

文章编号: 1005-0523(2010)02-0057-06

基于 DACS3 的改进蚁群算法求解 TSP 问题

王更生, 俞云新, 蔡求元, 尹 慧

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要: 蚁群算法是优化领域中新出现的一种仿生进化算法。该算法采用分布式并行计算机制, 具有较强的鲁棒性, 易与其他算法结合, 但存在运行时间长, 容易陷入局部最优解, 导致出现停滞现象等缺点。针对蚁群算法, 首先介绍其基本原理及不足之处。随后提出了一种改进算法, 该算法在选择路径时仅考虑信息素强度, 在信息素强度更新时采用基于 3 层动态信息素更新(Dynamic Ant Colony System with 3 level updates, DACS3)机制, 更好地模仿了自然蚂蚁。最后通过仿真验证该算法, 结果表明该算法可以取得较好的搜索效果。

关键词: 蚁群算法; 信息素强度; DACS3; TSP 问题

中图分类号: TP301.6

文献标识码: A

蚁群算法(Ant Colony Optimization, ACO)^[1-2]是意大利学者 Marco Dorigo 于 1992 在他的博士论文中引入, 灵感源于蚂蚁在寻找食物过程中发现较短路径的行为, 该算法最早成功地用于解决著名的旅行商问题^[3]。经过十几年的发展, 蚂蚁算法有诸多改进算法^[4-12], 经过各种数值仿真结果表明: 该算法具有许多优良的性质, 是求解组合最优化问题的新型通用启发式方法, 但也有相关文献资料表明该算法还存在一些不足, 还有提升的空间^[13-15]。

1 蚁群算法原理及优缺点

该算法模型可以通过 TSP(旅行商)问题描述^[3]。TSP(旅行商)问题是指完全遍历 n 个城市 1 次且仅 1 次所走过的最短距离。其蚁群算法数学模型^[3]如下。

首先引入如下符号: m 表示算法中蚂蚁的数量, $d_{ij}(i, j=1, 2, \dots, n)$ 表示边 (i, j) 之间的距离, n 为城市个数, $\tau_{ij}(t)$ 表示 t 时刻在 (i, j) 上残留的信息素量, 初始时刻, 各边信息素量相等。蚂蚁 k 在 t 时刻由城市 i 转移到城市 j 的概率为

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \notin \text{tabu}_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} & \text{if } j \notin \text{tabu}_k \\ 0 & \text{else} \end{cases} \quad (1)$$

式中: η_{ij} 为先验知识或能见度, 针对具体问题根据启发式规则而定; α 为边 (i, j) 上残留信息的重要程度; β 为启发信息的重要程度; tabu_k 为蚂蚁 k 的禁忌表即蚂蚁 k 所走过的城市集。

随着时间的推移, 以前蚂蚁留下的信息逐渐消逝, 用参数 ρ ($\rho \in (0, 1)$) 表示信息素挥发率, 当蚂蚁完成 1 次循环后, 各路径上的信息量根据下列公式做调整

$$\tau_{ij}(t)^n = (1 - \rho) \tau_{ij}(t)^0 + \Delta \tau_{ij} \quad (2)$$

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

收稿日期: 2009-10-21

作者简介: 王更生(1965-), 男, 教授, 硕士, 研究方向为 GIS/GPS。

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{if 蚂蚁 } k \text{ 过边 } (i,j) \\ 0 & \text{else} \end{cases} \quad (4)$$

式中: $\tau_{ij}(t)^n$ 表示更新后边 (i,j) 上的信息素量; $\tau_{ij}(t)^0$ 表示更新前边 (i,j) 上的信息素量; $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁本次循环中留在边 (i,j) 上的信息素量; $\Delta\tau_{ij}$ 表示本次循环中边 (i,j) 上信息量增量; Q 为常数; L_k 表示第 k 只蚂蚁在本次循环中所走过的路径长度。当所有蚂蚁都完成 1 次周游后,因每只蚂蚁本次周游的禁忌表已满,此时应及时清空,准备下 1 次周游。当周游次数达到设定值时算法结束。

自蚁群算法提出后,由于采用分布式并行计算机制,具有较强的鲁棒性,易与其他方法结合等优点被应用到了许多领域,取得了卓有成效的成果,但该算法也存在一些缺陷,如在算法的初始阶段,各条道路上的信息素水平基本相等,蚂蚁的搜索呈现出较大的盲目性,只有经过较长时间后,信息素水平才呈现出明显的指导作用,消耗了大量的时间,计算容易陷入局部最优解,算法收敛较慢,容易出现停滞现象等等。造成这些问题的主要原因是蚁群算法中的人工蚂蚁未能很好地模仿自然蚂蚁^[4]。

2 改进的蚁群算法

2.1 初始信息素的改进

在任意两个城市之间安排的信息素是等量的,但是这等量的信息素要平均到两个之间的路径上,由于城市之间的距离是不相同的,所以平均到每一小段上的信息量就是距离的倒数与分配到这两城市之间的信息量的乘积。

$$\text{任一路径 } (i,j) \text{ 上的初始信息素强度: } \tau_{ij} = \begin{cases} Q/d_{ij} & \text{if } i \neq j \\ 0 & \text{else} \end{cases}$$

式中: τ_{ij} 表示城市 i 与城市 j 之间的信息素强度; Q 表示初始信息素量且 $Q > 0$; d_{ij} 表示城市 i 与城市 j 之间的距离。

2.2 路径选择的改进

因为自然蚂蚁无法预测距离的长短,为更好模拟自然蚂蚁,在选择下一个城市时不再考虑距离因素,仅考虑信息素强度。同时为有效提高优化速度,降低局部最优解停滞的可能性,本文采用确定性和随机性选择相结合的选择策略,并在搜索过程中动态地调整确定性选择的概率^[5,6]。蚂蚁 $k(1 \cdots m)$ 在 t 时刻由城市 i 到城市 j 的转移概率

$$J = \begin{cases} \max\{\tau_{iu}\} & \text{if } q < q_0 \\ J & \text{else} \end{cases} \quad (5)$$

在选择下一个城市之前随机生成 q , 如果 q 的值小于 q_0 , 从城市 i 所有可行的城市找出 τ_{iu} 最大的城市,即为下一个要选择的的城市; 否则按 J 来选择下一个城市。

$$J = \begin{cases} \frac{\tau_{ij}}{\sum_{k \notin \text{tabu}_k} \tau_{ik}} & j \notin \text{tabu}_k \\ 0 & \text{else} \end{cases} \quad (6)$$

式中: τ_{ij} 为边 (i,j) 上的信息素强度; tabu_k 为蚂蚁 k 的禁忌表。在实际操作时,常采用赌盘法来确定 j 。在上面转移概率的求解过程中仅考虑了信息素强度。

2.3 信息素更新的改进

自然蚂蚁在爬行时均匀释放信息素,释放与挥发是同步的,而现有蚁群算法中的人工蚂蚁是在走完 1 个循环或 1 个边之后再进行信息素的释放与挥发,为更好地模拟自然蚂蚁和提高收敛速度,本文中基于 DACS3 的改进蚁群算法在信息素更新时采用 3 层动态更新的机制。

第 1 层 实时更新。每只蚂蚁 k 在建立 1 条路径的过程中,在蚂蚁 k 选择完下 1 个城市后按照下面的更新规则进行信息素强度更新:

对全部边 (i,j) 都进行1次信息素的挥发操作,即

$$\tau_{ij}^n = (1 - \rho) \tau_{ij}^0 \quad (7)$$

式中: τ_{ij}^n 表示边 (i,j) 更新后的信息素强度; τ_{ij}^0 表示边 (i,j) 更新前信息素强度。

当蚂蚁 k 走过边为 (p,q) 时,按公式(8)更新信息素强度:

$$\tau_{pq}^n = \tau_{pq}^0 + \frac{\rho' Q'}{d_{pq}} \quad (8)$$

式中: τ_{pq}^n 为更新后的信息素强度; τ_{pq}^0 表示更新前信息素强度; Q' 表示蚂蚁 k 在路径 (p,q) 释放的信息素量; ρ' 表示蚂蚁 k 释放的同时信息素挥发率,取0到1之间的小数; d_{pq} 表示边 (p,q) 的长度。

加入信息素强度的实时更新可以更好地模仿自然蚂蚁的信息素随时间的动态更新。虽然没有完全按时间连续地对信息素强度进行更新,但是在把时间离散化(即把蚂蚁走完1个边的时间作为1个时间跨度)的情况下已经达到了信息素强度的动态更新。

第2层 局部最优最差更新。为了加快收敛速度,当所有蚂蚁都完成1条路径后比较 m 条路径,记录最短最长路径,并按下列公式进行局部信息素强度更新:

$$\tau_{rs}^n = (1 - \rho) \tau_{rs}^0 + \Delta C \quad (9)$$

$$\Delta C = \begin{cases} p \cdot \Delta \tau_{rs} & \text{if } (r,s) \in LBT \\ -p \cdot \Delta \tau_{rs} & \text{if } (r,s) \in LWT \\ 0 & \text{else} \end{cases} \quad (10)$$

$$\Delta \tau_{rs} = \begin{cases} \left(\frac{L_{grb}}{L} \right) - 1 \\ \left(\frac{L_{grw}}{L} \right) - 1 \end{cases} \quad (11)$$

式中: τ_{rs}^n 表示更新后边 (r,s) 的信息素强度; τ_{rs}^0 表示更新前边 (r,s) 的信息素强度; p 表示加强的信息素总量; LBT 表示局部最优路径; LWT 表示局部最差路径; L_{grb} 表示局部最优路径的长度; L_{grw} 表示局部最差路径的长度。

第3层 全局最优最差更新。当前迭代从第2层产生的所有局部最优最差路径中,选出最优最差路径,按下列公式进行信息素强度的全局更新:

$$\tau_{rs}^n = (1 - \rho) \tau_{rs}^0 + \Delta C \quad (12)$$

$$\Delta C = \begin{cases} p \cdot \Delta \tau_{rs} & \text{if } (r,s) \in GBT \\ -p \cdot \Delta \tau_{rs} & \text{if } (r,s) \in GWT \\ 0 & \text{else} \end{cases} \quad (13)$$

$$\Delta \tau_{rs} = \begin{cases} \left(\frac{L_{gb}}{L} \right) - 1 \\ \left(\frac{L_{gw}}{L} \right) - 1 \end{cases} \quad (14)$$

式中: τ_{rs}^n 表示更新后边 (r,s) 的信息素强度; τ_{rs}^0 表示更新前边 (r,s) 的信息素强度; p 表示加强的信息素量; GBT 表示全局最优路径; GWT 表示全局最差路径; L_{gb} 表示全局最优路径的长度; L_{gw} 表示全局最差路径的长度。

3 改进蚁群算法求解TSP问题的算法实现

(1) 设定终止迭代步数 NC ,城市个数 n ,蚂蚁数量 m ,初始各城市信息素量 Q ,城市间距离 d [1] [1],初始化个路径的信息强度 $init\tau$ [1] [1],信息素挥发率 ρ ,信息素加强量 p ,信息素释放与挥发共存的挥发率 ρ' 及常数 q_0 。

(2) m 只蚂蚁随机选取初始位置 i ,并把 i 放入禁忌表 $tabu_k$ 中,对每只蚂蚁按照(5)(6)式来选择下一目标位置 j ,将 j 移入 $tabu_k$ 中, j 作为蚂蚁当前所在位置,并按(7)(8)式进行整体信息素强度调整。

(3) 经过 n 个时刻,所有蚂蚁完成1次循环,即结束1次迭代过程,记录局部最优及最差路径,按(9)

(10) (11) 进行信息素强度局部最优最差更新。

(4) 经过 c 次迭代,记录最优及最差路径,按(12) (13) (14) 式进行全局最优最差信息素强度更新,记录最优路径。

(5) 判断是否达到搜索次数 NC ,如果达到搜索次数则输出结果,结束算法,否则清空禁忌表,转(2)。

4 实例仿真

本文根据 TSP 问题中的 Eil51 和 Eil76 两组数据对算法做了仿真,分别用 C++ 语言为基本蚁群算法和本文提出的改进蚁群算法设计了程序并进行运算。

参数的选定如下:基本蚁群算法: $\alpha = 1, \beta = 5, \rho = 0.5, Q = 100$ 。改进蚁群算法: $\rho = 0.5, \rho' = 0.3, q_0 = 0.5, Q = 100, Q' = 50$ 。

对于两种算法在求解 Eil51 问题时都取蚂蚁数为 34,在求解 Eil76 问题时都取蚂蚁总数为 51,循环都取 1 000 代,每一种算法都分别运算 50 次然后选出较优的 20 次结果并记录,见表 1 和表 2。

表 1 Eil51 的运算结果

序号	基本算法结果	改进算法结果
1	442.230 046	438.722 958
2	445.926 089	439.457 932
3	441.235 737	441.253 338
4	444.857 433	439.067 695
5	442.751 706	433.442 384
6	439.287 932	437.138 990
7	438.741 338	435.487 343
8	448.420 543	440.249 074
9	442.547 960	440.706 745
10	446.533 913	434.590 920
11	440.460 959	438.445 234
12	445.987 319	440.253 456
13	441.109 041	439.879 845
14	442.369 984	435.675 432
15	444.021 913	433.923 421
16	443.154 714	434.523 871
17	442.827 441	440.769 847
18	448.005 910	437.719 277
19	447.341 070	438.657 382
20	446.897 321	436.789 642

表 2 Eil76 的运算结果

序号	基本算法结果	改进算法结果
1	559.756 410	555.274 521
2	562.560 473	557.402 633
3	561.614 223	556.992 639
4	562.331 073	556.655 761
5	559.169 869	555.611 449
6	560.115 382	557.194 533
7	561.981 801	557.040 307
8	561.046 332	556.992 639
9	560.660 194	560.659 605
10	560.580 366	558.193 210
11	562.041 599	555.275 471
12	561.249 608	558.193 210
13	559.281 524	556.662 639
14	561.459 193	559.305 234
15	560.869 895	559.405 876
16	560.723 708	555.769 324
17	561.205 513	557.452 359
18	560.585 148	557.439 442
19	563.027 751	560.784 220
20	561.614 223	558.079 696

对 Eil51 问题用基于 DACS3 的改进算法所得的较优路径长度为: 433.442 384,对应的路径为: 39,18,40,12,24,13,17,46,11,45,50,26,5,47,22,23,42,6,25,7,30,27,2,35,34,19,28,20,33,49,15,1,21,0,31,10,37,4,48,8,29,9,38,32,44,14,43,36,16,3,41。

对 Eil76 问题用基于 DACS3 的改进算法所得的较优路径长度为: 555.274 521,对应的路径为: 68,35,46,47,28,44,26,51,33,45,7,34,6,25,66,75,74,3,29,1,67,5,50,16,39,11,57,71,38,8,31,43,2,15,62,32,72,61,27,73,20,60,21,63,41,40,42,0,22,55,48,23,17,49,24,54,30,9,37,64,10,65,58,13,52,18,53,12,56,14,4,36,19,69,59,70。

在图 1 和图 2 中分别给出了基于 DACS3 的改进蚁群算法在求解 Eil51 和 Eil76 问题时取得较优路径

的收敛曲线。

从仿真结果可以看出,基于 DACS3 的改进蚁群算法求解各个 TSP 实例比基本蚁群算法性能好,说明改进的蚁群算法比基本蚁群算法更能使搜索具有多样性,增大搜索空间。曲线图的说明:随着迭代次数的增加,最优路径的长度在减小,最终找到最短长度。结果表明,基于 DACS3 的改进蚁群算法具有全局收敛性,是一种行之有效的优化算法。

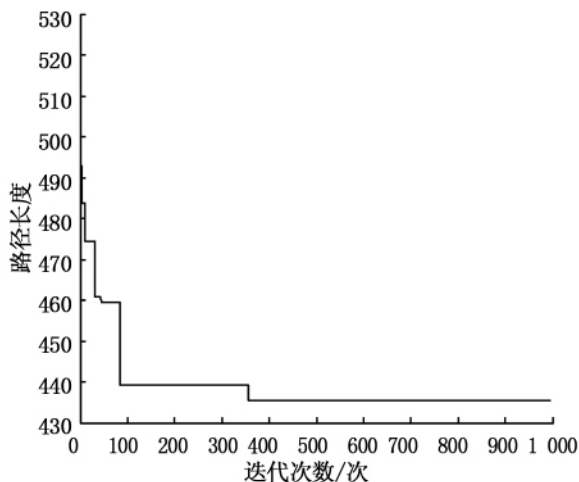


图1 Eil51 的收敛过程图

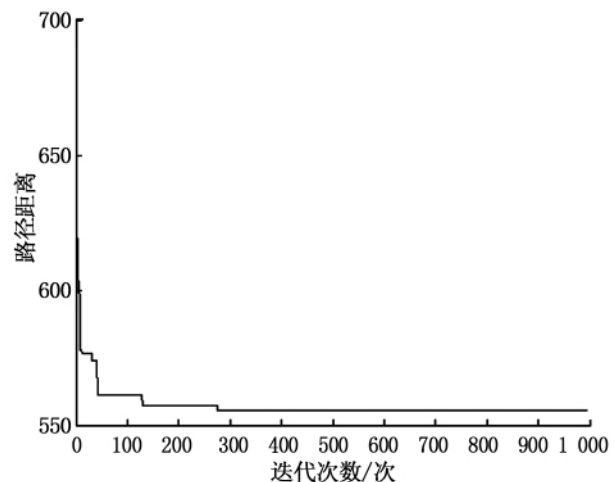


图2 Eil76 的收敛过程图

5 结束语

提出的基于 DACS3 的改进蚁群算法在各路径信息素初始化时引入信息素强度的概念,在蚂蚁的转移概率求解时仅考虑了信息素强度这一因素,在信息素强度更新时运用了用距离代替时间的理念与3 层动态更新的机制。通过 C++ 编程仿真建立模型并把获得的结果进行了比较分析,我们得出,基于 DACS3 的改进蚁群算法可以得到较好的收敛效果。由于改进算法在搜索下一城市的时候仅考虑信息素强度的特点使得运算简单、快捷,并且比现有的蚁群算法更好地模仿了自然蚂蚁,为进一步的模仿自然蚂蚁的其他优点提供了可能。

提出基于 DACS3 的改进蚁群算法是基于单纯考虑较优路径的,但由于 TSP 问题是属于 NP 问题,因此在实际应用中往往耗费大量的时间。所以从实用性角度,应该综合考虑时间效率和较优路径两个方面,这有待日后的研究。另外,没有进行理论上的证明缺乏理论基础。同时缺乏统一的理论,既是蚁群算法中的 1 个难点也是该算法未来的 1 个重要研究方向。总之,对蚁群算法的研究才刚开始,算法中的许多问题有待进一步的研究。

参考文献:

- [1] BLUM C. Ant colony optimization: introduction and recent trends [J]. Physics of Life Reviews, 2005, 2(4): 353-373.
- [2] COLORM A, DORIGOM, MINIEZZO V. Distributed optimization by ant colonies [C]. Proceeding of the First European Conference on Artificial Life. Paris France: Elsevier Publishing, 191: 134-142.
- [3] 劳眷. 蚁群算法求解 TSP 问题若干改进策略的研究 [J]. 科学技术与工程, 2009, 9(9): 2459-2462.
- [4] 杨洁, 杨胜, 曾庆光. 基于信息素强度的蚁群算法 [J]. 计算机应用, 2009, 29(3): 865-867.
- [5] 段海滨, 王道波, 朱家强. 蚁群算法理论及应用研究的进展 [J]. 控制与决策, 2004, 19(12): 1322-1340.
- [6] 杨中秋, 张延华, 郑志丽. 基于改进蚁群算法对最短路径问题的分析与仿真 [J]. 沈阳化工学院学报, 2009, 23(2): 150-153.
- [7] 李士勇. 蚁群算法及其应用 [M]. 哈尔滨: 哈尔滨工业大学出版社, 2004: 33-34.
- [8] 孙燕, 王秀坤, 刘业欣. 一种简单蚂蚁算法及其收敛性分析 [J]. 小型微型计算机系统, 2003, 21(8): 1524-1526.

- [9] 吴庆洪,张纪会,徐心和. 具有变异特征的蚁群算法 [J]. 计算机研究与发展,1999,36(10):1 240-1 245.
- [10] 丁建立,陈增强,袁著祉. 基于自适应蚂蚁算法的动态最优路由选择 [J]. 控制与决策,2003,18(6):751-753.
- [11] DORIGO M,MANIEZZO V,COLORNI A. The ant system: optimization by a colony of cooperating agent [J]. IEEE Transactions on Systems,Man,and Cybernetics,Part B,1996,26(1):29-41.
- [12] DORIGO M,GAMBARDELLA L M,MIDDENDORF M. Guest editorial: special section on ant colony optimization [J]. IEEE Transactions on Evolutionary Computation,2002,6(4):317-319.
- [13] 胡祥培,丁秋雷,李永先. 蚁群算法研究评述 [J]. 管理工程学报,2008,22(2):74-79.
- [14] 杨剑锋. 蚁群算法及其应用研究 [D]. 浙江:浙江大学,2007.
- [15] 陈昊. 蚁群优化算法的原理及其应用 [J]. 湖北大学学报:自然科学版,2006,28(4):350-352.

An Analysis and Simulation of TSP with Improved Ant Colony Algorithm Based on DACS3

Wang Gengsheng, Yu Yunxin, Cai Qiuyuan, Yin Hui

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: Ant colony algorithm is a new category of bionic algorithm in optimization field. Adopting parallel computation mechanism, ant colony algorithm has strong robustness and is easy to combine with other methods in optimization. But it has the shortcoming of long running time and easily getting into local best solution, and falling into stagnating state. Firstly, the basic principle and shortcoming of ant colony algorithm is introduced. Then, the paper proposes a new ant colony algorithm which only depends on the intensity of pheromone when selecting the path, adopts the system based on Dynamic Ant Colony System with 3 level updates when updating the pheromone, and better simulates the natural ants. Finally, the proposed algorithm is verified by simulation. The experiments demonstrate that the proposed algorithm can obtain good searching results.

Key words: ant colony optimization(ACO); intensity of pheromone; DACS3; TSP

(责任编辑 刘棉玲)