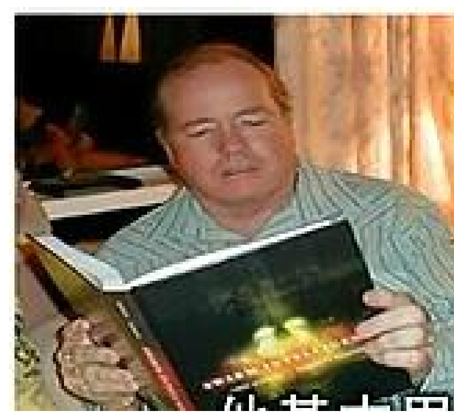
粒子群优化算法
Particle Swarm Optimization

粒子群算法发展历史简介

由美国心理学家Kennedy和电气工程师Eberhart于1995年提出.

- 模拟鸟群的觅食行为

- 基本思想：
  通过群体中个体之间的协作和信息共享寻找最优解

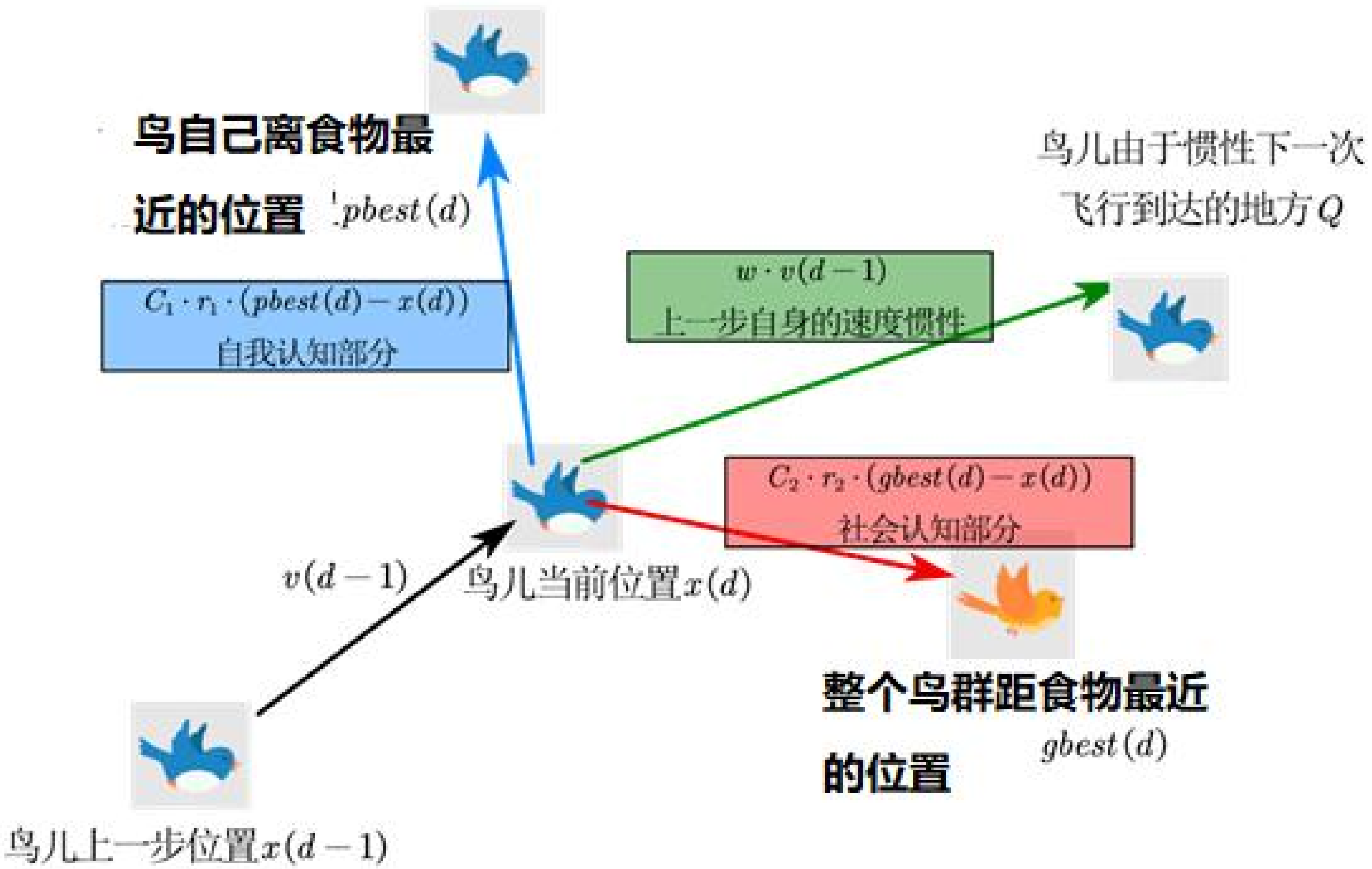


优势：

简单容易实现并且没有许多参数的调节。

目前已被广泛应用于函数优化、神经网络训练、模糊系统控制以及其他遗传算法的应用领域。

搜索策略



鸟自己离食物最
近的位置 $pbest(d)$
$C_1 \cdot r_1 \cdot (pbest(d) - x(d))$
自我认知部分
$w \cdot v(d-1)$
上一步自身的速度惯性
鸟儿由于惯性下一次
飞行到达的地方Q
$C_2 \cdot r_2 \cdot (gbest(d) - x(d))$
社会认知部分
$v(d-1)$
鸟儿当前位置$x(d)$
整个鸟群距食物最近
的位置
$gbest(d)$
鸟儿上一步位置$x(d-1)$

基本概念

粒子：优化问题的候选解

位置：候选解所在的位置

速度：候选解移动的速度

适应度：评价粒子优劣的值，一般设置为目标函数值

个体最佳位置：单个粒子迄今为止找到的最佳位置

群体最佳位置：所有粒子迄今为止找到的最佳位置

算法原理

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

$$\vec{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$$

◆ 粒子速度和位置的更新

第i个粒子搜索到的最优位置为

$$pbest_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$$

整个粒子群搜索到的最优位置为

$$gbest^{\mathrm{k}} = (g_1, g_2, \cdots, g_D)$$

第i个粒子接下来的位置和速度更新为：

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

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

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

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

“惯性部分”，又称为记忆项，

上次速度的影响

对自身运动状态的信任，

“自身认知部分”，

从当前点指向粒子自身最好的一个矢量。

对粒子本身的思考，即来源于自己经验的部分

“群体认知部分”，粒间子的信息共享，来源于群体中的其它优秀微粒的经验

其中，w称为惯性权重，值越大，全局搜索能力强。

c_1和c_2为称为学习因子。rand为[0,1]随机数

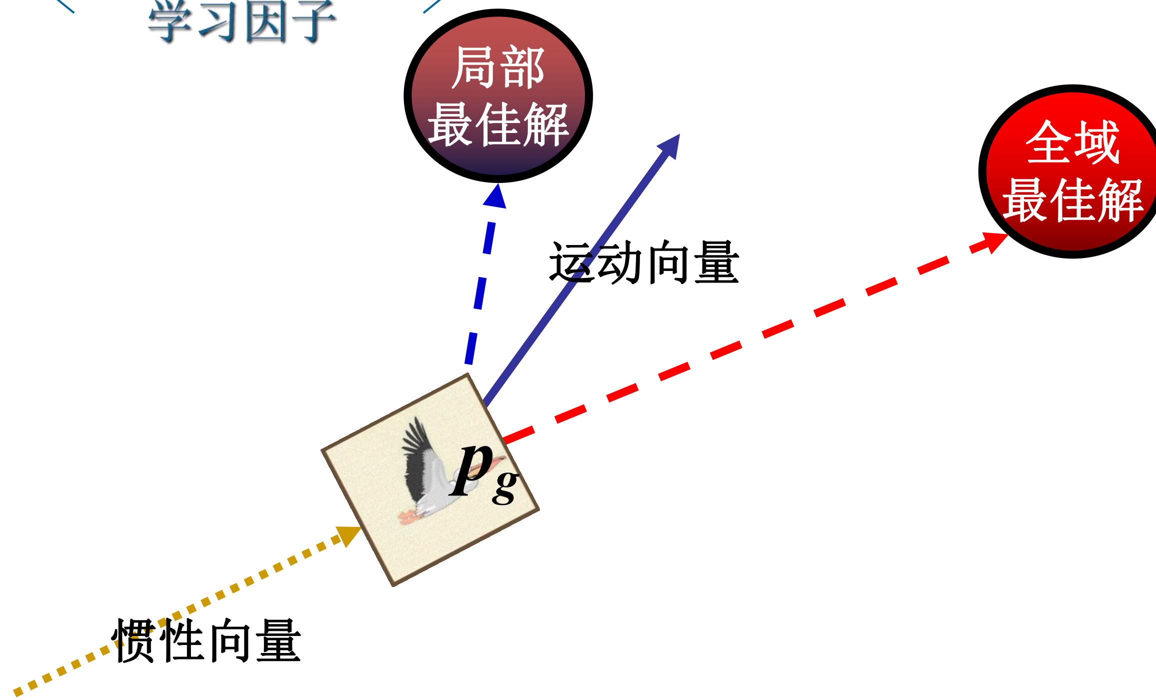
$$v_{id}(t+1)=w\cdot v_{id}(t)+c_1\cdot rand()\cdot(p_{id}-x_{id}(t))+c_2\cdot rand()\cdot(p_{gd}-x_{id}(t))$$

$$x_i(t+1)=x_i(t)+v_i(t)$$

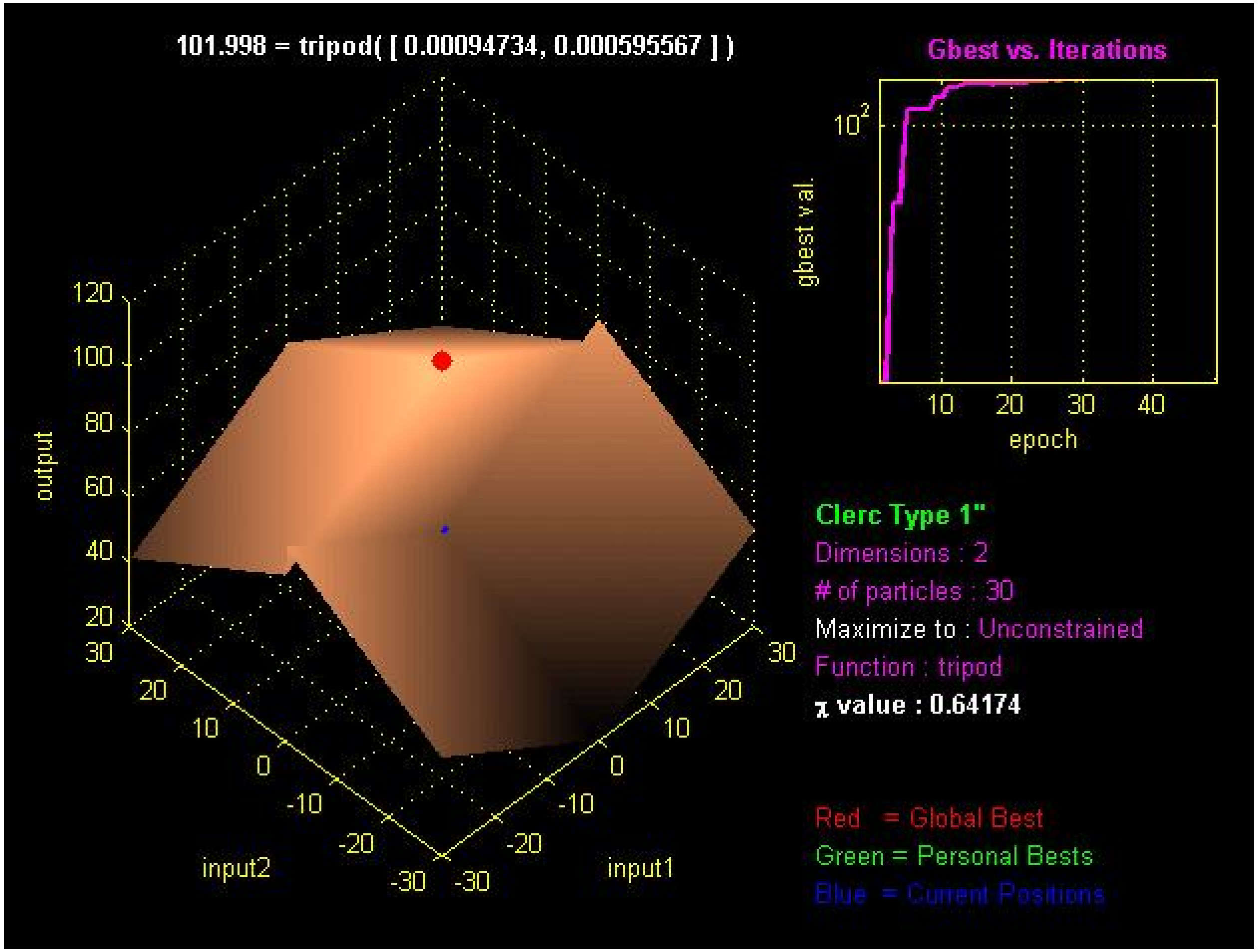
学习因子

$$V_i=\{V_{i1},V_{i2},\dots,V_{id}\}$$

$$X_i=\{X_{i1},X_{i2},\dots,X_{id}\}$$



粒子群算法描述

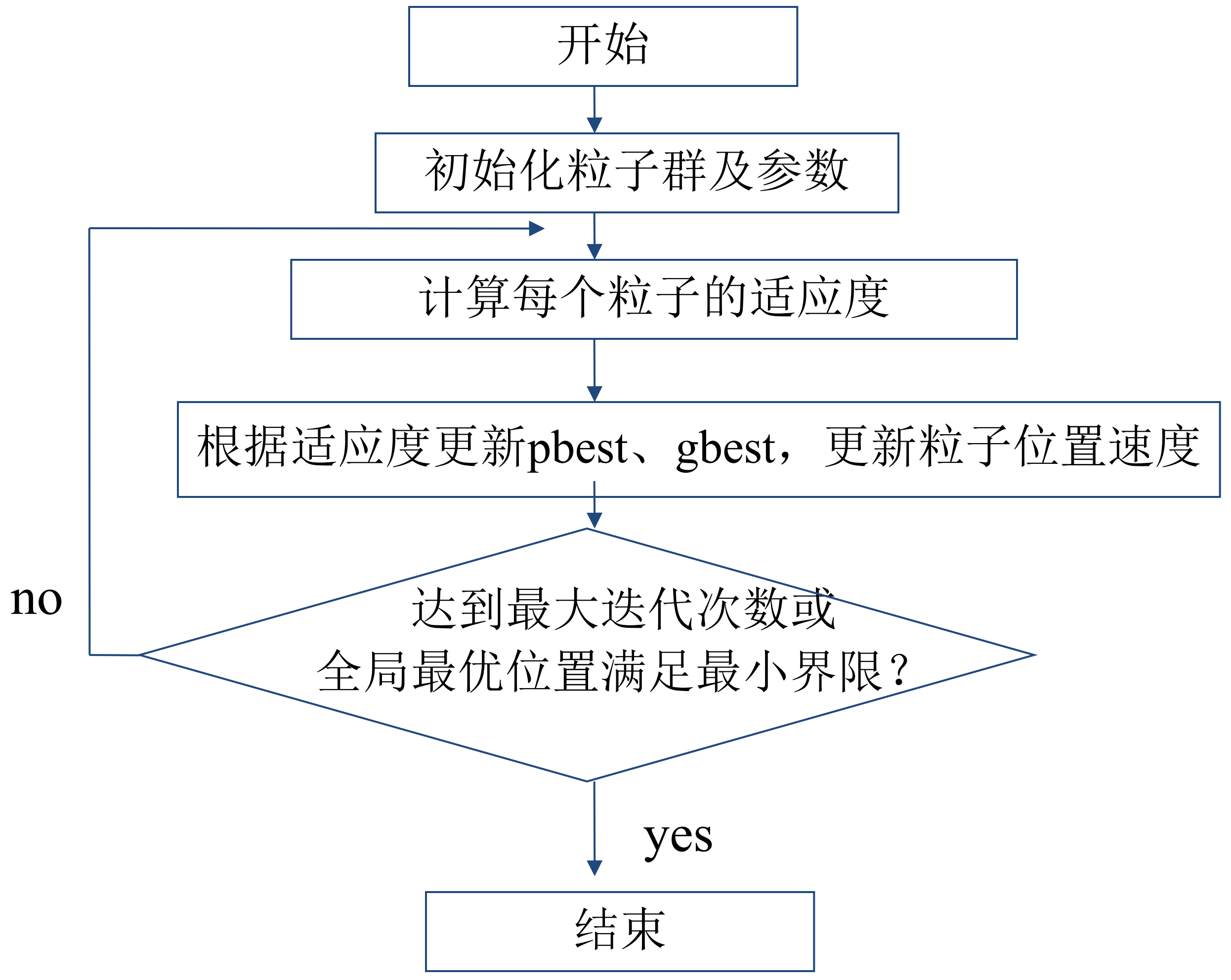


101.998 = tripod([0.00094734, 0.000595567])
output
120
100
80
60
40
20
30
20
10
0
-10
-20
-30
input2
-30
-20
-10
0
10
20
30
input1
Gbest vs. Iterations
gbest val.
10^2
10
20
30
40
epoch
Clerc Type 1"
Dimensions : 2
# of particles : 30
Maximize to : Unconstrained
Function : tripod
χ value : 0.64174
Red = Global Best
Green = Personal Bests
Blue = Current Positions

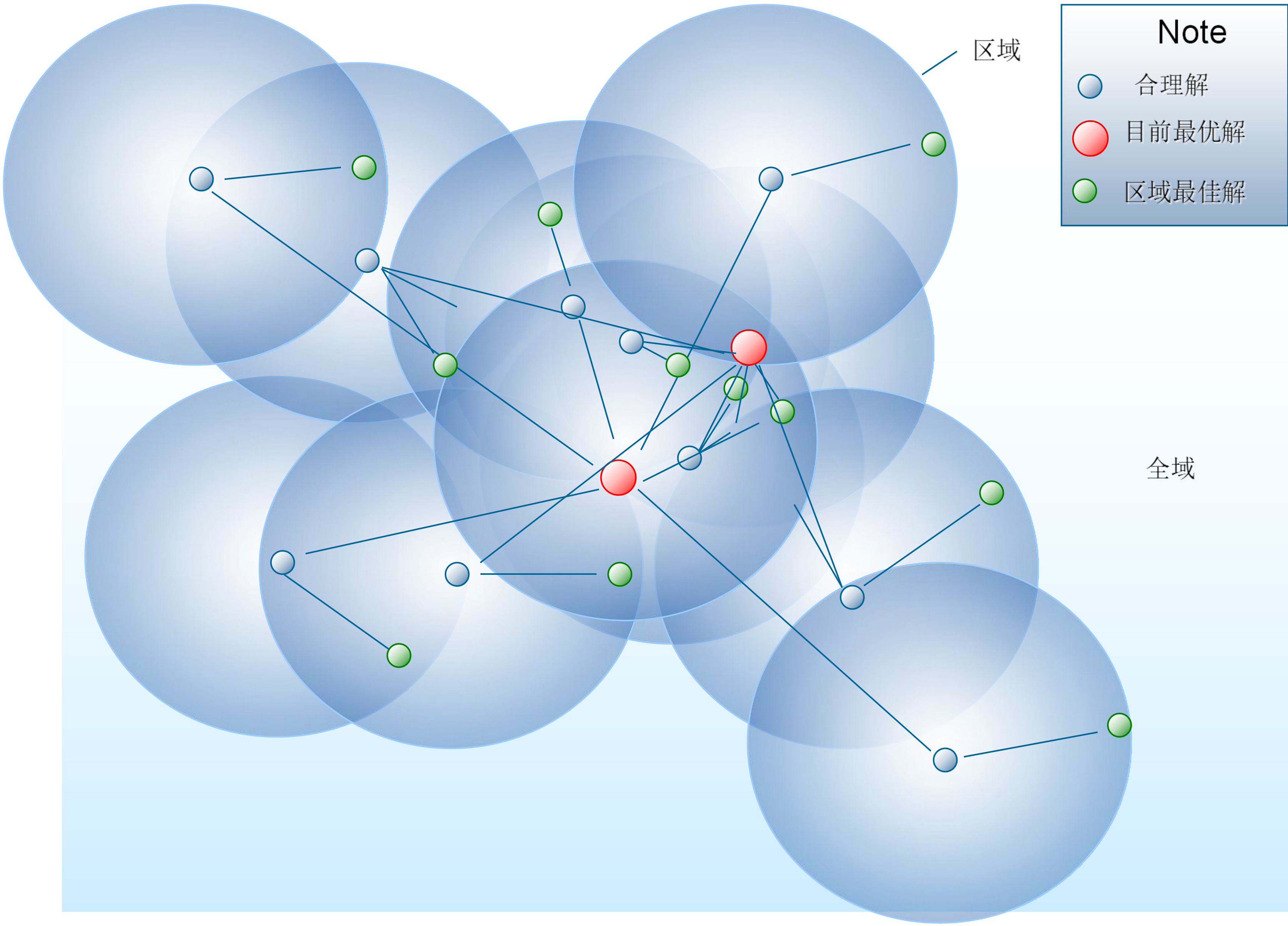
标准算法流程

1. **初始化**
   初始化粒子群体（群体规模为n），包括随机位置和速度。
2. **评估粒子的适应度**
   根据适应度函数 ，评价每个粒子的适应度。
3. **获取每个粒子的局部最优位置**
   对每个粒子，将其当前适应值与其个体历史最佳位置（pbest）对应的适应值做比较，如果当前的适应值更高，则将用当前位置更新历史最佳位置pbest。
4. **获取鸟群的全局最优位置**
   对每个粒子，将其当前适应值与全局最佳位置（gbest）对应的适应值做比较，如果当前的适应值更高，则将用当前粒子的位置更新全局最佳位置gbest。
5. **更新速度与位置**
   根据公式更新每个粒子的速度与位置。
6. **如未满足结束条件，则返回步骤2**
   通常算法达到最大迭代次数或者最佳适应度值的增量小于某个给定的阈值时算法停止。

算法流程图


开始
初始化粒子群及参数
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

简单例子



$$\mathrm{v}_{\mathrm{id}}^{\mathrm{k}}=\mathrm{wv}_{\mathrm{id}}^{\mathrm{k-1}}+c_1r_1(pbest_{id}-x_{id}^{k-1})+c_2r_2(gbest_d-x_{id}^{k-1})$$

自我认知部分 $c_1r_1(pbest_{id}-x_{id}^{k-1})$

社会经验部分 $c_2r_2(gbest_d-x_{id}^{k-1})$

c_1，c_2都不为0，称为完全型粒子群算法

完全型粒子群算法更容易保持收敛速度和搜索效果的均衡，是较好的选择．通常可选取2

惯性权重

$$\mathrm{v}_{\mathrm{id}}^{\mathrm{k}} = \mathrm{w}\mathrm{v}_{\mathrm{id}}^{\mathrm{k-1}} + c_1 r_1 (pbest_{id} - x_{id}^{k-1}) + c_2 r_2 (gbest_d - x_{id}^{k-1})$$

惯性权重 **w**
对算法的收敛起到很大的作用，
其值越大，粒子飞跃的范围就越广，
更容易找到全局最优，但是也会错失局部搜寻的能力。

取0.9－1.2是比较合适

例 2

已知函数 $y = f(x_1, x_2) = x_1^2 + x_2^2$

其中 $-10 \le x_1, x_2 \le 10$

用粒子群优化算法求解y的最小值。

步骤1：初始化。

假设种群大小是N=3；在搜索空间中随机初始化每个解的速度和位置，计算适应函数值，并且得到粒子的历史最优位置和群体的全局最优位置。

$$p_1=\begin{cases} v_1=(3,2) \\ x_1=(8,-5) \end{cases} \begin{cases} f_1=8^2+(-5)^2=64+25=89 \\ \boldsymbol{pBest}_1=\boldsymbol{x}_1=(8,-5) \end{cases}$$

$$p_2=\begin{cases} v_2=(-3,-2) \\ x_2=(-5,9) \end{cases} \begin{cases} f_2=(-5)^2+9^2=25+81=106 \\ \boldsymbol{pBest}_2=\boldsymbol{x}_2=(-5,9) \end{cases}$$

$$p_3=\begin{cases} v_3=(5,3) \\ x_3=(-7,-8) \end{cases} \begin{cases} f_3=(-7)^2+(-8)^2=49+64=113 \\ \boldsymbol{pBest}_3=\boldsymbol{x}_3=(-7,-8) \end{cases}$$

$$\boldsymbol{gBest}=\boldsymbol{pBest}_1=(8,-5)$$

步骤2：粒子的速度和位置更新。
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。

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

$$p_2 = \begin{cases} v_2 = \omega \times v_2 + c_1 \times r_1 \times (pBest_2 - x_2) + c_2 \times r_2 \times (gBest - x_2) \\ \Rightarrow v_2 = \begin{cases} 0.5 \times (-3) + 0 + 2 \times 0.3 \times (8 - (-5)) = 6.1 \\ 0.5 \times (-2) + 0 + 2 \times 0.1 \times ((-5) - 9) = 1.8 \end{cases} = (6.1,1.8) \\ x_1 = x_1 + v_1 = (-5,9) + (6.1,1.8) = (1.1,10.8) = (1.1,10) \end{cases}$$

$$p_3 = \begin{cases} v_3 = \omega \times v_3 + c_1 \times r_1 \times (pBest_3 - x_3) + c_2 \times r_2 \times (gBest - x_3) \\ \Rightarrow v_3 = \begin{cases} 0.5 \times 5 + 0 + 2 \times 0.05 \times (8 - (-7)) = 3.5 \\ 0.5 \times 3 + 0 + 2 \times 0.8 \times ((-5) - (-8)) = 6.3 \end{cases} = (3.5, 6.3) \\ x_1 = x_1 + v_1 = (-7, -8) + (3.5, 6.3) = (-3.5, -1.7) \end{cases}$$

步骤3：评估粒子的适应度函数值。
更新粒子的历史最优位置和全局的最优位置。

$$f_1^* = 9.5^2 + (-4)^2 = 90.25 + 16 = 106.25 > f_1 = 89$$

$$\begin{cases} f_1 = 89 \\ \boldsymbol{pBest}_1 = (8, -5) \end{cases}$$

$$f_2^* = 1.1^2 + 10^2 = 1.21 + 100 = 101.21 < 106 = f_2$$

$$\begin{cases} f_2 = f_2^* = 101.21 \\ \boldsymbol{pBest}_2 = X_2 = (1.1, 10) \end{cases}$$

$$f_3^* = (-3.5)^2 + (-1.7)^2 = 12.25 + 2.89 = 15.14 < 113 = f_3$$

$$\begin{cases} f_3 = f_3^* = 15.14 \\ \boldsymbol{pBest}_3 = x_3 = (-3.5, -1.7) \end{cases}$$

$$\boldsymbol{gBest} = \boldsymbol{pBest}_3 = (-3.5, -1.7)$$

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继续执行。

例 3 粒子群算法求解旅行商问题

1. 粒子的表示：TSP问题的一个解为一个序列，可以表示为一个粒子；

$x_i = (1, 3, 5, 2, 4)$，代表着该TSP问题一共有五个城市需要访问

2. 速度的表示：用一个序列的交换序列表示粒子的速度。

交换序列为一组有前后顺序的交换子的集合，

交换子定义为 $s = Swap\, x(i, j)$，

$x_i = (1, 3, 5, 2, 4)$，$s = Swap\, x(1, 3)$，

得到新的序列 $x_j = (5, 3, 1, 2, 4)$

即 $ss = [Swap_1, Swap_2, \ldots]$，

e.g. $x_i = (1, 3, 5, 2, 4)$

经过交换序列 $ss = [(3, 2), (1, 5)]$ 的交换

得到 $x_i = (4, 5, 3, 2, 1)$

之前的速度迭代公式:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (Pbest - X_i) + c_2 r_2 (Gbest - X_i)$$

在这里我们重新定义速度个位置更新公式:

$$V_i^{t+1} = \omega V_i^t \oplus c_1 r_1 (Pbest - X_i) \oplus c_2 r_2 (Gbest - X_i)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

3. 适应度函数的定义： 当前序列的路径长度即为适应度值

4. 惯性因子的定义：自身的交换序列即惯性因子

对于TSP问题，已知$X_i(t-1)=\{1,2,3,4,5\}$，$V_i(t-1)=\{(3,5),(1,2)\}$，，

$P_i(t-1)=\{3,1,5,2,4\}$，$G(t-1)=\{1,2,3,5,4\}$，$\alpha=0.6$、$\beta=0.4$，。试求$X_i(t)=$？

$$\begin{cases} V_i(t)=V_i(t-1)+\alpha(P_i(t-1)-X_i(t-1))+\beta(G(t-1)-X_i(t-1)) \\ X_i(t)=X_i(t-1)\oplus V_i(t) \end{cases}$$

首先对于$X_i(t-1)$的第 1 维，运算后应该等 于$P_i(t-1)$的第 1 维，

为此应该进行$Q_1=\text{swap}(1,3)$交换操作，运算结果为$\{3,1,2,4,5\}$；

在$\{3,2,1,4,5\}$中的第 2 维要想与$P_i(t-1)$的第 2 维相同， 进行$Q_2=\text{swap}(2,1)$的交换操作，这样得到的运算结果为$\{3,1,2,4,5\}$； 进行两个交换子的操作$Q_3=\text{swap}(2,5)$、$Q_4=\text{swap}(4,2)$，

因此$P_i(t-1)-X_i(t-1)=\{Q_1,Q_2,Q_3,Q_4\}=\{(1,3),(2,1),(2,5),(4,2)\}$。

$$G(t-1)-X_i(t-1)=\{(4,5)\}$$

$$\begin{aligned} V_i(t)&=V_i(t-1)+\alpha(P_i(t-1)-X_i(t-1))+\beta(G(t-1)-X_i(t-1)) \\ &=\{(3,5),(1,2)\}+\{(1,3),(2,5),(4,2)\}+\{(4,5)\} \\ &=\{(3,5),(1,2),(1,3),(2,5),(4,2),(4,5)\} \end{aligned}$$

$X_i(t)=X_i(t-1)\oplus V_i(t)$表示按照$V_i(t)$中交换子的排列顺序依次对$X_i(t-1)$进行交换操作，因此最终结果$X_i(t)=\{4,3,5,2,1\}$。

PSO应用

PSO 比较有潜力的应用包括系统设计、多目标优化、分类、模式识别、调度、信号处理、决策、机器人应用等。其中具体应用实例有: 模糊控制器设计、车间作业调度、机器人实时路径规划、自动目标检测、时频分析等 。

PSO存在问题

- 一种新兴的优化算法：其数学基础薄弱,在收敛性理论、计算性能、实现技术和参数的设置等方面缺乏严密的数学基础,其应用大多数仍然依靠经验和实验。
- PSO算法的理论研究： 纵观PSO的研究成果,大部分研究都集中在算法的设计上,对算法的性能、收敛性、收敛速度、参数选取及参数的鲁棒性等理论性的研究则很少,偶有一些理论研究,但仅仅局限在对算法的参数、状态及概念等方面,且理论分析的内容和深度都很浅,因此理论研究大大滞后于PSO在工程中的应用。

PSO研究热点

2004 年, IEEE 进化计算会议PSO 专集 (Guest Editorial Special Issue on Particle Swarm Optimization)， 卷首语中指出了当前研究的几个主要方向及热点:

(1) 算法收敛性分析 PSO在实际应用中被证明是有效的, 但目前还没有给出收敛性、收敛速度估计等方面的数学证明,已有的工作还远远不够。

（2）**粒子群拓扑结构** 不同的粒子群邻居拓扑结构是对不同类型社会的模拟,研究不同拓扑结构的适用范围,对PSO算法推广和使用有重要意义。

（3）**参数选择与优化** 参数w、φ1、φ2的选择分别关系粒子速度的3个部分:惯性部分、社会部分和自身部分在搜索中的作用。如何选择、优化和调整参数,使得算法既能避免早熟又能比较快速地收敛,对工程实践有着重要意义。

(4) 与其它演化计算的融合　如何将其它演化的优点和PSO的优点相结合,构造出有特色有实用价值的混合算法是当前算法改进的一个重要方向。

(5) 算法应用　算法的有效性必须在应用中才能体现,广泛地开拓PSO的应用领域,也对深化研究PSO算法非常有意义。

作业

使用粒子群算法 求解函数y=-x*(x-1) 在[0,2]上最大值

相关参数如下：2个粒子

粒子n0: x = 0.004 v = 0.004
粒子n1: x = 0.0 v = -4.06577

w = 0.4 , c1 = c2 = 2; vmax = 0.1

请写出迭代3步的详细过程，和相关输出结果

程序作业

使用粒子群优化算法求解以下函数的最小值

$$f(x)=\sum_{i=1}^{30} x_i^2 + x_i - 6$$

粒子群规模为50， w = 0.5， c1 = 1.5， c2 = 2.5，迭代100步，定义域为[-1,1]