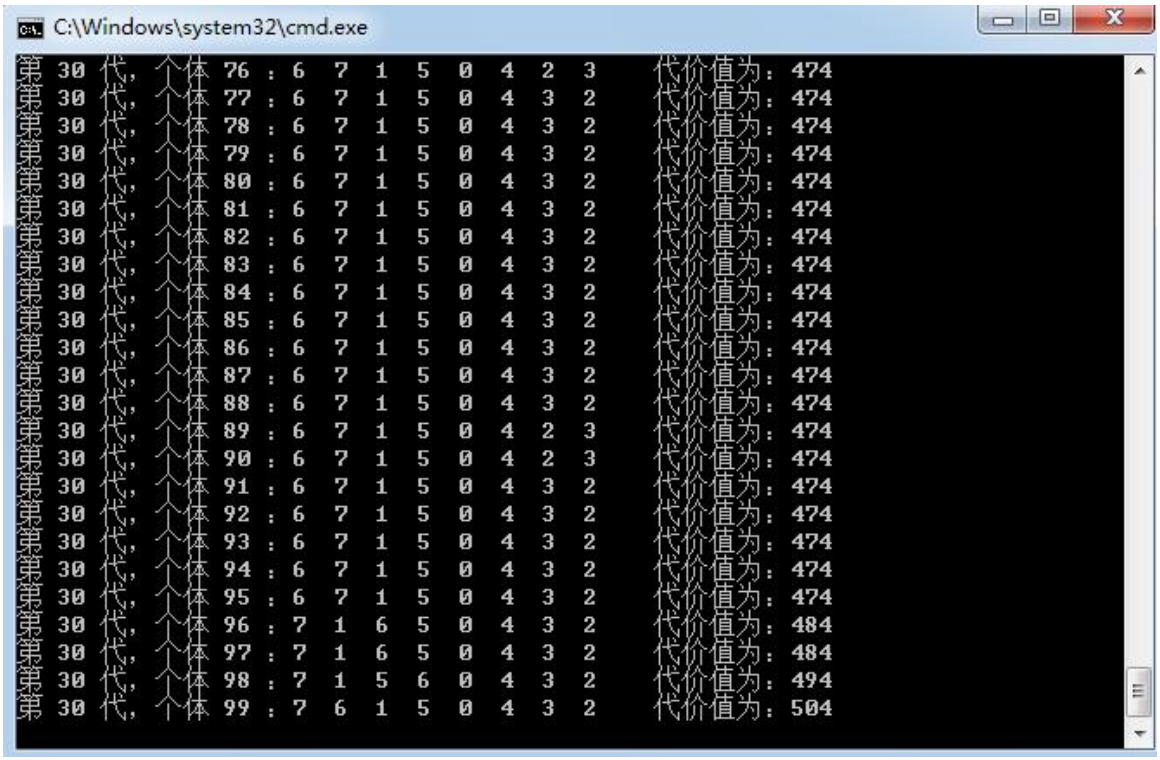
变异：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。

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)。

终止条件判断:若 t<=T,则 t=t+1，转到第 3 步，否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。

C 语言程序代码：

结果分析如下图：



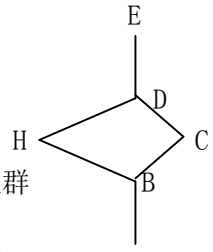

由图我们可知，遗传算法收敛于 474。与 LINGO 编程求解的结果是一致的。

### 3. 蚁群算法求解

#### 蚁群算法原理

蚁群算法是受到对真实的蚁群行为的研究启发而提出的，像蚁群、蜜蜂等群居昆虫，虽然单个昆虫的行为很简单，但是组成的群体却表现出极其复杂的行为。仿生学家经过大量细致观察研究发现，蚂蚁个体之间是通过一种称为外激素的物质进行信息传递的，蚂蚁在运动过程中，能够在它所经过的路径上留下外激素，而且蚂蚁在运动过程中能够感知这种物质，并且以此指导自己的运动方向，所以，大量的蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象。我们并不想完全模拟蚁群，而是对使用人工蚁群方法来解决优化问题感兴趣因此，我们的蚁群与实际的蚁群有三个主要的区别：

- 人工蚁群具有记忆性，
- 人工蚁群不是完全盲目的，
- 人工蚁群处在离散的时间环境中。



图(1)

虽然有区别，我们仍然可以用蚂蚁群的行为来形象地说明人工蚁群算法的原理。如图(1)所示，设ＤＨ＝ＨＢ＝１，ＤＣ＝ＣＢ＝0.5。我们假定在每个离散的等时间间隔：ｔ＝０，１，２，……有３０个蚂蚁从Ａ到达Ｂ，同时有３０个蚂蚁从Ｅ到Ｄ，每只蚂蚁的速度为１／Ｓ，并且，每有一只蚂蚁经过时，在时间ｔ留下信息素密度为１。

蚂蚁在选择路径时，那些有更多蚂蚁曾经选择过的路径（也就是具有更高信息素密度的路径），被再次选中的可能性最大。

当ｔ＝0时，没有信息素，有３０只蚂蚁分别在Ｂ和Ｄ。蚂蚁走哪条道路是完全随机的。因此，在每个点上蚂蚁将有１５只经过Ｈ，另外１５只经过Ｃ。

当 t ＝ 1 时有 3 0 只蚂蚁从 A 到 B，它们发现指向 H 道路上的信息素密度是 1 5，是由从 B 出发的蚂蚁留下的；指向 C 道路上的信息素密度是 3 0，其中 1 5 是由 B 出发蚂蚁留下，另外 1 5 是从 D 出发经过 C 已经到达 B 的蚂蚁留下。因此，选择经过 C 到 D 的可能性就更大，从 E 出发到 D 的 3 0 只蚂蚁也面临着同样的选择，由此产生一个正反馈过程，选择经过 C 的蚂蚁越来越多，直到所有的蚂蚁都选择这条较近的道路。蚁群算法就是利用蚂蚁的这一特性，解决最优化问题。

## 蚁群算法解决 T S P 问题

我们来介绍一下如何用蚁群算法求解 n 个城市的 T S P 问题。设 $d_{ij}$ 为城市 i， j 之间的几何距离，$d_{ij}=\left[\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2\right]^{1/2}$。设 $b_i(t)$ 表示 t 时刻位于城市 i 的蚂蚁的个数，蚂蚁总数 m＝$\sum_{i=1}^{n} b_i(t)$，$\tau_{ij}(t)$ 表示 t 时刻在 i j 连线上残留的信息量，初始时刻各条路径上的信息量为 $\tau_{ij}(0)$＝C （C 为常数）。用参数 $\rho$ 表示信息量的保留度，则经过 n 个时刻后，路径 i j 上的信息量根据下式作调整：

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij} \tag{1}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{2}$$

$\Delta\tau_{ij}^{k}$ 表示第 k 只蚂蚁在本次循环中留在路径 i j 上的信息量，$\Delta\tau_{ij}$ 表示本次循环所有经过的蚂蚁留在 i j 上的信息量。

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k} & \text{当第 k 只蚂蚁经过 i j 时}\\ 0 & \text{当不经过时}\end{cases} \tag{3}$$

定义 $\eta_{ij}$＝1 / $d_{ij}$ 。蚂蚁 k （ k＝1， 2，…， m）在运动过程中，$p_{ij}^{k}$ 表示在 t 时刻蚂蚁 k 由位置 i 转移到位置 j 的概率：

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}(t)}{\sum_{s\in allowed_k}\tau_{is}^{\alpha}\eta_{is}^{\beta}(t)} & j\in allowed_k\\ 0 & \text{其他}\end{cases} \tag{4}$$

我们用 $tabu_k(k=1,2,\cdots,m)$ 记录蚂蚁 k 目前已经走过的城市集合，allow $d_k$ 表示蚂蚁 k 下一步允许选择的城市集合，它等于全部的城市集合除去第 k 只蚂蚁已走过的集合 $tabu_k$。定

义$L_k$为第k只蚂蚁在本次循环中走过的路径和。

用蚁群算法解决TSP问题是一个递推过程，当t＝0时，将蚂蚁放在各城市，设定每条路径上的信息量初值$\tau_{ij}(0)$＝C，每只蚂蚁根据公式⑷决定的概率从城市i到城市j。$\tau_{ij}(t)$表示曾经有多少蚂蚁经过路径（i，j）；$\eta_{ij}$说明较近的城市有更大的可能性被选中。α，β用来控制两者对蚂蚁选择的影响力程度。经过一个循环后，根据公式⑴⑵⑶计算更新每条路径的信息量$\tau_{ij}(t)$。将所有的$tabu_k(k=1,2,\cdots,m)$复原，最后求出本次循环的最短路径min $L_k$。这个过程不断重复，直到所有的蚂蚁都选择同样的路径，或者循环次数达到预先设定的最高次数$NC_{\max}$。

解决n个城市的TSP问题算法设计如下：

1．初始化：

设定t＝0，循环计数器NC＝0，对每条路径设定初始信息量$\tau_{ij}(0)$＝C，$\Delta\tau_{ij}$＝0

将m只蚂蚁放在n个城市上（为了使问题简化，设定m＝n）。

2．设定tabu集合的索引s＝1，对k从1到m，把第k只蚂蚁放在起始位置，对应的设定集合$tabu_k(s)$

3．重复下面的步骤，直到集合tabu满为止（这一步将重复n－1次）：

设定s＝s＋1；

对k从1到m，根据公式⑷确定的概率，选择下一步移动的目标城市j｛在时间t时，第k只蚂蚁所在的城市是i＝$tabu_k(s-1)$｝；

将第k只蚂蚁移到城市j；

把j加入到集合$tabu_k(s)$中。

4．对k从1到m：

将第k只蚂蚁从$tabu_k(n)$移动到$tabu_k(1)$；

计算第k只蚂蚁所走过的路程和$L_k$，并更新最小路径min $L_k$；

对每条路径（i，j）：$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{L_k} & \text{当第k只蚂蚁经过i j时}\\ 0 & \text{当不经过时}\end{cases}$

$$\Delta\tau_{ij}=\Delta\tau_{ij}+\Delta\tau_{ij}^k;$$

5．对每条路径（i，j）根据$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij}$计算$\tau_{ij}(t+n)$；

设定t＝t＋n；

设定ＮＣ＝ＮＣ＋１；

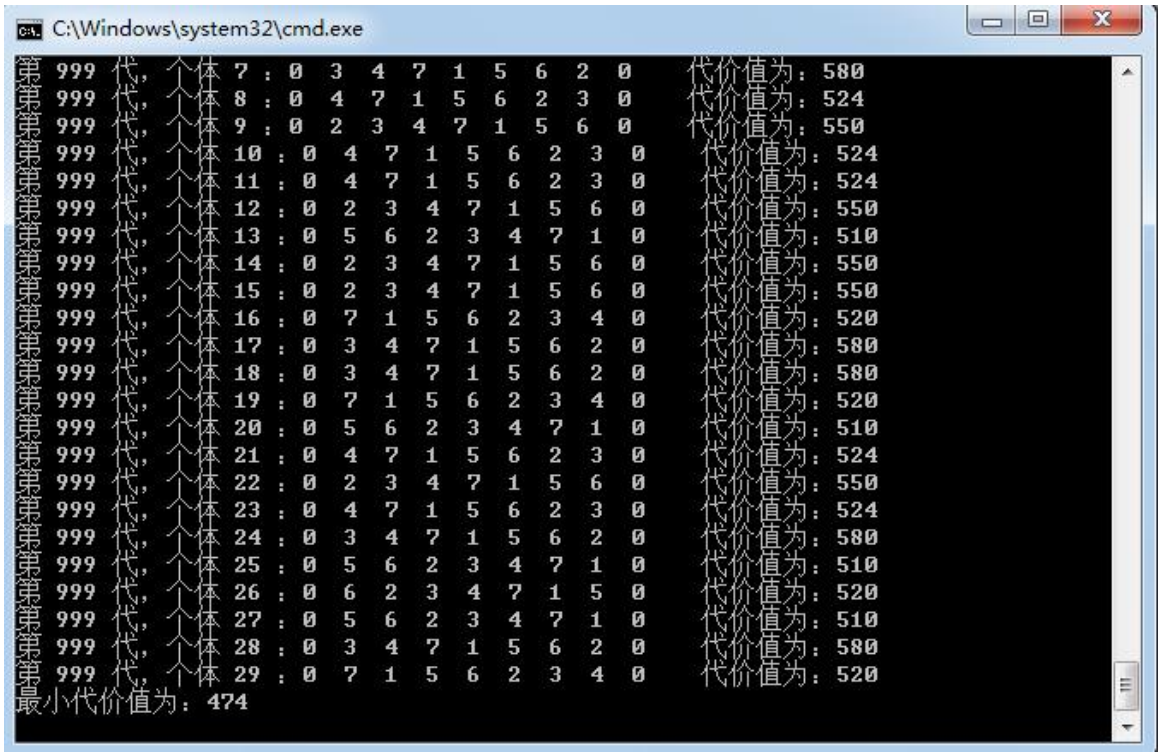
对每条路径（ｉ，ｊ），设定$\Delta\tau_{ij}$＝０。

６．如果ＮＣ＜$NC_{\max}$，

则清空所有的集合ｔａｂｕ
转到第二步；
否则，得出最短的路径。

C 语言程序如下：

程序结果如下：




虽然找出来最小值 474，但是好像蚁群算法并没有收敛于最小值，这里我修改了很多次程序与参数，但是还是没有找到问题的所在，我想这需要更多的学习与研究。

## 六．心得与结论

本文通过一个具体的与生活相关的实例讨论了一个运筹学的基本问题 TSP 问题，该问题在生活中有很多的运用，比如题中的如何安排国庆假期的旅游计划，比如如何规划一天的生活作息等，然后建立了该问题的一个具体的数学模

型，然后用 LINGO 软件和 C 语言进行了程序的编写与仿真，得出了一致的结论。但是，由于数据收集的准确性和一些假设的前提，该模型的出来的结果不一定是实际上最优化的结果，仍然有它需要改进与提高的地方。最后，在蚁群算法的仿真上，程序的结果出现了不收敛于最小值的情况，可能出现了局部收敛，这是蚁群算法的一个缺点之一，另一个缺点是蚁群算法的收敛速度较慢，遗传算法差不多只要 30 代左右就得到了最终的收敛值。所以，这也是另一个值得讨论与学习的地方。

# 参考文献

[1] 林家恒　贺庆　《一种求解 TSP 的蚁群算法》 [j]
[2] 遗传算法总结