# 概述

# 设计

1. 参数设计/约束条件

一般对于传统的路径规划算法，参数设计只考虑空间距离，但是这里需要引入多约束条件，因此参数设计主要包括：

1. 空间距离
2. 天气状况
3. 交通状况
4. 经济约束

在上述参数设计的时候，最终是需要给出一个约束函数。

在进行距离计算的时候，一般可以采用如下的几种方法：

1. 欧氏距离：计算平面距离（假设平面，忽略地球曲率，仅适用于小范围区域）
2. 曼哈顿距离：计算网格布局距离（用于网格路网，不适用于经纬度）

3）Haversine距离：最常用的 GPS 坐标距离计算方法，考虑地球曲率（强烈推荐）

4）闵可夫斯基距离：p 值调节为 1 或 2 时即为曼哈顿或欧氏距离，这样可以支持不同的景点之间采用不同的距离计算，比如两个景点都是平地且很近直接采用欧氏距离，如果一个是假山或者地势较高的凉亭，则可以采用曼哈顿距离

参考文献：

[基于蚁群算法的多目标最优旅游线路规划设计](https://www.zgstly.net/CN/10.12342/zgstly.20220059)

[旅游者时间约束和空间行为特征的景区旅游线路设计方法研究](https://d.wanfangdata.com.cn/periodical/ChlQZXJpb2RpY2FsQ0hJTmV3UzIwMjQwNzA0Eg1seXhrMjAxNjA5MDE2Ggh0cHI1c21rMQ%253D%253D)

2、理论模型

在参数设计的时候，需要调查具体参数设计的模型，这里的模型选择主要包括：

1. 选择具体的模型获取用户/游客的倾向度
2. 选择交通、气象等模型预测相关的非人意志的影响因子

参考文献：

[基于视野范围和遗传算法的三维地形路径规划](https://d.wanfangdata.com.cn/periodical/jsjgcyyy202115034)

3、路径规划算法选择

在进行路径规划算法选择的时候，需要考虑如下的一些因素：

1. 节点个数（如果比较少可以直接穷举、动态规划）
2. 是否走遍所有的节点
3. 是否需要回到起点





这里路径规划算法只是一个手段，最终呈现的目标，也就是目标函数，是游客用户体验系数（即保证用户体验系数最大）。

4、多约束条件

如果是简单的路径规划，则不具备灵活性，这里采用多约束条件的规划算法，即可以根据用户的不同诉求动态选择路线：

1. 用户对于景点个数有限制
2. 用户有不希望游玩的景点
3. 用户不希望旅游有门票的景区或者对于总门票金额有限制（即动态经济约束）

参考文献：

[基于蚁群算法的多约束动态旅游路线规划](https://www.hanspub.org/journal/paperinformation?PaperID=64442&)

[基于蚁群算法的多目标最优旅游线路规划设计](https://www.zgstly.net/CN/10.12342/zgstly.20220059)

# 实现

1. 设计模型的影响因素

主要包括：

1. 景区内景点的距离：寻找一个模型获取不同景点之间的精确距离，而非直线距离，比如使用GIS工具等
2. 景点之间的拥挤程度：寻找一个模型获取景点的拥挤程度，比如调查问卷、历史数据等
3. 游客的体验程度：寻找一个模型获取游客的旅游倾向
4. 算法实现
5. 根据设计模型，定义目标函数
6. 改进算法的流程图
7. 算法验证

* 有效性验证

在程序执行过程中，目标函数是收敛的，这样说明是有效的。

* 鲁棒性

1. 起点出发后又重新回到起点的闭环
2. 起点出发后指定到某个终点
3. 起点出发排除某些节点后回到起点（可选）
4. 起点出发排除某些节点后达到某个终点（可选）

这里的验证指标为，路径长度（不一定是最短，尽可能短）、拥挤程度（尽可能不拥堵）、用户体验（体验度越高越好）

* 高效性

优化算法的响应时间对比/效率

# 创新点

1. 融合用户旅游动机等理论进行路径规划（用于满意度等）
2. 可以根据用户的不同诉求规划不同路线：比如有的用户只希望旅游最多10个景点、用户不想看某几个景点，即基于多约束的动态旅游路线规划、用户可以选择回到原点和不回到原点（这里就设置起始点一样、指定终点两种情况，都遍历所有的节点）
3. 融合A\*和遗传算法，适用多种不同情况（这个可以作为对比）