智能算法

第七章：蚁群算法

谷俊峰

工业装备结构分析国家重点实验室

工程力学系

运载工程与力学学部

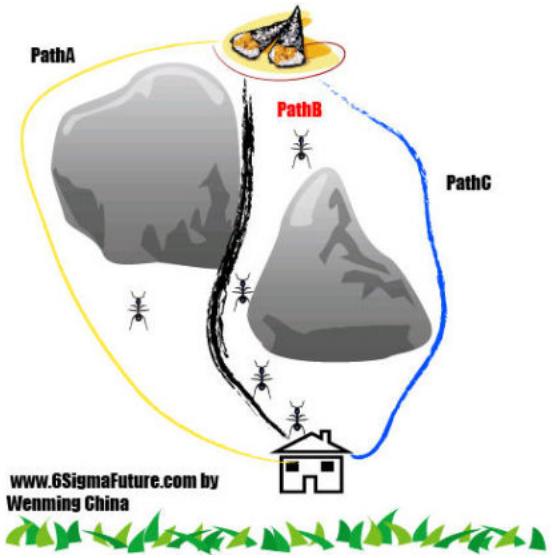
提纲

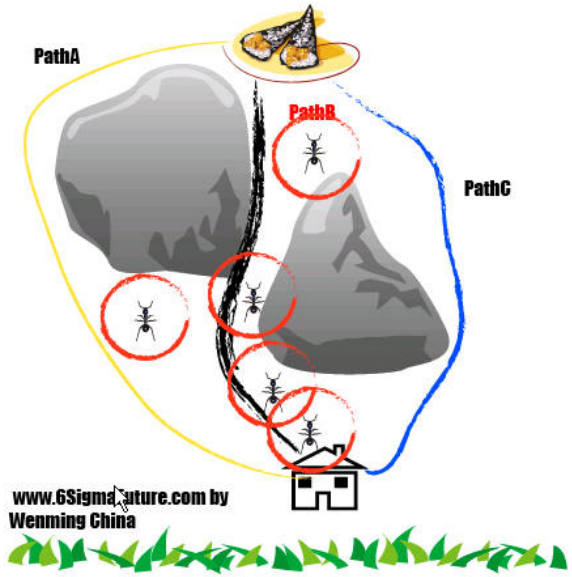
7.1 蚁群优化算法起源

- 蚂蚁属于群居昆虫
  - 单个蚂蚁的行为极其简单,但由这样的单个简单的个体所组成的蚁群群体却表现出极其复杂的行为,能够完成复杂的任务。

7.1 蚁群优化算法起源

- 蚂蚁觅食
  - 蚂蚁没有发育完全的视觉感知系统，甚至很多种类完全没有视觉，他们在寻找食物的过程中是如何选择路径的呢？
  - 蚂蚁往往像军队般有纪律、有秩序的搬运食物，他们是通过什么方式进行群体间的交流协作呢？
  - 大部分的蚂蚁都是按照途中的最近的路线走，小小的蚂蚁是如何有这么高的智能的呢？





7.1 蚁群优化算法起源

20世纪50年代中期创立了仿生学，人们从生物进化的机理中受到启发。提出了许多用以解决复杂优化问题的新方法，如进化规划、进化策略、遗传算法等，这些算法成功地解决了一些实际问题。

- 1991年 意大利米兰理学院 M. Dorigo 提出Ant System, 用于求解TSP等组合优化问题。
- 1995年 Gramdardella和Dorigo提出Ant-Q算法，建立了AS和Q-learning的联系。
- 1996年 二人又提出Ant Colony System
- 1997年 有人提出Max-Min Ant System
- 1999年 Dorigo等人把先前各种算法归结为Ant Colony Optimization meta-heuristic的统一框架下，给出抽象而规范的算法描述.
- 目前，被较广泛的应用

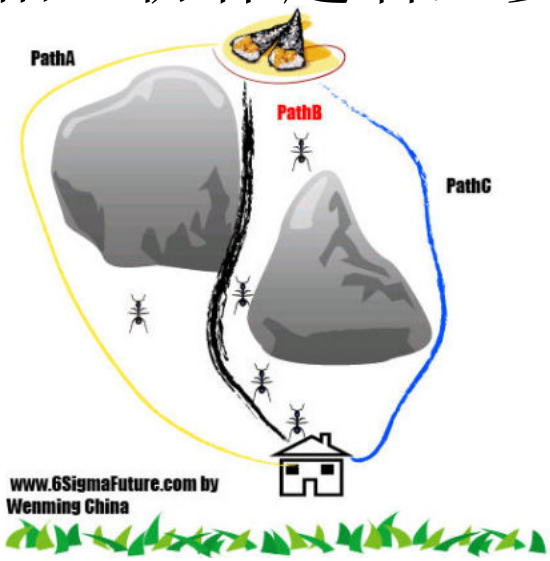
Dorigo

7.1 蚁群优化算法应用领域

- 蚁群优化算法自1991年由Dorigo提出并应用于TSP问题以来，已经发展了近20年。
- 具有鲁棒性强、全局搜索、并行分布式计算、易与其他问题结合等优点
- 应用领域不断扩张，如车间调度问题、车辆路径问题、分配问题、子集问题、网络路由问题、蛋白质折叠问题、数据挖掘、图像识别、系统辨识等。
- 这些问题大都是NP难的组合优化问题，用传统算法难以求解或者无法求解，各种蚁群算法及其改进版本的出现，为这些难题提供了有效而高效的手段。

7.1 蚁群算法原理

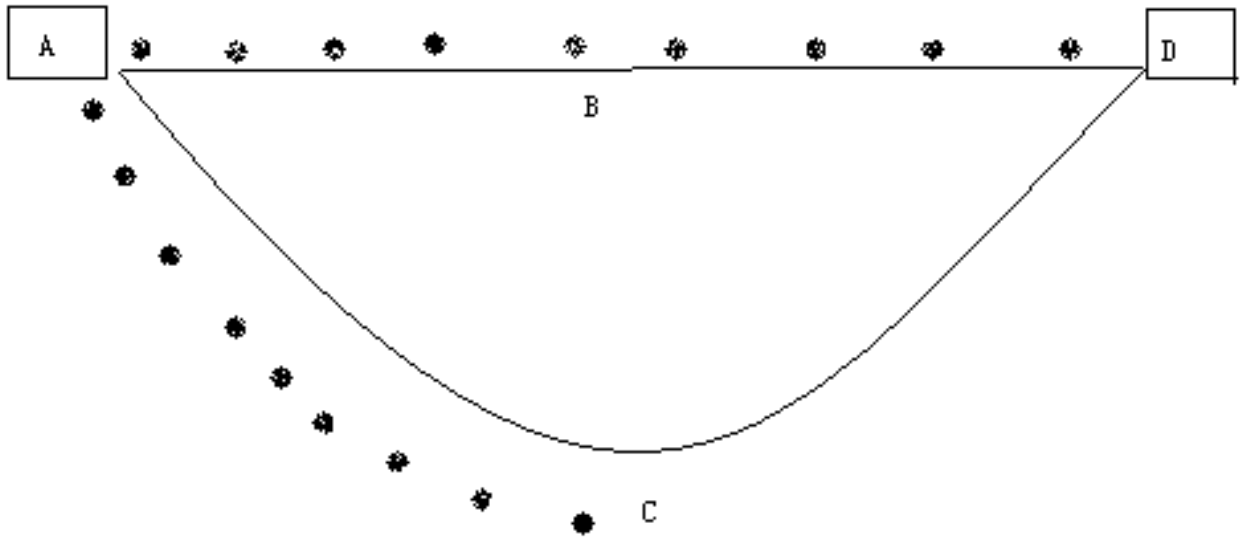
- 蚁群是如何完成这些复杂的任务的呢? 人们经过大量研究发现,蚂蚁个体之间是通过一种称之为外激素(pheromone) 的物质进行信息传递. 从而能相互协作,完成复杂的任务.
- 蚁群之所以表现出复杂有序的行为,个体之间的信息交流与相互协作起着重要的作用.



7.1 蚁群算法原理

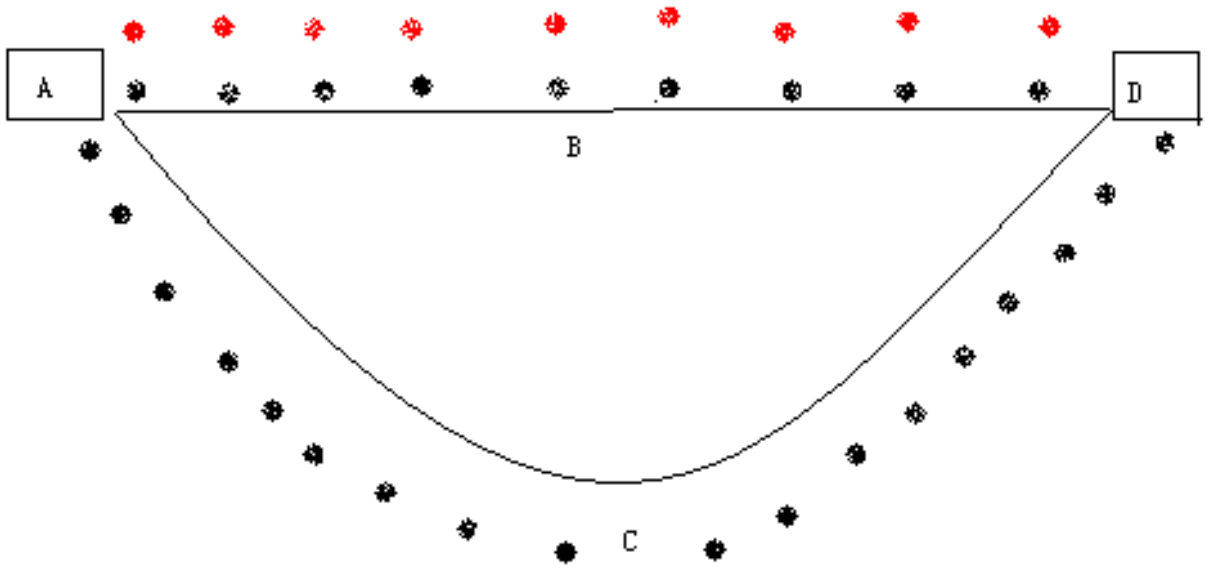
- 蚂蚁在运动过程中，能够在它所经过的路径上留下外激素，而且蚂蚁在运动过程中能够感知外激素的存在及其强度，并以此指导自己的运动方向，蚂蚁倾向于朝着外激素强度高的方向移动.
- 由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象：某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大. 蚂蚁个体之间就是通过这种信息的交流达到搜索食物的目的。

7.1 简化的蚂蚁寻食过程



蚂蚁从A点出发，速度相同，食物在D点，可能随机选择路线ABD或ACD。假设初始时每条分配路线一只蚂蚁，每个时间单位行走一步，本图为经过9个时间单位时的情形：走ABD的蚂蚁到达终点，而走ACD的蚂蚁刚好走到C点，为一半路程。

7.1 简化的蚂蚁寻食过程



本图为从开始算起，经过**18**个时间单位时的情形：走**ABD**的蚂蚁到达终点后得到食物又返回了起点**A**，而走**ACD**的蚂蚁刚好走到**D**点。

7.1 简化的蚂蚁寻食过程

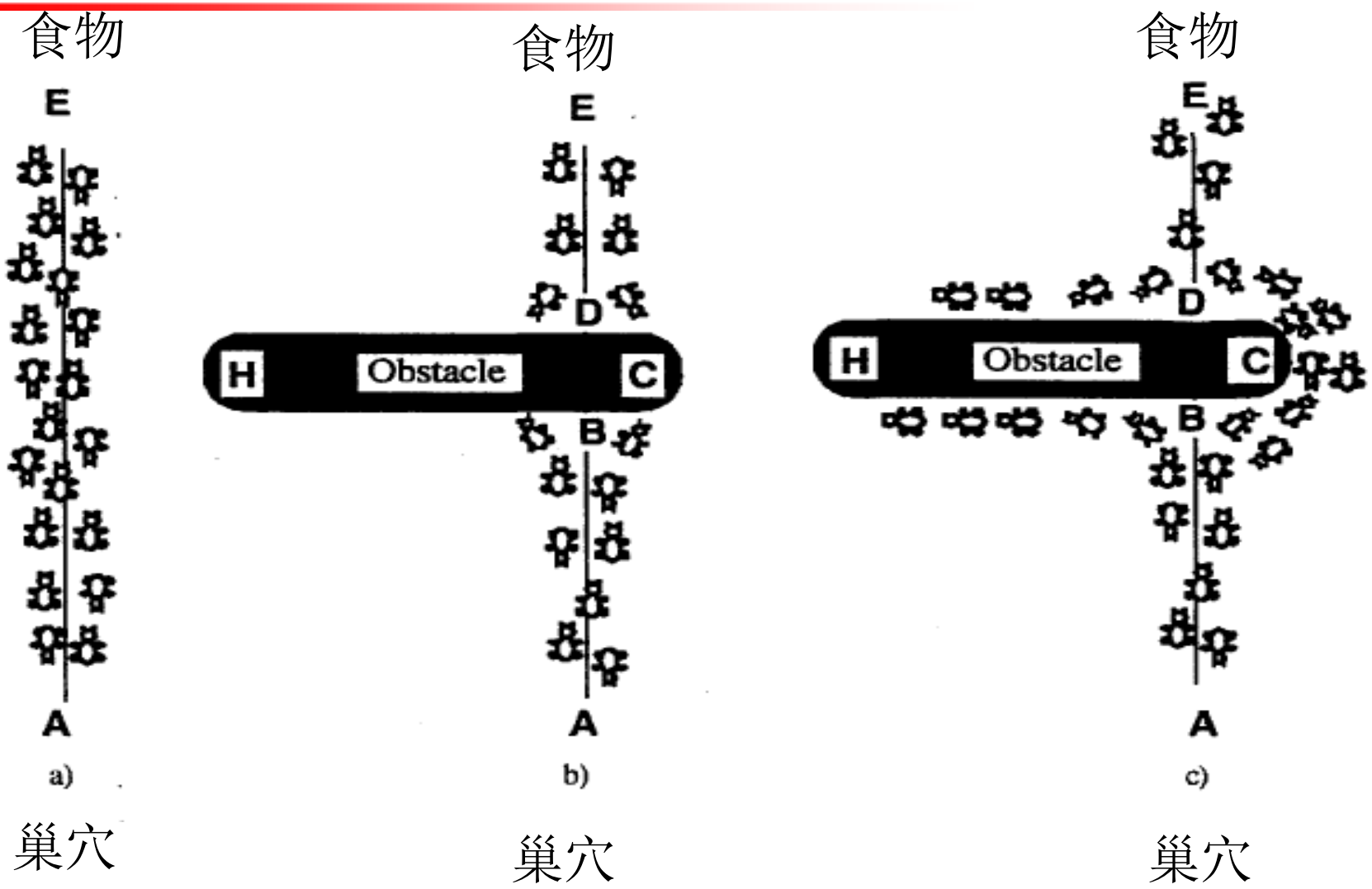
假设蚂蚁每经过一处所留下的信息素为一个单位，则经过36个时间单位后，所有开始一起出发的蚂蚁都经过不同路径从D点取得了食物，此时ABD的路线往返了2趟，每一处的信息素为4个单位，而 ACD的路线往返了一趟，每一处的信息素为2个单位，其比值为2：1。

寻找食物的过程继续进行，则按信息素的指导，蚁群在ABD路线上增派一只蚂蚁（共2只），而ACD路线上仍然为一只蚂蚁。再经过36个时间单位后，两条线路上的信息素单位积累为12和4，比值为3：1。

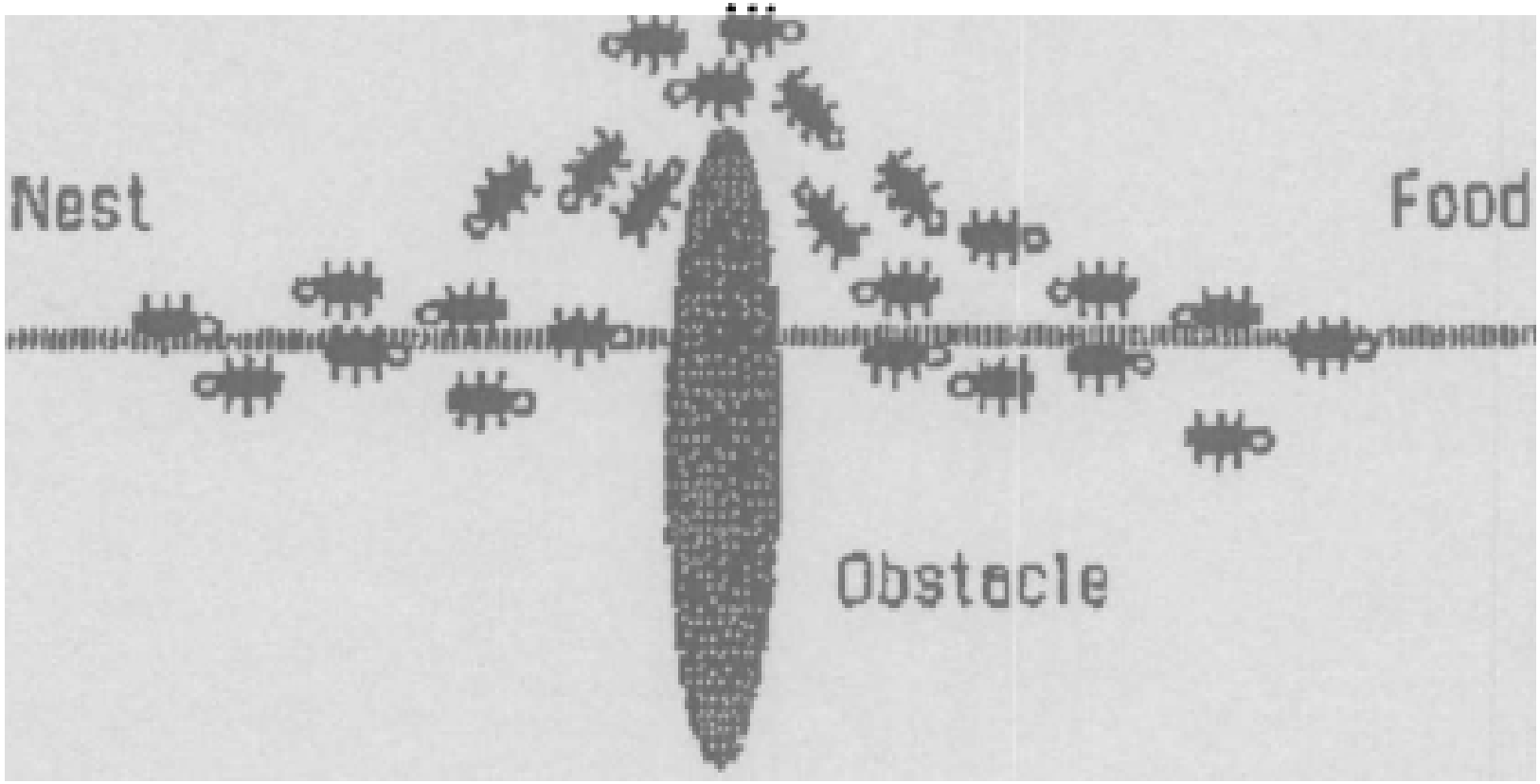
若按以上规则继续，蚁群在ABD路线上再增派一只蚂蚁（共3只），而ACD路线上仍然为一只蚂蚁。再经过36个时间单位后，两条线路上的信息素单位积累为24和6，比值为4：1。

若继续进行，则按信息素的指导，最终所有的蚂蚁会放弃ACD路线，而都选择ABD路线。这也就是前面所提到的正反馈效应。

7.1 简化的蚂蚁寻食过程



7.1 简化的蚂蚁寻食过程



7.1 自然蚁群与人工蚁群算法

- 基于以上蚁群寻找食物时的最优路径选择问题，可以构造人工蚁群，来解决最优化问题，如TSP问题。
- 人工蚁群中把具有简单功能的工作单元看作蚂蚁。二者的相似之处在于都是优先选择信息素浓度大的路径。较短路径的信息素浓度高，所以能够最终被所有蚂蚁选择，也就是最终的优化结果。
- 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。

7.1 自然蚁群与人工蚁群算法

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

7.2 蚁群优化算法的基本流程

蚂蚁系统（Ant System, AS）

- 蚂蚁系统是以TSP作为应用实例提出的，是最基本的ACO算法，比较易于学习和掌握。
- 本节将以AS求解TSP问题的基本流程为例描述蚁群优化算法的工作机制。

7.2 蚁群优化算法的基本流程

TSP问题表示为一个**N**个城市的有向图***G*** **=（*N*，*A*）**，

其中 $N=\{1,2,...,n\}$ $A=\{(i,j)\mid i,j\in N\}$

城市之间距离 $(d_{ij})_{n\times n}$

目标函数为 $f(w)=\sum_{l=1}^{n} d_{i_{l-1}i_l}$

其中 $w=(i_1,i_2,\cdots,i_n)$ 为城市**1,2，…n**的一个排列，$i_{n+1}=i_1$ 。

7.2 蚁群优化算法的基本流程

- TSP问题的人工蚁群算法中，假设m只蚂蚁在图的相邻节点间移动，从而协作异步地得到问题的解。每只蚂蚁的一步转移概率由图中的每条边上的两类参数决定：1 信息素值也称信息素痕迹。2 可见度，即先验值。

- 信息素的更新方式有2种，一是挥发，也就是所有路径上的信息素以一定的比率进行减少，模拟自然蚁群的信息素随时间挥发的过程；二是增强，给评价值“好”(有蚂蚁走过)的边增加信息素。

- 蚂蚁向下一个目标的运动是通过一个随机原则来实现的，也就是运用当前所在节点存储的信息，计算出下一步可达节点的概率，并按此概率实现一步移动，逐此往复，越来越接近最优解。

- 蚂蚁在寻找过程中，或者找到一个解后，会评估该解或解的一部分的优化程度，并把评价信息保存在相关连接的信息素中。

7.2 蚁群优化算法的基本流程

AS算法对TSP的求解有两大步骤：路径构建和信息素更新

1. 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。

7.2 蚁群优化算法的基本流程

- 随机比例规则

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

- ***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。

7.2 蚁群优化算法的基本流程

2. 信息素更新

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

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。

AS中：$C = m / C^{nn}$

7.2 蚁群优化算法的基本流程

为了模拟蚂蚁在较短路径上留下更多的信息素，当所有蚂蚁到达终点时，必须把各路径的信息素浓度重新更新一次，信息素的更新也分为两个步骤

- 首先，每一轮过后，问题空间中的所有路径上的信息素都会发生蒸发
- 然后，所有的蚂蚁根据自己构建的路径长度在它们本轮经过的边上释放信息素

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

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所留下来的信息素

7.2 蚁群优化算法的基本流程

- $\Delta\tau_{ij}^{k}$ 的定义

$$\Delta\tau_{ij}^{k}=\begin{cases}(C_k)^{-1} & if\ \ the\ k^{th}\ \ ant\ \ traverses\ \ (i,j)\\ 0 & others\end{cases}$$

C_k 是第k只蚂蚁走完整条路径后所得到的总路径长度

7.2 蚁群优化算法的基本流程

信息素更新的作用

1. 信息素挥发（**evaporation**） 信息素痕迹的挥发过程是每个连接上的信息素痕迹的浓度自动逐渐减弱的过程，这个挥发过程主要用于避免算法过快地向局部最优区域集中，有助于搜索区域的扩展。

2. 信息素增强（**reinforcement**）增强过程是蚁群优化算法中可选的部分，称为离线更新方式（还有在线更新方式）。这种方式可以实现由单个蚂蚁无法实现的集中行动。基本蚁群算法的离线更新方式是在蚁群中的***m***只蚂蚁全部完成***n***城市的访问后，统一对残留信息进行更新处理。

7.2 蚁群优化算法的基本流程

TSP问题的蚁群算法伪代码

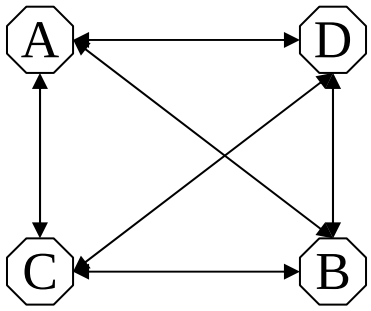
```
Procedure AS
  for each edge
    set initial pheromone value t_0
  end for
  while not stop
      for each ant k
        randomly choose an initial city
        for i=1 to n
          choose next city j with probability
        end for
      end for
      compute the length C_k of the tour constructed by the kth ant
      for each edge
        update the pheromone value
       end for
   end while
   print result
end procedure
```

7.2 蚁群优化算法的基本流程

应用举例

四个城市的**TSP**问题，距离矩阵和城市图示如下：

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



7.2 蚁群优化算法的基本流程

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

7.2 蚁群优化算法的基本流程

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

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

$$p(B)=0.033/(0.033+0.3+0.075)=0.081$$

$$p(C)=0.3/(0.033+0.3+0.075)=0.74$$

$$p(D)=0.075/(0.033+0.3+0.075)=0.18$$

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。

7.2 蚁群优化算法的基本流程

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^l=(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:

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

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

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

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。

7.2 蚁群优化算法的基本流程

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

7.2 蚁群优化算法的基本流程

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

$$\tau_{AB}=(1-\rho)\times\tau_{AB}+\sum_{k=1}^{3}\Delta\tau_{AB}^{k}=0.5\times0.3+(1/10+1/10)=0.35$$

$$\tau_{AC}=(1-\rho)\times\tau_{AC}+\sum_{k=1}^{3}\Delta\tau_{AC}^{k}=0.5\times0.3+(1/12)=0.16$$

…

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。

7.3 蚁群优化算法研究现状

一般蚁群算法的框架和基本蚁群算法大致相同，有三个组成部分：

- 蚁群的活动；
- 信息素的挥发；
- 信息素的增强；

主要体现在转移概率公式和信息素更新公式的不同。

7.3 蚁群优化算法研究现状

1：蚁群优化算法的发展

- **90年代Dorigo最早提出了蚁群优化算法——蚂蚁系统（Ant System, AS），并将其应用于解决计算机算法学中经典的旅行商问题（TSP）。**

- **从蚂蚁系统开始，基本的蚁群算法得到了不断的发展和完善，并在TSP以及许多实际优化问题求解中进一步得到了验证；**

- **这些AS改进版本的一个共同点就是增强了蚂蚁搜索过程中对最优解的探索能力，它们之间的差异仅在于搜索控制策略方面；**

- **取得最好结果的ACO是通过引入局部搜索算法实现的，实际上是一些结合了标准局域搜索算法的混合型概率搜索算法，有利于提高蚁群各级系统在优化问题中的求解质量。**

7.3 蚁群优化算法研究现状

2：蚂蚁系统的三种基本版本

Ant-density、Ant-quantity和Ant-cycle。

在**Ant-density**和**Ant-quantity**中蚂蚁在两个位置节点间每移动一次即更新信息素；

在**Ant-cycle**中当所有的蚂蚁都完成了自己的行程后才对信息素进行更新，而且每个蚂蚁所释放的信息素被表达为反映相应行程质量的函数。

通过与其它各种通用的启发式算法相比，在不大于**75**城市的**TSP**中，这三种基本算法的求解能力还是比较理想的，但是当问题规模扩展时，**AS**的解题能力大幅度下降。

7.3 蚁群优化算法研究现状

3：精英策略的蚂蚁系统(Elitist Ant System, EAS)

- AS算法中，蚂蚁在其爬过的边上释放与其构建路径长度成反比的信息素量，蚂蚁构建的路径越好，则属于路径的各个边上的所获得的信息素量就越多，这些边以后在迭代中被蚂蚁选择的概率也就越大。

- 我们可以想象，当城市的规模较大时，问题的复杂度呈指数级增长，仅仅靠这样一个基础单一的信息素更新机制引导搜索偏向，搜索效率有瓶颈。

- 我们是否可以通过一种“额外的手段”强化某些最有可能成为最优路径的边，让蚂蚁的搜索范围更快、更正确的收敛呢？

7.3 蚁群优化算法研究现状

3：精英策略的蚂蚁系统 (Elitist Ant System, EAS)

精英策略(Elitist Strategy)：

- 一种较早的改进方法；

- 在算法开始后即对所有已发现的最好路径给予额外的增强，并将随后与之对应的行程记为T_b(全局最优行程)；

- 当进行信息素更新时，对这些行程予以加权，同时将经过这些行程的蚂蚁记为“精英”，从而增大较好行程的选择机会。

7.3 蚁群优化算法研究现状

3：精英策略的蚂蚁系统 (Elitist Ant System, EAS)

$$\tau_{ij}(t)=(1-\rho)\tau_{ij}+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}+e\Delta\tau_{b}(i,j)$$

$$\Delta\tau_{k}(i,j)=\begin{cases}(C_k)^{-1}, & (i,j)\in R^{K}\\ 0, & others\end{cases}$$

$$\Delta\tau_{b}(i,j)=\begin{cases}(C_b)^{-1}, & (i,j)\in T_{b}\\ 0, & others\end{cases}$$

这种改进能够以更快的速度获得更好的解。但是若选择的精英过多则算法会由于较早的收敛于局部次优解而导致搜索的过早停滞。

7.3 蚁群优化算法研究现状

4: 基于排列的蚂蚁系统 (Rank-based AS, AS_{rank})

- 在精华蚂蚁系统被提出之后，人们又想到，有没有更好的一种信息素更新方式，它同样使得Tb各边的信息素浓度得到加强，且对其余边的信息素更新机制亦有改善？

7.3 蚁群优化算法研究现状

4: 基于排列的蚂蚁系统 Rank-based AS

- 与“精英策略”相似，在此算法中总是更新更好进程上的信息素，选择的标准是其行程长度决定的排序；
- 每只蚂蚁放置信息素的强度通过下式中的排序加权处理确定。

$$\tau_{ij}(t)=(1-\rho)\tau_{ij}+\sum_{k=1}^{\omega-1}(\omega-k)\Delta\tau_{ij}^{k}+\omega\Delta\tau_{b}(i,j)$$

$$\Delta\tau_{k}(i,j)=\begin{cases}(C_k)^{-1}, & (i,j)\in R^{K}\\ 0, & others\end{cases}$$

$$\Delta\tau_{b}(i,j)=\begin{cases}(C_b)^{-1}, & (i,j)\in T_{b}\\ 0, & others\end{cases}$$

其中， ω 为每次迭代后放置信息素的蚂蚁总数，一般设置 ω =6

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

首先思考几个问题：

问题一：对于大规模的**TSP**，由于搜索蚂蚁的个数有限，而初始化时蚂蚁的分布是随机的，这会不会造成蚂蚁只搜索了所有路径中的小部分就以为找到了最好的路径，所有的蚂蚁都很快聚集在同一路径上，而真正优秀的路径并没有被搜索到呢？

问题二：当所有蚂蚁都重复构建着同一条路径的时候，意味着算法的已经进入停滞状态。此时不论是基本**AS**、**EAS**还是$\mathbf{AS}_{\mathbf{rank}}$，之后的迭代过程都不再可能有更优的路径出现。这些算法收敛的效果虽然是“单纯而快速的”，但我们都懂得欲速而不达的道理，我们有没有办法利用算法停滞后的迭代过程进一步搜索以保证找到更接近真实目标的解呢？

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

在对AS进行直接完善的方法中，**MAX-MIN Ant System** 是一个典型代表。

1. 该算法修改了AS的信息素更新方式，只允许迭代最优蚂蚁（在本次迭代构建出最短路径的蚂蚁），或者至今最优蚂蚁释放信息素；
2. 路径上的信息素浓度被限制在[MAX，MIN]范围内；
3. 另外，信息素的初始值被设为其取值上限，这样有助于增加算法初始阶段的搜索能力。
4. 为了避免搜索停滞，问题空间内所有边上的信息素都会被重新初始化。

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

- 改进1借鉴于精华蚂蚁系统，但又有细微的不同。
- 在EAS中，只允许至今最优的蚂蚁释放信息素，而在MMAS中，释放信息素的不仅有可能是至今最优蚂蚁，还有可能是迭代最优蚂蚁。
- 实际上，迭代最优更新规则和至今最优更新规则在MMAS中是交替使用的。
- 这两种规则使用的相对频率将会影响算法的搜索效果。
- 只使用至今最优更新规则进行信息素的更新，搜索的导向性很强，算法会很快的收敛到T_b附近；反之，如果只使用迭代最优更新规则，则算法的探索能力会得到增强，但搜索效率会下降。

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

- 改进2是为了避免某些边上的信息素浓度增长过快，出现算法早熟现象。
- 蚂蚁是根据启发式信息和信息素浓度选择下一个城市的。
- 启发式信息的取值范围是确定的。
- 当信息素浓度也被限定在一个范围内以后，位于城市i的蚂蚁k选择城市j作为下一个城市的概率$p_k(i,j)$也将被限制在一个区间内。

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

- 改进3，使得算法在初始阶段，问题空间内所有边上的信息素均被初始化 τ_{max}的估计值，且信息素蒸发率非常小（在MMAS中，一般将蒸发率设为0.02）。
- 算法的初始阶段，不同边上的信息素浓度差异只会缓慢的增加，因此，MMAS在初始阶段有着较基本AS、EAS和AS_{rank}更强搜索能力。
- 增强算法在初始阶段的探索能力有助于蚂蚁“视野开阔地”进行全局范围内的搜索，随后再逐渐缩小搜索范围，最后定格在一条全局最优路径上。

7.3 蚁群优化算法研究现状

5: 最大最小蚂蚁系统(MAX-MIN Ant System, MMAS)

- 之前的蚁群算法，包括AS、EAS以及AS_{rank}，都属于“一次性探索”，即随着算法的执行，某些边上的信息素量越来越小，某些路径被选择的概率也越来越小，系统的探索范围不断减小直至陷入停滞状态。
- MMAS中改进4，当算法接近或者是进入停滞状态时，问题空间内所有边上的信息素浓度都将被初始化，从而有效的利用系统进入停滞状态后的迭代周期继续进行搜索，使算法具有更强的全局寻优能力。

7.3 蚁群优化算法研究现状

6：蚁群系统(Ant Colony System)

上述EAS、AS_{rank}以及MMAS都是对AS进行了少量的修改而获得了更好的性能。1997年，蚁群算法的创始人Dorigo在Ant colony system：a cooperative learning approach to the traveling salesman problem一文中提出了一种新的具有全新机制的ACO算法——蚁群系统，是蚁群算法发展史上的又一里程碑。

7.3 蚁群优化算法研究现状

6：蚁群系统(Ant Colony System)

ACS与AS之间存在三方面的主要差异：

1. 使用一种伪随机比例规则选择下一个城市节点，建立开发当前路径与探索新路径之间的平衡。
2. 信息素全局更新规则只在属于至今最优路径的边上蒸发和释放信息素。
3. 新增信息素局部更新规则，蚂蚁每次经过空间内的某条边，他都会去除该边上的一定量的信息素，以增加后续蚂蚁探索其余路径的可能性。

7.3 蚁群优化算法研究现状

(1) 状态转移规则：

- 一只位于节点i的蚂蚁通过应用下式给出的规则选择下一个将要移动到的城市j

$$j=\begin{cases}\arg\max\limits_{j\in allowed_k}\{[\tau(i,j)]\cdot[\eta(i,j)]^{\beta}\} & if\ q\le q_0\\ S & else\end{cases}$$

其中，S根据下列公式得到

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}(t)}, & j\in allowed_k\\ 0, & otherwise\end{cases}$$

- q是在[0,1]区间均匀分布的随机数
- q_0的大小决定了利用先验知识与探索新路径之间的相对重要性。
- 上述状态转移规则被称为伪随机比例规则

7.3 蚁群优化算法研究现状

(2) 信息素全局更新规则

- **ACS**采用了更为大胆的行为选择规则；
- 只增强属于全局最优解的路径上的信息素。其中，**0< ρ <1**是信息素挥发参数，C_b是从寻路开始到当前为止全局最优的路径长度。

$$\tau_{ij}(t) = (1-\rho)\tau_{ij} + \rho\Delta\tau_b(i,j)$$

$$\Delta\tau_b(i,j) = \begin{cases} (C_b)^{-1}, & (i,j) \in T_b \\ 0, & others \end{cases}$$

7.3 蚁群优化算法研究现状

(3) 信息素局部更新规则

- 引入了负反馈机制：每当一只蚂蚁由一个节点移动到另一个节点时，该路径上的信息素都按照如下公式被相应的消除一部分，实现一种信息素的局部调整，减小已选择过的路径再次被选择的概率。

$$\tau_{ij} = (1-\xi)\cdot\tau_{ij} + \xi\cdot\tau_0 \qquad 0<\xi<1$$

7.3 蚁群优化算法研究现状

7: 各种算法求解TSP问题中的性能比较：

Rank-based AS与**AS**、精英策略**AS**、遗传算法和模拟退火算法进行了比较：

- 在大型**TSP**问题中（最多包含**132**座城市），基于**AS**的算法都显示出了优于**GA**和**SA**的特性；
- **Rank-based AS**和精英策略**AS**均优于基本**AS**；前者**Rank-based AS**获得了比精英策略**AS**更好的解。

7.4 蚁群优化算法—技术问题

7.4.1 解的表达形式与算法的实现

7.4.2 每一节点的记忆信息和系数的确定

7.4.3 蚁群的规模和停止规则

7.4.4 信息素的更改

7.4.1 解的表达形式与算法实现 —解的表达形式

- 解的表达形式　基于**TSP**问题的蚁群优化算法，其解的形式是所有城市的一个排列（闭圈，这种情况下谁在第一并不重要），信息素痕迹按每个弧记录。
- 对于一般以顺序作为解的优化问题，谁在第一是很重要的。蚁群算法在解决这类问题时，只需要建立一个虚拟的始终点，就可以把**TSP**问题的解法推广，用于诸多的优化问题。
- 诸如车间作业及下料等问题，他们的共同特点是解以一个顺序表示。**TSP**问题寻找的是最短回路，而一般优化问题中的判断条件需要根据实际问题进行修改。

7.4.1 解的表达形式与算法实现 —算法的实现

例：**0-1**背包问题的解顺序表达形式与算法实现。设有一个容积为**b**的背包，**n**个尺寸分别为$a_i(i=1,2,...,n)$，价值分别为 $c_i(i=1,2,...,n)$的物品，**0-1**背包问题的数学模型为：

$$\max \sum_{i=1}^{n} c_i x_i$$

$$s.t.\sum_{i=1}^{n} a_i x_i \le b$$

$$x_i \in \{0,1\}, i = 1,\ldots,n$$

假设其解的顺序表达形式为 $(0, i_1, i_2, \ldots, i_n)$，其中 $(i_1, i_2, ..., i_n)$ 为 $(1,2,3,...,n)$ 的一个排列。

7.4.1 解的表达形式与算法实现—算法的实现

建立有向图 $G=(V,A)$ ，其中 $V=\{0,1,2,....,n\}, A=\{(i,j)\mid \forall i,j\in V\}$ **A**中共有 $n(n+1)$条弧。初始信息素痕迹定义为 $\tau_{ij}=1/n(n+1)$ 。设第**s**只蚂蚁第**k**步所走的路线为 $L(s)=(0,i_1,i_2,....,i_k)$ ，表示蚂蚁从**0**点出发，顺序到达 $i_1,i_2,....,i_k$ 。第 $k+1$步按**TSP**算法的转移概率公式行走选择 i_{k+1} 。若 $\sum_{j=1}^{k+1}a_{i_j}\le b$ 则 $L(s)=(0,i_1,i_2,....,i_k,i_{k+1})$ ，否则，此蚂蚁不再继续行走，退回起点。

7.4.1 解的表达形式与算法实现—算法的实现

对蚁群重复以上过程，比较**m**只蚂蚁的装包值 $\sum_{\substack{i\in L(s)\\ i\neq 0}} c_i, s=1,2,...,m$ 并记忆具有最大装包值的蚂蚁为**t**。把**AS**算法中步骤**3**中的 $f(L(t))<f(w)$ 改为 $f(L(t))>f(w)$ ，若满足此条件则替换当前最好解为 $W:=L(t)$ ，对**W**上的弧进行信息素的加强，其他弧进行信息素的挥发。

算法中记录了三个信息：信息素痕迹 τ_{ij} ；行走路线 $L(s)=(0,i_1,i_2,....,i_k,i_{k+1})$ ；和问题的约束条件 $\sum_{j=1}^{k+1} a_{i_j}\leq b$ ，以确定是否将 i_{k+1} 加入。

7.4.2 每一节点记忆信息和系数的确定—需要记忆的信息

算法中需要记忆的信息有三部分。

第一部分信息是存在每个节点的路由表数据结构 $A_i = \{a_{ij} \mid (i,j) \in A\}$ ，由此决定的的转移概率为

$$P_{ij} = \begin{cases} \dfrac{a_{ij}(k-1)}{\sum_{l \in T} a_{il}(k-1)}, & j \in T \\ 0, & j \notin T \end{cases}$$

其中**T**可以看成节点**i**的邻域。 $A_i(k-1) = \{a_{ij}(k-1) \mid (i,j) \in A\}$

$$a_{ij}(k-1) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(k-1)\eta_{ij}^{\beta}(k-1)}{\sum_{l \in T} \tau_{il}^{\alpha}(k-1)\eta_{il}^{\beta}(k-1)}, & j \in T \\ 0, & j \notin T \end{cases}$$

7.4.2 每一节点记忆信息和系数的确定—需要记忆的信息

第二部分需要记忆的信息是每个蚂蚁的记忆表中存储着的自身的历史信息，这一部分主要由算法的中的 $L(s)$记忆，表示蚂蚁已经行走过的节点。

第三部分为问题的约束条件。在**AS**中，**T**集合表示满足约束条件的候选集，在背包问题的蚁群算法中由判别条件

$$\sum_{j=1}^{k+1} a_{i_j} \le b \quad , \qquad L(s) = (0, i_1, i_2, ..., i_k, i_{k+1})$$

来实现记忆功能。

7.4.2 每一节点记忆信息和系数的确定 —系数的确定

残留信息的相对重要程度 α 和预见值的相对重要程度 β 体现了相关信息痕迹和预见度对蚂蚁决策的相对影响。**Dorigo**在求解**TSP**问题时，推荐参数的最佳设置为：

$$\alpha = 1, \beta = 5, \rho = 0.5$$

7.4.3 蚁群的规模和停止规则

一、蚁群大小：

一般情况下蚁群中蚂蚁的个数不超过TSP图中节点的个数。

二、终止条件：

1 给定一个外循环的最大数目，表明已经有足够的蚂蚁工作；

2 当前最优解连续K次相同而停止，其中K是一个给定的整数，表示算法已经收敛，不再需要继续；

3 目标值控制规则，给定优化问题（目标最小化）的一个下界和一个误差值，当算法得到的目标值同下界之差小于给定的误差值时，算法终止。

7.4.4 信息素的更改

信息素的更新分为离线和在线两种方式。

离线方式（同步更新方式）：在若干只蚂蚁完成**n**个城市的访问后，统一对残留信息进行更新处理。

信息素的在线更新（异步更新方式）：蚂蚁每行走一步，立即回溯并且更新行走路径上的信息素。

7.4.4 信息素的更改

离线方式的信息素更新可以进一步分为单蚂蚁离线更新和蚁群离线更新。

蚁群离线更新方式是在蚁群中的**m**只蚂蚁全部完成**n**城市的访问（第**k-1**次蚁群循环）后，统一对残留信息进行更新处理。

$$\tau_{ij}(k)=\tau_{ij}(k-1)+\Delta\tau_{ij}(k-1)$$

其中，$\tau_{ij}(k)$为第**k-1**次循环后的的信息素的痕迹值。

单蚂蚁离线更新是在第**s**只蚂蚁完成对所有**n**个城市的访问后，进行路径回溯，更新行走路径上的信息素，同时释放分配给它的资源。更新公式为：

$$\tau_{ij}(s+1)=\tau_{ij}(s)+\Delta\tau_{ij}(s)$$

第**s+1**只蚂蚁根据 $\tau_{ij}(s+1)$ 重新计算路由表。

7.4.4 信息素的更改

TSP问题中，蚁群优化算法根据信息素痕迹更新方式不同可以分为不同的算法，采用离线方式，并且 $\Delta\tau_{ij}(k-1)$或$\Delta\tau_{ij}(s)$为

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{Q}{|W|},(i,j)\in W\\ 0 \quad (i,j)\notin W\end{cases}$$

时，其中**W**为**t**循环中**m**只蚂蚁所行走的最佳路线或第**t**只蚂蚁所行走的一条路径。**Q**为一个常数，该算法名为蚁环算法（**ant-cycle algotithm**），特点是行走的路径越短对应保存的信息素的值就越大。

7.4.4 信息素的更改

EAS算法是典型的离线信息素更新方式。该算法中，蚁群中蚂蚁的先后出行顺序没有相关性，但是每次循环需要记忆m只蚂蚁的行走路径，以进行比较选择最优路径。相对而言，单蚂蚁离线更新方式记忆信息少，只需要记忆第s只蚂蚁的路径，并通过信息素更新后，释放该蚂蚁的所有记录信息。实际上这种方式等价于蚁群离线方式中只有一只蚂蚁。

7.4.4 信息素的更改

与单蚂蚁离线更新方式相比，信息量记忆更小的是信息素在线更新方式，即蚂蚁每走一步，马上回溯并且更新刚刚走过的路径上的信息素，其规则为

$$\tau_{ij}(k+1)=\tau_{ij}(k)+\Delta\tau_{ij}(k)$$

其中，**k**为蚂蚁行走的第**k**步。

7.4.4 信息素的更改

蚁量算法（**ant-quantity algorithm**）的信息素更新为

$\Delta\tau_{ij}(k)=\dfrac{Q}{d_{ij}}$，**Q**为常量，$d_{ij}$ 表示**i**到**j**的距离，这样信息浓度会随城市距离的减小而加大。

蚁密算法（ **ant-density algorithm** ）信息素更新为 $\Delta\tau_{ij}(k)=Q$。

以上三种算法中，蚁环算法效果最好，因为用的是全局信息，而其余两种算法用的是局部信息。蚁环离线更新方法很好地保证了残留信息不至于无限积累，非最优路径会逐渐随时间推移被忘记。

7.5 蚁群优化算法应用现状

各种工程优化问题：

随着群智能理论和应用算法研究的不断发展，研究者已尝试着将其用于各种工程优化问题，并取得了意想不到的收获。多种研究表明，群智能在离散求解空间和连续求解空间中均表现出良好的搜索效果，并在组合优化问题中表现突出。

蚁群优化算法并不是旅行商问题的最佳解决方法，但是它却为解决组合优化问题提供了新思路，并很快被应用到其它组合优化问题中。比较典型的应用研究包括：网络路由优化、数据挖掘以及一些经典的组合优化问题。

7.5 蚁群优化算法应用现状

蚁群算法在电信路由优化中的应用：

蚁群算法在电信路由优化中已取得了一定的应用成果。**HP**公司和英国电信公司在**90**年代中后期都开展了这方面的研究，设计了蚁群路由算法（**Ant Colony Routing, ACR**）。

每只蚂蚁就像蚁群优化算法中一样，根据它在网络上的经验与性能，动态更新路由表项。如果一只蚂蚁因为经过了网络中堵塞的路由而导致了比较大的延迟，那么就对该表项做较大的增强。同时根据信息素挥发机制实现系统的信息更新，从而抛弃过期的路由信息。这样，在当前最优路由出现拥堵现象时，**ACR**算法就能迅速的搜寻另一条可替代的最优路径，从而提高网络的均衡性、负荷量和利用率。目前这方面的应用研究仍在升温，因为通信网络的分布式信息结构、非稳定随机动态特性以及网络状态的异步演化与**ACO**的算法本质和特性非常相似。

7.5 蚁群优化算法应用现状

蚁群算法聚类问题的应用：

基于群智能的聚类算法起源于对蚁群蚁卵的分类研究。**Lumer**和**Faieta**将**Deneubourg**提出将蚁巢分类模型应用于数据聚类分析。其基本思想是将待聚类数据随机地散布到一个二维平面内，然后将虚拟蚂蚁分布到这个空间内，并以随机方式移动，当一只蚂蚁遇到一个待聚类数据时即将之拾起并继续随机运动，若运动路径附近的数据与背负的数据相似性高于设置的标准则将其放置在该位置，然后继续移动，重复上述数据搬运过程。按照这样的方法可实现对相似数据的聚类。

7.5 蚁群优化算法应用现状

蚁群算法在组合优化问题中的应用：

ACO还在许多经典组合优化问题中获得了成功的应用，如二次规划问题（**QAP**）、机器人路径规划、作业流程规划、图着色（**Graph Coloring**）等问题。

经过多年的发展，**ACO**已成为能够有效解决实际二次规划问题的几种重要算法之一。**AS**在作业流程计划（**Job-shop Scheduling**）问题中的应用实例已经出现，这说明了**AS**在此领域的应用潜力。利用**MAX-MIN AS**解决**QAP**也取得了比较理想的效果，并通过实验中的计算数据证明采用该方法处理**QAP**比较早的**SA**算法更好，且与禁忌搜索算法性能相当。利用**ACO**实现对生产流程和特料管理的综合优化，并通过与遗传、模拟退火和禁忌搜索算法的比较证明了**ACO**的工程应用价值。

7.5 蚁群优化算法应用现状

蚁群算法的其它应用：

- 武器攻击目标分配和优化问题
- 车辆运行路径规划
- 区域性无线电频率自动分配
- **Bayesian networks**的训练
- 集合覆盖
- **Costa**和**Herz**还提出了一种**AS**在规划问题方面的扩展应用—图 着色问题，取得了与其他启发式算法相似的效果。

第七章：蚁群算法

第七章 结束