第34卷 第19期 计 算 机 工 程 2008年10月
Vol.34 No.19 Computer Engineering October 2008
·人工智能及识别技术· 文章编号：1000—3428(2008)19—0200—03 文献标识码：A 中图分类号：TP18

基于改进ACS-3-opt蚁群算法的TSP

马文霜[1]，张洪伟[2]
(1. 四川大学计算机学院，成都 610064；2. 成都信息工程学院计算机系，成都 610225)

摘 要：在ACS-3-opt算法求解中，大规模TSP问题易于停滞。该文提出一种改进的算法，在ACS-3-opt算法停滞后，自适应地调整具有局部搜索能力蚂蚁的数量，并通过提高最小信息素的阈值扩大搜索空间，当算法再次停滞时，增强算法两次停滞时最优路径的公共路径上的信息素，为算法的运行提供较好的初始信息，并引导算法朝最优解的方向进行求解。大中型规模TSP问题的求解结果表明，该算法能够有效地跳出局部最优，解的质量优于ACS-3-opt算法。

关键词：蚁群算法；信息素阈值；公共路径

TSP Based on Improved ACS-3-opt Ant Colony Algorithm

MA Wen-shuang[1], ZHANG Hong-wei[2]
(1. College of Computer, Sichuan University, Chengdu 610064;
2. Department of Computer, Chengdu University of Information Engineering, Chengdu 610225)

【Abstract】 To overcome the drawbacks of solving middle-scale and big-scale TSP problem such as easy to stagnation, when using ACS-3-opt(Ant Colony System plus 3-opt), an improved algorithm is proposed, whose cores are that after stagnation of ACS-3-opt increase the quantity of ants with local search ability and enhance minimum pheromone threshold to enlarge search space. When algorithm stagnates again, it reinforces the pheromone of the common path of the two best tours generated. The reinforcement for the common path provides a good initial information for later running, guiding the algorithm to the best tour. The results of solving middle-scale and big-scale TSP problem show that the algorithm can skip from local optimum effectively and the solution is better than ACS-3-opt.

【Key words】 ant colony algorithm; pheromone threshold; common path

1 概述

基本蚁群算法[1]是由M. Dorigo等人首先提出的一种新型的仿生优化算法，并成功地解决了旅行商问题(TSP)、作业调度问题(JSP)等组合优化问题，取得了较好的参数。该算法采用了分布式并行计算机制，易于与其他方法结合，而且具有较强的鲁棒性。然而，基本蚁群算法存在搜索时间过长、易于停滞的缺点。当问题规模扩大时，所得的解效果较差。为了克服这些缺点，各国学者提出了许多改进的蚁群优化算法。如B. Bullnheimer等人提出的RAS(Rank-based Ant System)算法[2]，RAS算法在每次迭代后，对所有蚂蚁的遍历路径长度由小到大排序，选取路径最短的 ω 只蚂蚁来更新路径的信息素浓度，相比基本蚁群算法，RAS能够加速收敛但是也容易停滞，陷入局部最优；O. Cordon等人提出了BWAS (Best-Worst Ant System)算法[3]，该算法对基本蚁群算法做了如下改进：在每次迭代后找到迄今为止最优路径和本次迭代最差路径，增加最优路径上的信息素，减少最差路径的信息素；算法停滞后重置信息素；变换信息素矩阵等措施来加速算法收敛的同时避免停滞。相比RAS, BWAS，该算法能够找到更高质量的解。但是对于中大规模的TSP问题效果仍不理想；M. Dorigo等人提出了ACS(Ant Colony System)算法[4]，该算法对基本蚁群算法的路径选择策略做了调整，并且对信息素采取全局更新和局部更新相结合的策略，ACS算法使得蚁群在开发新路径和利用已有路径之间达到更好的平衡，获得的解具有很好的全局性；T. Stutzle等人提出的Max-Min Ant System(MMAS)算法[5]，MMAS算法的基本思想是将路径的信息素限制在 $(\tau_{\min},\tau_{\max})$ 之间，来抑制算法早熟，同时在算法停滞后重置信息素，以避免无效的迭代。

然而，单纯的依靠蚁群的自启发特性来构造路径并不能提高求解问题的规模。在应用启发算法解决组合优化问题的研究中发现，将蚁群算法和局部优化算法结合起来，在早期，局部优化算法为蚁群算法提供较高质量的解来指导蚁群算法的学习，蚁群算法在此基础上构造新的初始解来引导以后的局部搜索，如此反复可以显著地提高解的质量。本文在对增加了3-opt局部搜索能力的ACS算法(ACS-3-opt)的原理和性能做了详细分析后提出一种改进算法，并实现该算法。对中大规模TSP问题的求解结果表明，该算法所获得的解的质量优于ACS-3-opt。

2 ACS-3-opt算法对基本蚁群算法的改进

下面以平面上 n 个城市的TSP问题为例，说明ACS-3-opt对基本蚁群算法的改进。

2.1 路径选择规则

处于第 i 个城市的蚂蚁 k 将按照下面的规则选择下一个城市：

$$s=\begin{cases}\arg\max\limits_{j\in allowed_i}\{[(\tau_{ij})(\eta_{ij}^{\beta})]\} & q\le q_0\\ S \text{ 否则}\end{cases} \tag{1}$$

作者简介：马文霜(1983－)，男，硕士研究生，主研方向：智能信息系统，数据库与计算机网络；张洪伟，教授、博士后

收稿日期：2008-02-27 **E-mail：**mws_007@163.com

$$S=\begin{cases}\dfrac{\tau_{ij}\eta_{ij}^{\beta}}{\sum\limits_{j\in allowed_i}\tau_{ij}\eta_{ij}^{\beta}} & j\in allowed_i\\ 0 \text{ 否则}\end{cases} \tag{2}$$

$$\eta_{ij}=\frac{1}{d_{ij}} \tag{3}$$

其中，τ_{ij} 表示第i个城市到第j个城市路径上的信息素浓度；式(3)为启发函数，d_{ij}为两个城市之间的距离；β表示能见度的相对重要性，其值越大，蚂蚁越容易向离自己越近的城市移动；q是一个在[0,1]之间均匀分布的随机数；q_0 是一个给定的位于[0,1]之间的数。

2.2 信息素局部更新

蚂蚁从一个城市转移到另一个城市后需要按照下面的公式更新这两个城市之间路径的信息素：

$$\tau_{ij}^{\text{new}}=(1-\rho)\bullet\tau_{ij}^{\text{old}}+\rho\tau_0 \tag{4}$$

$$\tau_0=(nL_{nn})^{-1} \tag{5}$$

其中，L_{nn} 表示按照优先选取最近城市的原则对所有的城市做遍历后得到的一个回路的长度；ρ 表示局部信息素挥发因子，是一个给定的位于[0,1]之间的数。由式(4)可知，每当有一只蚂蚁选择了一条路径转移到下一个城市后，这条路径上的信息素就会减少，那么后来的蚂蚁选择这条路径的概率就会降低，相应地，蚂蚁选择其他路径的概率增加，增加蚂蚁发现更好路径的概率。

2.3 路径局部优化

在 ACS-3-opt 算法中，当所有的蚂蚁对城市完成遍历后，可以采用 3-opt 算法对蚂蚁走过的路径进行优化。

2.4 信息素全局更新

当所有的蚂蚁对城市完成遍历后，需要对路径上的信息素进行全局更新。只有迄今为止路径最短的蚂蚁能够在它经过的路径上留下信息素，信息素更新的规则如下：

$$\tau_{ij}^{\text{new}}=(1-\alpha)\bullet\tau_{ij}^{\text{old}}+\alpha\Delta\tau_{ij} \tag{6}$$

$$\Delta\tau_{ij}=\begin{cases}(L_{global-best})^{-1} & \text{if}(i,j)\in\text{全局最优}\\ 0 \text{ 否则}\end{cases} \tag{7}$$

3 改进的 ACS-3-opt 算法

3.1 ACS-3-opt 算法的改进

虽然 ACS-3-opt 算法可以在利用已有的路径信息和开发新路径之间达到一个很好的平衡，所获得解的质量较高，但是在解决中大规模 TSP 问题时易陷入局部最优。本文主要从跳出局部最优的角度对 ACS-3-opt 算法做如下改进：

3.1.1 蚂蚁的数量的自适应调整

当算法停滞后，按照式(8)增加蚂蚁的数量：

$$m^{\text{new}}=m^{\text{old}}+\Delta m \tag{8}$$

由于m和 ρ 存在如下关系[4]：

$$m=\frac{\lg(\varphi_1-1)-\lg(\varphi_2-1)}{q_0\lg(1-\rho)} \tag{9}$$

得 $\rho=1-\sqrt[mq_0]{\dfrac{\varphi_1-1}{\varphi_2-1}}$ 且 $\dfrac{\varphi_1-1}{\varphi_2-1}$ 的最优值为 0.4，因此

$$\rho^{\text{new}}=1-\sqrt[mq_0]{0.4} \tag{10}$$

当蚂蚁的数量变化时，需按照式(10)来调整 ρ 的值。

3.1.2 最小信息素阈值的自适应调整

当算法停滞后，在增加蚂蚁的同时，按照式(8)将所有路径上的信息素浓度置为

$$\tau_0^{\text{new}}=\tau_0^{\text{old}}+\delta(nL_{global-best})^{-1} \tag{11}$$

其中，δ 是一个给定的位于[0,1]之间的数。在 MMAS 算法中，每条路径上信息素浓度限制在 $(\tau_{\min},\tau_{\max})$ 之间，避免了某些路径由于前期的漏选而到后期无法被选中。因此，从长远来讲，信息素阈值解决了过早停滞。受 MMAS 算法的启发，在改进的算法中每当出现停滞时，可以自适应地提高信息素浓度的最小阈值来增大搜索空间，以期发现更优解。

3.1.3 更新公共路径的信息素

调整参数后算法再次停滞时，如果获得的最优解比原来的还差(图 1、图 2 为算法调整参数前后停滞时得到的最优路径，图 2 的路径较差)，那么找到调整参数前后最优路径的公共路径(见图 3)，根据式(6)，加强公共路径上的信息素，并且将其他路径上的信息素置为初始值公共路径，是算法的两次运行(尽管参数不同)的最优解都走过的路径，因此，该路径有很大可能是实际最优路径的组成部分，见图 3、图 5。加强公共路径的信息素使得算法在有公共路径连接的城市处选择下一条路径时将以较大的概率选择公共路径。将算法朝最优路径的方向引导。算法的后续运行实质上是从处于非公共路径的城市中构造一条最短路径将它们和公共路径相连组成一条回路，使得在算法停滞前易于发现更好的路径。由图 3、图 4 可知，由于公共路径上的强信息素的引导改进算法找到了更优的路径，这条路径基本包含了公共路径，同时这条路径已经十分接近实际最优路径，见图 4、图 5。

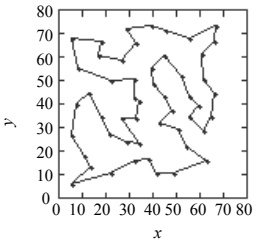

图 1 eil51 停滞时的路径 (长度 430)

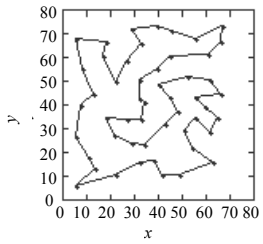

图 2 eil51 再次停滞时的路径 (长度 432)

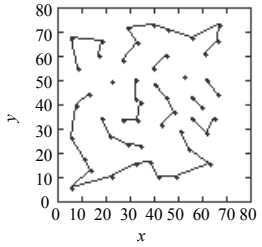

图 3 两次停滞的公共路径

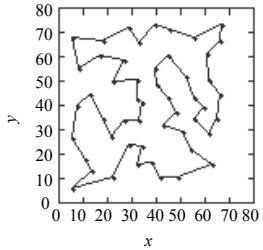

图 4 从公共路径找到的路径 (长度 428)

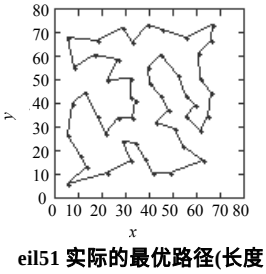

图 5 eil51 实际的最优路径(长度 426)

3.2 改进的 ACS-3-opt 算法伪代码

改进的 ACS-3-opt 算法伪代码如下：

设置 NCmax 为最大迭代次数；

设 m 为蚂蚁的数目；

n 为城市数目；

设置 so_far_best 为迄今为止最好的路径；
reinitial_best 为信息素重置后找到的最好路径；
由式(5)计算出τ_0的值，作为路径的信息素初始值
while(NC < NCmax) do{
for(i=1 to m){
将第 i 只蚂蚁随机的放到某一个城市；
修改搜索禁忌表将其他的 n-1 个城市置为未访问；
for(j=1 to n){
由式(1)选择下一个需要访问的城市；
修改搜索禁忌表将所访问的城市置为已访问过；
按照式(4)局部更新刚刚走过路径的信息素
}
}
使用 3-opt 算法对当前蚂蚁构造的回路做局部优化
更新 so_far_best 和 reinitial_best
使用式(6)对 reinitial_best 上的信息素做全局更新
if(分支因子<1.000 01 && 经 1 000 次迭代仍不能找出比 reinitial_best 更好的路径){ //算法停滞
if(so_far_best 长度 == reinitial_best 长度){
//第 1 次停滞或信息素重置后找到了更好的路径
按照式(8)更新蚂蚁的数量
按照式(10)重置局部信息挥发因子 ρ 的值
按照式(11)计算最小信息量的阈值 τ_0^{new}
将每条路径上的信息素置为 τ_0^{new}
else{//如果再次停滞后并且找到的路径比原来还差则增强公共 //路径上的信息素
求出 so_far_best 和 reinitial_best 的公共路径，按照式(6)增强这条路径上的信息素
将其他路径的信息素置为 τ_0^{new}
}
reinitial_best=优先选取最近城市获得的一条回路
}//end if 算法停滞
}//while 循环结束
打印 so_far_best

4 实验结果

改进算法采用 ANSI C 实现，实验运行环境：赛扬 1.6 GHz CPU, 256 MB 内存，Windows XP 操作系统。参数采用文献[4]的建议值：$\beta = 2, q_0 = 0.9, \alpha = \rho = 0.1, \tau_0 = (nL_{nn})^{-1}$，$\Delta m$ 和 λ 选取实验时取得最好效果的值：Δm =5, λ =0.5，每个算例运行 15 次，每次作 10 000 次迭代。

表 1 以 pr1002.tsp 和 gr666.tsp 为例，列举出了改进算法从 ACS-3-opt 的停滞路径出发(在算法开始运行时，将 ACS-3-opt 算法停滞时的路径作为当前的最优路径，更新这条路径上的信息素)所获得的解。由表 1 可知，改进算法从 ACS-3-opt 的停滞路径出发可以找到更优的路径，证明了改进算法能够有效地跳出局部最优。

表 1 从 ACS-3-opt 的停滞路径出发找到新路径

问题名称	最优路径	停滞路径	改进算法路径
pr1002.tsp	259 045	259 410	259 045
pr1002.tsp	259 045	259 769	259 045
gr666.tsp	294 358	294 423	294 379
gr666.tsp	294 358	294 852	294 476

表 2 对比了 ACS-3-opt 算法和改进算法的求解效果。其中，偏移率=(求解平均值-已知最优解)/已知最优解，由表 2 可知，改进算法解的质量优于 ACS-3-opt。

表 2 ACS-3-opt 和改进算法的求解效果对比

问题名称 (城市数目)	已知最优解	ACS-3-opt 算法			改进算法		
		最优值	平均值	偏移率/(%)	最优值	平均值	偏移率/(%)
lin318.tsp(318)	42 029	42 029	42 029.0	0.000	42 029	42 029.0	0.000
att532.tsp(532)	27 686	27 693	27 718.2	0.116	27 686	27 686.0	0.000
rat783.tsp(783)	88 06	8 818	8 837.9	0.362	8 806	8 806.0	0.000
gr666.tsp*(666)	294 358	294 358	294 472.0	0.038	294 358	294 364.3	0.002
pr1002.tsp*(1 002)	259 045	259 045	259 299.8	0.098	259 045	259 118.0	0.028
pr2392.tsp*(2 392)	378 032	378 873	379 174.2	0.301	378 127	378 581.4	0.145

5 结束语

本文针对 ACS-3-opt 算法后期会停滞陷入局部最优这一不足，从跳出局部最优的角度对 ACS-3-opt 算法进行改进，改进算法融合了 MMAS 算法的优点，自适应地调整参数来扩大搜索空间，以达到发现更优解的目的，并且通过增强算法两次停滞时公共路径上的信息素来引导算法跳出局部最优，增强了算法的全局寻优能力。实验证明，在改进算法求解中，大规模 TSP 问题的效果十分理想。

参考文献

[1] Dorigo M, Maniezzo V, Colomi A. The Ant System: Optimization by A Colony of Cooperating Angents[J]. IEEE Transactions on Systems, Man & Cybernetics, 1996, 26(1): 29-41.

[2] Bullnheimer B, Hartl R F, Strauss C. A New Rank-based Version of the Ant System: A Computational Study[J]. Central European Journal for Operations Research and Economics, 1999, 7(1): 25-38.

[3] Cordon O. A New ACO Model Integrating Evolutionary Computation Concepts: the Best-worst Ant System[Z]. 2000: 22-29.

[4] Dorigo M, Gambardella L M. Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[5] Stutzle T, Hoos H H. Max-Min Ant System and Local Search for the Traveling Salesman Problem[C]//Proceedings of IEEE International Conference on Evolutionary Computation. New York, USA: IEEE Press, 1997: 309-314.

（上接第 186 页）

参考文献

[1] Cooper G R J, Cowan D R. The Detection of Circular Features in Irregularly Spaced Data[J]. Computers and Geosciences, 2004, 30(1): 101-105.

[2] Tsai R Y. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Miami Beach, USA: IEEE Press, 1986.

[3] Li Hyosoon, Song Youngshik. SFIDA: A Software Implemented Fault Injection Tool for Distributed Dependable Applications[C]//Proceedings of the 4th International Conference/Exhibition on High Performance Computing in the Asia-pacific Region. Beijing, China: IEEE Computer Society, 2000.

[4] Edward P, Owen L, Mitchell R. Subpixel Measurement Using a Moment-based Edge Operator[J]. IEEE Trans. on PAMI, 1989, 11(12): 1293-1309.

[5] Benso A, Prinetto P. A Fault Injection Environment for Microprocessor-based Boards[C]//Proceedings of the IEEE International Test Conference. Washington, USA: IEEE Press, 1998.