

混合遗传蚁群算法求解多旅行商问题

麦启深

摘 要 多旅行商问题 (Multiple Travelling Salesman Problem, MTSP) 是旅行商问题的一个变化模型, 是一种典型的组合优化问题, 可以应用于物流配送、城市环卫等行业。多旅行商问题与旅行商问题一样, 可以利用遗传算法、蚁群算法、模拟退火算法、粒子群算法等算法进行求解。针对蚁群算法在求解问题过程中对信息素的强依赖, 以及遗传算法全局搜索能力强且收敛速度慢的特点, 使用将遗传算法与蚁群算法混合的策略进行求解多旅行商问题。该策略先利用遗传算法在全局范围内搜索较优解, 以此为基础, 再利用蚁群算法进一步优化得出最优解。分别采用了遗传算法 (Genetic Algorithm, GA)、蚁群算法 (Ant Colony Algorithm, ACA) 以及混合遗传蚁群算法 (Hybrid Genetic Algorithm and Ant Colony Algorithm, GA-ACA) 对 Benchmark 基准上的 4 个不同人数的多旅行商问题进行了求解, 并记录了 10 组测试数据通过求其平均值以及标准差进行比较分析, 验证了混合遗传蚁群算法在多旅行商问题上的正确性和有效性。

关键词 遗传算法; 蚁群算法; 混合遗传蚁群算法; 多旅行商问题

Hybrid Genetic Algorithm and Ant Colony Algorithm for Multiple Travelling Salesman Problem

Qishen Mai

Abstract Multiple Travelling Salesman Problem (MTSP) is an extended model of Travelling Salesman Problem (TSP). It is a typical Combinatorial Optimization Problem (COP), which can be applied to logistics distribution, public health, and other industries. MTSP, like TSP, can be solved by intelligent algorithms such as Genetic Algorithm (GA), Ant Colony Algorithm (ACA), Simulated Annealing (SA) and Particle Swarm Optimization (PSO). Aim at ACA's strong dependence on pheromones in the process of solving MTSP, and the strong global searching ability and slow convergence speed of GA, a hybrid strategy of GA and ACA can be used to solve the MTSP. This strategy uses GA to search the optimal solution in the global scope, and then uses ACA to further optimize the optimal solution. GA, ACA and Hybrid Genetic Algorithm and Ant Colony Algorithm (GA-ACA) were used to solve 4 MTSP with different numbers of people on the Benchmark, and 10 groups of test data were recorded and compared by finding their average values and standard deviations, which verified the availability and feasibility of GA-ACA on the MTSP.

Key words Genetic Algorithm; Ant Colony Algorithm; Hybrid Genetic Algorithm and Ant Colony Algorithm; Multiple Travelling Salesman Problem

1 引言

随着现代科学研究与工程实践的进一步发展, 求解组合优化问题成为各行业企业获取更高经济效益的重要策略之一。而旅行商问题作为经典的组合优化问题, 由于其在各个领域的广泛应用, 一直备受关注与研究。自从第一个城市规模为 49 个的

旅行商问题在 1954 年被 Dantzig^[1]等人利用线性规划解决以来, 求解旅行商问题的算法发展得愈发成熟与多样。旅行商问题是指: 在 N 个城市中, 要求旅行商到达每个城市有且仅有一次, 并回到起点, 且路线长度要最短。而多旅行商问题是指: 在 N 个城市中, 有 M 个旅行商从同一个城市出发, 分别走不同路线, 使得除出发城市外的每个城市有且仅有一个旅行商经过, 且旅行商们的总路程长度要最

短。

多旅行商问题作为旅行商问题的一种拓展模型，可以在诸多实际问题中得到有效利用，例如，印刷中的调度问题、人员调度问题、校车行驶路径问题、航空路线规划问题等^[2]。张硕航^[3]等人对多旅行商模型应用及其研究进行了综合描述；王海龙^[4]等人基于遗传算法提出了矩阵解码方式求解多旅行商问题并证明了该算法的有效性；刘明^[5]等人以应急物资配送为例，提出并设计了一类新的混合遗传算法求解多旅行商问题；林东梅^[6]等人提出双层降解混合算法求解多旅行商问题并证明了该算法的有效性和高效性。

传统的遗传算法具有较强的全局搜索能力但其难以收敛；而传统的蚁群算法具有局部的搜索能力并且收敛迅速。针对以上两个特点，为更加有效地求解多旅行商问题，本文将混合遗传算法和蚁群算法用于求解多旅行商问题：先利用遗传算法得到较优解，再利用蚁群算法进行进一步优化，实现这两个算法的互补融合^[7]。对以上混合遗传蚁群算法在 Benchmark 基准中的 eil75 实例进行仿真测试，并与传统遗传算法和蚁群算法进行结果比较，证明了混合遗传蚁群算法的可行性和有效性。

2 多旅行商问题描述

本文研究的多旅行商问题可以描述为：给定 n 个城市、 m 个旅行商以及城市对应的二维坐标列表 Pos，根据城市的二维坐标，计算城市间的距离矩阵 $D_{ij} = (d_{ij})_{n \times n}$ ，每个旅行商在同一个起点出发并各自访问一定数量的城市，要求所有的城市有且仅有一次被访问，每个城市只能被其中一个旅行商访问，每个旅行商至少访问除源点外两个城市并需要返回到出发的城市。

多旅行商问题的数学模型如下：

多旅行商问题可以用有向图 $G = (V, E)$ 表示，其中 V 表示 n 个城市顶点的集合， E 表示每个城市间边的集合。该图关联一个距离矩阵 $D_{ij} = (d_{ij})_{n \times n}$ 。以点 1 作为旅行商出发的城市，称作源点，点 $2, 3, \dots, n$ 表示 m 个旅行商所需访问的城市。

定义变量

$$x_{ij} = \begin{cases} 1, & \text{边 } A(i, j) \text{ 被选中} \\ 0, & \text{否则} \end{cases}$$

d_{ij} 表示旅行商在距离矩阵中对应从城市 i 到城市 j 的距离。

则有以下模型：

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

$$\text{s. t.} \quad \sum_{j=2}^n x_{1j} = m \quad (2)$$

$$\sum_{j=2}^n x_{j1} = m \quad (3)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 2, \dots, n \quad (4)$$

$$\sum_{j=1}^n x_{ij} = 1, i = 2, \dots, n \quad (5)$$

$$x_{1i} + x_{i1} \leq 1, i = 2, \dots, n \quad (6)$$

$$u_i - u_j + (n - m) \cdot x_{ij} \leq n - m - 1, \quad (7)$$

$$2 \leq i \neq j \leq n$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (8)$$

在该模型中，式(1)表示使 m 个旅行商的旅行距离总和最小化；式(2)和(3)保证恰好有 m 个旅行商离开和回到源点；式(4)和(5)是度约束，即每个顶点只有一次入度和出度；式(6)强制每个旅行商至少要访问初源点以外的 2 个顶点；式(7)是经典的子环消除约束，防止任何不包括源点的子环形成^[8]。

3 算法描述

3.1 混合遗传蚁群算法 (GA-ACA) 描述

图 1 给出了 GA-ACA 混合算法的算法流程图，基本步骤如下：

步骤 1 遗传算法随机生成种群

以输入的城市编号作为染色体，随机初始化染色体，染色体数目为城市个数，形成个体，种群规模为 N

步骤 2 选择操作

添加强者存活率 r_{retain} 以及弱者存活率 r_{random} 进行种群筛选，增加强者的个体，剔除弱者的个体，并使种群个体总数不变

步骤 3 交叉操作

在种群中随机选择两个染色体，分为父代与母代。在父代染色体序列中随机选择一个区间交叉到母代染色体序列上对应的位置，形成子代个体

步骤 4 变异操作

随机在种群中选择一个个体，并将该个体染色体序列中随机选择两条染色体进行交换，完成变异

步骤 5 判断是否满足遗传算法结束条件

在遗传算法中设置最大迭代次数为 GA_{Iter} ，当步骤 2 到步骤 4 迭代次数达到 GA_{Iter} ，进入步骤 6；否则，返回步骤 2

步骤 6 根据遗传算法较优解初始化蚁群信息素分布

根据距离矩阵 D_{ij} 构建完全有向图模型，读取较优解中每个旅行商路径的数据，称为关键路径，并在此路径上初始化蚁群信息素 P_{path} ，以及初始化其它路径上的默认信息素 P_{ini}

步骤 7 应用 ACA 进一步优化

应用蚁群算法寻找最优路径，蚂蚁数量为 Num_{ant} ，每只蚂蚁从图 G 的初始顶点出发，遍历所有顶点，构建 MTSP 问题可行解，获取最终路径长度

步骤 8 更新信息素并判断是否满足结束条件

在蚁群算法中设置最大循环次数 C_{max} ，当循环次数到达 C_{max} 时，算法结束并输出旅行商路径图以及适应度曲线；否则，返回步骤 7

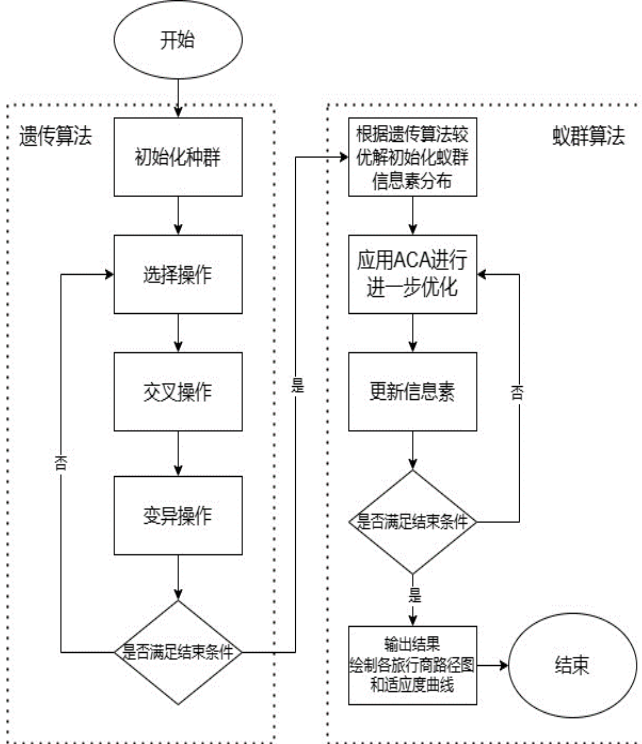


图 1 GA-ACA 算法流程图

3.2 蚁群算法信息素更新

对于蚁群算法，令图 G 中有向图各边的距离权重 $weight$ 信息素，各边初始化信息素为 $1/weight$ ，在每次迭代中会进行蒸发，紧接着是所有蚂蚁根据本次构建路径的边上更新信息素，信息素更新函数如下：

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (9)$$

其中 $\tau_{ij}(t)$ 表示时间 t 时从 i 到 j 的信息素， m 为蚂蚁个数， ρ 表示信息素蒸发率， $\Delta \tau_{ij}^k$ 是第 k 只蚂蚁在路径 i 到 j 留下的信息素。

4 实验仿真与分析

为比较和验证算法效果，笔者在 CPU 为 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz/16 GB 的计算机上，以 PyCharm 为实验环境，选取 MTSP 问题中 Benchmark 基准的 eil51 进行求解。在验证过程中，主要对蚁群算法与混合遗传-蚁群算法进行比较，参数设置如下：对于蚁群算法， $\alpha=1$ ， $\beta=1$ ， $\rho=0.03$ ，初始化信息素为 $1/weight$ ，蚂蚁数量为，循环次数为 500 次；对于混合遗传-蚁群算法，遗传算法中迭代次数为 5000 次，种群个数 $N=40$ ，变异概率 $P_m=0.1$ ，强者存活率 $r_{retain}=0.85$ ，弱者存活率 $r_{random}=0.15$ ，蚁群算法中关键路径的初始化信息素为 $10/weight$ ，其它参数保持不变。

为比较 ACA 与 GA-ACA 算法在解决 eil51 中 4 个实例的性能，获取最优值 T ，迭代过程中数据的标准差 S 与迭代次数 I ，定义最优值提升率 R_T ，标准值提升率 R_S 与迭代次数提升率 R_I 如下：

$$R_T = \frac{T_{ACA} - T_{GA-ACA}}{T_{ACA}} \times 100\%$$

$$R_I = \frac{I_{ACA} - I_{GA-ACA}}{I_{ACA}} \times 100\%$$

$$R_S = \frac{S_{ACA} - S_{GA-ACA}}{S_{ACA}} \times 100\%$$

表 1 ACA 在 ei151-m2 实例上的结果

序号	最优解	迭代次数
1	503	445
2	515	382
3	510	373
4	495	225
5	505	351
6	499	477
7	497	245
8	498	375
9	503	463
10	505	240
平均值	503	357.6

表 4 ACA 在 ei151-m7 实例上的结果

序号	最优解	迭代次数
1	685	248
2	672	264
3	688	118
4	686	403
5	694	286
6	691	381
7	682	384
8	681	365
9	679	353
10	689	406
平均值	684.7	320.8

表 2 ACA 在 ei151-m3 实例上的结果

序号	最优解	迭代次数
1	551	301
2	530	284
3	534	462
4	531	387
5	532	279
6	538	434
7	547	268
8	534	440
9	545	220
10	527	487
平均值	536.9	356.2

表 5 GA-ACA 在 ei151-m2 实例上的结果

序号	最优解	迭代次数
1	498	447
2	497	404
3	509	376
4	508	399
5	505	393
6	486	191
7	515	224
8	500	308
9	493	259
10	502	277
平均值	501.3	327.8

表 3 ACA 在 ei151-m5 实例上的结果

序号	最优解	迭代次数
1	608	171
2	609	306
3	610	312
4	611	460
5	605	466
6	592	422
7	612	472
8	593	342
9	603	306
10	612	407
平均值	605.5	366.4

表 6 GA-ACA 在 ei151-m3 实例上的结果

序号	最优解	迭代次数
1	533	247
2	551	249
3	529	374
4	524	246
5	532	464
6	542	359
7	539	329
8	529	370
9	544	386
10	522	268
平均值	534.5	329.2

表 7 GA-ACA 在 eil151-m5 实例上的结果

序号	最优解	迭代次数
1	611	237
2	608	441
3	605	236
4	608	457
5	595	428
6	597	381
7	591	199
8	590	448
9	607	422
10	610	307
平均值	602.2	355.6

表 8 GA-ACA 在 eil151-m7 实例上的结果

序号	最优解	迭代次数
1	665	318
2	688	329
3	685	410
4	674	239
5	684	388
6	690	278
7	674	330
8	671	198
9	681	233
10	684	325
平均值	679.6	304.8

表 9 GA-ACA 与 ACA 对比

实例	R_T (%)	R_I (%)	迭代次数的 R_S (%)
eil151-m2	0.34%	8.33%	6.21%
eil151-m3	0.45%	7.58%	22.47%
eil151-m5	0.55%	2.95%	-4.71%
eil151-m7	0.74%	4.99%	25.94%
平均值	0.52%	5.96%	12.48%

表 1 到表 8 记录了 ACA 与 GA-ACA 运行 eil151 上的 4 个实例的结果, 每个算法的每个实例各记录了 10 组数据。表 9 记录的是 GA-ACA 相较于 ACA 算法的最优值、迭代次数以及迭代次数的标准差提升率的比较。

5 结语

在表 9 中, 可以看出在最优解方面, GA-ACA 相较于传统 ACA 有些许提升; 在迭代次数方面, GA-ACA 具有一定优势, 可以使得最优值更快速稳定地收敛, 验证了 GA-ACA 的有效性和可行性。

本文中的 GA-ACA 算法仍有改进之处, 可以优化 ACA 中的参数以及信息素的策略, 并且期望可以利用更多 MTSP 数据集进行进一步的实验分析, 得出更加令人满意的结论。

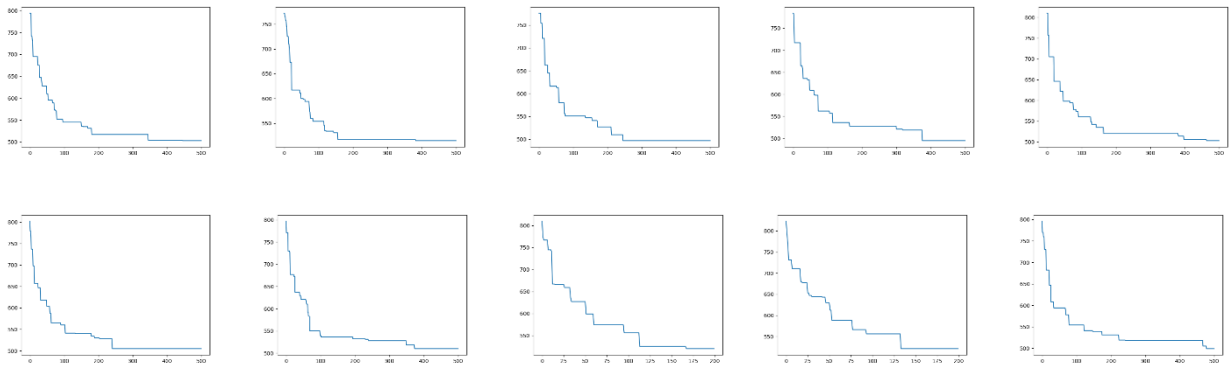
参 考 文 献

- [1] RABINER L. Book reviews:combinatorial optimization:algorithms and complexity[J]. IEEE ASSP Magazine, 1984,1(4):22
- [2] 王壮志,汪定伟,闫杨.一类多旅行商问题的计算及仿真分析[J].系统仿真学报,2009,21(20):6378-6381.
- [3] 张硕航,郭改枝.多旅行商模型及其应用研究综述[J].计算机科学与探索,2022,16(07):1516-1528.
- [4] 王海龙,周辉仁,郑丕谔等.基于遗传算法的多旅行商问题研究[J].计算机应用研究,2009,26(05):1726-1728+1732.
- [5] 刘明,张培勇.求解多旅行商问题的新混合遗传算法:以应急物资配送为例[J].系统管理学报,2014,23(02):247-254.
- [6] 林冬梅,王东,李娅.一种求解多旅行商问题双层降解混合算法[J].计算机应用研究,2011,28(08):2876-2879.
- [7] 肖宏峰,谭冠政.基于遗传算法的混合蚁群算法[J].计算机工程与应用,2008(16):42-45+134..
- [8] Bektas, T.: The multiple traveling salesman problem: an overview of for-mulations and solution procedures. Omega, 34(3) (2006), pp. 209–219.

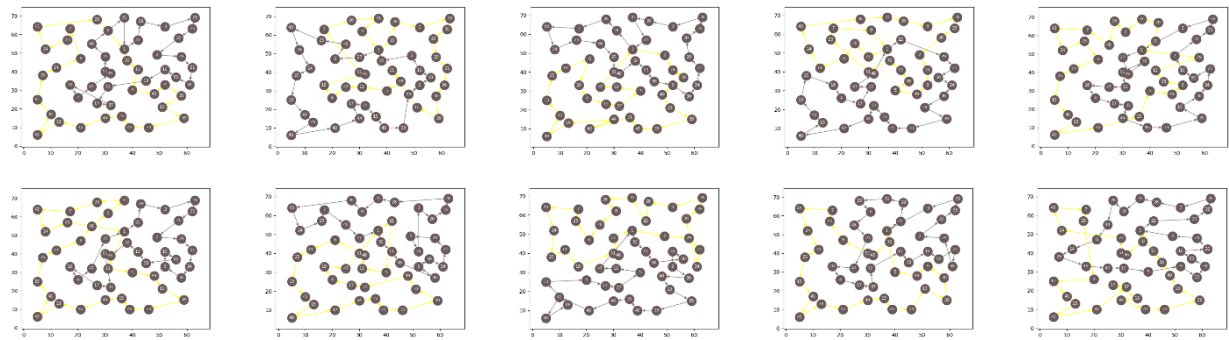
附录 1. 两种算法运行结果，以ei151-m2 为例：

ACA 在 ei151-m2 中的结果：

适应度曲线：

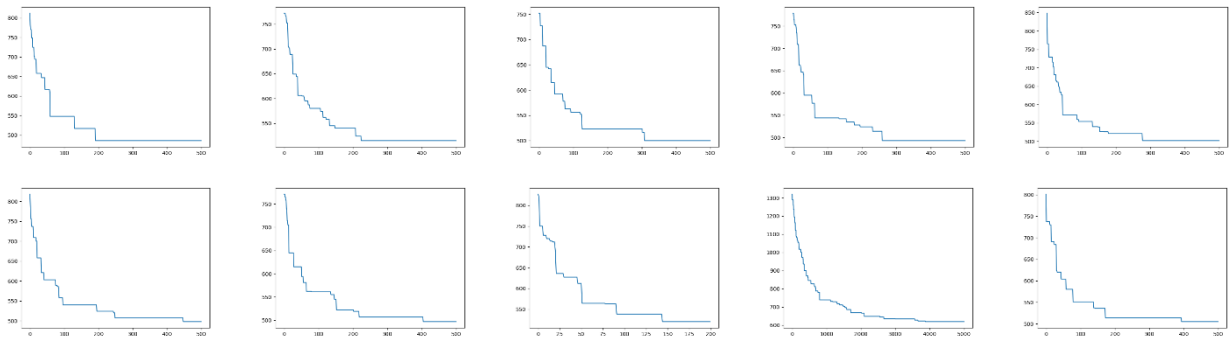


路径图：



GA-ACA在ei151-m2 中的结果：

适应度曲线：



路径图：

