

在解决这个问题的过程中，我尝试了两类算法：

一、遗传算法

我设计的这个 TSP 遗传算法慢而且不好用，所以没有采用遗传算法。

遗传算法设计如下：

1、初始化序列（有两种初始化方法：1、随机初始化；2、随机一个点然后贪心地选择最近的未访问点进行初始化）

2、用锦标赛方法筛选出两组亲代

3、亲代进行交叉，交叉的方法是第一组亲代中保留随机长度随机位置的片段，然后在其他位置填入第二组亲代，如果有冲突节点的话以第一组的节点优先。

4、对交叉的结果进行变异，变异方法是随机交换随机数量的节点的位置。

如果随机初始化的话效果很差，但是如果稍微加一些算法初始化（比如贪心地找最近的点）的话，遗传算法就在初始化出来的点再也不能改进一步了。

二、A*搜索

对于每一个节点来说，全图中剩下的所有节点都是可达的，但是对每一个节点都遍历所有剩下的未访问节点在计算复杂度上不可行。所以，在 A*搜索中我引入了一个参数，控制一个节点的孩子节点数量。对于一个节点来说，最近的前 n 个未访问节点才是可达的。基于这个设定，我预设了两种 $h(n)$ ：第一种 $h(n)$ 涉及三个结点：要计算 $h(n)$ 的结点，记为当前节点；搜索路径的初始节点，最后需要回到这个节点；距离当前节点最远的未访问节点，记为最远未访问节点。 $h(n)$ 是当前节点到最远未访问节点的距离加上最远未访问节点到初始节点的距离。第二种 $h(n)$ 的计算方式为：当前节点到所有未访问节点的距离的均值。

经过实验验证，第二种 $h(n)$ 性能比第一种 $h(n)$ 好，所以采取第二种 $h(n)$ 。对应的访问序列、结果图片、路径长度已经在文件中提交，路径长度为 6313938.200223035。

在提交的文件中只放了最好的结果，其他结果和过程日志可以在 [github](https://github.com/Cangshanqingshi/MonaLisa-TSP) 链接：<https://github.com/Cangshanqingshi/MonaLisa-TSP> 里查看。

三、后处理

对于已经有的结果，首先计算所有线段长度的均值和方差，长度在三倍方差之外的线段的端点视为异常点。对于所有的异常点，均尝试与自身周围一定阈值内的点交换顺序，如果可以让长度变短就保留，从而优化路径。

优化后的路径长度为 6303257.088980756，图片已在文件中提交，相关内容也已经在 [github](https://github.com) 上进行同步。