

# 一种快速全局优化的改进蚁群算法及仿真

段海滨, 王道波

(南京航空航天大学自动化学院, 江苏 南京 210016)

**摘要:** 在介绍基本蚁群算法原理的基础上, 对其作了许多改进以提高其全局优化寻优速度, 并给出了详尽的新算法编程仿真实现步骤, 最后将未改进的基本蚁群算法与本文改进后的蚁群算法分别应用于 TSPLIB 中的 Att532TSP 问题进行了仿真实验。仿真研究表明, 改进后的算法具有优良的全局优化性能, 效果令人满意。

**关键词:** 全局优化; 蚁群算法; 信息素; 仿真

**中图分类号:** TP13

**文献标识码:** A

## A Novel Improved Ant Colony Algorithm with Fast Global Optimization and its Simulation

DUAN Hai-bin, WANG Dao-bo

(College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

**Abstract:** Firstly, the principle of basic ant colony algorithm (ACA) is introduced. Then, a series of schemes to improve the global optimization properties are proposed, and a detailed programming process is presented. In the end, an example of Att532TSP is given, which is simulated by using basic ACA and improved ACA. The simulation results show that the improved ACA has excellent global optimization properties, and the effect of simulation is satisfactory.

**Keywords:** global optimization; ant colony algorithm; pheromone; simulation

### 1 引言 (Introduction)

随着近代仿生学的发展, 人们越来越关注自然界中一些看似微不足道的生物行为。20 世纪 90 年代初期, 意大利学者 Dorigo Macro 等人通过模拟自然界中蚂蚁集体寻径的行为而提出了蚁群算法<sup>[1]</sup> (Ant Colony Algorithm, 简称 ACA), 这是一种基于种群的启发式仿生进化算法。该算法最早成功应用于解决著名的旅行商问题 (TSP)。它采用分布式并行计算机制, 易于与其它方法结合, 具有较强的鲁棒性<sup>[2]</sup>, 最近几年开始引起了国内外专家学者的关注<sup>[3~7]</sup>, 但蚁群算法搜索时间长、收敛速度慢的弱点一直制约着它在众多领域的进一步推广应用。本文对基本的蚁群算法作了一系列改进, 使其能在全局优化的实现过程中快速找到全局最优解。

### 2 基本蚁群算法原理 (Basic ant colony algorithm)

#### 2.1 蚁群行为描述

根据仿生学家长期的研究发现: 蚂蚁虽没有视觉, 但运动时会在路径上释放出一种特殊的分泌物——信息素 (Pheromone)——寻找路径<sup>[8]</sup>。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, 同时会释放出与路径长度有关的信息素。蚂蚁走得路径越长, 则释放的信息素数量越小。当后来的蚂蚁再次碰到这个路口的时候, 选择信息素数量较大路径概率就会相对较大, 这样形成了一个正反馈机制。最优路径上的信息素数量越来越大, 而其它的路径上信息素数量却会随着时间的流逝而消减, 最终整个蚁群会找出最优路径。而且蚂蚁还能够适应环境的变化, 当蚁群的运动路径上突然出现障碍物时, 蚂蚁亦能够很快地重新找到最优路径。可见在整个寻径过程中, 虽然单个蚂蚁的选择能力有限, 但是通过信息素的作用使整个蚁群的行为具有非常高的自组织性, 蚂蚁之间交换着路

径信息,最终通过蚁群的集体自催化行为找出最优路径。

## 2.2 数学模型

蚁群算法包含两个基本阶段:适应阶段和协作阶段。在适应阶段,各候选解根据积累的信息不断调整自身结构,路径上经过的蚂蚁越多,信息素数量越大,则该路径越容易被选择;时间越长,信息素数量越小。在协作阶段,候选解之间通过信息交流,以期产生性能更好的解。为了能够清楚地理解蚁群算法的数学模型,本文借助了经典的对称 TSP 问题。

设  $C = \{c_1, c_2, \dots, c_n\}$  是  $n$  个城市的集合,  $L = \{l_{ij} | c_i, c_j \in C\}$  是集合  $C$  中元素(城市)两两连接的集合,  $d_{ij}(i, j = 1, 2, \dots, n)$  是  $l_{ij}$  的 Euclidean 距离:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

$G = (C, L)$  是一个有向图, TSP 问题的目的是从有向图  $G$  中寻出长度最短的 Hamilton 圈,此即一条对  $C = \{c_1, c_2, \dots, c_n\}$  中  $n$  个元素(城市)访问且只访问一次的最短封闭曲线。

设  $b_i(t)$  表示  $t$  时刻位于元素  $i$  的蚂蚁的个数,  $\tau_{ij}(t)$  为  $t$  时刻路径  $(i, j)$  上的信息素数量,  $m$  为蚁群中蚂蚁的数目,则  $m = \sum_{i=1}^n b_i(t)$ ;  $\Gamma = \{\tau_{ij}(t) | c_i, c_j \in C\}$  是  $t$  时刻集合  $C$  中元素(城市)两两连接  $l_{ij}$  上的残留信息素数量集合。在初始时刻各条路径上信息素数量相等,设  $\tau_{ij}(0) = \text{const}$  ( $\text{const}$  为常数)。蚁群算法的寻优是通过有向图  $g = (C, L, \Gamma)$  实现的。

**准则 1 (转移概率准则):** 蚂蚁  $k$  ( $k = 1, 2, \dots, m$ ) 在运动过程中,根据各条路径上的信息素数量决定转移方向。算法中人工蚂蚁与实际蚂蚁不同,具有记忆功能。

禁忌表  $\text{tabu}_k$  ( $k = 1, 2, \dots, m$ ) 用来记录蚂蚁  $k$  当前所走过的城市,集合随着  $\text{tabu}_k$  进化过程做动态调整。在搜索过程中,蚂蚁根据各个路径上的信息素数量及路径的启发信息来计算转移概率。 $p_{ij}^k(t)$  表示在  $t$  时刻蚂蚁  $k$  由元素(城市)  $i$  转移到元素(城市)  $j$  的转移概率:

$$p_{ij}^k(t) = \begin{cases} \frac{|\tau_{ij}(t)|^\alpha \cdot |\eta_{ij}(t)|^\beta}{\sum_{s \in \text{allowed}_k} |\tau_{is}(t)|^\alpha \cdot |\eta_{is}(t)|^\beta}, & \text{若 } j \in \text{allowed}_k \\ 0, & \text{否则} \end{cases} \quad (2)$$

公式(2)中,  $\text{allowed}_k = \{C - \text{tabu}_k\}$  表示蚂蚁  $k$

下一步允许选择的城市。 $\alpha$  表示轨迹的相对重要性,反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起的作用,其值越大,该蚂蚁越倾向于选择其它蚂蚁经过的路径,蚂蚁之间协作性越强; $\beta$  表示能见度的相对重要性,反映了蚂蚁在运动过程中启发式因子在蚂蚁选择路径中的受重视程度,其值越大,则该转移概率越接近于贪心规则<sup>[4]</sup>。 $\eta_{ij}(t)$  为启发函数:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (3)$$

对蚂蚁  $k$  而言,  $d_{ij}$  越小,则  $\eta_{ij}(t)$  越大,  $p_{ij}^k(t)$  也就越大。显然,该启发函数表示出了蚂蚁从元素(城市)  $i$  转移到元素(城市)  $j$  的期望程度。

**准则 2 (局部调整准则):** 局部调整是每只蚂蚁在建立一个解的过程中进行的。随着时间的推移,以前留下的信息逐渐消逝,经过  $h$  个时刻,两个元素(城市)状态之间的局部信息素数量要根据下式作调整:

$$\tau_{ij}(t+h) = (1-\zeta) \cdot \tau_{ij}(t) + \zeta \cdot \tau_0 \quad (4)$$

$$\tau_0 = \frac{1}{nl_{\min}} \quad (5)$$

式中,  $\zeta \in [0, 1]$ ,  $l_{\min}$  表示集合  $C$  中两个最近元素(城市)之间的距离。

**准则 3 (全局调整准则):** 只有生成了全局最优解的蚂蚁才有机会进行全局调整,全局调整规则为:

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

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (7)$$

式中,  $\rho$  为挥发系数,  $\rho \in [0, 1]$ ,  $\Delta\tau_{ij}(t)$  表示本次循环中路径  $ij$  上的信息素数量的增量,初始时刻  $\Delta\tau_{ij}(t) = 0$ ;  $\Delta\tau_{ij}^k(t)$  表示第  $k$  只蚂蚁在本次循环中留在路径  $ij$  上的信息量。

## 2.3 算法模型种类

文献[1,2]中给了三种不同的蚁群算法模型,分别称之为蚂蚁圈(Ant-Cycle)模型、蚂蚁数量(Ant-Quantity)模型及蚂蚁密度(Ant-Density)模型,它们的差别在于  $\Delta\tau_{ij}^k(t)$  求法的不同。

在 Ant-Cycle 模型中:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{若第 } k \text{ 只蚂蚁在本次循环中经过 } ij \\ 0, & \text{否则} \end{cases} \quad (8)$$

式中,  $Q$  是信息素强度,它影响算法的收敛速度。 $L_k$  表示第  $k$  只蚂蚁在本次循环中所走路径的总

长度.

在 Ant-Quantity 模型中:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{d_{ij}}, & \text{若第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 之间经过 } ij \\ 0, & \text{否则} \end{cases} \quad (9)$$

在 Ant-Density 模型中:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q, & \text{若第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 之间经过 } ij \\ 0, & \text{否则} \end{cases} \quad (10)$$

区别:模型(9)和(10)中利用的是局部信息;而(8)中利用的是整体信息,在求解 TSP 问题时性能较好,因而通常采用(8)作为基本模型.

由算法复杂性分析理论,  $m$  个蚂蚁要遍历  $n$  个元素(城市), 经过  $N_c$  次循环, 则算法复杂度为  $O(N_c \cdot m \cdot n^2)$ .

### 3 快速全局优化的改进蚁群算法 (Improved ant colony algorithm with fast global optimization)

#### 3.1 改进策略

显然, 蚂蚁数量越大, 算法的全局搜索能力越强, 但这会使算法的全局收敛速度变慢. 因此当蚂蚁数量  $m$  很大时, 对集合  $C$  元素(城市)的搜索机制由一只蚂蚁改为两只蚂蚁从两头寻找, 两只蚂蚁共用一个禁忌表  $tabu_k (k=1, 2, \dots, m)$ , 这种并行处理策略可有效地提高算法的全局收敛性.

为了在一定程度上减小算法陷入局部最优解的概率, 可将转移概率准则改为可得到多样性解的新准则<sup>[9]</sup>, 即让第  $k$  只蚂蚁按公式(11)概率从元素(城市)  $i$  转移到元素(城市)  $j$ :

$$j = \begin{cases} \underset{\alpha \in Allowed_k}{\operatorname{argmax}} \{ |\tau_{is}(t)|^\alpha \cdot |\eta_{is}(t)|^\beta \}, & \text{若 } q \leq q_0 \\ \text{根据公式 2 选择 } j, & \text{否则} \end{cases} \quad (11)$$

其中,  $q$  是  $(0, 1)$  均匀分布的随机数,  $q_0 \in [0, 1]$ ,  $q_0$  越小, 蚂蚁随机选择的概率越大.

在信息素的更新方面, 设集合

$$\begin{aligned} D &= \{D[k] | D[k] = \sum_{i,j \in C} d_{ij}[k], k=1, 2, \dots, m\} \\ d(t)_{\min} &= \min\{D[1], D[2], \dots, D[m]\} \\ d(t)_{\text{aver}} &= \frac{\min\{D[1] + D[2] + \dots + D[m]\}}{m} \end{aligned} \quad (13)$$

$d(t)_{\min}$  为第  $t$  个搜索周期得到的最短路径长度,  $d(t)_{\text{aver}}$  为第  $t$  个搜索周期  $m$  只蚂蚁周游路径长度的平均值. 只有当  $d(t)_{\min} < d(t-1)_{\min}$  且  $D[k] < d(t)_{\text{aver}}$  的时候, 蚂蚁  $k$  才按照公式(8)计算  $\Delta\tau_{ij}^k(t)$ .

由于信息素强度  $Q$  是表征蚂蚁所留轨迹数量的一个常数, 它影响算法的全局收敛速度. 其值过大, 会使算法收敛于局部最小值; 过小, 会使全局收敛速度减慢. 当集合  $C$  包含的元素(城市)很多时,  $Q$  的值也需要随之变化, 即将信息素强度由常数转化为阶梯函数<sup>[10]</sup>:

$$Q(t) = \begin{cases} Q_1, & 0 \leq t \leq T_1 \\ Q_2, & T_1 < t \leq T_2 \\ Q_3, & T_2 < t < \infty \end{cases} \quad (14)$$

式中  $Q_1$ 、 $Q_2$ 、 $Q_3$  分别对应阶梯函数不同的常数值.

在集合  $C$  元素(城市)很多的情况下, 若挥发系数  $\rho$  太大, 算法的全局搜索能力就会降低. 若减小  $\rho$ , 算法的全局搜索能力会随之提高, 但收敛速度会变慢. 可以对  $\rho$  值采取自适应控制策略, 即  $\rho$  的初始值可以取大, 随着循环次数的不断增加, 若每次的最优值相差不大, 说明过程陷入了某个极值点, 不一定是全局最优解. 此时, 应将  $\rho$  改为阈值函数:

$$\rho(t+1) = \begin{cases} \xi \cdot \rho(t), & \text{若 } \xi \cdot \rho(t) > \rho_{\min} \\ \rho_{\min}, & \text{否则} \end{cases} \quad (15)$$

$\xi$  为挥发约束系数, 且  $\xi \in (0, 1)$ .

将各条寻优路径上可能的残留信息素数量限制在  $[\tau_{\min}, \tau_{\max}]$ ,  $\tau_{\min}$  可以有效地避免算法停滞;  $\tau_{\max}$  可以避免某条路径上的信息量远大于其它路径, 使所有的蚂蚁都集中到同一条路径上面, 从而限制了算法的扩散<sup>[10]</sup>. 每次循环结束后, 保留最优路径, 一个循环中只有路径最短的蚂蚁才有权修改  $\tau_{ij}(t)$ . 修改策略在公式(6)的基础上, 再加上公式(16)进行阈值判断选择.

$$\tau_{ij}(t+n) = \begin{cases} \tau_{\min}, & \text{若 } \tau_{ij}(t) \leq \tau_{\min} \\ \tau_{ij}(t), & \text{若 } \tau_{\min} < \tau_{ij}(t) < \tau_{\max} \\ \tau_{\max}, & \text{若 } \tau_{ij}(t) \geq \tau_{\max} \end{cases} \quad (16)$$

#### 3.2 编程仿真步骤

改进后蚁群算法编程仿真的主要步骤如下:

第 1 步: 参数初始化

令时间  $t=0$  和循环次数  $N_c=0$ , 设置最大循环次数  $N_{c\max}$ , 将  $m$  蚂蚁置于  $n$  个元素(城市)上, 令信息量最大值  $\tau_{\max} = \tau_{ij}(0)$ , 经验上取  $\tau_{\min} = \tau_{\max}/(2 \cdot n)$  且初始时刻  $\Delta\tau_{ij}(0) = 0$ ,  $\rho(t_0)$ 、 $Q_1$ 、 $Q_2$ 、 $Q_3$  均为常数, 且  $Q_1 < Q_2 < Q_3$ ,  $q_0$  为常数,  $k=0$ ;

第 2 步: 循环次数  $N_c \leftarrow N_c + 1$ ;

第 3 步: 蚂蚁数目  $k \leftarrow k + 2$ ;

第4步: 每两只蚂蚁作为一组共同选择一个元素(城市)作为起点;

第5步: 蚂蚁甲根据公式(11)计算的概率选择元素(城市) $j_1$ 并前进,  $j_1 \in \{C - tabu_k\}$ ; 蚂蚁乙也根据公式(11)计算的概率选择元素(城市) $j_2$ 并前进, 但  $j_2 \in \{C - tabu_k - j_1\}$ ;

第6步: 当前路径长度大于本次  $m$  只蚂蚁相遇循环的最短路径, 则终止此次相遇循环;

第7步: 若集合  $C$  中元素(城市)未遍历完, 则跳转到第5步;

第8步: 若  $k < m$ , 则跳转到第3步;

第9步: 按照公式(12)计算本次  $m$  只蚂蚁派对循环的最短路径长度  $d(t)_{\min}$ , 记录当前最优解, 并令  $tabu_k = \Omega$ ;

第10步: 按照公式(13)计算本次  $m$  只成对的蚂蚁循环的路径长度平均值  $d(t)_{\text{aver}}$ ;

第11步: 只有当  $d(t)_{\min} < d(t-1)_{\min}$  且  $D[k] < d(t)_{\text{aver}}$  的时候, 蚂蚁  $k$  才按照公式(8)计算  $\Delta\tau_{ij}^k(t)$ , 同时按照公式(14)自适应地选择  $Q$ ;

第12步: 根据公式(15)自适应地调整挥发系数值;

第13步: 根据公式(6)、(7)及(16)更新每条路径上的残留信息量;

第14步: 如果蚂蚁群全部收敛到一条路径或循环次数  $N_c \geq N_{c\max}$ , 则循环结束, 同时打印出全局寻优的最佳路径; 否则, 跳转到第2步。

#### 4 仿真结果分析 (Analysis of simulation results)

将 Dorigo Macro 提出的基本蚁群算法与本文提出的针对实现快速全局优化的蚁群算法分别应用于 TSPLIB 中的 Att532TSP (即由美国 532 座城市组成的旅行商问题) 进行仿真实验。

实验采用 Borland C++ 语言编程, 硬件为内存为 256M、CPU 为 Pentium III 的计算机。设置算法参数  $\alpha = 1$ ,  $\beta = 5$ ,  $Q_1 = 2$ ,  $Q_2 = 4$ ,  $Q_3 = 6$ ,  $T_1 = 20$ ,  $T_2 = 40$ ,  $T_3 = 60$ ,  $\rho(t_0) = 0.95$ ,  $\rho_{\min} = 0.5$ ,  $\Delta\tau_{ij}(0) = 0$ ,  $N_{c\max} = 600$ ,  $\tau_{\max} = \tau_{ij}(0)$ ,  $\tau_{\min} = \tau_{\max}/(1064)$ ,  $q_0 = 0.45$ 。仿真实验结果见表 1。

由对 Att532TSP 的仿真计算结果可知, 本文提出的改进蚁群算法与基本蚁群算法相比较, 能够在较短的仿真时间内找到全局最优解, 并且使算法的相对误差大大减小, 其全局搜索速度和优化性能都有了很大改善。

表 1 Att532TSP 仿真结果比较

Tab. 1 Comparison of Att532TSP simulation results

算法类型	仿真最优解	仿真时间(s)	实际最优解	相对误差%
基本 ACA	28085	359	27686	1.444
本文 ACA	27691	62	27686	0.018

## 5 结论 (Conclusion)

本文提出的一系列改进策略使基本蚁群算法的全局优化速度有了显著的提高, 有效地克服了该算法收敛速度慢、易陷于局部最小点的缺陷, 算法的优化性能得到了有效的改善。

蚁群算法的研究虽然才刚刚开始, 但这种模仿自然生物寻优思想的仿生进化算法已经展现出十分广泛的推广应用前景。

## 参 考 文 献 (References)

- [1] Colomi A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies [A]. Proceedings of ECAL91 (European Conference on Artificial Life) [C]. Paris, France; 1991. 134 ~ 142.
- [2] Dorigo M, Maniezzo V, Colomi A. The ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics - Part B, 1996, 26(1): 29 ~ 41.
- [3] Verbeeck K, Nowe A. Colonies of learning automata [J]. IEEE Transactions on Systems, Man, and Cybernetics—Part B, 2002, 32(6): 772 ~ 780.
- [4] Montgomery J, Randall M. Anti-pheromone as a tool for better exploration of search space [A]. Proceedings of Third International Workshop ANTS [C]. Brussels, Belgium; 2002. 100 ~ 110.
- [5] 赵学峰. 基于蚁群算法的一类扩展型 TSP 研究 [J]. 系统工程, 2003, 21(1): 17 ~ 21.
- [6] 王 颖, 谢剑英. 一种基于改进蚁群算法的多点路由算法 [J]. 系统工程与电子技术, 2001, 23(8): 98 ~ 101.
- [7] 覃刚力, 杨家本. 自适应调整信息素的蚁群算法 [J]. 信息与控制, 2002, 31(3): 198 ~ 201.
- [8] Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behaviour [J]. Nature, 2000, 406(6): 39 ~ 42.
- [9] Dorigo M, Gambardella L M. Solving symmetric and asymmetric TSPs by ant colonies [A]. Proceedings of the IEEE Conference on Evolutionary Computation [C]. Nagoya, Japan; 1996. 622 ~ 627.
- [10] Thomas S, Holger H H. MAX - MIN ant system [J]. Future Generation Computer Systems, 2000, 16(8): 889 ~ 914.

## 作者简介

段海滨 (1976 -), 男, 博士生. 研究领域为智能进化算法, 控制与仿真等。

王道波 (1957 -), 男, 教授, 博士生导师. 研究领域为控制与仿真, 智能进化算法, 无人机控制技术。