## 碎纸复原问题

### 问题分析

根据题意，将一张印有文字的纸沿纵向等宽切割成 19 张条形纸片，并打乱顺序。设计一个算法将打乱顺序的纸片按正确的顺序排列，并复原。该问题是一个 TSP（旅行商）问题，属于组合优化问题，被证明具有 NPC 计算复杂性。TSP 问题可以利用枚举法得到精确解，但当问题的维度增大时，会导致组合爆炸。因此考虑使用近似算法，常用的近似算法有贪婪算法、模拟退火算法、遗传算法、蚁群算法等。针对此问题，我们使用蚁群算法进行处理。

### 一维切割的情况

#### 问题的形式化描述

利用 python 的 PIL 模块读取附件 1 中的图片，灰度化后转换为矩阵。为第 个纸片对应的灰度矩阵，形式如下

矩阵中的每个元素的取值范围为 。定义所有纸片灰度矩阵的集合为 。该问题中，定义 为未分割前完整纸片的灰度矩阵， 为以任意顺序拼接的得到的灰度矩阵。

#### 问题的解

的一个最短的 Hamilton 回路（有向环）就是该问题的一个解， 的所有的 Hamilton 回路就构成了问题的解空间。由此，该问题的时间复杂度为 。以矩阵的形式表达，则矩阵 即为问题的一个解， 构成的集合 即为问题的搜索空间，求解的过程就是在 中搜索 的过程。

#### 蚁群算法

In [computer science](https://en.wikipedia.org/wiki/Computer_science) and [operations research](https://en.wikipedia.org/wiki/Operations_research), the **ant colony optimization** [algorithm](https://en.wikipedia.org/wiki/Algorithm) (**ACO**) is a [probabilistic](https://en.wikipedia.org/wiki/Probability) technique for solving computational problems which can be reduced to finding good paths through [graphs](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)). **Artificial Ants** stand for [multi-agent](https://en.wikipedia.org/wiki/Multi-agent) methods inspired by the behavior of real ants. The pheromone-based communication of biological [ants](https://en.wikipedia.org/wiki/Ant) is often the predominant paradigm used.[[2]](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms#cite_note-2) Combinations of Artificial Ants and [local search](https://en.wikipedia.org/wiki/Local_search_(optimization)) algorithms have become a method of choice for numerous optimization tasks involving some sort of [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)), e. g., [vehicle routing](https://en.wikipedia.org/wiki/Vehicle_routing_problem) and Internet [routing](https://en.wikipedia.org/wiki/Routing). The burgeoning activity in this field has led to conferences dedicated solely to Artificial Ants, and to numerous commercial applications by specialized companies such as [AntOptima](https://en.wikipedia.org/w/index.php?title=AntOptima&action=edit&redlink=1). As an example, [Ant colony optimization](https://en.wikipedia.org/wiki/Ant_colony_optimization)[[3]](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms#cite_note-3) is a class of [optimization](https://en.wikipedia.org/wiki/Optimization_(computer_science)) [algorithms](https://en.wikipedia.org/wiki/Algorithm) modeled on the actions of an [ant colony](https://en.wikipedia.org/wiki/Ant_colony). Artificial ‘ants’ (e.g. simulation agents) locate optimal solutions by moving through a [parameter space](https://en.wikipedia.org/wiki/Parameter_space) representing all possible solutions. Real ants lay down [pheromones](https://en.wikipedia.org/wiki/Pheromone) directing each other to resources while exploring their environment. The simulated ‘ants’ similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.[[4]](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms#cite_note-4) One variation on this approach is [the bees algorithm](https://en.wikipedia.org/wiki/Bees_algorithm), which is more analogous to the foraging patterns of the [honey bee](https://en.wikipedia.org/wiki/Honey_bee), another social insect.

for cycle in cycles:  
 for ant in ants:  
 select the most probably node to move  
 move ant to next node  
 record the pheromone that the ant left  
 update pheromone

为节点 与 节点 之前的距离， 为第 只蚂蚁爬过的路径总长， 为每只蚂蚁留下的 pheromone 总量， 为第 只蚂蚁爬过后对节点 到 之间路径 pheromone 浓度的改变， 是一个衰减系数，表征每次循环后 pheromone 浓度的减小， 表征第 次循环时 对 的 pheromone 浓度， 表征第 只蚂蚁从 爬到 的可能性。

##### 节点间的距离

蚁群算法中，蚂蚁从 到 的可能性与 到 的距离和 到 的 pheromone 浓度有关，因此需要一种表征 的方法。

对于两张纸片，我们认为它们边缘重合的越好，则它们之间的距离越相近。定义纸片 与纸片 之间的相似度为 ，则定义 。

|  |  |  |
| --- | --- | --- |
| 符号 | 含义 | numpy 调用 |
|  | 第一行元素构成的向量 |  |
|  | 最后一行元素构成的向量 |  |
|  | 第一列元素构成的向量 |  |
|  | 最后一列元素构成的向量 |  |

对于相邻的两片纸片 与 ， 的右边缘应与 的左边缘相似，即 与 相似。参考论文中采用的是比较 与 中黑色像素点的个数，即比较 与 ，只考虑了黑色像素点的个数，并没有考虑其位置。因此我们考虑采用余弦相似度来表征 与 的匹配度。

因为 中所有元素都为正数，所以 。

1. 当 时，，此时 与 的匹配度达到最大。
2. 当 与 正交时，匹配度最小。

### 二维切割的情况

以 进行切割，得到 个纸片，如果直接对所有碎片求距离矩阵，需要计算 次余弦相似度，算法复杂度太高。因此需要对纸片进行聚类达到降维的目的。

#### 聚类