

桂林电子科技大学

第十八届大学生数学建模竞赛

承 诺 书

我们仔细阅读了桂林电子科技大学第十八届大学生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

参赛队员信息：

序号	姓名	学号	专业	电话号码
队长	韦纯福	2000710337	信息与计算科学	17877850692
组员	魏鹏波	2000710338	信息与计算科学	15719388475
组员	刘浩	2000710125	信息与计算科学	13536577085

基于蚁群算法的多目标规划

摘 要

本文针对游客游玩景区的满意度评价特性，考虑行程安排、景点、餐饮购物、时间的分配等因素，针对评价指标建立了路径与满意度结合的多目标规划模型，通过尝试以路径安排的合理性，路途中的游客满意度的提升作为优化目标，通过结合蚁群算法、Dijkstra 算法对模型进行求解。并且通过对三个问题的具体分析，引入不同的约束条件，得到每个问题的解决方案。

首先对数据进行预处理，将园区地图转换成几何模型进行处理，结合实际的路程、景点等信息，为每条路径赋予不同的权重信息从而进行后续的分析与处理。

对于问题一，通过将总体游玩园区路径最短，路径上的景点数目、餐厅分布、特色商品和公共设施作为评价指标，并在此基础上假定游客的游玩路线是单向的，并且从一个大门进入后一直沿着一个大致方向进行游玩，最后将从剩余的大门之一离开园区。进一步，对两个目标进行权重的分配，最终得到综合考虑满意度与路径最短的模型，并给出对应的求解方法。

对于问题二，需要进一步的考虑园方的表演时间的安排对游客的路线安排的影响。由于园方的表演时间并未是一个固定的时间段，而是动态变化不断调整的。基于此，通过将整个园区的几何模型中表演的地点进行标注，在第一问的基础上，对游客路线上经过表演地点进行约束，使得在有限的时间内，游玩时更多的经过表演的场所从而保证游客能够观赏到对应的表演。将目标函数进行对应的权重比较、处理，从而得到对应的游玩路线。

对于问题三，由于疫情的限制，导致了园区的容纳人数在正常情况下进一步的降低。在此基础上，我们考虑将目标函数进行修改。通过在保证满意度一定的情况下，保证游客的正常体验的前提下，引入定义路段上的游客人数与路段人口容量之比为段拥挤度这一维度，并且将其处理为随着路段上的游客人数实时变化的动态变量。在原先模型的基础上，引入路程的拥挤度这一因素对模型进行限制处理。并基于此建立关于疫情防控时期的路径模型。

关键词：最优化模型、多目标规划、蚁群算法、Dijkstra 算法

一、前言

1.1 问题背景

自改革开放以来，国内的经济焕发了前所未有的活力。国内经济发展的突飞猛进也带动了人们生活水平的提升，旅行已经成为了我们在日常休闲生活中的一部分。但与此同时，一些问题也随之而来，在人们进行游玩的时候，总希望能够在有限的休闲时间中去游览更多的景点。另一方面，从景点管理者的角度来说，了解人们的路线选择能够有效的帮助馆方进行资源的合理分配从而获取更好的口碑吸引人群。设计一条科学合理的游玩路线从各个角度来说都是十分重要的。一条优秀合理的路线在能够满足游客的需求同时，还可以在面对一些特殊时期的要求时帮助园方进行各条路线上的人流量的管理，更好的对各类资源进行分配。

在有限的时间内进行多个景点的游览，并且同时需要尽可能的获得更高的体验感是每个游客所追求的。单纯将路线最短是不充分的，人们在游玩过程中更注重各个景点的游览体验，那么在规划路线的同时如何考虑满意程度将是一项重要的指标，例如考虑路线上的特色商品的宣传、人文氛围、公共设施的搭配等。游客在不同的需求选择下将得到不同的路线。

1.2 问题重述

根据题目背景以及附件解决如下问题：

1. 通过查阅相关文献资料，用自己的语言描述一个你认为比较理想的游览路线规划模型。要求有具体的数学模型表达、解释和相应的求解方法，同时应标注清楚所描述的模型参考的文献来源。

2. 附件 1、附件 2 给出了广州长隆欢乐世界里长隆野生动物园的地图和园内一天的表演时间段，依据问题 1 所描述的模型寻找一条在规定的时间内（一般为园区的一个工作日）的合适游览路线，给出你的具体分析和计算过程，值得注意的是满意程度的体现。（提示：长隆野生动物园内各景点间的路线可在百度地图测量获得）

3. 由于疫情防控原因，整个园区内不能在同一个区域聚集过多人员，从园区的管理方面，还需考虑多种不同的游览路线以引导游客进行游览，请在以上两问题基础上基于合理的数学建模方法提出一套合理的游客游览管理方案以供园区参考（参考：无疫情时，长隆动物园日最大接待能力为 15 万人）。

二、问题分析

2.1 概论

在进行路线的选取时，我们需要的就是在有限的资源条件下，将游客的体

验最大化，很明显，这是一个多目标的规划模型。我们在考虑游客精力有限、馆方的表演投入一定的情况下，考虑到不同需求的游客的实际情况，得到了多个目标函数。进一步，我们对目标函数的权重进行了相关处理，通过客观的评价体系进行用户需求指标的权重量化。在此基础上，将初步得到的模型引入更多的客观约束条件进行完善。

2.2 问题一的分析

对于合理规划游览路线，设计路线要考虑不同的因素，单纯的考虑路线最短是不够的，人们在游玩过程中更注重各个景点的游览体验，针对以上的多种情况，若同时对景区内所有景点都进行设计的话，计算复杂度太大，不具有实用性和操作性，所以我们考虑用结合归一的方式，在相应的约束条件下，考虑游客从园内中的三个大门中任意一个门进入，在园区内进行游玩活动后，将从剩下的大门之一离开园区。除此之外，我们将小的景点进行归一处理，把一条路线上很多的小景区与这条路线的长度归为一个标准，以一个统一的标准来得出最优的路线，将整个园区的地图转换几何模型，便于后续处理，其中的结点是园区内道路的分叉口以及各种大型景区。

将数据与地图转换完成后，我们就得到了一个整个景区抽象出来的几何模型，不难看出，整个模型的求解过程对应最短路径问题。并且结合满意度的目标优化，基于此进行模型的求解。

2.3 问题二的分析

在游客路线的规划中，基于前两问的路径最短、满意度水平的两个对应的目标函数基础上，引入了关于时间的约束条件，对游客的游玩路线进行处理。我们希望在保证前两个目标的基础上，希望得到的路线更加多的经过园方的表演场地，从而在普遍的情况下，游客能够有更高的概率能够观赏到对应的表演。从而对模型进行优化处理，使其更加具有普遍性。此外，对两个目标函数的处理中，通过为两者分配合理的权重从而通过第一问模型的求解方法得到基于园方表演安排的游玩路线。

2.4 问题三的分析

第三问的题目要求基于疫情情况下对园区路线进行安排。通常情况下游客是一个动态的流动，需要考虑单位时间内游客的流出量，流出量，并建立微分方程模型进行预测的同时，还需要对单位时间内将人群在园区内通过优化模型得到对应的浏览路线。流入量、流出量、园区本身的储存量三者构成了一个动态的模型。

对此，我们为了保证模型的通用性，舍弃了人口流动的动态性，在原先的模型基础上进一步的进行修改。

在游客游玩中运动聚集过程中的拥挤状况一般用拥挤度表示。为反映结点 i 到结点 j 路段的拥挤情况，为此、我们定义路段上的游客人数与路段人口容量之比为路段拥挤度，且随路段上的游客人数实时变化。基于此建立对应的引入路段拥挤度的模型。

三、 模型假设与符号表示

3.1 模型的相关假设

1.通过百度地图所收集到的数据准确可靠

在进行模型的构建中，我们使用到的各景点之间的数据信息是通过百度地图中进行测算得到的，不考虑由于导航系统、信号波动与经纬度与实际距离单位转换等因素造成的误差。

2.不考虑交通拥堵、天气变化、意外事故等不可抗力因素造成的时间耽搁

实际上游客进行游览时，由于天气的突然变化、人群的拥堵聚集以及各种客观因素造成的时间耽搁以此引发的园方的活动的延迟是无法控制的，对此我们不对其进行考虑。

3.认为游览期间游客已经处理完其他事情

在游览时，由于各种游客各种的突发情况例如工作、家庭琐事导致的路线发生改变是具有随机性的，在此我们排除此因素。认为游客游览期间专注于游玩本身，路线的起点的终点都是在园区内，不考虑游玩时因为意外情况原路返回离开园区，游玩的方向是相对稳定的。

4.游客游玩的方向并未进行区分

游客进行园区的游玩时，可以从三个大门任意之一进入作为起点，终点为剩余的两个大门之一。同时未强制规定游客一定要遍历整个园区。

3.2 模型的符号表示

符号	含义	单位
x_{ij}	表示从第 <i>i</i> 个路口到第 <i>j</i> 个路口	\
p_{ij}	表示从第 <i>i</i> 个路口到第 <i>j</i> 个路口的满意度	\
d_{ij}	表示从第 <i>i</i> 个路口到第 <i>j</i> 个路口的路程	<i>m</i>
m_{ij}	表示从第 <i>i</i> 个路口到第 <i>j</i> 个路口之间的小景点数	\
t_{ij}	表示从第 <i>i</i> 个路口到第 <i>j</i> 个路口所需要的时间	min
$t_i + t_j$	表示从第 <i>i</i> 到第 <i>j</i> 个路口中的景点处停留时间	min
v	表示一个成年人正常旅游时的行走速度(1.2)	<i>m/s</i>
T	表示游客在一个工作日游玩所需时间	<i>min</i>
α_i	表示从景区入口处到 <i>i</i> 结点所需时间	<i>min</i>
α_j	表示从景区入口处到 <i>j</i> 结点所需时间	<i>min</i>
$\Delta\tau$	表示结点 <i>i</i> 到 <i>j</i> 的时间差	<i>min</i>

四、模型建立与求解

4.1 数据预处理

在进行模型的构建之前，我们将园区的整个地图的几何模型进行的重新优化处理，结果如下图所示：

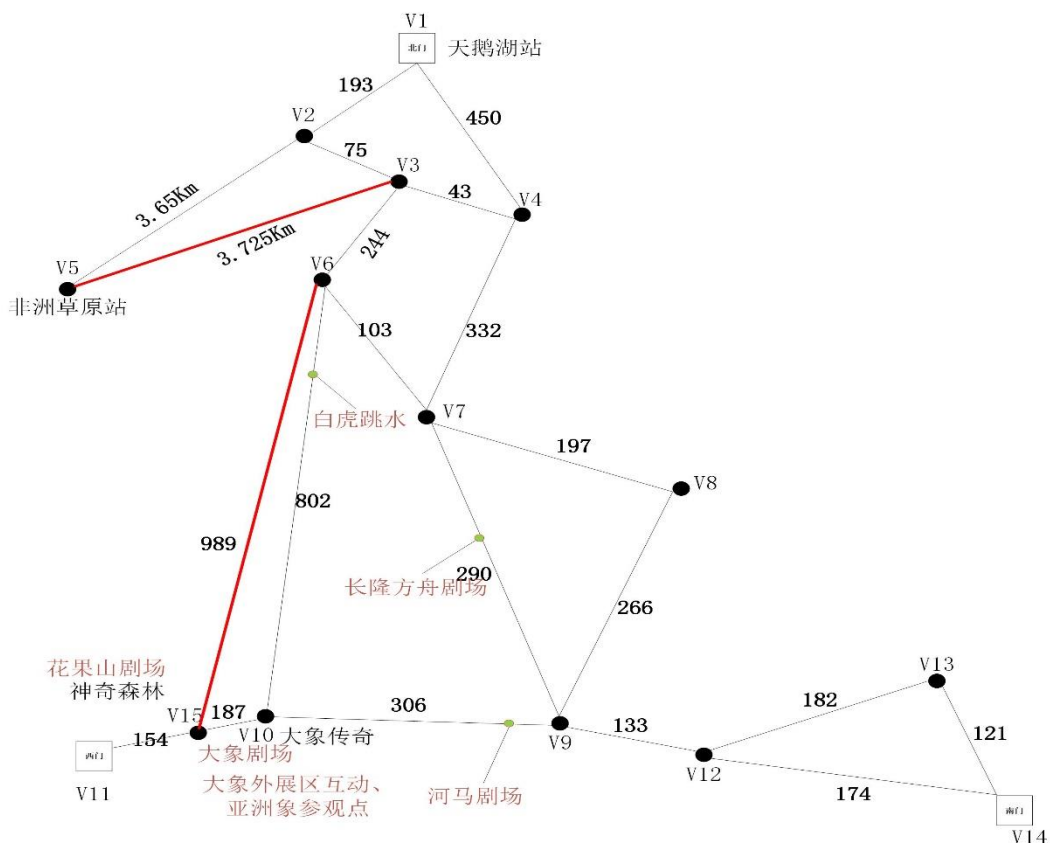


图 1 园区地方的几何模型

在模型的构建中，我们将地图中的岔口、园区大门以及部分的景点当成结点进行处理。并且，通过百度地图的测量工具，我们将各个路口之间，路口与景点之间以及园区大门和路口之间的距离以及测量得到，并将其作为边的权重进行了标注。默认单位是米，如有特殊情况已在图中标注出相关单位。

为了方便后续问题的处理，我们将小景点当作图的另一权值进行处理。主要关心的重要景点已在图中进行了说明。园方的后续展览点已通过红色地点进行标注，大门也已标出。

特别的，为了优化模型，我们人为的添加了两条边，分别是 v_5-v_3 以及 v_6-v_{15} ，均已在图中红色线加粗描述。它们分别代表的含义为：

- 如果模型计算出了从 v_5-v_3 的一条路径，那么实际的路径为 v_5-v_2 ， v_2-v_3 这一路径。
- 如果模型计算出了从 v_6-v_{15} 的一条路径，那么实际的路径为 $v_{10}-v_{15}$ ， v_6-v_{10} 这一路径。

此外：

- 在结点的处理过程中, 在一些相近的景区之间, 例如大象传奇与亚洲象参观点, 由于两个景区充分接近, 将其处理成一个结点在实际上对游客

游玩时不会产生过大影响，故我们将其统一起来作为一个结点从而进行模型的简化处理。例如结点 v_{10} 实际上代表着两个景区。

- 边的实际方向由游客的游玩的实际情况决定，在此处理为无向图，但是在模型的计算过程中，由于游客的游玩的起点已经随机生成，终点在模型假设中已经定义过，我们在实际求解过程时将其当成有向图进行处理。

另外一方面，我们也对我们规定的路径上的小景点数进行了统计，结果如下图所示：

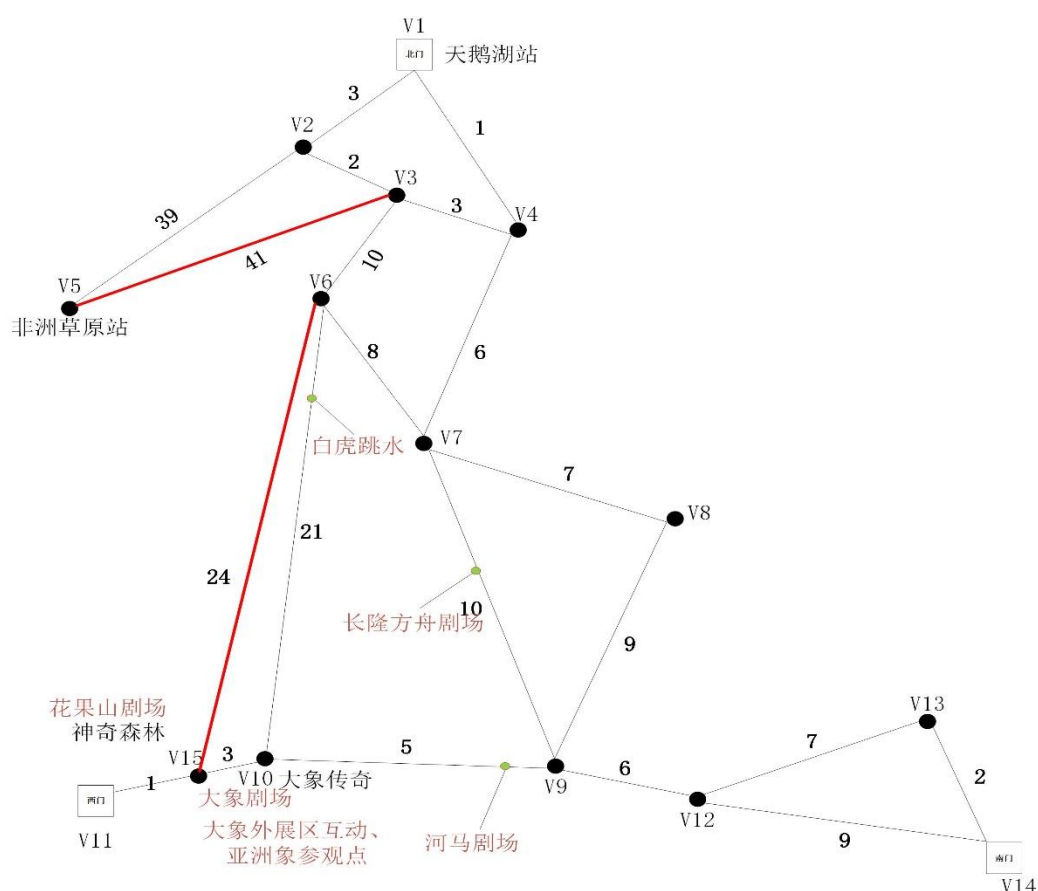


图 2 园区路径景点信息

进行模型的处理的时，我们将上述图转换为邻接矩阵进行存储，其中矩阵的数据为两个结点之间的实际距离，若结点之间不同直达，我们将二者之间的距离处理为无穷($\text{inf}=9999$)。

inf	193	inf	451	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	75	inf	3650	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	inf	43	inf	244	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	332	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	3725	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	103	inf	inf	802	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	197	290	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf	266	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	306	inf	133	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	306	inf	187	inf	inf	341
inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	182	174	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	182	inf	121	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf

进一步，对动物园的表演场所、时间进行标注，如下表所示：

表 1 动物园表演时间表

时间	地点
10:30	白虎跳水
11:00	白虎跳水、花果山剧场
11:30	白虎跳水
11:45	河马剧场
12:00	大象外展区互动、白虎跳水
12:30	花果山剧场（六，七）、白虎跳水
13:00	大象剧场、白虎跳水
13:30	白虎跳水
14:00	大象外展区互动、白虎跳水
14:30	长隆方舟剧场
15:00	大象外展区互动、花果山剧场、白虎跳水
15:30	河马剧场、白虎跳水
16:00	大象剧场、白虎跳水
16:30	长隆方舟剧场、白虎跳水
17:00	大象剧场（六，七）、大象外展区互动（一，五）、白虎跳水

4.2 问题一的模型建立与求解

● 确定问题的决策变量

我们将整个园区的地图处理成一个网格图。若旅客选择走从路口*i*到路口*j*之间的路线，则 x_{ij} 为 1，否则，则 x_{ij} 为 0。引入 0—1 整数变量表示如下：

$$x_{ij} = \begin{cases} 1, & \text{从第 } i \text{ 个路口走到第 } j \text{ 个路口} \\ 0, & \text{否则} \end{cases}$$

● 确定目标函数

由题意不难得到，我们需要得到的是一条让游客的满意度最大的一条旅游路线，那么我们便将游客满意度确定为此问中的目标函数。首先，我们将考虑一条优秀的路线基本的前提就是能够保证路线尽可能的短，从而保证用户的基本体验感。我们将整个园区的网络图结点的个数记为 n ，从而得到其中一个优化目标函数记为 z_1 ：

$$\text{MIN } z_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

另一方面，仅仅考虑路线最短对游客的体验感而言是不充足的。我们进一步考虑了游客在游玩路线上的景点数目，保证在相对短的游玩路线中能够使游客更多的欣赏到更多的景点。我们将从第 i 个园区地图的路口到第 j 个路口的满意度记为 p_{ij} ，并且定义满意度定义为两个路口形成的这一条路径上小景点的数目与路程的比值。同时，考虑到影响游客满意度的有多种因素，在行程本身外，也要考虑住宿、餐饮、购物^[1]等。对此，我们将路径上的餐饮、公共基础设施当成一个景点进行处理。得到第二个目标函数^[5]，记为 z_2 ：

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}}\right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

● 确定约束条件

首先，在之前的模型假设中，我们认为游客游玩的路径时路径是单个方向的，由此我们得到了两个约束条件：

$$\sum_{i=1}^n x_{ij} = 1, j = 1, 2 \dots n$$

$$\sum_{j=1}^n x_{ij} = 1, i = 1, 2 \dots n$$

这两个约束条件表示游客只到达一次分岔口，并且也只离开一次。

综上，我们得到了多目标的规划模型：

$$\text{MIN } z_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}}\right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\begin{aligned}
s.t \quad & \sum_{i=1}^n x_{ij} = 1, j = 1, 2 \dots n \\
& \sum_{j=1}^n x_{ij} = 1, i = 1, 2 \dots n \\
& x_{ij} = 0 \text{ 或 } 1
\end{aligned}$$

● 模型的求解方法

对于最优化问题，随着人们研究的深入，逐渐意识到在很多情况下想要完全准确的求出最优解是不现实的。基于此，人们研究出了求解最优化问题的众多方法。进一步我们考虑算法运行效率、结果的收敛性等特点^[2]，采用蚁群算法进行模型的求解，将结点之间路程上的小景点数与路程的比值为路程的权重，即满意度；来规划该路径下的满意程度，利用蚁群算法求得在该满意度矩阵下的最短路径。

● 蚁群算法基本介绍

蚁群算法是通过模拟自然界蚂蚁觅食行为来求解优化组合问题的仿生算法。意大利学者 M.Dorigo 于 1992 在博士论文中第一次提出蚁群算法概念。其原理在于：自然界的蚂蚁群在寻觅食物的过程中，会在途经道路上释放信息素，根据道路已有信息素选择路径。经过搜索，道路信息素浓度不断更新和消散而最终收敛于某条最短路径，此过程表现出了蚂蚁群体自组织、自动学习的能力。算法本质是一种正反馈机制，蚂蚁倾向于选择信息素浓度较大的路径作为自己移动方向。因此环境是否变化，蚁群总能挑选合适路径，实现选择最短路径目的^[3]。

● 数学模型

设定标记如下： m 表示蚁群个体数量； d_{ij} 表示位置 i 和 j 之间的距离； $\eta_{ij}(t)$ 表示启发函数即由位置 i 转移到 j 的启发程度； $\tau_{ij}(t)$ 表示 t 时刻边 (i, j) 上信息素量， $\tau_{ij}(t) = C$ ， C 取常数； $\Delta\tau_{ij}(t)$ 表示一组蚂蚁遍历完成后位置 i 到 j 间新产生的信息素量； P_{ij}^k 表示蚂蚁 k 的转移概率， j 是未访问的节点； $table_k$ 表示蚂蚁 k 下一步允许运行的节点集合。

① 初始时刻 $t=0$ ，各条路径信息素浓度相等，蚂蚁 k 在运动过程中根据各条路径信息素浓度计算转移概率：

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in J_k(i)} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & \text{如果 } j \in J_k(i) \\ 0, & \text{否则} \end{cases}$$

其中 α 是信息启发因子, β 是期望启发因子, 然后蚂蚁 k 会根据概率 P 选择下一步节点, 其中 p_{ij}^k 表示该节点被蚂蚁 k 选中的概率。

② 为了防止路径信息素过多造成残存信息 屏蔽启发信息, 在一组蚂蚁遍历完成后, 要对残存信息素进行更新, 更新规则为:

$$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ji}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{第 } k \text{ 只蚂蚁在该轮经过}(i, j) \\ 0 & \text{否则} \end{cases}$$

其中 $\rho(\rho \in [0,1))$ 表示信息素挥发系数, $\Delta\tau_{ij}^k(t)$ 表示本轮循环中蚂蚁 k 在位置 i 和 j 之间 新增的信息素量, Q 表示信息素强度, L_k 表示本轮循环中蚂蚁 k 走过路径的总长度。

② 经过若干轮的循环之后, 该算法会收敛, 蚂蚁在信息素正反馈的指引之下最终会选出一条优化路径作为全局最优解。

算法流程图如下:

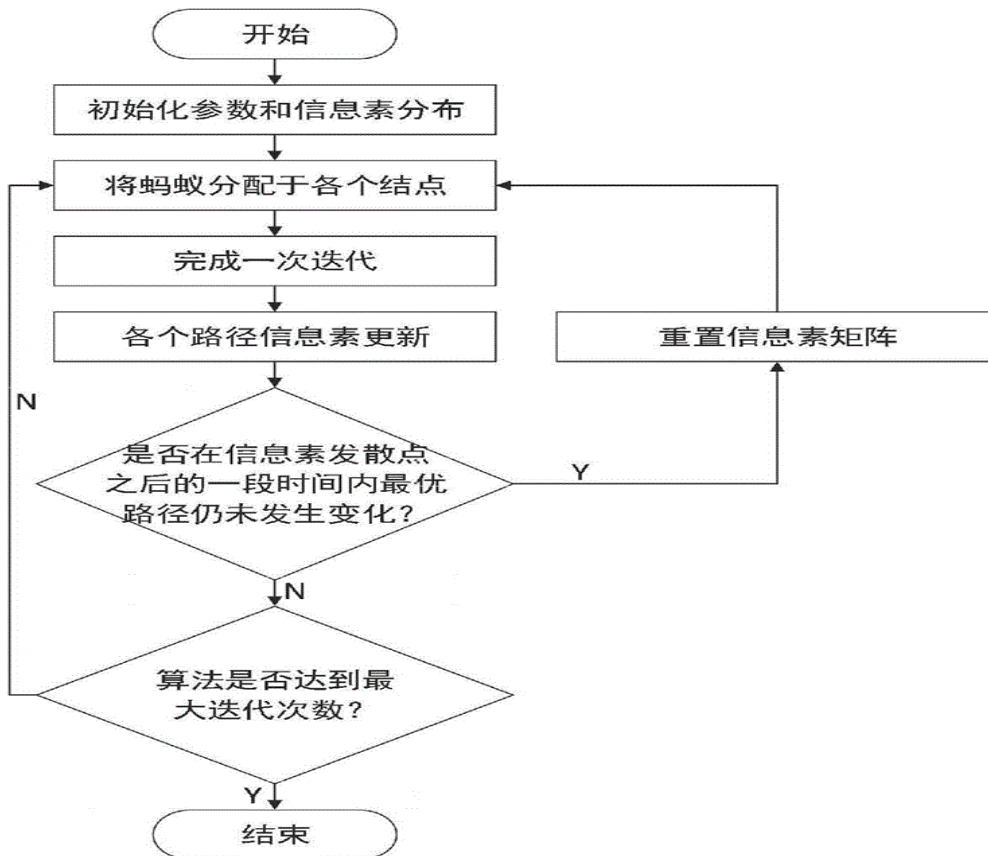


图 3 蚁群算法流程

利用该蚁群算法求解模型得到的最优、平均函数变化趋势如下图：

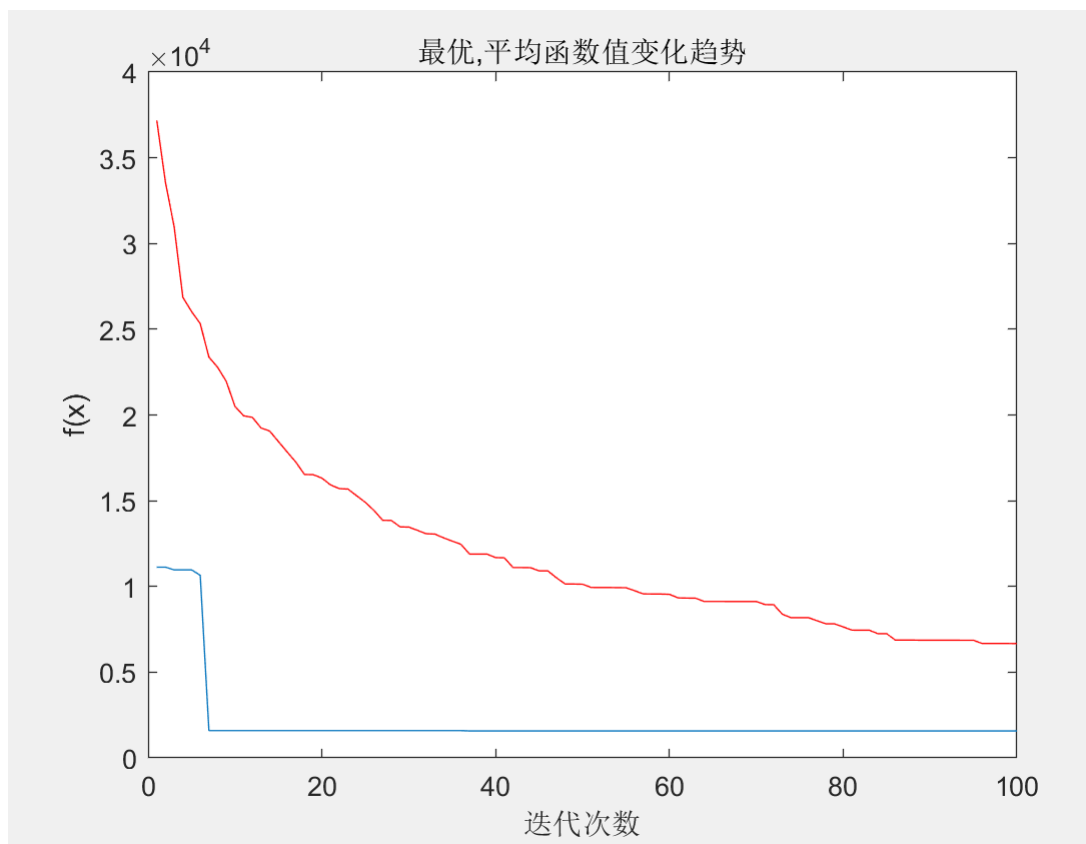


图 4 最优函数值与迭代次数之间的关系

基于满意度下的最短路线结果如下图所示：

```
Shortroad is 1    2    3    6    7    9    10    15
Mininum is 1573
```

图 5 基于满意度下最短路径搜寻结果

其代表的含义为：

◇ V1(北门)结点——> V2 结点——> V3 结点——> V6 结点——> V7 结点
——> V9 结点——> V10 结点——>V15 结点(西门)
最短路程为 1573m。

不考虑满意度情况下，寻找最短路径结果如下图所示：

```
Shortroad is 1    2    3    6    10    15
```

图 6 最短路径搜寻结果

其代表的含义为：

✧ V1(北门)结点——> V2 结点——> V3 结点——> V6 结点——> V10 结点——>V15 结点(西门)

4.3 问题二的模型建立与求解

● 确定问题的决策变量

对问题二的处理模型的构建时，仍是基于第一问基础上进行模型的改进。因此，问题二中的决策变量仍然与问题一相一致，即：

$$x_{ij} = \begin{cases} 1, & \text{从第 } i \text{ 个路口走到第 } j \text{ 个路口} \\ 0, & \text{否则} \end{cases}$$

● 确定目标函数

在保留问题一对应的游客游玩长度的路径的最短、满意度最大的两个目标函数的前提下：

$$\text{MIN } z_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}}\right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

● 确定约束条件

进一步考虑到在实际的路线中，为了使游客尽可能的能够观赏到园方的表演，因此我们添加了以下约束条件，使得游客的游玩路线尽可能的经过园方的表演场地，即：

$$T = \sum_{i=1}^n \sum_{j=1}^n t_{ij} * x_{ij} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n x_{ij} (t_i + t_j), i = 1, 2 \dots n, j = 1, 2 \dots n$$

综上，我们得到了

$$\text{MIN } z_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij} * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}}\right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

s.t

$$T = \sum_{i=1}^n \sum_{j=1}^n t_{ij} * x_{ij} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n x_{ij} (t_i + t_j) \leq 540, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\Delta\tau = Q_{ij}(\alpha_j - \alpha_i) \leq 30, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\sum_{i=1}^n x_{ij} \leq 1, j = 1, 2 \dots n$$

$$\sum_{j=1}^n x_{ij} \leq 1, i = 1, 2 \dots n$$

$$x_{ij} = 0 \text{ 或 } 1$$

$$Q_{ij} = 0 \text{ 或 } 1$$

(说明: $\sum_{j=1}^n x_{ij} \leq 1, i = 1, 2 \dots n$ 表示从第 i 结点出发只能去往对应唯一

的 j 结点;当结点 i 到结点 j 中有表演活动时 $Q_{ij} = 1$ 否则为 0; $t_{ij} = \frac{d_{ij}}{v}$)

● 模型的求解方法

通过第一问对满意度矩阵求解得到的最短路线中的两种情况中, 加上动物表演时间进行规划, 线性统一求解目标函数, 利用蚁群算法将两中最短路径进行再优化可以得到基于表演活动时间及满意度的最短路线:

Shortroad is 1 2 3 6 10 15

图 7 基于表演活动时间及满意度的最短路线

其代表的含义为:

◇ V1 结点——> V2 结点——> V3 结点——> V6 结点——> V10 结点——> V15 结点(西门)

4.4 问题三的建立与求解

在游客游玩中运动聚集过程中的拥挤状况一般用拥挤度表示^[4]。为反映结点 i 到结点 j 路段的拥挤情况, 定义路段上的游客人数与路段人口容量之比为路段拥挤度, 且随路段上的游客人数实时变化, 路段游客人员数表达式为:

$$n_{ij}^t = \int_0^t (\lambda_{ij}^t - \theta_{ij}^t) dt, i = 1, 2 \dots n, j = 1, 2 \dots n$$

式中为 λ_{ij}^t 是 t 时刻路段 (i, j) 的人群输入率, θ_{ij}^t 为 t 时刻路段 (i, j) 的人群输出率。

则时刻路段 (i, j) 的拥挤度为

$$\omega_{ij}^t = \frac{n_{ij}^t}{c_{ij}}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

为防止路段拥堵, 当路段上的人员数达到容量 c_{ij} 时, 不允许人流继续输入,

即 $0 < \omega_{ij}^t \leq 1$, 而为了在道路不发生拥挤的情况下使游客得到更好体验, 则可以在第一问和第二问得出的结果中建立新的动态平衡关系:

● **约束条件**

每一时刻的*i*结点到*j*结点的游客数 都应小于该段路程中的最大容纳量

$$n_{ij}^t \leq \frac{S_{ij}}{S_{\text{总}}}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

要保持动态平衡需要限制拥挤度:

$$0 < \omega_{ij}^t \leq 1 \quad i = 1, 2 \dots n, j = 1, 2 \dots n$$

为了疫情防控, 整个园区内不能在同一个区域聚集过多人员, 所以探索路线的过程中不能仅仅根据最短路径进行游玩所以保留满意度而舍弃最短路径:

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}} \right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

综上, 我们得到了

$$\text{MIN } z_2 = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{p_{ij}} \right) * x_{ij}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$\text{s.t} \quad n_{ij}^t \leq \frac{S_{ij}}{S_{\text{总}}}, i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$0 < \omega_{ij}^t \leq 1 \quad i = 1, 2 \dots n, j = 1, 2 \dots n$$

$$x_{ij} = 0 \text{ 或 } 1$$

$$\sum_{i=1}^n x_{ij} \leq 1, j = 1, 2 \dots n$$

$$\sum_{j=1}^n x_{ij} \leq 1, i = 1, 2 \dots n$$

从问题本身来看, 我们找到了一条很符合的路径, 对于在路径上游玩的旅客, 在保证景区最大人口容纳量的同时, 也提供了一条很长的路线, 以保证旅客们不会在游玩过程中造成拥挤。下图是我们找到的最佳路径:

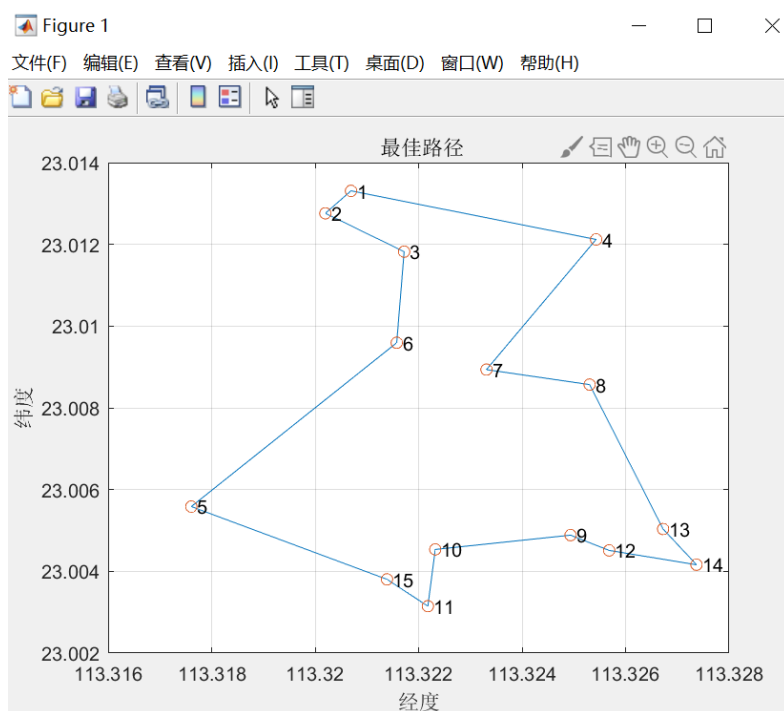


图 8 基于人员聚集的最佳路线

五、模型的评价与改进方向

5.1 模型的评价

5.1.1 模型优点:

首先保证了对数据的筛选，减少了无效数据量，将附件中园方的地图转换为几何模型，数据更加便于处理。在目标函数的设计上，不仅仅只考虑了路径最短问题，同时也综合了满意度的评判。模型采用了蚁群算法；蚁群算法具有较强的鲁棒性，所以适用广；并且是一种概率型的全局搜索算法。在蚁群算法中，蚂蚁位置的移动只依靠也是完全依靠两种因素，即路径的信息激素和路径长度，这种状态转移规则的移动规律可以提高算法的全局搜索能力，蚁群算法与其他算法结合起来相对容易便于改进和完善模型，在问题三中实现了景区游客的动态平衡，模型更具实用性。

5.1.2 模型不足

整体重要指标的权重的量化过程单一。多目标函数的处理过程单薄。

在保证充分了考虑题目要求的前提下，尽量使得模型能够更加具有普遍性、通用性。此外，多目标函数的权重处理也复合客观实际。但是在模型的构架中由于舍弃了时间这一维度，得到的结果在实际应用时并未能够提供更加确切个性化的帮助。

5.1.3 模型的改进

在进行模型的构建中，引入时间变量，罗列有限的园方表演安排从而得到更加精准的路线。在权重量化的过程中可以进行更深入的分析考虑，模型的求解选择可以结合多层贪心算法。在次基础上引入微分方程模型进行人口的流动性处理，同时预测人口的时间变化，结合动态的人口变化，从而得到更加准确的人口分配问题。

六. 参考文献

- [1]基于感知情境的北京不同类型旅游景区游客满意度研究[J]. 刘建国,张妍,黄杏灵.地理科学.2018(04):3-6.
- [2]基于改进蚁群算法的冰鲜水产品配送路径优化方法研究[J]. 田鸽,薛冬娟,梁斌,温凤仙,李鹏飞.大连海洋大学学报.2019(05):2-7.
- [3]改进蚁群算法求解最短路径问题[J] 张森.电子世界 2013(08):2-4.
- [4] 一种基于时空拥挤度的应急疏散路径优化方法[J].李清泉,李秋萍,方志祥.测绘学报, 2011,40(4) :3-8.
- [5] 基于多目标进化算法和遗传算法的最佳旅游套餐设计——以上海市为例[J].周辰红,李 中.同济大学数学系,200092,2014(03):1-4

附录:

问题一代码:

```
function shortroad_ant_main
% Ant main program
clear all;close all;clc;%清屏
tic;%计时开始
Ant=50;Ger=100;%运行参数初始化
power=[9999    193 9999    451 9999    9999    9999    9999
9999    9999    9999    9999    9999    9999;
9999    9999    75 9999    3651    9999    9999    9999
9999    9999    9999    9999    9999    9999;
9999    9999    9999    43 9999    244 9999    9999    9999    9999
9999    9999    9999    9999    9999;
```

```

9999 9999 9999 9999 9999 9999 332 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 3725 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 113 9999 9999
812 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 197 291 9999
9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 266
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
316 9999 133 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 316
9999 187 9999 9999 9999 341;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 182 174 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 182 9999 121 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999]

```

```

[PM PN]=size(power);
% 初始化蚂蚁位置
v=init_population(Ant,PN);
v(:,1)=1;v(:,PN)=1;% 始点和终点纳入路径
% 把距离当信息素浓度
fit=short_road_fun(v,power);
% 距离越小越好，所以要和信息素浓度相对应。
T0 = max(fit)-fit;
% 画出图形
figure(1);grid on;hold on;
plot(fit,'k*');
title('(a)蚂蚁的初始位置');
xlabel('x');ylabel('f(x)');
% 初始化
vmfit=[];vx=[];
P0=0.2; % P0----全局转移选择因子
P=0.8; % P ----信息素蒸发系数
%C=[];
% 开始寻优
for i_ger=1:Ger

```

```

lamda=1/i_ger;    % 转移步长参数
[T_Best(i_ger),BestIndex]=max(T0);%最大信息素浓度
for j_g=1:Ant    % 求取全局转移概率
    r=T0(BestIndex)-T0(j_g);%与最佳的蚂蚁的距离
    Prob(i_ger,j_g)=r/T0(BestIndex);%应该以多大的速率向它靠拢
end
for j_g_tr=1:Ant
    if Prob(i_ger,j_g_tr)<P0
        %局部转移----小动作转移
        M=rand(1,PN)<lamda;
        temp=v(j_g_tr,:)-2.*(v(j_g_tr,:).*M)+M;
    else
        %全局转移----大步伐转移
        M=rand(1,PN)<P0;
        temp=v(j_g_tr,:)-2.*(v(j_g_tr,:).*M)+M;
    end
    %始点和终点重新加入。即不能在移动过程中发生改变。
    temp(:,1)=1;temp(:,end)=1;
    if short_road_fun(temp,power)<short_road_fun(v(j_g_tr,:),power)
        %记录
        v(j_g_tr,:)=temp;
    end
end
%信息素更新 ， 准备下一次迭代
fit=short_road_fun(v,power);
T0 = (1-P)*T0+(max(fit)-fit);%信息素蒸发
[sol,indb]=min(fit);
v(1,:)=v(indb,:);%记录本次迭代的状态
media=mean(fit);
vx=[vx sol];
vmfit=[vmfit media];
end
%%%% 最后结果
disp(sprintf('\n'));    %空一行
% 显示最优解及最优值
disp(sprintf('Shortroad is %s',num2str(find(v(indb,:)))));%num2str 数据转
换成字符。
disp(sprintf('Mininum is %d',sol));
v(indb,:)
% 图形显示最优结果
figure(2);grid on;hold on;
plot(fit,'r*');
title('蚂蚁的最终位置');
xlabel('x');

```

```

ylabel('f(x)');
% 图形显示最优及平均函数值变化趋势
figure(3);
plot(vx);
title('最优,平均函数值变化趋势');
xlabel('Generations');ylabel('f(x)');
hold on;plot(vmfit,'r');hold off;
runtime=toc%时间结束
end
%%
function fit=short_road_fun(v,power)
[vm vn]=size(v);
fit=zeros(vm,1);%记录每一条路径的距离
for i=1:vm
    I=find(v(i,:)==1);%寻找在路径上的点
    [Im,In]=size(I);
    for j=1:In-1
        fit(i)=fit(i)+power(I(j),I(j+1));%求路径的距离
    end
end
end
end
%%
%Function init_population
function v=init_population(n1,s1)
v=round(rand(n1,s1));%初始化所有的蚂蚁
end

```

问题二代码：

```

function shortroad_ant_main
% Ant main program
clear all;close all;clc;%清屏
tic;%计时开始
Ant=50;Ger=100;%运行参数初始化
power=[9999    193/3    9999    451    9999 9999    9999    9999
9999    9999    9999    9999    9999    9999    9999;
9999    9999    75/2        9999        3651/39 9999    9999    9999
9999    9999    9999    9999    9999    9999    9999;
9999    9999    9999    43/3        9999 244/10 9999    9999    9999
9999    9999    9999    9999    9999    9999;
9999    9999    9999    9999    9999    9999    332/6 9999    9999
9999    9999    9999    9999    9999    9999;
9999    9999    3725/41 9999    9999    9999    9999    9999    9999
9999    9999    9999    9999    9999    9999;
9999    9999    9999    9999    9999    9999    113/8 9999    9999
812/21 9999    9999    9999    9999    9999;

```

```

9999 9999 9999 9999 9999 9999 9999 197/7 291/10
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 266/9
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
316/5 9999 133/6 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 316/5
9999 187/3 9999 9999 9999 341/4;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 182/7 174/9 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 182/7 9999 121/2 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999;
9999 9999 9999 9999 9999 9999 9999 9999 9999
9999 9999 9999 9999 9999 9999]

```

```

[PM PN]=size(power);
% 初始化蚂蚁位置
v=init_population(Ant,PN);
v(:,1)=1;v(:,PN)=1;%始点和终点纳入路径
%把距离当信息素浓度
fit=short_road_fun(v,power);
%距离越小越好，所以要和信息素浓度相对应。
T0 = max(fit)-fit;
% 画出图形
figure(1);grid on;hold on;
plot(fit,'k*');
title('(a)蚂蚁的初始位置');
xlabel('x');ylabel('f(x)');
% 初始化
vmfit=[];vx=[];
P0=0.2; % P0----全局转移选择因子
P=0.8; % P ----信息素蒸发系数
%C=[];
% 开始寻优
for i_ger=1:Ger
    lamda=1/i_ger; % 转移步长参数
    [T_Best(i_ger),BestIndex]=max(T0);%最大信息素浓度
    for j_g=1:Ant % 求取全局转移概率
        r=T0(BestIndex)-T0(j_g);%与最佳的蚂蚁的距离
        Prob(i_ger,j_g)=r/T0(BestIndex);%应该以多大的速率向它靠拢
    end
end

```



```

title('最优,平均函数值变化趋势');
xlabel('迭代次数');ylabel('f(x)');
hold on;plot(vmfit,'r');hold off;
runtime=toc%时间结束
end
%%
function fit=short_road_fun(v,power)
[vm vn]=size(v);
fit=zeros(vm,1);%记录每一条路径的距离
for i=1:vm
    I=find(v(i,:)==1);%寻找在路径上的点
    [Im,In]=size(I);
    for j=1:In-1
        fit(i)=fit(i)+power(I(j),I(j+1));%求路径的距离
    end
end
end
%%
%Function init_population
function v=init_population(n1,s1)
v=round(rand(n1,s1));%初始化所有的蚂蚁
end

```

问题三代码：

%% I. 清空环境变量

clear all

clc

%% II. 导入数据

citys=[113.320693 23.013309

113.320198 23.01276

113.321721 23.011821

113.32543 23.01212

113.317607 23.005584

113.321577 23.009592

113.323311 23.008935

113.325305 23.008569

113.324937 23.004885

113.322322 23.004536

113.322179 23.003147

113.325683 23.004511

113.326725 23.005035

113.327371 23.004161

```

113.321389 23.003804
];%景点经纬度坐标信息
%% III. 计算城市间相互距离
n = size(citys,1);
D = zeros(n,n);
for i = 1:n
    for j = 1:n
        if i ~= j
            D(i,j) = sqrt(sum((citys(i,:) - citys(j,:)).^2));
        else
            D(i,j) = 1e-4; %如果是 0 会导致矩阵对角线都是 0 导致启发函数无穷大 因此取个很小的值
        end
    end
end

%% IV. 初始化参数
m = 50; % 蚂蚁数量
alpha = 1; % 信息素重要程度因子
beta = 5; % 启发函数重要程度因子
rho = 0.1; % 信息素挥发因子
Q = 1; % 常系数
Eta = 1./D; % 启发函数
Tau = ones(n,n); % 信息素矩阵
Table = zeros(m,n); % 路径记录表,每一行代表一个蚂蚁走过的路径
iter = 1; % 迭代次数初值
iter_max = 200; % 最大迭代次数
Route_best = zeros(iter_max,n); % 各代最佳路径
Length_best = zeros(iter_max,1); % 各代最佳路径的长度
Length_ave = zeros(iter_max,1); % 各代路径的平均长度

%% V. 迭代寻找最佳路径
while iter <= iter_max
    % 随机产生各个蚂蚁的起点城市
    start = zeros(m,1);
    for i = 1:m
        temp = randperm(n);
        start(i) = temp(1);
    end
    Table(:,1) = start;
    citys_index = 1:n;
    % 逐个蚂蚁路径选择
    for i = 1:m
        % 逐个城市路径选择

```



```

for j = 2:n
    tabu = Table(i,1:(j - 1));           % 已访问的城市集合(禁忌表)
    allow_index = ~ismember(citys_index,tabu);
    allow = citys_index(allow_index);    % 待访问的城市集合
    P = allow;
    % 计算城市间转移概率
    for k = 1:length(allow)
        P(k) = Tau(tabu(end),allow(k))^alpha * Eta(tabu(end),allow(k))^beta;
    end
    P = P/sum(P);
    % 轮盘赌法选择下一个访问城市
    Pc = cumsum(P);
    target_index = find(Pc >= rand);
    target = allow(target_index(1));
    Table(i,j) = target;
end
end
% 计算各个蚂蚁的路径距离
Length = zeros(m,1);
for i = 1:m
    Route = Table(i,:);
    for j = 1:(n - 1)
        Length(i) = Length(i) + D(Route(j),Route(j + 1));
    end
    Length(i) = Length(i) + D(Route(n),Route(1));
end
% 计算最短路径距离及平均距离
if iter == 1
    [min_Length,min_index] = min(Length);
    Length_best(iter) = min_Length;
    Length_ave(iter) = mean(Length);
    Route_best(iter,:) = Table(min_index,:);
else
    [min_Length,min_index] = min(Length);
    Length_best(iter) = min(Length_best(iter - 1),min_Length);
    Length_ave(iter) = mean(Length);
    if Length_best(iter) == min_Length
        Route_best(iter,:) = Table(min_index,:);
    else
        Route_best(iter,:) = Route_best((iter-1),:);
    end
end
end
% 更新信息素
Delta_Tau = zeros(n,n);

```

```

% 逐个蚂蚁计算
for i = 1:m
    % 逐个城市计算
    for j = 1:(n - 1)
        Delta_Tau(Table(i,j),Table(i,j+1)) = Delta_Tau(Table(i,j),Table(i,j+1)) +
Q/Length(i);
    end
    Delta_Tau(Table(i,n),Table(i,1)) = Delta_Tau(Table(i,n),Table(i,1)) + Q/Length(i);
end
Tau = (1-rho) * Tau + Delta_Tau;
% 迭代次数加 1，清空路径记录表
iter = iter + 1;
Table = zeros(m,n);
end

%% VI. 结果显示
[Shortest_Length,index] = min(Length_best);
Shortest_Route = Route_best(index,:);
disp(['最短距离:' num2str(Shortest_Length)]);
disp(['最短路径:' num2str([Shortest_Route Shortest_Route(1)])]);

%% VII. 绘图
figure(1)
plot([citys(Shortest_Route,1);citys(Shortest_Route(1),1)],...
[citys(Shortest_Route,2);citys(Shortest_Route(1),2)],'o-');
grid on
for i = 1:size(citys,1)
    text(citys(i,1),citys(i,2),[' ' num2str(i)]);
end
%text(citys(Shortest_Route(1),1),citys(Shortest_Route(1),2),'      起点');
%text(citys(Shortest_Route(end),1),citys(Shortest_Route(end),2),'      终点');
xlabel('城市位置横坐标')
ylabel('城市位置纵坐标')
title(['蚁群算法优化路径(最短距离:' num2str(Shortest_Length) ')'])

```