

人工智能概论

慕彩红  
人工智能学院  
mucaihongxd@foxmail.com

第七章 群智能算法  
——粒子群算法

群智能概述

- 群智能（Swarm Intelligence, SI）  
对于群居昆虫，如蚂蚁、蜜蜂、鱼群、鸟群等，个体在结构上是很简单的，而它们的集体行为却可能变得相当复杂。  
人们把群居昆虫的集体行为称作“群智能”，即低智能的主体通过合作表现出高智能行为的特性。  
群智能算法是一种基于生物群体行为规律的计算技术。

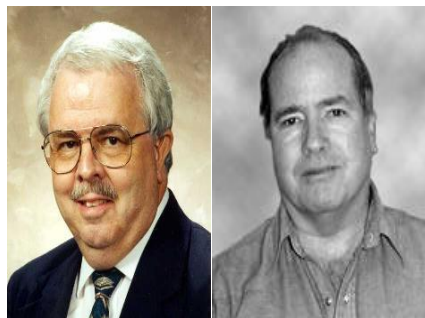
- 特点
  1. 个体的行为很简单，但当它们一起协同工作时却能够突现出非常复杂（智能）的行为特征。
  2. 个体间的交互作用在构建群行为中起到重要的作用。
  3. 没有集中控制且不提供全局模型。

- 优点
  1. 灵活性：群体可以适应随时变化的环境；
  2. 稳健性：个体失败，群体仍能完成任务；
  3. 自组织：活动既不受中央控制，也不受局部监管。
- 典型算法  
粒子群优化算法（鸟群捕食）  
蚁群算法（蚂蚁觅食）



粒子群算法

目录	1	算法的提出
	2	基本原理
	3	算法介绍
	4	算法流程
	5	算例及参数
	6	算法优缺点分析



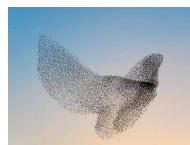
Russell Eberhart  
电子工程学博士



James Kennedy  
社会心理学博士

#### 算法的提出

- 1995年，受到鸟群活动的规律性启发， Russell Eberhart和James Kennedy建立了一个简化模型，并最终形成了**粒子群优化算法**（Particle Swarm Optimization, PSO），也可称为**粒子群算法**。



算法的提出

粒子群算法具有**收敛速度快、简单易实现**的优点，目前仍然受到众多研究者的青睐，不断扩展着它的应用领域。

作为一种新兴的群体搜索算法，粒子群算法得到了长足的发展，并成功应用于很多领域，比如**函数优化、图像处理、电力系统优化、医疗诊断等**。

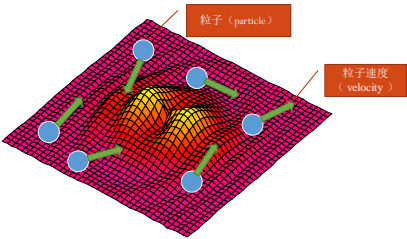


基本原理

鸟群有**准确的位置信息**和**准确的速度信息**，它们会向食物最多的地方



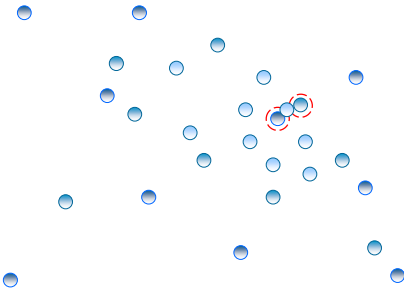
基本原理



觅食行为是利用社会中所存在的**互相合作**的概念，来引领所有个体朝向**最佳解位置**进行搜索

基本原理

粒子群搜索示意图



算法介绍

假设在D维搜索空间中，有m个粒子。

1) 其中第i个粒子的位置向量表示为：

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

2) 其飞行速度向量表示为：

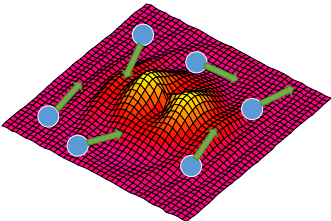
$$\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$$

3) 第i个粒子搜索到的最优位置为：

$$\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$$

4) 整个粒子群搜索到的最优位置为：

$$\mathbf{p}_{gbest} = (p_{gbest1}, p_{gbest2}, \dots, p_{gbestD})$$



算法介绍

粒子速度和位置的更新

$$v_{id}^{k+1} = wv_{id}^k + c_1rand_1(p_{id} - x_{id}^k) + c_2rand_2(p_{gbest} - x_{id}^k)$$


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

其中w为惯性权重，

d=1,2, ..., D, i=1,2, ..., m。

c<sub>1</sub>和c<sub>2</sub>为两个正常数,称为加速因子，rand<sub>1</sub>和rand<sub>2</sub>为均匀分布于[0,1]的随机数





西安电子科技大学  
XI'AN UNIVERSITY

算例

### 步骤2：粒子速度和位置更新


根据自身的历史最优位置和全局的最优位置，更新每个粒子的速度和位置。

$$p_1 = \begin{cases} v_1 = \omega \times v_1 + c_1 \times r_1 \times (pBest_1 - x_1) + c_2 \times r_2 \times (gBest - x_1) \\ \Rightarrow v_1 = \begin{cases} 0.5 \times 3 + 0 + 0 = 1.5 \\ 0.5 \times 2 + 0 + 0 = 1 \end{cases} = (1.5, 1) \\ x_1 = x_1 + v_1 = (8, -5) + (1.5, 1) = (9.5, -4) \end{cases}$$

$w$ 是惯性权重，一般取 $[0,1]$ 区间的数，本例中取为0.5；  
 $c_1$ 和 $c_2$ 为加速系数，本例中取为2；  
 $r_1$ 和 $r_2$ 是区间 $[0,1]$ 区间的随机数；

$$p_2 = \begin{cases} v_2 = w \times v_2 + c_1 \times r_1 \times (pBest_2 - x_2) + c_2 \times r_2 \times (gBest - x_2) \\ \Rightarrow v_2 = \begin{cases} 0.5 \times (-3) + 0 + 2 \times 0.3 \times (8 - (-5)) = 6.1 \\ 0.5 \times (-2) + 0 + 2 \times 0.1 \times ((-5) - 9) = 1.8 \end{cases} = (6.1, 1.8) \\ x_2 = x_2 + v_2 = (-5, 9) + (6.1, 1.8) = (1.1, 10.8) = (1.1, 10) \end{cases}$$

对于越界的位置需要进行合法性调整



西安电子科技大学  
XI'AN UNIVERSITY


算例

### 步骤2：粒子速度和位置更新

根据自身的历史最优位置和全局的最优位置，更新每个粒子的速度和位置。

$$p_3 = \begin{cases} v_3 = w \times v_3 + c_1 \times r_1 \times (pBest_3 - x_3) + c_2 \times r_2 \times (gBest - x_3) \\ \Rightarrow v_3 = \begin{cases} 0.5 \times 5 + 0 + 2 \times 0.05 \times (8 - (-7)) = 3.5 \\ 0.5 \times 3 + 0 + 2 \times 0.8 \times ((-5) - (-8)) = 6.3 \end{cases} = (3.5, 6.3) \\ x_3 = x_3 + v_3 = (-7, -8) + (3.5, 6.3) = (-3.5, -1.7) \end{cases}$$

$w$ 是惯性权重，一般取 $[0,1]$ 区间的数，本例中取为0.5；  
 $c_1$ 和 $c_2$ 为加速系数，本例中取为2；  
 $r_1$ 和 $r_2$ 是区间 $[0,1]$ 区间的随机数；



西安电子科技大学  
XI'AN UNIVERSITY


算例

### 步骤3：评估粒子的适应度函数值

根据适应度函数值更新粒子的历史最优位置和全局的最优位置

```

graph TD
    Start([开始]) --> Init[随机初始化每个粒子]
    Init --> Eval[评估每个粒子并得到全局最优]
    Eval --> Loop{ }
    Loop -- 是 --> End([结束])
    Loop -- 否 --> Update[更新每个粒子的速度和位置]
    Update --> EvalFunc[评估每个粒子的函数适应值]
    EvalFunc --> UpdateHist[更新每个粒子历史最优位置]
    UpdateHist --> UpdateGlobal[更新群体的全局最优位置]
    UpdateGlobal --> Loop
  
```



西安电子科技大学  
XI'AN UNIVERSITY

算例

### 步骤3：评估粒子的适应度函数值


根据适应度函数值更新粒子的历史最优位置和全局的最优位置。

$$f_1' = 9.5^2 + (-4)^2 = 90.25 + 16 = 106.25 > f_1 = 89 \quad \begin{cases} f_1 = 89 \\ pBest_1 = (8, -5) \end{cases}$$

$$f_2' = 1.1^2 + 10^2 = 1.21 + 100 = 101.21 < 106 = f_2 \quad \begin{cases} f_2 = f_2' = 101.21 \\ pBest_2 = x_2 = (1.1, 10) \end{cases}$$

$$f_3' = (-3.5)^2 + (-1.7)^2 = 12.25 + 2.89 = 15.14 < 113 = f_3 \quad \begin{cases} f_3 = f_3' = 15.14 \\ pBest_3 = x_3 = (-3.5, -1.7) \end{cases}$$

$$gBest = pBest_3 = (-3.5, -1.7)$$



西安电子科技大学  
XI'AN UNIVERSITY


算例

### 步骤4：结束判定

判断算法是否满足结束条件，在本例中将结束条件设置为达到一定的循环次数。

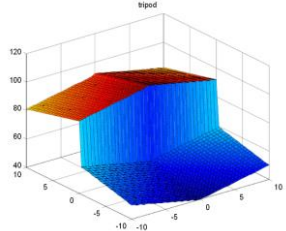
如果算法达到预设的循环次数，算法结束，此时的种群最优位置为算法找到的最优解；

如果没有达到预设的循环次数，转向步骤2循环执行；

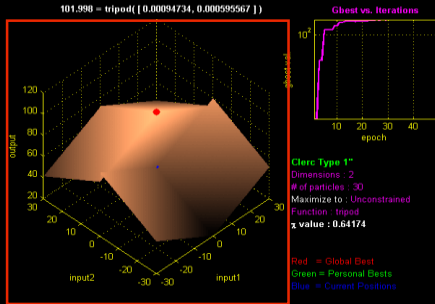


西安电子科技大学  
XI'AN UNIVERSITY

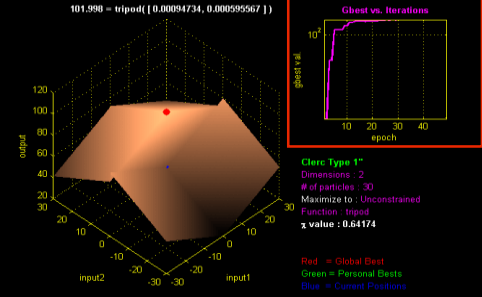
算法演示



## 算法演示



## 算法演示



## 算法参数

$$v_{id}^{k+1} = wv_{id}^k + c_1 \text{rand}_1(p_{id} - x_{id}^k) + c_2 \text{rand}_2(p_{gbestd} - x_{id}^k)$$

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

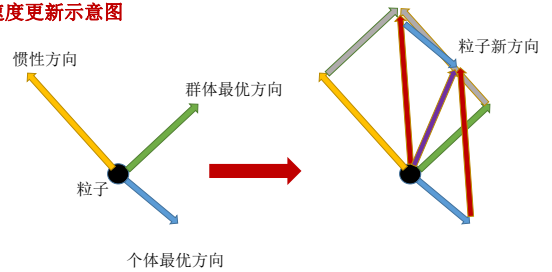
(1) 惯性权重  $w$ 

使粒子保持运动惯性，使其有搜索扩展空间的趋势，有能力探索新的区域。

也表示粒子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。

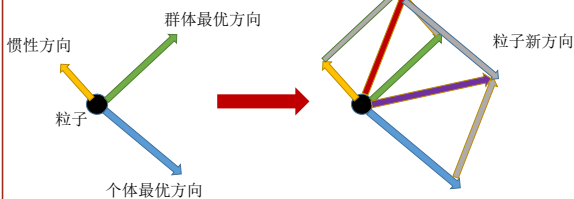
## 算法参数

## 速度更新示意图



## 算法参数

## 速度更新示意图



## 算法参数

$$v_{id}^{k+1} = wv_{id}^k + c_1 \text{rand}_1(p_{id} - x_{id}^k) + c_2 \text{rand}_2(p_{gbestd} - x_{id}^k)$$

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

## 惯性权重

惯性速度大的粒子更趋向于探索未知的空间，保证算法的探索能力。

惯性速度小的粒子更趋向于跟随种群最优方向，保证算法的收敛能力。

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。

算法参数

$$v_{id}^{k+1} = w_{id}^k + c_1 rand_1(p_{id} - x_{id}^k) + c_2 rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i=1,2,\dots,m; \quad d=1,2,\dots,D$$

(2) 改进的惯性权重 $w$

在解决实际优化问题时，往往希望先采用全局搜索，使搜索空间快速收敛于某一区域，然后采用局部精细搜索以获得高精度的解。

因此提出了自适应调整的策略，即随着迭代的进行，线性地减小  $w$  的值。

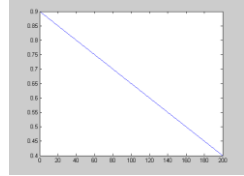
$$v_{id}^{k+1} = w_{id}^k + c_1 rand_1(p_{id} - x_{id}^k) + c_2 rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i=1,2,\dots,m; \quad d=1,2,\dots,D$$

(2) 改进的惯性权重 $w$

$w_{\max}$ 、 $w_{\min}$  分别是  $w$  的最大值和最小值；

$iter$ 、 $iter_{\max}$  分别是当前迭代次数和最大迭代次数。

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter$$



$$v_{id}^{k+1} = wv_{id}^k + c_1 rand_1(p_{id} - x_{id}^k) + c_2 rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i=1,2,\dots,m; \quad d=1,2,\dots,D$$

(3) 加速因子 $c_1$ 和 $c_2$

是代表将粒子推向个体最佳位置  $p_{best}$  和群体最佳位置  $g_{best}$  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。

低的值使粒子在目标区域外徘徊，而高的值导致粒子越过目标区域。

$$v_{id}^{k+1} = wv_{id}^k + c_1 rand_1(p_{id} - x_{id}^k) + c_2 rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i=1,2,\dots,m; \quad d=1,2,\dots,D$$

(4) 改进的加速因子 $c_1$ 和 $c_2$

通常将  $c_1$  和  $c_2$  统一为一个控制参数， $\phi = c_1 + c_2$

如果  $\phi$  很小，粒子群运动轨迹将非常缓慢；

如果  $\phi$  很大，则粒子位置变化非常快；

通过仿真可以获得  $\phi$  的经验值，当  $\phi=4.0$

( $c_1=2.0$ ,  $c_2=2.0$ ) 时，具有很好的收敛效果。

$$v_{id}^{k+1} = wv_{id}^k + c_1 rand_1(p_{id} - x_{id}^k) + c_2 rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i=1,2,\dots,m; \quad d=1,2,\dots,D$$

(5) 粒子数

通常一般取 20~40，对较难或特定类别的问题可以取 100~200。

(6) 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。

鸟类觅食与粒子群算法的对应关系

鸟	森林	粒子	求解空间
类		子	目标函数值
觅	食物的量	群	空间中的一个解
食	每只鸟所处的位置	算	(粒子位置)
	食物量最多的位置	法	全局最优解



算法优缺点分析

粒子群算法的优点是什么？

- 1. 简单易实现；
- 2. 收敛速度快；
- 3. 粒子有记忆功能。

粒子群算法的缺点是什么？

- 1. 缺乏速度的自适应调节，容易陷入局部最优，可能导致收敛精度低或不收敛；
- 2. 标准粒子群算法不能有效求解离散及组合优化问题；
- 3. 参数难以确定，对不同的问题，需选择合适的参数来达到最优效果。



粒子群算法的应用

找到约222,000条相关结果

按相关性

Particle swarm optimization

A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is

Dynamic clustering using particle swarm optimization with application in image segmentation

A new Improved particle swarm optimization and application to portfolio

selection

A research on particle swarm optimization and its application in robot manipulators

Design chip position of printed circuit board based on particle swarm optimization

Printed Circuit Board (PCB) is a critical component in the IC packaging process. Manufactured PCBs are typically designed for use by a specific kind of chip and cannot be used for multiple chip types.

However, allowing different kinds. Te-Jen Su, Yi-Feng Chen, Kun-Liang Lo - Modern Physics Letters B - 2019年

.....

小结

- 算法的提出
- 基本原理
- 算法介绍
- 算法流程
- 算例及参数
- 算法优缺点分析

思考

- 粒子在进行速度更新时，受到哪些因素的影响？
- 惯性权重的取值对算法性能有什么影响呢？



小结

$$v_{id}^{k+1} = wv_{id}^k + c_1rand_1(p_{id} - x_{id}^k) + c_2rand_2(p_{gbestd} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad i = 1, 2, \dots, m; \quad d = 1, 2, \dots, D$$

算法启示

- 遗传算法（GA），粒子群优化算法（PSO）
- 基于群体的优化
- 合作
- 如何合作？
- GA：交叉；PSO：速度更新
- 合作是解决复杂问题的最佳方式之一



Copyright by Lrc&Mch