

带有近邻选择策略和遗传算子的蚁群算法

陶利民^{1,2}, 郭俊恩³

(1 杭州师范大学 信息科学与工程学院, 浙江 杭州 310036; 2 浙江工业大学 信息工程学院, 浙江 杭州 310032;
3 洛阳理工学院 计算机与信息工程系, 河南 洛阳 471023)

摘 要: 针对蚁群算法的不足, 文中提出一种带有近邻节点选择策略和遗传算子的蚁群算法. 在算法执行时, 每只蚂蚁选择下一个城市采用近邻选择策略, 在每轮循环结束时, 对较优个体进行交叉运算的操作方法, 以期提高蚁群算法的收敛速度. 实验结果表明该算法是有效的.

关键词: 蚁群算法; 收敛速度; 近邻选择策略; 遗传算子; 交叉运算

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2010)04-0179-03

Ant Colony Optimization Algorithm Based on the Nearest Neighbor Node Choosing Strategy and Genetic Operator

TAO Li-min^{1,2}, GUO Jun-en³

(1 School of Information Science and Engineering, Hangzhou Normal University, Hangzhou 310036, China;
2 School of Information Engineering, Zhejiang University of Technology, Hangzhou 310032, China; 3 Department of
Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang 471023, China)

Abstract: To aim at the lacks, an improved ACO is presented. This algorithm is an ACO having the nearest neighbor node choosing strategy and genetic operator. The ACO's convergence speed is expected to be raised by means of using the nearest neighbor node choosing strategy when choosing the next city to move, and making crossover operator among better results at the end of each iteration. The experiment results indicate that the improved ACO is effectual.

Key words: ant colony optimization; convergence speed; nearest neighbor node choosing strategy; genetic operator; crossover operator

1 引言

蚁群优化(Ant Colony Optimization, ACO)算法是模拟自然界中真实蚁群的觅食行为而形成的一种模拟进化算法. 蚁群算法用于组合优化问题求解时具有较强发现解的能力, 具有分布式计算的优势, 易于与其它算法相融合的优点, 在动态环境下也表现出高度的灵活性和健壮性^[1]. 但是蚁群优化算法始终存在收敛速度慢, 易于陷入局部最优解的问题.

为了改善蚁群优化算法这两方面的性能, 学者们提出了许多改进方法^[2-4], 这些研究对蚁群优化

算法(ACO)有了一定程度的改进, 但对提高收敛速度, 获取全局最优解的效果不是特别的明显, 仍然制约蚁群算法在大规模优化问题中的应用. 为此, 基于旅行商问题(Traveling Salesman Problem, TSP)的求解, 文中研究了一种带有近邻节点选择策略和遗传算子操作的快速收敛于最优解的算法, 简称NNGOACO算法(ACO algorithm based on the nearest neighbor node choosing strategy and genetic operator). 实验结果证明, 文中提出的算法对蚁群优化算法的收敛速度有较大的提高.

收稿日期: 2009-03-09; 修回日期: 2009-06-10

基金项目: 浙江省高等教育学会项目(Y200801); 杭州师范大学科技创新与文化创意基金项目(2009XJ076)

2 蚁群优化算法(ACO)

以 n 个城市的 TSP 问题为例来描述基本蚁群系统模型^[5]. 设 m 是蚁群中蚂蚁的数量, 用 $d_{ij}(i, j = 1, 2, \dots, n)$ 表示城市 i 和 j 之间的距离, 用 $b_i(t)$ 表示 t 时刻位于城市 i 的蚂蚁个数, 则有 $m = \sum_{i=1}^n b_i(t)$.

下面给出 ACO 算法描述.

Step 1: 初始化. 将每个边上的信息素的量初始化为一个很小的常数值; 将 m 只蚂蚁随机地分配到 n 个城市, 同时把出发点城市设置到禁忌表中.

Step 2: 每只蚂蚁根据式(1) 选择下一个城市, 并修改禁忌表.

$$j = \begin{cases} \arg \max_{j \in \text{tabu}_k} [\tau_{ij}(t)] [\eta_{ij}(t)]^\beta, & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \quad (1)$$

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{k \in \text{tabu}_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}(t)]^\beta}, & j \notin \text{tabu}_k \\ 0, & j \in \text{tabu}_k \end{cases} \quad (2)$$

其中, $q_0 \in [0, 1]$ 是初始设定的参数; q 是一个随机数, $q \in [0, 1]$; S 是由式(2) 决定的. η_{ij} 为蚂蚁从城市 i 转移到城市 j 的启发信息, 一般取 $\eta_{ij} = 1/d_{ij}$; d_{ij} 为城市 i 与城市 j 之间的距离; α 为在路径 i 和 j 上残留信息的重要程度; β 为启发信息的重要程度; $\text{tabu}_k (k = 1, 2, \dots, m)$ 用以记录蚂蚁 k 当前所走过的城市, 称为禁忌表, 即下一步不允许选择的城市; 集合 tabu_k 随着进化过程作动态调整. 该策略显然增强了搜索的多样性, 从而避免过早地陷入局部最优解而停滞.

Step 3: 信息素局部更新. 每只蚂蚁选择一个城市以后, 按式(3) 更新该边上的信息素的量.

$$\tau_{ij}(t+1) = (1 - \rho) \tau_{ij}(t) + \rho \Delta \tau_{ij}^k \quad (3)$$

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{l_{jb}}, & \text{当蚂蚁 } k \text{ 走过城市 } ij \\ 0, & \text{否则} \end{cases} \quad (4)$$

其中, l_{jb} 是蚂蚁 k 从开始城市到当前城市已走过的路径长度. $\Delta \tau_{ij}^k$ 表示第 k 只蚂蚁经过城市 i 和 j 时在该路径上留下的信息素的量.

Step 4: 计算最佳路径. 当 m 只蚂蚁走完所有城市以后, 按式(5) 计算最佳路径长度:

$$l_{\min} = \min\{l_k\}, k = 1, 2, \dots, m \quad (5)$$

其中, l_k 是第 k 只蚂蚁所走的路径长度.

Step 5: 信息素全局更新. 当所有蚂蚁走完所有城市以后, 仅对最佳路径上的信息素的量按式(6) 进行更新:

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

$$\Delta \tau_{ij} = \begin{cases} \frac{1}{l_k}, & \text{如果全局最优解经过路径 } ij \\ 0, & \text{否则} \end{cases} \quad (7)$$

其中, α 为全局信息素挥发系数, l_k 为最佳路径的长度.

Step 6: 如果没有达到指定的搜索次数则清空禁忌表, 重复上述过程.

3 NNGOACO 算法

3.1 NNGOACO 算法的基本原理

3.1.1 近邻节点选择策略

在 ACO 算法中, 蚂蚁选择下一城市时, 需要计算由城市 i 到城市 j 的转移概率, 然后选择概率较大的城市 j 行走, 而概率计算中用城市距离作为启发因子, 所以 ACO 算法计算得到的下一个城市 j 可能并不是距离城市 i 最近的城市. 当城市较多时, 计算城市 i 到其它城市的转移概率的计算量相当大, 计算速度非常慢, 算法收敛也就特别慢. 因此, 文中设计了近邻节点选择规则, 每次计算概率时只计算到符合规则的那些城市的概率, 从而可以大大节省计算时间, 加快算法的收敛速度.

近邻节点选择规则为: 首先根据 n 个城市的坐标文件建立坐标系对 n 个城市进行定位, 并计算出两两城市之间的距离, 形成一个 $n \times n$ 的距离矩阵 A ; 对 A 的每一列按城市距离由小到大进行排序, 并将排序后的城市距离用城市编号代替; 最后删除最后一行(因为预设城市 i 到城市 i 之间的距离为无穷大, 即不可达, 所以排序后肯定在最后位置), 形成一个 $(n-1) \times n$ 的距离索引矩阵 B . 这样矩阵 B 中的第 i 列从第一行至第 $(n-1)$ 行就是距城市 i 由近及远的城市编号排列.

在 NNGOACO 算法中, 指定只计算与第 i 个城市最近的 d 个城市的概率, 这样当城市规模较大时, 指定一个较小的 d 值, 即可以加快算法的收敛速度, 又可以提高算法的精度.

3.1.2 交叉算子选择策略

由于 ACO 算法易于陷入局部最优解而出现停滞现象, 文中设计出一种对较优解进行交叉操作的策略, 从而解决了算法快速收敛到某一解的情况, 使

解空间出现多样性,从而增大了寻求最优解的可能.

交叉算子主要用于进化算法.文中采用 Davis 提出的顺序交叉算子进行交叉操作.

在本算法中,首先在每轮循环结束时,计算每只蚂蚁所走路径的长度,按式(8)计算第 k 只蚂蚁被选做父代的概率.然后,计算出每只蚂蚁被选做父代的概率,按轮盘赌的方式选择其中两个做为交叉算子的两个父代进行交叉操作.最后,计算产生的两个新后代的路径长度,如果其结果比本轮最优解好,则用其代替本轮最优解,且用式(6)更新该条路径上的信息素的量;如果其结果不比本轮最优解好,则放弃对后代的处理.

$$p_{\text{parent}}^k = \frac{\frac{1}{L_k}}{\sum_{i=1}^m \frac{1}{L_i}}, \quad k = 1, 2, \dots, m \quad (8)$$

3.2 NNGOACO 算法

下面给出 NNGOACO 算法的描述:

Step 1: 初始化. 根据 n 个城市的坐标文件,建立近邻节点矩阵 B ,指定查询的近邻节点个数 d ($d \leq n$),设置计数器 $NC = \text{MAX}$,设置每条路径上的初始信息量为一常量,将 m 个蚂蚁随机地放置在 n 个城市,并将该蚂蚁所在城市设置到该蚂蚁的禁忌表中.

Step 2: 对每一只蚂蚁,以当前城市 i 为中心,按近邻选择规则选出 d 个蚂蚁未走过的城市并计算它们的转移概率,如果未走过的城市个数小于 d ,则全部计算;然后选择一个转移概率较大的城市 j 进行转移.在 d 个城市中,按式(1)和式(2)选择节点 j .

式(1)和式(2)中,当 $q > q_0$ 时,计算 d 个城市的转移概率 p_{ij}^k ,并根据赌轮盘规则选择城市 j ,将 j 加入该蚂蚁的禁忌表 tabu_k 中.

Step3: 信息素局部更新.步骤同 ACO 算法的信息素局部更新.

Step4: 计算最佳路径.当 m 只蚂蚁走完所有城市以后,计算每只蚂蚁所走的路径长度,从这 m 只蚂蚁所走的路径中找出最佳路径,保存它的行走路径及路径长度.

Step5: 信息素全局更新.当所有蚂蚁走完所有城市以后,首先对 Step4 中找出的最佳路径上的信息素按式(6)进行更新.然后进行交叉操作,更新最佳路径长度和全局信息素.

Step6: 如果没有达到指定的搜索次数则清空禁忌表,重复上述过程.

4 实验结果比较

为了验证文中提出算法的有效性,从 TSPLIB 下载了著名的 TSP 范例 st70 进行了实验,并与 ACO 算法、文献[6]在算法收敛性和实验结果方面进行了比较.由于参数选择没有理论指导,所以只能通过实验的方法来确定参数.本算法设置的参数为: $\alpha = 1, \beta = 4, \rho = 0.5, Q = 20000, q_0 = 0.8, d = 25$;其中式(6)中的参数取: $\alpha = 0.6$.

4.1 收敛特性比较

用不同方法求解 st70 TSP 问题,经过 10 次实验得到的最优解的收敛性进行比较.实验表明:ACO 算法和文献[6]需要 2600 次以上的迭代才能收敛到较优解,而文中所提出的算法仅需 964 次就能收敛到较优解.由此可见本算法能加快算法的收敛速度.

4.2 优化 TSP 问题的结果比较

表 1 给出了求解 st70 的 TSP 问题时所求得的较优解结果比较.从表 1 可以看出本方法所求得的最优解虽然还没有达到目前公布的最优解,但是由图 1 可以看出它的收敛速度明显比其它两种算法要快的多^[7-8].

表 1 st70 TSP 问题较优解的比较

ACO	682.233286
文献[6]的算法	679.925452
文中算法	680.566094
网站公布的结果	675

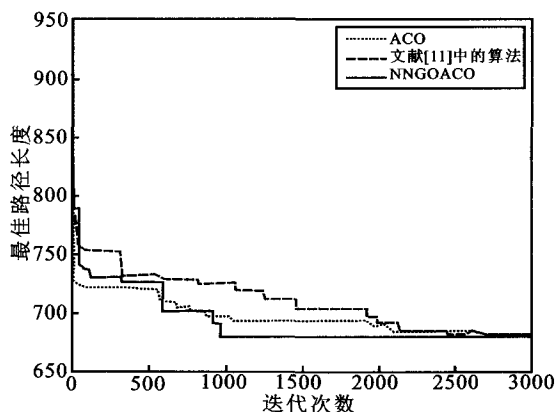


图 1 三种蚁群算法求解 st70 TSP 问题的收敛性比较

5 结束语

文中方法对加快算法的收敛速度是行之有效的,然而蚁群算法不像遗传算法那样,具有坚实的数学基础和系统的分析方法,目前形成的研究成果大

(下转第 185 页)

$$\text{MSE} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{ij} - y_{ij})^2}{M \times N} \quad (5)$$

实验中分别采用 $4 \times 4 \times 64$ 的码书, $4 \times 4 \times 128$ 的码书和 $4 \times 4 \times 256$ 的码书, 即码字的维数为 16, 码书的尺寸分别为 64、128、256. 实验分别采用三种尺寸的码书, 重构图像的 PSNR 值见表 1. 可以看出, 码书的尺寸越大, 重构图像的失真越小.

表 1 重构图像的失真 PSNR

码书尺寸	PSNR
64	27.05 dB
128	27.39 dB
256	27.89 dB

文中在研究了 AEI 算法和 DM642 板卡的基础上将矢量量化算法与 DSP 平台相结合, 成功完成将 AEI 算法向 DM642 板卡的移植, 实现了视频的实时性处理. 并验证了矢量量化技术和 DSP 在视频和图像处理方面的性能, 取得了较为理想的结果.

参考文献:

- [1] 孙圣和, 陆哲明. 矢量量化技术及应用[M]. 北京: 科学出版社, 2002.
- [2] Bei C D, Gray R M. An improvement of the minimum dis-

tortion encoding algorithm for vector quantization[J]. IEEE Transactions on Communications, 1985, 33(10): 1132-1133.

- [3] 王婷婷, 乔瑞萍, 潘志斌, 等. 基于 DM642 的矢量量化算法研究及应用[C]// TI DSP 大奖赛论文集. USA: TI 公司, 2008: 143-161.
- [4] 李东平, 潘志斌, 乔瑞萍, 等. 一种改进的 AEI 算法中初始匹配码字的快速查找方法[J]. 中国图像图形学报, 2009, 14(9): 1808-1812.
- [5] 乔阳, 潘志斌, 乔瑞萍, 等. 基于 Hadamard 变换和自适应顺序搜索的码字快速搜索算法[J]. 微电子学与计算机, 2009.
- [6] Texas Instruments. Video driver examples for DM642 EVM demonstration software release report (SPRA932a) [M]. USA: Texas Instruments Incorporated, 2003.
- [7] Texas Instruments. TMS320C64X DSP Video Port/VCX-O Interpolated Control (VIC) port reference guide [M]. USA: Texas Instruments Incorporated, 2003.
- [8] Texas Instruments. TMS320DM642 technical overview (SPRU615) [M]. USA: Texas Instruments Incorporated, 2002.

作者简介:

谢敏 男, (1982-), 硕士研究生. 研究方向为矢量量化算法的研究及其在 DSP 上的实现.

(上接第 181 页)

多都是基于实验的. 但蚁群算法将像其它的智能算法一样, 越来越多的被应用到更多的领域.

参考文献:

- [1] 朱庆保, 杨志军. 基于变异和动态信息素更新的蚁群优化算法[J]. 软件学报, 2004, 15(2): 185-192.
- [2] Stutzle T, Hoos H H. MAX-MIN ant system and local search for the traveling salesman problem[C]// IEEE Int'l Conf. on Evolutionary Computation. Indianapolis: IEEE Press, 1997: 309-314.
- [3] Tsai C F, Tsai C W. A new approach for solving large traveling salesman problem using evolution ant rules[C]// Neural Networks, IJCNN 2002, Proc. of the 2002 Int'l Joint Conf. on. Honolulu: IEEE Press, 2002: 1540-1545.
- [4] 吴斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解

算法[J]. 计算机学报, 2001, 24(12): 1328-1333.

- [5] 郭俊恩, 王士同, 徐红林. 基于蚁群遗传算法的氨基酸序列比对方法[J]. 计算机应用, 2007, 27(6): 1434-1437.
- [6] Dorigo M, Caro G D. Ant colony optimization: A new meta-heuristic[C]// Proc. of the 1999 Congress on Evolutionary Computation. Washington: IEEE Press, 1999: 1470-1477.
- [7] 徐克奇. 蚁群优化算法在电镀试验台作业调度的应用[J]. 微电子学与计算机, 2008, 25(12): 210-212.
- [8] 郭新华, 蒋艳, 曹建霞. WSN 中动态自适应蚁群路由算法[J]. 微电子学与计算机, 2008, 25(12): 157-160.

作者简介:

陶利民 男, (1975-), 讲师, 博士研究生. 研究方向为人工智能、网络计算.

郭俊恩 男, (1978-), 硕士, 讲师. 研究方向为人工智能.