

求解旅行商问题的几种算法的比较研究

李 敏,吴 浪,张开碧
(重庆邮电大学 自动化学院,重庆 400065)

摘 要:旅行商问题具有重要的理论和实际研究价值,在工程实践中应用广泛。采用遗传算法、蚁群算法和模拟退火算法对旅行商问题进行求解,并选取中国旅行商问题进行仿真,比较了 3 种算法的优劣,得出了它们各自不同的适用范围:蚁群算法适用于缓慢地较精确的求解场合;模拟退火算法适用于快速精确的求解;遗传算法适用于快速求解,但结果准备度要求不高的情况。
关键词:旅行商问题;遗传算法;蚁群算法;模拟退火算法;中国旅行商问题
中图分类号:TP393 **文献标识码:**A **文章编号:**1673-825X(2008)05-0624-03

Comparative study of several algorithms for traveling salesman problem

LI Min, WU Lang, ZHANG Kai-bi
(College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, P. R. China)

Abstract:Traveling salesman problem (TSP) is of important theoretical and practical significance and applied widely in engineering practice. The genetic algorithm, ant colony algorithm and simulated annealing were adopted to solve the traveling salesman problem, and the Chinese traveling salesman problem was chosen to simulate. Through the comparison of these three algorithms' advantages and disadvantages, their different applications were gained: the ant colony algorithm is suitable for slow and accurate solving, the simulated annealing applies to quick and accurate solving, and the genetic algorithm is for quick but low accurate solving.
Key words:traveling salesman problem (TSP); genetic algorithm; ant colony algorithm; simulated annealing; Chinese traveling salesman problem

0 引 言

TSP 问题(traveling salesman problem, TSP)是一种典型的组合优化问题,已经被证明是一个 NP-complete(non-deterministic poly-nominal complete)难题。经典的 TSP 的描述为:给定 n 个城市和两两城市之间的距离,有一个旅行商从某一城市出发,要求确定一条经过各城市当且仅当一次的最短路线。
TSP 问题易于陈述但难于求解。目前求解旅行商问题的方法可分为 2 大类:精确算法和近似算法。常用的精确求解方法主要有分枝定界法、线形规划法和动态规划法,其计算时间呈指数复杂度,难以解决大规模问题。近似算法又分为巡回路径构造算法、巡回路径优化算法和智能算法,其算法简单,计

算量小,大多数情况下求得的满意解能满足要求。而其中,智能算法的研究最为引人注目,如模拟退火、禁忌搜索、粒子群优化算法、遗传算法、免疫算法、神经网络、蚁群算法及其改进算法等^[1-3],它们给 TSP 问题的解决提供了新的途径。本文中我们采用遗传算法、蚁群算法和模拟退火算法对旅行商问题进行求解,并选取中国旅行商问题进行仿真,根据仿真结果比较了上述几种算法的优缺点,得出了不同算法的适用范围。

1 不同算法求解 TSP

如果用图论来描述 TSP^[4],那就是已知赋权图 $G = (C, L)$,找出总权值最小的哈密顿圈。其中 $C = \{c_1, c_2, \dots, c_n\}$ 为顶点集,这里表示 n 个城市的集合; $L = \{l_{ij} | c_i, c_j \in C\}$ 为各顶点相互连接组成的边集,这里是集合 C 中元素(城市)两两连接的集

合。每一条边 l_{ij} 都存在与之对应的权值 d_{ij} , 构成 $D = (d_{ij})$ 矩阵, 实际应用中 d_{ij} 可以表示距离、费用、时间、油量等。

对于对称旅行商问题, $d_{ij} = d_{ji}, \forall i, j = 1, 2, 3, \dots, n$ 。若对于城市 $C = \{c_1, c_2, \dots, c_n\}$ 的一个访问顺序为 $T = \{t_1, t_2, t_3, \dots, t_i, \dots, t_n\}$, 其中 $t_i \in C (i = 1, 2, 3, \dots, n)$, 且记 $t_{n+1} = t_1$ 。则旅行商问题的数学模型为

$$\min L = \sum_{i=1}^n d_{t_i, t_{i+1}}$$

1.1 遗传算法求解 TSP

遗传算法 (genetic algorithm, GA) 是一种基于生物自然选择与遗传机理的随机搜索算法^[5]。一般地, 遗传算法以一群随机产生的可行解开始, 每个解用一串编码表示为个体, 由优化目标函数确定个体的适应度对个体进行评价。通过交叉、变异等遗传算子的操作对种群进行组合产生下一代个体, 逐步向优化的种群进化。

遗传算法求解 TSP 问题的基本步骤如下:

- 1) 在搜索空间中随机产生初始种群;
- 2) 计算每个个体的目标函数, 经调整得到相应的适应度值 (路线长度);
- 3) 通过遗传算子操作产生下一代个体;
- 4) 在下一代个体的基础上回到步骤 2), 记录下每代的最好个体, 直至达到最大迭代次数;
- 5) 输出最好个体, 终止整个程序。

1.2 蚁群算法求解 TSP

蚁群算法^[6] (ant colony algorithm, ACA) 是一种全新的通过模拟自然界蚂蚁寻径的行为的进化算法。蚂蚁群找到食物时, 它们总能找到一条从食物到巢穴之间的最优路径。这是因为蚂蚁在寻找路径时会在路径上释放出一种特殊的信息素。当它们碰到一个还没有走过的路口时, 就随机地挑选一条路径前行。与此同时释放出与路线长度有关的信息素。路径越长, 释放的激素浓度越低。当后来的蚂蚁再次碰到这个路口的时候, 选择激素浓度较高路径概率就会相对较大。这样形成了一个正反馈。最优路径上的激素浓度越来越大, 而其它的路径上激素浓度却会随着时间的流逝而消减。这样, 整个蚁群最终会找出最优路径。

蚁群算法求解 TSP 问题的基本步骤如下。

- 1) 令 $nc = 0$ (nc 为迭代步数或搜索次数); 每条边上的 $\tau_{ij}(0) = c$ (常数), 并且 $\Delta\tau_{ij} = 0$; 放置 m 个

蚂蚁到 n 个城市上。

- 2) 将各蚂蚁的初始出发点置于当前解集 $tabu_k$ 中; 对每个蚂蚁 $k (k = 1, 2, \dots, m)$, 按概率 P_{ij}^k 移至下一城市 j ; 将城市 j 置于 $tabu_k$ 中。

- 3) 经过 n 个时刻, 蚂蚁 k 可走完所有的城市, 完成一次循环。计算每个蚂蚁走过的总路线长度 L_k , 更新找到的最短路径。

- 4) 更新每条边上的信息量 $\tau_{ij}(t + n)$ 。

- 5) 对于每一条边, $\Delta\tau_{ij} = 0; nc = nc + 1$ 。

- 6) 若 $nc <$ 预定的迭代次数 $ncmax$, 则转步骤 2); 否则, 打印出最短路径, 终止整个程序。

1.3 模拟退火算法求解 TSP

模拟退火算法 (simulated annealing, SA) 是一种模拟熔化状态下物体由逐渐冷却至最终达到结晶状态这一物理过程的近似全局优化算法^[7]。它利用所求解问题的求解过程与熔化物体退火过程的相似性, 采用随机模拟物体退火过程来完成问题的求解, 即在控制参数 (温度) 的作用下对参数的值进行调整, 直到找到能使能量函数达到全局极小值的参数。为了在退火过程中不丢失最低内能状态, 降温时初始温度要足够高, 下降速率要小。

模拟退火算法求解 TSP 问题的基本步骤如下:

- 1) 选取一个起始点 z , 并设一个较高的起始温度 T , 令迭代次数 k 等于 1;
- 2) 求目标函数 (能量函数) $E = f(x)$ 的函数值;
- 3) 按照由生成函数 $g(\Delta x, T)$ 确定的概率选择 Δx , 令新点 $x_N = x + \Delta x$;
- 4) 计算新的目标函数值 $E_N = f(x_N)$;
- 5) 按照由接收函数 $h(\Delta E, T)$ 确定的概率将 x 设为 x_N , 其中, $\Delta E = E_N - E$;
- 6) 按照退火时间表降低温度 T ;
- 7) 增加迭代次数 k , 如果 k 达到最大迭代次数, 停止迭代; 否则返回步骤 3)。

2 算法仿真及结果比较

为了显示各算法的适用范围和各自的优缺点, 选取中国旅行商问题 (chinese traveling salesman problem, 简称 CTSP 或 C-TSP) 对以上 3 种算法进行仿真。C-TSP 是指在中国 31 个主要城市 (包括省会城市、自治区首府和直辖市) 之间进行旅行 (没有考虑重庆、香港和澳门), 找出一条最短的经过每个城市仅一次的且回到起点的路径。

2.1 测试环境及仿真结果

本次测试采用 Matlab6.5 为编程工具, 在 CPU 为

Intel Pentium4 2.93 GHz,内存为 256 Mbit,操作系统为 Windows XP(SP2)的计算机上对 C-TSP 求解。

(%)”按 $\frac{\text{平均}-\text{最优}}{\text{最优}} \times 100\%$ 计算,其中“最优”是指曾经得到过的最优近似解 15 379 km^[8]。

3 种算法求解 C-TSP 的结果见表 1。“平均

表 1 测试结果比较

Tab. 1 Comparison of testing results

算法	最短路线长度/km	平均路径长度/km	方差	程序平均运行时间/s	平均/%
遗传算法	16 369	18 216	936 910	20.44	18.45
蚁群算法	15 602	15 648	14 822	69.82	1.75
模拟退火算法	15 567	16 981	683 180	12.38	10.42

2.2 结果分析

如表 1 所示,首先,比较 3 种算法的运行时间。遗传算法的程序平均执行时间 20.44 s;蚁群算法为 69.82 s;模拟退火算法为 12.38 s。容易看出,模拟退火算法运行速度最快,蚁群算法运行速度最慢。其次,比较 3 种算法的准确度。遗传算法求得平均路线长度 18 216 km,平均(%) 18.45%;蚁群算法为 15 648 km,平均(%) 1.75%;模拟退火算法为 16 981 km,平均(%) 10.42%。就解的分布而言,蚁群算法的结果分布最集中,均在目前最优解所在区间[15 000,16 000],如图 1 所示。

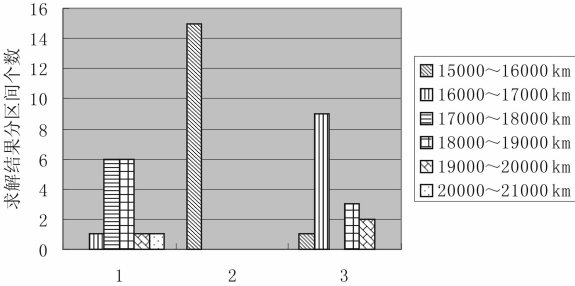


图 1 3 种算法得到的结果分区间分布

Fig. 1 Results distribution of three algorithms

最后,比较 3 种算法的巡回路线与目前最好结果的相似程度,如图 2 所示:遗传算法的巡回路线除

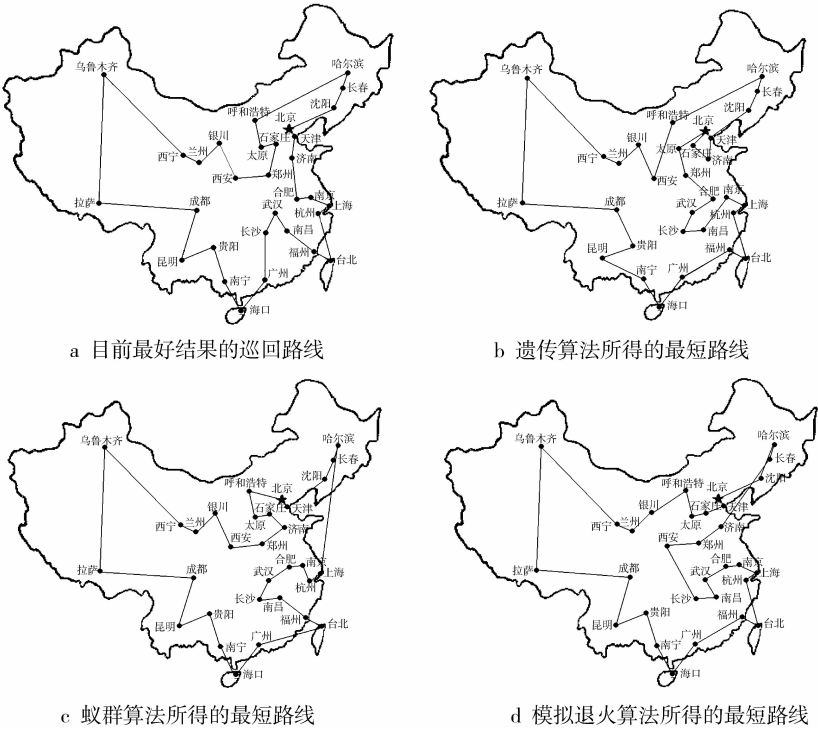


图 2 目前最好结果的巡回路线与 3 种算法得到的最短路线

Fig. 2 Best itinerant route currently and the shortest routes obtained from three algorithms