高级人工智能

Advanced Artificial Intelligence

蚁群算法

蚁群算法概述

（Ant Colony Optimization, ACO）

- 意大利学者M.Dorgio 于1991年提出
- 受蚁群觅食行为启发而设计的群智能随机优化算法
- 蚂蚁在觅食时沿途留下信息素，无需其它先验知识，蚁群根据信息素可找到从蚁穴至食物源的最短路径

蚁群算法概述

- 自然界中的蚂蚁能找到最短路径依靠的是前驱的蚂蚁在途中留下的信息素，蚂蚁本能地选择信息素较浓的路径

- 信息素会随时间挥发，导致蚂蚁较少的路径信息素浓度减弱。

- 同样时间间隔最短路径有更多蚂蚁通过，该路径信息素增强而其他路径则减弱，最终蚁群选择最短路径

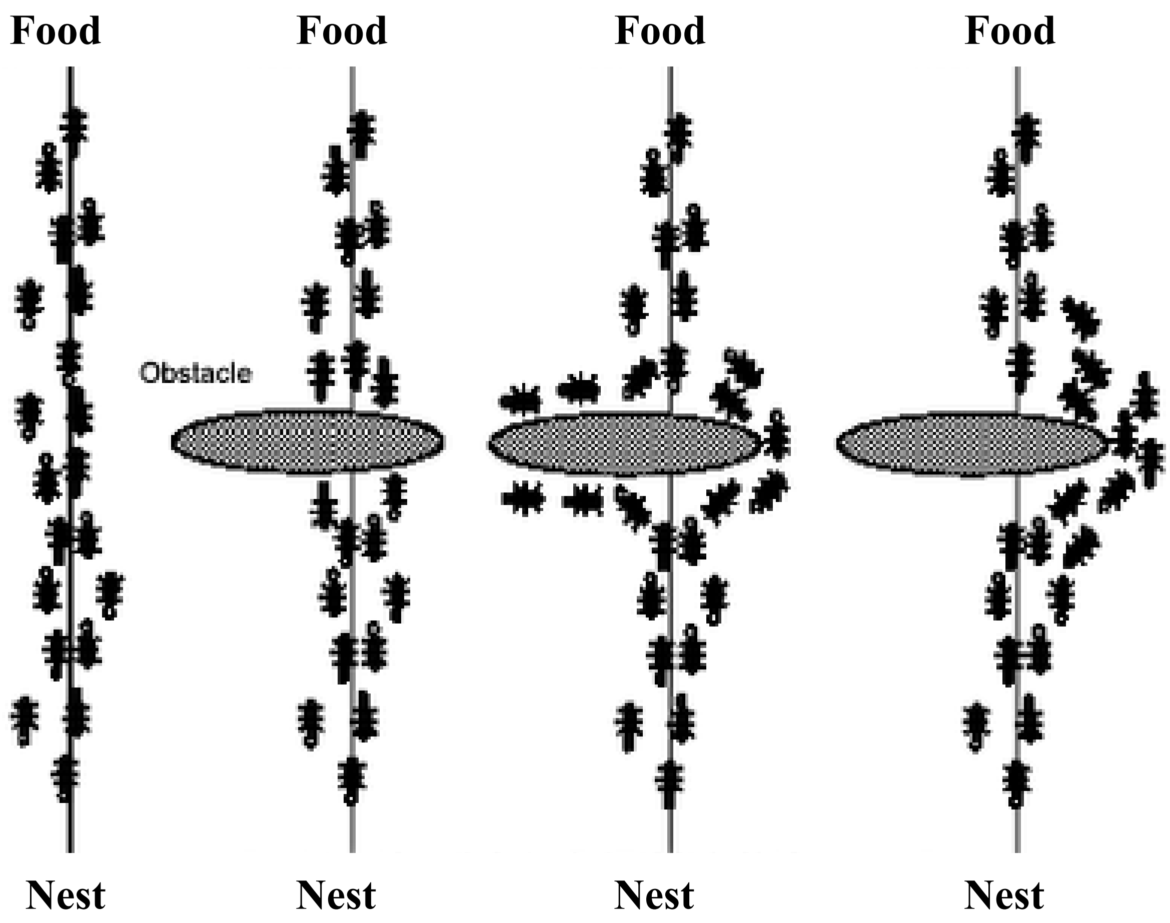
蚁群算法概述
（Ant Colony Optimization, ACO）

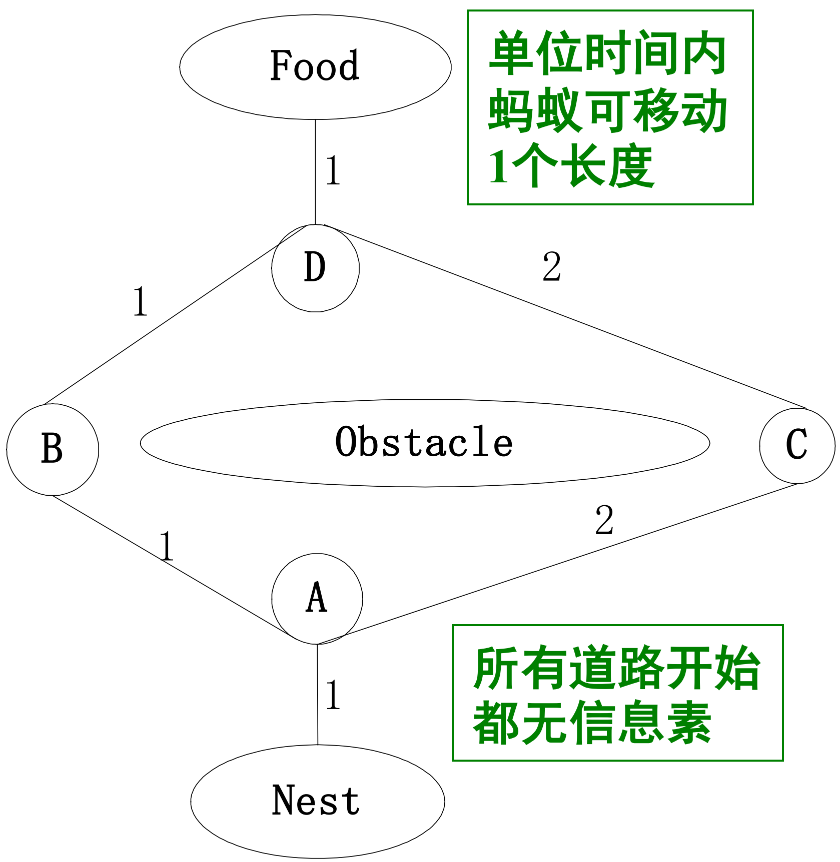
■正反馈机制（增强型学习系统）。通过【最优路径上蚂蚁数量的增加→信息素强度增加→后来蚂蚁选择概率增大→最优路径上蚂蚁数量进一步增加】达到最终收敛于最优路径

■通用型随机优化方法, 吸收了蚂蚁的行为特征(内在搜索机制),使用人工蚂蚁仿真来求解问题。

■并非对实际蚂蚁的简单模拟, 融进了智能策略：有一定的记忆; 随机但不盲目; 时空是离散的

蚁群算法基本原理



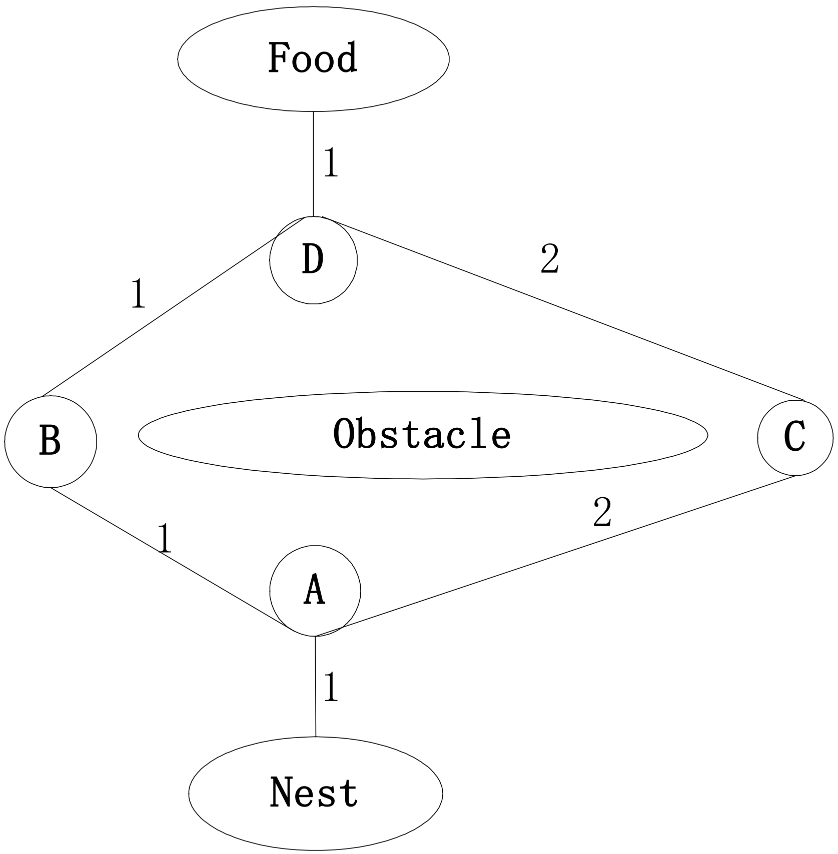


t=0：20只蚂蚁从Nest移到A。等概率地10只走左侧①组，10只走 右侧②组

t=4：①组到达Food，将折回

t=5：两组蚂蚁在D相遇。此时BD的信息素与CD相同，故①组5只返回选BD，另5只选CD

t=8：①组BD线的5只返回Nest，另5只走到AC中点；②组返回已过D点各有5只在B点、CD上，故AC，CD和B点上各有5只蚂蚁。



t=9：新一批蚂蚁从Nest移到A并再次面对选择。这时:
AB上的信息素是20而AC上是15，AB线被增强。

AB(20)：去①组10只;
回①组5只(N)+②组5只(A)

AC(15):去②组10只;
回①组5只 (A)

②组5只(C)

蚁群算法流程设计

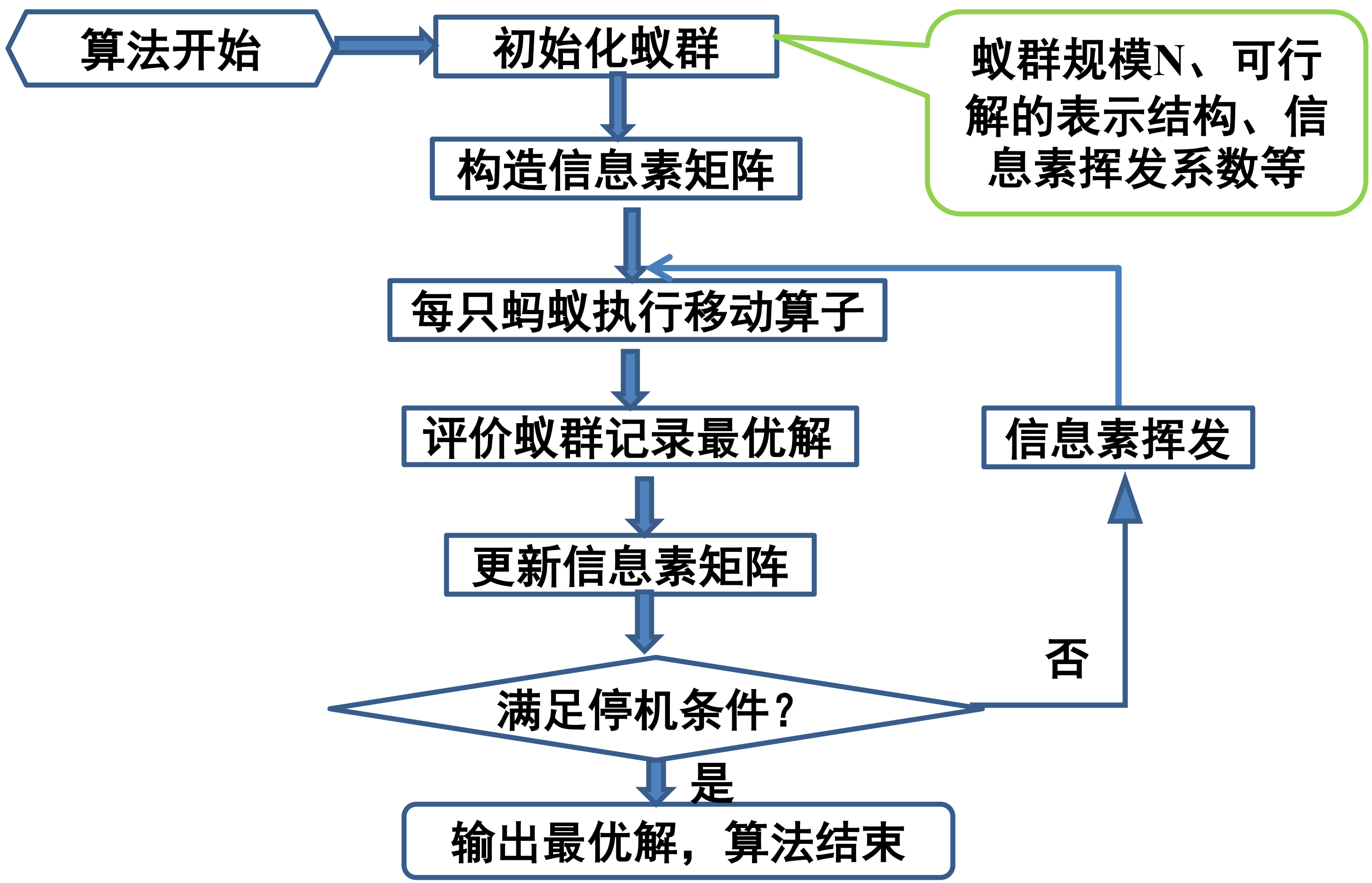
- 问题建模为图结构寻找最短路径问题

- 信息素的表达：信息素矩阵 $T(t)=[\tau_{ij}(t)]_{M\times M}$
  信息素的更新：$\tau_{ij}(t+1)=a\tau_{ij}(t)+\Delta\tau_{ij}$
  M为图节点个数

- 信息素的利用：在蚂蚁下一步走步选择时设计选择概率与边的信息素成正比

- 一次迭代中，所有蚂蚁都要获得一条路径，评估更新后进入下一次迭代

蚁群算法流程图



算法开始
初始化蚁群
蚁群规模N、可行解的表示结构、信息素挥发系数等
构造信息素矩阵
每只蚂蚁执行移动算子
评价蚁群记录最优解
信息素挥发
更新信息素矩阵
满足停机条件？
否
是
输出最优解，算法结束

蚁群算法构成要素-移动算子

蚂蚁从图中一个节点移向另一个节点的操作，赌轮选择法

■第t次循环时节点 i 的蚂蚁k移向节点 j 的选择概率如下：

$$P_{ij}^{k}(t)=\frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in allowed_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)},j\in allowed_k$$

$allowed_k$ ：蚂蚁k可访问但尚未访问过的节点集

$\eta_{ij}(t)$ ：边(i,j)的能见度（从节点i移向j用到的启发信息项）

$\tau_{ij}(t)$ ：边(i,j)上的信息素

$$\tau_{ij}(t)\eta_{ij}(t)\Rightarrow\frac{\tau_{ij}(t)\eta_{ij}(t)}{\sum_{s\in allowed_k}\tau_{is}(t)\eta_{is}(t)}\Rightarrow$$

α>0:轨迹的相对重要性（≈1）

β>0:能见度的相对重要性（2~5）

α=0：算法演变成传统的随机贪心算法，最邻近城市被选中的概率最大。
β=0：蚂蚁完全只根据信息素浓度确定路径，算法收敛快，但获得的最优路径往往与实际目标有较大差异，算法的性能较差

蚁群算法构成要素

信息素更新算子

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

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

$\Delta\tau_{ij}$ ：边(i,j)上的信息素增量

$\Delta\tau_{ij}^{k}$ ：第k只蚂蚁本次循环中在 边(i,j)上释放的信息素

$0<\rho<1$：信息素挥发系数（～0.5）

$$\tau_0 = \frac{N}{L_0}$$

蚁群算法构成要素

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

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

信息素更新算子

蚁密系统（局部）

$$\Delta\tau_{ij}^{k}=\begin{cases} Q & if\ \ k^{th}\text{蚂蚁在本次循环经过路径}(i,j) \\ 0 & otherwise \end{cases}$$

Q:体现蚂蚁所留轨迹数量的一个常数

蚁量系统（局部）

$$\Delta\tau_{ij}^{k}=\begin{cases} Q/d_{ij} & if\ \ k^{th}\text{蚂蚁在本次循环经过路径}(i,j) \\ 0 & otherwise \end{cases}$$

d_{ij}:边(i,j)的长度

蚁周系统（整体）

$$\Delta\tau_{ij}^{k}=\begin{cases} Q/L_{k} & if\ \ k^{th}\text{蚂蚁在本次循环经过路径}(i,j) \\ 0 & otherwise \end{cases}$$

L_k:第k只蚂蚁在本次循环中所走路径总长

蚁群算法构成要素

适应度函数与禁忌表(Tabu List)

- 需要一个适应度函数$F(t)$来评价搜索得到的解的优劣
- 蚂蚁k的移动受到一定的约束，有些节点在当前位置是不允许访问的，可通过设置该蚂蚁的禁忌表来完成。

例：在TSP问题中，

$\text{Tabu}_k(t)$={蚂蚁k已经访问过的节点}

当$\text{Tabu}_k(t)$含有n个顶点时，完成一次循环

可进行适应值评价（如路径长度L_k）

清空$\text{Tabu}_k(t)$，并重新开始循环。

蚁群算法构成要素

控制参数选择

■**种群大小N**：大可增加找到最优解的概率但算法复杂度亦高；小难以保证找到最优解但计算量小。求解TSP问题通常N=城市数

■**信息素挥发系数 ρ**：太小增加已访问的区域被再次访问的可能，全局搜索能力减弱；过大有较好的随机性和全局搜索能力但收敛速度可能降低

蚁群算法举例-对称n-TSP问题求解

For $t = 1$ to t_{max} do(迭代次数)

For k =1 to N do（种群个数）

T[k](t)：蚂蚁k第t次迭代获得的路径，该路径需执行如下步骤n-1次：

■第t次循环时节点 i 的蚂蚁k移向节点 j 的选择概率如下：

$$P_{ij}^{k} = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{s \in J^{k}}[\tau_{is}(t)]^{\alpha}[\eta_{is}]^{\beta}}$$

i:蚂蚁所在的当前节点

J^{k} :蚂蚁k当前可访问的节点集合

Update $J^{k} \Leftarrow J^{k} \setminus \{j\}$;

End For (k)

蚁群算法举例-对称n-TSP问题求解

For k=1 to N do

　计算蚂蚁k产生的路径$T^k(t)$的长度 $L^k(t)$

End For

If 路径有改进then

　　　update T* and L*

End If

For 每条边(i, j) do

　更新信息素： $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}$

End For

End For(t)

- **例5.1 给出用蚁群算法求解一个四城市的TSP问题的执行步骤，四个城市A、B、C、D之间的距离矩阵如下**

$$W = d_{ij} = \begin{bmatrix} \infty & 3 & 1 & 2 \\ 3 & \infty & 5 & 4 \\ 1 & 5 & \infty & 2 \\ 2 & 4 & 2 & \infty \end{bmatrix}$$

- **假设蚂蚁种群的规模m=3,参数α=1，β=2，ρ=0.5。**

❖解：

步骤1：初始化。首先使用贪心算法得到路径($ACDBA$)，则$C^{nn}=f(ACDBA)=1+2+4+3=10$。求得$\tau_0=m/C^{nn}=3/10=0.3$。初始化所有边上的信息素$\tau_{ij}=\tau_0$。

步骤2.1：为每只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。

步骤2.2：为每只蚂蚁选择下城市。我们仅以蚂蚁1为例，当前城市i=A,可访问城市集合$J_1(i)=\{\mathrm{B, C, D}\}$。计算蚂蚁1选择B,C,D作为下一访问城市的概率：

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

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

用轮盘赌法则选择下城市。假设产生的随机数q=random(0,1)=0.05，则蚂蚁1将会选择城市B。

用同样的方法为蚂蚁2和3选择下一访问城市，假设蚂蚁2选择城市D，蚂蚁3选择城市A。

步骤2.3：当前蚂蚁1所在城市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选择C,D作为下一城市的概率：

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

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

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

用轮盘赌法则选择下城市。假设产生的随机数q=random(0,1)=0.67，则蚂蚁1将会选择城市D。

用同样的方法为蚂蚁2和3选择下一访问城市，假设蚂蚁2选择城市C，蚂蚁3选择城市C。

步骤2.4：实际上此时路径已经构造完毕，蚂蚁1构建的路径为($ABDCA$)。蚂蚁2构建的路径为($BDCAB$)。蚂蚁3构建的路径为($DACBD$)。

步骤3：信息素更新。

蚂蚁1：ABDCA
蚂蚁2：BDCAB
蚂蚁3：DACBD

计算每只蚂蚁构建的路径长度：C_1=3+4+2+1=10，C_2=4+2+1+3=10，C_3=2+1+5+4=12。更新每条边上的信息素：

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

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

……

蚁密？
蚁量？
蚁周？

τ_{AB}与τ_{BA}可以是不同的。这一点与实际蚂蚁寻径有差异

步骤4：

如果满足结束条件，则输出全局最优结果并结束程序，否则，转向步骤2.1继续执行。

机器人最短路径规划的蚁群算法求解

- 问题建模为图结构寻找最短路径问题，0-1矩阵 $A=[a_{ij}]_{M\times N}$ 表示大小为M*N的地形结构

- 信息素的表达：信息素矩阵 $T(t)=[\tau_{ij}(t)]_{MN\times MN}$

- 启发式信息 $\eta_{ij}, i=1,2,\cdots,M; i=1,2,\cdots,N$ 的设计：

$$\eta_{ij}=\begin{cases}\dfrac{1}{\sqrt{(i-i_g)^2+(j-j_g)^2}}, & a_{ij}=0\\ 0, & a_{ij}=1\end{cases}$$

- 禁忌表的构建：当前像素8邻域

一个数集选择问题的蚁群算法求解（选讲）

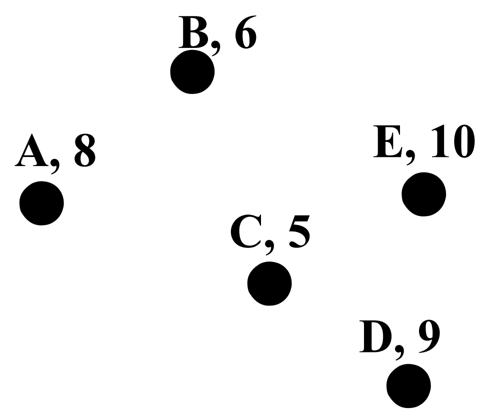
问题：从给定的N个互不相同的自然数中，选出k个数，使得这k个数的和与一个给定的数M之差的绝对值最小。

思考：

1）如何模型化为ACO算法能应用的形式

2）信息素矩阵如何构建，如何初始化，如何更新？

3）*启发式信息如何设计？



蚁群算法特征

蚁群算法构成一个系统

- S被称为系统，若满足：
  1）S中至少包含2个元素；
  2）S中的元素按照一定的方式互相联系；
  3）整体性能>∑各部分功能

- 蚁群具有：多元性，相关性，整体性；构成系统。

- 加和性：∑各部分功能=最终功能，非系统性

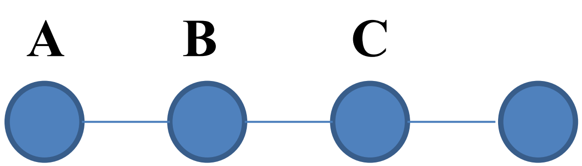
例：无约束优化问题的牛顿法，具有加和性，不能看成是一个系统

蚁群算法特征

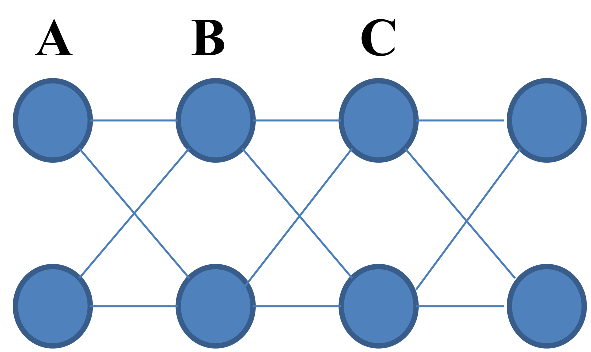
蚁群算法是一个分布式多智能体系统

- 生命系统是分布式的

- 在问题空间的多点同时独立地进行解搜索，整个问题的求解不会因为某只蚂蚁的失误而受到影响，具有较强的全局搜索能力，也增加了算法的可靠性。



非分布式行为



分布式行为

蚁群算法特征

蚁群算法具有自组织特征

- 自组织：一个系统的组织力或指令源于系统内部，也即系统在获得时间、空间或者功能的结构过程中无外界的特定干预；
- 蚁群算法、遗传算法、粒子群算法、后续的人工蜂群算法都具有自组织性
- 自组织即在无外界作用下使得系统熵减少的过程（即系统从无序到有序的进化过程）
- 自组织性增强了算法的稳健性和应用的普适性，不要求对应用问题有全面清晰的了解

蚁群算法特征

蚁群算法具有反馈机制

- 反馈是指将系统现在的行为作为影响系统未来行为的因素；正反馈：以现在的行为去加强未来的行为；负反馈：以现在的行为去削弱未来的行为；

- 蚁群算法在较优解路径上留下更多信息素，吸引后续更多蚂蚁，引导整个系统向最优解进化，属于正反馈过程；

- 蚁群隐含的负反馈：搜索过程的概率策略，是有可能接受退化解的

- 正反馈求精(exploitation)，保证算法向着最优解方向前进；负反馈保持对搜索范围的探索(exploration)，避免算法早熟。

蚁群算法总结

- 分布式优化方法, 不仅适合串行计算机, 而且适合未来的并行计算机

- 全局优化方法, 不仅可用于求解单目标优化问题, 而且可用于求解多目标优化问题

- 启发式算法, 计算复杂性为$O(Nc*n^2*N)$, 其中Nc 是迭代次数, N是蚂蚁数目, n 是节点数目

- 应用于组合优化问题中，在图着色、车间流、车辆调度、机器人路径规划、路由算法设计问题等领域均取得了良好的效果

蚁群算法待研究的问题

- 如何将现实的任务转换成可用蚁群求解的问题空间，并用适当的方式表达。如何定义“人工蚂蚁”以及蚂蚁间的非直接通信方式（如路径上的信息素、对象的分布状态等）的选择。

- 如何建立正反馈机制，定义启发函数，递增地进行问题求解，并且使得到的解与问题定义中现实世界的情况相对应。

蚁群算法待研究的问题

■大量的参数要初始化，其值对算法的性能有较大影响，但选取方法和原则目前尚无理论依据，只能实验调优，因此参数的最佳设置原则有待研究。

■搜索时间较长，如何将蚁群算法与遗传算法、免疫算法等优化算法相结合，改善和提高算法性能，以适应海量数据库的知识发现

■ 多头绒泡菌进行路网优化（扩展阅读：日本北海道大学的研究人员将一些燕麦片放在一个潮湿的表面上，其放置的各个点相当于东京周围的各个城市，并让多头绒泡菌从中心向外生长。观察到该粘菌进行自我组织、向外扩散并形成一种网络，该网络在功效、可靠性以及成本上都堪比真实世界的东京铁路网。）