



# 启发式算法原理与应用

数模加油站 2024美赛保奖大班课

 主讲：张老师



# 目录

## CONTNETS

01

模拟退火

02

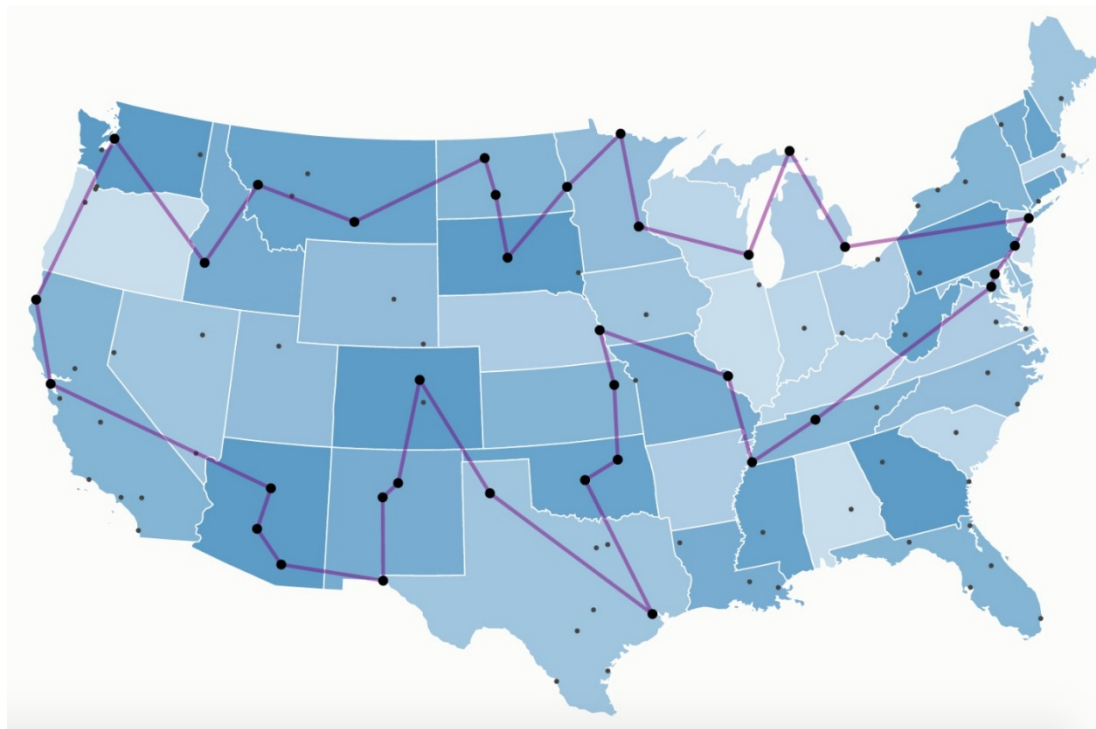
遗传算法

03

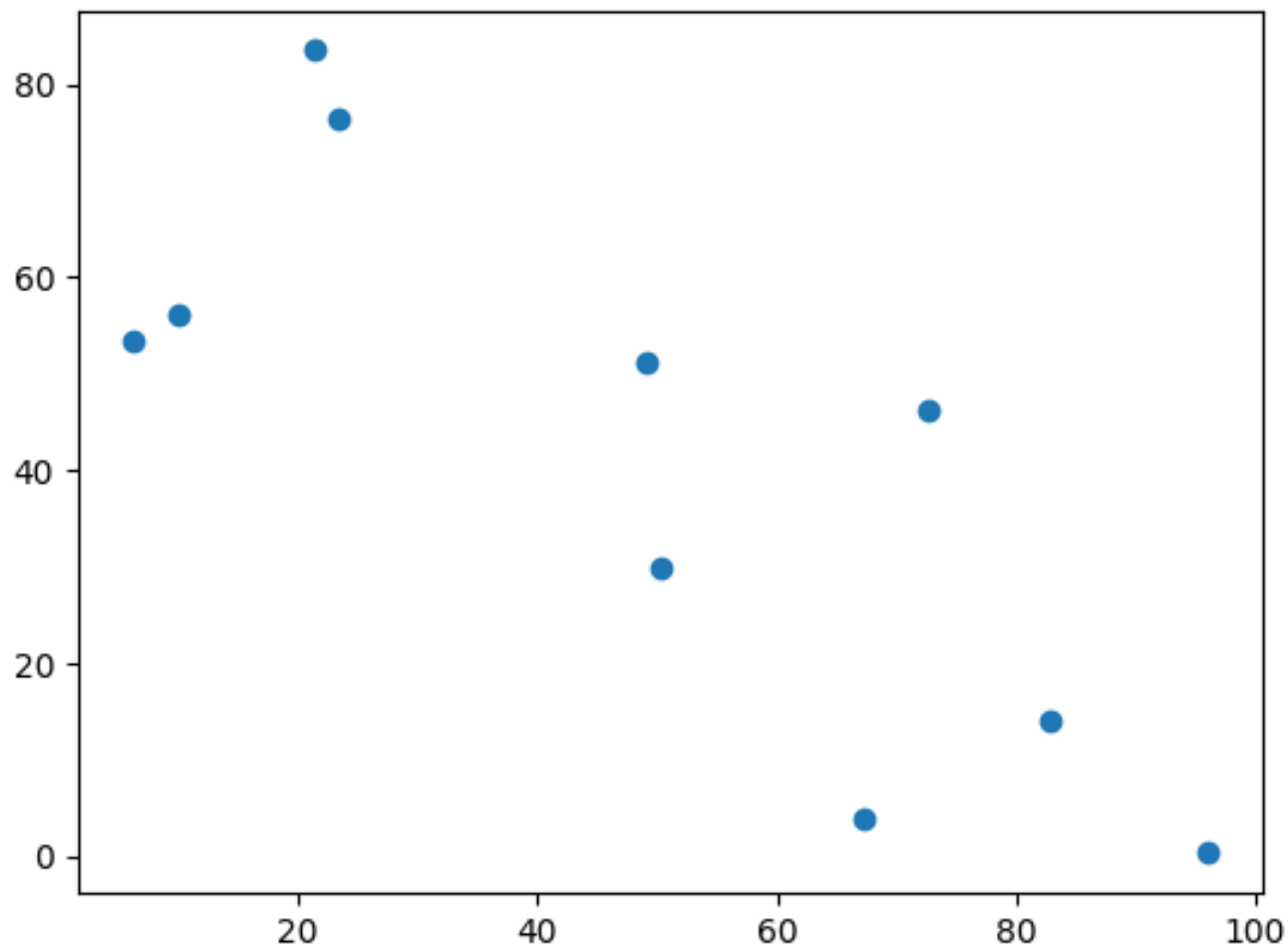
蚁群算法

# 旅行商问题 (Travelling salesman problem)

假设有一个旅行商人要拜访 $N$ 个城市，每个城市只能拜访一次，最后回到原来出发的城市，求最短的路径。



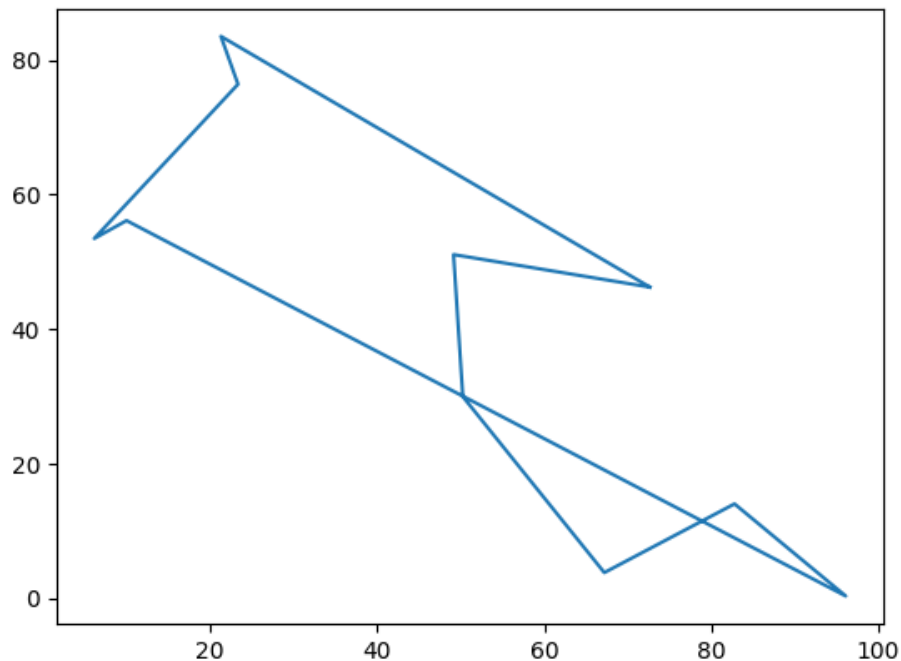
本次课程以下面的点为例



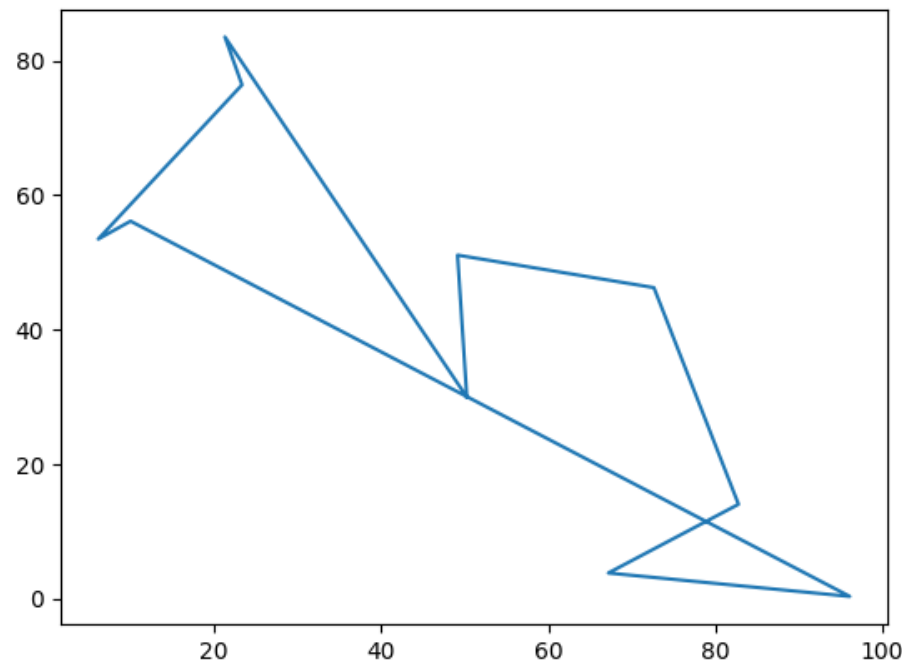
思路1: 贪心算法

从任意一个点出发, 每次选择离自己最近的点, 最后回到初始位置

这种方法不能保证最优解, 而且初始点会影响结果



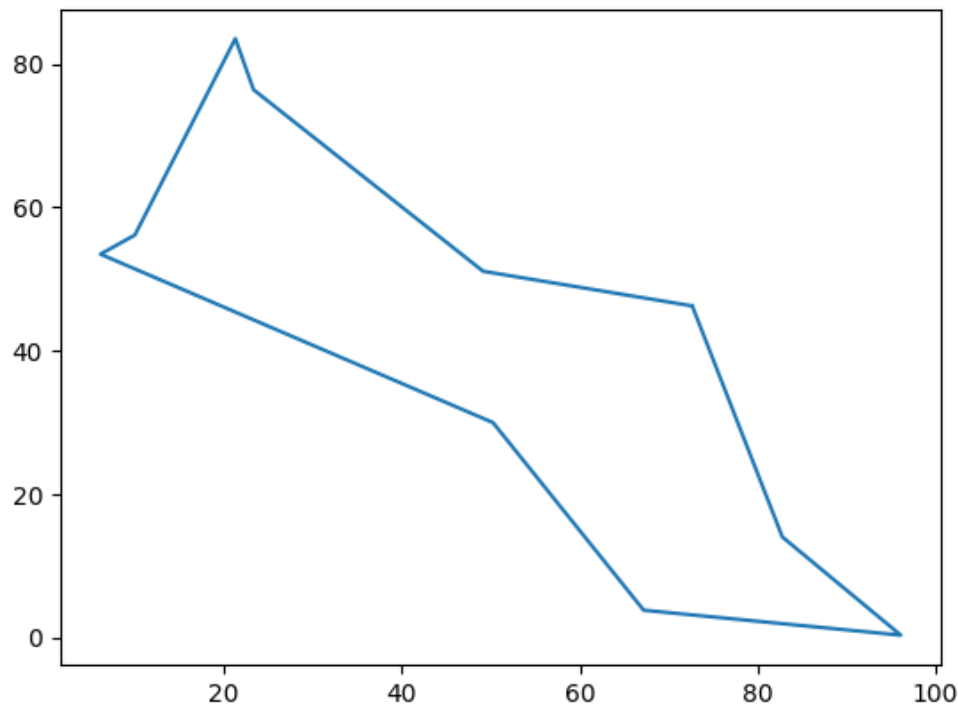
route=320.93



route=330.93

旅行商问题属于NP难问题（NP-hard），要想得到最优解，只能把所有可能的情况全部枚举出来，算出路径长度再做比较。

如果有 $n$ 个城市，则路径可以表达为 $1 \sim n-1$ 的一个全排列，共 $(n-1)!$ 种可能的路径。能保证最优解，但当 $n$ 变大时，很快就会突破计算机的运算限制。





## 旅行商问题

启发式算法是基于直观或经验构造的算法，在可接受的计算时间和空间条件下，给出待解决优化问题的一个可行解。

Remember：启发式算法并不保证找到最优解，只是在有限资源下找到还不错的解  
经典的启发式算法包括模拟退火、遗传算法、蚁群算法、神经网络等。

启发式算法的共同的目标：求NP-hard组合优化问题的全局最优解，NP-hard问题的限制它们只能以启发式的算法去求解实际问题。

常见的启发式算法都有其实际背景，但启发式算法的目标并非尽可能贴近其来源，而是高效解决要解决的问题。

其他的组合优化问题：

- 任务分配问题：  $m$ 个人完成  $n$ 项工作，每人做各项工作的效率不同，问如何分配能使工作最快完成。
- 装箱问题：把一定数量的物品放入容量相同的一些箱子中，使得每个箱子中的物品大小之和不超过箱子容量，并使所用的箱子数目最少。
- 背包问题：从一定数量的物品中选择若干装入背包，在不超过背包容量的前提下，使装入背包物品的价值最大。



# 模拟退火算法

---

01

PART ONE



# 模拟退火算法 (Simulated Annealing Algorithm)

模拟退火算法得益于材料统计力学的研究成果。

统计力学表明材料中粒子的不同结构对应于粒子的不同能量水平。在高温条件下，粒子的能量较高，可以自由运动和重新排列。在低温条件下，粒子能量较低。

如果从高温开始，非常缓慢地降温（这个过程被称为退火），粒子就可以在每个温度下达到热平衡。当系统完全被冷却时，最终形成处于低能状态的晶体。

模拟退火算法的思想是：为了不被局部最优解困住，需要以一定概率跳出当前位置，暂时接受一个不太好的解。在搜索最优解的过程中逐渐降温，初期跳出去的概率比较大，进行广泛搜索；后期跳出去的概率比较小，尽量收敛到较优解。

模拟退火算法的步骤如下：

- 随机生成一个初始解 $x$ ，设定一个初始温度 $T$
- 在上一次解的基础上进行调整，生成新的解 $x'$ ，并对比旧解 $f(x')$ 和新解 $f(x)$ 
  - 如果新解更好了，那就接受
  - 如果新解更差了，那就以 $e^{-\frac{f(x')-f(x)}{T}}$ 的概率接受
- 降温并重复上述步骤，直至迭代一定的次数

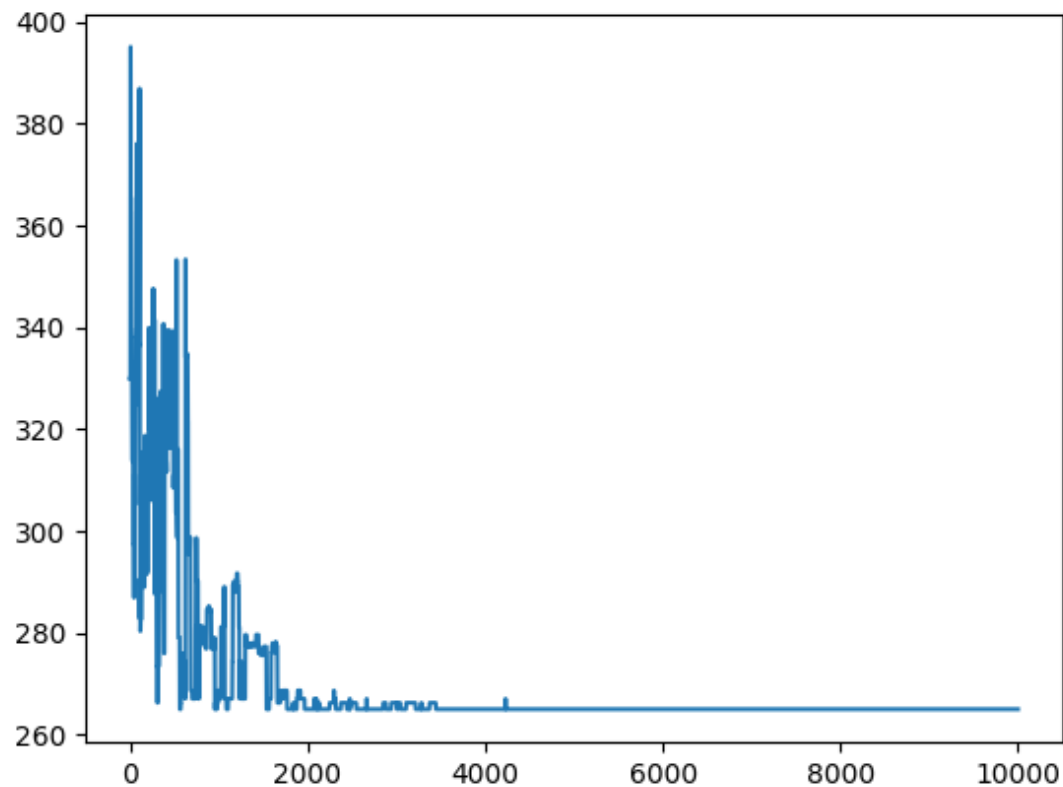
- 解空间
  - 解空间就是所有valid解的集合，每次生成的解必须valid
  - 旅行商问题的解空间就是1~n的全排列
- 初始解
  - 初始解可以随意选取，但比较好的初始解可以帮助算法尽快收敛
  - 可先随机生成少量几个序列，选一个比较小的；或者采用贪心算法选择初始解
- 目标函数
  - 目标函数根据我们要优化的目标确定，旅行商问题的目标函数就是路径长度

- 降温方法
  - 初始温度、降温方法的选择比较tricky，没有固定标准，只能试
  - 例如初始温度为1，每次降温就乘以99%，降到 $10^{-30}$ 就停止
  - 需要根据具体问题的数值、能接受的迭代次数等确定
  - 注：也不需要每次迭代都降温，也可以迭代几次才降一次温
- 温度和接受概率的关系
  - 根据模拟退火算法，如果新解更差了，那就以 $e^{-\frac{\Delta f(x)}{T}}$ 的概率接受
  - 可见，如果温度较高、新解变差的程度不大，则接受的概率较大

## 如何生成新的解

- 新解的生成是最体现creativity的地方，可以自己提出如何将当前解变换为新解，而且保证新解是valid，并且和旧解的差异不要太大
- 例如：
  - 任意选择两个标号，调换位置
  - 任意选择两个标号，颠倒它们中间的序列
  - 任意选择三个标号，将前两个标号中间的序列放到第三个标号之后
  - .....

## 运行实例



# 遗传算法

---

02

PART TWO

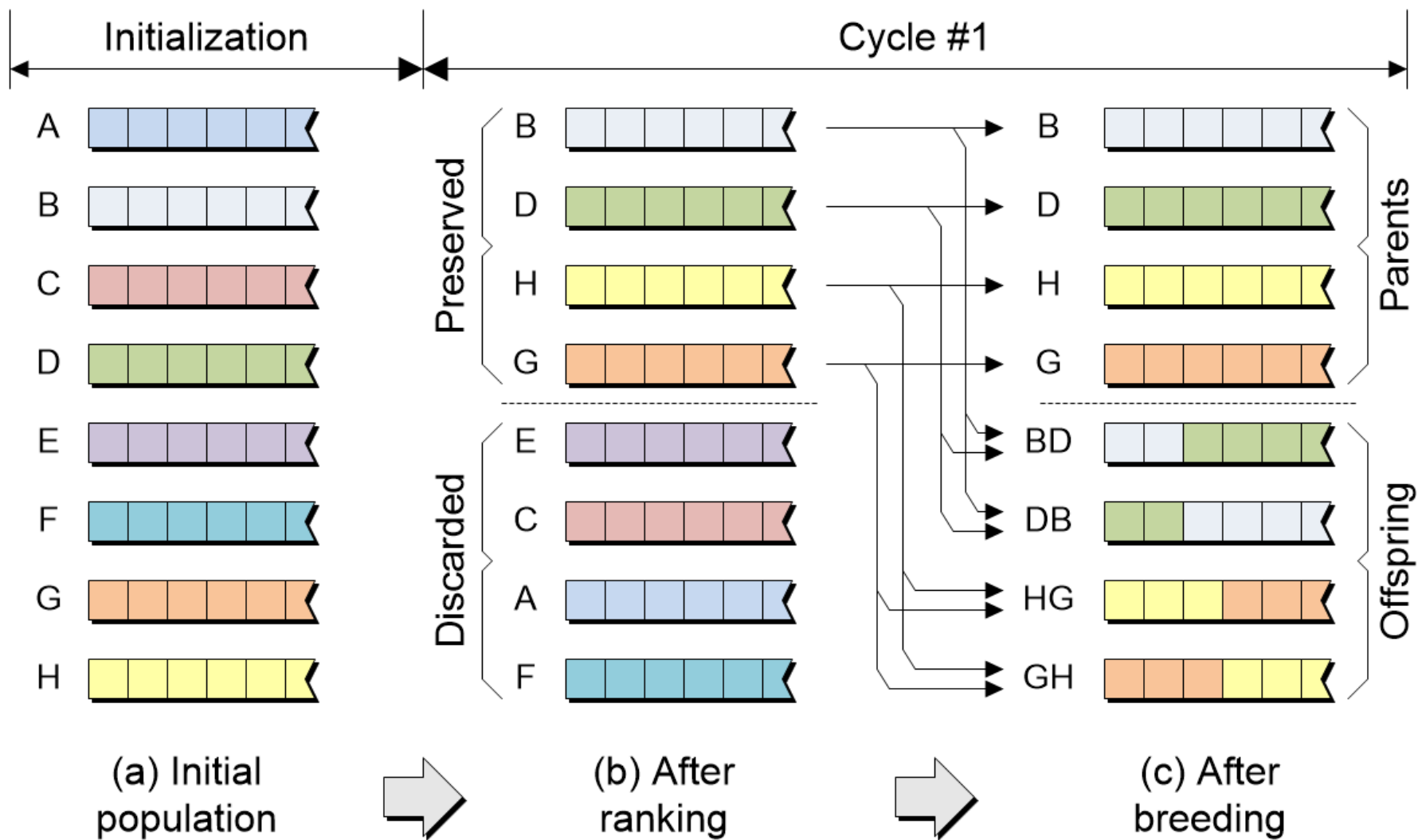




遗传算法是一种基于自然选择原理和自然遗传机制的搜索（寻优）算法，它是模拟自然界中的生命进化机制，在人工系统中实现特定目标的优化

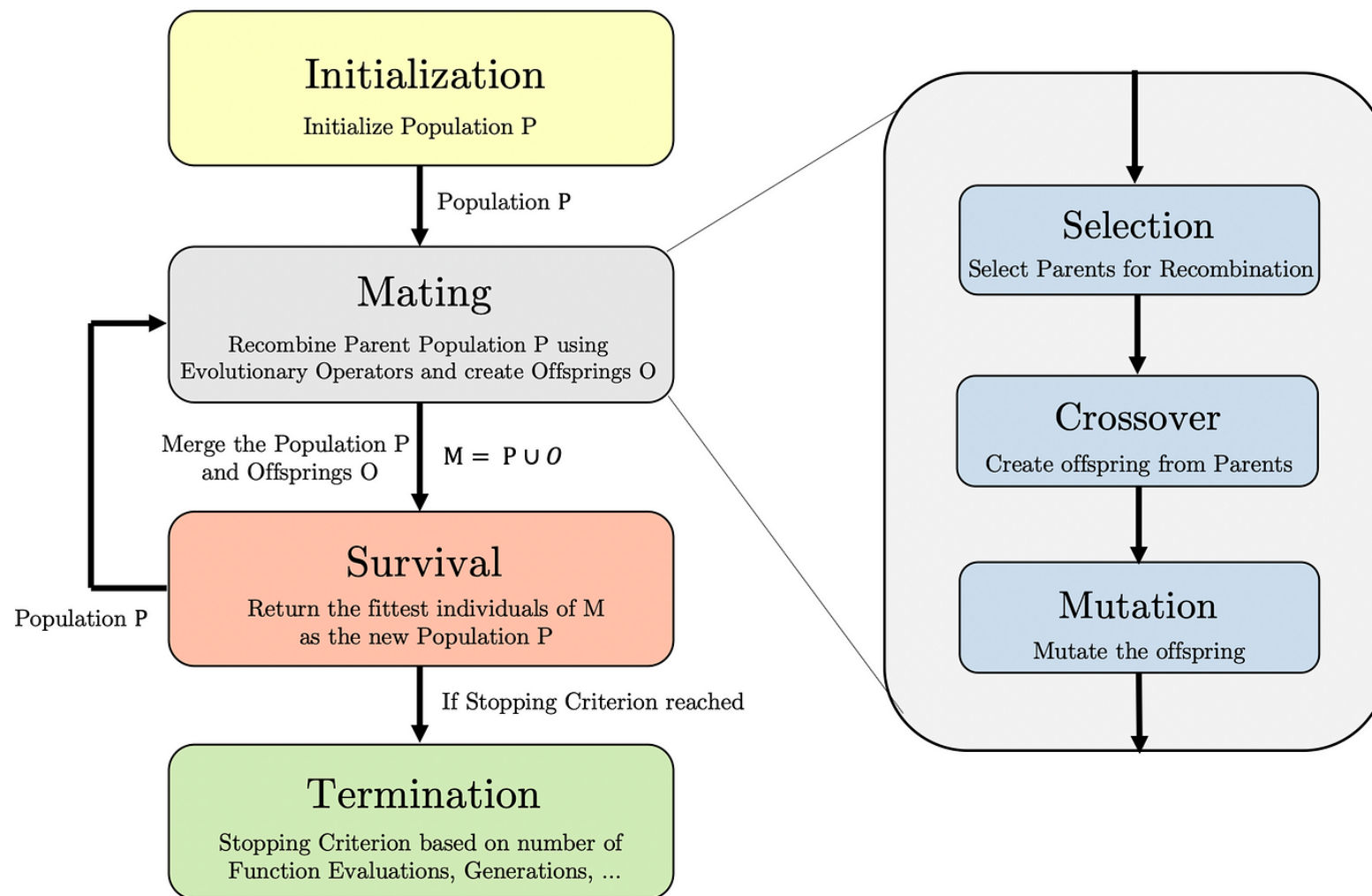
突变和基因重组是进化的原因，遗传算法是通过群体搜索技术，根据适者生存的原则逐代进化，最终得到准最优解

操作包括：初始群体的产生、求每一个体的适应度、根据适者生存的原则选择优良个体、被选出的优良个体两两配对，通过随机交叉其染色体的基因并随机变异某些染色体的基因生成下一代群体，按此方法使群体逐代进化，直到满足进化终止条件



遗传算法的步骤如下：

- 产生M个初始解，构成初始种群
- 每对父母以一定概率生成一个新解（交配产生后代）
- 每个个体以一定概率发生突变（即将自己的解变换产生新解）
- 父代和子代合在一起，留下M个最好的个体进入下一轮，其余淘汰（进行自然选择）
- 重复以上迭代，最后输出最好的个体



### 如何选择参数

- 遗传算法中要选择的参数很多：种群数量 $M$ 、变异概率、生成子代的数量、迭代次数
- 但很遗憾，参数的选择没有固定标准，只能自己试
- 种群数量 $M$ 越大、迭代次数越多、生成的子代越多，当然更有希望找到最优解，但相应的计算资源消耗也会增大，只能在可接受范围内进行选择

### 如何选择初始种群

- 其实可以随便初始化，但是较好的初始种群可以帮助更快收敛
- 例如随机生成若干个选最好的、贪心算法等

### 如何交配产生子代

- 交配方法是最能体现creativity的地方，应该尽量继承父代，但也要进行足够的调整

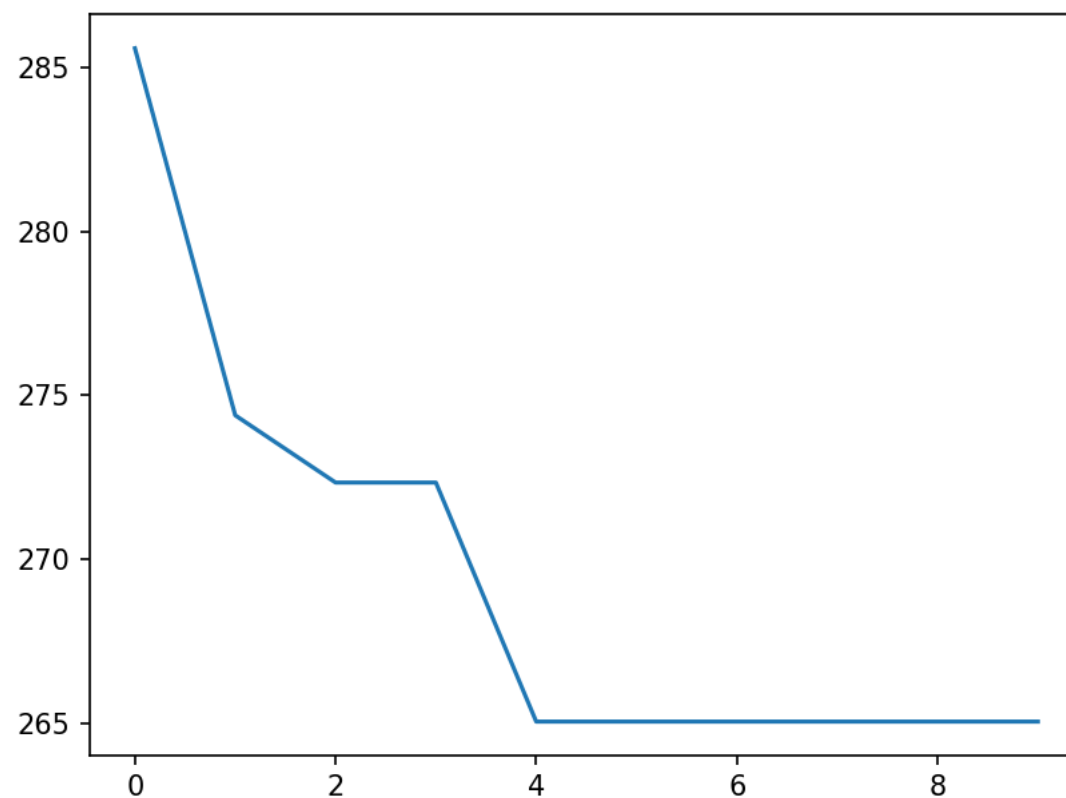
例如：

- 选择父亲的一个标号 $t$ ，在母亲那里找到它后面的全部数字，并依序取出
- 把父亲标号 $t$ 后面的部分接到母亲后面
- 把母亲取出来的数字接到父亲后面

### 如何突变产生新的解

- 突变就是根据自己的解生成一个新的解，方法可以和模拟退火中的方法相同

## 运行实例



# 蚁群算法

---

03

PART THREE

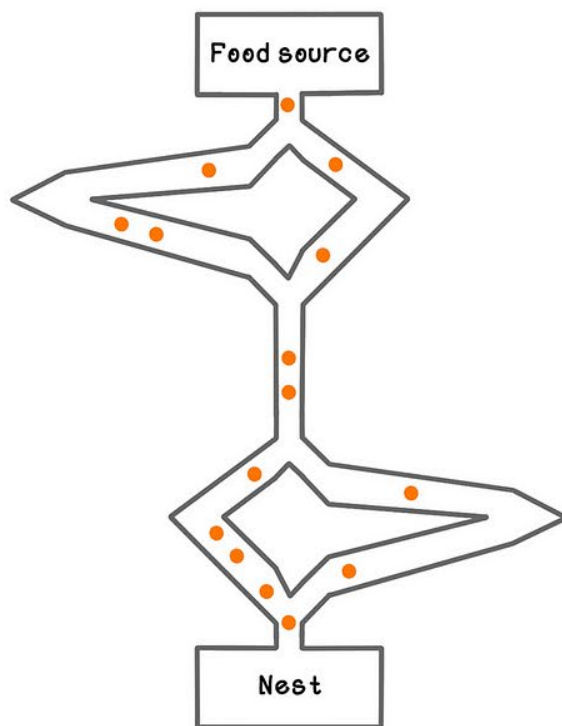




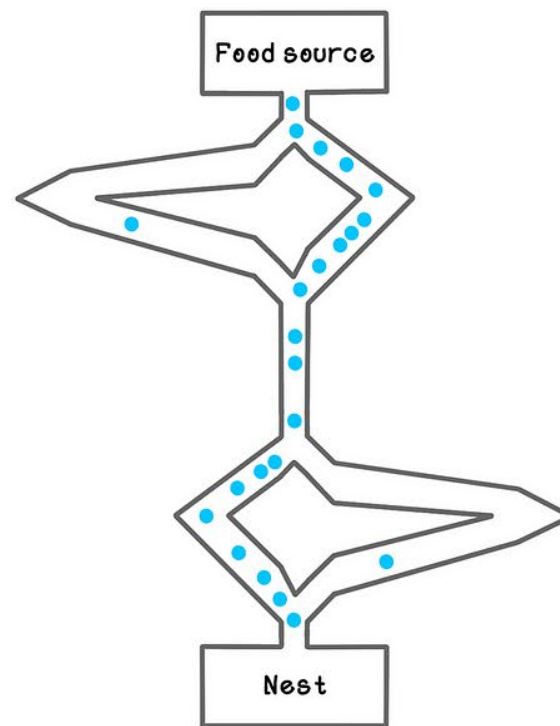
蚁群算法来自于蚂蚁寻找食物过程中发现路径的行为。蚂蚁并没有视觉却可以寻找到食物，这得益于蚂蚁分泌的信息素，蚂蚁之间相互独立，彼此之间通过信息素进行交流，从而实现群体行为

蚁群算法的基本原理就是蚂蚁觅食的过程。首先，蚂蚁在觅食的过程中会在路径上留下信息素（pheromone），并在寻找食物的过程中感知这种物质的强度，并指导自己的行为方向，他们总会朝着浓度高的方向前进

因此可以看得出来，蚂蚁觅食的过程是一个正反馈的过程，该路段经过的蚂蚁越多，信息素留下的就越多，浓度越高，更多的蚂蚁都会选择这个路段



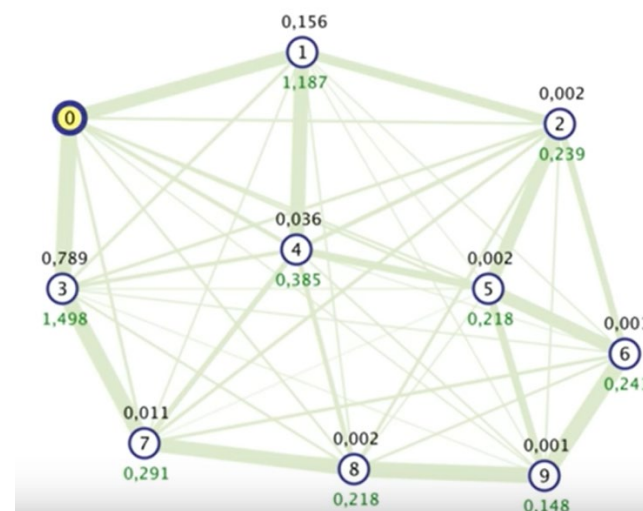
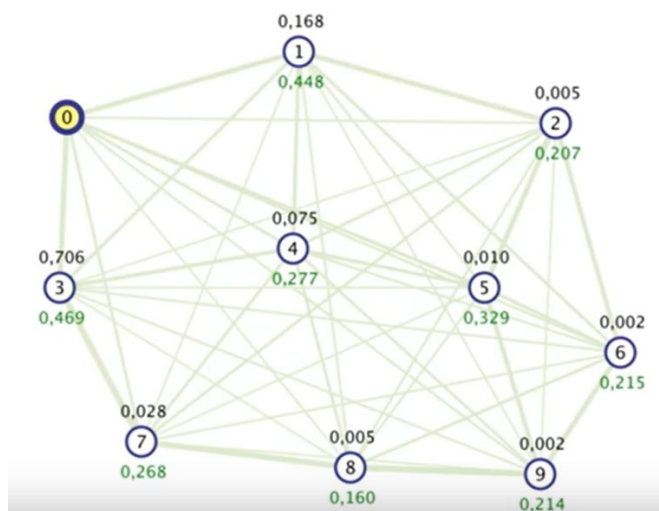
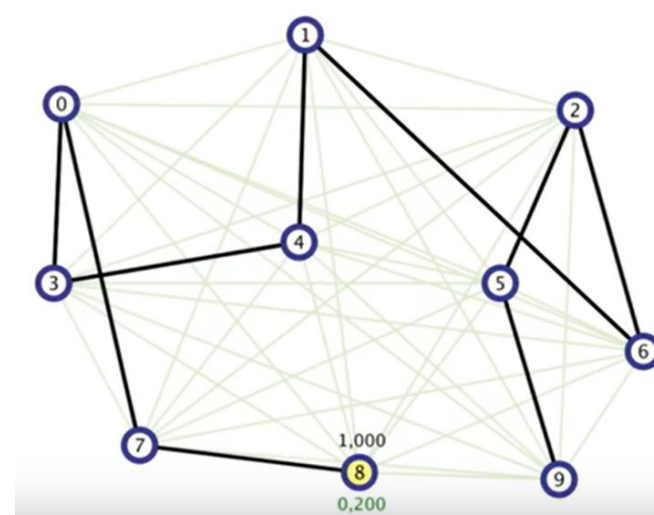
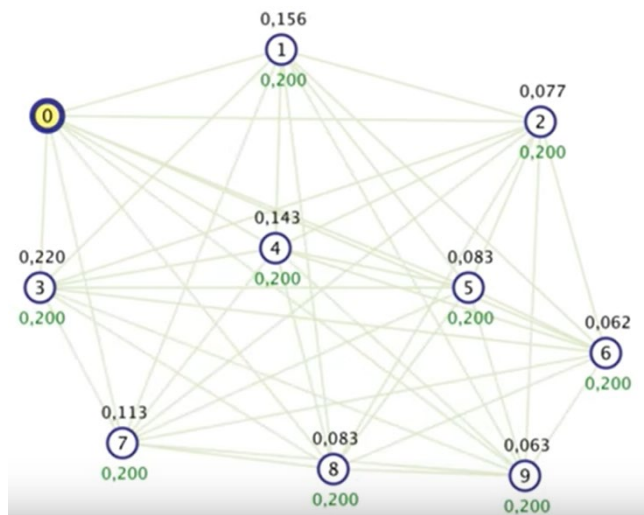
After 4 minutes



After 8 minutes

蚁群算法的步骤如下：

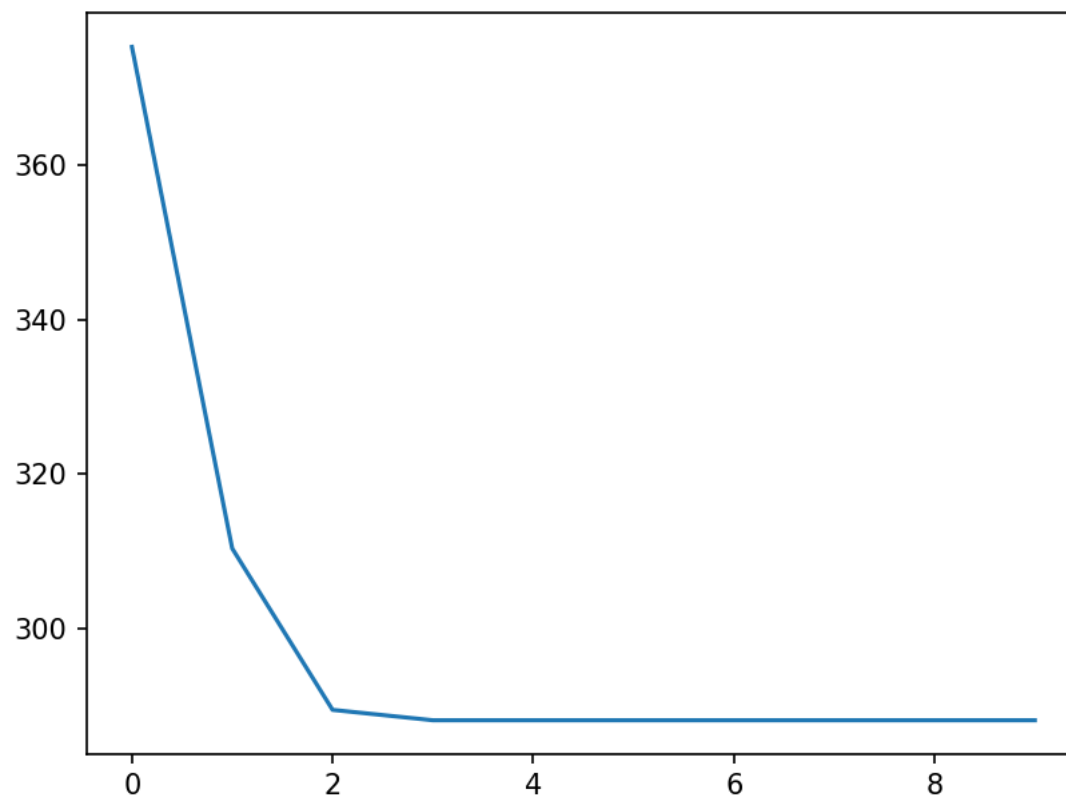
- 选择蚂蚁的数量，每个蚂蚁随机选择一个起点，初始化所有路线上信息素的浓度相等
- 每一个蚂蚁从不同顶点出发，依次选择自己的路径
  - 在每一个节点处，根据路径本身信息 $\eta$ 和信息素浓度 $\tau$ 选择下一个节点
  - 选择每个节点的概率正比于 $\tau^\alpha \eta^\beta$
- 选择自己的路径后，在被选择路径上留下信息素 $\Delta\tau = \frac{Q}{\text{路径长}}$
- 每一轮的信息素会以一定比例挥发，更新信息素浓度 $\tau = p\tau_0 + \Delta\tau$
- 重复以上迭代若干次，输出找到最优的解



需要选择哪些参数

- 蚂蚁的数量：一般蚂蚁的数量和顶点数量一样就可以
- 迭代的次数：根据计算资源选择
- $\alpha, \beta$ ：影响路径本身信息和信息素浓度的相对重要性
- 挥发系数 $p$ 、信息素总量 $Q$ ，只能根据具体问题来试

## 运行实例





**THANKS**