第6章 粒子群优化算法

Contents

6.1 粒子群优化算法简介

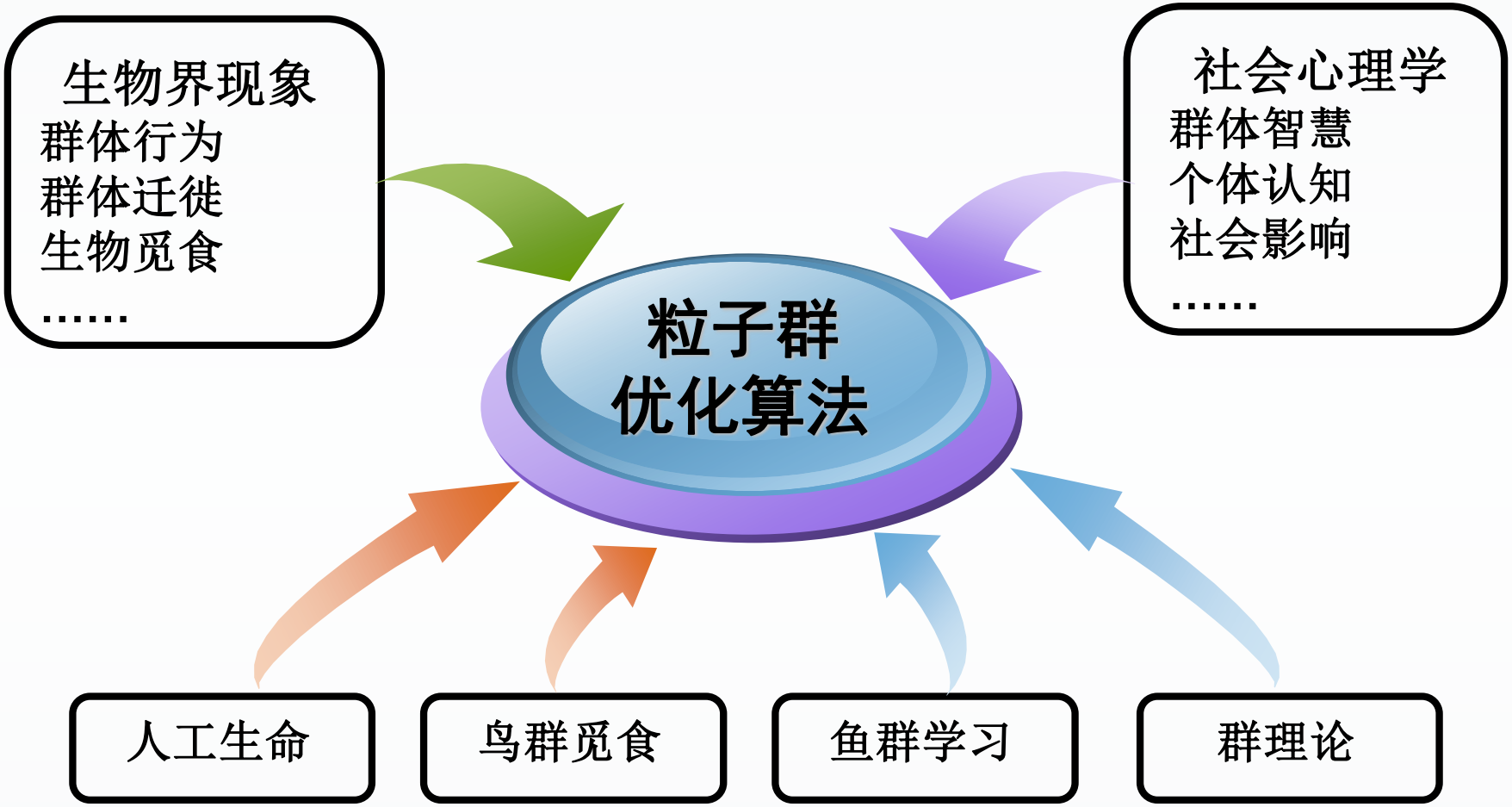
粒子群优化算法是什么？

粒子群优化算法的思想来源是怎样的?
它由谁提出的?

粒子群优化算法
（Particle Swarm Optimization，PSO）
是进化计算的一个分支，
是一种模拟自然界的生物

PSO模拟了自然界鸟群捕食和鱼群捕食的过程。
通过群体中的协作寻找到问题的全局最优解。
它是1995年由美国学者Eberhart和Kennedy提出的，
现在已经广泛应用于各种工程领域的优化问题之中。

6.1.1 思想来源

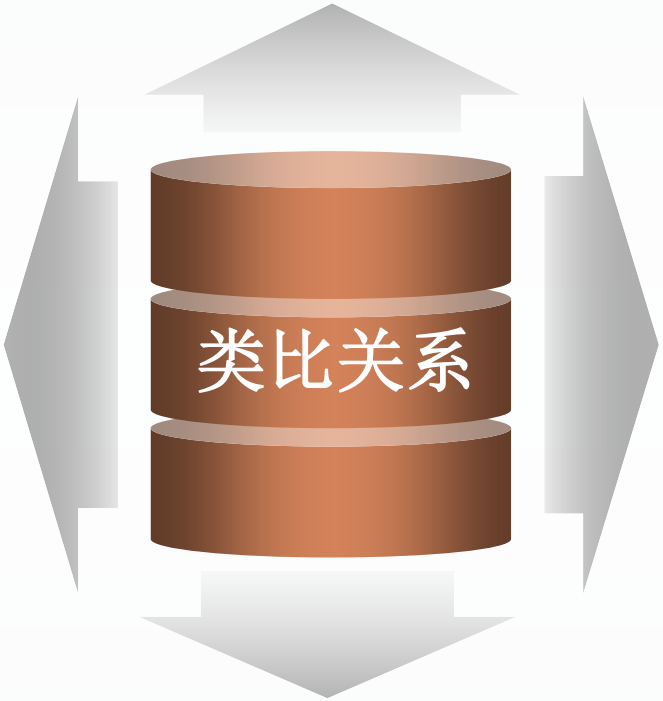

生物界现象
群体行为
群体迁徙
生物觅食
……
社会心理学
群体智慧
个体认知
社会影响
……
粒子群
优化算法
人工生命
鸟群觅食
鱼群学习
群理论

6.1.2 基本原理

鸟群觅食现象

鸟群觅食现象

- 鸟群
- 觅食空间
- 飞行速度
- 所在位置
- 个体认知与群体协作
- 找到食物

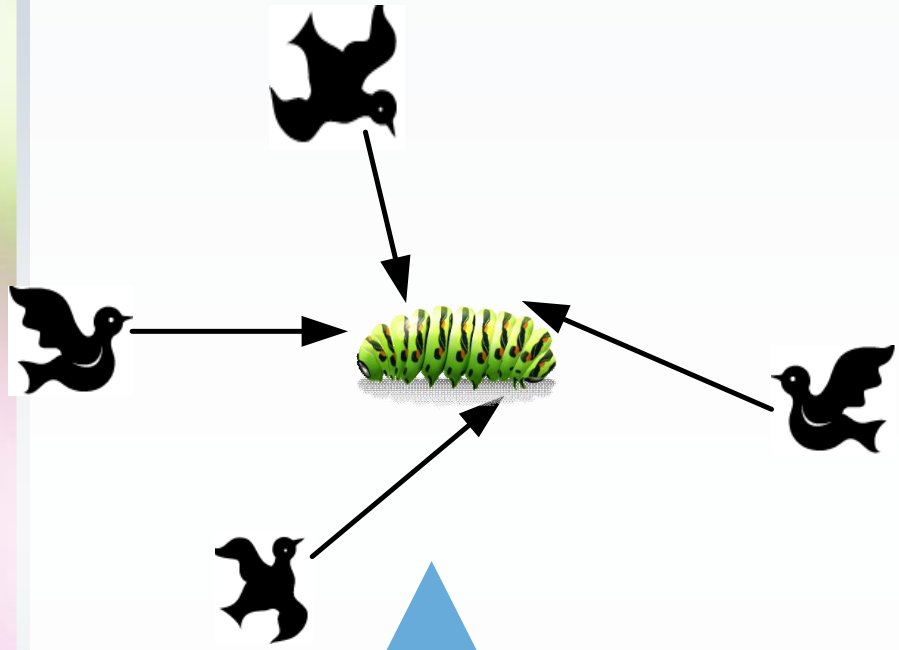


粒子群优化算法

- 搜索空间的一组有效解
- 问题的搜索空间
- 解的速度向量
- 解的位置向量
- 速度与位置的更新
- 找到全局最优解

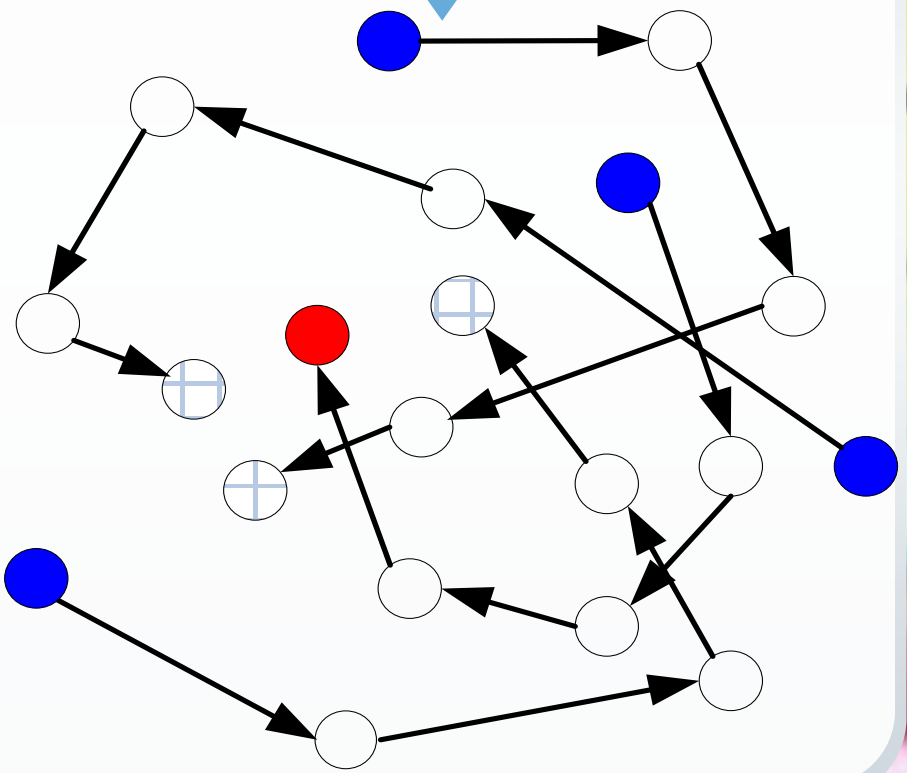
粒子群优化算法

6.1.2 基本原理



鸟群觅食现象

粒子群优化算法



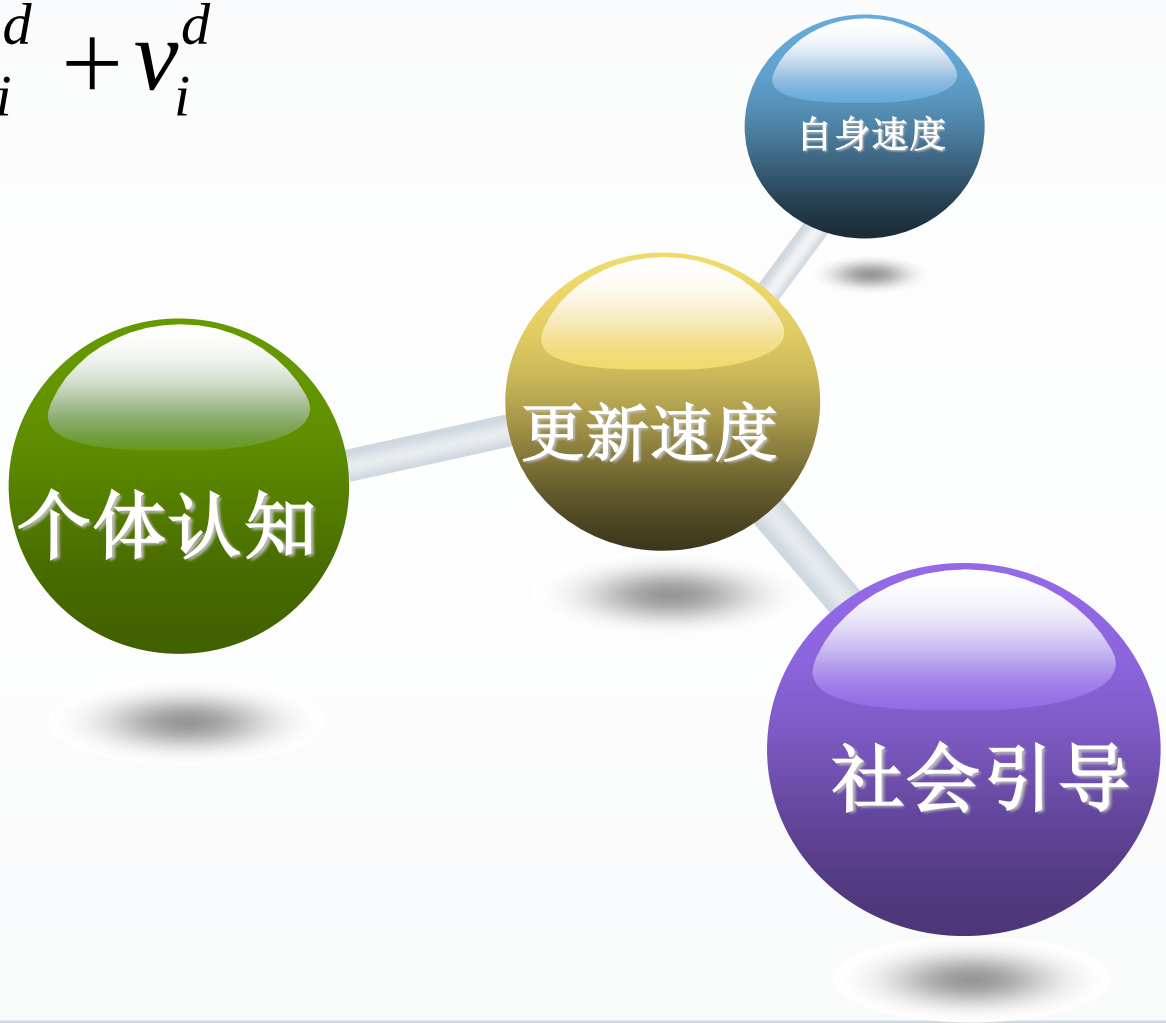
6.2 粒子群优化算法的基本流程

- 基本流程
  - 速度与位置更新公式
  - 速度与位置更新示意图
  - 算法流程图和伪代码
- 应用举例
  - 函数最小化问题
  - 算法的执行步骤示意图

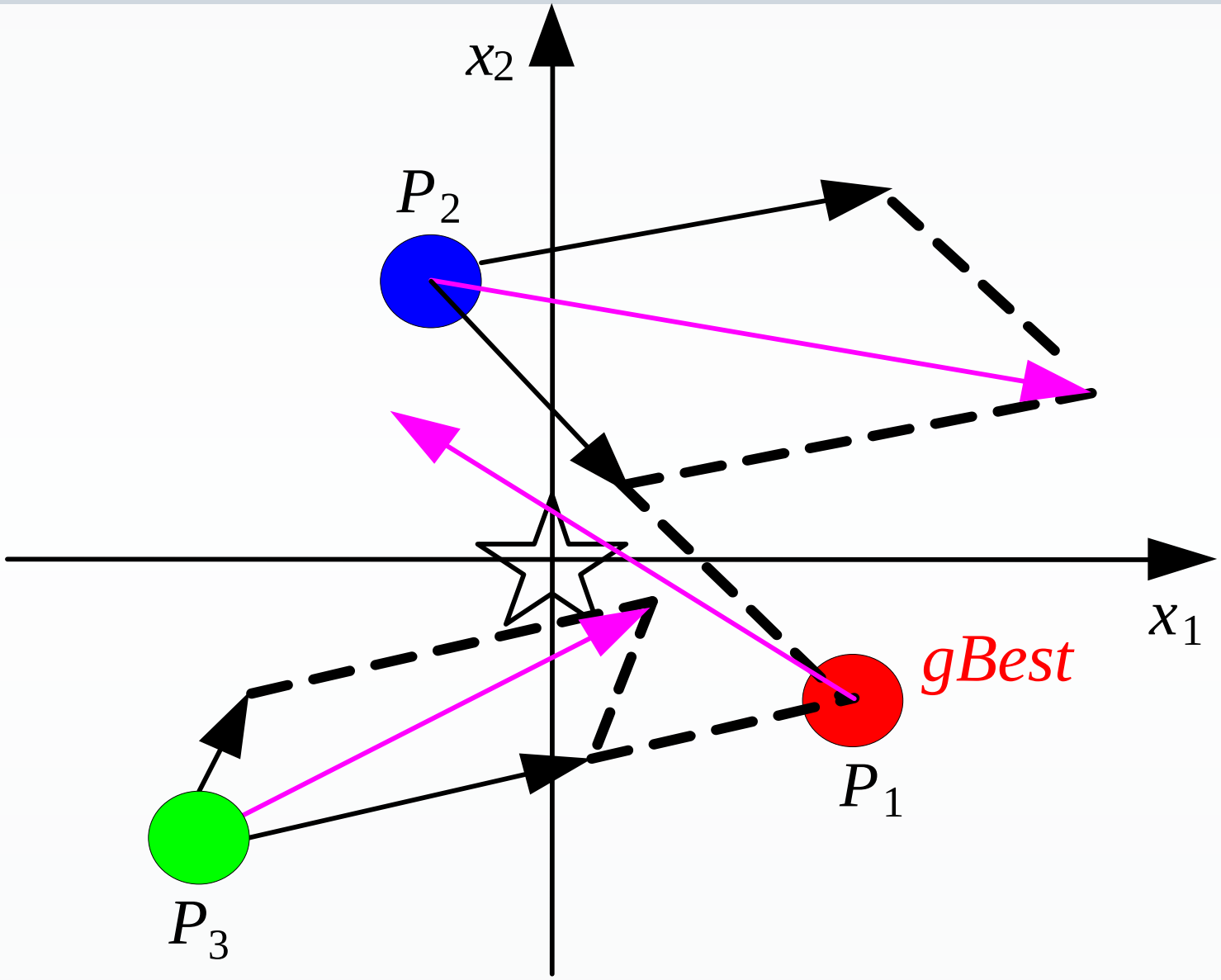
粒子的个体速度与位置更新公式

$$v_i^d = \omega \times v_i^d + c_1 \times r_1^d \times (pBest_i^d - x_i^d) + c_2 \times r_2^d \times (gBest^d - x_i^d)$$

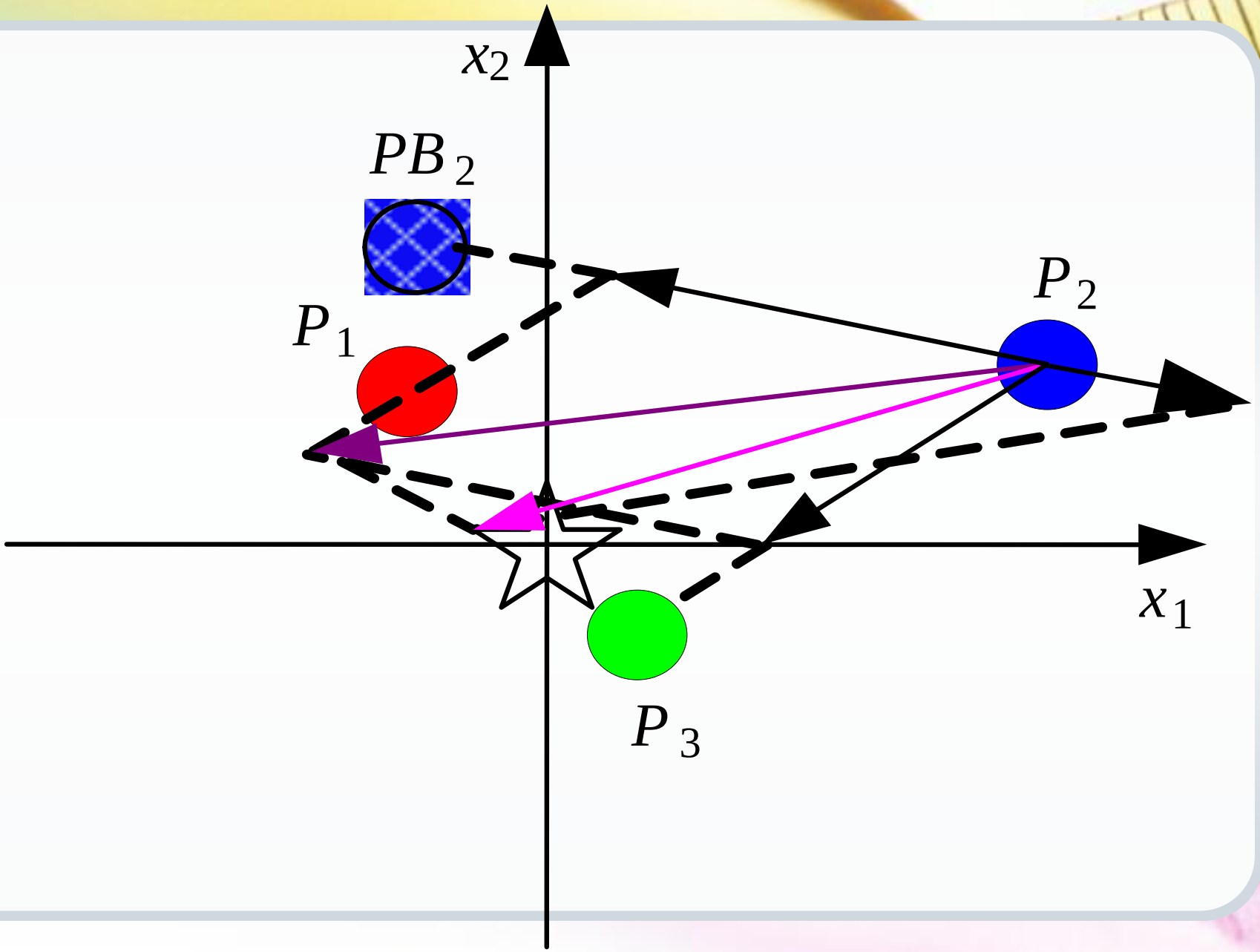
$$x_i^d = x_i^d + v_i^d$$



速度与位置更新示意图

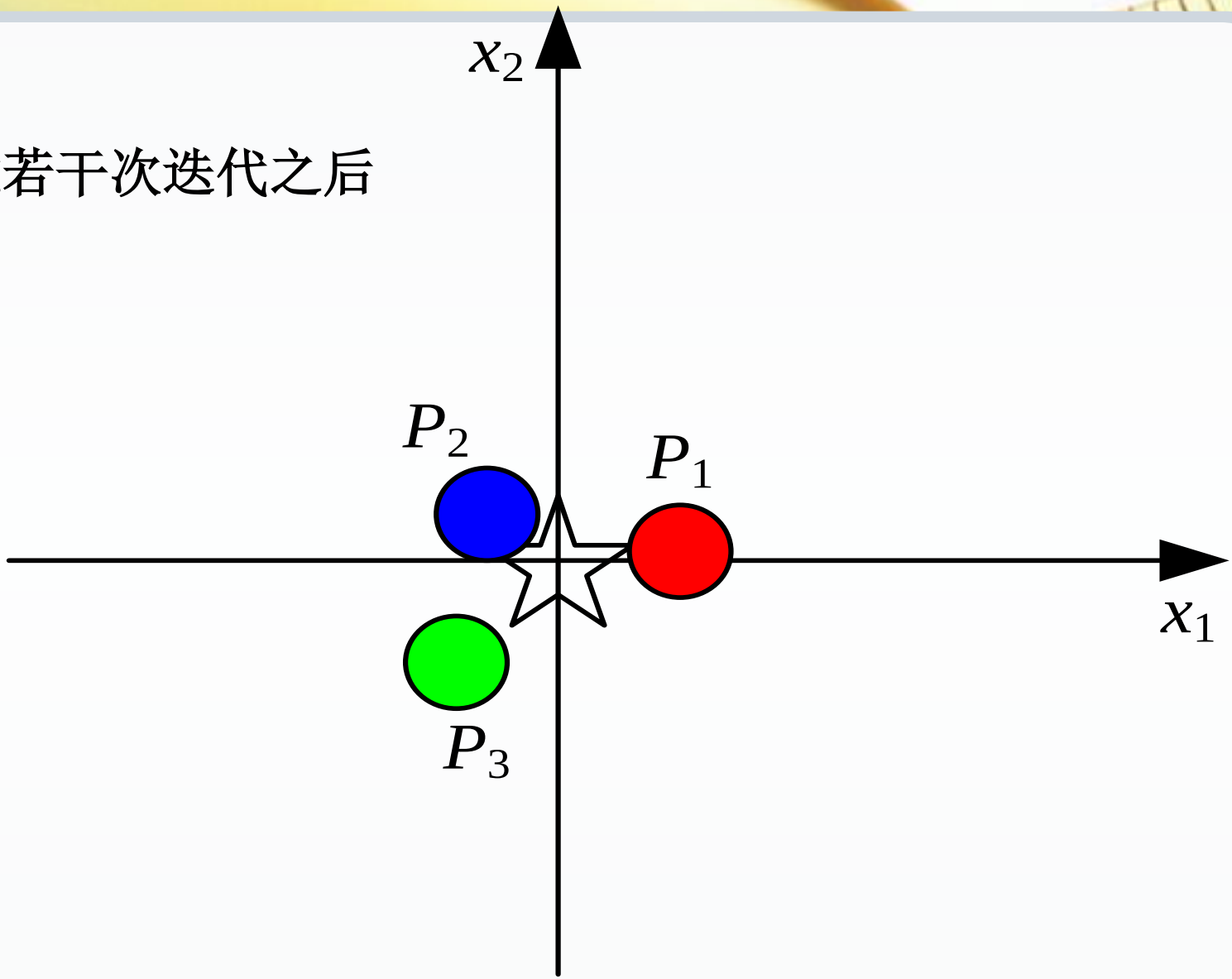


速度与位置更新示意图

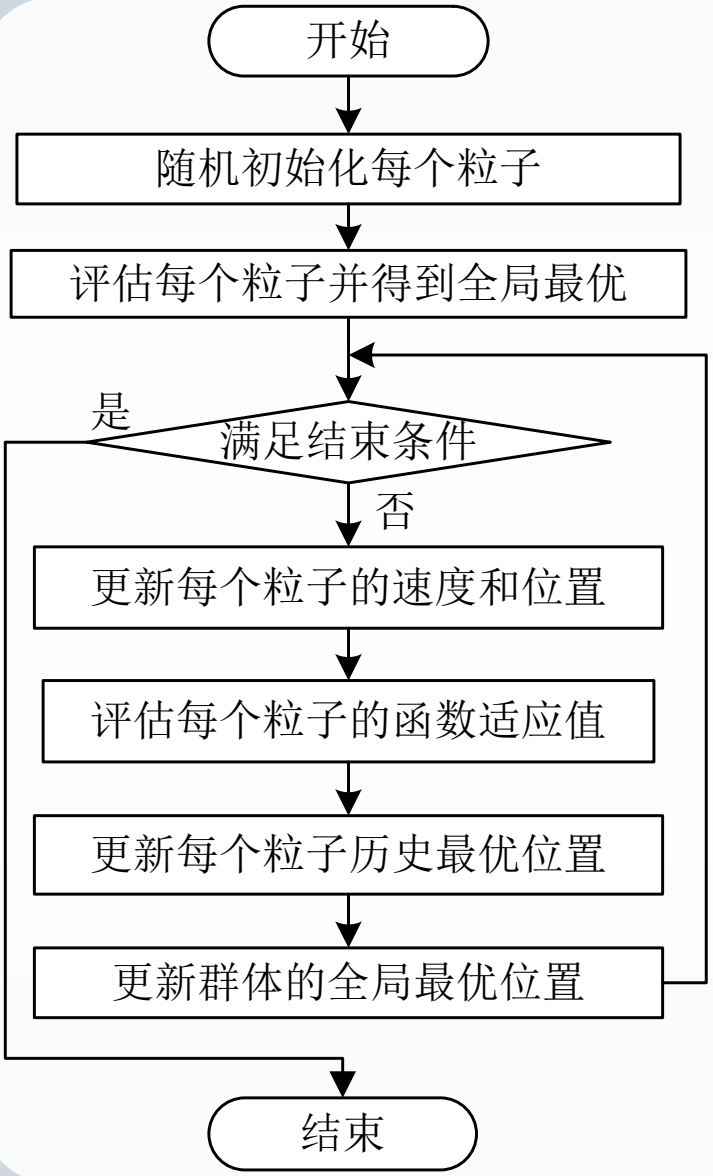


速度与位置更新示意图

经过若干次迭代之后



PSO算法流程图和伪代码




```
//功能：粒子群优化算法伪代码
//说明：本例以求问题最小值为目标
//参数：N为群体规模

procedure PSO
   for each particle i
      Initialize velocity Vi and position Xi for particle i
      Evaluate particle i and set pBesti = Xi
   end for
   gBest = min {pBesti}
   while not stop
      for i=1 to N
         Update the velocity  and position of particle i
         Evaluate particle i
         if fit (Xi) < fit (pBesti)
               pBesti = Xi;
         if fit(pBesti) < fit (gBest)
               gBest = pBesti;
       end for
   end while
   print gBest
end procedure
```

6.2.2 应用举例

- 例6.1　　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。

运行步骤

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。

$p_1=\begin{cases}v_1=(3,2)\\ x_1=(8,-5)\end{cases}$ $\begin{cases}f_1=8^2+(-5)^2=64+25=89\\ pBest_1=x_1=(8,-5)\end{cases}$

$p_2=\begin{cases}v_2=(-3,-2)\\ x_2=(-5,9)\end{cases}$ $\begin{cases}f_2=(-5)^2+9^2=25+81=106\\ pBest_2=x_2=(-5,9)\end{cases}$

$p_3=\begin{cases}v_3=(5,3)\\ x_3=(-7,-8)\end{cases}$ $\begin{cases}f_3=(-7)^2+(-8)^2=49+64=113\\ pBest_3=x_3=(-7,-8)\end{cases}$

$gBest=pBest_1=(8,-5)$

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。

$v_1=\omega\times v_1+c_1\times r_1\times(pBest_1-x_1)+c_2\times r_2\times(gBest-x_1)$

$\Rightarrow v_1=\begin{cases}0.5\times3+0+0=1.5\\ 0.5\times2+0+0=1\end{cases}=(1.5,1)$

$x_1=x_1+v_1=(8,-5)+(1.5,1)=(9.5,-4)$

$v_2=\omega\times v_2+c_1\times r_1\times(pBest_2-x_2)+c_2\times r_2\times(gBest-x_2)$

$\Rightarrow v_2=\begin{cases}0.5\times(-3)+0+2\times0.3\times(8-(-5))=6.1\\ 0.5\times(-2)+0+2\times0.1\times((-5)-9)=\dots\end{cases}=(6.1,1.8)$

$x_2=x_2+v_2=(-5,9)+(6.1,1)=(1.1,10)$

$v_3=\omega\times v_3+c_1\times r_1\times(pBest_3-x_3)+c_2\times r_2\times(gBest-x_3)$

$\Rightarrow v_3=\begin{cases}0.5\times5+0+2\times0.05\times(8-(-7))=3.5\\ 0.5\times3+0+2\times0.8\times((-5)-(-8))=6.3\end{cases}=(3.5,6.3)$

$x_3=x_3+v_3=(-7,-8)+(3.5,6.3)=(-3.5,-1.7)$

w是惯量权重，一般取[0,1]区间的数，这里假设为0.5

c_1和c_2为加速系数，通常取固定值2.0

r_1和r_2是[0,1]区间的随机数

注意：对粒子的速度和位置要进行合法性调整

步骤3：评价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。

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

$\begin{cases}f_1=89\\ pBest_1=(8,-5)\end{cases}$

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

$\begin{cases}f_2=f_2^*=101.21\\ pBest_2=x_2=(1.1,10)\end{cases}$

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

$\begin{cases}f_3=f_3^*=15.14\\ pBest_3=x_3=(-3.5,-1.7)\end{cases}$

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

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。

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。

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

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

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

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

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。

$$p_1=\begin{cases}v_1=\omega\times v_1+c_1\times r_1\times(pBest_1-x_1)+c_2\times r_2\times(gBest-x_1)\\ \Rightarrow v_1=\begin{cases}0.5\times3+0+0=1.5\\ 0.5\times2+0+0=1\end{cases}=(1.5,1)\\ x_1=x_1+v_1=(8,-5)+(1.5,1)=(9.5,-4)\end{cases}$$

$$p_2=\begin{cases}v_2=\omega\times v_2+c_1\times r_1\times(pBest_2-x_2)+c_2\times r_2\times(gBest-x_2)\\ \Rightarrow v_2=\begin{cases}0.5\times(-3)+0+2\times0.3\times(8-(-5))=6.1\\ 0.5\times(-2)+0+2\times0.1\times((-5)-9)=1.8\end{cases}=(6.1,1.8)\\ x_1=x_1+v_1=(-5,9)+(6.1,1.8)=(1.1,10.8)=(1.1,10)\end{cases}$$

注意！ 对于越界的位置，需要进行合法性调整

$$p_3=\begin{cases}v_3=\omega\times v_3+c_1\times r_1\times(pBest_3-x_3)+c_2\times r_2\times(gBest-x_3)\\ \Rightarrow v_3=\begin{cases}0.5\times5+0+2\times0.05\times(8-(-7))=3.5\\ 0.5\times3+0+2\times0.8\times((-5)-(-8))=6.3\end{cases}=(3.5,6.3)\\ x_1=x_1+v_1=(-7,-8)+(3.5,6.3)=(-3.5,-1.7)\end{cases}$$

w是惯量权重，一般取[0,1]区间的数，这里假设为0.5

c_1和c_2为加速系数，通常取固定值2.0

r_1和r_2是[0,1]区间的随机数

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。

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

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

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

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

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

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

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

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。

6.3 粒子群优化算法的改进研究

与PSO相关的重要学术期刊与国际会议

- 重要学术期刊
  - IEEE Transactions on Evolutionary Computation
  - IEEE Transactions on Systems, Man and Cybernetics
  - IEEE Transactions on ……
  - Machine Learning
  - Evolutionary Computation
  - ……

与PSO相关的重要学术期刊与国际会议

- 重要国际会议
  - IEEE Congress on Evolutionary Computation (CEC)
  - IEEE International Conference on Systems, Man, and Cybernetics (SMC)
  - ACM Genetic and Evolutionary Computation Conference (GECCO)
  - International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS)
  - International Conference on Simulated Evolution And Learning (SEAL)
  - ……

6.3.1 理论研究改进

2002

Clerc&Kennedy 2002年设计了一个称为压缩因子的参数。在使用了此参数之后，PSO能够更快地收敛

2003

Trelea 2003年指出PSO最终最终稳定地收敛于空间中的某一个点，但不能保证是全局最优点

2006

Kadirkamanathan等人2006年在动态环境中对PSO的行为进行研究，由静态分析深入到了动态分析

2006

F. van den Bergh等人2006年对PSO的飞行轨迹进行了跟踪，深入到了动态的系统分析和收敛性研究

6.3.2 拓扑结构改进

静态拓扑结构

全局版本：

星型结构

局部版本：

环形结构

齿形结构

金字塔结构

冯诺依曼结构

……

动态拓扑结构

逐步增长法

Suganthan 1999

最小距离法

Hu & Eberhart 2002

重新组合法

Liang&Suganthan2005

随机选择法

Kennedy 等人 2006

……

其它拓扑结构

社会趋同法

Kennedy 2000

Fully Informed

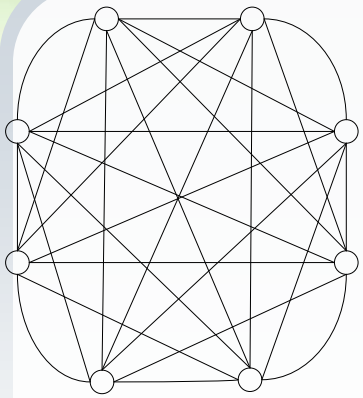
Mendes 等人 2004

广泛学习策略

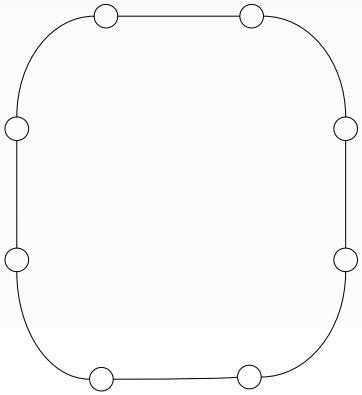
Liang 等人 2006

……

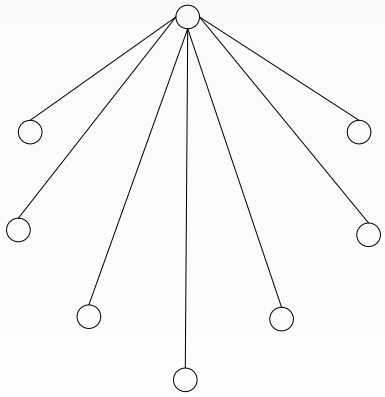
几种典型的拓扑结构示意图



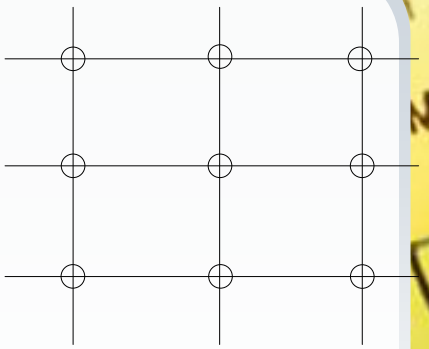
(a) 星型结构



(b) 环型结构



(c) 齿型结构



(d) 冯诺依曼结构

全局版本PSO和局部版本PSO在收敛特点：

1. GPSO由于其很高的连接度，往往具有比LPSO更快的收敛速度。但是，快速的收敛也让GPSO付出了多样性迅速降低的代价
2. LPSO由于具有更好的多样性，因此一般不容易落入局部最优，在处理多峰问题上具有更好的性能

在解决具体问题的时候，可以遵循以下一些规律：

(A)邻域较小的拓扑结构在处理复杂的、多峰值的问题上具有优势，例如环型结构的LPSO

(B)随着邻域的扩大，算法的收敛速度将会加快，这对简单的、单峰值的问题非常的有利，例如GPSO在这些问题上就表现很好

6.3.3 混合算法改进

混合进化算子的改进

选择算子
交叉算子
变异算子
……
进化规划
进化策略
蚁群算法
……

混合其它搜索算法的改进

结合模拟退火算法
结合人工免疫算法
结合差分进化算法
结合局部搜索算法
……

混合其它技术的改进

单纯形技术
函数延伸技术
混沌技术
量子技术
协同技术
小生境技术
物种形成技术
……

6.3.4 混合算法改进

二进制编码

Kennedy和Eberhart 1997 年对PSO进行了离散化，形成了二进制编码的PSO(BPSO)，并且在对De Jong的五个标准测试函数的测试中取得较好的效果

整数编码

Salman等人2002 年将粒子的位置变量四舍五入为最接近的合法的离散值

Yoshida等人 2000 年将连续的值域分区间，每个区间赋予一个相应的离散值

其它形式

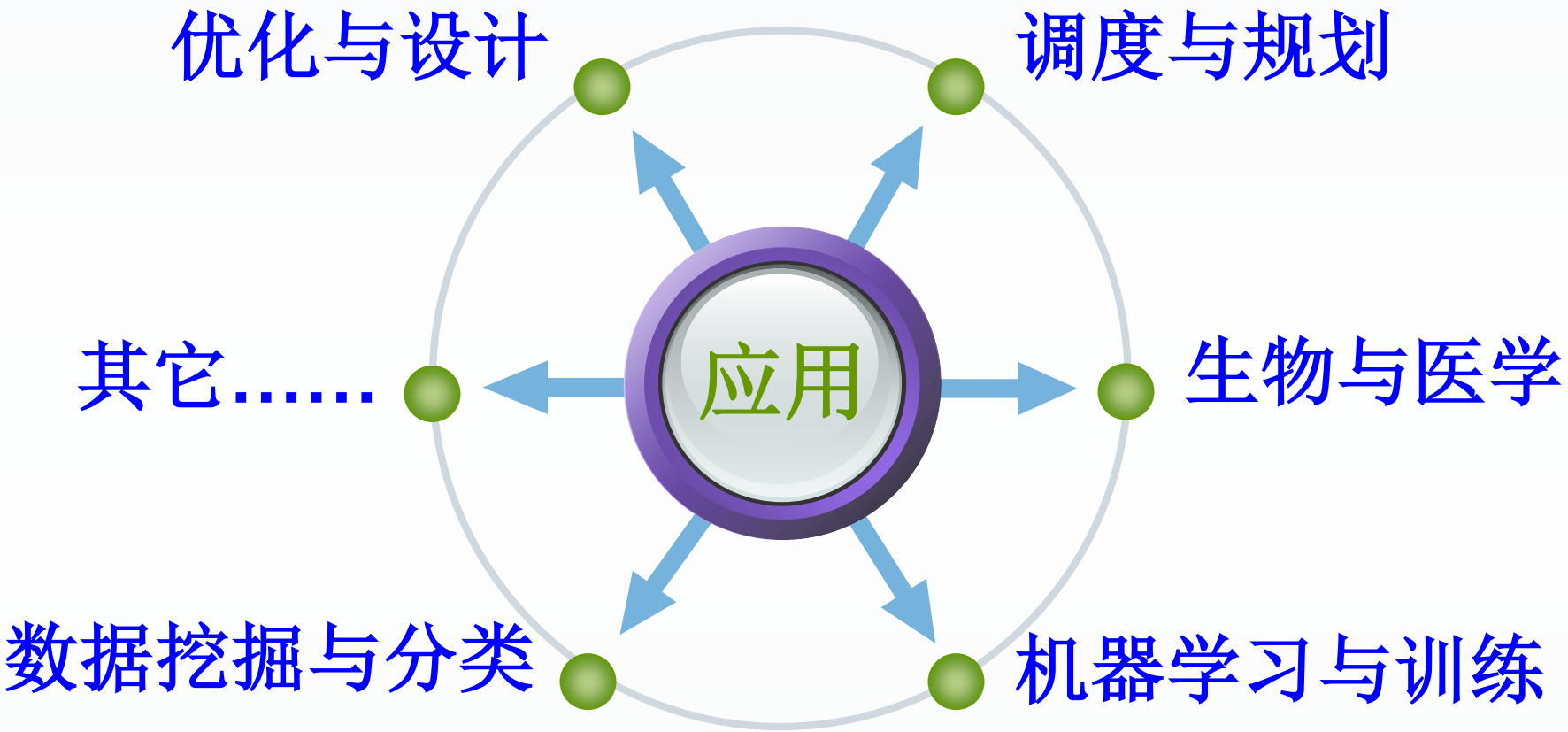
Schoofs和Naudts 2002 年重新定义了PSO的“加减乘”法，并且应用到了约束可满足问题（CSP）中

Hu等人2003 年将速度定义为位置变量相互交换的概率，从而将PSO离散化并用于解决*n*皇后问题

Clerc 2004 年为PSO定义了合适的“加减乘”法而实现离散化，并且应用于解决旅行商问题（TSP）

Chen等人2009年基于集合论的技术，重新定义了PSO速度和位置的更新公式实现了离散化

6.4 粒子群优化算法的相关应用



6.5 粒子群优化算法的参数设置

- 种群规模N
- 粒子的长度D
- 粒子的范围R
- 最大速度$V_{\max}$
- 惯性权重ω
- 压缩因子χ
- 加速系数c_1和c_2
- 终止条件
- 全局和局部PSO
- 同步和异步更新

种群规模*N*

- 影响着算法的搜索能力和计算量
  - **PSO对种群规模要求不高，一般取20-40就可以达到很好的求解效果**
  - **不过对于比较难的问题或者特定类别的问题，粒子数可以取到100或200**

- 粒子的长度D由优化问题本身决定，就是问题解的长度
- 粒子的范围R由优化问题本身决定，每一维可以设定不同的范围

最大速度V_{max}

- 决定粒子每一次的最大移动距离，制约着算法的探索和开发能力
  - V_{max}的每一维 V_{max}^{d} 一般可以取相应维搜索空间的10%-20%，甚至100%
  - 也有研究使用将V_{max}按照进化代数从大到小递减的设置方案

惯性权重ω

- 控制着前一速度对当前速度的影响，用于平衡算法的探索和开发能力
  - 一般设置为从0.9线性递减到0.4，也有非线性递减的设置方案
  - 可以采用模糊控制的方式设定，或者在[0.5, 1.0]之间随机取值
  - ω设为0.729的同时将c_1和c_2设1.49445，有利于算法的收敛

压缩因子χ

- 限制粒子的飞行速度的，保证算法的有效收敛
  - **Clerc等人通过数学计算得到χ取值0.729，同时c_1和c_2设为2.05**

加速系数c_1和c_2

- 代表了粒子向自身极值*pBest*和全局极值*gBest*推进的加速权值
  - c_1和c_2通常都等于2.0，代表着对两个引导方向的同等重视
  - 也存在一些c_1和c_2不相等的设置，但其范围一般都在0和4之间
  - 研究对c_1和c_2的自适应调整方案对算法性能的增强有重要意义

终止条件

- 决定算法运行的结束，由具体的应用和问题本身确定
  - 将最大循环数设定为500，1000，5000，或者最大的函数评估次数，等等
  - 也可以使用算法求解得到一个可接受的解作为终止条件
  - 或者是当算法在很长一段迭代中没有得到任何改善，则可以终止算法

全局和局部PSO

- 决定算法如何选择两种版本的粒子群优化算法—全局版PSO和局部版PSO
  - 全局版本PSO速度快，不过有时会陷入局部最优
  - 局部版本PSO收敛速度慢一点，不过不容易陷入局部最优
  - 在实际应用中，可以根据具体问题选择具体的算法版本

同步和异步更新

- 两种更新方式的区别在于对全局的*gBest*或者局部的*lBest*的更新方式
  - 在同步更新方式中，在每一代中，当所有粒子都采用当前的*gBest*进行速度和位置的更新之后才对粒子进行评估，更新各自的*pBest*，再选最好的*pBest*作为新的*gBest*
  - 在异步更新方式中，在每一代中，粒子采用当前的*gBest*进行速度和位置的更新，然后马上评估，更新自己的*pBest*，而且如果其*pBest*要优于当前的*gBest*，则立刻更新*gBest*，迅速将更好的*gBest*用于后面的粒子的更新过程中
  - 一般而言，异步更新的PSO具高效的信息传播能力，具有有更快的收敛速度


Thank You !