



数学模型课程 智能算法

蚁群算法飞机巡航实验报告

指导教师： 殷 翔

专 业： 交通运输工程

学 号： 238908

姓 名： 张晨皓



目录

1.算法说明	1
1.1 蚁群算法	1
1.2 局部 2-邻边调整.....	2
1.3 算法框架	3
2.实验及参数设置.....	3
2.1 自拟的五角星小型案例（用于算法初学）	4
2.1.1 实验数据	4
2.2.2 参数设置.....	4
2.2.3 程序及结果分析	6
2.2 飞机巡航案例（本次作业）	8
2.2.1 实验数据	8
2.2.2 参数设置	9
3.飞机巡航案例程序及结果分析	10
3.1 程序介绍及结果说明	10
3.1.1 主要函数和功能	10
3.1.2 运行行能及结果	11
3.2 程序结果及参数分析	12
3.2.1 程序结果	12
3.2.2 参数分析	13

1. 算法说明

1.1 蚁群算法

个人理解，蚁群算法就是借鉴了蚂蚁的“群体智慧”，进而设计出的一种算法。单个蚂蚁在寻找最优路径的时候，其策略具有随机性，而当若干只蚂蚁共同行动的时候，对于最优路径的寻觅则表现得较为智慧，展现出群体智能的效果（有点像黏菌的最优路径寻觅？同时有种“量变产生质变”的味道，又有点像 AI/神经网络里面的“涌现”）。

算法设计人员受此启发，认为关键原因在于蚂蚁和蚂蚁之间存在“互相交流”，交流的媒介是信息素，又称为“弗洛蒙”。关于信息素的主要内容被我概括为以下几点：（1）单只蚂蚁会在行走过的路径上留下信息素，使得对应路径信息素浓度增加；（2）蚂蚁走的距离越长留下的信息素越少，走的距离越短信息素越多；（3）且蚂蚁更倾向于走信息素浓度高的路径，但决策是轮盘赌随机的（走信息素高的路径实现了正反馈机制、而轮盘赌随机则保证了跳出局部最优解的能力，易于寻找到全局最优解）；（4）信息素会随着时间的推移而挥发，在算法中是按照轮次挥发。

与其他优化算法相比，蚁群算法具有以下几个特点：(1)采用正反馈机制，使得搜索过程不断收敛，最终逼近最优解。(2)每个个体可以通过释放信息素来改变周围的环境，且每个个体能够感知周围环境的实时变化，个体间通过环境进行间接地通讯。(3)搜索过程采用分布式计算方式，多个个体可以同时进行并行计算，大大提高了算法的计算能力和运行效率（在后续的实际编程过程中，我并没有使用并行运算）。(4)启发式的概率搜索方式不容易陷入局部最优，易于寻找到全局最优解。

对于一个 TSP 商旅问题，在使用蚁群算法的求解过程中，存在若干轮次（每一次轮次信息素都会挥发）；在一个轮次当中，存在若干个蚂蚁，从随机的起点出发，按照根据可见度和信息素所计算的概率，轮盘赌式地、不重复地访问所有结点直到访问完毕回到起点；根据所有蚂蚁的行走路线和距离更新信息素矩阵的对应部分，本轮计算结束，信息素矩阵挥发，进入到下一轮次。

个人按照理解将蚁群算法的算法伪代码归纳如下。其中可见度矩阵 η_{mtx} 是由对应点对之间距离的倒数组成的，信息素矩阵 τ_{mtx} 也被成为弗洛蒙矩阵，一般初始化形式为全一矩阵。

使用蚁群算法求解 TSP 商旅问题伪代码

```
1  输入：信息素矩阵 tau_mtx, 信息素指数 alpha, 可见度矩阵 eta_mtx, 可见度指数 bet
    a, 信息素更新常量 Q, 信息素挥发系数 rho, 计算轮次 R, 蚂蚁个数 M
2  对于每一轮次 r
3      信息素挥发 tau_mtx = rho*tau_mtx (0 < rho < 1)
4      对于每一只蚂蚁 m (可并行计算)
5          随机生成初始起点 start_id
6          根据起点生成城市禁忌访问列表 taboo_l
7          若禁忌列表 taboo_l 中存在可访问的城市
8              根据信息素矩阵、可见度矩阵、禁忌列表, 更新可访问城市的选择概率
9              根据轮盘赌算法, 按照选择概率选择下一个访问城市 next_id
10             更新城市禁忌访问列表 taboo_l
11         记录蚂蚁 m 此轮行走总长度 d, 及对应的路径序列 path
12         更新蚂蚁 m 所经过对应的路径信息素 tau_mtx[] = tau_mtx[] + Q/d
13     保留截止到此轮次的 TSP 商旅问题的最优距离 D*, 及其对应的最优路径序列
14 输出：最优距离 D*, 最优路径序列 L*
15 结束
```

1.2 局部 2-邻边调整

由于蚁群算法是一种启发式的智能算法, 与精确算法相比缺少相关的最优性条件 (类似 KKT 这种的?) 的推导、数理证明, 因此在对凸问题或者 NP 问题的实际求解过程中免不了存在落入局部最优解的可能。为了避免或减少这种情况的发生, 可以使用一些动态规划的算法对部分解进行调整。在本次作业中, 老师要求使用 20 次的 **2-邻边调整** (复杂度较高), 我将主要的调整效果绘制如图 1 (主要是帮助自己理解)。

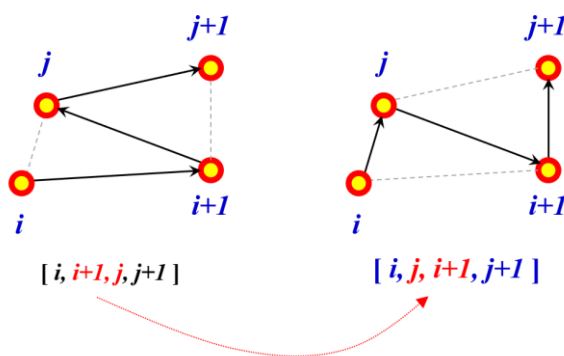


图 1 局部 2-邻边调整示意图

1.3 算法框架

将前述的局部 2-邻边调整技巧应用在蚁群算法当中，便得到了本次实验所使用的算法。其中，为区别前述的蚁群算法，特将使用局部 2-邻边调整的部分进行加粗和标红，算法伪代码描述如下。

使用局部 2-邻边调整的蚁群算法求解 TSP 商旅问题伪代码

```
1   输入：信息素矩阵 tau_mtx，信息素指数 alpha，可见度矩阵 eta_mtx，可见度指数 bet
    a，信息素更新常量 Q，信息素挥发系数 rho，计算轮次 R，蚂蚁个数 M
2   对于每一轮次 r
3       信息素挥发 tau_mtx = rho*tau_mtx (0 < rho < 1)
4       对于每一只蚂蚁 m（可并行计算）
5           随机生成初始起点 start_id
6           根据起点生成城市禁忌访问列表 taboo_l
7           若禁忌列表 taboo_l 中存在可访问的城市
8               根据信息素矩阵、可见度矩阵、禁忌列表，更新可访问城市的选择概率
9               根据轮盘赌算法，按照选择概率选择下一个访问城市 next_id
10              更新城市禁忌访问列表 taboo_l
11              记录蚂蚁 m 此轮行走的路径序列 path
12              对路径序列 path 进行 20 轮的 2-邻边算法
13              调整路径，得到优化后路径 path_op，该路径的行走总长度 d_op
14              更新蚂蚁 m 所经过对应的路径信息素 tau_mtx[] = tau_mtx[] + Q/d_op
15      保留截止到此轮次的 TSP 商旅问题的最优距离 D*，及其对应的最优路径序列
16  输出：最优距离 D*，最优路径序列 L*
17  结束
```

2.实验及参数设置

为学习复现算法，加深印象，我没有选择参考 MATLAB 代码的细节，而是首先自己设计了一个类似的五结点场景，从五角星小型案例入手，编写了可以运行的程序（麻雀虽小五脏俱全，主要用于进行可行性学习）；后将其进一步改写和扩展为作业所要求的飞机巡航案例，最后使用 MATLAB 代码进行细节的校正。

算法主要通过调用 NumPy 库编写，在 python 3.8 平台上运行，所有计算在六核 3.0GHz AMD Ryzen 5 4600H 处理器和 32GB RAM 的计算机上进行。现

分别将用于算法初学的五角星小型案例，以及本次作业的飞机巡航案例简单介绍如下。

2.1 自拟的五角星小型案例（用于算法初学）

2.1.1 实验数据

将案例中的五个结点分别标记为结点 1、2、3、4、5，五个结点分别为正五边形的 5 个顶点，如图 2 所示，五个结点并设定具体的位置坐标，而是展示了逻辑关系。边长设为 1 个单位长度，则内部五角星的边长为 1.62 个单位长度（按照五边形的角度计算得到的平面距离，在该案例下并未考虑球面距离）。

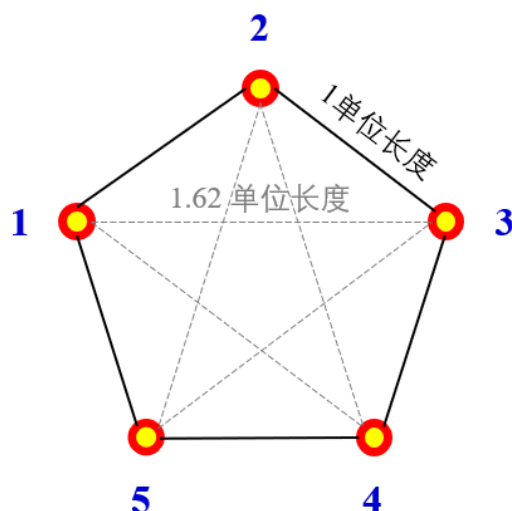


图 2 五角星小案例示意图

2.2.2 参数设置

在该案例中，一些比较重要的参数主要包括：仿真轮次 rounds、蚂蚁个数 ant_number、信息素指数 alpha（我当时编写程序时都将信息素称之为弗洛蒙了）、可见度指数 beta、信息素挥发系数 rho 等。

将上述参数分别设置为表 1，作为小型案例的参数设置情况，以备于为后续的结果分析，尤其是敏感性分析服务。

表 1 五角星小型案例参数设置

轮次数	蚂蚁数	信息素指数	可见度指数	挥发系数
10	10	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9
	100	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9
100	10	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9
	100	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9

2.2.3 程序及结果分析

五角星小案例的代码已经上传至 <https://github.com/Reminscene/ACO-Algorithm-for-solving-TSP/blob/main/1.%EF%BC%88E7%94%A8%E4%BA%8E%E7%AE%97%E6%B3%95%E5%88%9D%E5%AD%A6%EF%BC%89%E8%9A%81%E7%BE%A4%E7%AE%97%E6%B3%95%E4%BA%94%E8%A7%92%E6%98%9F%E5%B0%8F%E7%AE%97%E4%BE%8B.py>，命名为：“（用于算法初学）蚁群算法五角星小案例.py”，并未随作业案例文件一同发送至邮箱，其运行结果见表 2。

表 2 五角星小型案例程序结果

轮次数	蚂蚁数	信息素指数	可见度指数	挥发系数	运行时间/s	最短距离
10	10	0	0	0.1	0.0183	5*
				0.9	0.0185	5*
			1	0.1	0.0185	5*
				0.9	0.0185	5*
		1	0	0.1	0.0188	5*
				0.9	0.0196	5*
			1	0.1	0.0195	5*
				0.9	0.0188	5*
	100	0	0	0.1	0.1804	5*
				0.9	0.1812	5*
			1	0.1	0.1801	5*
				0.9	0.1814	5*
		1	0	0.1	0.1849	5*
				0.9	0.1805	5*
			1	0.1	0.1808	5*
				0.9	0.1836	5*
100	10	0	0	0.1	0.1865	5*
				0.9	0.1849	5*
			1	0.1	0.1916	5*
				0.9	0.1866	5*
		1	0	0.1	0.1837	5*
				0.9	0.1881	5*
			1	0.1	0.1862	5*
				0.9	0.1890	5*
				0.1	1.8454	5*
				0.9	1.8447	5*
	100	0	0	0.1	1.8443	5*
				0.9	1.8335	5*
			1	0.1	1.8395	5*
				0.9	1.8303	5*
		1	1	0.1	1.8214	5*
				0.9	2.2611	5*

最短距离中，*表示是 TSP 的最优距离

现对上表中红色加粗标注的 6 种情况对应的收敛曲线进行展示，其中红线为每轮中蚂蚁平均 TSP 距离，蓝线为最短 TSP 距离，具体内容如图 3 所示。

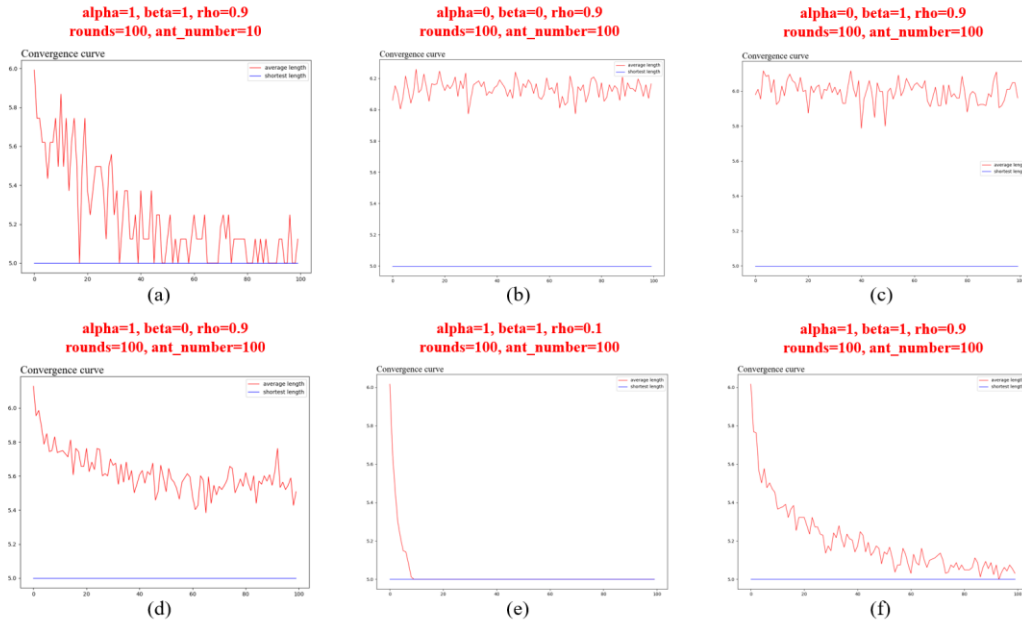


图 3 小案例 6 种情景对应的收敛曲线

根据上图收敛曲线，可初步对相关的参数进行研究分析，分析发现如下：

- **2-邻边算法的引入：**由于引入了 2-邻边算法对 TSP 路径进行了调整，故对于 5 结点的小型案例，尽管平均距离都在波动（红色），但还是非常容易地就得到了最短距离（蓝色），蓝色线并没有明显的下降特征。
- **蚂蚁数量 ant_number：**通过比较上图的子图（a）和子图（f）可知，两者的区别在于蚂蚁数量的不同，分别对应为 10 只和 100 只的情况；在小案例中，初步发现增加蚂蚁数量可以让每轮蚂蚁 TSP 的平均距离波动更小，但总体趋势不变。
- **信息素指数 alpha：**通过比较上图的子图（c）和子图（f）可知，两者的区别在于信息素指数 α 的不同，分别对应为 0 和 1 的情况；在小案例中，初步发现当将 α 从 0 调整到 1 的时候，每一轮的 TSP 平均距离收敛性变好，出现明显的向最短 TSP 距离收敛的趋势。
- **可见度指数 beta：**通过比较上图的子图（d）和子图（f）可知，两者的区别在于可见度指数 β 的不同，分别对应为 0 和 1 的情况；在小案例中，初步发现当将 β 从 0 调整到 1 的时候，每一轮的 TSP 平均距离收敛速度加快，能够更快地向最短 TSP 距离收敛。

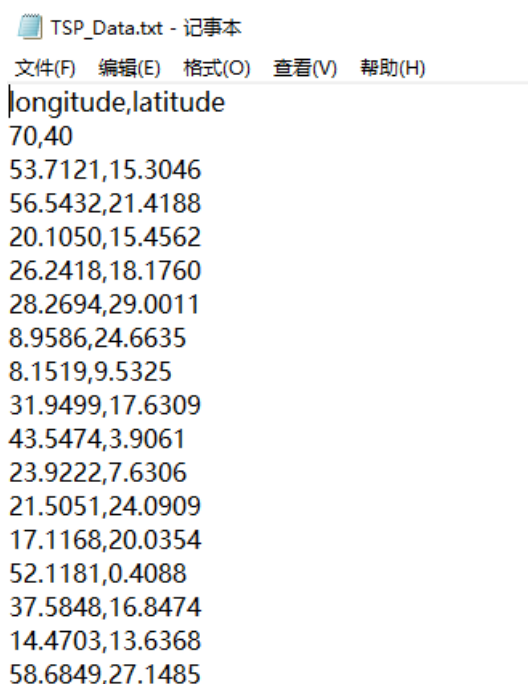
- **挥发系数 ρ :** 通过比较上图的子图 (e) 和子图 (f) 可知, 两者的区别在于信息素挥发系数 ρ 的不同, 分别对应为 0.1 和 0.9 的情况; 在小案例中, 初步发现当将 ρ 从 0.1 调整到 0.9 的时候, 每一轮的 TSP 平均距离的结果出现波动, 收敛速度变慢, 这应当是在小案例中与其他数值共同作用的结果。

根据上述的讨论和对比分析, 可推测在小型案例中, 作为动态规划算法的 2-邻边算法的引入直接有助于找到最优 TSP 路径; 增加蚂蚁数量, 可以减少每一轮平均路径结果的波动性; 信息素指数变为 0 时, 容易陷入局部最优解, 无法实现很好的收敛; 可见度指数变为 0 时, 有一定的收敛倾向, 但不是最优。

2.2 飞机巡航案例 (本次作业)

2.2.1 实验数据

在飞机巡航的大案例作业当中, 给定了出发点 (程序中将其命名为 1-start) 和 100 个需要访问的其他城市结点, 飞机的飞行速度为先前给定的 1000km/h (对客机来说稍微有点快哈)。所设置的 101 个点的经度和纬度坐标以文本形式给出, 储存在 data 文件夹的 'TSP_Data.txt' 当中。与老师所给数据不同, 为了方便, 我将其整理为以下格式, 并将起点 1-start (经度 70, 纬度 40) 存放于其中, 如图 4 所示。



```
TSP_Data.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
longitude,latitude
70,40
53.7121,15.3046
56.5432,21.4188
20.1050,15.4562
26.2418,18.1760
28.2694,29.0011
8.9586,24.6635
8.1519,9.5325
31.9499,17.6309
43.5474,3.9061
23.9222,7.6306
21.5051,24.0909
17.1168,20.0354
52.1181,0.4088
37.5848,16.8474
14.4703,13.6368
58.6849,27.1485
```

图 4 飞机巡航大案例的结点数据 'TSP_Data.txt'

2.2.2 参数设置

在该飞机巡航大型案例中，将仿真轮次 rounds、蚂蚁个数 ant_number、信息素指数 alpha、可见度指数 beta、信息素挥发系数 rho 等参数分别设置为表 3，作为大型案例的参数设置情况，以备于为后续的结果分析，尤其是敏感性分析服务，表格最后一行的设置即为本次作业所直接要求的内容。

表 3 飞机巡航大型案例参数设置

轮次数	蚂蚁数	信息素指数	可见度指数	挥发系数
10	10	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9
	50	0	1	0.1
				0.9
		1	0	0.1
				0.9
			1	0.1
				0.9
50	10	0	0	0.1
				0.9
		1	1	0.1
				0.9
			0	0.1
				0.9
	50	0	1	0.1
				0.9
		1	0	0.1
				0.9
			1	0.1
				0.9
1000	100	1	5	0.1

3.飞机巡航案例程序及结果分析

3.1 程序介绍及结果说明

3.1.1 主要函数和功能

为了更好地帮助理解程序，现对程序的主要函数和功能进行介绍：本次编写的程序主要包括了以下函数，主要实现的功能分别如下表 4 所示。

表 4 飞机巡航大型案例程序主要函数和功能介绍

函数	功能介绍
progress_bar(process_rate)	输入： 进度的比率 输出： 进度条 功能： 可视化展示计算进度
distance_calculation(i_index, j_index)	输入： 结点 i 的索引，结点 j 的索引 输出： 结点 i 和 j 之间的球面距离 功能： 计算两个坐标点（城市）之间的球面距离
random_selection(rate)	输入： 由概率分布组成的列表 输出： 对应的下标索引 功能： 按照概率的分布，以轮盘赌的形式随机输出某概率对应的下表索引
calculate_probability(tau_m, eta_m, taboo_l, current_index)	输入： 信息素矩阵、可见度矩阵、禁忌访问列表、当前结点的位置索引 输出： 前往可访问的各个结点（城市）的概率 功能： 生成由概率分布组成的列表
ant_colony_optimization(start_index, tau_m)	输入： 蚂蚁的起点索引、信息素矩阵 输出： 单只蚂蚁的行走路径序列 功能： 以蚁群算法的概率计算规则，用来指导单只蚂蚁完成整条路径行走

3.1.2 运行性能及结果

本次作业的飞机巡航大案例的代码、结果、图片文件一同打包发送至老师邮箱，分别见 result 和 figure 文件夹，请查阅。

此外，代码已经上传至 <https://github.com/Reminscene/ACO-Algorithm-for-solving-TSP/blob/main/2.%EF%BC%88%E6%9C%AC%E6%AC%A1%E4%BD%9C%E4%B8%9A%EF%BC%89%E8%9A%81%E7%BE%A4%E7%AE%97%E6%B3%95%E9%A3%9E%E6%9C%BA%E5%B7%A1%E8%88%AA%E6%A1%88%E4%BE%8B.py>，命名“（本次作业）蚁群算法飞机巡航案例.py”。

轮次数 rounds=1000，蚂蚁数量 ant_number=100，信息素指数 alpha=1，可见度指数 beta=5，信息素挥发系数 rho=0.1，计算的相关指标如下所述。

- 计算用时：9500s+（写得比较菜）；
- 最短路长度：39266.0959087068 千米；
- 飞机用时：39.266 小时；
- 最短路序列：[1, 17, 3, 45, 67, 2, 92, 87, 83, 74, 30, 82, 48, 72, 14, 27, 10, 84, 97, 31, 79, 77, 85, 65, 64, 11, 76, 69, 94, 70, 19, 63, 62, 66, 26, 29, 34, 90, 86, 8, 39, 78, 47, 23, 58, 81, 25, 68, 7, 22, 71, 37, 32, 13, 24, 49, 28, 57, 88, 61, 16, 91, 41, 4, 73, 33, 75, 5, 54, 53, 12, 89, 6, 96, 55, 44, 38, 9, 60, 40, 18, 20, 42, 15, 50, 80, 51, 98, 100, 56, 21, 99, 101, 52, 46, 59, 93, 43, 36, 35, 95, 1]

计算过程的收敛曲线如图 5 所示，计算结果的 TSP 最短路可视化展示如图 6 所示，其中图 6 需要运行“3.可视化巡航路径.py”文件才能得到，已一并与上述代码打包发送至邮箱。

另，“3.可视化巡航路径.py”已经上传至 <https://github.com/Reminscene/ACO-Algorithm-for-solving-TSP/blob/main/3.%E5%8F%AF%E8%A7%86%E5%8C%96%E5%B7%A1%E8%88%AA%E8%B7%AF%E7%BA%BF.py>。

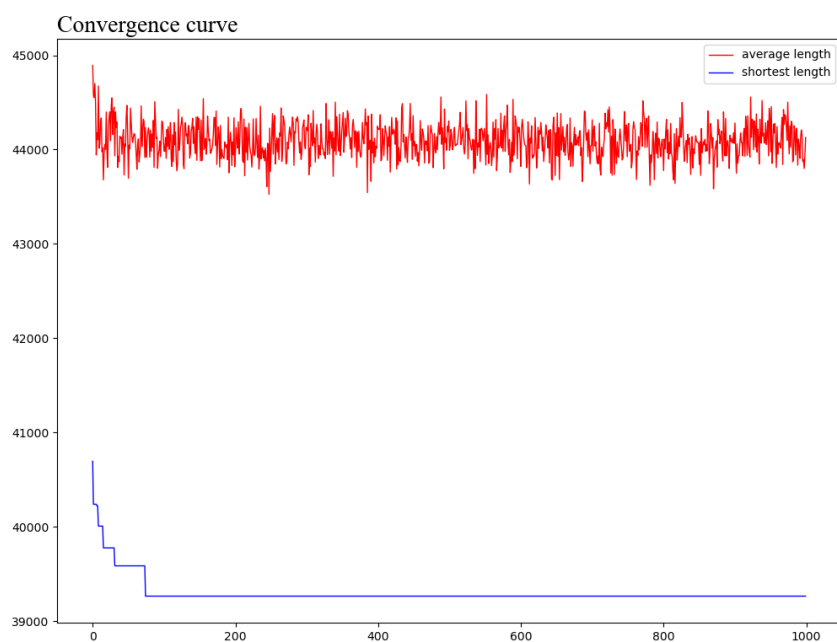


图 5 飞机巡航大案例的收敛曲线

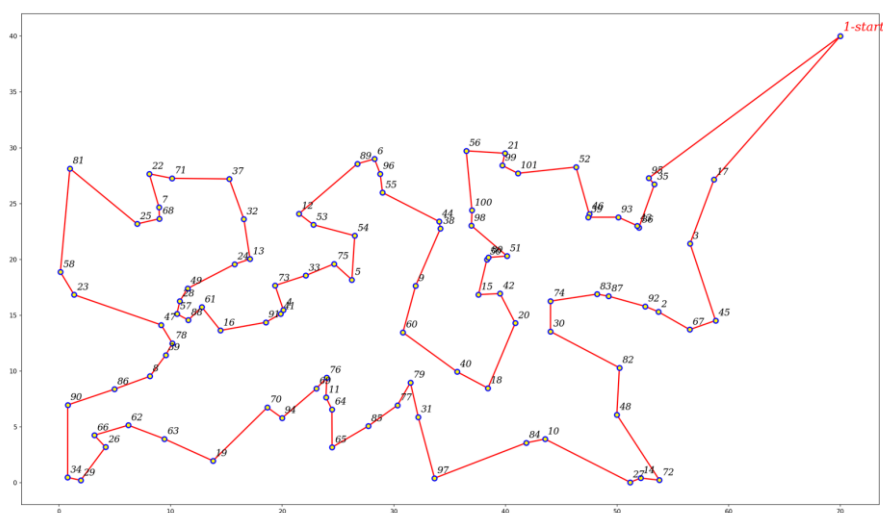


图 6 飞机巡航大案例的最优 TSP 路径

3.2 程序结果及参数分析

3.2.1 程序结果

除了 3.1 中的，作业所要求的给定场景下的运行结果之外，在其他不同参数设置情境下，运行结果见表 5。

表 5 飞机巡航大型案例程序结果

轮次数	蚂蚁数	信息素指数	可见度指数	挥发系数	运行时间/s	最短距离/km
10	10	0	0	0.1	9.7621	41751.5193
				0.9	9.8279	42155.0133
			1	0.1	9.6538	41878.3169
				0.9	9.8350	41033.2448
		1	0	0.1	9.8092	40794.4538
				0.9	10.0089	41096.8302
			1	0.1	9.7058	40804.7959
				0.9	9.7262	42086.5683
	50	0	0	0.1	48.5635	41039.5799
				0.9	48.6761	40817.7995
			1	0.1	48.9905	41410.9106
				0.9	48.8335	41606.5113
		1	0	0.1	47.8750	40751.2638
				0.9	48.6026	40608.2545
			1	0.1	47.6500	40405.0869
				0.9	48.5014	41264.7262
50	10	0	0	0.1	48.4296	41159.4285
				0.9	47.8382	40746.2304
			1	0.1	48.8569	41300.7974
				0.9	48.2094	41114.9367
		1	0	0.1	47.7763	40395.6466
				0.9	48.4982	41186.8759
			1	0.1	48.6081	39939.1947
				0.9	48.8771	41196.0192
	50	0	0	0.1	238.8407	41244.1083
				0.9	247.1692	41150.7931
			1	0.1	240.0884	40108.2917
				0.9	242.1187	40544.5307
		1	0	0.1	234.8949	40012.8270
				0.9	240.5881	41009.5220
			1	0.1	232.4664	39971.5286
				0.9	235.7804	40851.5841
1000	100	1	5	0.1	9500.0000+	39266.0959*

最短距离中，*表示是 TSP 的最优距离（计算可能有一定偏差，但差别不大）

3.2.2 参数分析

现对前表中红色加粗标注的 6 种情况对应的收敛曲线进行分析和展示，其中红线为每轮中蚂蚁平均 TSP 距离，蓝线为最短 TSP 距离，具体内容见图 7 及分析内容。

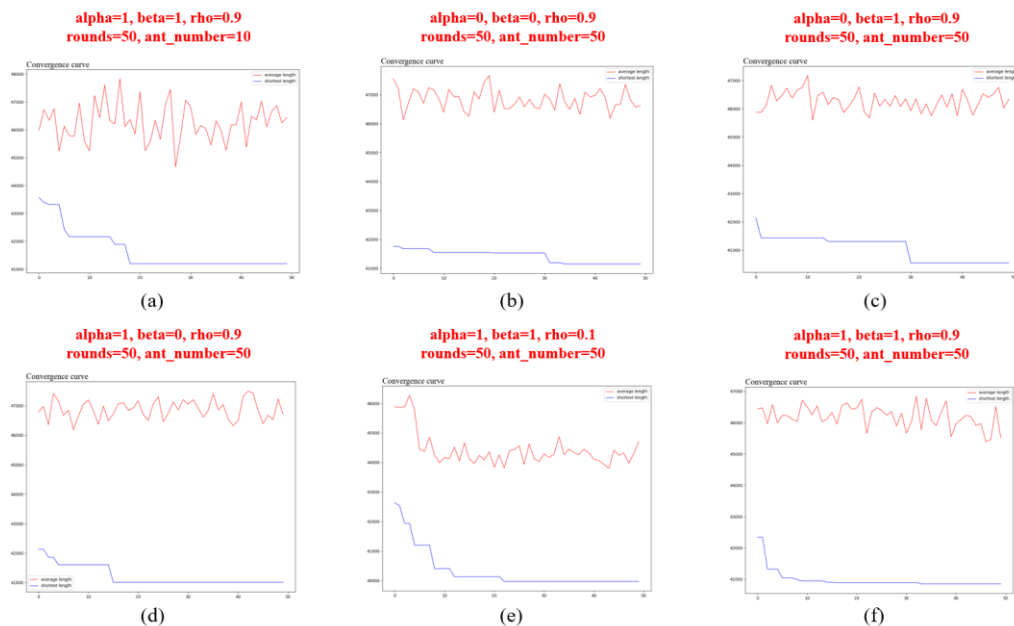


图 7 飞机巡航大案例 6 种情景对应的收敛曲线

根据上图收敛曲线，可初步对相关的参数进行研究分析，分析发现如下：

- **蚂蚁数量 ant_number:** 通过比较上图的子图（a）和子图（f）可知，两者的区别在于蚂蚁数量的不同，分别对应为 10 只和 50 只的情况；在飞机巡航大型案例中，发现增加蚂蚁数量可以让每轮蚂蚁 TSP 的平均距离波动更小，这与小型案例相同，也不难理解，抽样次数增加的时候数据结果的波动幅度会更小。
- **信息素指数 alpha:** 通过比较上图的子图（c）和子图（f）可知，两者的区别在于信息素指数 alpha 的不同，分别对应为 0 和 1 的情况；在飞机巡航大型案例中，发现当将 alpha 从 0 调整到 1 的时候，每一轮的 TSP 最短距离（蓝色折线）收敛性速度加快、收敛明显提前，出现明显的向最优 TSP 距离收敛的趋势，但平均 TSP 距离依然存在波动和未收敛的情况。
- **可见度指数 beta:** 通过比较上图的子图（d）和子图（f）可知，两者的区别在于可见度指数 beta 的不同，分别对应为 0 和 1 的情况；在飞机巡航大型案例中，初步发现当将 beta 从 0 调整到 1 的时候，每一轮的 TSP 最短距离（蓝色折线）收敛性速度加快、收敛明显提前，能够更快地向最优 TSP 距离收敛，但平均 TSP 距离依然存在波动和未收敛的情况。
- **挥发系数 rho:** 通过比较上图的子图（e）和子图（f）可知，两者的区别在于信息素挥发系数 rho 的不同，分别对应为 0.1 和 0.9 的情况；在飞机

巡航大型案例中，初步发现当将 ρ 从 0.1 调整到 0.9 的时候，每一轮的 TSP 平均距离的结果出现波动，收敛速度变慢，而较小挥发系数的时候，平均距离的波动值有明显收敛倾向。

根据上述的讨论和对比分析，可推测在飞机巡航大型案例中，与小型案例相似，增加蚂蚁数量，可以减少每一轮平均路径结果的波动性；信息素指数变为 0 时，无法实现很好的收敛，收敛速度较慢；可见度指数变为 0 时，同样无法实现很好的收敛，收敛速度较慢；挥发系数受具体数据和情景影响，并没有明显的规律。