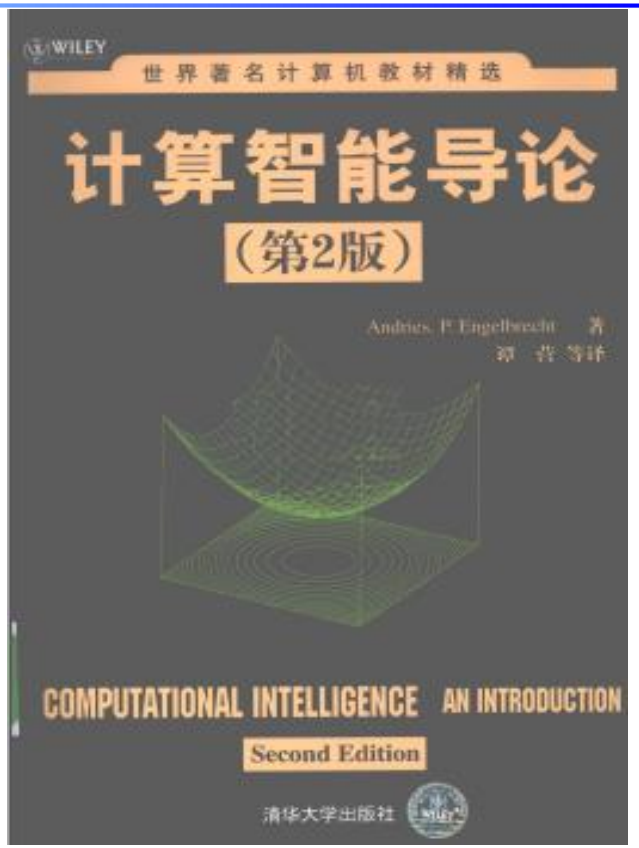
大数据挖掘与统计学习

软件工程系
文化遗产数字化国家地方工程联合中心
可视化技术研究所
张海波
讲师/博士(后)

计算智能（Computational Intelligence）

- 定义：目前还没有一个统一的的定义，计算智能是在**神经网络（Neural Networks,NN）、演化计算（Evolutionary Computation,EC）及模糊系统（Fuzzy System,FS）这3个领域**发展相对成熟的基础上形成的一个统一的学科概念。

- 计算智能是以生物进化的观点认识和模拟智能。按照这一观点，智能是在生物的遗传、变异、生长以及外部环境的自然选择中产生的。在用进废退、优胜劣汰的过程中，适应度高的（头脑）结构被保存下来，智能水平也随之提高。因此说计算智能就是基于结构演化的智能。



AI
- 计算智能
- 符号智能

现代智能优化算法

基于“从大自然中获取智慧”的理念，通过人们对自然界独特规律的认知，提取出适合获取知识的一套计算工具。总的来说，通过自适应学习的特性，这些算法达到了全局优化的目的。

此外：新近提出的鱼群算法、布谷鸟算法、狼群算法、头脑风暴算法、多目标进化算法、蜜蜂算法等层出不穷。

模拟自然相关现象

↓

构造统计量

↓

确定目标函数

↓

求解目标函数的极值（解）

计算智能的产生与发展

- 1992年，贝慈德克在《Approximate Reasoning》学报上首次 提出了“计算智能”的概念。

- 1994年6月底到7月初，IEEE在美国佛罗里达州的奥兰多市召开了首届国际计算智能大会(简称WCCI’94)。会议第一次将神经网络、演化计算和模糊系统这三个领域合并在一起，形成了“计算智能”这个统一的学科范畴。

- 在此之后，WCCI大会就成了IEEE的一个系列性学术会议，每4年举办一次。1998年5月，在美国阿拉斯加州的安克雷奇市又召开了第2届计算智能国际会议WCCI’98。2002年5月，在美国州夏威夷州首府火奴鲁鲁市又召了第3届计算智能国际会议WCCI’02。2014.07.06在北京召开。此外，IEEE还出版了一些与计算智能有关的刊物。

- 目前，计算智能的发展得到了国内外众多的学术组织和研究机构的高度重视，并已成为智能科学技术一个重要的研究领域。

计算智能的研究方法

基础：模型、算法、实验

模型：

符号系统及其上的操作，是三元组(数据集，操作，规则)

CI研究对象是具有以下特征的数学模型

符合模型的严格定义而又非常具体

兼有生物学背景知识

描述某一智能行为

计算智能的研究方法

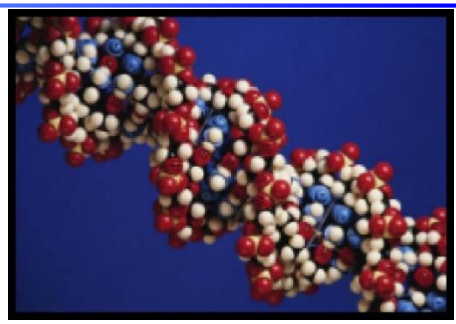
基础：模型、算法、实验

算法：

以计算理论、计算技术和计算工具研究对象模型的核心

特征：具有计算功能的算法，一般应具有数值构造性、迭代性、收敛性、稳定性和实效性

- 数值构造性：解是由数值量构造的
- 迭代性：计算公式上表现为递推，理论上表现为动力学性质，算法实现上表现为循环
- 收敛性：算法结束于稳定的结果上
- 稳定性：初始误差在迭代过程中可以得到控制
- 实效性：在有限的存储空间和有效的运算时间内得到有意义的计算结论

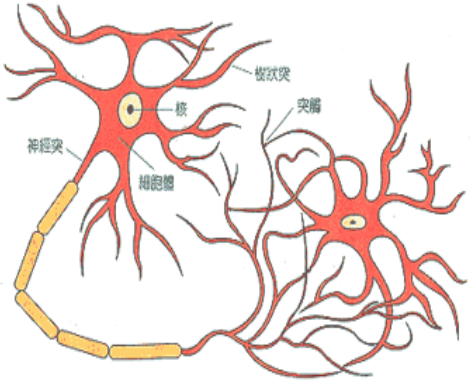


进化计算（Evolutionary Computation,EC）

GA（Genetic Algorithm）为代表

群智能(粒子群优化方法，蚁群算法)

神经网络计算

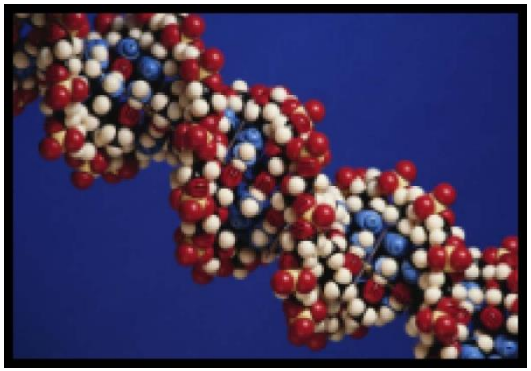


模糊计算

遗传算法（Genetic Algorithm）

- 遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

- 最初由美国Michigan大学J.Holland教授于1975年首先提出来，并出版了颇有影响的专著《Adaptation in Natural and Artificial Systems》，GA这个名称才逐渐为人所知，J.Holland教授所提出的GA通常为简单遗传算法（SGA）。



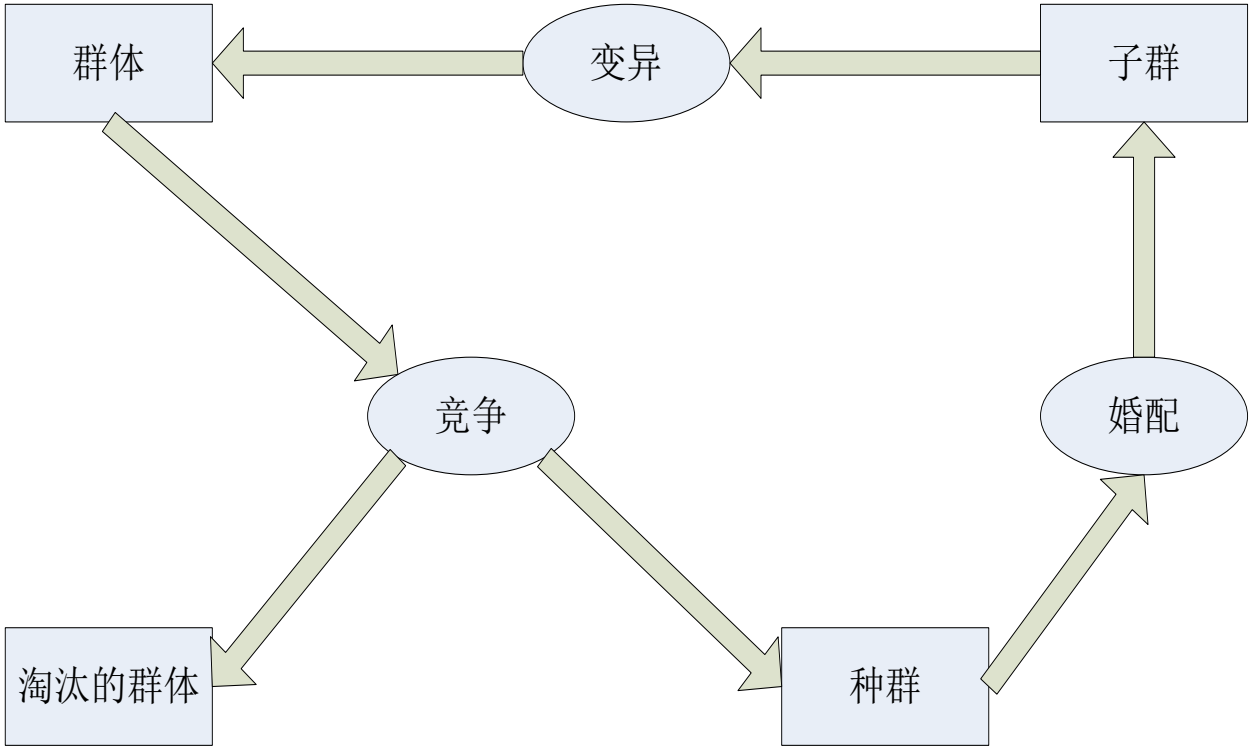
●生物进化理论和遗传学基本知识

达尔文的自然选择说

遗传（**heredity**）：
子代和父代具有相同或相似的性状，保证物种的稳定性

变异（**variation**）：
子代与父代,子代不同个体之间总有差异，是生命多样性的根源

生存斗争和适者生存：
具有适应性变异的个体被保留，不具适应性变异的个体被淘汰



生物进化循环图

生物进化理论和遗传学基本知识

(2) 遗传学基本概念和术语

个体		染色体
9	----	1001

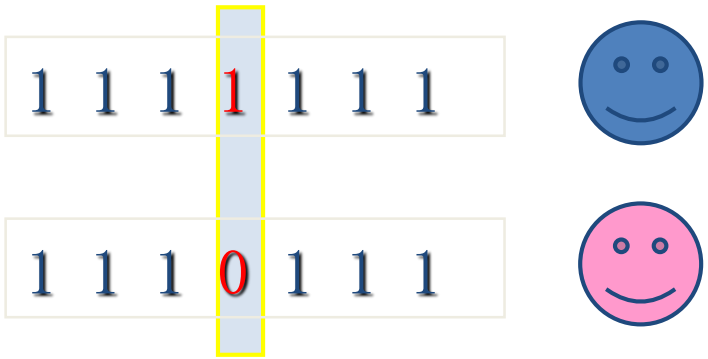
基因(gene)：染色体的一个片段

染色体(Chromosome)：问题中个体的某种字符串形式的编码表示

种群(Population)：个体的集合,该集合内的个体数称为种群的大小

基因型(genetype)：基因组合的模型，染色体的内部表现

表现型(phenotype)：染色体决定性状的外部表现



生物进化理论和遗传学基本知识

(2) 遗传学基本概念和术语

进化(evolution)：个体逐渐适应生存环境，不断改良品质的过程

适应度(fitness)：反映个体性能的一个数量值

适应度函数(fitness function)：

问题中的全体个体与其适应度之间的一个对应关系,一般是一个实值函数。该函数就是遗传算法中指导搜索的评价函数。

编码(coding)：从表现型到基因型的映射

解码(decoding)：从基因型到表现性的映射

算法相关基本概念

1. 个体与种群

- 个体就是模拟生物个体而对问题中的对象（一般就是问题的解）的一种称呼，一个个体也就是搜索空间中的一个点。
- 种群(population)就是模拟生物种群而由若干个体组成的群体，它一般是整个搜索空间的一个很小的子集。

2. 适应度与适应度函数

- 适应度(fitness)就是借鉴生物个体对环境的适应程度,而对问题中的个体对象所设计的表征其优劣的一种测度。
- 适应度函数(fitness function)就是问题中的全体个体与其适应度之间的一个对应关系。它一般是一个实值函数。该函数就是遗传算法中指导搜索的评价函数。

3. 染色体与基因

染色体（chromosome）就是问题中个体的某种字符串形式的编码表示。字符串中的字符也就称为基因（gene）。

例如：

个体		染色体
9	----	1001
（2，5，6）	----	010 101 110

4. 遗传操作

亦称遗传算子(genetic operator)，就是关于染色体的运算。遗传算法中有三种遗传操作:

- 选择-复制(selection-reproduction)
- 交叉(crossover，亦称交换、交配或杂交)
- 变异(mutation，亦称突变)

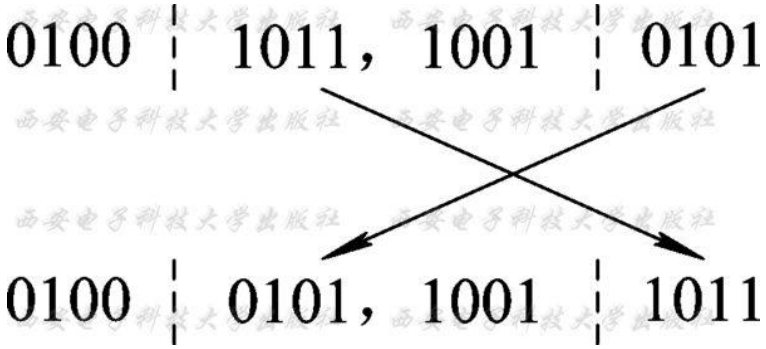
选择-复制 通常做法是：对于一个规模为N的种群S,按每个染色体$x_i \in S$的选择概率$P(x_i)$所决定的选中机会，分N次从S中随机选定N个染色体，并进行复制。

这里的选择概率$P(x_i)$的计算公式为

$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

交叉 就是互换两个染色体某些位上的基因。

例如，设染色体 s_1=01001011，s_2=10010101，交换其后4位基因，即



s_1'=01000101，　s_2'=10011011

可以看做是原染色体s_1和s_2的子代染色体。

变异 就是改变染色体某个(些)位上的基因。

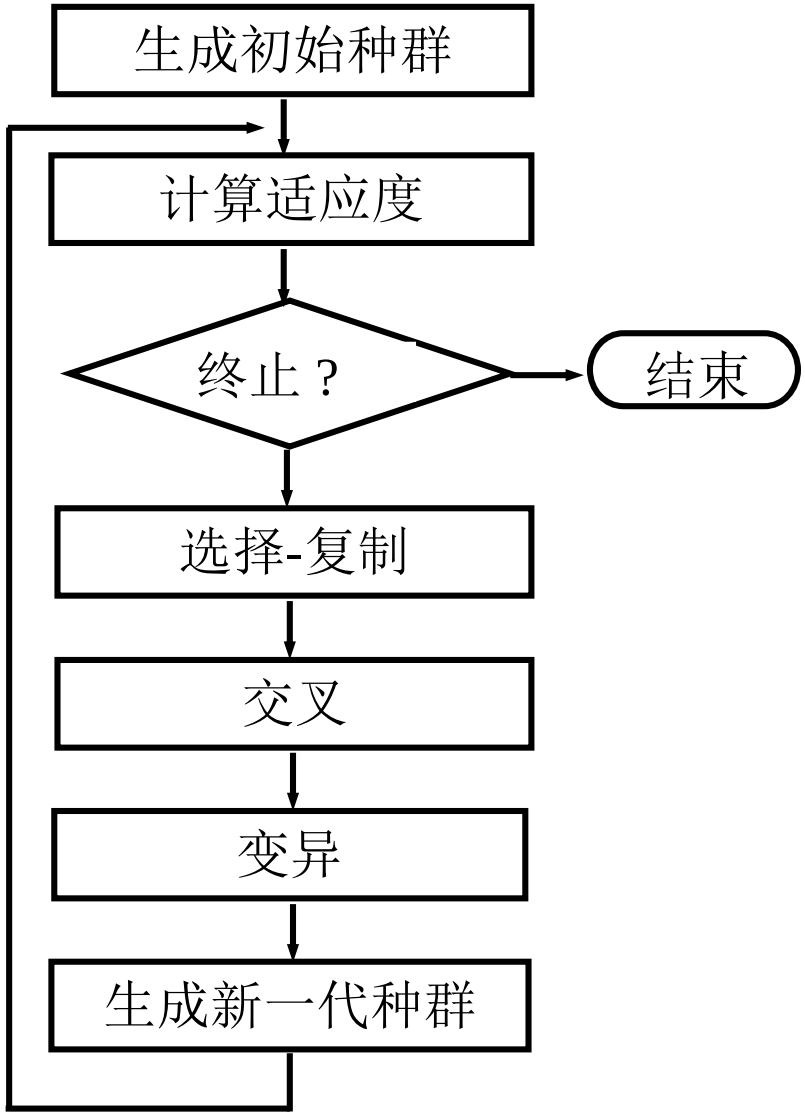
例如，设染色体 s=11001101

将其第三位上的0变为1, 即

s=11<u>0</u>01101 →11<u>1</u>01101= s'。

s'也可以看做是原染色体s的子代染色体。

1.2 基本遗传算法



遗传算法基本流程框图

算法中的一些控制参数：

- 种群规模
- 最大换代数
- 交叉率(crossover rate)就是参加交叉运算的染色体个数占全体染色体总数的比例，记为P_c，取值范围一般为0.4～0.99。
- 变异率(mutation rate)是指发生变异的基因位数所占全体染色体的基因总位数的比例，记为P_m，取值范围一般为0.0001～0.1。

基本遗传算法

步1 在搜索空间U上定义一个适应度函数$f(x)$，给定种群规模N，交叉率P_c和变异率P_m，代数T；

步2 随机产生U中的N个个体s_1，s_2，…，s_N，组成初始种群$S=\{s_1, s_2, \ldots, s_N\}$，置代数计数器$t=1$；

步3 计算S中每个个体的适应度f()；

步4 若终止条件满足，则取S中适应度最大的个体作为所求结果，算法结束。

步5 按选择概率$P(x_i)$所决定的选中机会，每次从S中随机选定1个个体并将其染色体复制，共做N次，然后将复制所得的N个染色体组成群体S_1；

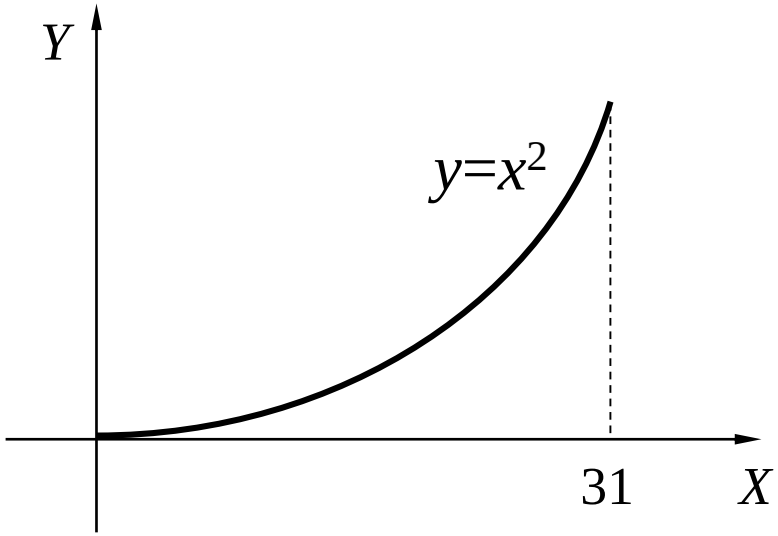
步6 按交叉率P_c所决定的参加交叉的染色体数c，从S_1中随机确定c个染色体，配对进行交叉操作，并用产生的新染色体代替原染色体，得群体S_2；

步7 按变异率P_m所决定的变异次数m，从S_2中随机确定m个染色体，分别进行变异操作，并用产生的新染色体代替原染色体，得群体S_3；

步8 将群体S_3作为新一代种群，即用S_3代替S，$t = t+1$，转步3；

1.3 遗传算法应用举例

例4.1 利用遗传算法求解区间［0,31］上的二次函数$y=x^2$的最大值。



分析

原问题可转化为在区间［0, 31］中搜索能使y取最大值的点a的问题。那么，［0, 31］中的点x就是个体, 函数值$f(x)$恰好就可以作为x的适应度，区间［0, 31］就是一个(解)空间 。这样, 只要能给出个体x的适当染色体编码, 该问题就可以用遗传算法来解决。

解

(1) 设定种群规模, 编码染色体，产生初始种群。

将种群规模设定为4；用5位二进制数编码染色体；取下列个体组成初始种群S_1:

s_1= 13 (01101), s_2= 24 (11000)

s_3= 8 (01000), s_4= 19 (10011)

(2) 定义适应度函数,

取适应度函数：$f(x)=x^2$

(3) 计算各代种群中的各个体的适应度，并对其染色体进行遗传操作,直到适应度最高的个体(即31（11111）)出现为止。

首先计算种群S_1中各个体

s_1= 13(01101), s_2= 24(11000)

s_3= 8(01000), s_4= 19(10011)

的适应度$f(s_i)$。

容易求得

$$f(s_1) = f(13) = 13^2 = 169$$
$$f(s_2) = f(24) = 24^2 = 576$$
$$f(s_3) = f(8) = 8^2 = 64$$
$$f(s_4) = f(19) = 19^2 = 361$$

再计算种群S_1中各个体的选择概率。

选择概率的计算公式为

$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

由此可求得

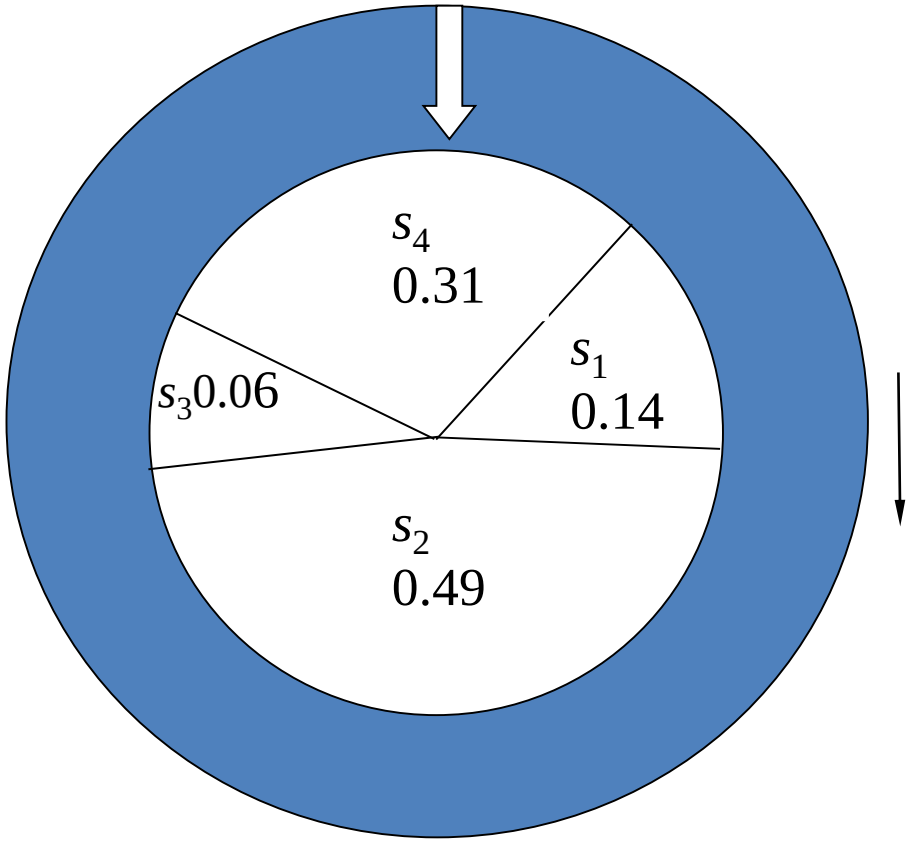
$$P(s_1)=P(13)=0.14$$

$$P(s_2)=P(24)=0.49$$

$$P(s_3)=P(8)=0.06$$

$$P(s_4)=P(19)=0.31$$

- 赌轮选择法



赌轮选择示意

在算法中赌轮选择法可用下面的子过程来模拟:

① 在［0, 1］区间内产生一个均匀分布的随机数r。

② 若$r \le q_1$,则染色体x_1被选中。 $q_i = \sum_{j=1}^{i} P(x_j)$

③ 若$q_{k-1} < r \le q_k (2 \le k \le N)$, 则染色体$x_k$被选中。 其中的$q_i$称为染色体$x_i$ (i=1, 2, …, n)的**积累概率**, 其计算公式为

选择-复制

设从区间［0, 1］中产生4个随机数如下:

$$r_1 = 0.450126, \quad r_2 = 0.110347$$

$$r_3 = 0.572496, \quad r_4 = 0.98503$$

染色体	适应度	选择概率	积累概率	选中次数
s_1=01101	169	0.14	0.14	1
s_2=11000	576	0.49	0.63	2
s_3=01000	64	0.06	0.69	0
s_4=10011	361	0.31	1.00	1

于是，经复制得群体：

s_1'=11000（24），s_2'=01101（13）

s_3'=11000（24），s_4'=10011（19）

交叉

设交叉率p_c=100%，即S_1中的全体染色体都参加交叉运算。

设s_1'与s_2'配对，s_3'与s_4'配对。分别交换后两位基因，得新染色体：

s_1''=11001（25），s_2''=01100（12）

s_3''=11011（27），s_4''=10000（16）

变异

设变异率p_m=0.001。

这样，群体S_1中共有

$$5\times4\times0.001=0.02$$

位基因可以变异。

0.02位显然不足1位，所以本轮遗传操作不做变异。

于是，得到第二代种群S_2：

s_1=11001（25），s_2=01100（12）

s_3=11011（27），s_4=10000（16）

第二代种群S_2中各染色体的情况

染色体	适应度	选择概率	积累概率	估计的选中次数
s_1=11001	625	0.36	0.36	1
s_2=01100	144	0.08	0.44	0
s_3=11011	729	0.41	0.85	2
s_4=10000	256	0.15	1.00	1

假设这一轮选择-复制操作中，种群S_2中的**4个染色体都被选中**，则得到群体：

s_1'=11001（25），s_2'= 01100（12）

s_3'=11011（27），s_4'= 10000（16）

做交叉运算，让s_1’与s_2’，s_3’与s_4’分别交换后三位基因，得

s_1''=11100（28）, s_2''= 01001（9）

s_3''=11000（24）, s_4''= 10011（19）

这一轮仍然不会发生变异。

于是，得第三代种群S3：

s_1=11100（28），s_2=01001（9）

s_3=11000（24），s_4=10011（19）

第三代种群S_3中各染色体的情况

染色体	适应度	选择概率	积累概率	估计的选中次数
s_1=11100	784	0.44	0.44	2
s_2=01001	81	0.04	0.48	0
s_3=11000	576	0.32	0.80	1
s_4=10011	361	0.20	1.00	1

设这一轮的选择-复制结果为：

s_1'=11100（28），s_2'=11100（28）

s_3'=11000（24），s_4'=10011（19）

做交叉运算，让s_1’与s_4’，s_2’与s_3’分别交换后两位基因，得

s_1''=11111（31），s_2''=11100（28）

s_3''=11000（24），s_4''=10000（16）

这一轮仍然不会发生变异。

于是，得第四代种群S_4：

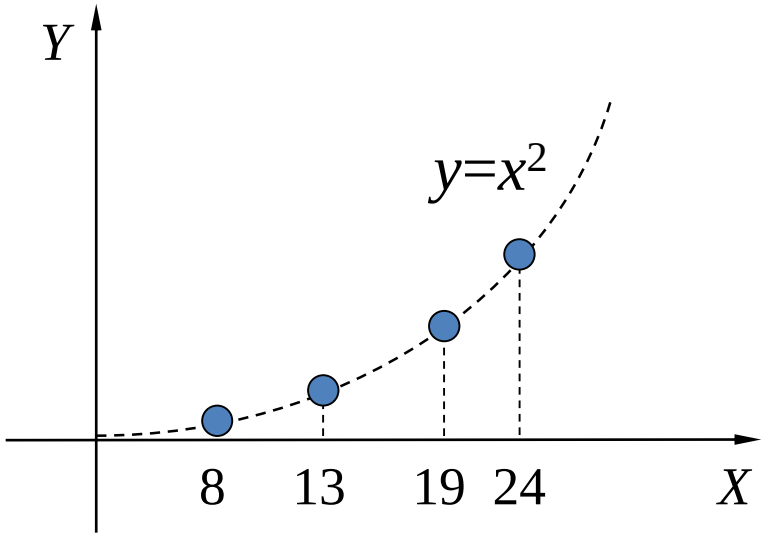
s_1=11111（31），s_2=11100（28）

s_3=11000（24），s_4=10000（16）

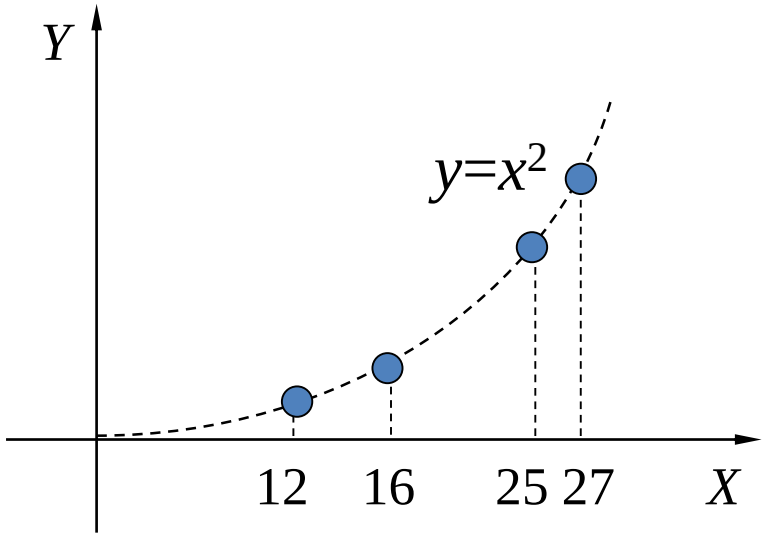
显然，在这一代种群中已经出现了适应度最高的染色体s_1=11111。于是，遗传操作终止，将染色体“11111”作为最终结果输出。

然后，将染色体“11111”解码为表现型，即得所求的最优解：31。

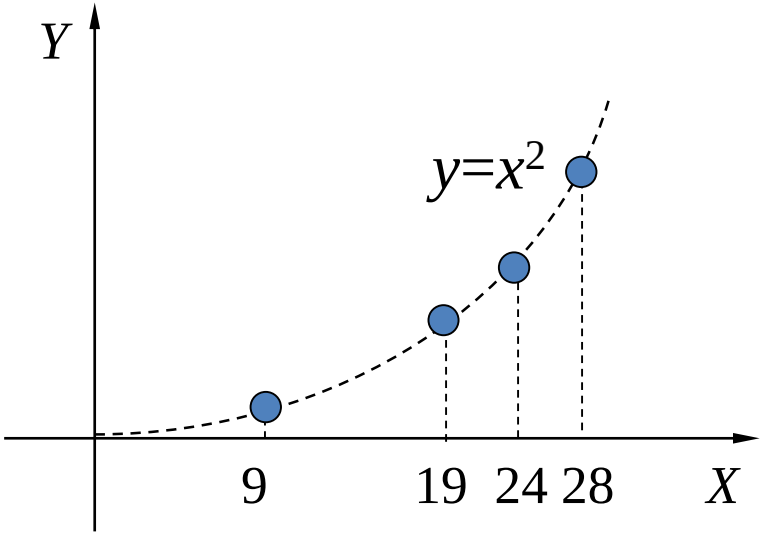
将31代入函数$y=x^2$中，即得原问题的解，即函数$y=x^2$的最大值为961。



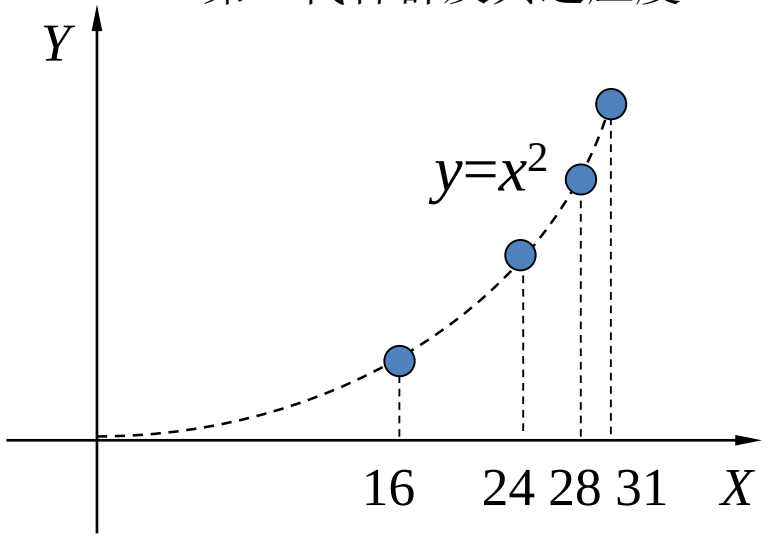
第一代种群及其适应度



第二代种群及其适应度



第三代种群及其适应度



第四代种群及其适应度

遗传算法的本质

遗传算法本质上是对染色体模式所进行的一系列运算，即通过选择算子将当前种群中的优良模式遗传到下一代种群中，利用交叉算子进行模式重组，利用变异算子进行模式突变。通过这些遗传操作，模式逐步向较好的方向进化，最终得到问题的最优解。

- **群智能（Swarm Intelligence, SI）**

  人们把群居昆虫的集体行为称作“群智能”（“群体智能”、“群集智能”、“集群智能”等）



- **特点**

  个体的行为很简单，但当它们一起协同工作时，却能够突现出非常复杂（智能）的行为特征。

- **描述**

  群智能作为一种新兴的演化计算技术已成为研究焦点，它与人工生命，特别是进化策略以及遗传算法有着极为特殊的关系。

- **特性**

  指无智能的主体通过合作表现出智能行为的特性，在没有集中控制且不提供全局模型的前提下，为寻找复杂的分布式问题求解方案提供了基础。

- **优点**

  灵活性：群体可以适应随时变化的环境；

  稳健性：即使个体失败，整个群体仍能完成任务；

  自我组织：活动既不受中央控制，也不受局部监管。

- **典型算法**

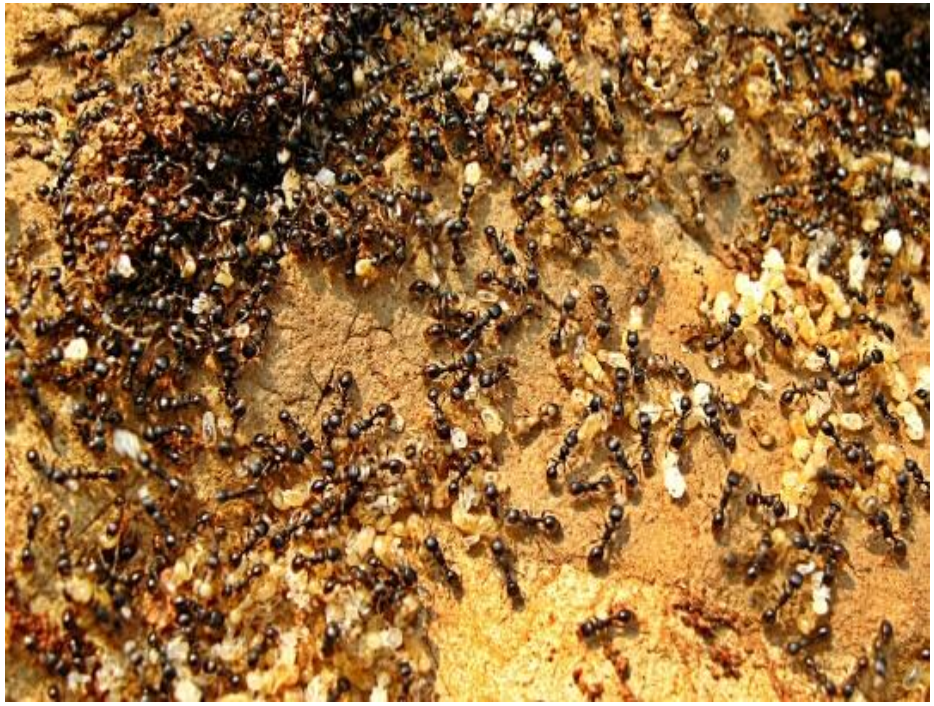
  蚁群算法（蚂蚁觅食）

  粒子群算法（鸟群捕食）

蚁群算法

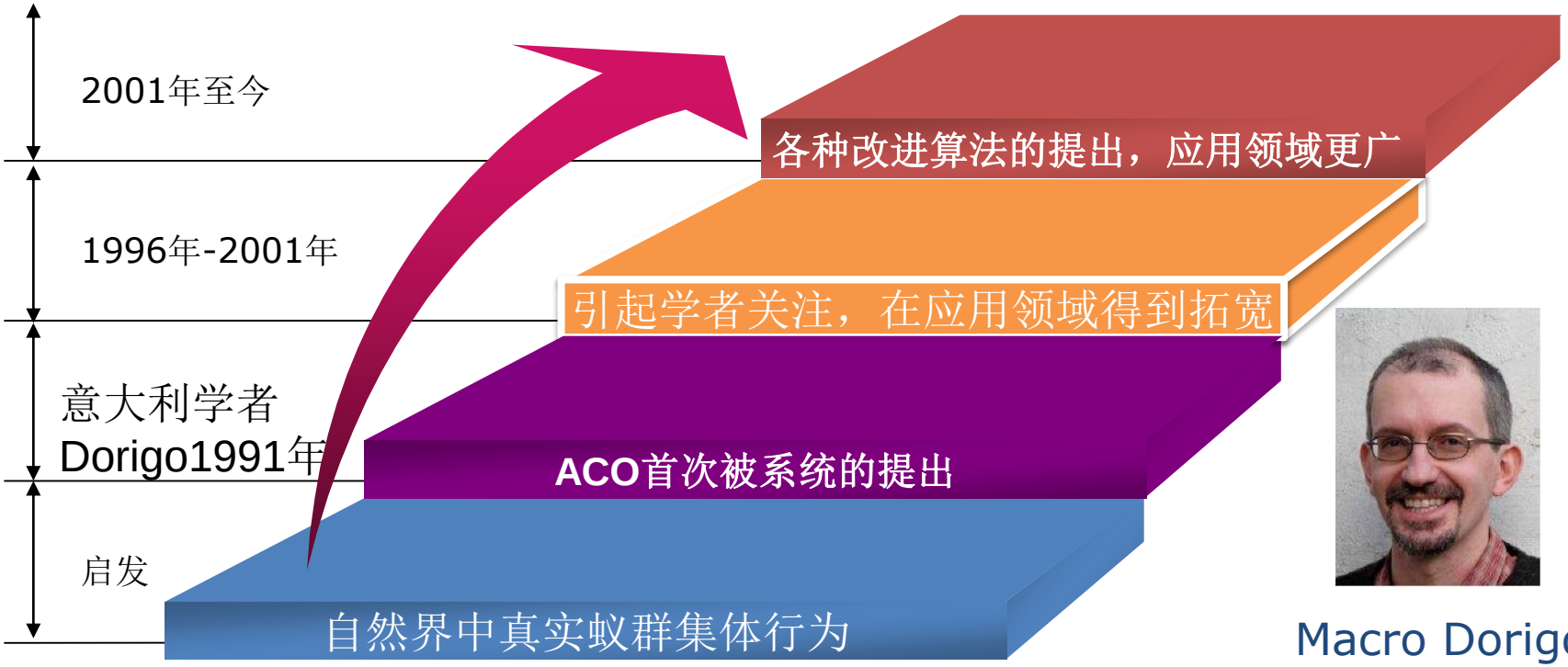
一、蚁群算法的基本原理

- 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。



一、蚁群算法的基本原理

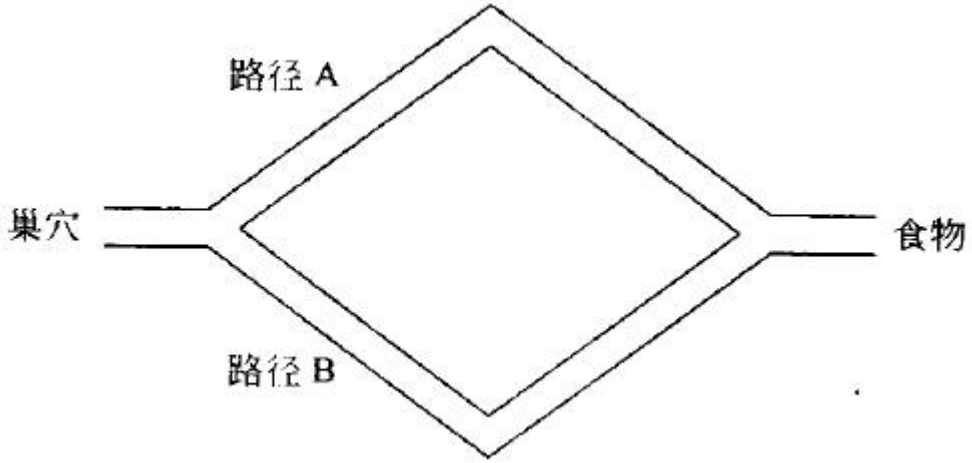
◆ 背景



Macro Dorigo
MiLanCollege

一、蚁群算法的基本原理

◆ 双桥实验：



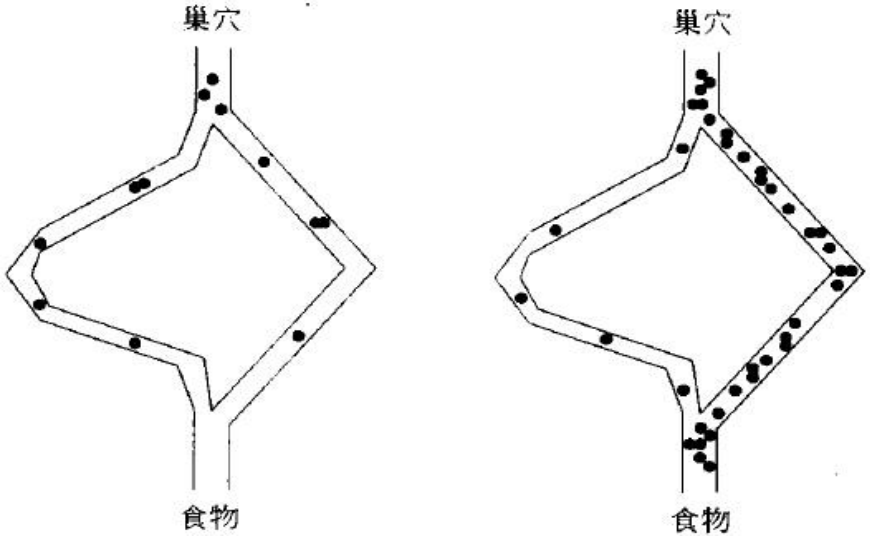
等长路径，开始时路径上无信息素，蚂蚁以等概率进行路径选择。
实验结果：总会有一个分支被绝大多数的蚂蚁选择。

蚁群的自组织性（神奇的自动化）

一、蚁群算法的基本原理

◆ 双桥实验的扩展实验：

不等长路径，开始时路径上无信息素，蚂蚁以等概率进行路径选择。



实验结果：短分支被绝大多数的蚂蚁选择。

信息素（food/path）

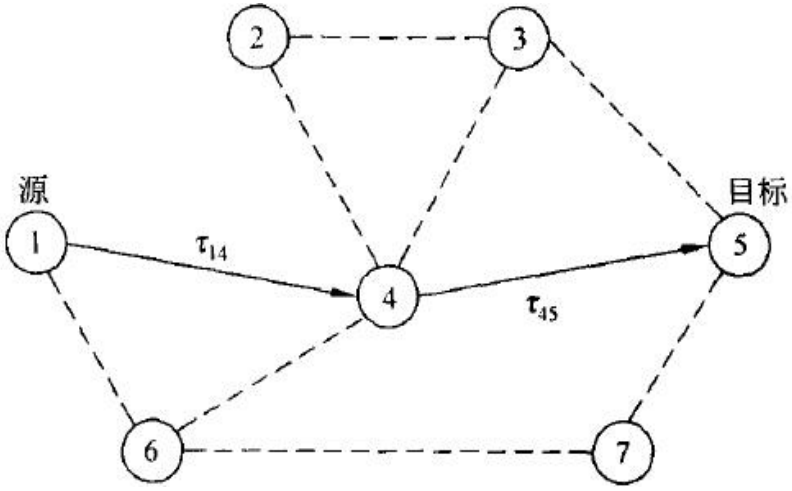
正反馈

一、蚁群算法的基本原理

◆ 蚂蚁寻找最短路径的分析与建模

以简单图为例：

G=(V,E),V为节点集合，E为边的集合；对于n_k只蚂蚁中的蚂蚁K，当前节点I,进行决策选择下一个节点j的概率为：



$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)}{\sum_{j\in J_i^k}\tau_{ij}^{\alpha}(t)} & \text{如果} j\in \mathcal{N}_i^k \\ 0 & \text{如果} j\notin \mathcal{N}_i^k\end{cases} \quad (1)$$

$\mathcal{N}_i^k$ 对于蚂蚁K指的是与节点i相连的可选节点集合（若为空集，先构成环/回路）；

α 为一个正的常亮，放大信息素的影响（经验设定，过大次优）

一、蚁群算法的基本原理

当所有蚂蚁到达目标节点，去环，每只蚂蚁按原路径返回源节点，并在沿途的每个边上（i,j）上释放一定量的信息素 $\Delta\tau_{ij}^{k}$：

$$\Delta\tau_{ij}^{k} \propto \frac{1}{L^{k}(t)} \qquad (2)$$

式中：$L^{k}(t)$（1/优良度）为蚂蚁K在第t步路径的长度：

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^{k}(t) \qquad (3)$$

$\Delta\tau_{ij}^{k}$ 正比于路径的优良度(不唯一，多种测度形式)；t+1时刻的信息素取决于（3）式。

考虑信息素的挥发因素，引入挥发机制，ρ 为归一化因子，即，ρ 控制之前已搜索过得路径的影响，显然与信息素挥发速率成正比（越大越随机，ρ =1，完全随机搜索）

（4）

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

一、蚁群算法的基本原理

◆ 简单图蚁群优化算法框架：

Step1:初始化信息素 $\tau_{ij}(0)$

t=0;

将K只蚂蚁置于源节点(nest)

Step2:为每只蚂蚁建立路径 $x^k(t)$,即：

for(k=1,…,n_k)

while($x^k(t)$!= ∅)

由（1）式计算下一节点的概率

更新 $x^k(t)$

if 到达目标节点

去环，并计算路径长度 $f(x^k(t))$

end

for(k=1,…,n_k)

由（4）式计算信息素的挥发，得到新的信息素 $\tau_{ij}(t)$

一、蚁群算法的基本原理

end

Step3: for(k=1,…,n_k)

for 每条边的路径长度 $x^k(t)$

由（3）式更新信息素；

end

end

Step4: t=t+1

Step5: if(True) //终止条件

Return Min($x^k(t)$)

else

continue

关于终止条件的选取：迭代次数n_t or 解得精度 or All ands in one path

对于简单蚁群算法优化性能的总结：

1、适于简单图（无环无重）；

2、对于较大的图或非简单图，优化性能差（不鲁棒，参数敏感）；

3、图越复杂，性能越依赖于参数ρ ，过大易收敛于次优路径；

4、参数 α ,适当最优，过大次优。

蚁群算法的主要改进

改进关键：探索—开发，如何平衡？

近年来一些学者的主要平衡策略：

选择后续节点时，引入额外的启发信息（避免环或非优点）；

不同的信息素的更新策略（全局/局部）；

蚂蚁数量；

多种群并行开发策略；

蚁群算法的主要改进

◆ 蚂蚁系统

针对简单蚁群优化算法做了两点改进：

1、增加启发式信息，改变转移概率；

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{u\in N_i^k(t)}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)} & \text{如果} j\in \mathcal{N}_i^k(t)\\ 0 & \text{如果} j\notin \mathcal{N}_i^k(t)\end{cases} \quad (5)$$

相对于（1）式，在保存先验信息（代表之前较优的移动）同时，增加了后验信息（即下一次移动的倾向性），从而很好的处理了探索—开发之间的关系：

$\alpha = 0$ ⟹ 算法演变为贪心搜索

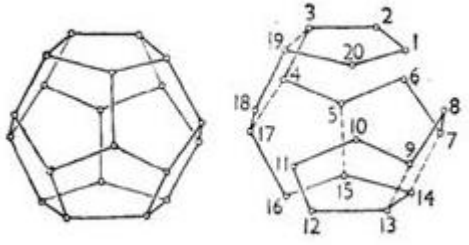
$\beta = 0$ ⟹ 算法演化为简单蚁群算法

对于 η_{ij}^{β} ，针对具体问题具体而定，比如对于寻找最短路径长度问题

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

2、式中的 $\mathcal{N}_i^k$ 同（**1**）式，此外为每只蚂蚁增加禁忌表，存放已访问的节点，禁忌表中的节点不会被包含在 $\mathcal{N}_i^k$ 中，对于求**Hamilton Problem**尤为重要,还有学者提出转移概率的另一种形式，减少了参数 β ：

$$p_{ij}^k(t)=\begin{cases}\dfrac{\alpha\tau_{ij}(t)+(1-\alpha)\eta_{ij}}{\sum_{u\in\mathcal{N}_i^k(t)}(\alpha\tau_{iu}(t)+(1-\alpha)\eta_{iu}(t))} & \text{如果} j\in\mathcal{N}_i^k(t)\\ 0 & \text{其他情况}\end{cases}$$



信息素的挥发同（4）式，即：

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

信息素的更新采用下式：

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (6)$$

$\Delta\tau_{ij}^{k}(t)$表示蚂蚁K在边ij的释放的信息素，针对$\Delta\tau_{ij}^{k}(t)$，不同学者依据各自的理解提出了不同的形式，从而出现了蚂蚁系统的三种变种：

1、蚁周系统：

(7)

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{f(x^{k}(t))} & \text{如果边}(i,j)\text{处于路径}x^{k}(t) \\ 0 & \text{其他情况} \end{cases}$$

$\Delta\tau_{ij}^{k}(t)$，即信息素的释放量与路径质量 $f(x^{k}(t))$ 成反比，也就是说信息素的浓度是通过全局信息来更新，Q为正的常量（正向激励）。

2、蚁密系统：

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}Q & \text{如果边}(i,j)\text{处于路径}\mathbf{x}^{k}(t)\\ 0 & \text{其他情况}\end{cases}$$

信息素浓度通过蚂蚁数量来更新（每只蚂蚁在路径边上释放相同量的信息素）。

3、蚁量系统：

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{d_{ij}} & \text{如果边}(i,j)\text{处于路径}\mathbf{x}^{k}(t)\\ 0 & \text{其他情况}\end{cases}$$

d_{ij} 为边ij的一种测度，通常定义为长度。

此外，对于蚂蚁系统的初始化阶段，

1、k只蚂蚁的放置：

针对源节点到目标的最短路径；

→ nest；

针对Hamilton Problem；

→ 随机放置到图中；

2、信息素的初始化：

同简单群算法。

◆ 蚁群系统

学者针对蚂蚁系统进一步做了如下改进：1、不同的转移规则；2、不同的信息素更新规则；3、局部信息素更新规则；4、引进候选列表给予某些特别节点更多的倾向性。

具体来讲：

1、对于j点的选取（分段思想）：

$$j=\begin{cases}\arg\max_{u\in \mathcal{S}_i^k(t)}\{\tau_{iu}(t)\eta_{iu}^{\beta}(t)\} & \text{如果} r\leqslant r_0\\ J & \text{如果} r>r\end{cases} \qquad (8)$$

j点的分段策略便得转移规则倾向选择短边的点或是信息素浓度大的点，通过设置合适的 γ_o 来平衡探索—开发，即：

$\gamma \leq \gamma_o$ ⟹ 开发；

$\gamma > \gamma_o$ ⟹ 探索。

J的取值：

$$p_{ij}^k(t)=\frac{\tau_{ij}(t)\eta_{ij}^{\beta}(t)}{\sum_{u\in \mathcal{S}_i^k}\tau_{iu}(t)\eta_{iu}^{\beta}(t)}$$

2、与蚂蚁系统不同，信息素的更新不再取决于所有蚂蚁，而是取决于全局最优蚂蚁（最短路径）：

$$\tau_{ij}(t+1)=(1-\rho)_1\tau_{ij}(t)+\rho_1\Delta\tau_{ij}(t)$$

其中：

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{1}{f(x^+(t))} & \text{如果}(i,j)\in x^+(t)\\ 0 & \text{其他情况}\end{cases}$$

对于最短路径问题：

$$f(x^+(t))=|x^+(t)|$$

通过信息素的全局更新策略，使得蚁群总是集中在当前的最优解周围进行搜索;此外，挥发因子设置（静/动）也至关重要，越大，最优路径的影响越大，同时之前的影响也被忽略，更加倾向于探索。

3、此外，除了全局更新规则外，还有学者提出了局部更新规则：

$$\tau_{ij}(t) = (1-\rho_2)\tau_{ij}(t) + \rho_2\tau_0$$

其中τ_0的选取为一个小的正常量，对于Hamilton Problem，$\tau_0 = (n_G L)^{-1}$；其中L为最近启发信息生成的长度。

4、增加一个候选节点列表，即，将靠近i的n个节点按距离升序排列，优先选择距离最近的节点，若候选节点为空，则依靠转移概率从非候选节点列表中选择离i最近的节点。

◆ 最大最小蚂蚁系统：

尽管学者们对于蚁群算法的弱点做了很大的改进，提出了著名的上述两个系统，但是他们在解决复杂的优化问题时，系统会出现早熟停滞的现象，停滞是指所有蚂蚁沿着同一路经（比如当蚂蚁太少并且迅速开发最高信息素浓度路径的时候），进而有学者针对此问题提出了最大最小蚂蚁系统。

4、一直在改进，从未被间断：(1)转移概率；（2）信息素的更新规则；（3）避免停滞的。

◆ 参数设置的总结：

η_k 的选取：探索能力与迭代时间（算法复杂度）的矛盾；

η_t 的选取：非最优解与计算复杂度的矛盾；

τ_0 的设置：0/正数（改变倾向性）。

蚁群的其他社会行为

生物学家经过长期的观察发现，在蚂蚁群体中存在一种本能的聚集行为：

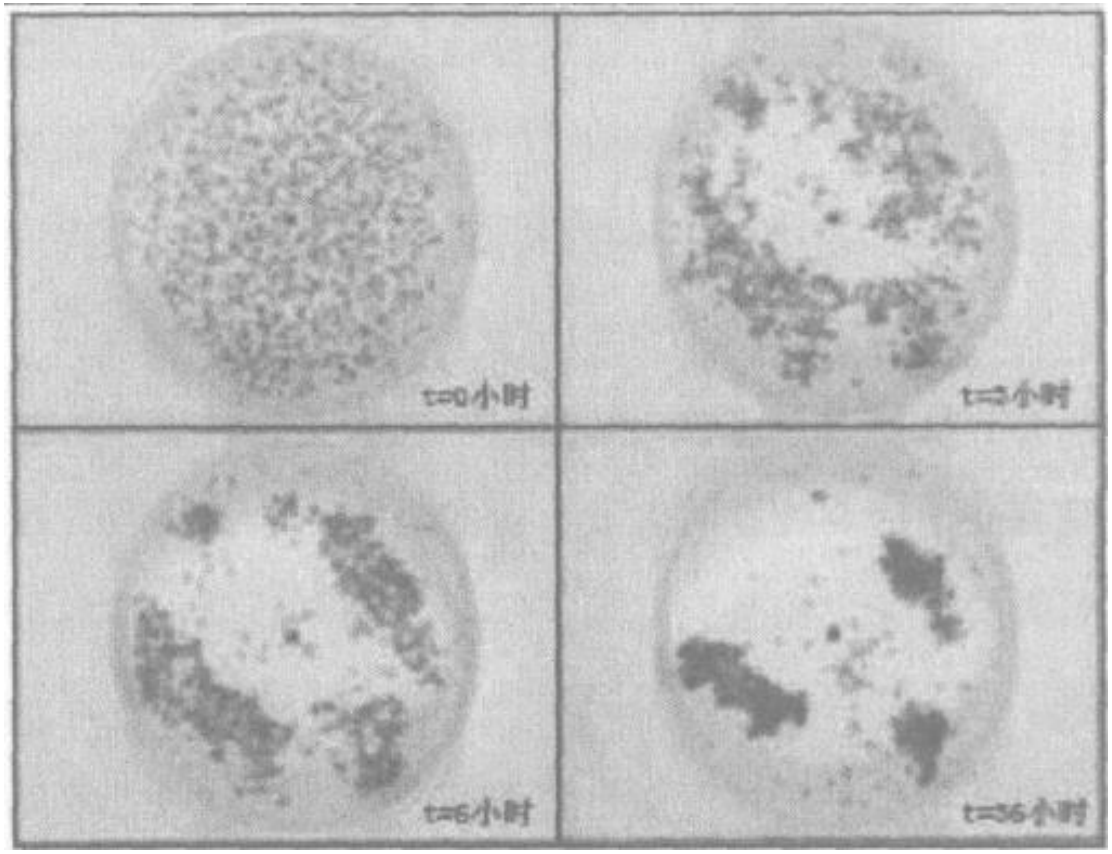
1、蚂蚁往往能在没有关于蚂蚁整体的任何指导性信息情况下，将其死去的同伴的尸体安放在一个固定的场所,表面上看每只蚂蚁是单独行动，捡拾或遗弃尸体时在空间中随机行动，但进一步观察发现，最终蚂蚁都会将尸体等等聚集成堆放到一定的地点。

2、对于幼虫，生物学家也发现了类似的聚类现象，幼虫按大小（年龄）摆放的位置呈现出一种圆周现象。

蚁群除了觅食的群体行为外，还有一种本能的聚类行为。

国外学者在十几年前针对蚂蚁的聚类行为做了相关的验证试验（幼体的摆放）

→ 自组织性



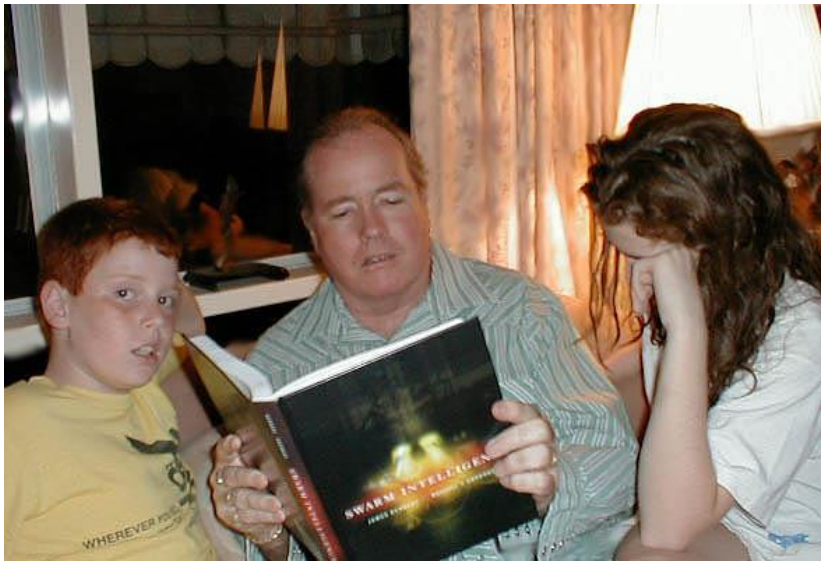
粒子群算法

由Kennedy和Eberhart于1995年提出.

群体迭代，粒子在解空间追随最优的粒子进行搜索.

粒子群算法：
- 简单易行
- 收敛速度快
- 设置参数少

已成为现代优化方法领域研究的热点.



粒子群算法的思想源于对鸟群捕食行为的研究.
模拟鸟集群飞行觅食的行为，鸟之间通过集体的协作使群体达到最优目的，是一种基于Swarm Intelligence的优化方法。

马良教授在他的著作《蚁群优化算法》一书的前言中写到：

“自然界的蚁群、鸟群、鱼群、
羊群、牛群、蜂群等，其实时时刻刻都在给予
我们以某种启示，只不过我们常常忽略了
大自然对我们的最大恩赐！.....”

粒子群算法的基本思想

设想这样一个场景：一群鸟在随机搜索食物

已知{在这块区域里只有一块食物；
所有的鸟都不知道食物在哪里；
但它们能感受到当前的位置离食物还有多远.

那么:找到食物的最优策略是什么呢？

搜寻目前离食物最近的鸟的周围区域．
根据自己飞行的经验判断食物的所在。

PSO正是从这种模型中得到了启发.

PSO的基础：信息的社会共享

- 生物学家对鸟(鱼)群捕食的行为研究
- 社会行为 (Social-Only Model)
- 个体认知 (Cognition-Only Model)



算法介绍

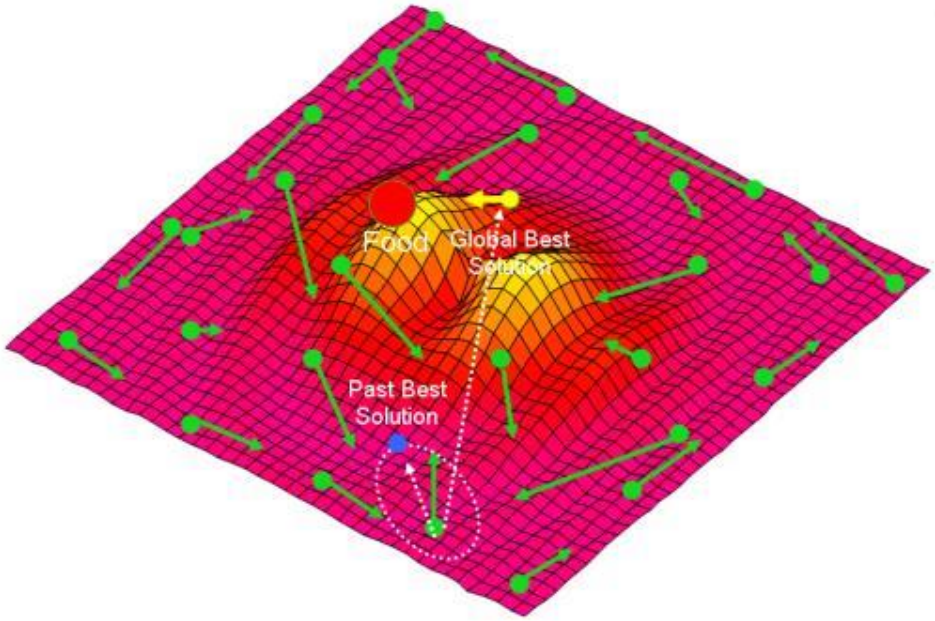
- 每个寻优的问题解都被想像成一只鸟，称为“粒子”。所有粒子都在一个D维空间进行搜索。
- 所有的粒子都由一个**fitness function** 确定适应值以判断目前的位置好坏。
- 每一个粒子必须赋予记忆功能，能记住所搜寻到的最佳位置。
- 每一个粒子还有一个速度以决定飞行的距离和方向。这个速度根据它本身的飞行经验以及同伴的飞行经验进行动态调整。

- **PSO算法**

  初始化为一群随机粒子，通过迭代找到最优。

  每次迭代中，粒子通过跟踪“个体极值（pbest）”和“全局极值(gbest)”来

  更新自己的位置。



- **粒子速度和位置的更新**

假设在*D*维搜索空间中，有*m*个粒子；

其中第*i*个粒子的位置为矢量 $\vec{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$

其飞翔速度也是一个矢量，记为 $\vec{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$

第*i*个粒子搜索到的最优位置为 $\vec{p}_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$

整个粒子群搜索到的最优位置为 $\vec{p}_{gbest} = (p_{gbest1}, p_{gbest2}, \cdots, p_{gbestD})$

第*i*个粒子的位置和速度更新为：

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

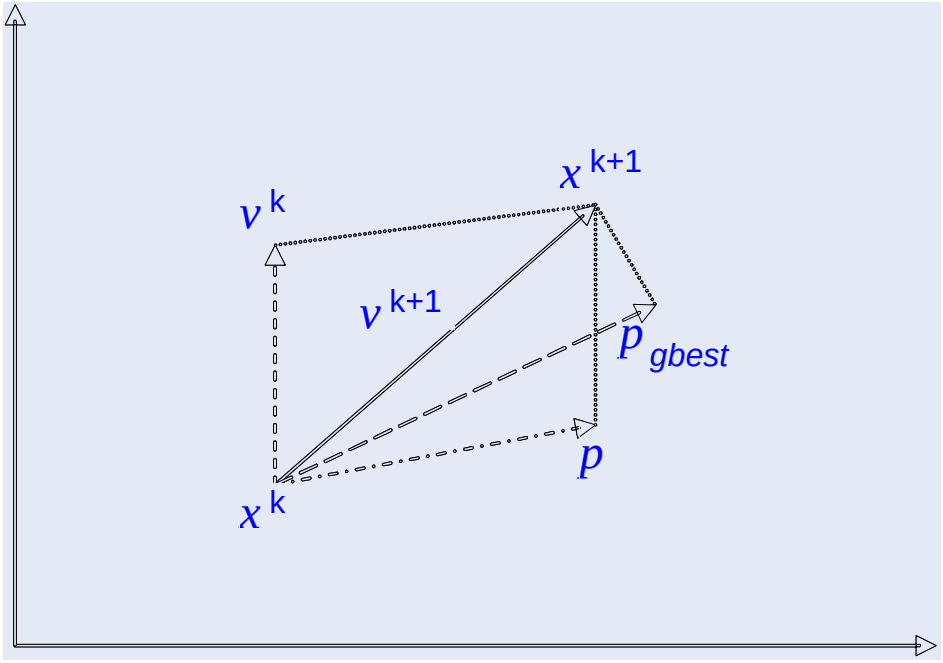
- **粒子速度和位置的更新**

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$
$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

其中，w称为惯性权重，

c_1和c_2为两个正常数，称为加速因子。

将 v_{id}^{k} 限制在一个最大速度 v_{max} 内。



- **粒子速度和位置的更新**

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

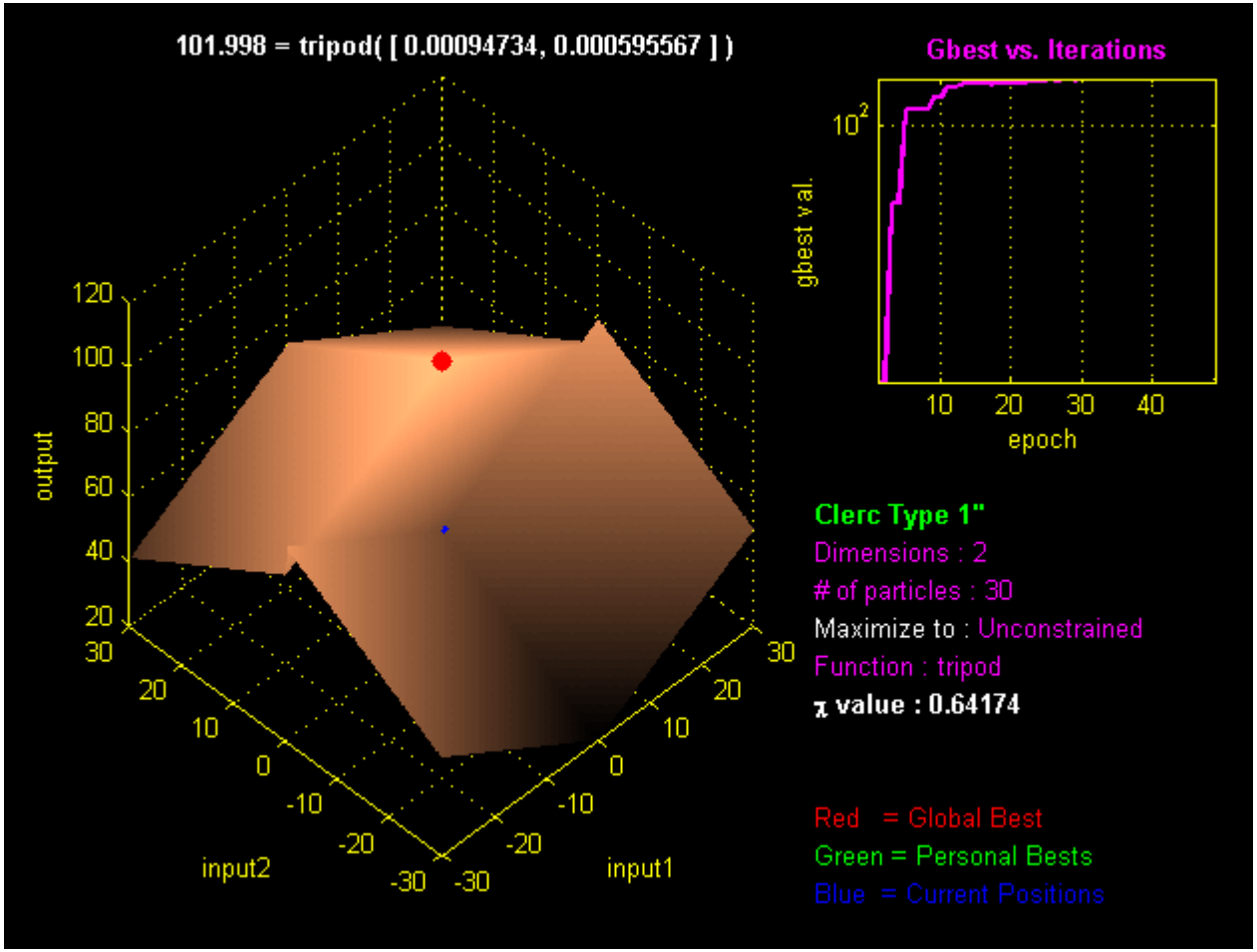
$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

“惯性部分”，对自身运动状态的信任

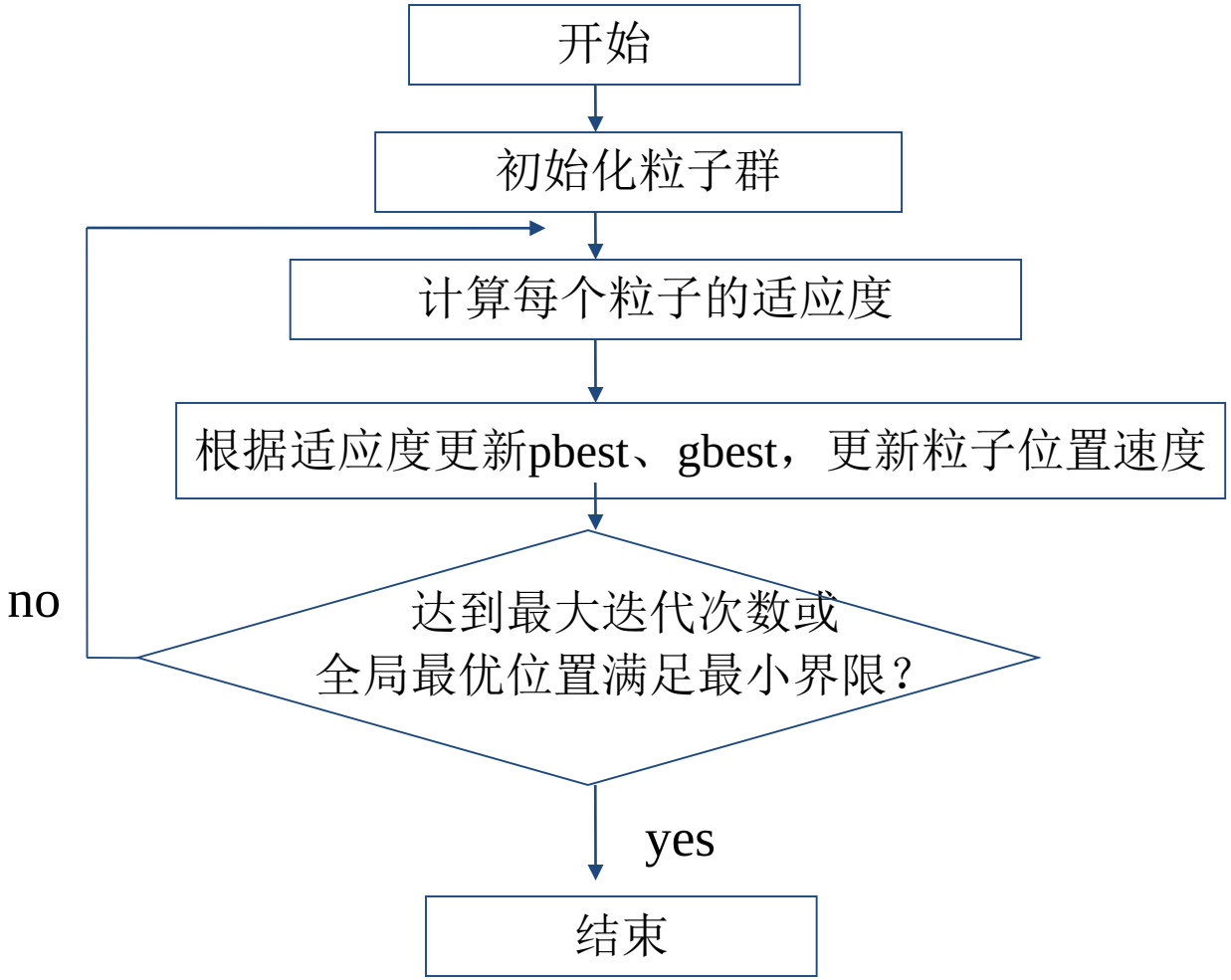
“认知部分”，对微粒本身的思考，即来源于自己经验的部分

“社会部分”，微粒间的信息共享，来源于群体中的其它优秀微粒的经验

- 迭代过程


101.998 = tripod([0.00094734, 0.000595567])
Gbest vs. Iterations
gbest val.
epoch
output
input1
input2
Clerc Type 1"
Dimensions : 2
# of particles : 30
Maximize to : Unconstrained
Function : tripod
χ value : 0.64174
Red = Global Best
Green = Personal Bests
Blue = Current Positions

粒子群优化算法流程图


开始
初始化粒子群
计算每个粒子的适应度
根据适应度更新pbest、gbest，更新粒子位置速度
达到最大迭代次数或
全局最优位置满足最小界限？
no
yes
结束

参数分析

- **惯性权重*w***

使粒子保持运动惯性，使其有扩展搜索空间的趋势，有能力探索新的区域。

表示微粒对当前自身运动状态的信任，依据自身的速度进行惯性运动。

较大的*w*有利于跳出局部极值，而较小的*w*有利于算法收敛。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

- **加速常数c_1和c_2**

  代表将微粒推向**pbest**和**gbest**位置的统计加速项的权重。

  表示粒子的动作来源于自己经验的部分和其它粒子经验的部分。

  低的值允许粒子在被拉回之前可以在目标区域外徘徊，而高的值则导致粒子突然冲向或越过目标区域。

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 rand()(p_{id} - x_{id}^{k}) + c_2 rand()(p_{gbest} - x_{id}^{k})$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad i = 1,2,\cdots,m;\ d = 1,2,\cdots,D$$

- **加速常数c_1和c_2**

  将c_1和c_2统一为一个控制参数，$\varphi = c_1 + c_2$

  如果φ很小，微粒群运动轨迹将非常缓慢；

  如果φ很大，则微粒位置变化非常快；

  实验表明，当φ=4.1（通常c_1=2.0，c_2=2.0）时，具有很好的收敛效果。

- **粒子数**

  一般取20～40，对较难或特定类别的问题可以取100～200。

- **最大速度v_{max}**

  决定粒子在一个循环中最大的移动距离，通常设定为粒子的范围宽度。

- **终止条件**

  最大循环数以及最小错误要求。