

# 算法分析与理论-小组作业

## 旅行商问题 (TSP)

假设有一个旅行商需要访问若干个城市，并且每个城市之间有不同的距离，旅行商需要找到一条最短路径来访问每个城市一次，并最终返回起点。

## 蚁群优化算法 (Ant Colony Optimization)

蚁群优化算法模拟蚂蚁的行为，通过信息素来指引搜索路径，以找到最短路径。

该算法模拟了自然界中蚂蚁觅食的行为，蚂蚁在寻找食物源时，会在其经过的路径上释放一种信息素，并能够感知其他蚂蚁释放的信息素。

信息素浓度的大小表征路径的远近，浓度越高，表示对应的路径距离越短。

**正反馈**：蚂蚁以较大概率优先选择信息素浓度较高的路径，并释放一定量的信息素，以增强该条路径的信息素浓度。

## 随机重启爬山算法 (Hill Climbing)

**爬山算法**一种简单的贪心搜索算法，该算法每次从当前解的临近解空间中选择一个最优解作为当前解，直到达到一个局部最优解。

**缺点**：容易陷入局部最优解

**随机重启爬山算法**是一种改进的爬山算法，旨在解决爬山算法容易陷入局部最优解的问题。其核心思想是：如果当前的搜索过程没有找到全局最优解，那么就**重新随机生成一个初始状态**（随机重启），继续进行爬山搜索，直到找到全局最优解。

## 模拟退火算法 (Simulated Annealing)

其实也是一种贪心算法，但是它在搜索过程引入了**随机因素**。

模拟退火算法以一定的概率来接受一个比当前解要差的解，因此有可能会跳出这个局部的最优解，达到全局的最优解。

**随机因素**：若移动后得到更优解，则总是接收移动；若移动后的解比当前解要差，则以**一定的概率**接受移动，而且这个概率随着时间推移逐渐降低（逐渐降低才能趋向稳定）

“一定的概率”的计算参考了金属冶炼的退火过程，因此叫做模拟退火算法。

## FAQ

Q：我看到你们的输入用例中的图是从**稀疏到稠密**的，请问这个稀疏和稠密是怎么定义的？

A：最稀疏的图即刚好连通的一个**单环**，最稠密的图是**完全图**。前者有  $n$  条边，后者有  $n(n-1)/2$  条边，比如稠密度 25% 指的是图中有  $n + (n(n-1)/2 - n) * 0.25$  条边。

Q：按照你的定义，非 100% 稠密的图不是完全图，算法中对于**非完全图**是怎么去处理那些“不存在”的边的？

A: 对于不存在的边, 我们将这些边的边权设置为了一个很大的值, 这样一来算法在迭代过程中几乎就不会去选择这些边。实验证明我们这样做确实是对的。