一种求解TSP的混合型蚁群算法

赵学峰
(西北师范大学 数学与信息科学学院,甘肃 兰州 730070)

摘 要:针对基本蚁群算法存在的过早收敛问题,提出一种采用混合模式调整信息素的改进蚁群算法,当陷入局部最优解时便启用新的信息素调整规则,从而使算法跳出局部解.计算机仿真结果表明,这种混合型蚁群算法对求解TSP难题有较好的改进效果.

关键词:蚁群算法;TSP;信息素
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1001-998X(2003)04-0031-04

A hybrid ant colony algorithm for solving TSP

ZHAO Xue-feng
(College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, Gansu, China)

Abstract: A hybrid ant algorithm is presented to prevent the premature in canonical ant colony algorithm in this paper. The main idea of the meeting algorithm is that a hybrid pheromone update strategy is used according to the solution that artificial ants have found. Computer simulation shows that the proposed algorithm can efficiently find better minimum beyond premature convergence for hard TSP problem.

Key words: ant colony algorithm; TSP; pheromone

地球上的生物物种在漫长的演化过程中形成了丰富的行为特性,并且不断完善和发展,以更好地适应生存的环境.生物化的过程本质上是一个优化过程,这一特点使得生物系统对计算科学具有直接启发意义.随着计算机的诞生,人们期望计算机借助程序的形式创造出一些新型的智能体,于是对人类的大脑活动、人类的学习方式以及生物界的进化过程进行了模拟研究,促成了人工神经网络、遗传算法和免疫算法等智能计算模型的兴起.意大利学者Dorigo M等提出的蚁群算法(Ant colony system)类比自然界蚁群的觅食行为与旅行售货商问题(TSP)的相似之处,成为求解NP-hard问题的一种有潜力的随机优化算法[1-4].和遗传算法等基于自然法则的随机启发式搜索算法一样[5-7],蚁群算法存在过早收敛现象——算法陷入局部最优解而停止搜索更优的解.为此,笔者对基本蚁群算法的信息素调整规则作了改进,并应用于典型TSP进行了求解验证.

1 基本蚁群算法模型

1.1 TSP模型

给定 n 个城市(结点)的集合 $\{1,2,\cdots,n\}$,及城市 i 和 j 之间的欧氏距离 $d_{ij}(i,j=1,\cdots,n,d_{ii}=0)$,模型的目标是寻找一条经过 n 个城市各1次且仅1次最终回到出发点的最短回路 H^*.

TSP是著名的NP难题[8,9],目前尚无一个多项式时间算法,一些传统的确定性算法在求解这一问题时显得无能为力,例如用Backtrack法搜索问题的解空间时,算法的计算时间复杂性为 $O(n!)$[9].

1.2 蚁群行为分析

蚂蚁在搜寻食物时会在它所经过的路径上释放一种特殊的信息素(Pheromone)来传递信息,后来的蚂蚁走到前面蚂蚁走过的路口时可以感知到环境中这种物质的存在及强度,倾向于朝着信息素强度高的方向移动.某一路径上走过的蚂蚁越多,则信息素越来越强,后来者选择该路径的概率就越

收稿日期:2002-12-30;**修改稿收到日期**:2003-09-06
作者简介:赵学峰(1965—),男,甘肃渭源人,讲师.主要研究方向为算法分析.

大;而其他路径上的信息素却随着时间减弱.通过信息素的作用,蚁群行为与环境之间形成一个信息正反馈机制.这种间接的通信方式形成一种分布式自组织行为,个体简单的蚂蚁从初始的随机路径开始,经过群体演化行为最终找到最优路径.

1.3 蚁群算法的关键因素

蚁群算法用一类简单的智能体(人工蚂蚁),并用数量化的人工信息素来模拟自然界中的蚂蚁与环境的正反馈.基本蚁群算法可形式化描述为

$$ACA = (t, \Omega, H, \mu, \lambda, PHE_{\text{init}}, H^*, \varepsilon),$$

其中 t 为时间基,描述模型的离散搜索时间;Ω 为信息素分布的状态空间;H 为问题的解空间;$\mu:\Omega \longrightarrow \Omega$ 为状态转移函数;$\lambda:\Omega \longrightarrow H$ 为输出函数;$PHE_{\text{init}} \in \Omega$ 为初始信息素分布;$H^* \in H$ 为得到的最优解;ε 表示算法终止条件.

设蚂蚁的数量为 m,初始时将 m 只蚂蚁随机放在 n 个结点上,设置每边的初始信息素.在时刻 t,设第 $k(k=1,2,\cdots,m)$ 只蚂蚁所在的结点为 i,基本蚁群算法的优化过程体现在:

1) 路径选择机制.蚂蚁从环境中感知信息素的存在并选择要移动的边.蚁群算法中人工蚂蚁存在于离散的状态中,它们从 i 到另一个点 j 的运动采用下列概率转移规则

$$P_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{j\in \text{allowed}_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & j \in \text{allowed}_k, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

其中 $\tau_{ij}(t)$ 表示第 t 个搜索周期时边 (i,j) 上的信息素存量,是蚂蚁之间的通信通道;$\eta_{ij}(t)$ 通常取为 $1/d_{ij}$;参数 α,β 表示相对权重,用以调整 $\eta_{ij}(t)$ 与 $\tau_{ij}(t)$ 的相对重要性.α 越大,蚂蚁越倾向于选择其他蚂蚁使用的路段,体现了蚂蚁之间的协作;β 越大,则该概率选择越接近贪心规则.allowed_k 表示蚂蚁 k 下一步允许选择的城市集合,按照 TSP 的约束,在蚂蚁的周游完成以前只能访问未曾到过的城市,而 $\{1,2,\cdots,n\} - \text{allowed}_k$ 则成为禁忌表.

2) 信息素调节机制.当 m 只蚂蚁都完成一次周游后,构造了一个回路集合 $H(t)$,形成问题的一个可行解集合,其中第 k 只蚂蚁的回路长度记为 $L(k,t)$.当第 t 个搜索时刻结束时,蚂蚁便在各自的回路上留下信息素,其增量与该解的质量(即 $L(k,t)$)成反比,m 只蚂蚁留在边 (i,j) 上总的信息素增量为

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m}\Delta\tau_{ij}^k(t),$$
$$\Delta\tau_{ij}^k(t) = Q/L(k,t), \tag{2}$$

表示当蚂蚁 k 在循环 t 经过边 (i,j) 时释放的信息素,否则为 0.Q 是一正常数.

蚁群算法模拟了生物信息素随着时间逐渐消失的特点,假设人工信息素的保留比例为 ρ,则 $1-\rho$ 表示信息素的挥发程度,于是边 (i,j) 上的信息素的全局更新公式为

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t),$$
$$\rho \in (0,1). \tag{3}$$

(3)式综合了循环 t 以前的信息素与循环 t 所产生的信息素,是蚁群协同选择的结果,为蚁群的 $t+1$ 次循环搜索提供了基础,体现了蚁群不断从环境学习、优化的正反馈性.

算法中的参数 Q,C,α,β,ρ 目前用实验方法确定其较佳取值范围.

如果程序终止于 T 次循环后,则基本蚁群算法的时间复杂度 $O(T\cdot m\cdot n^2)$.

文献[3]将遗传算法的变异算子引入蚁群算法,文献[4]研究了用 2 只蚂蚁完成一次周游和具有并行性的分段算法,改进了蚂蚁搜索的路径方式.蚂蚁之间是通过信息素来实现相互通信、协同优化的,有必要探讨信息素的调整方式.

2 引入混合策略的蚁群算法

在基本蚁群算法中每次周游结束要给所有蚂蚁经过的边上增加信息素,实际上,在较差解对应的边上增加信息素是一种对资源的浪费,会造成大量的无效搜索,因此这种方式使信息素的释放范围偏大.在遗传算法中,经过选择算子的作用会丢弃一些染色体,借鉴这种方法,对蚁群算法在每次循环所得到的 m 个解中的部分较差解,可以停止在其经过的路径上释放信息素,有利于算法的快速收敛.算法陷于停滞,表明某些不在最优回路中的边上累积了较多的信息素,而某些最优回路的边上信息素没有得到有效的增强,正反馈机制使得算法无法逃出这种局面.在基本蚁群算法中,各边的初始信息素为 C,由(2)式和(3)式可知经过 t 次循环后 (i,j) 边上的累计信息素存量为

$$\tau_{ij}(t+1) = \rho^t \cdot C + \rho^{t-2}\cdot\Delta\tau_{ij}(1) + \cdots +$$

$\rho^1 \cdot \Delta\tau_{ij}(t-1) + \Delta\tau_{ij}(t)$.

如果边(i,j)不属于 H^*，而且初始随机搜索使 $\Delta\tau_{ij}(1)$较大，则正反馈机制使 $\Delta\tau_{ij}(2),\cdots,\Delta\tau_{ij}(k)$ 较大，该边在 t 次循环中被选中的概率随之增大，对此应启用新的策略克服.

n 个结点的 TSP 问题共有 $C_n^2 = n(n-1)/2$ 条边，蚁群算法的关键是信息素的释放范围和强度，即如何由 t 时刻的各边的信息素集

$$\{\tau_{ij}(t) \mid i,j = 1,2,\cdots,n\}$$

产生 $t+1$ 时刻的

$$\{\tau_{ij}(t+1) \mid i,j = 1,2,\cdots,n\}.$$

设 $L(t)$是在 t 时刻得到的最短回路长度，$OPTL(t)$是直到 t 时刻为止算法得到的最短回路长度，即

$$L(t) = \min\{L(k,t) \mid k = 1,2,\cdots,m\},$$

$$OPTL(t) = \min\{L(1),L(2),\cdots,L(t)\},$$

则新的释放信息素的方式可以分以下 3 种情况处理：

1) 如果 $L(t+1) < OPTL(t)$，表明解的质量在不断改进，算法目前运行良好，在循环 t 完成后，从 m 只蚂蚁的解集 $H(t)$中选出最优的 m_1 条回路组成优势解集 $H_1(t)$并在其上释放信息素，这一策略保证了 $OPTL(t) \in H_1(t+1)$. m_1 一般应取值较小，由于概率论中的正态分布具有普适性，可以认为一群数量较多的蚂蚁的寻优能力服从正态分布，按其固有的“3δ 原则”，随机变量与期望值的绝对差不超过 3δ 的概率为 99.7%，不超过 2δ 的概率为 95.4%，所以代表优势个体数量的 m_1 取值可以在 $0.01n$ 至 $0.05n$ 之间.

2) 如果 $L(t+1) \not< L(t)$，算法运行出现退化，在退化解对应的回路上释放信息素一般会出现长时间的震荡，所以仅在 $OPTL(t)$对应的边上释放信息素，依靠最佳个体的指导摆脱退化.

3) 如果 $L(t+1) = L(t)$，即算法陷于停滞，则采用模拟退火算法的思想从 $H(t)$中按某种选择机制组成有 m_2 条回路的退火解集 $H_2(t)$并释放信息素，在其余的边上实行“回溯”机制——指边上的信息素恢复为初始值，文中将边(i,j)上初值重新设置为 $rand \times \eta_{ij}$，其中 $rand$ 是 0,1 之间的随机数.

在非停滞情况下信息素的释放范围比基本蚁群算法缩小，摆脱了基本蚁群算法初期的长时间震荡，加速收敛过程，由于各边的信息素增量变少，所以信息素保留因子 ρ 应取值较大. 在停滞情况下，加大了信息素的释放范围，重新改变各边的信息素分布，使蚁群从局部极值中跃迁到新的搜索空间. 算法采用优势个体存储蚁群过去的学习能力，赋予蚁群以记忆特性，使得寻优过程能从混乱归于有序，并有可能向全局最优解收敛.

依据上述分析，现给出混合型蚁群算法的算法描述如下：

步骤 1 给出 Q,α,β,ρ，放置 m 个蚂蚁在 n 个城市上，计算 $\eta_{ij} = 1/d_{ij}$，置 $\tau_{ij}(0) = rand \times \eta_{ij}$，$\Delta\tau_{ij} = 0$； $t = 1$.

步骤 2 蚂蚁 k 从起点开始，按(1)式概率值的大小选择下一个城市 $j, k \in \{1,2,\cdots,m\}$，$j \in allowed_k$. 如果蚂蚁 k 转移到 j，从 $allowed_k$ 中删除 j，直至 $allowed_k = \varnothing$ 时重新回到起点；计算 $L(k,t)$，记录 $L(t)$.

步骤 3 按 $L(k,t)$，$k = 1,2,\cdots,m$ 的值，从 $H(t)$中选出 m_1 个本次循环的最优解集 $H_1(t)$.

步骤 4 若进入停滞状态，启用新的信息素调整对策：用基于回路长度的轮盘赌方式从 $H(t)$选择 m_2 条回路组成退火解集 $H_2(t)$并按(2)式释放信息素；其余边上的信息素恢复为初始值 $\tau_{ij}(0)$.

步骤 5 若 $L(t+1) < OPTL(t)$，在 $H_1(t)$上按(2)式释放信息素，$OPTL(t+1) = L(t+1)$，否则只在 $OPTL(t)$上按(2)式释放信息素.

步骤 6 按(3)式更新每边的信息素 $\tau_{ij}(t)$.

步骤 7 对每边(i,j)，置 $\Delta\tau_{ij} = 0$，算法完成了第 t 次搜索.

步骤 8 $t = t+1$，当 t 小于预设的搜索次数 T，转至步骤 2.

步骤 9 输出得到的最短回路及其长度.

蚁群算法的计算量主要在于选择机制的实现，即步骤 2，在这一步骤上述算法与基本蚁群算法是相同的，并且实际应用中 m_1 和 m_2 都取得很小，所以不会影响算法的复杂性，而且信息素的释放范围比原来缩小，使得该算法的计算量不会超过文献[4]的计算量.

3 实验结果

用改进的算法对 2 个 TSP 问题的标准实例进行了测试，其中实例 CHN144 (即中国 144 城市问题)

引自文献[6], Grötschel问题引自文献[7], 后者规模达442个点, 这2个实例是著名的NP-hard问题. 由于本文主要讨论蚁群算法的过早收敛问题, 所以算法的性能评价用本算法求得的解 L 与实例目前已知最优解 $OPTL$ 的相对误差 $(L-OPTL)/OPTL$ 为比较标准.

1) CHN144实例. 取 $m=144$, 每个城市各放一个蚂蚁; $rand$ 为0.4至0.7间的随机数; $\rho=0.95$; $\alpha=1.0$; $\beta=5.0$; $Q=500$ (Q 的值对算法的收敛性影响不大).

用基本蚁群算法运行10次所得到的最好解为30 380.50, 按改进算法搜索到的最短回路长度在浮点运算下为30 355.1, 具体如下: 兰州—青铜峡—银川—榆林—延安—临汾—运城—洛阳—郑州—开封—安阳—邯郸—长治—太原—石家庄—保定—天津—北京—张家口—大同—呼和浩特—包头—二连浩特—锡林浩特—承德—唐山—秦皇岛—锦州—营口—鞍山—沈阳—四平—长春—哈尔滨—白城—齐齐哈尔—海拉尔—满洲里—黑河—同江—佳木斯—鸡西—牡丹江—图们—通化—丹东—大连—烟台—荣城—青岛—潍坊—济南—济宁—徐州—连云港—蚌埠—合肥—芜湖—南京—扬州—无锡—上海—杭州—宁波—椒江—温州—金华—黄山—安庆—景德镇—鹰潭—南昌—九江—黄石—武汉—信阳—南阳—襄樊—十堰—西安—宝鸡—天水—汉中—绵阳—成都—宜宾—重庆—遵义—贵阳—都匀—柳州—桂林—怀化—常德—宜昌—沙市—岳阳—长沙—萍乡—衡阳—郴州—韶关—赣州—龙岗—三明—福州—台北—台中—高雄—厦门—汕头—深圳—香港—澳门—广州—肇庆—湛江—海口—三亚—北海—南宁—凭祥—个旧—昆明—六盘水—西昌—攀枝花—大理—畹町—拉萨—日喀则—和田—喀什—阿克苏—塔城—阿勒泰—乌鲁木齐—哈密—敦煌—格尔木—德令哈—玉门—张掖—西宁.

这一结果是该实例目前在浮点运算下的最优解, 所求得的最短回路与由分段并行算法[4]、佳点集遗传算法[5]等得到的最短回路是相同的.

2) Grötschel实例. 文献[7]记录的用遗传算法求得的最优解为51.21, 尚未见到有文献用蚁群算法求解这一问题的.

实验选取参数如下: $m=442$, 每个城市各放一个蚂蚁; $\rho=0.98$, $\alpha=1.8$, $\beta=1.7$, $Q=1$. 经计算机求解发现基本蚁群算法的过早收敛性比较明显, 运行10次所得到的最优解为55.50, 相对误差在8%以上按改进算法搜索到的最短路径长度为51.30, 相对误差为0.175 7%. 相比之下解的质量得到明显提高, 说明该算法具有逃离局部最优解的能力.

参考文献

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

[2] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithm and stigmery[J]. *Future Generation Computer Systems*, 2000, 16(8): 851—871.

[3] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240—1245.

[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] 张 铃, 张 钹. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 917—922.

[6] 康立山, 谢 云, 尤矢勇, 等. 非数值并行算法: 模拟退火算法[M]. 北京: 科学出版社, 1994.

[7] 刘 勇, 康立山, 陈毓屏. 非数值并行算法: 遗传算法[M]. 北京: 科学出版社, 1995.

[8] 王晓东. 计算机算法分析与设计[M]. 北京: 电子工业出版社, 2001. 114—146.

[9] Lewis H R, Papadimitriou C H. *Elements of the Theory of Computation* [M]. 北京: 清华大学出版社, 1999. 275—300.

(责任编辑 惠松骐)