

# 基于 MATLAB 的改进型基本蚁群算法

李虹, 孙志毅

(太原重型机械学院, 太原 030024)

**摘要:**蚁群算法是一种新型的模拟进化算法。是继 GA、SA、TS 等算法之后求解组合优化问题的一种新思路。人工蚁群算法通过模拟蚁群搜索食物的行为,采用正反馈结构、分布式计算与某种启发式算子相结合的方法,能够很快地发现较好解。本文给出一种基于 MATLAB 的改进型基本蚁群算法,有效地降低了算法的复杂度,缩短了搜索时间,具有较强发现最好解的能力。

**关键词:**蚁群算法;改进

**中图分类号:**TP301.6 **文献标识码:**A

## 0 引言

蚁群算法(Ant Colony Algorithm)是最近几年才由意大利学者 M. Dorigo, V. Manierio, A. Collorini 等人提出的一种新型的模拟进化算法。受到人们对自然界中真实蚁群集体行为研究成果的启发,考虑到蚁群搜索食物的过程与旅行商问题的相似性,利用蚁群算法求解旅行商问题(Traveling Salesman Problem)、指派问题(Assignment Problem)和调度问题(Scheduling Problem),取得了一些比较满意的实验结果。蚁群算法是一种适应性好、鲁棒性强,具有正反馈结构的并行算法。

## 1 蚁群系统基本原理

自然界中像蚂蚁这样几乎没有视力的昆虫有很多,他们是如何找到由其巢穴到食物源之最短路

径的?生物学家经过长期大量细致的观察研究后发现:最初单个蚂蚁的行为是随机的。蚂蚁在运动过程中会会在其经过的路径上留下一种叫做外激素(Pheromone)的信息物质。蚂蚁个体之间的信息传递就是依靠这种物质进行的。一方面,每只蚂蚁在其走过的线路上留下一定量的信息物质,且留在路径上的信息物质随时间逐渐衰减。另一方面,后来的蚂蚁能够感知这种外激素并以路径上残留信息量的多少指导其行为,信息量越大的路径,被选中的概率也越大。显然,蚁群搜索食物源的过程是信息量的一个正反馈过程。据此,蚁群可以快速找到由巢穴到食物源的最佳路径。图1所示为真实蚁群系统搜索食物时的路线示意图。图中A是蚁巢,E是食物所在的位置,HC为障碍物。

设D和H,B和H之间的距离均为2个单位,D与C、B与C之间的距离为1个单位,在一个时间单元内有30只蚂蚁由A到达B点,同样有30只蚂蚁由

收稿日期:2003-03-12

作者简介:李虹(1965-),女,电子信息分院副教授,研究方向为控制理论及其应用。

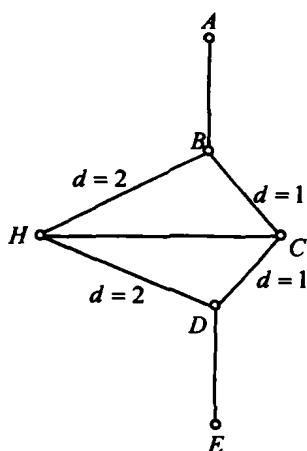


图1 蚁群搜索食物路线示意图

Fig.1 An example of real ant colony while searching food  
E到达D点,蚂蚁运动的速度是1单位距离/单位时间,每只蚂蚁在其走过的路径上留下一个单位的信息量。假设初始时刻 $t=0$ 时各条路径均无蚂蚁走过,位于B和D点的各30只蚂蚁选择所走路线的概率是相同的,按照统计规律,即有15只蚂蚁选择DH(BH),另外15只选择路径(BC)。由于 $d_{DH}=d_{BH}=2d_{DC}=2d_{BC}$ ,所以经过1个单位时间后,走过路径BC和DC的蚂蚁个数是走过BH、DH蚂蚁个数的两倍。这些蚂蚁留在路径上的信息量前者也是后者的2倍。在 $t=1$ 时,新的30只蚂蚁位于B点和D点,根据信息量的多少,他们选择路径DC(BC)的概率是选择DH(BH)的两倍,即有20只蚂蚁选择前者,而只有10只选择后者。这一过程一直继续下去,最终所有的蚂蚁都将选择由蚁穴至食物源的最短路径。

## 2 人工蚁群系统数学模型

我们不难发现蚁群寻找食物的过程与旅行商问题的相似性。利用蚁群系统原理求解一个城市的旅行商问题,首先应建立人工蚁群系统的数学模型。

在人工蚁群系统中,蚂蚁具有下面几个特征:以概率大小选择转移路线,概率则是城市之间距离和路径上残留信息量的函数;有记忆功能,在每一次循环中,每只蚂蚁的转移路径只能是它没有走过的;完成一次循环后,在其走过的路径上留下一定量的信息物质;蚂蚁留在路径上的信息量随时间逐渐衰减。

设 $m$ 是蚁群中蚂蚁的数量,用 $d_{ij}(i, j=1, 2, k,$

$n)$ 表示城市 $i$ 和城市 $j$ 之间的距离, $\tau_{ij}(t)$ 表示 $t$ 时刻残留在城市 $i, j$ 连线上的信息量。初始时刻 $t=0$ ,各条路径上有少量的信息 $\tau_{ij}(0)$ 。根据各条路径上信息量的多少, $t$ 时刻位于某一城市的蚂蚁 $k(k=1, 2, k, m)$ 一次只能选择其中一个目标城市, $n$ 次后回到起点。完成一次循环,按下面的公式修改各条路径上的信息量:

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

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{L_k} & \text{本次循环中蚂蚁 } k \text{ 经过连接城市 } i \text{ 和} \\ 0 & \end{cases}$$

$j$ 的路径 (2)

式中 $0 < \rho < 1$ 是表示在时间间隔 $(t, t+n)$ 内信息量衰减的系数。 $L_k$ 是蚂蚁 $k$ 在一次循环中走过路径的总长度, $Q$ 是一个大于0的常数。

综合考虑各转移线路上的信息量及其距离等因素,用 $allowed_k$ 表示蚂蚁 $k$ 在位置 $i$ 上时允许的转移目标位置,定义蚂蚁 $k$ 在 $t$ 时刻由位置 $i$ 转移至位置 $j$ 时的概率为:

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & j \in allowed_k \\ 0 & \end{cases} \quad (3)$$

启发因子 $\eta_{ij} = 1/d_{ij}$ , $\alpha, \beta$ 是两个可以调整的参数,其值决定了信息量和启发因子在蚂蚁选择路径时所起作用的强弱程度。 $P_{ij}(t)$ 是蚂蚁在 $t$ 时刻选择转移路径的依据。

## 3 基于 MATLAB 的改进型基本蚁群算法

基本蚁群算法优点是具有很强的发现较好解的能力,不容易陷入局部最优,与其他一些算法相比能够较快地收敛于解空间的某一子集。利用基本蚁群算法求解 Oliver30 TSP 的典型实验与其他一些优化算法的对比实验结果(表1)可以说明这一点。但是,当种群规模较大时算法复杂度高,而且搜索时间长的问题就显得尤为突出。造成这些问题的主要原因之一是随着解空间的增大,位于点上的蚂蚁个体在选择转移路径时,不同路径上的转移概率相互接近的可能性也越大,而基本蚁群算法在每次进化时,只能固定地选择其中一条路径。若在

每一个点上放置一只蚂蚁即,必然使算法复杂度增加,搜索时间延长。这里我们提出一种对基本蚁群算法进行改进的方法,令即只在任意一个节点上放置一只蚂蚁,并在算法循环过程中,每次随机选取一个节点作为起点,这样可以有效地克服搜索停滞,缩短搜索时间。

表1 不同算法求解 Oliver30 TSP 实验结果对照表[1]

Tab.1 Experimental result comparison using different algorithms to solve Oliver30 TSP[1]

算 法	达到收敛所需循环次数	2-opt	L-K
ant colony system	420		
near neighbor	587	437	421
far insert	428	421	420
near insert	510	492	410
space filling curve	464	431	421
sweep	486	426	420
random	1212	663	421

MATLAB 以矢量和矩阵为基本运算单元,基于 MATLAB 的改进型基本蚁群算法具有编程简单、容易实现等特点。利用基本蚁群算法求解  $n$  个城市的旅行商问题,在 MATLAB 的图形窗口下,可以方便地给出任意数目、任意组合的点阵,其分析结果更具一般性。经过计算找到最短路径后,有 MATLAB 强大绘图功能的支持,将所得路径标注出来,非常直观。基于 MATLAB 的改进型基本蚁群算法程序流程如下:

初始化:输入参数  $n, B, Q, \alpha, \beta, \rho$  及随机数的种子数在给定坐标系中输入  $n$  个城市的 TSP 模型  $[x, y]$

```

x = []; y = [];
for z = 1:n
    [x1, y1, button] = ginput(1);
    if (button == 1) break; end
    plot(x1, y1, '*', 'black');
    x = [x, x1];
    y = [y, y1];
    text(x(z) + 0.5, y(z) + 0.5, num2str(z))
end

```

主程序:按输入顺序编号生成城市序列号矢量  $C$ 、计算距离矩阵  $d$

随机产生初始信息量矩阵

循环计数:

for  $b = 1:B$

随机选择起点位置即  $c(1) = i$

$c(v) (v = 2, 3, \dots, n)$  为允许选择的目标城市

由起点开始计算转移路径:

for  $k = 1:n$

根据各条路径上的现有信息量计算最大

转移概率  $j$  为目标城市序列号:  $c(i+1)$

$= j$

end

$CB(b, :) = C, L$  分别为本次循环路径顺序及路径长度矢量,

$\Delta\tau, \tau(b+1) = \tau(b) + \Delta\tau$  分别为本次循环的信息量增量及新的信息量矩阵

end

计算最佳路径  $L_m = \min[L(b): B \subset B]$ , 获得最短路径

的循环次数 及对应的路径顺序矢量  $cs$

在原图形上绘制最佳路径:

hold on

for  $k = 1:n$

line([x(cs(k)), x(cs(k+1))], [y(cs(k)), y(cs(k+1))]);

end

line([x(cs(n)), x(cs(1))], [y(cs(n)), y(cs(1))]);

hold off;

clear

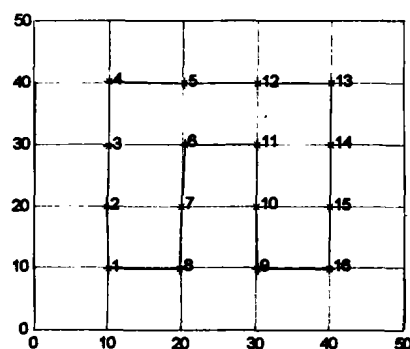


图2 16城市 TSP 问题最好解

Fig.2 The best resolve of TSP with 16 cities

## 4 实验与结论

选用 16 点的等距点阵和 30 点的不规则点阵分

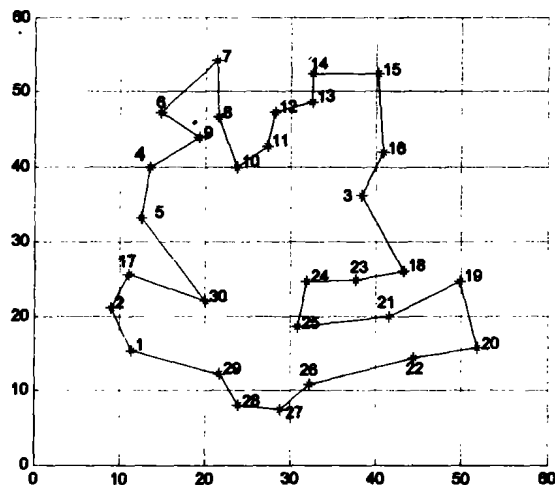


图3 改进型基本蚁群算法找到的最短路径

Fig.3 The shortest route obtained by improved algorithm

别进行仿真。

图2的仿真结果表明:改进型基本蚁群算法具有发现问题最好解的能力。

对于任意给定的30点TSP问题(如图3所

示),当 $\alpha = 1, \beta = 1, \rho = 0.5, Q = 50$ ,取30次仿真的平均结果,找到的最短路径为221.99,达到收敛所需的进化代数为126。实验结果表明:经过对基本蚁群算法的改进,具有较强发现最好解的能力。并且有效地缩短了搜索时间。

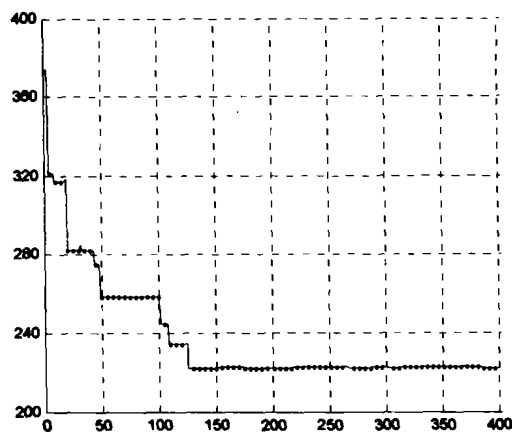


图4 30点TSP最好解进化曲线

Fig.4 Evolution of the best length for 30 TSP

## 参考文献:

- [1] Marco Dorigo Vittorio Maniezzo and Alberto Colomi, Ant System: Optimization by a Colony of Cooperating Agents[C], IEEE Trans on SMC, 1996, 26(1): 28-41.
- [2] 吴庆宏, 张纪会, 徐心和. 具有变异特征的蚁群算法, 计算机研究与发展[J], 1999, 36(10)31-34.
- [3] 张纪会, 高齐圣, 徐心和. 自适应蚁群算法[J]. 控制理论与应用, 2000, 17(1):15-17.

## Improved Basic Ant Colony Algorithm Based on MATLAB

LI Hong, SUN Zhi-yi

(Taiyuan Heavy Machinery Institute, Taiyuan 030024, China)

**Abstract:** Ant colony algorithm is a novel with algorithm simulated evolutionary. It provides a new way to solve complicated combinatorial optimization problems such as traveling salesman problem (TSP), assignment problem, job-shop etc as GA, SA, TS, and so on. Having been enlightened by the behavior of ant colony's searching for food, positive feedback construction and distributed computing combined with certain heuristics are adopted in the algorithm. All that makes it easier to find better solution. In this paper, an improved basic ant colony algorithm is proposed. Using this method the lever of algorithmic complication is depressed and the searching time is reduced. It has strong Ability in finding the best resolution. Based on MATLAB its program is simple and it is convenient to realize.

**Key words:** ant colony algorithm; improvement