

# 蚁群算法在旅行商问题中的应用

◆ 苑侗侗 王雪阳

**摘要:** 本文以旅行商问题为研究对象,介绍了蚁群算法实现的具体步骤,并以遗传算法、模拟退火算法为比较算法,分析了三种算法求解旅行商问题时的性能,实验证明:蚁群算法在计算小规模旅行商问题时,其性能优于遗传算法和模拟退火算法;在计算大规模旅行商问题时,遗传算法性能优于蚁群算法和模拟退火算法。

**关键词:** 蚁群算法;旅行商问题;性能

## 引言

蚁群算法作为一种新型群智能算法,最早应用于旅行商问题求解中。该算法是根据蚂蚁寻找食物过程中,通过释放信息素等方式进行相互协作,以寻求最短路径为启示而产生的算法。目前,蚁群算法已广泛在各个领域得到应用,一般多用于求解复杂的组合型难题。

本文以旅行商问题为研究背景,该问题旨在求解最短路径。其中最短路径需满足单一旅行者由一个城市出发,经过所有指定的城市,且每个城市只经过一次,最终回到出发点的要求。文章使用蚁群算法(ant colony algorithm,ACA)对旅行商问题进行求解,并用模拟退火算法(simulated annealing,SA)、遗传算法(genetic algorithm,GA)作为对比算法对ACA、SA、GA的性能进行了比较。

## 一、蚁群算法实现

本节详细阐述了实现旅行商问题的蚁群算法,具体步骤如下:

Step1: 参数初始化。初始化变量M、 $\rho$ 、city、visit等,其中M表示旅行商问题规模, $\rho$ 表示信息素的挥发因子,city中存放需遍历的所有城市,visit是空表。

Step2: 根据信息素浓度为每只蚂蚁从city表中选择将要移动到的下一个城市,移动结束后,将该城市添加到visit表,同时在city表中删除该城市。重复M-1次,直到所有的城市都被遍历过一次。

Step3: 更新信息素浓度。两个城市之间的信息素浓度随着时间的增长产生一个迭代增量 $\Delta\tau_{ij}$ 。若t时刻信息素浓度为 $\tau_{ij}(t)$ ,则t+n时刻信息素浓度 $\tau_{ij}(t+n)=\rho\tau_{ij}(t)+\Delta\tau_{ij}$ 。

Step4: 检查终止条件。若达到规定的最大迭代值,转向Step5,否则,重复Step1,2,3,4。

Step5: 输出最优解。

## 二、三种算法性能比较

本节使用ACA、SA、GA求解旅行商问题,并比较三种算法性能的优劣,仿真实验从通用TSPLB中选取Oliver30、Eil51及CH130。使用三种算法计算旅行商问题的最优值(Best)、最坏值(Worst)以及迭代次数(N),其中迭代次数最大值为500,结果如表1所示:

表1 ACA、SA、GA结果对比表

M	Compare	ACA	SA	GA
30	Best	425.65	429.38	439.36
	Worst	430.82	435.718	472.475
	N	88	175	213
51	Best	450.98	439.33	467.98
	Worst	455.82	444.61	490.16
	N	123	172	259
130	Best	6417.9	6350.44	7118.26
	Worst	6430.9	6471.04	7322.39
	N	151	266	432

分析表1可得,当城市数量为30时,ACA得到的最优值较好,并且迭代次数最少;当城市数量为50时,SA得到的最优值较好,但迭代次数比ACA所需迭代次数大。由此可见,在求解城市数量较少的旅行商问题时,使用ACA更有优势,当城市数量增多时,使用SA得到的结果较好。

## 三、结论

蚁群算法在解决组合型问题时,有其自身优势,但其优势同时受其他因素影响,如所求解问题的规模等,还有待进一步研究探究。

## 参考文献

[1] 李成兵,郭瑞雪,等.改进蚁群算法在旅行商问题中的应用[J].计算机应用,2014,34(S1):131-132,165.

(作者单位:河北农业大学信息科学与技术学院)