高级人工智能 群智能算法

Advanced Artificial Intelligence

Swarm intelligence algorithms

粒子群算法

粒子群算法概述

（Particle Swarm Optimization, PSO）

- Eberhart和Kennedy于1995年提出，优化算法
- 源于对鸟群觅食行为（一种复杂适应系统）的研究
- 4个基本特点
  - 个体是主动的、活动的
  - 个体与环境及其它个体相互有影响，这种影响是系统发展 变化的主要动力
  - 环境的影响是宏观的，个体间的影响是微观的，宏观与微观有机结合
  - 整个系统可能还要受一些随机因素的影响
- 具有进化计算和群智能的特点，较之遗传算法操作简单

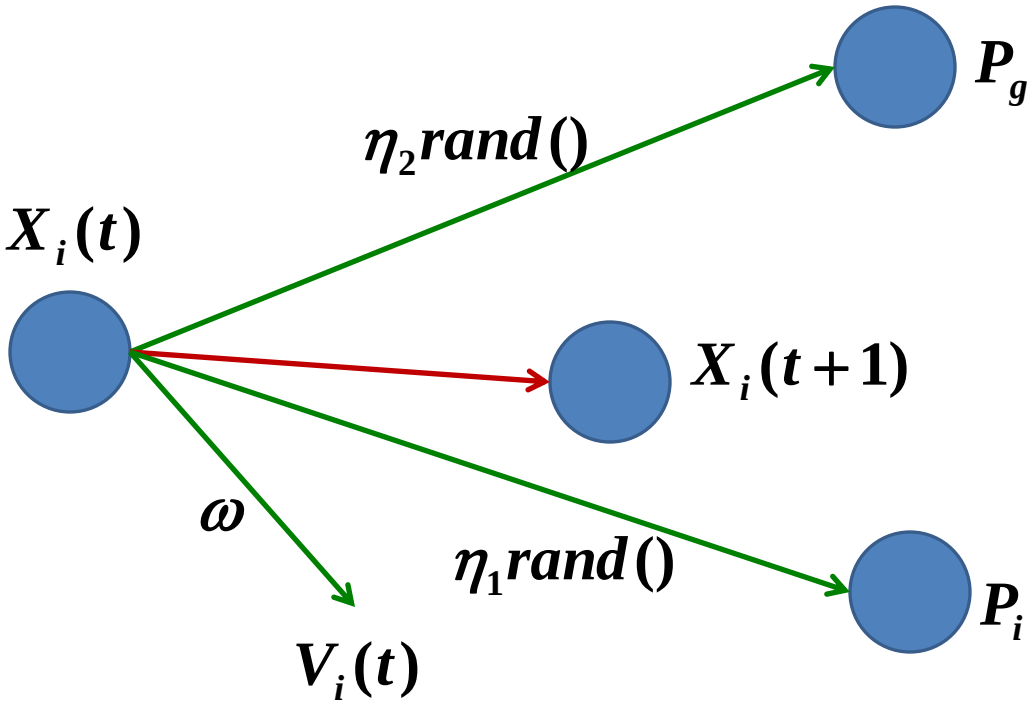
粒子群算法概述

- 可行解看成空中的一只鸟，即“粒子”
- 粒子在解空间中运动，通过速度公式确定飞行方向和距离
- 粒子追随当前的最优粒子在解空间中搜索：粒子本身找到的最优解+种群找到的最优解
- 鸟仅仅追踪有限数量的邻居但最终的结果是整个鸟群好像在一个中心的控制之下：复杂的全局行为是由简单规则的相互作用引起的。

蓝仔的十八班武艺 bilibili

如果从1出发需要经过所有点后

粒子群算法图示



搜索空间：n维

种群大小：N

t时刻粒子i的位置: $X_i(t)$

t时刻粒子i的速度: $V_i(t)$

粒子i的历史最优解:P_i

种群的历史最优解:P_g

粒子群算法的基本公式

符号说明：

搜索空间：n维

种群：$X=\{X_1,X_2,\cdots,X_N\}$

粒子i的位置：$X_i=(x_{i1},x_{i2},\cdots,x_{in})^T$

粒子i的历史最优解：$P_i=(p_{i1},p_{i2},\cdots p_{in})^T,i=1,2,\cdots,N$

种群的历史最优解：$P_g=(p_{g1},p_{g2},\cdots p_{gn})^T$

粒子i的速度：$V_i=(v_{i1},v_{i2},\cdots,v_{in})^T$

粒子群算法的基本公式（全局）

个人认知　　社会认知

$$V_i(t+1)=\omega V_i(t)+\eta_1 rand()(P_i-X_i(t))+\eta_2 rand()(P_g-X_i(t))$$

$$v_{id}(t+1)=\begin{cases} v_{\max}, v_{id}(t+1)>v_{\max} \\ -v_{\max}, v_{id}(t+1)<-v_{\max} \end{cases}, d=1,2,\cdots n$$

$$X_i(t+1)=X_i(t)+V_i(t+1)$$

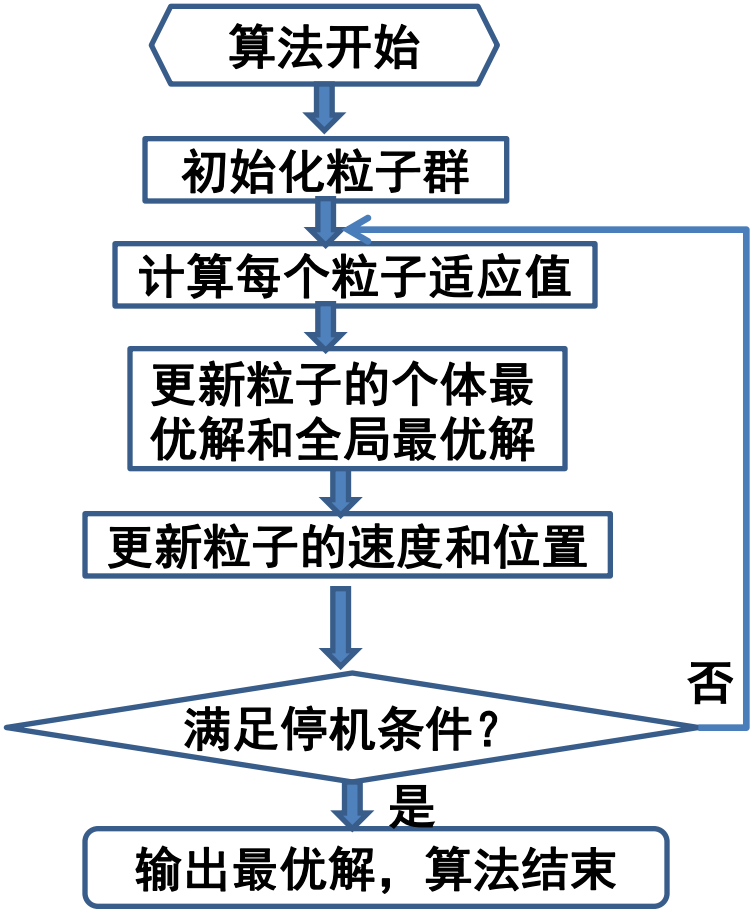
ω：惯性权重

η_1,η_2：加速常数

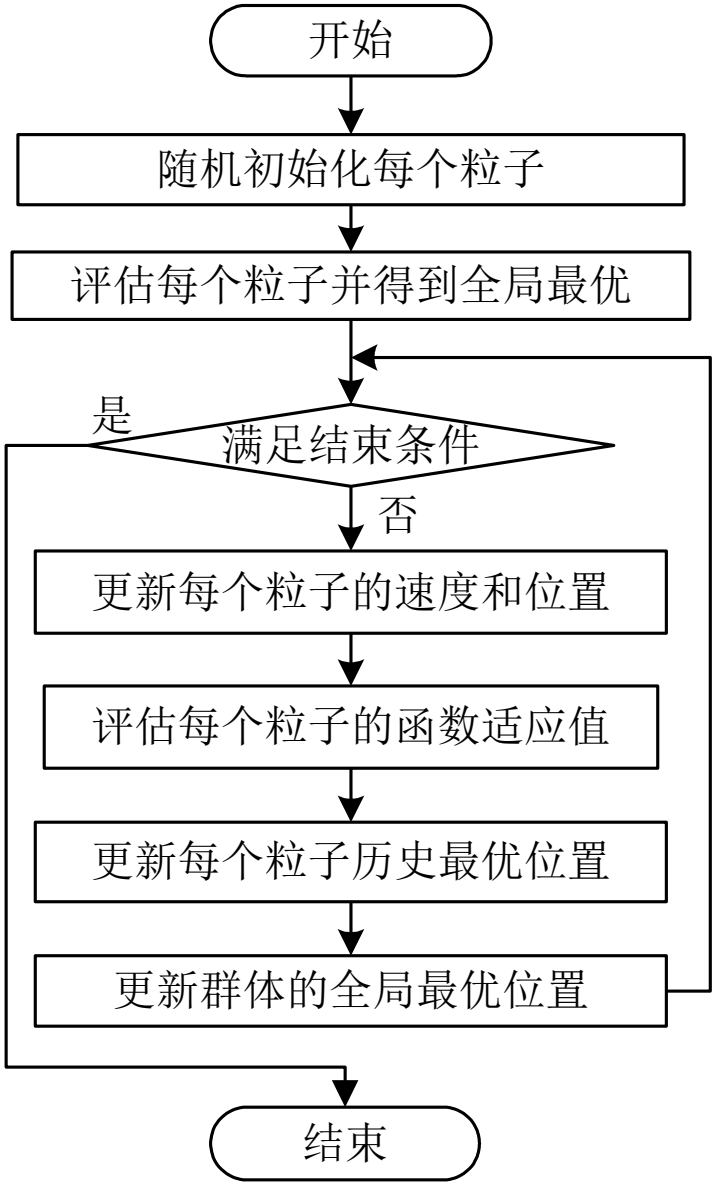
仅有个人认知：无信息共享，相当于单独n个粒子，难以找到最优解；

仅有社会认知：个体无信息认知，搜索速度可能更快，但易陷入局部最优。

粒子群算法流程图


算法开始
初始化粒子群
计算每个粒子适应值
更新粒子的个体最优解和全局最优解
更新粒子的速度和位置
满足停机条件？
否
是
输出最优解，算法结束

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
```

粒子群算法相关问题-1

搜索模式

- 全局模式：粒子运动受所有粒子状态的影响，追随局部及整体两个历史极值；收敛速度较快，群体多样性降低，但易陷入局部极值

- 局部模式：粒子运动只受自身认知和邻近粒子影响，不追随全局极值，收敛速度相对较慢，群体多样性较好，陷入局部极值的可能性相比全局模式较小。比全局模式增加了探索的几率

粒子群算法相关问题-2

参数设置

- ω(t)：0~1；初始设置较大，对解空间进行大范围探查(exploration)；后续较小，进行小范围求精(exploitation)。ω随着迭代过程趋向于0。
- 实验结果表明ω线性减少效果较好。

- $\eta_1(t)$:自我认知系数，≈2
- $\eta_2(t)$:社会认知系数，≈ 2
- 迭代初期一般$\eta_1(t) > \eta_2(t)$，后期$\eta_1(t) < \eta_2(t)$
- 研究$\eta_1(t)$和$\eta_2(t)$在迭代过程的自适应调节问题有助于提高算法的性能

- 有经验表明：ω设为0.729的同时将η_1和η_2设为1.49445，有利于算法的收敛

请解释参数设置的思路及对搜索的影响

粒子群算法相关问题-3

速度上限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；
- 较大时探索(exploration)能力较强，但可能错过最优解
- 较小时求精(exploitation)能力较强，但可能易陷入局部解
- 一般设为每维变量值域的10%~20%
- 也有研究将V_{max}按照进化代数从大到小递减的方案

种群规模

- 影响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就可达到很好的求解效果；
- 对于比较难的或特定类别的问题，粒子数可以取到100~200

粒子群算法相关问题-4

终止条件

- 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

粒子群算法相关问题-5

同步和异步更新

- 区别在于对全局*gBest*或局部*pBest*的更新方式
- 同步更新：每一代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*
- 异步更新：每一代粒子采用当前的*gBest*进行速度和位置的更新，然后马上评估，更新自己的*pBest*，且若其*pBest*优于当前的*gBest*，则立刻更新*gBest*，迅速将更好的*gBest*用于后面粒子的更新过程中
- 一般而言，异步更新的PSO具更高效的信息传播能力，收敛速度更快

局部模式粒子群算法

局部模式速度公式

- 每个粒子的行为受到其局部邻域粒子的影响。只追随自身极值P_i和邻域极值P_b，不追随全局极值P_g
- 速度更新公式：

$$V_i(t+1)=\omega V_i(t)+\eta_1 rand()(P_i-X_i(t))+\eta_2 rand()(P_b-X_i(t))$$

邻域的两类定义方式

- 以空间聚集定义而获得的P_b倾向于搜索状态空间中邻近的同质区域，搜索能力有限
- 以逻辑拓扑结构定义而获得的P_b可以搜索具有社会网络关系的邻域，邻域关系常用二值关系矩阵表达

局部模式粒子群拓扑结构

邻域结构的影响

- 邻域结构决定了粒子之间信息传递的方式、传递的速度与强度

- 粒子之间信息传递速度太快，系统易出现早熟；传递速度太慢，单个粒子难以迅速得到较远粒子的信息，造成算法收敛速度变慢，影响计算效率

- 不同的结构具有不同的特点和适用场合

典型的局部模式粒子群拓扑结构-1

常见的3种邻域结构

- 全连接形：也即全局模式。所有粒子相互均有连接。收敛速度快，有较强的局部搜索能力，但易陷入局部最优

- 星形：以其中一个粒子为中心，与邻域中其它所有粒子相联系，而其它粒子不进行信息交换。信息传递速度快，收敛速度也较快，但易陷入局部极值点

- 环形：所有粒子排成环状结构，结构中每个粒子与2个其它粒子相连。可有效保证种群的多样性，不易陷入局部最优，但搜索速度较慢，源于信息传递速度慢

典型的局部模式粒子群拓扑结构-2

3种邻域的折中方案

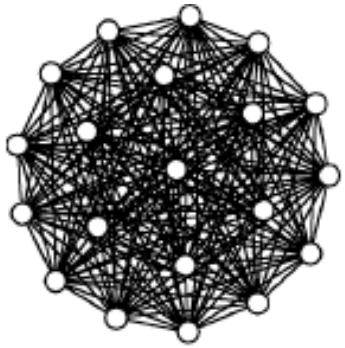
- **4簇结构：粒子分成4个子簇，子簇间通过网关连接**
- **金字塔形：四面体结构，粒子分布在四面体的四个顶点上，粒子间由三角形线框连接**
- **Square形（冯·诺依曼结构）：网格状,每个粒子与上下左右4个最邻近的粒子相互连接，类似花托形状。在对每个区域进行充分搜索的同时加强了粒子之间的信息交流**

术语

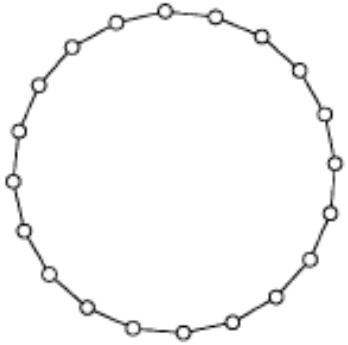
- **平均距离（Average Distance）:所有粒子之间的平均距离**
- **直径（Diameter）:粒子之间的最远距离。**

局部模式粒子群拓扑结构

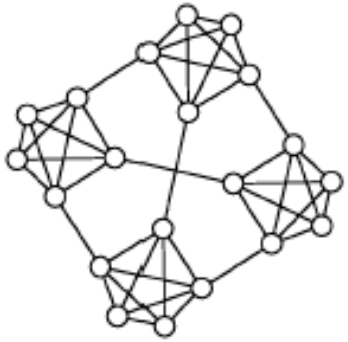
全连接



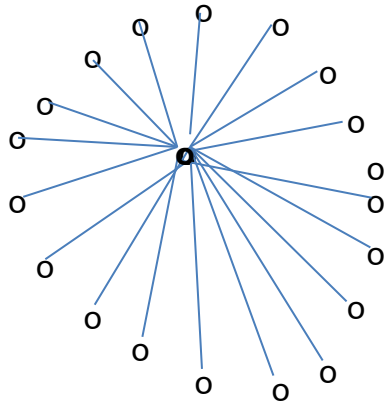
环状



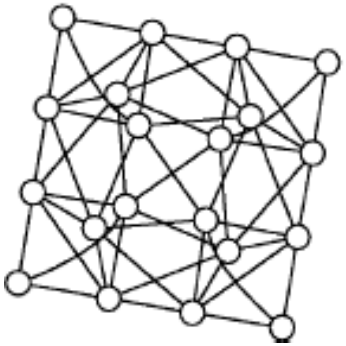
四簇



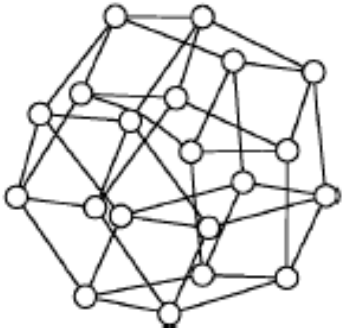
星形



金字塔



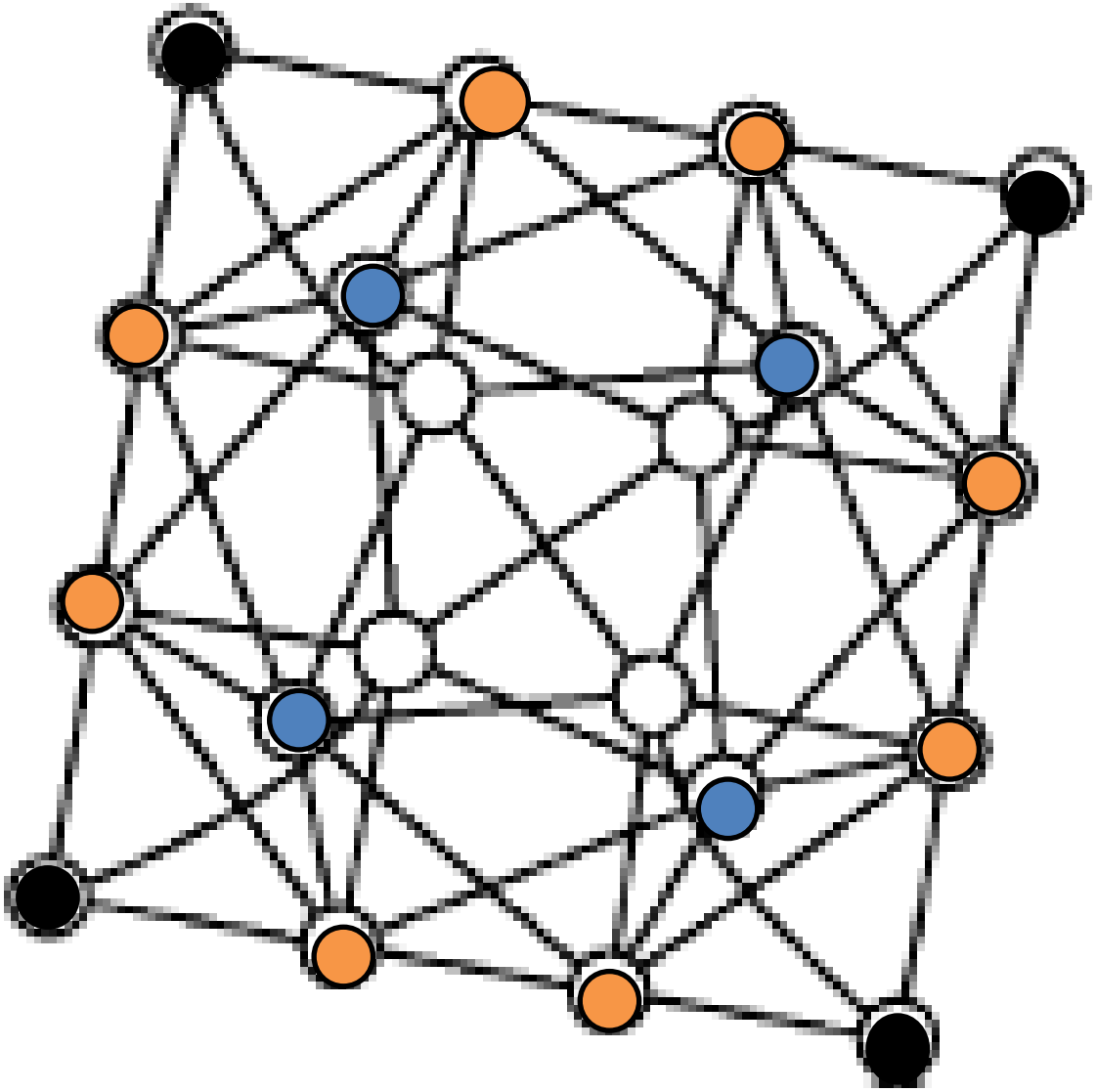
冯. 诺依曼

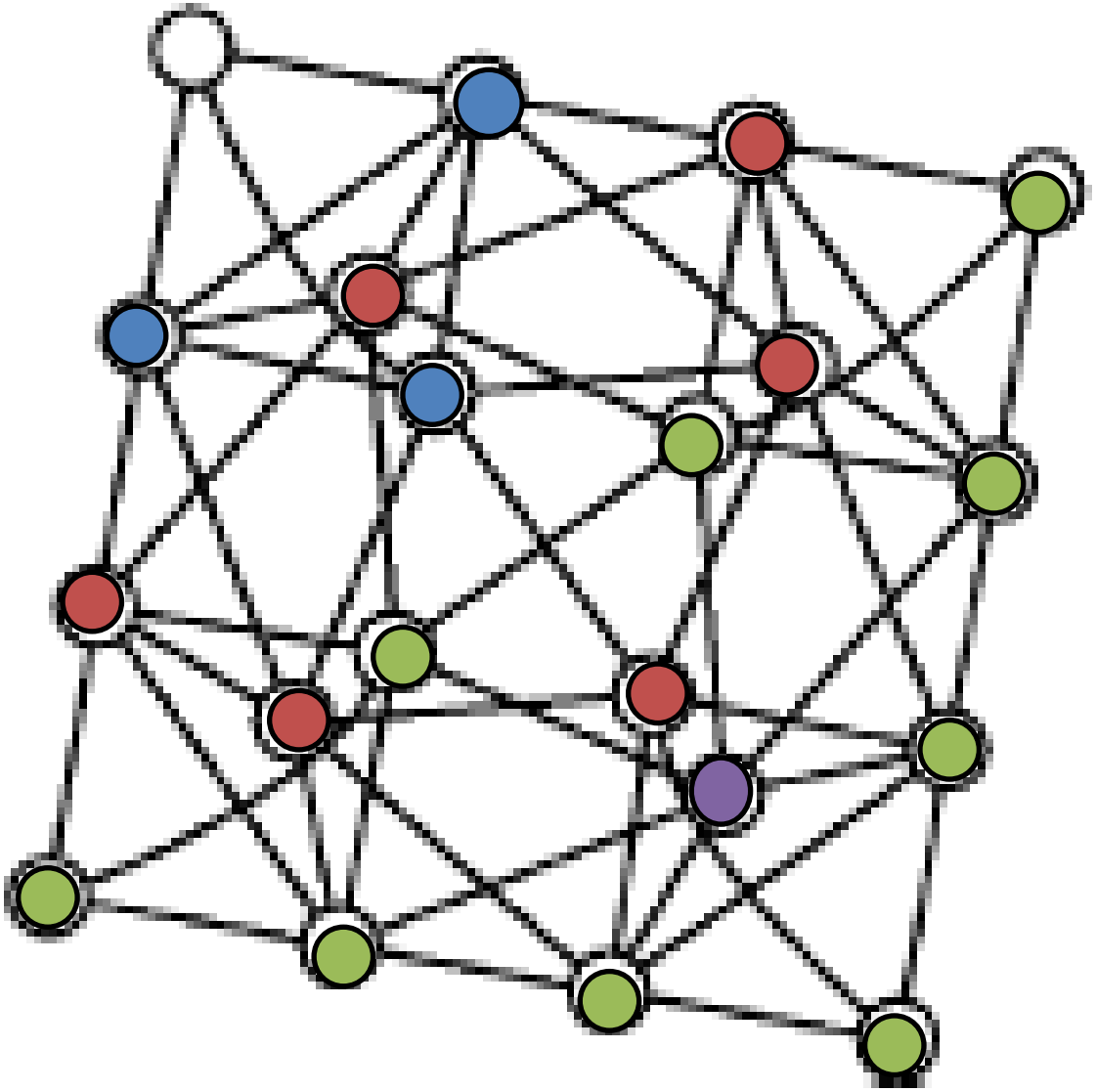


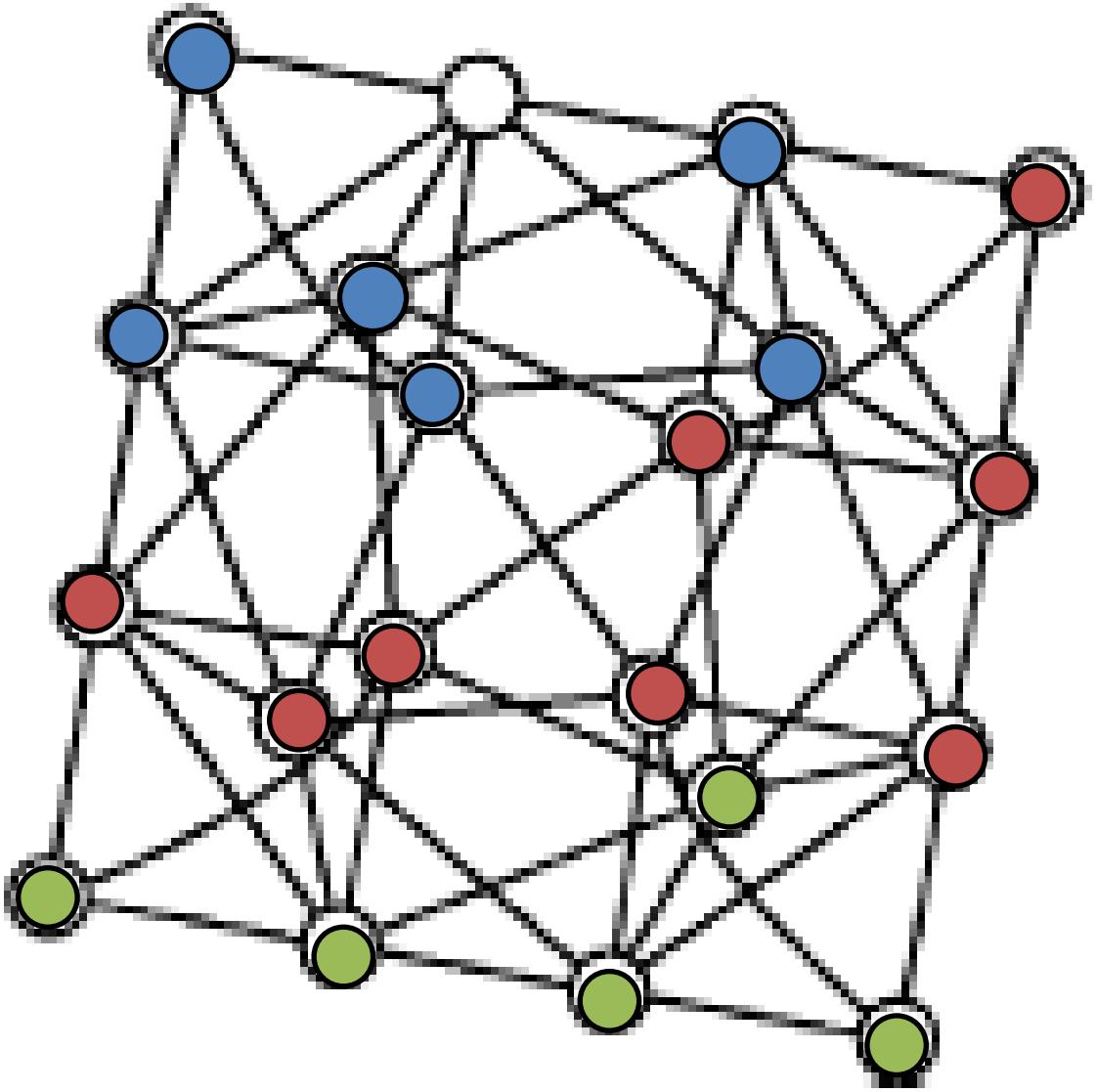
***N*=20时**

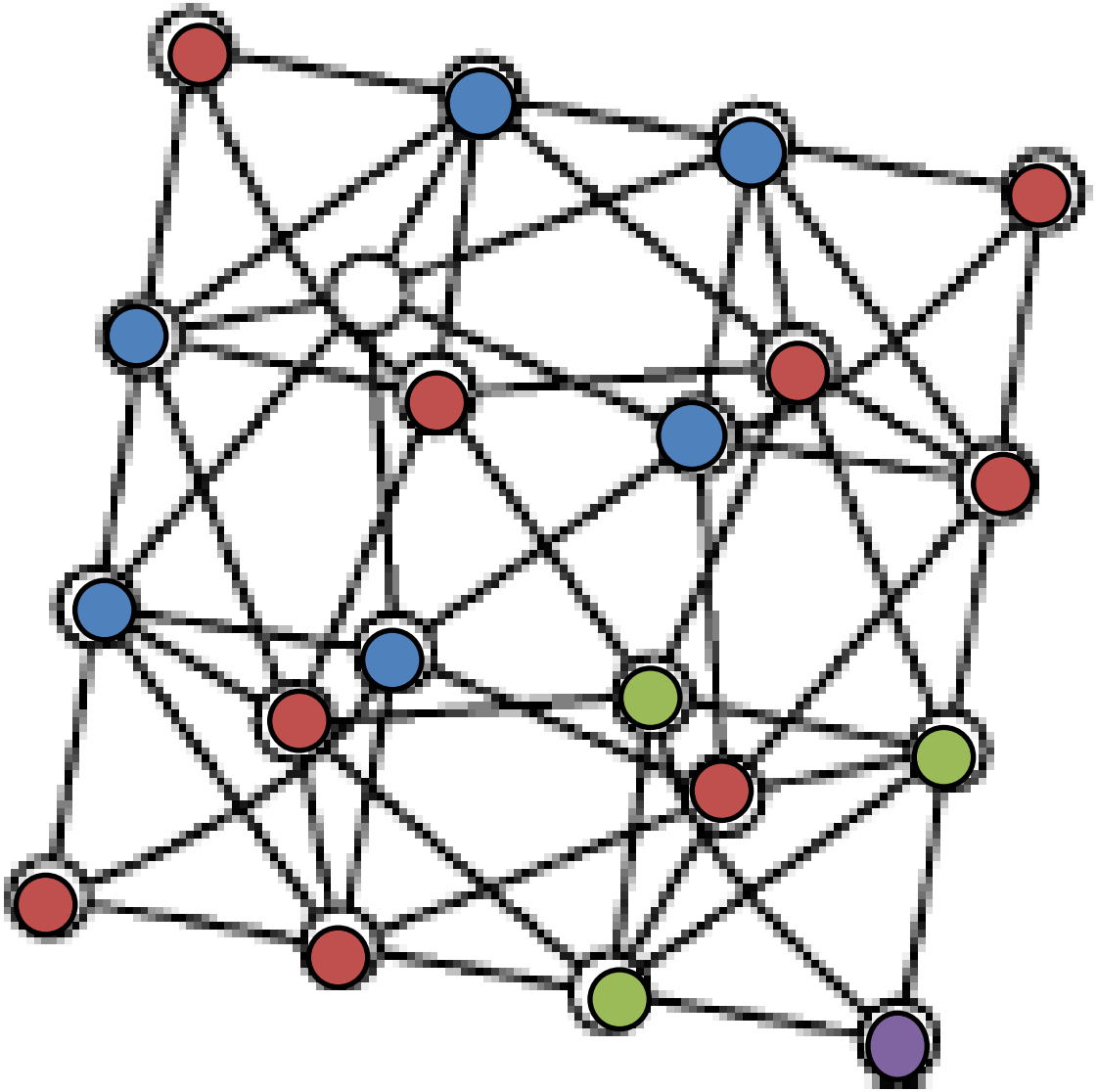
Topology	Average Distance	Diameter
All	1	1
Ring	5.26	10
Four Clusters	2.26	3
Pyramid	2.04	4
Square	2.32	4

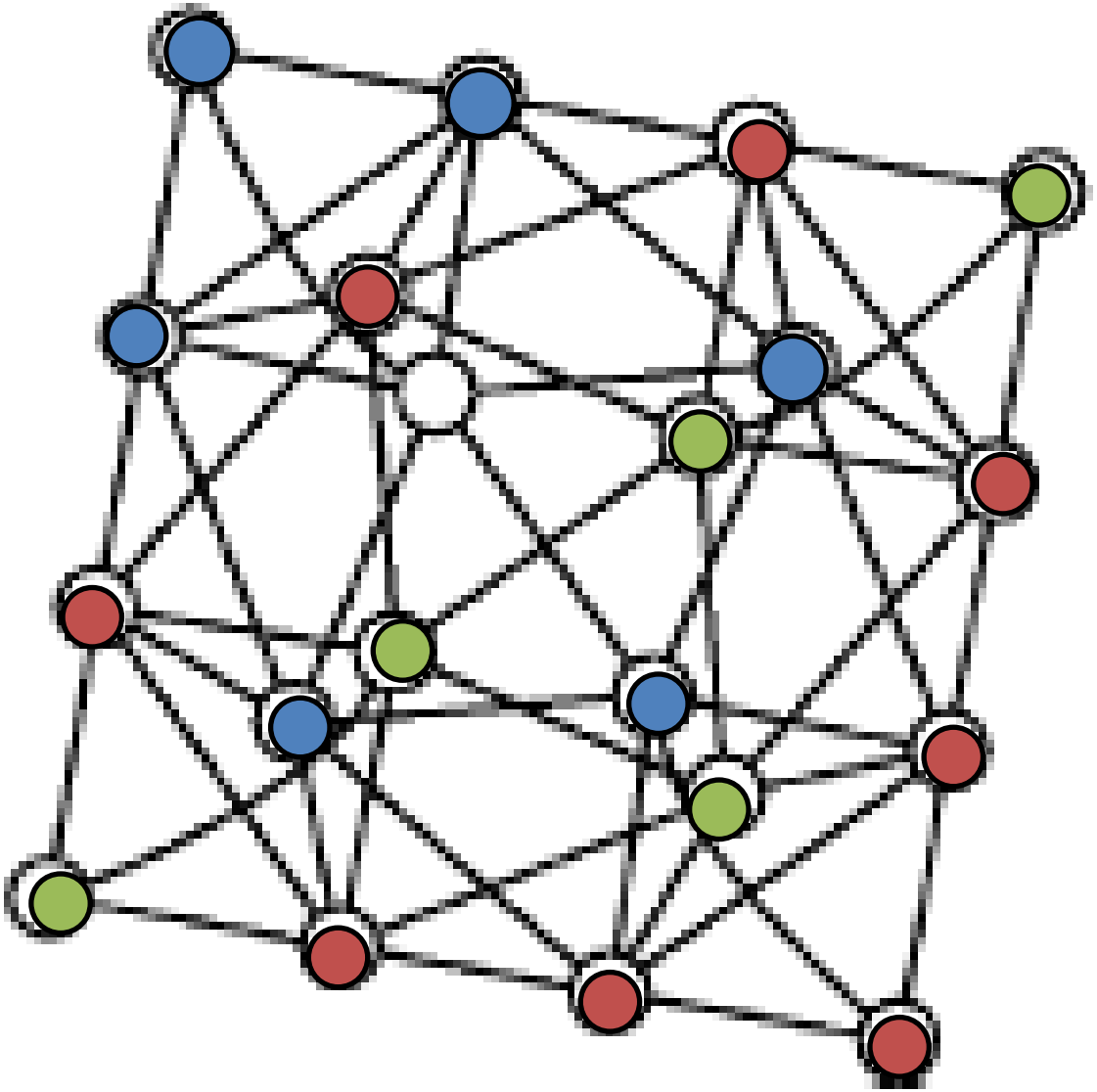
1）计算星形结构的AD值
2）思考Average Distance和Diameter的计算方法；
3）写出20个粒子的金字塔结构和冯诺依曼结构的邻接矩阵

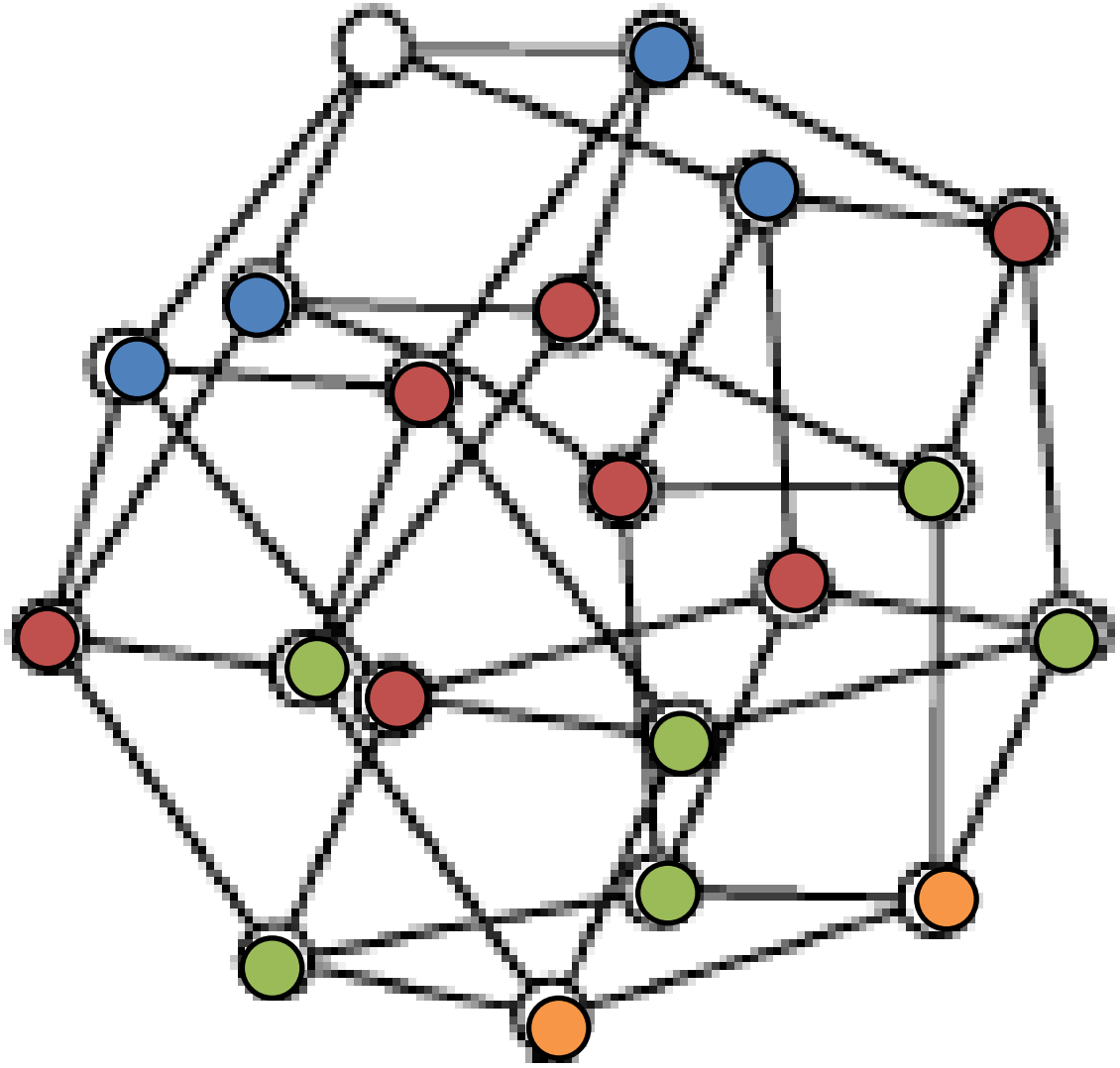












局部模式粒子群拓扑结构

- 结论：没有哪一种邻域结构适合于所有类型的测试函数。但冯·诺依曼结构具有最好的平均效果；
- 可通过数值实验寻找最合适的邻域结构，或随着计算进行，动态地改变邻域结构。

在解决具体问题的时候，可以遵循以下一些规律：

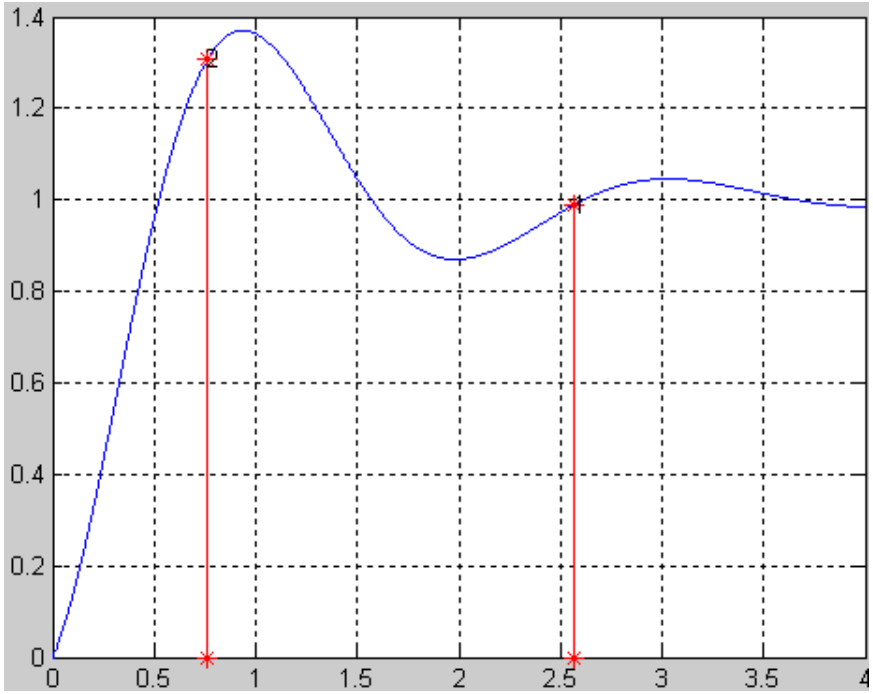
- 邻域较小的拓扑结构在处理复杂的、多峰值的问题上具有优势，例如环型结构的LPSO
- 随着邻域的扩大，算法的收敛速度将会加快，这对简单的、单峰值的问题非常有利，例如GPSO在这些问题上表现很好

粒子群算法举例

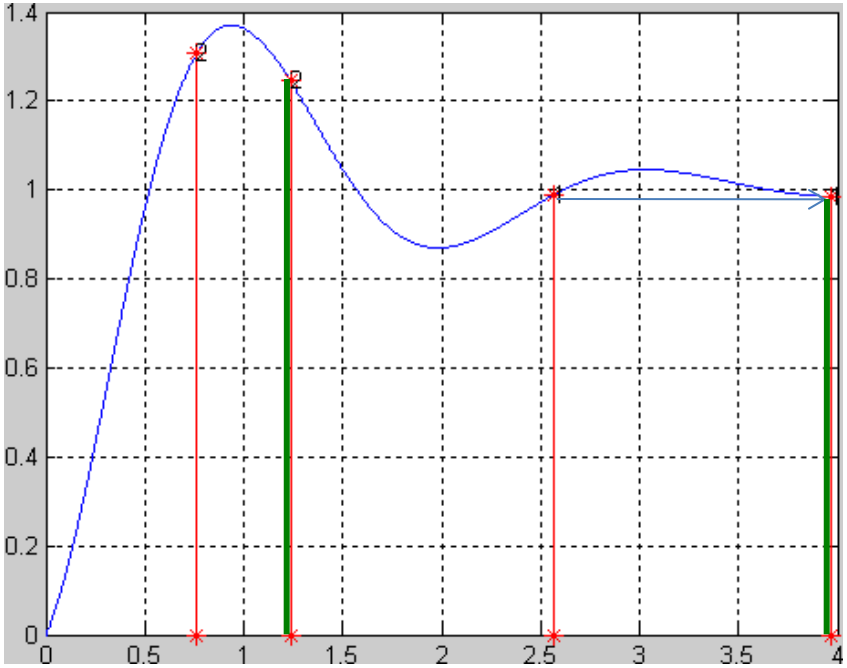
求函数y=1-*cos*(3**x*)**exp*(-*x*)在[0,4]最大值

- $x^* \approx 0.940$，$y_{max}=1.3706$
- 红色点：初始种群的粒子

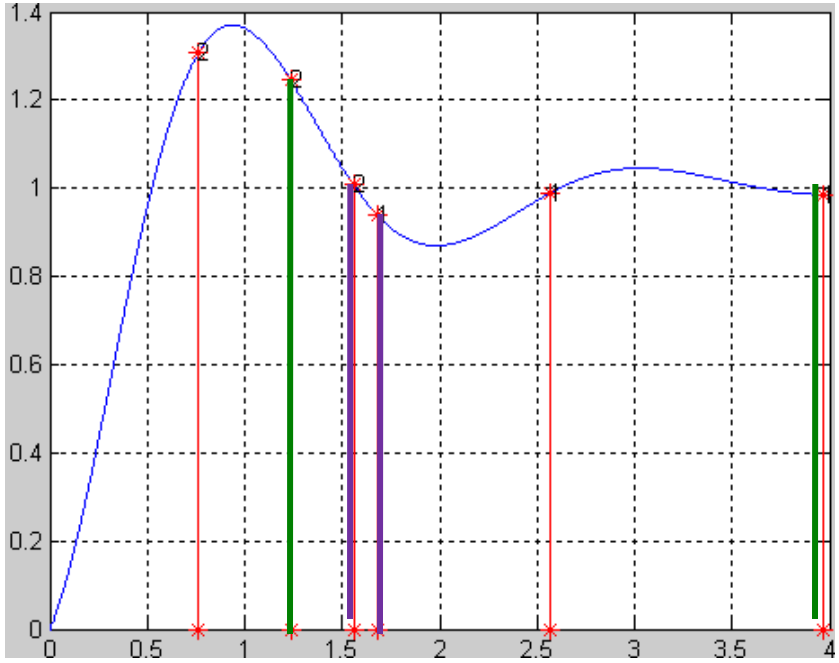
  $x1=2.52$, $x2=0.75$
- 函数的最大值即食物
- 适应值函数选择为y
- 粒子按位置速度公式更新
- 初始速度为0



粒子群算法举例

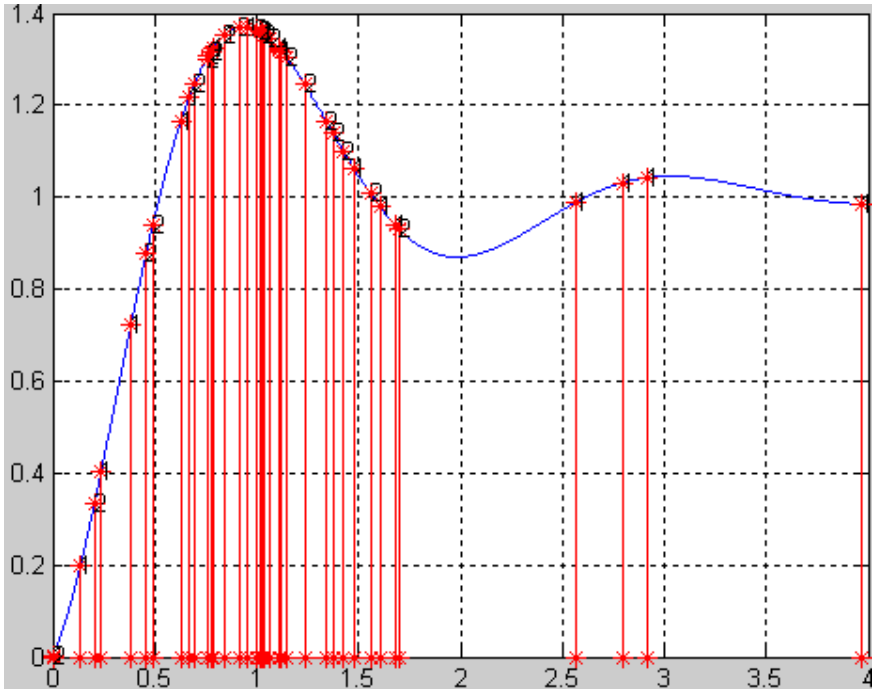


第一次更新

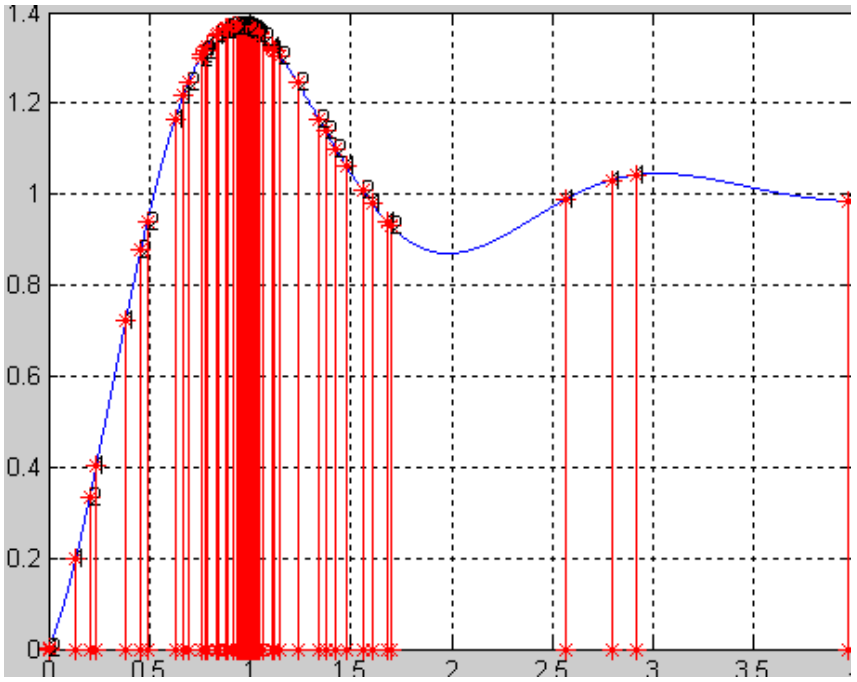


第二次更新

粒子群算法举例



第21次更新



第30次更新

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。

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

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

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

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

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。

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

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

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

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

对于越界的位置，需要进行合法性调整

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。

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

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

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

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

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

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

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

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。

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

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

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

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

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。

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

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

注意！ 对于越界的位置，需要进行合法性调整

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

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。

一种二进制编码的粒子群算法

Kennedy处理离散优化问题的二进制粒子群算法公式：

$$v_{ij}(t+1)=\omega v_{ij}(t)+\eta_1 r_{1j}(t)(x_{ij}^{(p)}(t)-x_{ij}(t))+\eta_2 r_{2j}(t)(x_j^{(g)}(t)-x_{ij}(t))$$

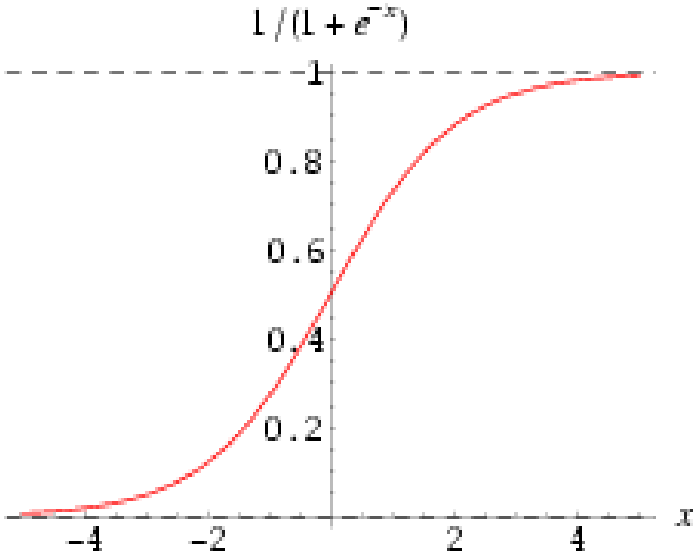
$$x_{ij}(t+1)=\begin{cases}1 & rand \le sig(v_{ij}(t+1))\\ 0 & rand > sig(v_{ij}(t+1))\end{cases}$$

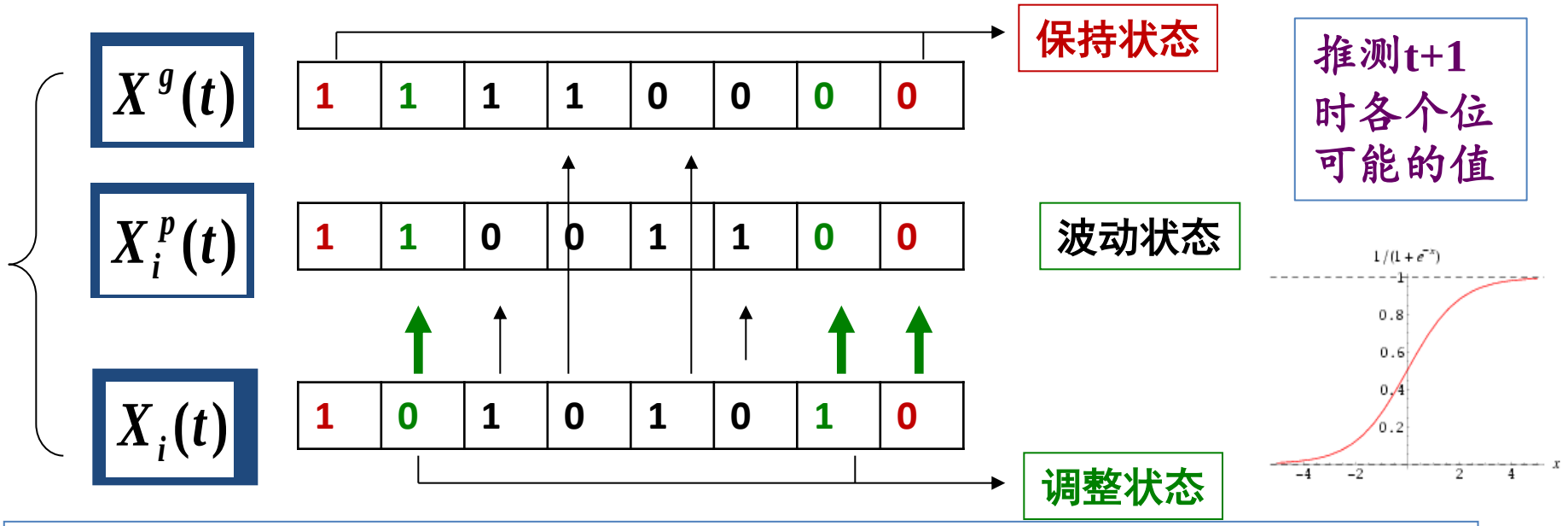
$i=1,2,\cdots,N$(种群大小)

$j=1,2,\cdots,n$（粒子维度）

$$f(x)=\frac{1}{1+e^{-x}}$$

$$f'(x)=f(x)\cdot(1-f(x))$$





$j=0$:（保持状态）

$x_0^{(g)}(t)=x_{i0}^{(p)}(t)=x_{i0}(t)=0 \Rightarrow v_{i0}(t+1)=\omega v_{i0}(t)$

$x_{i0}(t)=0 \rightarrow rand > sig(v_{i0}(t)) \rightarrow sig(v_{i0}(t))<0.5 \uparrow \rightarrow v_{i0}(t)<0 \uparrow$

$v_{i0}(t+1)=\omega v_{i0}(t)<0 \uparrow$

$x_{i0}(t+1)=0 \uparrow$

$$v_{ij}(t+1)=\omega v_{ij}(t)+\eta_1 r_{1j}(t)(x_{ij}^{(p)}(t)-x_{ij}(t))+\eta_2 r_{2j}(t)(x_j^{(g)}(t)-x_{ij}(t))$$

$$x_{ij}(t+1)=\begin{cases}1 & rand \le sig(v_{ij}(t+1)) \\ 0 & rand > sig(v_{ij}(t+1))\end{cases}$$

$j=1$:（调整状态）

$x_1^{(g)}(t)=x_{i1}^{(p)}(t)=0,\ \ x_{i1}(t)=1\Rightarrow v_{i1}(t+1)=\omega v_{i1}(t)-\eta_1 r_{11}(t)-\eta_2 r_{21}(t)$

$x_{i1}(t)=1\rightarrow rand\le sig(v_{i1}(t))\rightarrow sig(v_{i1}(t))>0.5\uparrow\rightarrow v_{i1}(t)>0\uparrow$

$v_{i1}(t+1)=\omega v_{i1}(t)-\eta_1 r_{11}(t)-\eta_2 r_{21}(t)<0\uparrow$

$x_{i1}(t+1)=0\uparrow$

$j=2$:（波动状态）

$x_2^{(g)}(t)=0,\ \ x_{i2}^{(p)}(t)=1,\ \ x_{i2}(t)=0\Rightarrow v_{i2}(t+1)=\omega v_{i2}(t)+\eta_1 r_{12}(t)$

$x_{i2}(t)=0\rightarrow rand>sig(v_{i2}(t))\rightarrow sig(v_{i2}(t))<0.5\uparrow\rightarrow v_{i2}(t)<0\uparrow$

$v_{i2}(t+1)=\omega v_{i2}(t)+\eta_1 r_{12}(t)$符号难判断

$x_{i2}(t+1)$取值有随机波动性

$$v_{ij}(t+1)=\omega v_{ij}(t)+\eta_1 r_{1j}(t)(x_{ij}^{(p)}(t)-x_{ij}(t))+\eta_2 r_{2j}(t)(x_j^{(g)}(t)-x_{ij}(t))$$

$$x_{ij}(t+1)=\begin{cases}1 & rand\le sig(v_{ij}(t+1))\\ 0 & rand>sig(v_{ij}(t+1))\end{cases}$$

粒子群算法求解TSP问题

如何定义轨迹，使得到的候选点为可行解？

交换子定义1：

设n个节点TSP问题的一个解序列为S = (a_{j1},… a_{jn})；交换子 SO($i1$, $i2$)定义为交换S中的点a_{i1}和 a_{i2}的操作，记 S’ = S + SO($i1$, $i2$) 为S经过SO($i1$, $i2$)操作后的新解。

例:（1 3 5 2 4) + SO(1, 2) = (3 1 5 2 4)

粒子群算法求解TSP问题

交换序定义2

一个或多个交换子的有序队列就是交换序，记作SS。

$$SS=(SO_1,SO_2,\ldots,SO_n),$$

其中SO_i是交换子。

S’= S +SS

=((S+ SO_1)+ SO_2)+…+ SO_{n-1})+SO_n

粒子群算法求解TSP问题

等价集合与合并算子

定义3：不同交换序可能产生同一个新解，所有产生同一个新解的交换序的集合称为交换序的等价集

定义4：若干交换序可以合并成一个新的交换序，用⊕标记两个交换序的合并算子。

定义5：在交换序的等价集中，拥有最少交换子的等价集称为基本交换序。

粒子群算法求解TSP问题

求基本交换序的一个例子

例：给定两条解路径A 和B，试构造一个基本交换序SS，使得B+SS=A 或 SS = A–B。

解：按照A的顺序逐次对B进行交换，直至得到A。

A: (1 2 3 4 5); B: (2 3 1 5 4)

B1= B+ SO(1, 3)=(1 3 2 5 4);

B2= B1+ SO(2, 3)=(1 2 3 5 4);

B3= B2+ SO(4, 5)= (1 2 3 4 5) =A

SS=A-B=(SO(1, 3), SO(2, 3), SO(4, 5))

粒子群算法求解TSP问题

变形后的PSO速度公式（全局模式）

$$V_i(t+1) = V_i(t) \oplus \alpha(P_i - X_i(t)) \oplus \beta(P_g - X_i(t))$$

$$X_{i+1}(t) = X_i(t) + V_i(t+1)$$

- V_i是一个交换序，α,β为[0，1]间的随机数
- α(P_i-X_i)表示基本交换序(P_i-X_i)中的所有交换子以概率α保留；α的值越大，(P_i-X_i)保留的交换子越多，P_i的影响越大；
- β(P_g-X_i)表示基本交换序(P_g-X_i)中的所有交换子以概率β保留; β的值越大，(P_g-X_i)保留的交换子越多，P_g的影响越大；

一种求解TSP问题的PSO 算法步骤

1）初始化粒子群：赋给每个粒子随机的初始解和交换序

2）若满足结束条件, 转步骤5)

3）根据粒子当前位置$X_i(t)$, 计算其下一个位置$X_i(t+1)$, 即新解

- 计算基本交换序$A = P_i - X_i$, 表示A 作用于X_i得到P_i
- 计算基本交换序$B = P_g - X_i$
- 根据式 $V_i(t+1) = V_i(t) \oplus \alpha(P_i - X_i(t)) \oplus \beta(P_g - X_i(t))$ 计算速 度$V_i(t+1)$
- 计算搜索到的新解 $X_i(t+1) = X_i(t) + V_i(t+1)$
- 若找到一个更好的解, 则更新P_i

4）若整个群体找到一个更好的解, 更新P_g ； 转步骤2）

5）显示结果

粒子群算法求解TSP问题实验分析

Table 1 TSP with 14 nodes

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Coordinate X	16.47	16.47	20.09	22.39	25.23	22.00	20.47	17.20	16.30	14.05	16.53	21.52	19.41	20.09
Coordinate Y	96.10	94.44	92.54	93.37	97.24	96.05	97.02	96.29	97.38	98.12	97.38	95.59	97.13	94.55

Table 2 Analyses of the algorithm performance

Size of solution space	(14– 1)! /2= 3 113 510 400
Number of particles in the swarm	100
Average number of iterations	20 000
Average size of search space	20 000* 100= 2 000 000
Search space/solution space	2 000 000/3 113 510 400= 0.064%
Best solution of the algorithm	1→10→9→11→8→13→7→12→6→5→4→3→14→2
Length	30.878 5(Equal to the best known result in the world)

2003

粒子群算法求解TSP问题实验分析

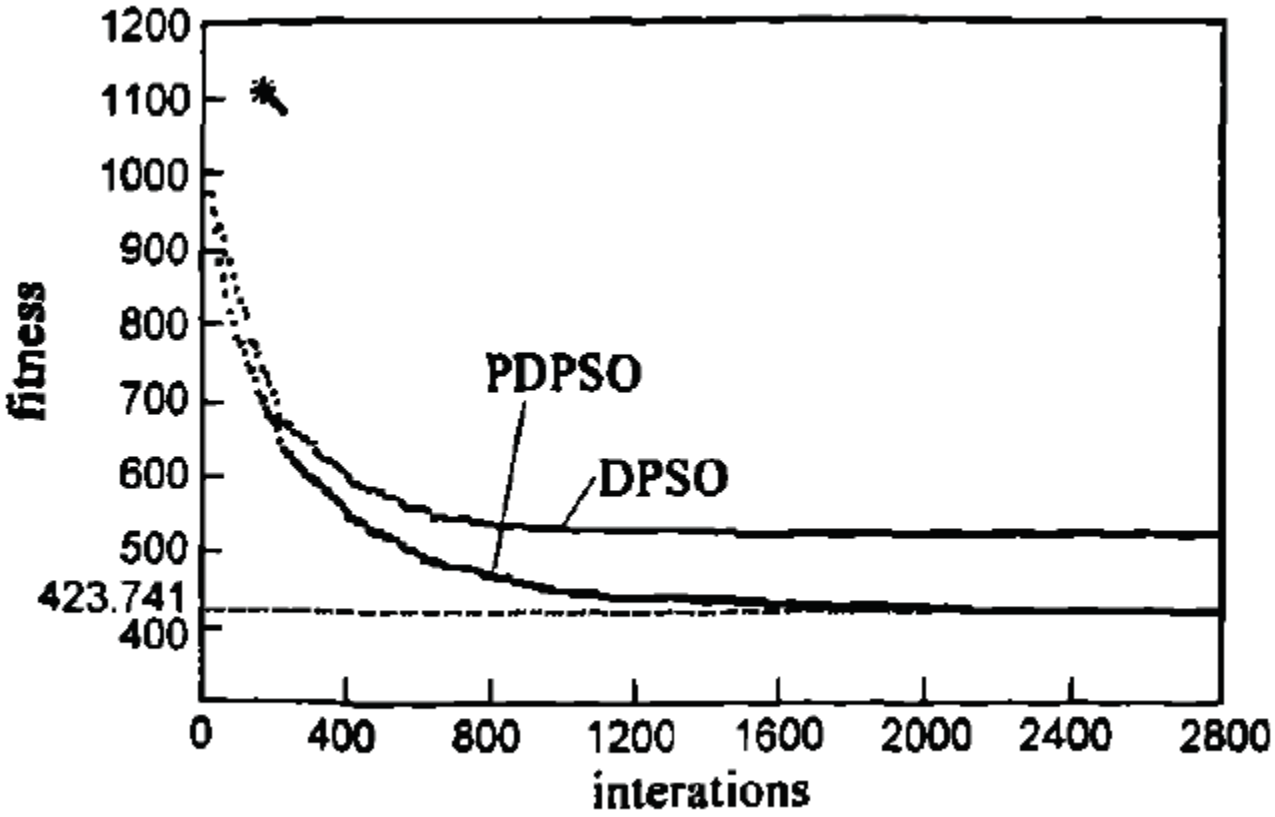


图3 Oliver30上算法的收敛曲线

2009
电子学报

粒子群算法求解TSP问题实验分析

表 2　TSP 问题的试验结果

问题	算法	得到的最优解	平均值	问题最优解
burma14.tsp	SADPSO	30.8785	30.8785	30.8785
	DPSO	30.8785	31.5551	
	ACO	30.8785	31.4075	
Oliver30.tsp	SADPSO	423.7410	424.8267	423.7406
	DPSO	453.4200	515.4413	
	ACO	434.2214	447.9351	
eil51.tsp	SADPSO	436.7730	440.7810	426.0000
	DPSO	476.9910	523.5421	
	ACO	449.6437	454.3333	

粒子群算法若干改进策略

- **Peram : Fitness-Distance-Ratio based Particle Swarm Optimization，FDRPSo，每个粒子根据一定的适应值-距离-比例原则，向附近具有较好适应值的多个粒子进行不同程度的靠近，而不仅只向当前发现的最好粒子靠近**
- **Bergh : Cooperative Particle Swarm Optimizer，CPSO，采用多个协同工作的子粒子群对解向量的不同部分分别进行优化，达到较好的寻优结果.**
- **Liang, Suganthan : Comprehensive Learning Particle Swarm Optimizer，CLPSO，每个粒子的速度更新基于所有其它粒子的历史最优位置，从而达到综合学习的目的.**