

文章编号: 1001-0920(2008)02-0225-04

动态调整选择策略的改进蚁群算法

郑 松, 侯迪波, 周泽魁

(浙江大学 a. 工业控制国家重点实验室, b. 控制科学与工程学系, 杭州 310027)

摘 要: 针对蚁群算法存在停滞现象的缺点, 提出一种动态调整的选择策略以强化其全局搜索能力. 改进的选择策略通过适当刺激蚂蚁尝试具有较弱信息素解, 以提高所得解的全局性. 给出了新算法仿真实验步骤, 并将改进后的蚁群算法与传统蚁群算法分别应用于旅行商问题(TSP)进行仿真实验. 仿真结果表明, 改进后的算法具有优良的全局优化性能, 可抑制算法过早收敛于次优解, 有效防止了停滞现象, 收敛速度也大大加快.

关键词: 蚁群算法; 停滞现象; 信息素; 全局优化

中图分类号: TP13

文献标识码: A

Ant colony algorithm with dynamic transition probability

ZHENG Song, HOU Di-bo, ZHOU Ze-kui

(a. National Laboratory of Industrial Control Technology, b. Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China. Correspondent: ZHENG Song, E-mail: szhengzju@163.com)

Abstract: Aiming at the disadvantage of stagnation behavior in ant colony algorithm (ACA), the dynamic transition is presented. The dynamic transition facilitates the exploration by increasing the probability of selecting solution components with low pheromone trail. Then a programming process is presented. An example of traveling salesman problem 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 faster the convergence speed, and it can avoid premature convergence of ACO.

Key words: Ant colony optimization; Stagnation behavior; Pheromone trail; Global optimization

1 引言

动物界的智能行为一直是科学家灵感的源泉, 近年来, 群居的昆虫表现出来的集体智能吸引了研究者的注意. 科学家发现, 昆虫在群落一级上的合作基本上是自组织的, 在许多场合中尽管这些合作可能很简单, 但它们却可以解决许多复杂的问题. 蚁群算法就是利用群集智能解决组合优化问题的典型例子. 蚁群算法(ACA)是继模拟退火算法、遗传算法、禁忌搜索算法、人工神经网络算法等元启发式搜索算法以后的又一种应用于组合优化问题的启发式搜索算法^[1,2].

蚂蚁系统是在 20 世纪 90 年代初被提出的, 并成功解决了旅行商问题(TSP)^[3,4]. 蚂蚁系统仍存在一些缺陷, 所以后来又出现了很多改进的蚁群算法^[5-8], 如带精英策略的蚂蚁系统(ASe), 蚁群系统(ACS)和最大最小蚁群系统(MMAS).

蚁群算法不仅能够智能搜索、全局优化, 而且具有稳健性(鲁棒性)、正反馈、分布式计算、易与其他算法结合等特点. 因此, 蚁群算法的问世为诸多领域解决复杂优化问题提供了有力的工具^[9-12]. 但蚁群算法在构造解的过程中, 利用了随机选择策略, 这种选择策略使得进化速度较慢, 容易出现停滞现象. 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. 本文针对这一问题, 引入了动态调整的选择策略, 以提高蚁群算法的全局搜索能力和搜索速度.

2 蚁群系统数学模型

1996 年, Gambardella 等提出一种修正的蚁群算法, 被称为蚁群系统(ACS)^[3,6]. 为了便于理解, 以求解平面上 n 个城市的 TSP 问题($0, 1, \dots, n-1$ 表示城市序号)为例说明蚁群系统模型. 人工蚂蚁的数量为 m . 每个人工蚂蚁的行为符合下列规律: 根据

收稿日期: 2006-11-16; 修回日期: 2007-05-15.

基金项目: 国家自然科学基金重点项目(NSFC-50505045).

作者简介: 郑松(1982—), 男, 贵州六盘水人, 博士生, 从事计算机控制与优化技术的研究; 周泽魁(1946—), 男, 江西临川人, 教授, 博士生导师, 从事计算机控制与优化技术等研究.

路径上的信息素浓度,以相应的概率来选取下一步路径.用一个数据结构(tabu list)来控制这一点;当完成了一次循环后,根据整个路径长度来释放相应浓度的信息素,并更新走过的路径上的信息素浓度.

现用 $\tau_{ij}(t)$ 表示在 t 时刻,边 (i, j) 上的信息素浓度.当蚂蚁完成一次循环之后,相应边上的信息素浓度必须进行更新处理.即模仿人类记忆的特点,对旧的信息进行削弱,同时将最新的蚂蚁访问路径的信息加入到 $\tau_{ij}(t)$. 蚁群系统有两种信息素更新方式:局部更新规则和全局更新规则.

局部更新规则是在每只蚂蚁完成一次转移后执行.随着时间的推移,以前留下的信息逐渐消逝,两个元素(城市)状态之间的局部信息素根据下式调节:

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

式中: $\rho \in [0, 1]$, L_m 是由最近的邻域启发产生的一个路径长度.

全局更新不再是对所有的蚂蚁,而是仅对全局最优的蚂蚁使用.全局调整规则如下:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + (1 - \rho) \cdot \tau_{gb}, \quad \tau_{gb} = \begin{cases} (L_{gb})^{-1}, & (i, j) \text{ 全局最优路径;} \\ 0, & \text{否则.} \end{cases} \quad (2)$$

式中: ρ 为一个取值范围在 $0 \sim 1$ 之间的常数系数, L_{gb} 为到目前为止找出的全局最优路径.

转移概率 $P_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的概率,有

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}{\sum_{j \in allowed_k} \tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta}, & j \in allowed_k; \\ 0, & \text{其他.} \end{cases} \quad (3)$$

其中: $\eta_{ij} = 1/C_{ij}$, C_{ij} 为经过路径 (i, j) 所需的花费; α 和 β 两个参数分别用来控制信息素和路径长度的相对重要程度; $allowed_k = (0, 1, \dots, n - 1)$ 表示蚂蚁 k 下一步允许选择的城市.

初始时刻, m 个蚂蚁被放置在不同的城市上,赋予每条边上的信息素浓度是相等的,浓度为 $\tau_{ij}(0) = C$ (C 为常数).每个蚂蚁的 tabu list 的第 1 个元素赋值为它所在的城市.当蚂蚁们完成了一次完整的寻径过程后,计算 τ_{ij}^k , 并且更新每条边上的信息素浓度;然后开始新一轮的循环.当循环的次数达到事先定义好的 $N_{c_{max}}$ 时或者所有的蚂蚁都选择

了同一种路径方式时,整个算法终止.

3 利于全局搜索的改进蚁群算法

3.1 动态调整的选择策略

停滞现象是造成蚁群算法不足之处的根本原因.本文对算法进行改进,在确定性选择和随机选择相结合的基础上,通过在搜索过程中动态调整转移概率构建更利于全局搜索的选择策略.

在进化过程中,路径上的信息素不断改变,搜索过的较好解的信息素得到加强,下次迭代被选择的可能性增大.而有些本是较优解但由于起始阶段较少蚂蚁经过而可能逐渐被遗忘,影响算法的全局搜索能力.如果在选择策略中,考虑适当刺激蚂蚁尝试那些很少经过的路径,有利于对解空间更全面地搜索,从而可以有效地克服基本蚁群算法的不足.本文改进后的选择策略为

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta \cdot Q_c(i, j)}{\sum_{j \in allowed_k} \tau_{ij}(t)^\alpha \cdot \eta_{ij}^\beta \cdot Q_c(i, j)}, & j \in allowed_k; \\ 0, & \text{其他.} \end{cases}$$

其中

$$Q_c(i, j) = \frac{m \cdot N_c}{m \cdot N_c + \rho \cdot Q_c(i, j) \cdot (i, j) / \max_{(i, j)} Q_c(i, j)} \quad (4)$$

m 是蚂蚁个数, N_c 表示当前迭代次数, $\max_{(i, j)} Q_c(i, j)$ 代表启发函数 $Q_c(i, j)$ 的最大值, $Q_c(i, j)$ 是从第 1 次迭代开始,目前经过路径 (i, j) 的蚂蚁总个数.同时考虑了参数 ρ 和 Q_c , 当迭代前期趋向次优解时,虽然次优解上的信息素不断增强,但通过它的蚂蚁个数 $Q_c(i, j)$ 不断增大,其值不断减小.这样在选择路径时,可以抑制信息素过度剧增而导致过早收敛,有利于算法的全局搜索.

本文在研究选择策略过程中,曾考虑另外一种形式的 $Q_c(i, j)$, 即不从第 1 次迭代开始计算,而是当前迭代中经过路径 (i, j) 的蚂蚁个数,这里用 $Q_{ci}(i, j)$ 表示.经过针对 Oliver30 和 Eil51 的仿真数据对比,采用考虑全局的 $Q_c(i, j)$ 比 $Q_{ci}(i, j)$ 的优化效果好,优化的平均结果更好,见表 1. 所以本文采用式 4 中的 $Q_c(i, j)$.

参数 ρ 具有上下限,由于 $Q_c = m \cdot N_c / (m \cdot N_c + \rho \cdot Q_c(i, j) \cdot (i, j) / \max_{(i, j)} Q_c(i, j))$, 从该表达式可知 $1 \leq Q_c \leq \frac{1}{1 - \rho}$.

参数 ρ 可以调整 Q_c 的强度, ρ 越小, $\frac{1}{1 - \rho}$ 越大,表示 Q_c 在选择策略中的权重越小,当 $\rho = 0$ 时, $Q_c = 1$, 改进的选择策略不考虑 Q_c , 与式 (3) 相同.以 Oliver30 和 Eil51 作为算例,针对不同参数 ρ 进行仿真,结果如表 2. 仿真结果表明,当 $\rho = 0.5$ 时,改进

选择策略的求解性能较好. 因此, 后面的仿真均采用 $\alpha = 0.5$.

参数 Q_c 包含信息素无法表示的信息, 是对以往选择策略很好的补充. 并且, 在迭代初期, 由于路径之间信息素 差异不明显, 在 Q_c 中, Q_c 的作用比较明显, 刺激蚂蚁去尝试搜索新的路径, 可以较好保证搜索的全局性, 防止过早收敛于次优解; 在迭代中后期, 进化方向已基本确定, 此时路径之间存在明显的信息素差异, 由于 Q_c 有下限限制, 它的作用已相当微弱, 不会影响整个蚁群算法的收敛性.

表 1 改进的选择策略采用不同形式 Q_c 效果对比

旅行商问题	ACS 算法		采用 Q_{c_i} 形式的改进蚁群算法		采用 Q_c 形式的改进蚁群算法	
	平均值	最优值	平均值	最优值	平均值	最优值
Oliver30	426.8	423.7	425.2	423.7	424.7	423.7
Eil51	445.7	441.9	443.6	436.0	440.1	436.0

注: 平均值指算法重复 20 次所得路径的平均长度(后同).

表 2 参数 α 对改进选择策略效果的影响

	0	0.2	0.5	1.0	1.5
Oliver30	426.8	425.6	424.7	425.1	425.8
Eil51	445.7	443.8	440.1	441.5	442.9

注: 表中显示结果为平均路径长度.

3.2 改进选择策略对收敛速度的影响

研究改进选择策略对收敛速度的影响, 首先要分析传统蚁群算法中局部更新规则与改进选择策略

的联系与区别. 局部更新规则是在蚂蚁完成一次转移后执行, 实质是避免后来的蚂蚁重复相同路径, 本身有抑制收敛同一路径的作用. 虽然改进后的选择策略和局部更新规则对抑制过早收敛都有效, 但它们的机理并不相同, 对信息素影响不同. 局部更新规则对信息素是直接的永久性更改, 而选择机制对信息素无直接性的更改. 另外, 动态调整选择机制是在选择路径时开始刺激蚂蚁搜索新路径; 局部更新规则是在蚂蚁经过路径之后, 再通过调整信息素来反馈引导蚂蚁搜索其他路径. 于是选择机制抑止过早收敛比局部更新具有超前性, 在收敛速度上即表现为具有更快的收敛速度. 传统算法大多采用局部更新规则来抑制早熟, 改进蚁群算法虽然没有完全放弃局部更新法则, 但是动态调整的选择策略的引入, 替代了局部更新法则的一部分功效. 动态调整的选择策略对局部更新法则功效的削弱, 以及新选择策略的超前性, 对蚂蚁的刺激也具有相应的超前性. 这些正是改进蚁群算法提高收敛速度的根本原因. 表 3 中结果表明, 一方面, 动态调整的选择策略比局部更新法则具有超前性, 能够加快算法的收敛速度, 在没有局部更新规则情形下, 它所需要的迭代次数会有显著减少; 另一方面, 两者相结合比单纯依靠其中一种来避免停滞更为有效, 且动态调整的选择策略的引入的确加快了算法的收敛速度.

表 3 动态选择策略与局部更新规则效果对比

旅行商问题	ACS 算法			采用动态调整选择策略, 没有局部更新规则的蚁群算法			改进的蚁群算法		
	平均值	最优值	迭代次数	平均值	最优值	迭代次数	平均值	最优值	迭代次数
Oliver30	426.8	423.7	1 259	427.6	424.6	89	424.7	423.7	521
Eil51	445.7	441.9	1 585	457.1	449.1	179	440.1	436.0	1 023

3.3 算法仿真步骤

改进蚁群算法编程仿真步骤如下:

Step1: 初始化. 令 $t = 0, N_c = 0, Q_c = 0$, 初始每条边上的信息素 $\tau_{ij}(0)$, 并且 $\tau_{ij} = 0$, 随机放置 m 个蚂蚁到 n 个城市上.

Step2: 令 $s = 1, s$ 是 tabu list 的下标, 把 m 个蚂蚁的初始城市号码放置到 $\text{tabu}(s)$ 中.

Step3: 根据动态调整的选择策略(式 4) 来选择下一步应该到达的城市, 将第 k 个蚂蚁移到城市 j , 并将 j 插入到 $\text{tabu}(s)$ 中, 同时对第 k 个蚂蚁经过的路径 (i, j) 进行信息素局部更新(式(1)), 并且更新 Q_c , 即 $Q_c(i, j) = Q_c(i, j) + 1$. 如此重复直到每只蚂蚁都完成路径搜索.

Step4: 计算每只蚂蚁的总路线长度, 更新找到的最短路径. 对最优蚂蚁经过的路径进行全局更新(式(2)).

Step5: 令 $N_c = N_c + 1, \tau_{ij} = 0$.

Step6: 如果 $N_c > N_{c_{\max}}$ 或者所有的蚂蚁选择同一条路径, 则循环结束, 同时打印全局优化的最佳路径; 否则跳转到 Step2.

4 仿真结果

将 Dorigo 等人提出的几种蚁群算法和本文改进的蚁群算法, 分别应用于 TSPLIB 中的几个典型 TSP 问题进行仿真实验.

本文设置算法参数 $m = 10, \alpha = 1, \beta = 5, \rho = 0.5$.

表 4 算法仿真对比结果

旅行商问题	ACS 算法	改进后蚁群算法	AS 算法
20cities	24.96	24.68	26.34
Oliver30	426.81	424.19	431.9
Eil51	445.71	442.06	459.12
Eil101	648.1	641.2	665.5

注: 表中显示结果为平均路径长度.

表 5 ACS 算法与改进蚁群算法对比详表

旅行商问题	算法名称	最短路径长度	平均路径长度	最差路径长度	平均迭代次数
20cities	ACS 算法	24.52(0.0 %)	24.96	26.47	349
	改进后蚁群算法	24.52(0.0 %)	24.68	24.95	323
Oliver30	ACS 算法	424.46(1.0 %)	426.81	434.63	1 259
	改进后蚁群算法	423.74(0.8 %)	424.19	425.27	521
Ei151	ACS 算法	431.72(1.3 %)	445.71	458.5	1 585
	改进后蚁群算法	430.2(1.0 %)	442.06	450.0	1 023
Eil101	ACS 算法	638.43(1.5 %)	648.12	662.3	4 415
	改进后蚁群算法	635.97(1.1 %)	641.24	654.1	3 195

0.9, $Q = 10$, $\alpha = 0.9$, $\beta = 0.5$, 最大迭代次数 $N_{c_{max}} = 100 \cdot n$. 3 种蚁群算法的仿真结果见表 4.

由表 4 的仿真结果可知,对于各种旅行商问题,改进后的蚁群算法要比其他蚁群算法(ACS,AS)能够寻找到更短的路径,说明改进后的蚁群算法具有更好的全局搜索能力.为清楚地了解动态调整选择策略的效果,本文将改进的蚁群算法与 ACS 算法作了详细对比(表 5),在处理 Oliver30 问题中,改进蚁群算法平均路径长度为 424.19,较优于 ACS 算法的 426.81,并且寻找到的最短路径也要好于 ACS 算法;另外,它的平均迭代次数大约是 ACS 算法的一半,收敛速度得到加快.为更直观地体现改进算法具有较快的收敛速度,本文将两种算法的收敛过程体现在图 1 上,改进蚁群算法在迭代 464 次后收敛于 423.74,而传统算法在迭代 962 次后才最后收敛,表明改进后蚁群算法收敛速度快于 ACS 算法.对其他旅行商问题的仿真数据也有类似的结论.大量仿真结果表明采用动态调整的选择策略后,蚁群算法的全局搜索能力和优化速度都有较大的改善.

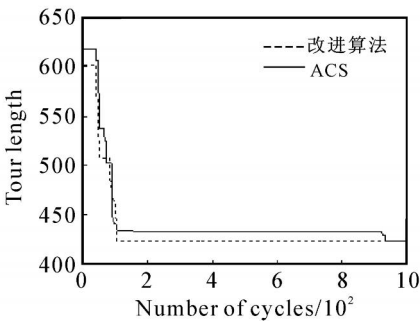


图 1 改进蚁群算法与 ACS 算法收敛速度对比

5 结 语

本文提出动态调整的选择策略,使蚁群算法的全局优化能力得到显著提高,而且能够有效地克服传统蚁群算法收敛速度慢、易过早收敛等缺陷,优化性能得到了有效的改善.蚁群算法不象其他启发式算法那样已形成系统的分析方法,具有坚实的数学基础.但从当前的应用效果看,这种新型系统寻优思想无疑具有十分光明的前景,更多深入细致的工作还有待于进一步展开.

参考文献(References)

[1] Dorigo M, Stützle T. Ant colony optimization [M]. Cambridge: MIT Press/Bradford Books, 2004.

[2] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics—Part B, 1996, 26(1): 29-41.

[3] Gambardella L M, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies[C]. Proc of the IEEE Int Conf on Evolutionary Computation (ICEC '96). Piscataway: IEEE Press, 1996: 622-627.

[4] Dorigo M. Optimization, learning, and natural algorithms[D]. Milan: Politecnico di Milano, 1992.

[5] Bahreininejad A, Hesamfar P. Subdomain generation using emergent ant colony optimization[J]. Computers and Structures, 2006, 84(5): 1719-1728.

[6] Issmail Ellabib, Paul Calamai, Otman Basir. Exchange strategies for multiple ant colony system [J]. J of Information Sciences, 2006, 3(1): 46-63.

[7] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.

[8] Thomas Stutzle, Holger H Hoos. MAX-MIN ant system [J]. Future Generation Computer Systems, 2000, 16(8): 889-914.

[9] Shu-Chuan Chu, John F Roddick, Jeng-Shyang Pan. Ant colony system with communication strategies[J]. Information Sciences, 2004, 167(7): 63-76.

[10] Paulo Henrique Siqueira, Maria Teresinha Arns Steiner, Sergio Scheer. A new approach to solve the traveling salesman problem [J]. Neurocomputing, 2006, 70(4): 1013-1021.

[11] Onwubolu G C, Clerc M. Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization [J]. Int J of Production Research, 2004, 42(3): 473-491.

[12] Bianchi L, Knowles J, Bowler J. Local search for the probabilistic traveling salesman problem: Correction to the 2-p-opt and 1-shift algorithms[J]. European J of Operational Research, 2005, 162(1): 206-219.