蚁群优化算法

Ant Colony Optimization

群智能算法

由简单智能的个体通过某种形式的聚集协同而表现出智能行为。

通过模仿生物界的群体协作行为而提出的仿生类随机搜索算法。

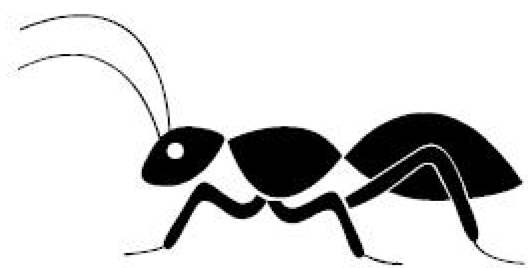
- 人工蜂群算法
- 细菌觅食算法
- 萤火虫算法
- 粒子群算法
- 人工鱼群算法

1. 蚁群算法起源

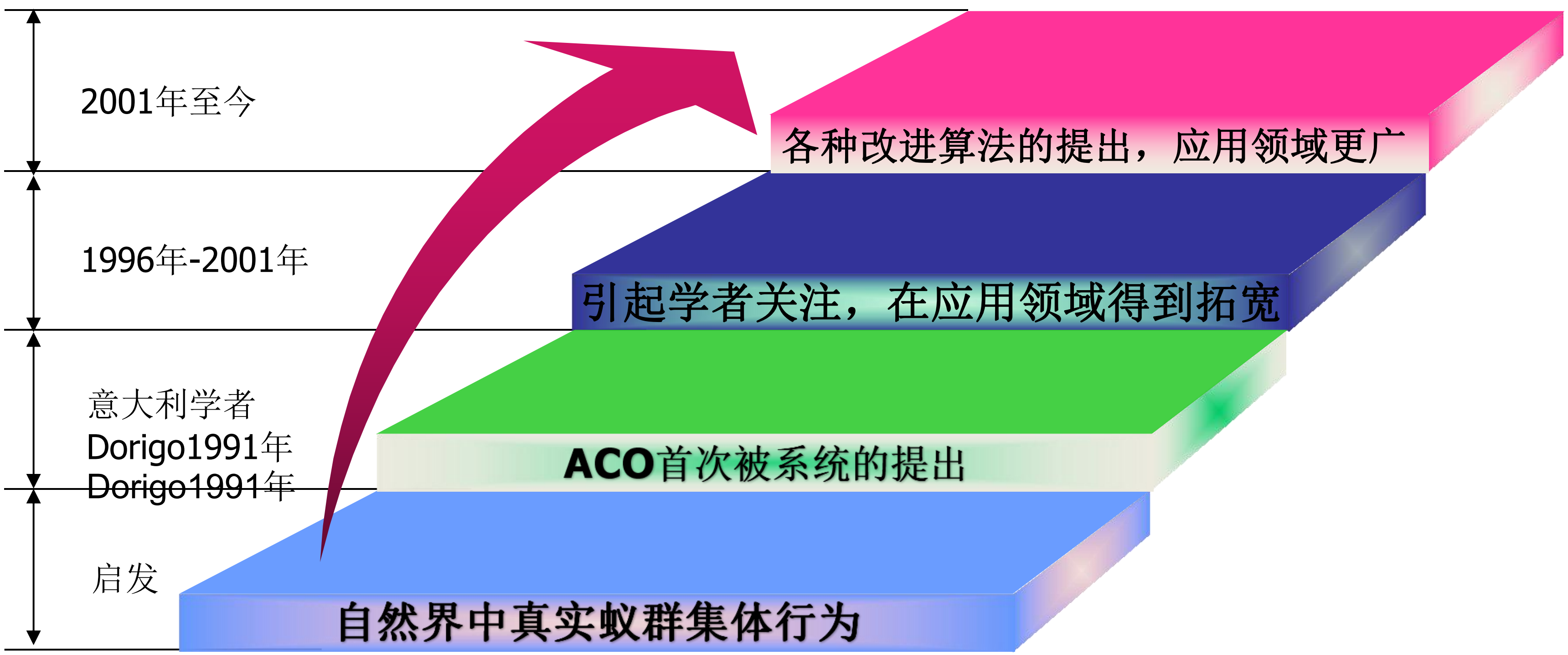
- 计算智能领域有两种基于群智能的算法：蚁群算法和粒子群算法，前者模仿蚂蚁觅食，后者模仿鸟类觅食
- **最早是由意大利学者Colorni A., Dorigo M. 等于1991年提出。经过30年的发展，蚁群算法在理论以及应用研究上已经得到巨大的进步。**

Macro Dorigo

Gambardella



蚁群算法的发展


2001年至今
各种改进算法的提出，应用领域更广
1996年-2001年
引起学者关注，在应用领域得到拓宽
意大利学者
Dorigo1991年
Dorigo1991年
ACO首次被系统的提出
启发
自然界中真实蚁群集体行为

蚁群算法概述

- **蚁群算法(ant colony optimization, ACO)，又称蚂蚁算法，是一种对自然界蚂蚁的觅食行为模拟而得到的一种仿生算法(蚂蚁有能力在没有任何提示的情形下找到从巢穴到食物源的最短路径。**

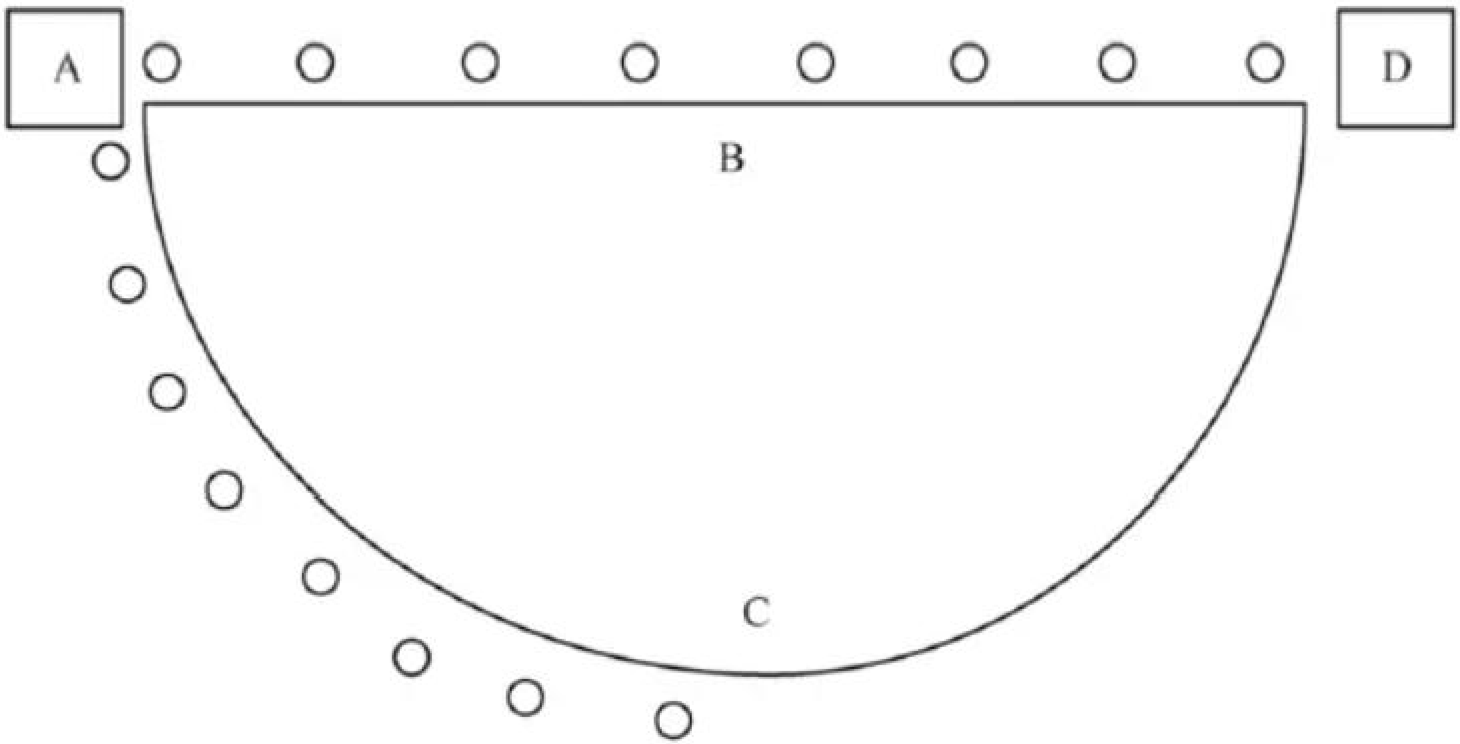
- 当蚂蚁寻找食物，会释放一种挥发性分泌物pheromone(**信息素**)，如果其中一条道路比原来的其他道路更短，**信息素的挥发相对变慢，**该道路上的信息素浓度会越来越大，后来的蚂蚁选择该道路的概率也就越高，最终找到最短路径

蚁群算法原理

- 如何找到最短路径 ？

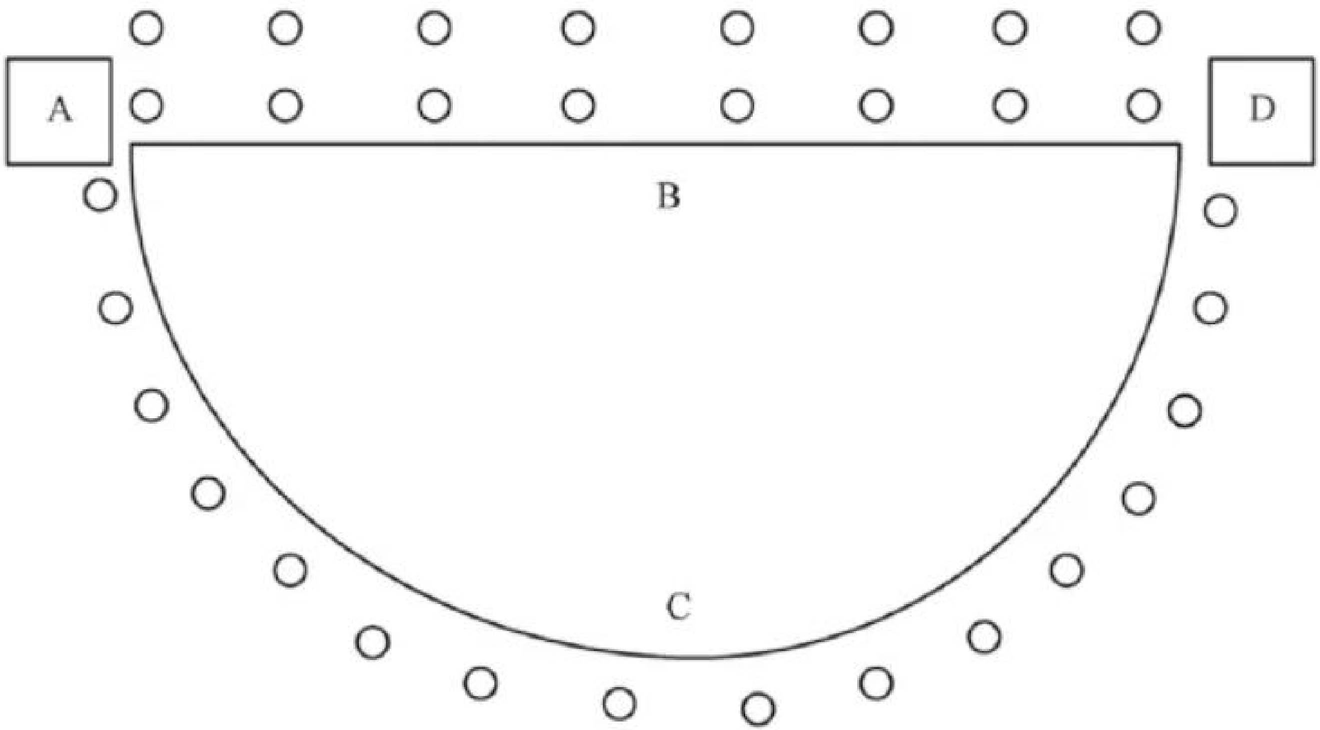
- 信息素：信息素多的地方显然经过这里的蚂蚁多，因而会有更多的蚂蚁聚集过来。
- 正反馈现象：某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大。

- 蚂蚁从A点出发到D点觅食，
- 随机从ABD或ACD中选择一条路。
- 则经过8个时间单位后，如下图所示：
- ABD路线的蚂蚁到达D点，ACD路线的蚂蚁到达C点。



蚂蚁出发后8个时间单位情形.png

- 再过8个时间单位：ABD路线的蚂蚁回到A点，ACD路线的蚂蚁到达D点。



蚂蚁出发后16个时间单位情形.png

- 经过32个单位时间后：ABD路径上的蚂蚁往返两趟，ACD往返一趟，
- 那么ABD与ACD路径上的信息素浓度比值为2：1。

- 根据信息素的引导：

接下来ABD路径就有2只蚂蚁选择，ACD路径仍然为1只，

- 经过32个单位时间后信息素就会达到3：1。

- 因此会有越来越多的蚂蚁选择ABD，ACD逐渐被放弃，这就形成了正反馈。

➢自然蚂蚁的智能特点

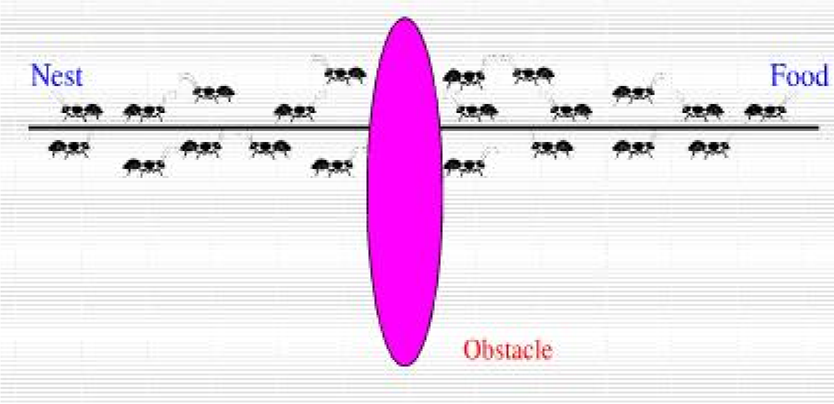


图1 蚂蚁正常行进，突然环境改变，增加了障碍物

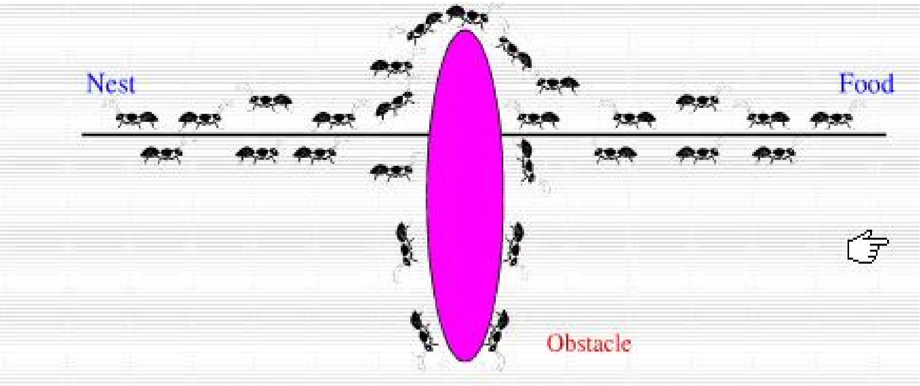


图2 蚂蚁以等同概率选择各条路径
较短路径信息素浓度高，选择该路径的蚂蚁增多

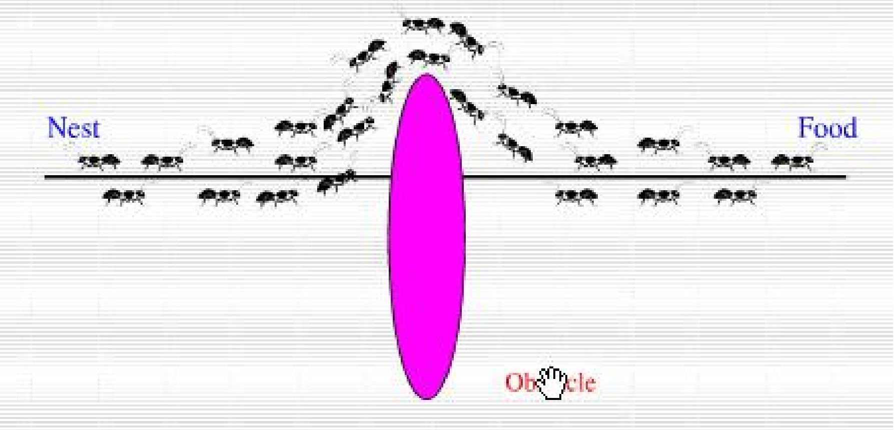


图4 蚂蚁最终绕过障碍物找到最优路径

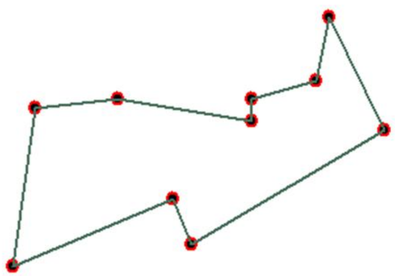
人工蚁群和自然蚁群的区别：

- 人工蚁群有一定的记忆能力，能够记忆已经访问过的节点；
- 人工蚁群选择下一条路径的时候是按一定算法规律有意识地寻找最短路径，而不是盲目的。例如在TSP问题中，可以预先知道当前城市到下一个目的地的距离。

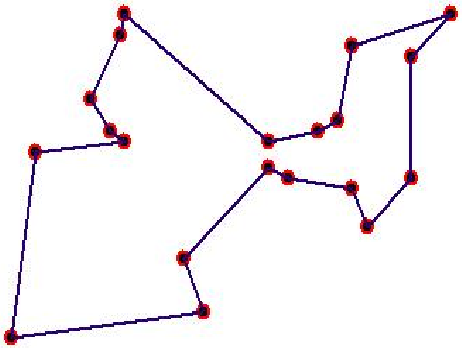
蚁群觅食	蚁群优化算法
蚁群	搜索空间的一组有效解（表现为种群规模N）
觅食空间	问题的搜索空间（表现为维数D）
信息素	信息素浓度变量
蚁巢到食物的一条路径	一个有效解
找到的最短路径	问题的最优解

□ 旅行商问题

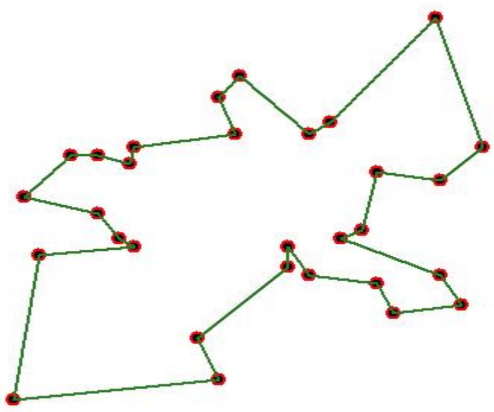
一位商人从自家出发，希望能找到一条最短路径，途径给定集合的所有城市最后返回家乡，并且每个城市都被访问且仅访问一次。



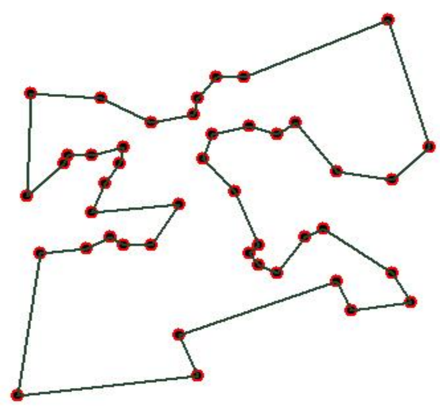
10城市TSP问题



20城市TSP问题



30城市TSP问题



48城市TSP问题

TSP问题数学描述

设 $C=\{c_1,c_2,\ldots,c_n\}$ 是n个城市的集合，

$L=\{l_{ij} \mid c_i,c_j \subset C\}$ 是集合C中元素两两连接的集合，

$d_{ij}(i,j=1,2,\ldots,n)$ 是 l_{ij} 的距离，

目标函数表示为

$\pi_1 \cdots \pi_n$

$d_{\pi_1\pi_2}+d_{\pi_2\pi_3}\cdots+d_{\pi_{n-1}\pi_n}$

$$f(\pi)=\min\left\{\sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)}+d_{\pi(n)\pi(1)}\right\}$$

$(\pi_1 \cdots \pi_n)$

$+d_{\pi_n\pi_1}$

AS算法求解TSP问题有两大步骤：

C_i

$C_i \cdots C_j$

C_{j+1} C_j

路径构建与信息素更新方式。

路径构建

每个蚂蚁都随机选择一个城市作为其出发城市，并维护一个**路径记忆**向量，用来存放该蚂蚁依次经过的城市。

蚂蚁在构建路径的每一步中，按照一个随机比例规则选 择下一个要到达的城市。

•随机比例规则

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\eta_{ij}(t)\right]^{\beta}}{\sum\limits_{k\in allowed_k}\left[\tau_{ik}(t)\right]^{\alpha}\times\left[\eta_{ik}(t)\right]^{\beta}} & if \quad j\in allowed_k\\ 0 & others\end{cases}$$

- i、j分别为起点和终点；
- $\eta_{ij}=1/d_{ij}$ 为能见度，是两点i、j路距离的倒数；
- $\tau_{ij}(t)$ 为时间 t 时由 i 到 j 的信息素强度；
- $allowed_k$ 为尚未访问过的节点集合；
- α,β 为两常数，分别是信息素和能见度的加权值。

信息素更新

初始化信息素浓度 $\boxed{\tau_{ij} = C, \qquad \forall i, j}$

如果C太小，算法容易早熟，蚂蚁会很快的全部集中到一条局部最优的路径上。反之，如果C太大，信息素对搜索方向的指导作用太低，也会影响算法性能。

$$\boxed{\tau_{ij}(t) = (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}}$$

$$= \frac{Q}{d_{ij}}$$

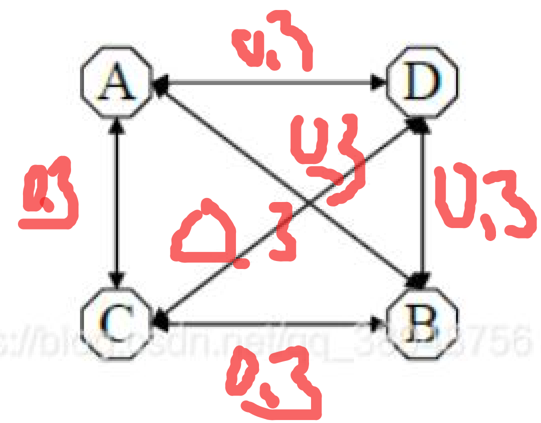
$$\frac{1}{d_{ij}}$$

m为蚂蚁个数，$0 < \rho <= 1$为信息素的蒸发率，在AS中通常设置为0.5，$\Delta\tau_{ij}^{k}$为第k只蚂蚁在路径i到j所留下來的信息素

算　例1

四个城市的TSP问题，距离矩阵和城市图示如下：

$$D=(d_{ij})=\begin{pmatrix} 0 & 3 & 1 & 2 \\ 3 & 0 & 5 & 4 \\ 1 & 5 & 0 & 2 \\ 2 & 4 & 2 & 0 \end{pmatrix}$$



假设共m=3只蚂蚁，参数 α =1， β =2， ρ =0.5

步骤1 初始化

首先使用贪婪算法得到路径的(ACDBA), 则 C_{nn}=1+2+4+3=10，求得 $\tau_0=m/C_{nn}=0.3$

$$\tau(0)=(\tau_{ij}(0))=\begin{pmatrix} 0 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0 \end{pmatrix}$$

步骤2 为每个蚂蚁随机选择出发城市，假设蚂蚁1选择城市A，蚂蚁2选择城市B，蚂蚁3选择城市D。禁忌表

步骤3.1 为每个蚂蚁选择下一访问城市，仅以蚂蚁1为例

当前城市i=A，可访问城市集合$J_1(i)=\{B,C,D\}$

计算蚂蚁1访问各个城市的概率

$$A \Rightarrow \begin{cases} B: \tau_{AB}^{\alpha} \times \eta_{AB}^{\beta} = 0.3^1 + (1/3)^2 = 0.033 \\ C: \tau_{AC}^{\alpha} \times \eta_{AC}^{\beta} = 0.3^1 + (1/1)^2 = 0.300 \\ D: \tau_{AD}^{\alpha} \times \eta_{AD}^{\beta} = 0.3^1 + (1/2)^2 = 0.075 \end{cases}$$

$$p(B) = 0.033/(0.033+0.3+0.075) = 0.081$$
$$p(C) = 0.3/(0.033+0.3+0.075) = 0.74$$
$$p(D) = 0.075/(0.033+0.3+0.075) = 0.18$$

用轮盘赌法选择下一个访问城市，假设产生的随机数q=0.05，则蚂蚁1会选择城市B

同样，假设蚂蚁2选择城市D，蚂蚁3选择城市A。

步骤3.2 为每个蚂蚁选择下一访问城市，仅以蚂蚁1为例

当前城市i=B，路径记忆向量R^1=(AB),可访问城市集合$J_1(i)$={C,D}

计算蚂蚁1访问C,D城市的概率:

$$B \Rightarrow \begin{cases} C: \tau_{BC}^{\alpha} \times \eta_{BC}^{\beta} = 0.3^1 + (1/5)^2 = 0.012 \\ D: \tau_{BD}^{\alpha} \times \eta_{BD}^{\beta} = 0.3^1 + (1/4)^2 = 0.019 \end{cases}$$

$$p(C) = 0.012/(0.012 + 0.019) = 0.39$$

$$p(D) = 0.019/(0.012 + 0.019) = 0.61$$

用轮盘赌法选择下一个访问城市。假设产生的随机数q=0.67，则蚂蚁1会选择城市D

同样，假设蚂蚁2选择城市C，蚂蚁3选择城市D。

此时，所以蚂蚁的路径都已经构造完毕

蚂蚁1：A→B →D →C →A

蚂蚁2：B→D →C →A →B

蚂蚁3：D→A →C →B →D 12

步骤4 信息素更新

计算每只蚂蚁构建的路径长度：C1=3+4+2+1=10；C2=4+2+1+3=10；C3=2+1+5+4=12。更新每条边上的信息素

$$\tau_{AB} = (1-\rho)\times\tau_{AB} + \sum_{k=1}^{3}\Delta\tau_{AB}^{k} = 0.5\times 0.3 + (1/10 + 1/10) = 0.35$$

$$\tau_{AC} = (1-\rho)\times\tau_{AC} + \sum_{k=1}^{3}\Delta\tau_{AC}^{k} = 0.5\times 0.3 + (1/12) = 0.16$$

步骤5

如果满足结束条件，则输出全局最优结果并结束程序，否则，则转向步骤2继续执行。

蚂蚁系统数学模型（一）

设n表示TSP规模，

i和j是集合C中的两个元素，

m为蚁群蚂蚁总数，

$b_i(t)$ **表示t时刻位于i的蚂蚁数目，则** $m = \sum_{i=1}^{n} b_i(t)$

设 $\tau_{ij}(t)$**为t时刻路径(i,j)上的信息素量，**

$\Gamma = \{\tau_{ij}(t) \mid c_i, c_j \subset C\}$ **是t时刻集合C中所有信息素的集合。**

初始时刻，各条路径上的信息量是相同的。

蚂蚁系统数学模型（二）

蚂蚁$k(k=1,2,...,m)$ 在运动过程中有三个因素决定

其转移方向信息素量$\tau_{ij}(t)$，启发式信息$\eta_{ij}(t)$ 和禁忌表 $tabu_k$

$\eta_{ij}(t)$为启发函数，其表达式一般表示为 $\eta_{ij}(t)=\dfrac{1}{d_{ij}}$ ；

禁忌表 $tabu_k$ 用于记录蚂蚁k当前走过的城市，

$allowed_k=\{C-tabu_k\}$ 表示蚂蚁k下步允许选择的城市。

蚂蚁系统数学模型（三）

$p_{ij}^{k}(t)$ **表示蚂蚁k在t时刻由i转到j的概率**

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s_k\subset allowed_k}[\tau_{is}(t)]^{\alpha}\cdot[\eta_{is}(t)]^{\beta}}, & if\ j\notin allowed_k\\ 0, & otherwise\end{cases}$$

上式中，α为信息素因子， β为启发式因子，用于控制信息素浓度和启发式信息作用的权重关系。值越大表示重要性越大

蚂蚁系统数学模型（四）

信息素更新公式

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij};\qquad \Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

1. 原有信息素的挥发 通常的做法是设置信息持久率 $\rho(0<\rho\leq 1)$
让所有 $\tau_{ij}(t)$ **乘以** ρ**。**

2. 新生信息素的释放 AS算法曾有过三种信息素释放策略

Ant-Density模型： $\Delta\tau_{ij}^{k}=Q$ **若蚂蚁k在t到t+1之间经过(i,j)**

Ant-Quantity模型： $\Delta\tau_{ij}^{k}=\dfrac{Q}{d_{ij}}$ **若蚂蚁k在t到t+1之间经过(i,j)**

Ant-Cycle模型： $\Delta\tau_{ij}^{k}=\dfrac{Q}{L_k}$ **若蚂蚁k在本次循环中经过(i,j)**

（二）参数含义及符号

m ——蚂蚁数量；

k ——蚂蚁编号；

t ——时刻；

n ——城市数；

d_{ij} ——城市 (i, j) 之间的距离；

η_{ij} ——启发式因子（能见度），反映蚂蚁由城市 i 转移到城市 j 的启发程度；

τ_{ij} ——边 (i, j) 上的信息素量；

$\Delta\tau_{ij}$——本次迭代边(i, j)上的信息素增量；

$\Delta\tau_{ij}^{k}$——第k只蚂蚁在本次迭代中留在边(i, j)上的信息素量；

ρ——信息素蒸发（或挥发）系数，

$1-\rho$——持久性（或残留）系数，$0<\rho<1$；

$P_{ij}^{k}(t)$——时刻t蚂蚁k由城市i转移到城市j的概率（转移概率）；

$tabu_k$——蚂蚁k的禁忌表。

（三）计算公式

1、转移概率 $p_{ij}^{k}(t)$ 计算公式：

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in J_k(i)}[\tau_{is}(t)]^{\alpha}\cdot[\eta_{is}(t)]^{\beta}}, & \text{如果} j\in J_k(i)\\ 0, & \text{否则}\end{cases}$$

α ——信息素的相对重要程度；

β ——启发式因子的相对重要程度；

$J_k(i)$——蚂蚁 k下一步允许选择的城市集合。

2、启发式因子计算公式：$\eta_{ij}=\dfrac{1}{d_{ij}}$

3、信息素计算公式

当所有蚂蚁完成1次周游后，各路径上的信息素为：

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

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁}k\text{在本次周游中经过边}\ (i,j)\\ 0, & \text{否则}\end{cases}$$

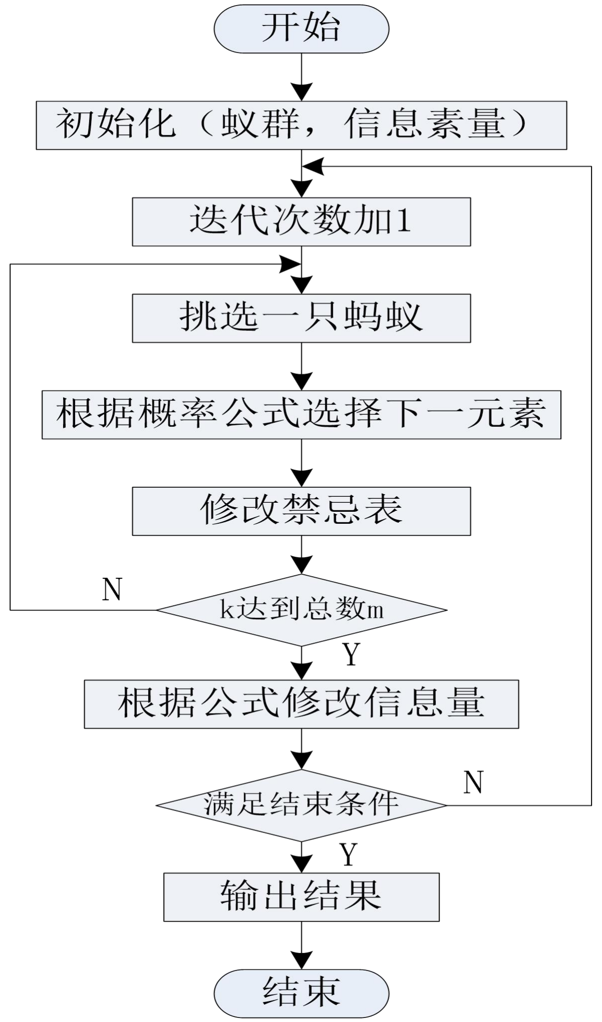
Q ——正常数，

L_k ——蚂蚁 k在本次周游中所走路径的长度。

开始时，令 $\tau_{ij}(0)=C$

```
(1)初始化  随机放置蚂蚁，为每只蚂蚁建立禁忌表，
(2)迭代过程
    k=1
    while k=<Count do (执行迭代)
        for i = 1 to m do  (对m只蚂蚁循环)
          for j = 1 to n - 1 do  (对n个城市循环)
            根据蚂蚁行动原则 选择下一个城市j并将j置入禁忌表,
        end for
      end for
      计算每只蚂蚁经过的路径长度
      依据信息素更新方法更新所有路径上的信息量;
      k = k + 1;
    end while
(3)输出结果,结束算法.
```

算法流程



开始
初始化（蚁群，信息素量）
迭代次数加1
挑选一只蚂蚁
根据概率公式选择下一元素
修改禁忌表
N
k达到总数m
Y
根据公式修改信息量
满足结束条件
N
Y
输出结果
结束

蚁群算法求解TSP问题

➢ 下面以TSP为例说明基本蚁群算法模型。

◆首先将m只蚂蚁随机放置在n个城市，位于城市i的第k只蚂蚁选择下一个城市j的概率为：

$$P^k(i,j)=\begin{cases}\dfrac{[\tau(i,j)]^{\alpha}\cdot[\eta(i,j)]^{\beta}}{\sum\limits_{s\notin tabu_k}[\tau(i,s)]^{\alpha}\cdot[\eta(i,s)]^{\beta}}, & if\ \ j\notin tabu_k \\ 0, & otherwise\end{cases} \tag{1}$$

◆ $\tau(i,j)$ 表示边（i，j）上的信息素浓度；$\eta(i,j)=1/d(i,j)$ 是启发信息，d是城市i和j之间的距离；α 和 β 反映了信息素与启发信息的相对重要性；

$tabu_k$ 表示蚂蚁k已经访问过的城市列表。

◆当所有蚂蚁完成周游后，按以下公式进行信息素更新。

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

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (2)$$

◆其中，ρ为小于1的常数，表示信息的持久性。

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & ij \in l_k \\ 0 & otherwise \end{cases} \qquad (3)$$

◆其中，Q为常数；L_k表示第k只蚂蚁在本次迭代中走过的路径长度。

实现过程

Step 1 初始化

置 t：=0；{t 表示时间}

置 NC：=0；{NC 为迭代次数}

对每条边 l_{ij} 设置 $\tau_{ij}(t)=C$，$\Delta\tau_{ij}(t)=0$；将 m 只蚂蚁随机放到 n 个城市上；

Step 2 置 s：=1；{s 为禁忌表中的索引}

for k：=1 **to** m **do**

将蚂蚁 k 的起点城市加入到禁忌表 $tabu_k$；

end for

Step 3 while (禁忌表 $tabu_k$ 不满)

置 s：$=s+1$；

for k：$=1$ **to** m **do**

按式（2.1）计算转移概率 $p_{ij}^k(t)$，根据赌轮方法选择下一个要到的

城市 j；{在时刻 t 时，蚂蚁 k 在城市 $i = tabu_k(s-1)$}

蚂蚁 k 移到城市 j；

将城市 j 加入到 $tabu_k$；

end for

end while

Step 4　for k：=1 to m do

蚂蚁 k 从 $tabu_k(n)$ 移到 $tabu_k(1)$；

计算蚂蚁 k 走过的周游长度 L_k；

更新当前的最优路径

end for

for 每条边 l_{ij}

for k：=1 to m do

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & \text{若蚂蚁}k\text{在本次周游中经过边}l_{ij}\\ 0, & \text{否则}\end{cases}$$

$$\Delta\tau_{ij}=\Delta\tau_{ij}+\Delta\tau_{ij}^{k};$$

end for

end for

Step 5 **for** 每条边 l_{ij} 按式（2.2）计算 $\tau_{ij}(t+1)$；

置 t：$=t+1$；

置 NC：$=NC+1$；

for 每条边 l_{ij}， 置 $\Delta\tau_{ij}(t)=0$

Step 6 **if** $(NC < NC_{MAX})$ **and** （没有出现停滞情况） **then**

清空所有的禁忌表；

goto step 2

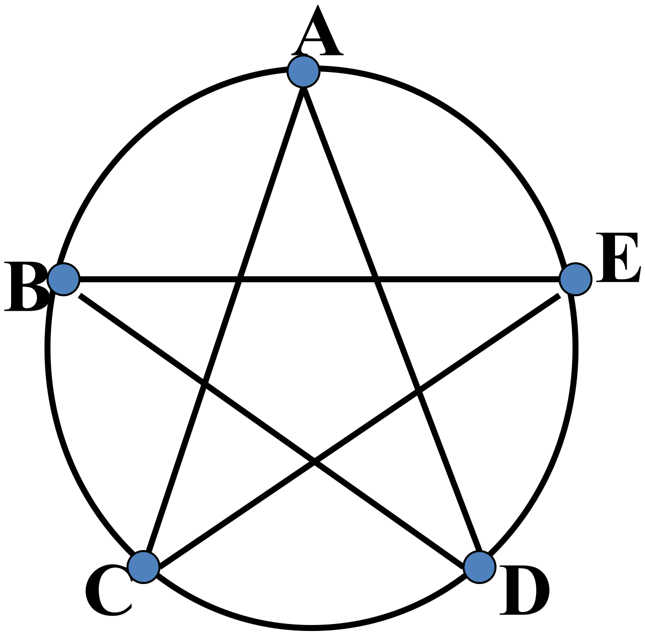
else

打印最优路径；

算法停止；

end

算　例2



已知资料表

	A	B	C	D	E
A	0	2	10	8	3
B	1	0	2	5	7
C	9	1	0	3	6
D	10	4	3	0	2
E	2	7	5	1	0

参数设置：$m=5,\quad \alpha=1,\quad \beta=1,\quad \rho=0.5,\quad Q=100,\quad \tau_{ij}(0)=2$

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}}{\sum\limits_{s\in J_k(i)}\left[\tau_{is}(t)\right]^{\alpha}\cdot\left[\eta_{is}(t)\right]^{\beta}}=\dfrac{X}{Y}, & \text{如果} j\in J_k(i)\\ 0, & \text{否则}\end{cases}$$

$$\eta_{ij}=\frac{1}{d_{ij}}$$

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
1	0	A	A	B C D E	2 2 2 2	0.47 0.095 0.118 0.315	11	9.1	2.117
		B	A,B	C D E	2 2 2	0.593 0.237 0.169			1.686
		C	A,B,C	D E	2 2	0.67 0.33			1.0
		D	A,B,C,D	E	2	1.0			1.0
		E	A,B,C,D,E	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
2	0	B	B	A C D E	2 2 2 2	0.54 0.27 0.11 0.08	9	11.1	3.686
		A	B,A	C D E	2 2 2	0.18 0.22 0.60			1.117
		E	B,A,E	C D	2 2	0.17 0.83			2.4
		D	B,A,E,D	C	2	1.0			0.667
		C	B,A,E,D,C	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
3	0	C	C	A B D E	2 2 2 2	0.069 0.62 0.207 0.103	9	11.1	3.222
		B	C,B	A D E	2 2 2	0.745 0.149 0.106			2.686
		A	C,B,A	D E	2 2	0.273 0.727			0.917
		E	C,B,A,E	D	2	1.0			2.0
		D	C,B,A,E,D	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
4	0	D	D	A B C E	2 2 2 2	0.084 0.211 0.287 0.422	11	9.1	2.367
		E	D,E	A B C	2 2 2	0.593 0.169 0.237			1.686
		A	D,E,A	B C	2 2	0.83 0.17			1.2
		B	D,E,A,B	C	2	1.0			1.0
		C	D,E,A,B, C	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
5	0	E	E	A B C D	2 2 2 2	0.271 0.078 0.109 0.543	9	11.1	3.686
		D	E,D	A B C	2 2 2	0.146 0.366 0.488			1.367
		C	E,D,C	A B	2 2	0.1 0.9			2.222
		B	E,D,C,B	A	2	1.0			2.0
		A	E,D,C,B,A	空集	-	-			

信息素矩阵 $\tau_{ij}(0+5)$

	A	B	C	D	E
A	0	9.1+9.1+1 =19.2	1	1	11.1+11.1+11.1 +1 =34.3
B	11.1+11.1+11.1 +1=34.3	0	9.1+9.1+1 =19.2	1	1
C	1	11.1+11.1+ 11.1+1 =34.3	0	9.1+9.1+1 =19.2	1
D	1	1	11.1+11.1+ 11.1+1 =34.3	0	9.1+9.1+1 =19.2
E	9.1+9.1+1 =19.2	1	1	11.1+11.1+ 11.1+1 =34.3	0

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^{k}(t)$	L_k	$\Delta\tau_{ij}^{k}$	Y
1	5	A	A	B C D E	19.2 1 1 34.3	0.45 0.005 0.006 0.538	9	11.1	21.258
		E	A,E	B C D	1 1 34.3	0.004 0.006 0.99			34.643
		D	A,E,D	B C	1 34.3	0.021 0.979			11.683
		C	A,E,D,C	B	34.3	1.0			34.3
		B	A,E,D,C,B	空集	-	-			

<table>
<tr><th>k</th><th>t</th><th>i</th><th>$tabu_k$</th><th>$J_k(i)$</th><th>$\tau_{ij}(t)$</th><th>$p_{ij}^k(t)$</th><th>L_k</th><th>$\Delta\tau_{ij}^k$</th><th>Y</th></tr>
<tr><td rowspan="5">2</td><td rowspan="5">5</td><td>B</td><td>B</td><td>A
C
D
E</td><td>34.3
19.2
1
1</td><td>0.775
0.217
0.005
0.003</td><td rowspan="5">9</td><td rowspan="5">11.1</td><td>44.24</td></tr>
<tr><td>A</td><td>B,A</td><td>C
D
E</td><td>1
1
34.3</td><td>0.009
0.011
0.98</td><td>11.66</td></tr>
<tr><td>E</td><td>B,A,E</td><td>C
D</td><td>1
34.3</td><td>0.006
0.994</td><td>34.5</td></tr>
<tr><td>D</td><td>B,A,E,D</td><td>C</td><td>34.3</td><td>1.0</td><td>11.43</td></tr>
<tr><td>C</td><td>B,A,E,D,
C</td><td>空集</td><td>-</td><td>-</td><td></td></tr>
</table>

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
3	5	C	C	A B D E	1 34.3 19.2 1	0.003 0.837 0.156 0.004	9	11.1	40.98
		B	C,B	A D E	34.3 1 1	0.99 0.006 0.004			34.64
		A	C,B,A	D E	1 34.3	0.011 0.989			11.56
		E	C,B,A,E	D	34.3	1.0			34.3
		D	C,B,A,E,D	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
4	5	D	D	A B C E	1 1 34.3 19.2	0.005 0.012 0.535 0.449	9	11.1	21.38
		C	D,C	A B E	1 34.3 1	0.003 0.992 0.005			34.58
		B	D,C,B	A E	34.3 1	0.996 0.004			34.44
		A	D,C,B,A	E	34.3	1.0			11.43
		E	D,C,B,A,E	空集	-	-			

k	t	i	$tabu_k$	$J_k(i)$	$\tau_{ij}(t)$	$p_{ij}^k(t)$	L_k	$\Delta\tau_{ij}^k$	Y
5	5	E	E	A B C D	19.2 1 1 34.3	0.217 0.003 0.005 0.775	9	11.1	44.24
		D	E,D	A B C	1 1 34.3	0.008 0.021 0.971			11.78
		C	E,D,C	A B	1 34.3	0.003 0.997			34.41
		B	E,D,C,B	A	34.3	1.0			34.3
		A	E,D,C,B,A	空集	-	-			

至此出现了停滞现象，算法结束。

已找到最优解：**AEDCBA,**目标函数值为**9**。

蚁群优化算法参数设置

参数	参数意义	参数经验值
蚂蚁数目m	影响算法搜索能力与计算量， 数目多，计算量大，收敛慢 数目少，探索能力降低，早熟	AS,EAS,MMAS m=n ACS，m=10
信息素权重 α 启发信息权重 β	决定算法的搜索导向 α 越小，偏向于眼前利益 β 越小，偏向于信息素浓度	各类ACO算法 $\alpha=1$ $\beta=2\sim5$
信息素维持因子ρ	影响蚂蚁个体间的相互影响强弱 ρ 大，较高全局搜索，收敛慢 ρ 小，信息素挥发快，易早熟	AS,EAS $\rho=0.5$ MMAS $\rho=0.98$ ACS $\rho=0.9$
初始信息素量τ_0	决定初始阶段探索能力	ACS $\tau_0=1/(n\cdot L_{nn})$

四、改进的蚁群优化算法

改进的蚂蚁算法

▲最优解保留策略蚂蚁系统（带精英策略的蚂蚁系统ASelite）

▲最大-最小蚂蚁系统（**MMAS**）

▲基于优化排序的蚂蚁系统（**ASrank**）

▲最优最差蚂蚁系统（**BWAS**）

▲一种新的自适应蚁群算法（**AACA**）

▲基于混合行为的蚁群算法（**HBACA**）

（一）带精英策略的蚂蚁系统 AS_{elite}

特点——在信息素更新时给予当前最优解以额外的信息素量，使最优解得到更好的利用。找到全局最优解的蚂蚁称为“精英蚂蚁”。

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

$$\Delta\tau_{ij}^{*}=\begin{cases}\sigma\cdot\dfrac{Q}{L^{gb}}, & \text{若边 } ij \text{ 是当前最优解的一部分}\\ 0, & \text{否则}\end{cases}$$

$\Delta\tau_{ij}^{*}$ ——精英蚂蚁在边 ij 上增加的信息素量；

σ ——精英蚂蚁个数；

L^{gb} ——当前全局最优解路径长度。

（二）最大最小蚂蚁系统 MMAS

特点
1、每次迭代后，只对最优解所属路径上的信息素更新。
2、对每条边的信息素量限制在范围 $[\tau_{\min}, \tau_{\max}]$ 内，目的是防止某一条路径上的信息素量远大于其余路径，避免过早收敛于局部最优解。

关于 $\tau_{\min}, \tau_{\max}$ 的取值，没有确定的方法，有的书例子中取为0.01，10；有的书提出一个在最大值给定的情况下计算最小值的公式。

（三）基于优化排序的蚂蚁系统 AS_{rank}

特点：每次迭代完成后，蚂蚁所经路径由小到大排序，并根据路径长度赋予不同的权重，路径越短权重越大。信息素更新时对 $\Delta\tau_{ij}^{k}$ 考虑权重的影响。

（四）最优最差蚂蚁系统 BWAS

特点：主要是修改了ACS中的全局更新公式，增加对最差蚂蚁路径信息素的更新，对最差解进行削弱，使信息素差异进一步增大。

（五）一种新的自适应蚁群算法 AACA

特点：将ACS中的状态转移规则改为自适应伪随机比率规则，动态调整转移概率，以避免出现停滞现象。

说明：在ACS的状态转移公式中，q_0是给定的常数；在AACA中，q_0是随平均节点分支数ANB而变化的变量。ANB较大，意味着下一步可选的城市较多，q_0也变大，表示选择信息素和距离最好的边的可能性增大；反之减小。

（六）基于混合行为的蚁群算法 HBACA

特点：按蚂蚁的行为特征将蚂蚁分成4类，称为4个子蚁群，各子蚁群按各自的转移规则行动，搜索路径，每迭代一次，更新当前最优解，按最优路径长度更新各条边上的信息素，如此直至算法结束。

蚂蚁行为——蚂蚁在前进过程中，用以决定其下一步移动到哪个状态的规则集合。

蚂蚁行为

1、蚂蚁以随机方式选择下一步要到达的状态。

2、蚂蚁以贪婪方式选择下一步要到达的状态。

3、蚂蚁按信息素强度选择下一步要到达的状态。

4、蚂蚁按信息素强度和城市间距离选择下一步要到达的状态 。

五、蚁群算法与遗传的比较

实验结果表明：

1、蚁群算法所找出的解的质量最高，遗传算法次之。

2、蚁群算法的收敛速度快。蚁群算法之所以能够快速收敛到全局最优解，是因为该算法的个体之间不断进行信息交流和传递。单个个体容易收敛于局部最优，多个个体通过合作可以很快地收敛于解空间的最优解的附近。

（六）AS算法的优点与不足

优点

- 较强的鲁棒性——稍加修改即可应用于其他问题。（鲁棒性就是系统的健壮性，用以表征控制系统对特性或参数摄动的不敏感性。）
- 分布式计算——本质上具有并行性。
- 易于与其他启发式算法结合。

不足

- 一般需要较长的搜索时间。
- 容易出现停滞现象。