求解 TSP 的改进蚁群算法 *

侯文静, 马永杰, 张 燕, 石玉军
(西北师范大学 物理与电子工程学院, 兰州 730070)

摘 要: 针对蚁群算法存在的搜索时间长、易限于局部最优解等缺陷,提出了一种改进的蚁群算法。通过在初始化信息素矩阵中采用候选城市列表减少劣质解,在局部搜索中采用聚类进行二次搜索,缩小了算法的搜索范围、改善了解空间的质量,提高了搜索速度。仿真结果表明,改进后的蚁群算法在 TSP 的求解中,收敛速度和全局寻优能力均得到较大的提高。

关键词: 蚁群算法(ACA); 旅行商问题; 候选城市列表; 聚类; 蚁群系统(ACS)
中图分类号: TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2010)06-2087-03
doi:10.3969/j.issn.1001-3695.2010.06.026

Improved ant colony algorithm for solving TSP

HOU Wen-jing, MA Yong-jie, ZHANG Yan, SHI Yu-jun
(*School of Physics & Electronic Engineering*, *Northwest Normal University*, *Lanzhou* 730070, *China*)

Abstract: Aimed at the shortcomings, which needing much time and easier to fall in local optimal solution in the ant colony algorithm, this paper proposed an improved algorithm. Through employing the list of candidate cities in the initial pheromone matrix to decrease inferior solutions and using cluster to do the second search in the local search, it could narrow the searching range of algorithm, could improve the quality of the solution space and raise the searching speed. The simulations result for TSP shows that the algorithm is improved greatly in convergence rate and ability of global optimization.

Key words: ant colony algorithm(ACA); TSP; list of candidate cities; clustering; ant colony system(ACS)

蚁群算法又称做蚂蚁算法,是一种用来在图中寻找优化路径的几率型算法,利用生物信息激素作为蚂蚁选择后续行为的依据。每只蚂蚁会对一定范围内其他蚂蚁散布的生物信息激素作出反应,依据生物信息激素的强度在每一个道口对多条路径选择作出概率上的判断并执行选择,由此观察并影响它们以后的行为。它由意大利学者 Dorigo 等人[1]于 20 世纪 90 年代提出,其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。目前已经应用于求解旅行商、二次指派、车辆调度等经典的理论问题,并迅速地推广到实际问题领域,成为解决组合优化问题最有效的算法之一。

然而,该算法的收敛性和连续空间寻优[2]仍是制约算法广泛应用的主要瓶颈。围绕这个问题,众多学者发表了大量有价值的学术文献。文献[3]研究了一种与真实蚂蚁系统更加贴近的信息素扩散算法,进一步真实地体现了蚂蚁信息系统;文献[4]在信息素的局部更新规则和全局更新规则中增加了参数控制;文献[5]加入了多种蚁群及与之相应的信息素调节机制;文献[6]提出 RAACS 算法,随机选择部分城市来计算状态转移概率;文献[7]采用了三种局部优化算子来交换搜索路径中城市的位置;文献[8]对 TSP 中的城市进行聚类,将其分解成若干小规模的子城市集合,然后分别进行求解,最后将各个子城市的解组合成完整的待求解;文献[9]采用了新信息素更新策略——内部更新系统;文献[10]提出的 KCC-Ants 和 ELU-Ants 算法,利用了新的局部信息素更新规则。上述算法都是单独针对信息素或单独针对路径构建和局部搜索的改进,而本文对蚁群算法的改进不仅仅是对信息素的修改,而是与路径构建和局部搜索相结合,使得新的算法既可以提高收敛速度和全局寻优能力,又可以提高解的质量。

1 蚁群算法

1.1 基本蚁群算法的数学模型[1,2,11]

以求解平面上 n 个城市的 TSP 为例来说明蚁群系统的基本模型。给定 n 个城市集合 $C=\{c_1,c_2,\cdots,c_n\}$ 及城市之间路径的长短 $d_{ij}(1\leqslant i,j\leqslant n,i\neq j)$,TSP 就是找到一条只经过每个城市一次且回到起点的、最短路径的回路。设 m 表示蚁群中蚂蚁的数目;n 表示 TSP 规模;d_{ij} 为城市 i 与 j 之间的距离;$\tau_{ij}(t)$ 为 t 时刻在路径(i,j)连线上残留的信息量。在初始时刻各条路径上的信息量相等,并设 $\tau_{ij}(0)=\mathrm{const}$(const 为常数)。

蚂蚁 $k(k=1,2,\cdots,m)$ 在运动过程中,根据各条路径上的信息量决定其转移方向。禁忌表 $\mathrm{tabu}_k(k=1,2,\cdots,m)$ 记录蚂蚁 k 已走过的城市。$P_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率,如式(1)所示。它表示蚂蚁在选择路径时会尽量选择离自己距离较近且信息素浓度较大的方向。

$$P_{ij}^k(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^\alpha[\eta_{ij}(t)]^\beta}{\sum\limits_{s\subset \mathrm{allowed}_k}[\tau_{is}(t)]^\alpha[\eta_{is}(t)]^\beta} & j\in\mathrm{allowed}_k\\ 0 & \text{否则}\end{cases}\tag{1}$$

收稿日期: 2009-10-29; **修回日期**: 2009-12-27 **基金项目**: 甘肃省自然科学研究基金资助项目(096RJZA115);甘肃省教育厅科研项目(0901B-08)

作者简介:侯文静(1983-),女,河北邯郸人,硕士研究生,主要研究方向为计算机测量与控制(houwenjing034@126.com);马永杰(1967-),男,副教授,博士,主要研究方向为物流技术、智能控制;张燕(1982-),女,硕士,主要研究方向为计算机测量与控制;石玉军(1972-),男,硕士研究生,主要研究方向为计算机测量与控制.

其中：$\text{allowed}_k = \{C - \text{tabu}_k\}$，表示在 t 时刻蚂蚁 k 下一步允许选择的城市；α 为信息启发式因子，反映了蚁群在运动过程中所残留的信息量的相对重要程度；β 为期望启发式因子，反映了期望值的相对重要程度；η_{ij} 为启发函数，表示由城市 i 转移到城市 j 的期望程度，其表达式如下：

$$\eta_{ij} = 1/d_{ij} \tag{2}$$

随着时间的推移，要对残留在各条路径上的信息素进行更新处理。由此 $(t+n)$ 时刻在路径 (i,j) 上的信息量可按式(3)和(4)的规则进行调整。

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t)\Delta\tau_{ij}(t) \tag{3}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{4}$$

其中：$\rho(0 \leqslant \rho < 1)$ 表示信息素的挥发系数，$1-\rho$ 表示信息素的残留因子；$\Delta\tau_{ij}(t)$ 表示本次循环中路径 (i,j) 上的信息素增量，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 (i,j) 上的信息量。

根据信息素更新策略的不同，Dorigo 等人给出了三种不同的算法模型，分别为 Ant-cycle、Ant-density 和 Ant-quantity，它们的区别在于 $\Delta\tau_{ij}^{k}(t)$ 求法不同。式(5)所示的为 Ant-cycle 模型。

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L_k & \text{第 } k \text{ 只蚂蚁在本次循环中经过 } (i,j) \\ 0 & \text{否则} \end{cases} \tag{5}$$

其中：Q 表示信息素强度，它在一定程度上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。

Dorigo 等人认为在 Ant-density 和 Ant-quantity 算法中，蚂蚁在构建解的同时释放信息素；而在 Ant-cycle 算法中，蚂蚁在完整地构造问题的解之后再释放信息素，利用的是整体信息，效果要好于前两种算法。因此通常采用式(5)作为蚁群算法的基本模型。

1.2　蚁群系统算法

蚁群系统(ant colony system，ACS)是在基本蚁群算法的基础上使用了未出现的新的机制和策略，进一步获得较好的性能。

在蚁群系统中，其状态转移规则采用伪随机比例规则，如式(6)所示：

$$j = \begin{cases} \arg\max\limits_{s \in \text{allowed}_k}\{[\tau_{is}]^{\alpha}[\eta_{is}]^{\beta}\} & \text{如果 } q \leqslant q_0 \\ \text{式(1)} & \text{否则} \end{cases} \tag{6}$$

其中：q 是均匀分布在区间[0,1]中的一个随机变量；q_0 是一个算法参数($0 \leqslant q_0 \leqslant 1$)。在这种规则下，每当蚂蚁要选择向哪一个城市转移时，就产生一个在[0,1]内的随机数，根据这个随机数的大小按式(6)确定用哪种方法产生蚂蚁转移的方向。

在蚁群系统中，其信息素更新规则采用全局更新规则和局部更新规则，如式(7)(8)所示：

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad \forall (i,j) \in T^{bs} \tag{7}$$

其中：T^{bs} 是最优路径，C^{bs} 是 T^{bs} 的长度，$\Delta\tau_{ij}^{bs} = 1/C^{bs}$。在蚁群系统中信息素的蒸发和信息素的释放都只在构成 T^{bs} 的边上执行，这种方式可以降低算法的时间复杂度，从 $O(n^2)$ 降低到 $O(n)$。

$$\tau_{ij} \leftarrow (1-\varepsilon)\tau_{ij} + \varepsilon\tau_0 \tag{8}$$

其中：ε 和 τ_0 是两个参数，ε 满足 $0 < \varepsilon < 1$，τ_0 是信息素的初始值。局部更新的作用在于，蚂蚁每经过 (i,j)，该边的信息素 τ_{ij} 将会减少，从而其他蚂蚁选中该边的概率相对减少，这将增加探索未使用过的边的机会，使得算法不易陷入停滞状态。

2　蚁群算法的改进

由于蚁群中个体根据信息素浓度正反馈来产生新解，随着正反馈的增强，降低了全局搜索能力，易陷入局部最优，影响求解质量；其次，由于蚁群中个体的运动是随机的，当群体规模较大时，很难在较短的时间内从大量杂乱无章的路径中找出一条较好的路径。为此提出以下改进。

2.1　采用新的初始化信息素

在对 TSP 的研究中发现，在其解空间中，大部分解都是劣质解，对优质解的寻找产生很大的阻力。如果能够减少这些劣质解、缩小解空间的搜索范围，就能提高解空间的质量，进而能够快速找到全局最优解，提高算法性能。

在 TSP 的最优解中绝对不含任何交叉线路的情况[11]；对于每个城市而言，从离其最近的几个城市中选择一个作为近邻城市，而不是从剩下城市中任选一个，也必然可以找到最优解。它选择的下一个城市应该是离它最近的城市[5]。因此，在改进算法中采取加入候选城市列表、加强城市间的相对位置对最优解的作用，以相对城市之间的距离及其排列顺序为依据，进行初始化 $\tau_{ij}(0)$ 的设置，即在给定城市坐标的同时，其特征也已固定，不会随着搜索的进行而改变。在加入初始化公式后，使得搜索由于静态特征的给定而更加具有目的性、指导性和确定性，能够很快得到最优的路径。算法改进的初始化信息素步骤如下：

a)计算城市之间的距离，并按照从小到大的顺序排列。

b)计算每个城市的最近邻列表，如果城市规模 $N < 50$，采用最近邻城市数目 $q = 6$。

c)按照式(9)计算初始化信息素：

$$\tau_{ij}(0) = \begin{cases} \dfrac{(q-l+1)\times d_{ij}}{n\times(d_{ij})_{\max}} & \text{如果城市 } j \text{ 在城市 } i \text{ 的最近邻列表中} \\ \dfrac{(d_{ij})_{\min}}{n\times(d_{ij})_{\max}} & \text{否则} \end{cases} \tag{9}$$

其中：$(d_{ij})_{\max}$ 是以城市 i 为中心，离城市 i 最远的城市距离；$(d_{ij})_{\min}$ 是以城市 i 为中心，离城市 i 最近的城市距离；q 是最近邻城市的数目；l 是城市 j 在城市 i 的最近邻列表中的排列次数；n 是城市的规模数。

2.2　局部搜索中通过聚类进行二次搜索

在对蚁群算法的研究中，经过大量的数据测试发现，对于一个静态的 N 规模的城市集合，在每次的搜索过程中，有某一部分或几部分城市的顺序是“已经固定的”。所谓已经固定的是指在一次搜索的迭代中，这些城市的顺序以固定的模式多次出现。在这种情况下，可以将这些城市看做一个整体，进而可以在此基础上减小城市的整体规模，利用蚁群算法在低维空间具有较好的搜索特性降低算法的复杂性、提高算法的性能。因此，在改进算法的局部搜索部分中：a)加入一个子程序 $f_\text{search}(d_1, d_2)$，即寻找两个迭代最优解中相同的城市集合；b)将这些城市集合看成一个整体，用子程序 Jiaoji 重新进行搜索，即检测交集中是否有交叉的现象。将这些城市集合变成一个虚拟的新城市加入到原来的城市集合中，降低原城市规模。c)对降低城市规模后的新城市集合进行“二次搜索”，进而得到较好的解。通过聚类进行二次搜索主要体现在较大城市规模的 TSP 上。实现步骤如下：

a)在第一次应用蚁群算法的基础上，从迭代最优解中取出任意的两条路径 d_1、d_2。

b)运行子程序 $f_\text{search}(d_1, d_2)$，并将交集结果保存到 K 中。

c)用子程序 Jiaoji 进行二次搜索，并将结果还原成原来的城市规模。

d)判断最终找到的最优路径和 d_1、d_2,输出经过两次搜索后的全局最优解。

由此,改进后的 ACS 算法流程可以描述如下:

a)初始化参数 Q、C、$NC_{\max}$(最大迭代次数)、α、β、ρ、q_0。

b)将 m 只蚂蚁随机地放在 m 个城市中,计算各个城市的距离矩阵及其最近邻矩阵,按照式(10)初始化信息素矩阵,并将结果存放到 $\tau_{ij}(0)$ 中。

c)设置迭代次数 $NC_{\max}$ 为 0,随机选择蚂蚁的初始位置,并将该位置放置到对应的 tabu 表中。

d)按照式(6)计算每只蚂蚁 k 将要转移的下一个城市,若当前城市为 i,选择的下一个城市为 j,此时将 j 放入到对应的 tabu_k 中,直到每只蚂蚁搜索完成一次路径回路。

e)计算每只蚂蚁的回路长度 L_k,并记录当前最好的解。

f)若迭代次数达到 $NC_{\max}$,则跳到 i),否则跳到 g)。

g)按照式(7)(8)更新全局信息素矩阵和局部信息素矩阵。

h)置空 $\Delta\tau_{ij}$,置 tabu_k 表为空,$NC \leftarrow NC+1$,跳转到 c)。

i)输出迭代最优解。

j)以上面得出的迭代最优解为基础,任意取出其中的两条回路 d_1、d_2,调用子程序 f_search(d_1,d_2),寻找交集部分并将结果保存到矩阵 K 中。

k)调用子程序 Jiaoji 进行二次搜索,并将结果 k_1 还原成原来的城市集合的路径。

l)将最后找到的最优解与 d_1、d_2 进行比较,输出全局最优解。

3　仿真结果及分析

仿真算例选用了 TSP 测试库 TSPLIB 中的 Oliver30 和 KroA100 以及 CTSP(中国 31 个省会城市的旅行商问题),并将改进的算法与基本蚁群算法进行比较。

实验中参数设置分别为:α = 1,β = 3,ρ = 0.1,q_0 = 0.6,Oliver30、CTSP,Q = 100;KroA100,Q = 1 000。实验结果如表 1 和图 1 ~4 所示。

表 1　Oliver30 问题的不同算法结果比较

最短路径	迭代次数(本文算法)	迭代次数(基本算法)	文献[5]算法
424.635 6	26	64	27
424.464 7	28	87	41
423.973 8	25	136	116
423.740 9	16	179	137

表 1 中最短路径长度为最优路径的长度,迭代次数是取 10 次实验的平均而得到的。从表 1 中可以看出,本文改进的蚁群算法平均 16 次迭代就能搜索到目前最好解(423.7409),而用基本蚁群算法和文献[5]中的算法分别要用 179 次和 137 次迭代才能够搜索到。就次优解而言,在取得相同结果的情况下,改进后的算法所需要的迭代次数比基本算法和文献[5]中的算法少得多。

图 1、2 是 Oliver30 问题中运用改进算法所找到的最优路径和进化曲线。从图中可以看出,改进后的蚁群算法不仅在求解能力上有明显优势,而且算法的执行效率也得到提高。

借鉴上述探讨的结果,将该算法用于 CTSP 和 KroA100 问题,实验结果如图 3 和 4(图中'+ + '表示基本蚁群算法的进化曲线;'—'表示本文算法的进化曲线)所示。从图中可以看出,对于基本蚁群算法易限于局部最优,而改进后的算法能够在较少的迭代次数中找到最优解,较好地避免了局部最优。

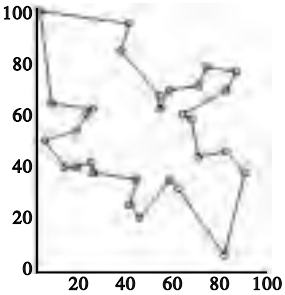

图1　最优路径

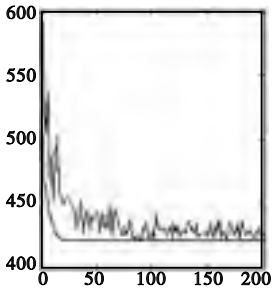

图2　进化曲线

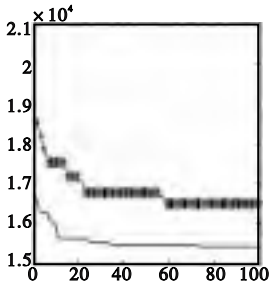

图3　CTSP最优解的进化曲线(15404)

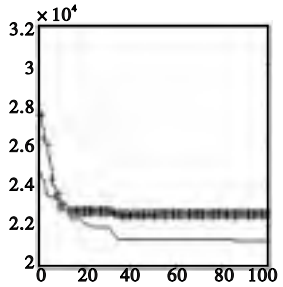

图4　KroA100最优解的进化曲线(21285)

4　结束语

针对基本蚁群算法存在搜索时间长、易限于局部最优的缺点,提出了将信息素管理与路径构建和局部搜索相结合的改进蚁群算法。通过对 TSP 的算例进行实验结果比较,显示了算法能够克服基本蚁群算法的缺陷,每次求解均能找到最优解,且求解速度优于同期的研究结果。对其他多例 TSP 的求解均获得了类似结果,证明本文采取的改进方法是有效和可行的。

参考文献:

[1] DORIGO M, MANIEZZO V, COLORNI A. The ant system:optimization by a colony of cooperating agents[J]. IEEE Trans on System, Man, and Cybernetics,1996,26(1):29-41.

[2] 段海滨,王道波,朱家强,等.蚁群算法理论及应用研究的进展[J].控制与决策,2004,19(12):1321-1326.

[3] 黄国锐,曹先彬,王煦法,基于信息素扩散的蚁群算法[J].电子学报,2004,32(5):865-868.

[4] 李金汉,杜德生.一种改进蚁群算法的仿真研究[J].自动化技术与应用,2008,27(2):58-60.

[5] 徐精明,曹先彬,王煦法.多态蚁群算法[J].中国科学技术大学学报,2005,35(1):59-65.

[6] QI Cheng-ming. An ant colony system hybridized with randomized algorithm for TSP[C]//Proc of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence,Networking,and Parallel/Distributed Computing. 2007:461-465.

[7] 龚本灿,李腊元,蒋廷耀,等.基于局部优化策略求解 TSP 的蚁群算法[J].计算机应用研究,2008,25(7):1974-1976.

[8] 胡小兵,黄席樾.对一类带聚类特征 TSP 问题的蚁群算法求解[J].系统仿真学报,2004,16(2):2683-2686.

[9] PINTEA C M, DUMITRESCU D. Improving ant systems using a local updating rule[C]//Proc of the 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. 2005:295-298.

[10] NAIMI H M, TAHERINEJAD N. New robust and efficient ant colony algorithms:using new interpretation of local updating process[J]. Expert Systems with Applications,2009(36):481-488.

[11] 段海滨.蚁群算法原理及其应用[M].北京:科学出版社,2005.