

目录

- 1 实验内容..... 1
 - 1.1 TSP 旅行商问题..... 1
- 2 算法原理..... 1
 - 2.1 Genetic Algorithm 遗传算法 1
 - 2.2 Ant Colony Optimization 蚁群算法 2
 - 2.3 Particle Swarm Optimization 粒子群算法 2
- 3 实验过程与结果分析..... 2
 - 3.1 TSP 旅行商问题..... 3
 - 3.1.1 遗传算法..... 4
 - 3.1.2 蚁群算法..... 7
 - 3.1.3 粒子群算法..... 12
 - 3.1.4 算法比较..... 16
- 参考文献..... 17

1 实验内容

1.1 TSP 旅行商问题

求解货郎担（旅行商）问题，即 TSP。要求尽可能快地给出较优结果，最好能优于已知最优解。使用 TSPLIB 数据

<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>

请分别使用 Genetic Algorithm、Ant Colony Optimization 、 Particle Swarm Optimization 等算法实现，并比较不同算法的优化结果和运行速度。

2 算法原理

2.1 Genetic Algorithm 遗传算法

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。它是由美国的 J.Holland 教授 1975 年首先提出，其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。遗传算法的流程如图 1 所示：

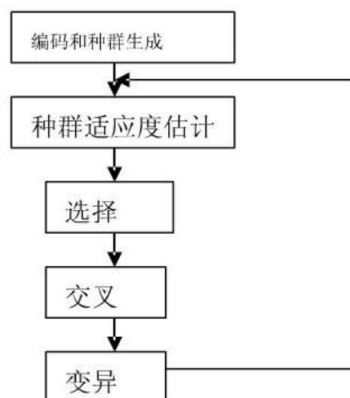


图 1 遗传算法流程图

由图 1 可知，遗传算法共分为五个部分，在下面的章节将对五个部分的定义进行具体说明。

2.2 Ant Colony Optimization 蚁群算法

蚁群系统(Ant System 或 Ant Colony System)是由意大利学者 Dorigo、Maniezzo 等人于 20 世纪 90 年代首先提出来的。他们在研究蚂蚁觅食的过程中，发现单个蚂蚁的行为比较简单，但是蚁群整体却可以体现一些智能的行为。例如蚁群可以在不同的环境下，寻找最短到达食物源的路径。这是因为蚁群内的蚂蚁可以通过某种信息机制实现信息的传递。后又经进一步研究发现，蚂蚁会在其经过的路径上释放一种可以称之为“信息素”的物质，蚁群内的蚂蚁对“信息素”具有感知能力，它们会沿着“信息素”浓度较高路径行走，而每只路过的蚂蚁都会在路上留下“信息素”，这就形成一种类似正反馈的机制，这样经过一段时间后，整个蚁群就会沿着最短路径到达食物源了。

蚁群算法的基本原理：

- 1、蚂蚁在路径上释放信息素。
- 2、碰到还没走过的路口，就随机挑选一条路走。同时，释放与路径长度有关的信息素。
- 3、信息素浓度与路径长度成反比。后来的蚂蚁再次碰到该路口时，就选择信息素浓度较高路径。
- 4、最优路径上的信息素浓度越来越大。
- 5、最终蚁群找到最优寻食路径。

2.3 Particle Swarm Optimization 粒子群算法

粒子群算法，也称粒子群优化算法或鸟群觅食算法（Particle Swarm Optimization），缩写为 PSO，是由 J. Kennedy 和 R. C. Eberhart 等开发的一种新的进化算法(Evolutionary Algorithm - EA)。PSO 算法属于进化算法的一种，和模拟退火算法相似，它也是从随机解出发，通过迭代寻找最优解，它也是通过适应度来评价解的品质，但它比遗传算法规则更为简单，它没有遗传算法的“交叉”(Crossover)和“变异”(Mutation)操作，它通过追随当前搜索到的最优值来寻找全局最优。这种算法以其实现容易、精度高、收敛快等优点引起了学术界的重视，并且在解决实际问题中展示了其优越性。粒子群算法是一种并行算法。

3 实验过程与结果分析

3.1 TSP 旅行商问题

旅行商问题，即 TSP 问题（Traveling Salesman Problem）又译为旅行推销员问题、货郎担问题，是数学领域中著名问题之一。假设有一个旅行商人要拜访 n 个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

迄今为止，这类问题中没有一个找到有效算法。倾向于接受 NP 完全问题（NP-Complete 或 NPC）和 NP 难题（NP-Hard 或 NPH）不存在有效算法这一猜想，认为这类问题的大型实例不能用精确算法求解，必须寻求这类问题的有效的近似算法。

我们采用 a280, att48 等数据集对 TSP 问题进行研究。att48 数据集给出了四十八个城市的城市坐标，同时给出了城市编号，可用欧式距离表示两城市之间的距离。同时，数据集给出了已知的最优路径：
[1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 20, 47, 21, 32, 39, 48, 5, 42, 24, 10, 45, 35, 4, 26, 2, 29, 34, 41, 16, 22, , 23, 14, 25, 13, 11, 12, 15, 40, 9]。
将该方案进行可视化处理，得到的路径如图 1 所示：

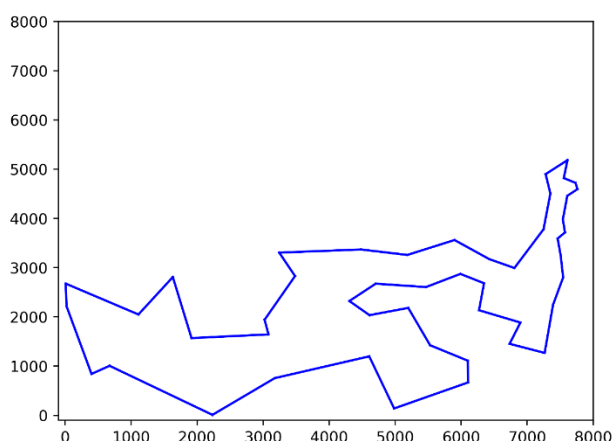


图 1 已知最优解路径图

计算图一路径长度，得到距离为 33061.63。

接下来分别通过遗传算法、蚁群算法、粒子群算法对 TSP 问题进行研究。并对各个算法的结果进行比较研究。

考虑到遗传算法、蚁群算法、粒子群算法均为概率算法，因此具有一定的随机性，同时均容易落入局部最优解，相同一组参数可能会得到不同的运行结果。因此在对不同参数进行评估的时候，不对参数能否得到较优结果进行讨论。另外，在比较各算法优劣性时，取该算法在实验中的平均表现。

3.1.1 遗传算法

根据遗传算法的思路，首先需要对数据进行编码及种群生成。数据集提供了每个城市的坐标与编号，二者对应，因此采用城市编号作为编码。初始化种群：随机生成一个染色体，任意交换两城市顺序位置，如果总距离减小，则更新改变染色体。

种群的适应度估计：采取方案的总距离作为适应度评判指标。总距离越短则认为适应度越高。

选择：设置参数存活率、弱者存活率两个参数。在对所有基因的适应度进行从高到低排序后，存活率指适应度从高到低选取的比例；弱者存活率指在本应淘汰的部分，随机选取部分“幸运儿”保留。

交叉：采用次序交叉法。首先随机地在双亲中选择两个杂交点，再交换杂交段，其他位置根据双亲城市的相对位置确定。例如：父代为

- $A1 = \{3, 1, 7, 6, \underline{8, 5, 2, 0}, \underline{9}, 4\}$
- $A2 = \{1, 5, 4, 6, \underline{9, 0, 3, 2}, \underline{8}, 7\}$

下划线标出了二者需要交换的部分，以 A1 为例，首先从需交换片段的下一位开始，将序列变为：

- $9-4-3-1-7-6-\underline{8-5-2-0}$

考虑到要将 A2 中的 9, 0, 3, 2 换入，因此去掉此序列中的 9, 0, 3, 2，防止出现非法序列，得到：

- $4-1-7-6-8-5$

最后，将新的序列插入到新的片段后，需要注意，交换的片段不改变相对位置。得到

- $B1 = \{7, 6, 8, 5, 9, 0, 3, 2, 4, 1\}$

A2 的处理方法同理。

变异：按照给定的变异率，对选定变异的个体，随机地取三个整数，满足 $1 < u < v < w < \text{城市数}$ ，把 v 、 u 之间（包括 u 和 v ）的基因段插到 w 后面。

接下来进行测试，取初始参数如表 1 所示：

表 1 遗传算法初始参数

种群数	改良次数	进化次数	存活率	弱者存活率	变异率	杂交率
2000	500	500	0.3	0.4	0.1	1.0

经过多次实验，得到实验平均最优路径为

[40, 33, 41, 23, 9, 44, 34, 3, 25, 1, 28, 4, 47, 38, 31, 20, 46, 12, 24, 13, 22, 10, 14, 45, 30, 43, 17, 6, 27, 5, 36, 18, 26, 16, 42, 29, 35, 32, 19, 11, 39, 8, 37, 7, 21, 2, 0]

最优距离为 36758.48，运行时间为 1: 52.00。各代最优距离的下降过程如图 2 所示

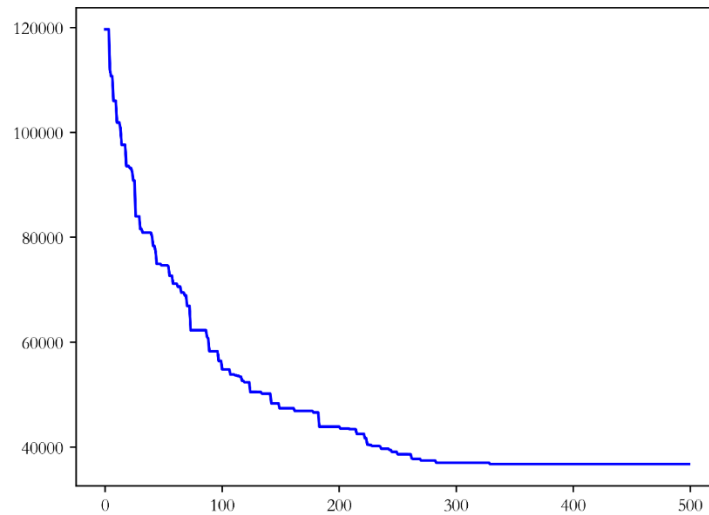


图 2 遗传算法各次迭代最优距离下降过程

由图 2 可以发现，迭代次数在 300 次左右时基本已经收敛。将得到的最优路径绘制得图 3

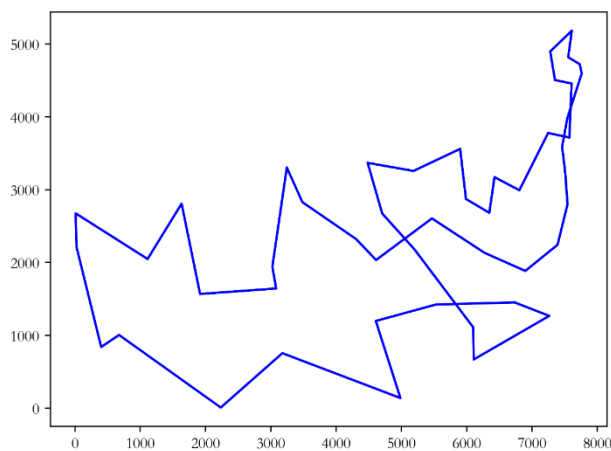


图 3 遗传算法所求路径

从图 3 可以较为直观地观察求解路径，与图 1 进行对比，直观上可认为表现较好。接下来对具体参数进行研究，在研究一个特定参数时，其他参数的设置如表 1 所示。

首先对变异率进行研究，变异率影响发生变异的概率。将变异率初始值设置为 0.1，变化区间为 0.1 到 0.4，变化步长为 0.1。得到在不同变异率下各次迭代距离下降过程，由图 4 所示

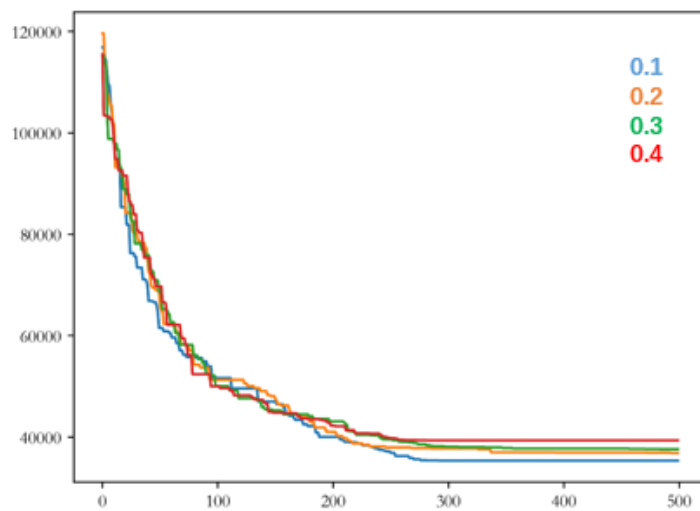


图 4 变异率对遗传算法的影响

由图 4 可得，固定其他参数，改变变异率，对算法没有明显影响。

接下来研究杂交率对实验的影响。杂交率指选中两个父代基因后，二者进行杂交的概率。将杂交率初始值设置为 1.0，变化区间为 1.0 到 0.6，变化步长为 0.1。得到在不同杂交率下各次迭代距离下降过程，由图 5 所示

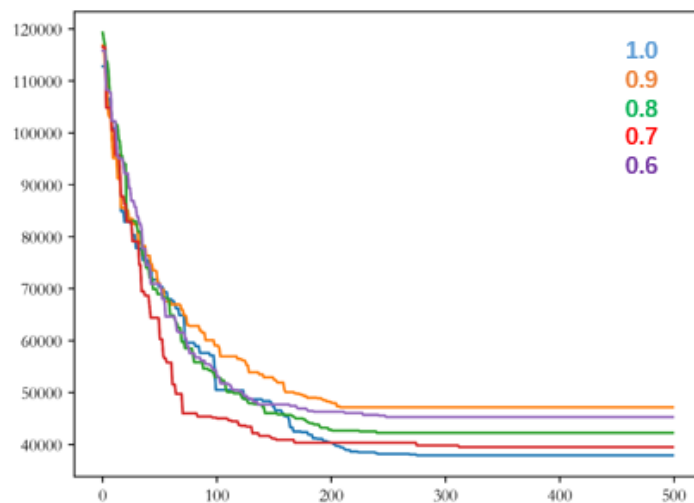


图 5 杂交率对遗传算法的影响

由图 5 可得，在变异率为 0.7 时，距离下降速度最快。总体上来说，固定其他参数，改变变异率，对算法没有明显影响。

接着研究存活率对实验的影响。存活率指每次选择适应度高的个体的比例。将杂交率初始值设置为 0.1，变化区间为 0.1 到 0.4，变化步长为 0.1。得到在不同杂交率下各次迭代距离下降过程，由图 6 所示

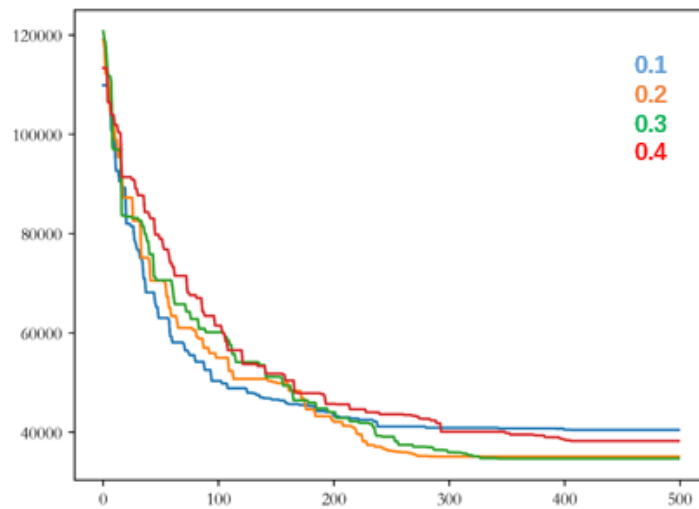


图 6 存活率对遗传算法的影响

由图 6 可得，固定其他参数，存活率越小，距离下降速度越快，收敛速度越快。将以上的研究结果整理，得到表 2：

表 2 遗传算法相关参数对算法的影响

参数名称	对算法的影响
变异率	需要考虑参数之间的共同作用
杂交率	0.7 时下降速度较快
信息素挥发速度	数值越小，下降速度越快，收敛速度越快

3.1.2 蚁群算法

基于蚁群算法的思路，结合 TSP 问题，对蚁群算法进行一定的调整。蚂蚁选择城市的概率与城市之间的距离和当前连接支路上所包含的信息素余量有关。为了强制蚂蚁进行合法的周游，直到一次周游完成后，才允许蚂蚁游走已访问过的城市（这可由禁忌表来控制）。

蚁群算法的信息素更新公式如式 1 式 2 所示：

$$\tau_{ij}(t+n)=(1-\rho)\cdot\tau_{ij}(t)+\Delta\tau_{ij} \tag{1}$$

$$\Delta\tau_{ij}(t)=\sum_{k=1}^m\Delta\tau_{ij}^k(t) \tag{2}$$

其中， $\tau_{ij}(t)$ 为 t 时刻(i, j)路径上的信息素浓度， ρ 为信息素挥发速度， $\Delta\tau_{ij}$ 为(i, j)路径上信息素浓度的增量， m 为蚂蚁总数， $\Delta\tau_{ij}^k(t)$ 为第 k 只蚂蚁经过后留下的信息素增量。

信息素采取蚁周模型与蚁量模型结合的更新方式。首先在一次迭代中，先对每个蚂蚁进行研究，所有蚂蚁环游结束后，对各个路径上的信息素浓度进行统一更新，信息素的增量与每两个城市之间的距离相关。

蚂蚁对路径的选择采用轮盘赌的方式，通过该路径上的信息素浓度以及启发函数进行。

接下来进行测试，取初始参数如表 3 所示：

表 3 蚁群算法初始参数					
蚂蚁 个数	信息素重 要因子	启发函数 重要因子	信息素挥发 速度	信息素参数	迭代次数
40	2	2	0.2	1000	200

经过多次实验，得到实验平均最优路径为
 [41. 4. 47. 38. 31. 20. 12. 24. 13. 22. 10. 11. 14. 39. 8. 0. 7. 37.
 30. 43. 17. 6. 27. 35. 29. 42. 16. 26. 18. 36. 5. 45. 32. 19. 46. 2.
 21. 15. 40. 33. 28. 1. 25. 3. 34. 44. 23. 9.]
 最优距离为 34986.71，运行时间为 57.49 秒。各代最优距离的下降过程如图 7 所示：

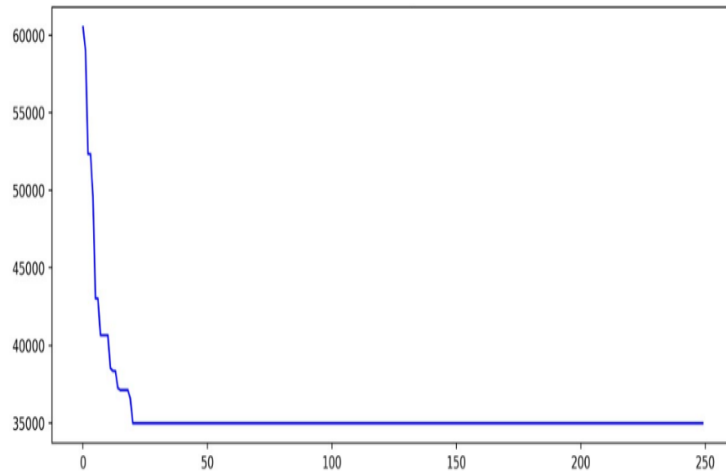


图 7 蚁群算法各次迭代最优距离下降过程

由图 7 可以发现，迭代次数在 40 次左右时基本已经收敛。将得到的最优路径绘制得图 8

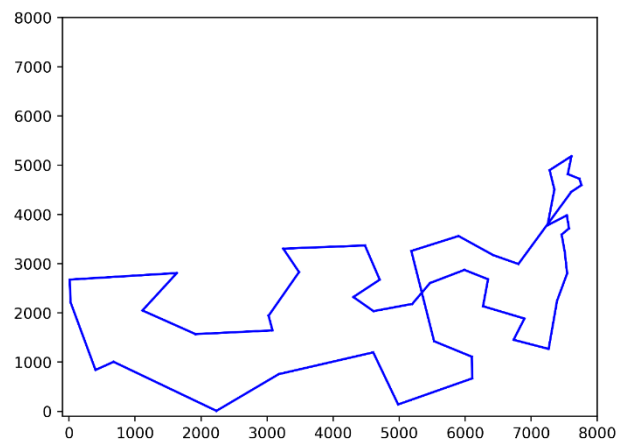


图 8 蚁群算法所求路径

从图 3 可以较为直观地观察求解路径，与图 1 进行对比，直观上可认为表现较好。接下来对具体参数进行研究，在研究一个特定参数时，其他参数的设置如表 3 所示。

首先对信息素重要因子 α 与启发函数 β 重要因子进行研究，二者代表在进行轮盘赌时概率选择是受信息素浓度影响更大还是瘦启发函数影响更大，相关因子数值越大代表影响越大。将信息素重要因子 α 初始值设置为 1，变化区间为 1 到 5，变化步长为 1；再将启发函数 β 重要因子初始值设置为 1，

变化区间为 1 到 5，变化步长为 1。得到在不同信息素重要因子及启发函数重要因子下各次迭代距离下降过程，由图 9，图 10 所示

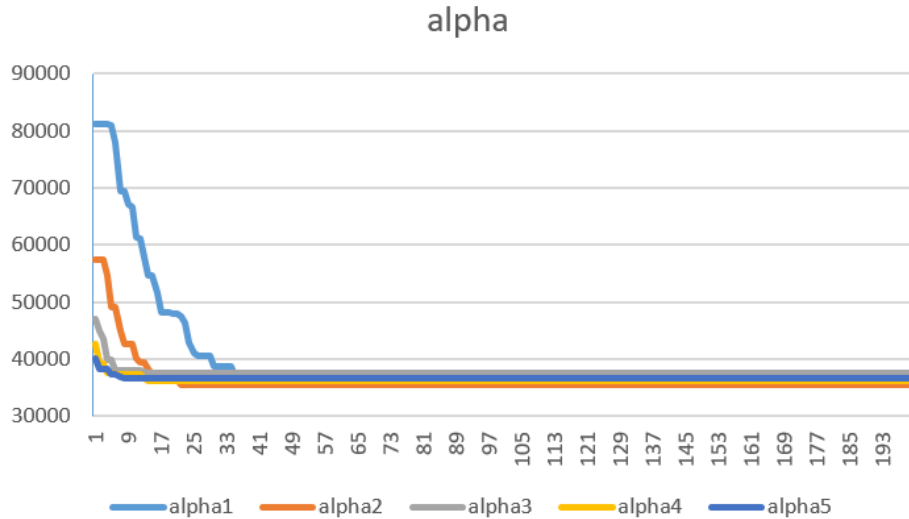


图 9 信息素重要因子对蚁群算法的影响

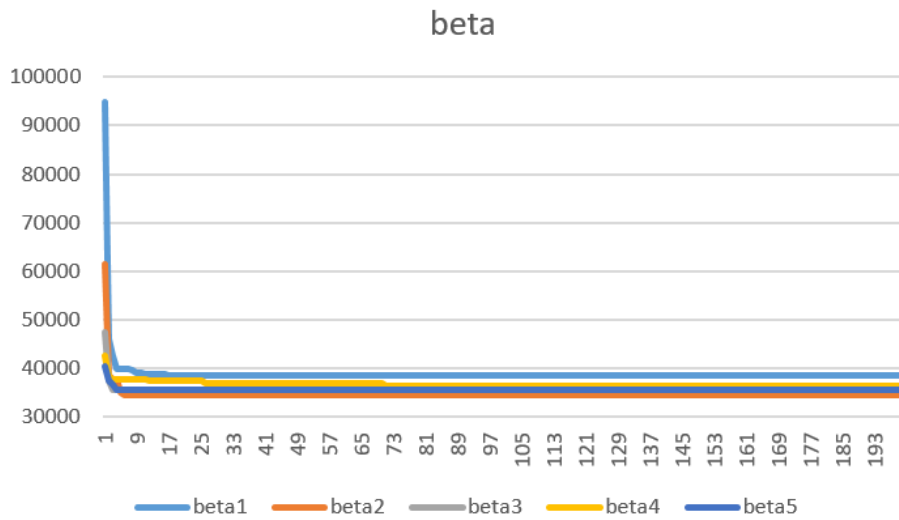


图 10 启发函数重要因子对蚁群算法的影响

由图 9 图 10 可得，固定其他参数，信息素重要因子 α 越大，初始值越小，下降速度越快；启发函数重要因子 β 越大，初始值越小，下降速度越快，由此可知，二者比例越极端，初始值越小，下降速度越快。

接下来研究信息素挥发速度对实验的影响。信息素挥发速度越大，受上一步的影响越小。将信息素挥发速度初始值设置为 0.2，变化区间为 0.2 到 0.8，变化步长为 0.2。得到在不同信息素挥发速度下各次迭代距离下降过程，由图 11 所示

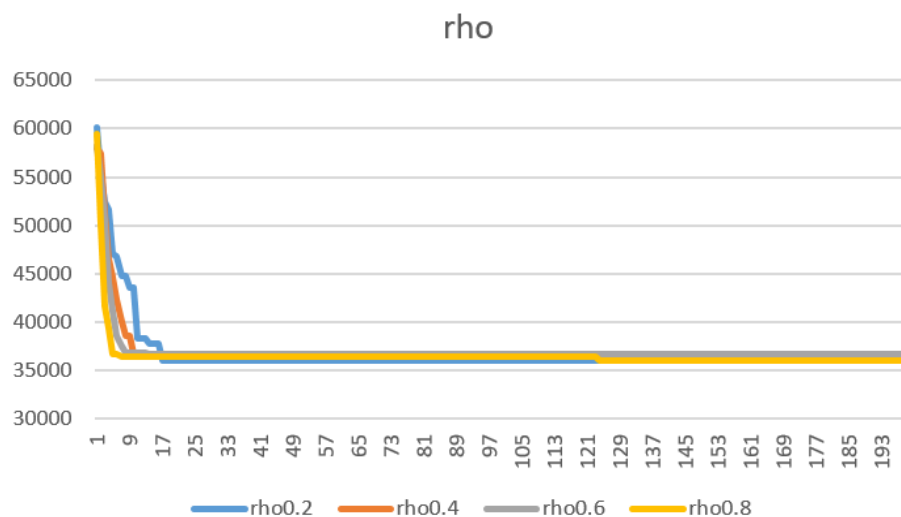


图 11 信息素挥发速度对蚁群算法的影响

由图 11 可得，信息素挥发速度越大，收敛速度越快。。

接着研究信息素参数对实验的影响。信息素参数为每次更新信息素浓度时的分子（分母为该路径长度）。将信息素参数初始值设置为 10，变化区间为 10 到 10000，每次扩大十倍。得到在不同信息素参数下各次迭代距离下降过程，由图 12 所示

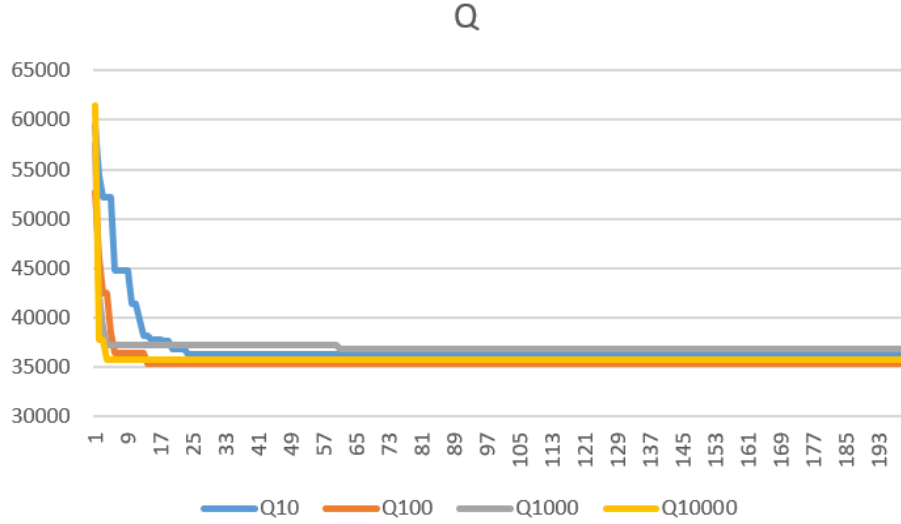


图 12 信息素参数对蚁群算法的影响

由图 12 可得，固定其他参数，信息素参数越大，距离下降速度越快，收敛速度越快。

将以上的研究结果整理，得到表 4：

表 4 蚁群算法相关参数对算法的影响

参数名称	对算法的影响
信息素重要因子	数值越大，初始值越小，下降速度越快
启发函数重要因子	数值越大，初始值越小，下降速度越快
信息素挥发速度	数值越大，下降速度越快，收敛速度越快
信息素参数	数值越大，下降速度越快，收敛速度越快

3.1.3 粒子群算法

基于粒子群算法的思路，结合 TSP 问题。粒子速度的更新公式如式 3 式 4 所示：

$$V_{id} = wV_{id} + C_1r_1(P_{id} - X_{id}) + C_2r_2(P_{gd} - X_{id}) \quad (3)$$

$$X_{id} = X_{id} + V_{id} \quad (4)$$

其中， V_{id} 表示第 i 个变量的速度的第 d 维。 w 成为惯性因子，其值为非负。 C_1 和 C_2 称为加速因子，前者为每个粒子的个体学习因子，后者为每个粒子的社会

学习因子。 r_1 和 r_2 为从 0 到 1 之间的随机数，而这可以相等。 P_{id} 表示第 i 个变量的个体极值的第 d 维， P_{gd} 表示全局最优解的第 d 维。

考虑速度的定义。速度定义采取两个思路。第一种思路为将速度定义为一个整型，代表一个粒子中可以随机变化的城市编号个数。通过实验，该方法表现效果很差，解出的最优距离一般都大于 10000。

分析原因，粒子群算法认为，每一个粒子周围应该会有较优解（除非该粒子已经是局部最优），因此定义速度，代表让该粒子向更优方向移动。但如果将速度按照第一种方式进行定义，由于城市的编号是随机编号的，进行随机交换后，难以体现出算法的“移动”思维，随机性过强，故表现较差。

第二种思路为定义速度为一个由二元组组成的列表，每一个二元组代表对路径序列中相应位置的城市编号进行交换。定义 w 为浮点数，二者相乘定义为以概率 w ，对速度中的一些二元组进行保留，丢弃其余的。定义减号为从一个序列到另一个序列需要的交换方式。加号的定义参考集合中的并运算。

同时，粒子群算法较容易陷入局部最优解。因此加入扰动因子，以一定的概率（这里定义为 0.5 并保持不变）对速度进行操作，若触发扰动，则丢弃速度列表中末尾的二元组。通过这样的方法帮助算法跳出局部最优解。

接下来进行测试，取初始参数如表 5 所示：

表 5 粒子群算法初始参数

惯性因子	粒子数	迭代次数	C_1	C_2
0.1	500	200	2	4

经过多次实验，得到实验平均最优路径为

[35, 19, 46, 12, 24, 22, 10, 11, 32, 45, 30, 37, 8, 39, 14, 20, 47, 4, 41, 9, 34, 44, 23, 31, 38, 13, 2, 15, 40, 1, 3, 25, 28, 33, 21, 0, 7, 43, 17, 29, 42, 26, 16, 18, 36, 5, 27, 6]

最优距离为 44157.71，运行时间为 1: 23.03 秒。各代最优距离的下降过程如图 7 所示：

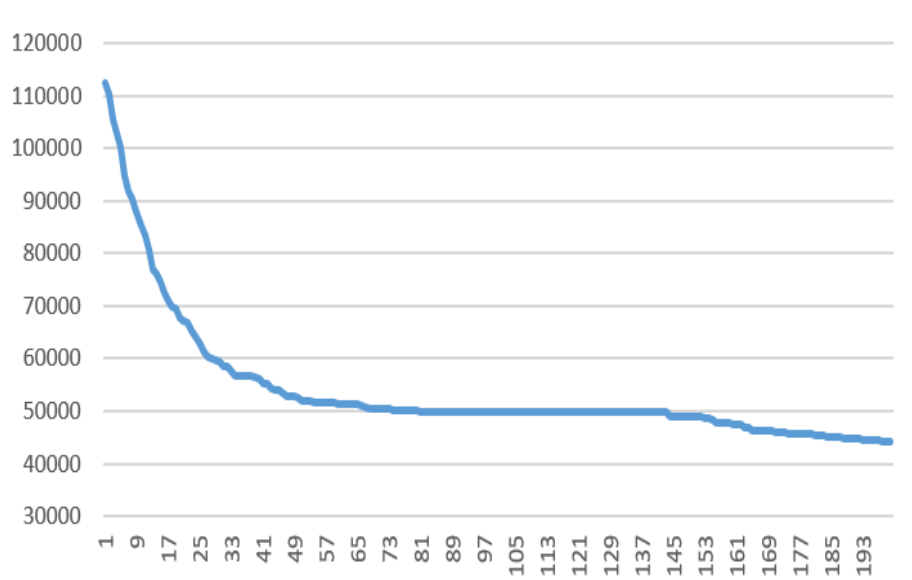


图 13 粒子群算法各次迭代最优距离下降过程

将得到的最优路径绘制得图 14

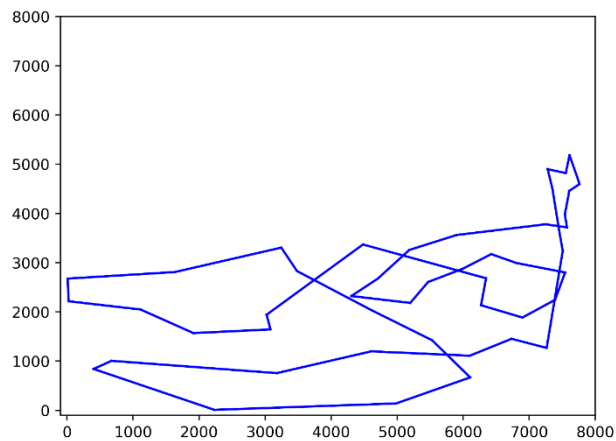


图 14 蚁群算法所求路径

从图 14 可以较为直观地观察求解路径，与图 1 进行对比，直观上可认为表现没有其他两种算法好。接下来对具体参数进行研究，在研究一个特定参数时，其他参数的设置如表 5 所示。

首先对惯性因子 w 进行研究。较大时，全局寻优能力强，局部寻优能力弱；较小时，全局寻优能力弱，局部寻优能力强，通过调整 w 的大小，可以对全局

寻优性能进行调整。将惯性因子 w 初始值设置为 0.1，变化区间为 0.1 到 0.4，变化步长为 0.1。得到在不同信息素重要因子及启发函数重要因子下各次迭代距离下降过程，由图 15 所示

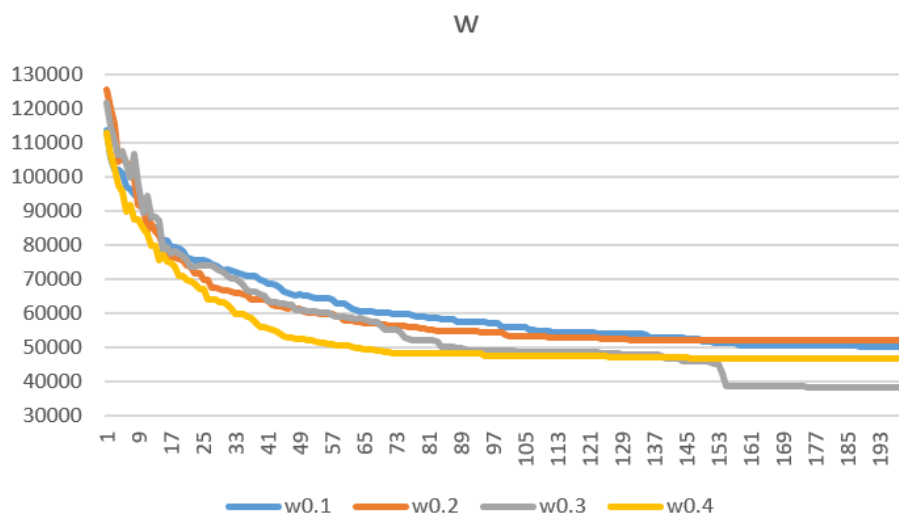


图 15 惯性因子 w 对粒子群算法的影响

由图 15 可得，固定其他参数，惯性因子 w 对算法结果无明显影响。

接下来研究 C_1 和 C_2 称为加速因子对实验的影响。前者为每个粒子的个体学习因子，后者为每个粒子的社会学习因子。考虑二者共同左右对算法进行影响，主要应考虑二者比例，因此仅变化 C_1 进行研究。将信加速因子 C_1 初始值设置为 0，变化区间为 0 到 4，变化步长为 1。得到在加速因子比例下各次迭代距离下降过程，由图 16 所示

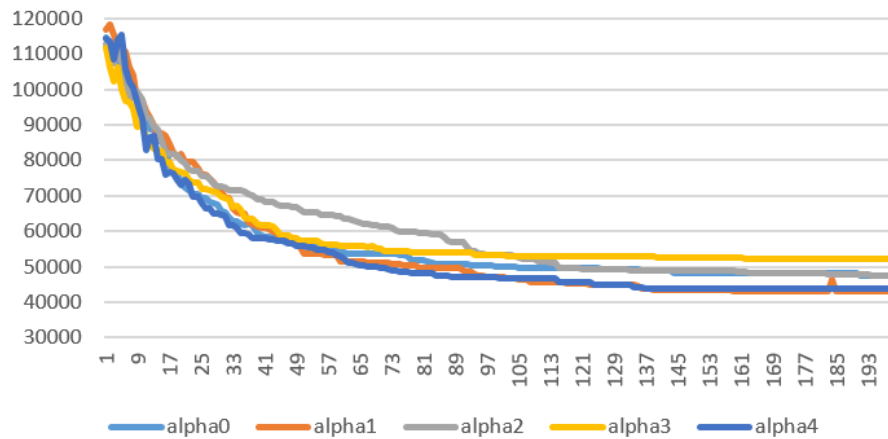


图 16 信息素参数对遗传算法的影响

由图 15 可得，固定其他参数，单纯变化加速因子比例对算法的影响不是特别明显。但当加速因子比为 1 时，表现效果一般较好。

将以上的研究结果整理，得到表 6：

表 6 蚁群算法相关参数对算法的影响

参数名称	对算法的影响
惯性因子 w	需要考虑参数之间的共同作用
加速因子比	

3.1.4 算法比较

将以上结果进行整理得到表 7（其中，运行时间将所有的算法迭代次数统一为 500 次），图 17

表 7 各算法针对 att48 数据集表现对比

算法名称	距离 (已知最优解为 33061.63)	误差	运行时间
遗传算法	36758.48	11.18%	1: 52.00
蚁群算法	34986.71	5.82%	2: 12.49
粒子群算法	44157.71	33.56%	1: 23.03

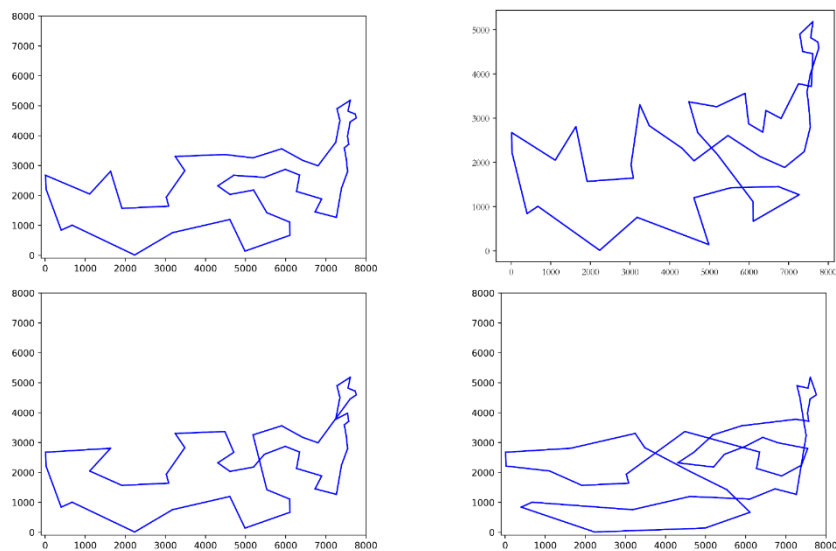


图 17 各算法针对 att48 数据集路径图
(左上已知最优解, 右上遗传算法, 左下蚁群算法, 右下粒子群算法)

由表 7 图 17 可得, 针对 att48 数据集, 蚁群算法结果与已知最优解最接近, 运行时间最长; 粒子群算法结果与已知最优解相差最大, 运行时间最短。

参考文献

1. Razali N M, Geraghty J. Genetic algorithm performance with different selection strategies in solving TSP[C]//Proceedings of the world congress on engineering. Hong Kong: International Association of Engineers, 2011, 2(1): 1-6.
2. Bai Q. Analysis of particle swarm optimization algorithm[J]. Computer and information science, 2010, 3(1): 180.
3. 甄文祥, 王文田. 遗传算法及其应用[J]. 小型微型计算机系统, 2000, 23(6):9-10.
4. 雷开友. 粒子群算法及其应用研究[D]. 西南大学, 2006.
5. <http://mnemstudio.org/particle-swarm-tsp-example-1.htm>

6. Liu B, Wang L, Jin Y, et al. An effective PSO-based memetic algorithm for TSP[M]//Intelligent computing in signal processing and pattern recognition. Springer, Berlin, Heidelberg, 2006: 1151-1156.