第27卷 第6期 福州大学学报(自然科学版) Vol.27 No.6
1999年12月 Journal of Fuzhou University(Natural Science) Dec. 1999

文章编号：1000-2243(1999)06-0005-05

② 5-9 022.4

凸整数规划问题的混合蚁群算法

林 锦，朱文兴

(福州大学计算机科学与技术系，福建 福州 350002)

摘要：混合蚁群算法是基于群体的一类仿生算法，适合于解困难的组合最优化问题．本文对其做适当改进，用于解凸整数规划问题．结果表明：用该算法求目标函数为正定二次型的整数规划问题的最小值，找到的解比多起始点局部搜索方法好得多，比原来的混合蚁群算法找到更好的解．

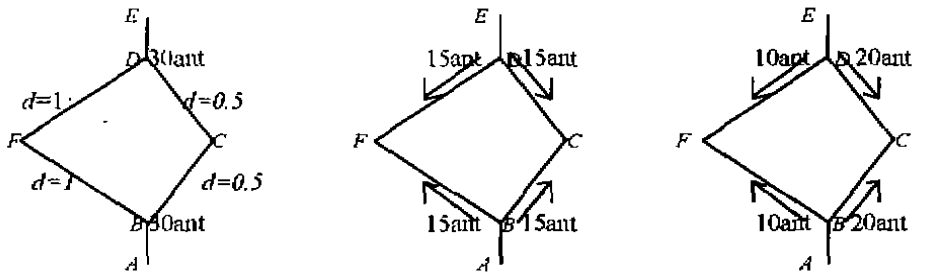
关键词：蚂蚁系统；蚁群优化；启发式

中图分类号：TP301.6 **文献标识码**：A

凸整数规划 混合蚁群算法 组合优化

蚁群算法是模仿蚂蚁工作方式的一种新的启发式算法．生物学研究表明一群互相协作的蚂蚁能够找到食物源和巢之间的最短路径，而单只蚂蚁则不能．蚂蚁间相互协作的方法是它们在所经过的路上留下一定数量的信息素(迹)，该迹能被其它蚂蚁检测出来，一条路上的迹越多，其它蚂蚁将以越高的概率跟随此路径，从而该路径上的迹会被加强．现举例说明：

假设蚂蚁以 1 单位长度 / 单位时间的爬行速度往返于食物源 A 和巢 E(见图 1)，其中 d 为距离，每过一个单位时间各有 30 只蚂蚁离开巢和食物源 (图 1(a)).



(a) 各有 30 只准备离开 B 和 D　(b) 各有 15 只选择 C 和 F　(c) 20 只选择 C，10 只选择 F

图 1　短路径被越来越多的蚂蚁选中

假设 $T=0$ 时，有 30 只蚂蚁在点 B 和点 D(图 1(b))．由于此时路上无迹，蚂蚁就以相同的概率走 2 条路中的一条，因而 15 只蚂蚁选择往 C，其余 15 只选择往 F．$T=1$ 时，路径 BCD 被 30 只蚂蚁爬过，而路径 BFD 只被 15 只蚂蚁爬过，从而 BCD 上的迹的数量是 BFD 的 2 倍．此时，又有 30 只蚂蚁离开 B(和 D)，于是各有 20 只蚂蚁选择往 C，另外 10 只蚂蚁选择往 F，这样更多的迹被留在更短的路径 BCD 上，这个过程一直重复，短路径 BCD 上的迹以更快的速度增长，越来越多的蚂蚁选择这条短路径．

收稿日期：1999-01-15

作者简介：林 锦 (1970—)，女，硕士研究生．

基金项目：福建省自然科学基金资助项目 (F97006)

通过上例可知蚂蚁工作方式的本质是：① 迹越多的路径，被选中的概率越大，即选则机制；② 路径越短，在上面的迹增长得越快，即迹更新机制；③ 蚂蚁之间通过迹进行通信，即协同工作机制.

选择机制和迹更新机制构成正反馈机制，在蚁群的协同作用下，正反馈过程使得越来越多的蚂蚁选择最短的路径. Dorigo 于 1991 年创建了蚂蚁算法的模型. 该模型已成功应用于求旅行商问题 (TSP)[1, 2]，二次指派问题[3]，排序问题[4] 等 NP − 困难的组合最优化问题，结果可与模拟退火，遗传算法等通用的启发式算法相媲美. 蚁群算法和局部搜索算法相结合（称为混合蚁群算法）应用于解二次指派问题和排序问题，得到的结果可以与专用算法相媲美[3, 4]. 由于大量的组合最优化问题可化为凸整数规划问题，故本文改进混合蚁群算法并用于解凸整数规划问题:

$$(\mathrm{P}) \qquad \begin{cases} \min\ f(x) \\ \text{s.t.}\quad 0 \leqslant x_i < L_i \qquad (x_i \text{是整数},\ i = 1,\ \cdots,\ n) \end{cases}$$

其中: $x = (x_1,\ \cdots,\ x_n)$, $f(x)$ 是凸函数且 $f(x) > 0$. 实验结果表明，该算法比多起始点局部搜索算法和蚁群算法得到好得多的解.

2 混合蚁群算法

混合蚁群算法是结合局部搜索算法的蚁群算法. 下面用 TSP 问题为例来解释混合蚁群算法. 把 TSP 问题表示为一个图 $G = (V,\ E)$, V 是表示城市的点集，E 是连接城市的边. 算法开始时，把 m 只蚂蚁按一定规则分布在各城市，每只蚂蚁的任务是建立一个解，即从各自的起始城市开始完成一个环游. 环游是一个城市接着一个城市地建立的，位于城市 r 的蚂蚁 k 根据某种选择策略选择下一个要去的城市 s(选择策略)，选择原则就是在与城市 r 连接的所有边中，边长越短，边上的迹越多的边，蚂蚁就以越高的概率选中这条边，当到达下一个城市时，蚂蚁就适当减少边 $(r,\ s)$ 上的迹（局部更新），从而其它蚂蚁以较低概率选择这条边，以免得到许多相似的解. 当所有蚂蚁都建立了各自的解，则以各自的解为起始点用某种局部搜索算法（例如 3 − OPT) 求局部最优解. 最后，根据局部最优解的好坏更新其边上的迹，原则是越短的环游增加越多的迹（全局更新). 迹更新结束后，开始下一轮迭代. 算法概要如下:

Step1: 初始化迹.

Step2: 下述过程迭代 $I_{\max}$ 次:

a) 每只蚂蚁($k = 1,\ \cdots,\ m$)根据选择策略建立一个新解同时局部更新迹;

b) 以所建立的解为起始点，求局部最优解;

c) 根据局部最优解全局更新迹.

3 解凸整数规划问题的混合蚁群算法

把凸整数规划问题 (P) 表示为一个二部图 $G = (V,\ U,\ E)$, V 是 n 个点的集合，分别表示解的的 n 个分量，U 是 L 个点的集合 ($L = \max\{L_i - 1,\ i = 1,\ \cdots,\ n\}$), $L_i - 1$ 是分量 i 的最大可能取值，V 中的点 i 与 U 中的点 0，1，…，$L_i - 1$ 有边相连，$q(i,\ j)$ 是边 $(i,\ j)$ 上的迹，如果 V 中的的点 i 与 U 中的点 j 无边相连，则 $q(i,\ j) = 0$，迹矩阵 $Q(i,\ j)_{n*L}$ 是一个 n 行 L 列矩阵.

算法开始时，所有蚂蚁都集中在第 1 个分量处，每个蚂蚁按照选择策略选择一条边，

当选择完一条边后马上更新该边上的迹，当 m 个蚂蚁都选择好各自的第一个分量的值后，都集中到第二个分量处，直到选择完第 n 个分量的值，从而建立了 m 个解．以所建立的 m 个解为起始点，进行局部搜索，根据得到的局部最优解计算迹的增量，全局更新迹，全局更新迹后继续迭代直到满足停止条件，停止条件是最大迭代次数．

1) 初始化．由于混合蚁群算法利用局部最优解计算迹的增量，因而迹的初始值 $\mathrm{ini}q$ 取为 $\mathrm{ini}q = W / f_{\min}$，$W$ 是常量，蚂蚁总数取为 $m = L$.

2) 改进的选择策略．位于第 i 个分量的蚂蚁，按下列公式选择边 $(i,\ j)$:

$$j:=\begin{cases}\arg\max\limits_{s\in\{0,\ 1,\ \cdots,\ L_i-1\}} q(i,\ s) & (\text{若}q\leqslant q_0)\\ j_1 & (\text{其它})\end{cases} \tag{1}$$

其中：$q_0 = 0.9$，$0 < q < 1$，q 依均匀分布在 (0，1) 内随机取值，$\arg\max\limits_{s\in\{0,\ 1,\ \cdots,\ L_i-1\}} q(i,\ s)$ 表示与端点 i 相连的迹最大的边的另一端点，j_1 依如下概率分布在 $U\setminus \arg\max\limits_{s\in\{0,\ 1,\ \cdots,\ L_i-1\}} q(i,\ s)$ 内随机取值:

$$P_k(i,\ j):=\frac{q(i,\ j)}{\sum\limits_{s\in U\setminus \arg\max\limits_{s\in\{0,\ 1,\ \cdots,\ L_i-1\}} q(i,\ s)} q(i,\ s)},\quad j\in U\setminus \arg\max_{s\in\{0,\ 1,\ \cdots,\ L_i-1\}} q(i,\ s) \tag{2}$$

该公式表明，迹最大的边以高概率 0.9 被选中，其余的边以概率 0.1 参与选择，而在文献 [1，2] 中，j_1 的取值范围是所有可能取值，这样，按公式 (2) 选择时，迹最大的边仍然以很高的概率被选中，造成算法太快收敛到次优解．本文缩小了 j_1 的取值范围，使得蚂蚁不仅以概率 0.9 选择迹最大的边，而且以概率 0.1 选择非最大迹的边，这样，在每次迭代时可以建立不同的解，在利用最大迹的同时，加大对解空间的探测力度，使算法不会太快收敛．实验结果表明，改进的算法大大提高得到最优解的概率．

3) 局部更新．当蚂蚁 K 选中边 $(i,\ j)$ 后，就更新边 $(i,\ j)$ 上的迹:

$$q(i,\ j):=(1-\alpha)*q(i,\ j)+\alpha*\min q[i] \tag{3}$$

其中：$\min q[i]$ 是上轮迭代得到的迹矩阵中第 i 行前 i 个元素的最小值，这是动态变化的．从公式 (3) 知，更新后的迹是原来的迹与最小迹的凸组合，这使得迹的减小基于可供选择的 L_i 条边上的迹的相对大小．由于公式 (1) 以概率 0.9 选择 L_i 条可供选择的边中迹最大的一条，因而蚂蚁常常选中迹最大的边，第一只蚂蚁选中它后，就更新该边上的迹，使迹的值小了一点；第二只蚂蚁选中它后，也是这样，结果是当迹最大的边被多次选中后，迹的量减少到 L_i 条可供选择的边上的迹的值的平均水平，从而蚂蚁选中其它边的概率增加，增加了所建立的解的多样性．

4) 局部搜索算法．当 m 只蚂蚁都建立好各自的解，分别以这些解为起始点，作局部搜索．局部搜索的邻域定义为 $N(x)=\{x,\ x+e_i,\ x-e_i,\ i=1,\ \cdots,\ n\}$，其中 e_i 是第 i 个分量为 1，其余分量为 0 的 n 维向量．邻域搜索算法如下:

Step1: 任给一初始整点 $x_0\in\{(x_1,\ \cdots,\ x_n):\ 0\leqslant x_i\leqslant L_i-1,\ i=1,\ \cdots,\ n\}$;

Step2: 若 x_0 是问题 (P) 的局部极小解，停机；否则，检查 x_0 的邻域 $N(x_0)$，找到一整点 $x\in N(x)\bigcap\{(x_1,\ \cdots,\ x_n):\ 0\leqslant x_i\leqslant L_i-1,\ i=1,\ \cdots,\ n\}$，使得 $f(x)\leqslant f(x_0)$.

Step3: 令 $x_0:=x$，转Step2.

5) 全局更新．当所有蚂蚁都得到局部最优解，就全局更新所有边上的迹，公式如下:

$$q(i,\ j):=(1-\alpha)*q(i,\ j)+\alpha*\Delta q(i,\ j) \tag{4}$$

$$\Delta q(i,\ j)=\sum_{k=1}^{m}\Delta q^k(i,\ j)$$

$$\Delta q^k(i,\ j)=\begin{cases} W/f(k) & \text{(如果蚂蚁}k\text{选中}j\text{作为第}i\text{个分量的值)} \\ 0 & \text{(其它)} \end{cases}$$

其中：α是参数，$0<\alpha<1$；$q(i,\ j)$是边$(i,\ j)$上的迹的增量；$f(k)$是蚂蚁k建立的解做局部搜索后得到的局部最优解的目标函数值．公式(4)表明被越多蚂蚁访问过的边得到越多的迹的增量，同时所有的边上的迹都蒸发掉一部分，避免迹的无限增长．

下面具体给出解凸整数规划问题的混合蚁群算法的类pascal描述．

1) 初始化阶段．

令 $m:=l$；$\alpha:=0.1$；$q:=0.9$；$W:=10$；在$\{x:0\leqslant x_i\leqslant L_i-1,\ x_i$是整数，$i=1,\ \cdots,\ n\}$内随机产生$m$个解作为起始点做局部搜索，求出最小的局部最优解$x_{\min}$，以及相应的目标函数值$f_{\min}$；令初始迹 iniq $=W\diagup f_{\min}$；并为每条边赋初始值$q(i,\ j):=$iniq; for $i:=1$ to n do; min$q[i]:=$iniq;

2) 迭代．

for num $:=1$ to $I_{\max}$ do

① for $i:=1$ to n do

for $k:=1$ to m do

begin

根据公式(1)求第i个分量的值j；$x,\ k(i):=j$;

按公式(3)更新边$(i,\ j)$上的迹$q(i,\ j)$;

end

② 以得到的m个解为起始点，根据本文给出的局部搜索算法求局部最优解及目标函数值；求出该轮迭代的最小目标函数值 $f\sim\min$; if $f\sim\min<=f_{\min}$ then

更新迭代以来的最小目标函数值$f_{\min}$及解$x_{\min}$，以及该轮迭代所得到的解；

③ for每条边$(i,\ j)$ 根据公式(5)计算迹增量$q(i,\ j)$;

for每条边$(i,\ j)$ $q(i,\ j):=(1-\alpha)*q(i,\ j)+\alpha*\Delta q(i,\ j)$;

④更新迹矩阵每行的最小值min$q(i)$;

3) 输出最小目标函数值$f_{\min}$以及相应的解$x_{\min}$．

4 实验结果

下面用该算法解正定二次型整数规划问题:

$$\begin{cases} \min(x-1)^{\mathrm{T}}C(x-1)+1 \\ \text{s.t.}\quad 0\leqslant x_i\leqslant L \end{cases} \qquad (x_i\text{是整数},\ i=1,\ \cdots,\ n)$$

其中：$x=(x_1,\ x_2,\ \cdots,\ x_n)$，$C$是随机产生的正定矩阵．表1给出了用3种算法解4个正定二次整数规划问题的结果，其中C_0，C_1，C_2，C_3是不同的正定矩阵；multi－start是多起始点局部搜索算法；原HAC是原混合蚁群算法；改进的HAC是本文第三部分描述的算法，最大迭代次数为$I_{\max}=200$．其中Best列中的数值是15次实验所得到的最小目标函数值，方括号中的数值表示取得该值的次数，Ave列中的数值是15次实验的目标函数值的平均值，偏差是这些目标函数值的均方差．

表 1　3 种算法的计算结果比较

参数	Multi-start			原 HAC			改进的 HAC		
	Best	Ave	偏差	Best	Ave	偏差	Best	Ave	偏差
$n=5,\ L=100,\ C_0$	18	73.2	66.2	1 [1]	24	76	1 [12]	3.3	22.3
$n=5,\ L=100,\ C_1$	52	56	25.7	1 [7]	28	96.6	1 [11]	14	85.6
$n=5,\ L=100,\ C_2$	3 600	4 093	714	1 [6]	522	4 690	1 [9]	2.3	7.0
$n=10,\ L=100,\ C_3$	48	292	592	1 [5]	17	126	1 [6]	1.7	2.4

从表 1 可见，多起始点局部搜索算法的性能很差，然而结合了该算法的混合蚁群算法的性能都比较好；改进的 HAC 的性能较原 HAC 大大提高，取得最优解的次数增加，平均解的质量大大提高.

表 2　2 种算法的计算结果比较

$n=5,\ L=100,\ C_2$	原 HAC		改进的 HAC	
	Best	Ave	Best	Ave
$I_{max}=100$	1 [6]	522	1 [3]	39.8
$I_{max}=200$	1 [6]	522	1 [9]	2.7

从表 2 可见，把迭代次数 $I_{max}=100$ 增至 $I_{max}=200$，原 HAC 不能再提高解的质量，而改进的 HAC 增加了找到最优解的次数，并大大提高了平均解的质量.

参考文献:

[1] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agent[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29～41.

[2] Luca M Gambardella, Marco Dorigo. Ant-Q: a reinforcement learning approach to the traveling salesman problem[A]. Proceeding of ML-95, Twelfth Intern Conf on Machine Learning[C]. Morgan: Kaufmann, 1995. 252～260.

[3] Gambardella L M, Taillard E D, Dorigo M. Ant colony for the QAP[R]. Technical Report IDSIA, 1997.

[4] Gambardella Luca Maria, Dorigo Marco. HAS-SOP: Hybrid ant system for the sequential ordering problem[R]. Technical Report IDSIA, 1997.

A Hybrid Ant Colony System for the Convex Ineger Programing Problem

LIN Jin, ZHU Wen-xing

(Department of Computer Science and Technology, Fuzhou University, Fujian Fuzhou 350002, China)

Abstract: The hybrid ant colony system(HAC), a class of population-based meta-heuristic algorithm, is suitable for solving hard combinatorial optimization problems. In this paper it is improved appropriately and applied to the convex integer programming problems. Computational results show that our improved HAC is much more efficient than the multi-start local search and outperforms the original HAC.

Keywords: ant systems; ant colony optimization; meta-heuristics