具有多态特征和聚类处理的蚁群算法

段凤玲,李龙澍,曹文婷
(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:现实蚁群中,蚁群的觅食是一种典型的聚类行为,文中针对一些带聚类特征的TSP,提出了新型的带聚类处理的多态蚁群算法。该算法思想是根据聚类特征对TSP中的城市进行处理,将待求问题分成许多小规模的子问题。对于每个子问题,融合多态蚁群算法,引入不同种类的蚁群。通过对每个子问题进行求解,得到类内最短距离。最后按文中给出的规则合并所有子问题的解得到最优解。算法实验测试结果表明,该算法能将局域搜索与全局搜索相结合,极大提高了算法的收敛速度和求解速度。

关键词:蚁群算法;旅行商问题;聚类;多态蚁群算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2009)12-0077-04

Ant Colony Algorithm with Polymorphism and Clustering Processing

DUAN Feng-ling, LI Long-shu, CAO Wen-ting
(School of Computer Science and Engineering, Anhui University, Hefei 230039, China)

Abstract: In reality, the act of finding food for ant is a typical clustering behavior. Aiming TSP with clustering, raise a new algorithm, which is integration of polymorphic ACA and improved ACA with clustering. The thought of the algorithm decomposes TSP into many sub-problems based on the characteristic of clustering, and then solves every sub-problem. While solving every sub-problem, it merges polymorphic ACA, introduces different types of ant colony, and merges Local Search and Global search. At last, it merges all the solutions of the sub-problems, and gets the bestanswer of the problem. Experiment result has proved that it can greatly improve convergence rate and solution speed of algorithm.

Key words: ant colony algorithm; TSP; clustering; polymorphic ACA

0 引 言

近些年来,伴随着模拟自然与生物机理为特征的仿生优化算法时代的兴起,一些求解复杂问题的仿生优化算法相继出现[1,2],如蚁群算法、遗传算法、人工免疫算法等。蚁群算法是一种模拟蚂蚁群体智能行为的仿生优化算法,由意大利学者Macro Dorigo[3,4]提出,并成功的用于解决旅行商问题(TSP),作业调度问题(JOP)等组合优化问题,取得了较好的实验数据。

目前人们对蚁群算法的研究已经从当初单一的旅行商问题领域渗透到多个应用领域,由解决一维静态组合优化问题发展到解决多维动态组合优化问题,由离散域范围内的研究逐渐扩展到连续域范围内的研究,而且在蚁群算法的硬件实现上也取得了很多突破性的进展。该算法最早成功应用与解决著名的TSP问题,具有较强的鲁棒性,优良的分布式计算机制,易于与其它方法相结合等优点,但搜索时间长,易陷入于局部最优解是其突出的特点[5]。

1 蚁群算法的基本原理

蚁群能够适应环境的变化,当蚁群路径上突然出现障碍物时,虽然单个蚂蚁的选择能力有限,但是通过信息素作用的调整,整个蚁群行为具有非常高的自组织性,最终通过蚁群的集体自催化行为找到最优路径,如图1所示。

设A为蚁穴,D为食物源,EF为一障碍物,由于障碍物,蚂蚁只能从经A经E或F到达D,或由D到达

收稿日期:2009-04-01;**修回日期**:2009-06-30
基金项目:国家自然科学基金项目(602730430);安徽省自然科学基金项目(050420204);安徽省教育厅自然科学基金项目(2006KJ098B);安徽省高校拔尖人才基金项目(05025120);安徽大学研究生创新基金项目(20073005)
作者简介:段凤玲(1983-),女,安徽阜阳人,硕士研究生,研究方向为蚁群智能优化算法;李龙澍,教授,博士生导师,研究方向为不精确信息处理技术和智能软件开发技术。

A。在初始时刻，由于 BF，FC，BE，EC 上均无信息素，位于 A 和 D 的蚂蚁可以随机选择，从统计学的角度看，可认为蚂蚁以相同的概率选择它们，经过一段时间后，在路径 BFC 上的信息量是路径 BEC 的信息量的 2 倍。随着时间的推移，蚂蚁将会以越来越大的概率选择 BFC，最终找到蚂蚁到食物源的最短路径。

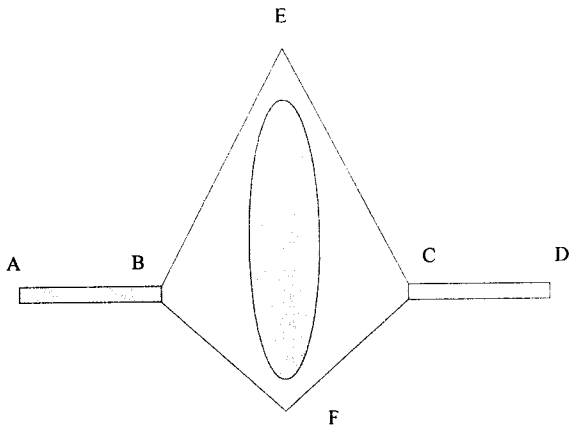

图 1 蚂蚁觅食原理模拟

下面以 TSP 问题为例，介绍蚁群算法的数学模型。设 $C=\{c_1,c_2,c_3,\cdots,c_n\}$ 是 n 个城市的集合，$L=\{l_{ij}\mid c_i,c_j\subset C\}$ 是集合 C 中元素两两的连接的集合，$d_{ij}(i,j=1,2,\cdots,n)$ 是 l_{ij} 的距离，$b_i(t)$ 表示 t 时刻位于城市 i 的蚂蚁的个数，$\tau_{ij}(t)$ 表示 t 时刻在 i 和 j 上连线上残留的信息量，n 为 TSP 的规模，m 为蚂蚁总数，则 $m=\sum_{i=1}b_i(t)$；$\Gamma=\{\tau_{ij}(t)\mid c_i,c_j\subset C\}$ 是 t 时刻 C 中元素两两连接 l_{ij} 上残留信息量的集合。在初始时刻各条路径上的信息量相等。

在搜索中，$P_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 由元素 i 到 j 的状态转移概率，公式如下：

$$P_{ij}^k(t)=\begin{cases}\dfrac{T_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\subset a_k}T_{is}^{\alpha}(t)\beta_{is}^{\beta}(t)}, & \text{若 } j\in a_k\\ 0 & \text{otherwise}\end{cases}\tag{1}$$

上式中，蚂蚁 k 下一步允许选择的城市为 $a_k=\{C-\text{tabu}_k\}$，α 为信息启发因子，β 为期望启发式因子，$\eta_{ij}(t)$ 为启发函数，其表达式如下：

$$\eta_{ij}(t)=\frac{1}{d_{ij}}\tag{2}$$

$t+n$ 时刻在路径 (i,j) 上的信息量可按如下规则调整：

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

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^k(t)\tag{3}$$

上式中，ρ 表示信息挥发系数，$1-\rho$ 表示信息残留因子，$\Delta\tau_{ij}(t)$ 表示本次循环中信息素的增量，在 Ant-Cycle 中，若蚂蚁 k 在本次循环中经过边 (i,j)，$\Delta\tau_{ij}^k(t)=\frac{Q}{l_k}$，否则，信息量增量为 0，式中 l_k 表示第 k 只蚂蚁在本次循环中所走路径的总长度。

2 带有多态和聚类特征的蚁群算法设计

2.1 多态蚁群算法的原理

基本蚁群算法中，每只蚂蚁在选择下一个城市时，多数情况下搜索子空间太大，单种蚁群，没有考虑蚂蚁种群的种类差异和动作差异，不能完全反映真实社会的复杂性。

寻优收敛速度慢，笔者引入不同种类的蚁群，每种蚁群都有不同的信息素调控机制，将局域搜索和全局搜索相结合，能够使算法的收敛速度大大提高。

引入多种蚁群易于找到更好的局部搜索方法，将蚁群分为探测类 A、搜索类 B 和取食类 C。A 类蚁群是将选择蚂蚁放在城市的中心，探测其它的可选范围 MAXPC 中的城市，这里的 MAXPC 来源于文献[6]，第 k 只蚂蚁由城市 i 选择下一个城市时，需按概率 p_{ij} 在剩下的 $m-x$ 中选择，只需在剩下的 $\{m-x,\text{MAXPC}\}$ 中选择。当城市规模 $\leqslant 100$ 时，$\text{MAXPC}<10$。当 $100\leqslant$ 城市规模 $\leqslant 1000$ 时，$\text{MAXPC}<20$[7]。

2.2 融合多态的带聚类处理的蚁群算法

如果一个 TSP 问题规模很小，用蚁群算法具有极高的求解性能，而对于大型规模的多城市 TSP 问题，则容易陷入局部最优，而且收敛速度慢，所以采取一种新方法，将问题分解为多个子问题，最后将每个小规模子问题的解合并成待求问题的解，即为新型的带聚类处理的蚁群算法。

设 TSP 问题的城市规模为 n，根据分布特征选择不同的聚类算法，应为基于距离的聚类算法，如果该 TSP 问题的类内模式为球状分布，选择 C-均值算法，非球状模式则选择近邻函数法。如果 m 为城市的类数，每个类 $c(c=1,2,\cdots,m)$ 有 k_c 个城市数目，则 $\sum_{c=1}^{m}k_c=n$，有两个城市 $w_1^c,w_{k_c}^c$ 分别与另外两个不同的类相连，称 $w_1^c,w_{k_c}^c$ 为类 c 的边界城市。

把每个类 c 本身看成一个复杂城市 c，由这 m 个复杂城市构成一个新的 TSP，该新的 TSP 的解即为所有类的连接顺序。在每个类中，从一个边界城市到下一个边界城市的路径是待求的 TSP 解的组成部分，且是最短路径。

在每个复杂城市 c 中，在求解类内最短路径中，采取多态蚁群算法，引入不同种类的蚁群，每种蚁群都有不同的信息素调控机制且只能在自己的搜索空间内查找，将局域搜索和全局搜索相结合，此时全局搜索空间

为除去该类中边界城市 $w_1^c, w_{k_c}^c$ 以外的城市。采用该算法能够大大提高收敛速度。

3 具有多态和聚类特征的蚁群算法实现

将 m 个复杂城市组成新的 TSP[8,9],选取每个复杂城市的中心为 $o_c(c = 1,2,\cdots m)$,其中 o_c 可以不是整个TSP中的城市,将 o_c 看成一个TSP,利用蚁群算法求解,求得的解的顺序即为各类的连接顺序。

如何确定类中边界城市,笔者采取求类中最短距离的方法,通过找到最短距离所在的两个城市,设类 p、类 q 按类的连接顺序相邻,$u_{pc}(c = 1,2,\cdots,k_c)$ 为类 p 中的城市,$u_{qk}(k = 1,2,\cdots,k_q)$ 为类 q 中的城市,则类 p、类 q 的边界城市由公式(4)确定:

$$\{u_{pm}, u_{qn}\} = \arg\min d(u_{pc}, u_{qk}) \quad (c = 1,2,\cdots, k_p, k = 1,2,\cdots,k_q) \tag{4}$$

在求解类内最短路径时,采用多态蚁群算法,在类 c 中,选择 k_c 只蚂蚁放到 k_c 中,每个探测蚁所在位置为中心,探测其它的 $k_c - 1$ 个城市,根据公式(5)得到探测素,记为 b_{ij},公式(5)如下:

$$b_{ij} = \frac{\min_{ij}}{\max_{ij}} \tag{5}$$

如果城市 j 不在 i 的 MAXPC 中,则 $b_{ij} = 0$。(5)式中,$\min_{ij}$ 表示以 i 为中心到其它 $k_c - 1$ 个城市的最小距离,$\max_{ij}$ 表示以 i 为中心到其它 $k_c - 1$ 个城市的最大距离。初始时刻各条路径上的信息素如下:

$$\tau_{ij}(0) = \begin{cases} C \cdot b_{ij} & \text{若城市 } j \text{ 在城市 } i \text{ 的 MAXPC 内} \\ C\dfrac{\min_{ij}}{\max_{ij}} & \text{否则} \end{cases} \tag{6}$$

该侦查素可为搜索蚁提供状态转移 P_{ij}^k 的计算,为各路径上的信息量调整提供辅助参考。搜索蚁在蚂蚁 $k(k = 1,2,\cdots,n)$ 在运动过程中的 t 时刻,从城市 i 转移到城市 j 的状态转移概率可按公式(7)计算:

$$P_{ij}^k(t) = \begin{cases} \dfrac{T_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \notin \text{tatu}_k} T_{is}^{\alpha}(t)\beta_{is}^{\beta}(t)} & \text{若 } j \notin \text{tatu}_k, \text{且 } b_{ij} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

蚂蚁完成一次循环,根据公式(3)调整信息素,当若蚂蚁 k 经过 i,j 且 $b_{ij} \neq 0$ 时,信息素的增量可根据下面公式:$\Delta\tau_{ij} = \dfrac{Q \cdot (\frac{\min_{ij}}{\max_{ij}})}{l_k}$ 计算,否则增量为 0,且 $\Delta\tau_{ij} = \sum_{k=1}^{k} \Delta\tau_{ij}^k$。

4 算法设计步骤

(1) 观察 TSP 中的城市,对城市进行聚类处理,根据城市分布情况选用 C - 均值或邻近函数。

(2) 设城市可以聚成 $U_1, U_2, \cdots, U_m$ 类,且 o_c 为城市的中心。把 $o_c(c = 1,2,\cdots,m)$ 看成一个 TSP,使用蚁群算法求解,设该解为 $L_{C1}, L_{C2}, \cdots, L_{Cm}, L_{C1}$,找到各类路径的连接顺序。

(3) 利用公式(4)计算各类的边界城市。

(4) 对每个类 $U_c(c = 1,2,\cdots,m)$,选择 k_c 只蚂蚁放到 k_c 中,每个探测蚁以所在位置为中心,探测其它的 $k_c - 1$ 个城市,根据公式(5)得到探测素,记为 b_{ij}。

(5) 将类内路径上的信息量按公式(6)计算,置迭代次数为 0。

(6) 将每只搜索蚁放在除边界城市以外的城市上,按公式(7)计算转移位置,直至搜索完成一个循环,计算除边界城市以外的城市组成的所有解,设该目标函数值为 $p_k(k = 1,2,\cdots,m)$,将 p_k 分别联合两个边界城市后再做计算比较,得到当前从一个边界城市到另一个边界城市的最优解 $P_c(c = 1,2,\cdots,m)$。

(7) 若所求的解在一定迭代次数之内无明显改进,则转步骤(8),否则,按公式(3)更新路径上的信息量,置 $\Delta\tau_{ij}$ 为 0,指令计数器加 1。

(8) 输出类内最优解。

(9)把所有类内最优解的城市连接起来,把类和类之间的边界城市按步骤(2)求得的连接顺序连接起来,构成 TSP 问题的一个可行解。

(10)对解进行局部搜索,记录本次迭代最优解,如果优于当前最优解,则用其替换当前最优解。

(11)若不满足算法结束条件,则跳转步骤(2),若满足,则退出算法。

5 仿真实验结果

选用 TSPLIB 中的实例,对 D2103、Pr136、Pr2392、D2319,通过基本蚁群算法与该文进行对比实验,根据不同的 TSP 规模,选用不同的参数设置如表 1 和表 2 所示,用文中算法和蚁群算法各运行 5 次,得到实验结果如表 3 所示。

表 1　文中算法针对不同规模 TSP 实例的参数设置

参数 / 设置	ρ	α	β	蚂蚁数	迭代次数
D2103	0.1	1.0	2.0	20	200
Pr136	0.1	1.0	2.0	30	100
Pr2392	0.1	1.0	2.0	10	300
D2319	0.1	1.0	2.0	20	200

6 结束语

文中提出的算法比较适用于对大规模城市的TSP,对于不同分布特征的城市,算法的优越性能有所不同,如果城市的分布带有明显的聚类特征,则用该算法具有明显的优越性,收敛速度快且能获得较好的解。采用带聚类处理的蚁群算法,可以大大加快蚁群算法的寻优速度,在进行聚类处理分类后引入多态蚁群来求解类内类最短路径,可以近一步加快算法的求解速度。与使用基本的蚁群算法相比该算法解的性能得到显著提高,有利用蚁群算法的应用和推广。

表 2 文中算法针对不同规模 TSP 实例的参数设置

参数 / 设置	ρ	α	β	τ_0
D2103	0.1	1.0	2.0	$(2103*8450)^{-1}$
Pr136	0.1	1.0	2.0	$(136*96772)^{-1}$
Pr2392	0.1	1.0	2.0	$(2392*378032)^{-1}$
D2319	0.1	1.0	2.0	$(2319*234256)^{-1}$

表 3 基本蚁群算法与文中算法求解 TSP 的结果对比

TSP 实例	ACA		文中算法		已知	运行时间比
	所求解	误差比(%)	所求解	误差比(%)	最优解	(ACA:文中)
D2103	88306	9.77	81204	0.94	80456	1:35
Pr136	100213	3.56	97208	0.45	96772	1:11
Pr2392	426218	12.75	392698	3.84	378032	1:17
D2319	262114	11.89	245371	4.58	234256	1:4

参考文献:

[1] Dorigo M, Maniezzo V, Colorni A. The Ant System: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 29 - 41.

[2] Dorigo M, Gambardella L M. Ant clolny system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53 - 56.

[3] Dorigo M, Aniezzo B M, Colorni A. Positive feedback as a search strategy[R]. [s. l.]: Dipartimento di Elettronica, Politecnico di Milano, IT, 1991: 23 - 101.

[4] 卢荣德,陈宗海,王 雷. 复杂工业过程计算机建模、仿真与控制的综述[J]. 系统工程与电子技术,2002,24(1):27 - 30.

[5] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社,2005:24 - 171.

[6] 吴 坚,王常禄. 中国蚂蚁[M]. 北京:中国林业出版社,1995.

[7] 李艳君. 拟生态系统算法及其在工业过程控制中的应用[D]. 杭州:浙江大学,2001:21 - 195.

[8] 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(2):2 - 4.

[9] 全惠云,文高进. 求解 TSP 的子空间遗传算法[J]. 数学理论与应用,2002,22(1):36 - 39.

(上接第 76 页)

来的字符会是:

New_add.jsp,News_del.jsp,News.acion

那么在需要做控制的页面首先利用:

```
String url = servletRequest.getRequestURI();
url=url.substring(url.lastIndexOf("/") + 1, url.length());
```

取出当前页的文件名,再将此文件名与该用户的权限字符做比较,如果该文件名在权限字符中可以找到,则表明该用户拥有此页面操作的权限。

5 结束语

对于静态组织机构型的企业,权根管理是依据组织机构的划分和行政级别来确定操作权限的;对于动态项目型的企业,在权限管理中是依据用户在项目中的不同角色来确定操作权限的。而对于混合型的企业而言,权限管理常常是以项目角色作为主线,以行政级别和组织机构作为补充。但是不管对于哪种应用场合,ACL(访问控制列)都可以比较灵活有效地实现用户要求的权限管理。

参考文献:

[1] 张晓群,董丽丽. 角色访问控制模型的研究及应用[J]. 计算机技术与发展,2007,17(2):42 - 45.

[2] 尹 涛,李 翔,林 祥. 基于 AOP 的角色访问控制模型设计与实现[J]. 计算机技术与发展,2008,18(10):136 - 138.

[3] 谷玉奎,曹宝香,袁玉珠. 基于 SOA 的通用权限管理服务[J]. 计算机技术与发展,2008,18(6):70 - 72.

[4] 王 昊,赵文静,边根庆. 基于三方通信构架电子政务安全系统的研究[J]. 计算机技术与发展,2007,17(10):242 - 244.

[5] 朱 磊. 一种面向服务的权限管理模型[J]. 计算机学报,2005,28(4):677 - 685.

[6] Yang F Q. Thinking on the development of software engineering technology[J]. Journal of Software, 2005, 16(1): 17 - 20.

[7] Sandhu R, Coyne E J, Feinstein H L. Role - Based access control models[J]. IEEE Computers, 1996, 29(2): 38 - 47.

[8] Fei Y K, Wang Z J. A concept model of Web components [C]//In: Proceedings of IEEE International Conference on Services Computing. Shanghai: [s. n.], 2004: 159 - 164.