

一种基于精英策略的改进蚁群算法及应用^①

张家善^{1,2}, 王志宏¹, 陈应显¹

¹(辽宁工程技术大学 工商管理学院, 葫芦岛 125000)

²(湛江师范学院 商学院, 湛江 524048)

摘 要: 针对基本蚁群算法存在求解速度慢, 容易出现“早熟”, 导致搜索停滞的缺点, 将遗传算法中排序的概念扩展到精英机制当中, 以一种新的加权方法进行信息素更新, 建立了改进蚁群算法模型. 对 30 城市物流配送问题仿真结果表明: 改进算法的求解速度和求解精确度都明显优于基本蚁群算法.

关键词: 蚁群算法; 搜索停滞; 精英策略; 排序; 物流配送

Improved Ant Colony Algorithm Based on Elitist Strategy and Application

ZHANG Jia-Shan^{1,2}, WANG Zhi-Hong¹, CHEN Ying-Xian¹

¹(College of Business Administration, Liaoning Technical University, Huludao 125000, China)

²(School of Business, Zhanjiang Normal University, Zhanjiang 524048, China)

Abstract: The basic ant colony algorithm converges slowly, is prone to plunge into partial optimum and results in search stagnation. In this paper, the optimized compositor is introduced into elitist strategy. According to ant compositor, new weighted coefficient is designed for pheromone updating. The simulation results of 30-city logistic distribution show that improved algorithm is more efficient and accurate than the basic ant colony algorithm.

Key words: ant colony algorithm(ACA); search stagnation; elitist strategy; compositor; logistic distribution

1 引言

蚁群算法(Ant Colony Algorithm, ACA)^[1]又称作蚂蚁算法, 是由意大利的 Marco Dorigo 于 1992 年在他的博士论文中首先提出, 其基本思想是模仿蚂蚁依赖信息素进行通信而显示出的社会行为, 在 Agent 总体的基础上, 由一个贪心算法指导下的自催化过程引导每个 Agent 的行动. 该算法模拟了蚂蚁觅食时的行为, 按照启发式思想, 通过信息素的诱导作用, 逐步收敛到问题的最优解. 其主要特点就是: 通过正反馈、分布式协作来寻找最优路径. 目前, 蚁群算法已经广泛应用于旅行商问题、二次分配问题、作业车间调度问题、物流配送问题, 成为解决组合优化问题最有效的算法之一.

基本蚁群算法复杂度较高, 一般需要较长的搜索时间, 而且容易出现搜索停滞现象(Stagnation Behaviour), 即搜索进行到一定程度后, 所有个体所发

现的解完全一致, 不能对解空间进一步进行搜索, 不利于发现更好的解. 围绕该问题, 学者们发表了不少文献, 如文献[3]在路径选择时提出了一种不再考虑距离因素, 仅考虑信息素强度的方法; 文献[4]通过在局部搜索中采用聚类进行二次搜索, 取得了较好求解效果; 文献[5]引入精英策略, 即在每次循环之后给予最优解以额外的信息素量, 以使到目前为止所找出的最优解在下一循环中对蚂蚁更有吸引力. 文献[6]基于蚂蚁分类, 提出了一种改进精英策略算法. 实验表明, 上述算法在一定程度上拓展了解的搜索空间, 提高了解的质量. 但是依然存在问题, 或者搜索速度慢, 或者虽然搜索快但却容易陷入局部最优. 本文基于精英策略并将遗传算法中排序的概念扩展到精英机制当中, 对蚁群算法进行改进, 以在较短时间内获得问题的满意最优解.

^① 基金项目: 国家自然科学基金(50904032)

收稿时间: 2012-02-16; 收到修改稿时间: 2012-03-14

2 基本蚁群算法^[1,2,8]

假设有 m 只蚂蚁放入到 n 个随机选择的需求节点中, 每一只蚂蚁将根据线路上的信息浓度选择下一个它还没有访问的节点; 同时在完成一步(从一个节点到达另外一个节点)或者一个循环(完成对所有 n 个节点的访问)后, 更新所有路径上的残留信息浓度. m 是蚁群中蚂蚁的数量, d_{ij} 是城市 i 到城市 j 之间的距离, η_{ij} 是边 (i, j) 的能见度, $\eta_{ij}=1/d_{ij}$, 反映由城市 i 转移到城市 j 的启发程度, τ_{ij} 是边 (i, j) 上的信息素轨迹强度, $\Delta\tau_{ij}^k$ 是蚂蚁 k 在边 (i, j) 上留下的单位长度轨迹信息素量, p_{ij}^k 是蚂蚁 k 从城市 i 转移到城市 j 的状态转移概率, j 是尚未访问的城市, 则状态转移概率 p_{ij}^k

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (1)$$

式中, $allowed_k = \{0, 1, \dots, n-1\}$ 表示蚂蚁 k 下一步允许选择的节点. α 和 β 为两个参数, 分别反映了蚂蚁在运动过程中积累的信息和启发信息在蚂蚁选择路径中的相对重要性^[7]. 为每只蚂蚁设计一个禁忌表 $tabu_k(k=1, 2, \dots, m)$, 记录在 t 时刻蚂蚁 k 已走过城市, 不允许该蚂蚁在本次循环中重复经过. 本次循环结束后禁忌表被清空.

当所有蚂蚁完成一次循环后, 根据各蚂蚁遍历的好坏(适应度值), 计算信息素增量, 更新相关路径上的信息素, 更新规则如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (2)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (3)$$

式中 ρ 为信息素轨迹的衰减系数, $\Delta\tau_{ij}^k(t, t+1)$ 表示第 k 只蚂蚁在 $(t, t+1)$ 时刻留在 (i, j) 路段上的信息素, $\Delta\tau_{ij}(t, t+1)$ 表示本次循环中路段 (i, j) 的信息素增量.

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & (i, j) \in l_k \\ 0, & 否则 \end{cases} \quad (4)$$

其中, Q 表示初始信息素量, 为常数, 其值由实验确定. L_k 表示第 k 只蚂蚁在本次循环中所走路线的费用值(可以表示为长度).

3 算法改进

1) 状态转移概率公式的改进

本文引入一个新的常量: $q_0 \in (0, 1)$, 蚂蚁 k 在每次选择路径之前, 先随机产生 $q_0 \in (0, 1)$, 蚂蚁 k 将根据下式选择路径 j :

$$j = \begin{cases} \arg \max_{j \in allowed_k} [\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)], & q < q_0; \\ \text{式}(1), & q \geq q_0; \end{cases} \quad (5)$$

式中, 当 $q \geq q_0$ 时, 该式表示基本 ACA 中的探索性搜索; 当 $q < q_0$ 时, 该式表示确定性搜索, 即从已得的结果中, 找出概率最大的路径作为选择. 确定性搜索利用了已得的经验知识来指导路径选择, 弥补了探索性搜索在收敛速度上受限制的缺陷.

2) 信息素更新的改进

精英策略(Elitist Strategy)的基本思想是: 在每次循环结束后即对所有已经发现的最优解给予额外的信息素增强, 其目的是为了使到目前为止所找出的最优解在下次循环中对蚂蚁更具有吸引力^[5]. 信息素根据下式进行更新:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^* \quad (6)$$

其中,

$$\Delta\tau_{ij}^* = \begin{cases} \sigma \frac{Q}{L^*} & \text{如果边 } (i, j) \text{ 是所} \\ & \text{找出最优解的一部分} \\ 0 & \text{否则} \end{cases}$$

$\Delta\tau_{ij}^*$ 表示精英蚂蚁引起的路径 (i, j) 上的信息素量的增加, σ 是精英蚂蚁的个数, L^* 为所找出的最优解的路径长度.

文献[5]认为所有经过边 (i, j) 蚂蚁对该边信息素更新的贡献是相同的, $\Delta\tau_{ij}$ 计算采用(4)式. 实验表明, 使用精英策略可以使蚂蚁系统经过较少的进化代数就能找出更优的解. 但是使用精英策略后, 解元素之间的差异减小了, 导致选择概率的差异也随之减小, 使得搜索过程集中在到目前为止所找出的最优解附近, 从而阻止了对更优解的进一步搜索.

为了维持选择压力, 本文将蚂蚁按路径长度排序, 认为蚂蚁对信息素更新的贡献与该蚂蚁所找到路径的质量有关, 路径越优(短), 对信息素更新的贡献越大; 路径越长, 对信息素更新的贡献越小. $\Delta\tau_{ij}$ 的确定按如下规则进行:

当所有蚂蚁都完成一次遍历后, 蚂蚁按路径长度排序($L_1 \leq L_2 \leq \dots \leq L_m$), 蚂蚁对激素轨迹量更新的贡献根据该蚂蚁的排名位次(记为 μ)进行加权, 只考虑前 $m/3$ 只蚂蚁.

在排列中, 前 $m/3$ 只蚂蚁中的一只所经过的边获得一定量的信息素, 其量正比于该蚂蚁的排名位次, 其权系数为 $\frac{L_{av} - L_\mu}{L_{av} - L^*}$. 信息素增加按下式进行:

$$\Delta\tau_{ij} = \sum_{\mu=1}^{m/3} \Delta\tau_{ij}^\mu \tag{7}$$
$$\Delta\tau_{ij}^\mu = \begin{cases} \frac{L_{av} - L_\mu}{L_{av} - L^*} \cdot \frac{Q}{L_\mu} & \text{如果第}\mu\text{只最好} \\ & \text{蚂蚁经过路径}(i, j) \\ 0 & \text{否则} \end{cases}$$

其中, μ 为蚂蚁排列的顺序号; $\Delta\tau_{ij}^\mu$ 表示由第 μ 只最好蚂蚁引起的路径(i, j)上的信息素量的增加; L_{av} 为本次循环平均路径长度, L_μ 是第 μ 只最优蚂蚁的路径长度.

此外, 到目前为止找出最优路径蚂蚁所经过的边也将获得额外的激素量(相当于带精英策略蚁群算法中精英蚂蚁的激素更新).

3)信息素设置上下界

本文在信息素更新过程中引入 MMAS 的思想, 对信息素实施上下界限制. 设信息素最小值和最大值分别 $\tau_{\min}$ 和 $\tau_{\max}$, 从而使得对所有信息素 $\tau_{ij}(t)$, 有 $\tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}$. 在进行信息素更新时, 若发现有

$\tau_{ij}(t) > \tau_{\max}$, 则设置 $\tau_{ij}(t) = \tau_{\max}$; 若 $\tau_{ij}(t) < \tau_{\min}$, 则设置 $\tau_{ij}(t) = \tau_{\min}$. 这样可以有效地避免某条路径上的信息量远大于其余路径, 使得所有的蚂蚁都集中到同一条路径上, 从而阻止了对解空间的进一步探索.

改进后算法流程如图 1 所示.

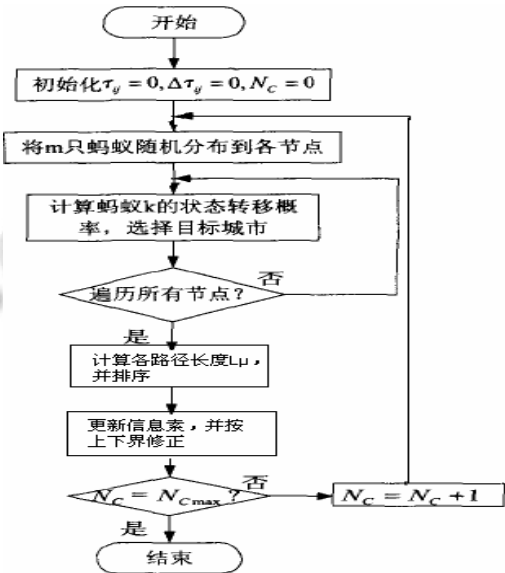


图 1 改进后算法流程

4 应用仿真

选取 VRP 测试库中的 Eil30 问题做仿真: 已知车辆额定载重量为 4500kg, 客户数为 29, 编号 1 对应的位置为配送中心, 各配送点坐标及需求量如表 1 所示.

表 1 各配送点坐标及需求量数据表

编号	x 坐标	y 坐标	需求量/kg	编号	x 坐标	y 坐标	需求量/kg
1	162	354	0	16	125	355	550
2	218	382	300	17	119	357	150
3	218	358	3100	18	115	341	100
4	201	370	125	19	153	351	150
5	214	371	100	20	175	363	400
6	224	370	200	21	180	360	300
7	210	382	150	22	159	331	1500
8	104	354	150	23	188	357	100
9	126	338	450	24	152	349	300
10	119	340	300	25	215	389	500
11	129	349	100	26	212	394	800
12	126	347	950	27	188	393	300
13	125	346	125	28	207	406	100
14	116	355	150	29	184	410	150
15	126	335	150	30	207	392	1000

(资料来源:VRP 测试库)

参数设置: $\rho=0.15$, $q_0=0.3$, $Q=50$. 蚂蚁总数取为 20, 循环取 100 代. 将本案例分别用基本蚁群算法、带精英策略蚁群算法、本文改进算法运行, 实验结果对照如下:

表 2 三种算法运行对照

算法名称	α	β	最优解	进化代数
基本蚁群算法	1	3	487.8218	62
	1	2	487.8218	74
带精英策略蚁群算法 ^[8]	1	3	485.4446	66
	1	2	485.4446	82
本文改进算法	1	3	485.4446	34
	1	2	485.4446	45

改进算法求解 Eil30 问题的进化曲线如图 2 所示:

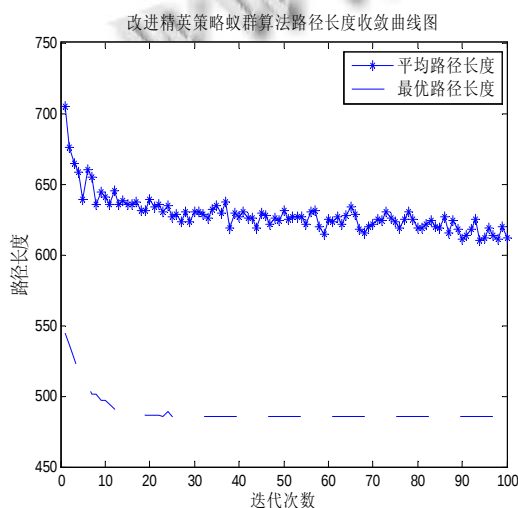


图 2 路径长度进化曲线图

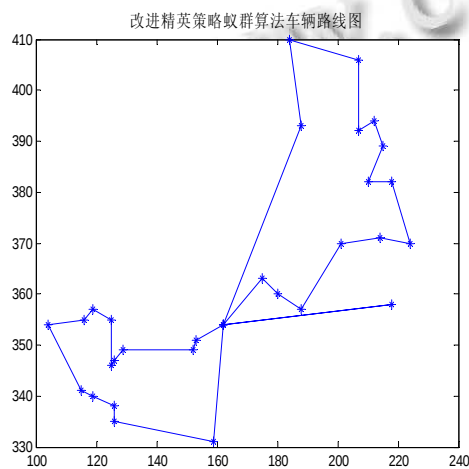


图 3 配送线路图

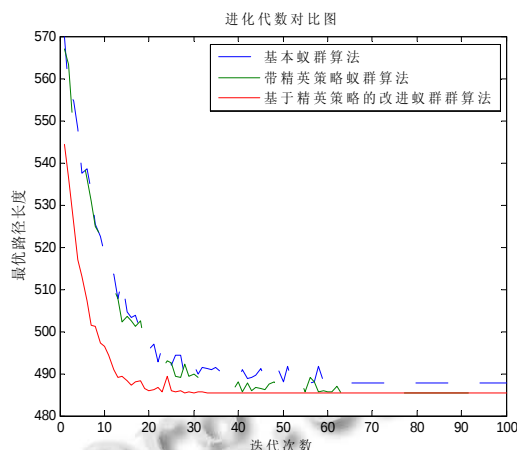


图 4 三种算法进化曲线对比图

三种算法的进化曲线如图 4 所示. 改进算法经过 34 代迭代后, 达到最优解, 而基本算法需要 62 代迭代, 才达到最优, 进化速度提高了约 45.2%. 改进算法求得最优配送线路如图 3 所示, 为:

第 1 辆车的路线:

1-20-21-23-4-5-6--2-7-25-26-30-28-29-27-1

第 2 辆车的路线:

1-22-15-9-10-18-8-14-17-16-13-12-11-24-19-1

第 3 辆车的路线: 1-3-1

配送共需 3 辆车, 最优路径长度为: 485.4446, 较文献 9 的最优结果 553.0731 明显改善. 实验中 100 次迭代, 其中 485.4446 的最优解共出现 5 次, 分别是第 34、44、94、97、100 次循环迭代, 485.4446 非常接近理论最优解了.

5 结语

本文针对蚁群算法计算速度慢, 容易陷入“早熟”, 导致算法搜索停滞的弊端, 提出了一种基于精英策略的改进蚁群算法. 新算法将遗传算法中排序的概念扩展到精英机制当中, 并对信息素设置上下界约束, 有效地避免某条路径上的信息量远大于其余路径, 使得所有的蚂蚁都集中到同一条路径上. 实例仿真表明, 运用改进蚁群算法的求解速度比基本算法提高了约 45.2%, 最优解的精度提高了约 3.3%.

参考文献

- 1 Dorigo M, Vittorio M, Alberto C. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE

(下转第 134 页)