7. 粒子群优化

堵威

华东理工大学 自动化系

2021.4.22

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

本章内容

1. **背景知识**
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

背景知识

- 群体行为
  - 自然系统中能普遍观察到群体智能（觅食、捕食等）
  - 动物群常常比独行的动物更有效地避开捕食者，捕食者很难盯住混在一大群动物中的单只动物——捕食者混淆效应
  - 当一大群动物在觅食/饮水时，随机效应决定了始终有几只动物在监视捕食者——多眼假说
  - 动物群比独行的动物移动得更快：骑单车的人骑成一排，跟在后面的人比领头人少花40%的能量；成群的鱼，成群的大雁

背景知识

- 群体行为

鸟群行为

鱼群行为

背景知识

- 鸟的智能

背景知识

- 鸟的智能

背景知识

- 鸟的智能

背景知识

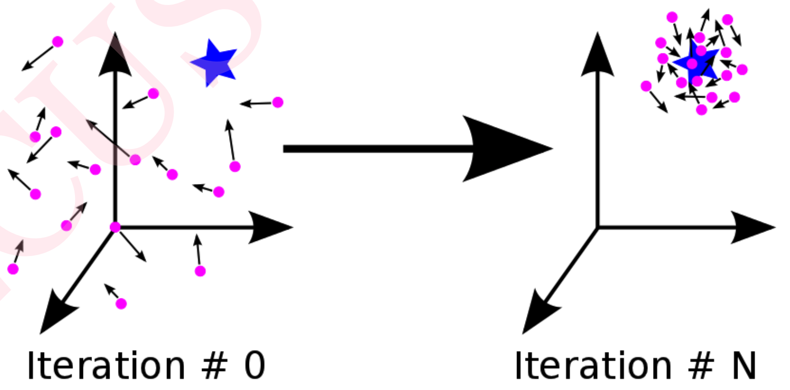
- 群体行为
  - 一群个体合作不仅能改进它们在某件任务上的集合性能，而且还能提高每一个个体的性能——粒子群优化的基础
  - 惯性：保留在过去已证明是成功的那些旧的方式
  - 受社会的影响：受他人启发
  - 受邻居的影响：从亲近的人那里学到的多，受朋友的影响，修正我们的行为

本章内容

1. 背景知识
2. **基本粒子群优化算法**
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

基本粒子群优化算法

- 基本框架
  - 定义d维连续域上的一个最小化问题，以及包含N个候选解的种群，记为{$\mathbf{x}_i$}, i∈[1,N]
  - 粒子群优化的精髓：每一个个体$\mathbf{x}_i$（粒子）以某个速度$\mathbf{v}_i$在搜索空间中移动；与其他进化算法不同，一代到下一代的建模，空间中移动的动态建模

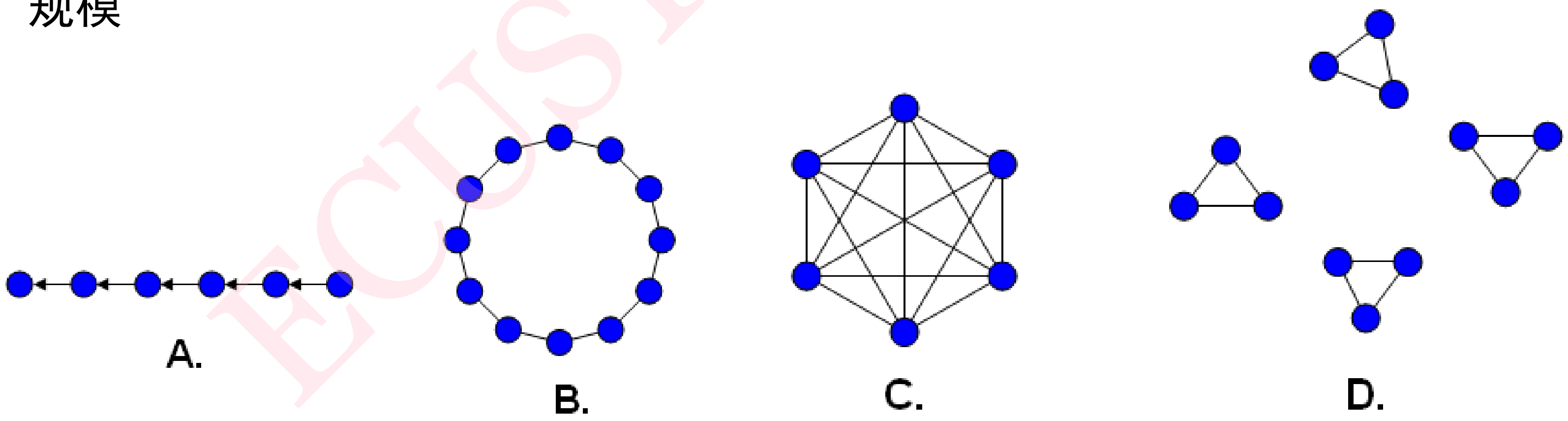


基本粒子群优化算法

- 基本框架
  - 当粒子在搜索空间中移动时，因为存在某种惯性，易于保持自己的速度，但是其速度也会受到一些因素的影响：

1）过去最好的位置：过去最好的位置影响着搜索

2）当前邻居的最好位置：邻居的最好位置也影响着搜索；定义邻居的规模



基本粒子群优化算法

- 基本框架
  - 最小化n元函数f(**x**)，**x**$_i$是第i个候选解，**v**$_i$是速度向量

初始化一个随机个体的种群 $\{\boldsymbol{x}_i\}$, $i \in [1, N]$
初始化每一个个体的 n 元速度向量 $\boldsymbol{v}_i, i \in [1, N]$
初始化每一个个体到目前为止最好的位置: $\boldsymbol{b}_i \leftarrow \boldsymbol{x}_i$, $i \in [1, N]$
定义邻域规模 $\sigma < N$
定义最大影响值 $\phi_{1,\max}$ 和 $\phi_{2,\max}$
定义最大速度 $\boldsymbol{v}_{\max}$
While not (终止准则)
    For 每一个个体 $\boldsymbol{x}_i$, $i \in [1, N]$
        $H_i \leftarrow \{\boldsymbol{x}_i$ 最近的 σ 个邻居 $\}$
        $\boldsymbol{h}_i \leftarrow \underset{\boldsymbol{x}}{\operatorname{argmin}}\{f(\boldsymbol{x}) : \boldsymbol{x} \in H_i\}$
        生成一个随机向量 $\boldsymbol{\phi}_1$, $\phi_1(k) \sim U[0, \phi_{1,\max}]$, $k \in [1, n]$
        生成一个随机向量 $\boldsymbol{\phi}_2$, $\phi_2(k) \sim U[0, \phi_{2,\max}]$, $k \in [1, n]$

速度更新
        $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{\phi}_1 \circ (\boldsymbol{b}_i - \boldsymbol{x}_i) + \boldsymbol{\phi}_2 \circ (\boldsymbol{h}_i - \boldsymbol{x}_i)$
        If $|\boldsymbol{v}_i| > \boldsymbol{v}_{\max}$ then[1]
                $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i v_{\max} / |\boldsymbol{v}_i|$
        End if

位置更新
        $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i$
        $\boldsymbol{b}_i \leftarrow \arg\min\{f(\boldsymbol{x}_i), f(\boldsymbol{b}_i)\}$
    下一个个体
下一代

基本粒子群优化算法

- 基本框架
  - 速度更新：

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{\phi}_1 \circ (\boldsymbol{b}_i - \boldsymbol{x}_i) + \boldsymbol{\phi}_2 \circ (\boldsymbol{h}_i - \boldsymbol{x}_i)$$

其中Φ_1和为Φ_2学习率，Φ_1为认知学习率，Φ_2为社会学习率，Φ_1,Φ_2为分布在[0,$\Phi_{1,max}$]和[0,$\Phi_{2,max}$]上的随机数；$\boldsymbol{b}_i$ 为每一个粒子迄今最好的位置，$\boldsymbol{h}_i$ 当前邻居的最好位置。

  - 最大速度：

$\mathbf{v}_{max}$的设置需考虑搜索空间动态范围，如果$\mathbf{v}_{max}$大于搜索空间的动态范围，在一代之内粒子很容易离开搜索空间，可以建议设置在搜索空间范围的10%和20%之间。

基本粒子群优化算法

- 基本框架
  - 位置更新：

$$x_i \leftarrow x_i + v_i$$

速度叠加在当前位置上，完成位置更新；若$\boldsymbol{x}_i$超出搜索域，需要进行位置重置处理。

  - 初始化：

除了初始化种群{$\mathbf{x}_i$}，i∈[1,N]，还需要初始化每一个粒子的n元速度向量$\mathbf{v}_i$, i∈[1,N]；初始化速度的方式：随机初始化、初始化为0。

  - 邻域规模σ：

σ<N；有些含糊，可以指每一个粒子有σ个近邻；也可以指在邻域内总共有σ个粒子，每一个粒子有(σ-1)个近邻；

基本粒子群优化算法

- 粒子群的拓扑
  - 每一个粒子会受到离它最近的σ个邻居的影响，影响粒子的邻居的布置成为群的拓扑
  - 上述算法中每一个粒子的邻域在每一代都会改变，因此称为动态拓扑；邻域是局部的（不包括整个种群），因此也称为lbest拓扑
  - 定义每一个粒子邻域的方法众多：

1）算法运行前事先定义，静态邻域；

2）优化停滞时，随机重新定义邻域；

3）全拓扑/gbest拓扑：极端情况下，采用包围整个种群的单个邻域（粒子群优化中最早提出的拓扑，至今仍广泛使用）；

基本粒子群优化算法

- 粒子群的拓扑

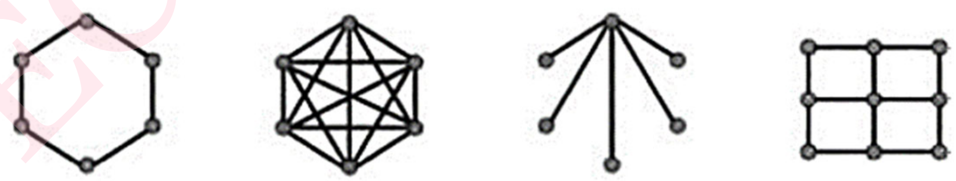
  – 定义每一个粒子邻域的方法众多：

4）环形拓扑：每一个粒子与另外两个粒子连接（常用）；

5）簇拓扑：每一个粒子与簇中的其他粒子全连接，同时在每一个簇中有少量的粒子会与另一个簇中的粒子连接；

6）轮胎拓扑：焦点粒子与所有粒子连接，所有粒子只与焦点粒子相连；

7）冯诺依曼拓扑/方形拓扑：每一个粒子与它的4个邻居相连。



环形拓扑　　全拓扑　　轮胎拓扑　　冯诺依曼拓扑

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. **速度限制**
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

速度限制

- v_{max}的作用
  - 在粒子群优化应用中，如果不用v_{max}，会出现粒子在搜索空间中乱窜的情形
  - 为了探究其中的原因，考虑基本粒子群算法中Φ_2 = 0时的简化版，此时，位置和速度更新为：

$$\left.\begin{aligned} \boldsymbol{v}_i(t+1) &= \boldsymbol{v}_i(t) + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i(t)), \\ \boldsymbol{x}_i(t+1) &= \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1), \end{aligned}\right\}$$

其中t为代数， Φ_1为认知学习率，上式可以写成：

$$\begin{bmatrix} \boldsymbol{x}_i(t+1) \\ \boldsymbol{v}_i(t+1) \end{bmatrix} = \begin{bmatrix} 1-\phi_1 & 1 \\ -\phi_1 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_i(t) \\ \boldsymbol{v}_i(t) \end{bmatrix} + \begin{bmatrix} \phi_1 \\ \phi_1 \end{bmatrix} \boldsymbol{b}_i.$$

速度限制

- v_{max}的作用
  - 方程右边矩阵的特征值为：

$$\lambda = \frac{2-\phi_1 \pm \sqrt{\phi_1^2 - 4\phi_1}}{2}$$

特征值支配着系统的稳定性。

若$\Phi_1 \in [0, 4]$，则两个特征值长度均为1，意味着系统临界稳定，且当$t \to \infty$，在某种初始条件下，$\mathbf{x}_i(t)$和$\mathbf{v}_i(t)$可能会变得无穷大；

若$\Phi_1 > 4$，则有一个特征值大于1，意味着系统不稳定，$\mathbf{x}_i(t)$和$\mathbf{v}_i(t)$对于几乎所有的初始条件都会无止境地增大；

上述简单的讨论说明用v_{max}来限制$\mathbf{v}_i$的大小非常重要。

速度限制

- v_{max}的作用
  - 如果要限制速度，可以采用几种不同的方式：

1）检查v_i的大小，如果大于v_{max}，就按比例缩小v_i的元素：

$$\boldsymbol{v}_i \leftarrow \frac{\boldsymbol{v}_i v_{\max}}{|\boldsymbol{v}_i|}$$

2）限制v_i的每一个元素的大小，种群中每一个个体都有n维，所以

$$\boldsymbol{v}_i = [v_i(1), v_i(2), \cdots, v_i(n)]$$

对第i个粒子的速度做如下限制：

$$v_i(j) \leftarrow \begin{cases} v_i(j), & \text{如果 } |v_i(j)| \leqslant v_{\max}(j), \\ v_{\max}(j)\operatorname{sign}(v_i(j)), & \text{如果 } |v_i(j)| > v_{\max}(j), \end{cases} \quad j \in [1, n]$$

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. **惯性权重与压缩系数**
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

惯性权重与压缩系数

- 惯性权重
  - 根据速度更新公式：

$$v_i(k) \leftarrow v_i(k) + \phi_1(k)(b_i - x_i(k)) + \phi_2(k)(h_i(k) - x_i(k)), \quad k \in [1, n],$$

尽管学习率允许粒子的速度从一代到下一代有一些变化，但粒子趋向于保持其速度。

根据以往的经验，在优化过程中减小惯性可能会得到更好的性能：

$$v_i(k) \leftarrow wv_i(k) + \phi_1(k)(b_i(k) - x_i(k)) + \phi_2(k)(h_i(k) - x_i(k)),$$

其中，ω为惯性权重，经常从第一代的约0.9减小到最后的0.4左右。当代数增加，惯性权重有助于降低每一个的速度从而改善收敛。

惯性权重与压缩系数

- 压缩系数

  – 惯性权重常常通过压缩系数实施，压缩系数能够实现惯性权重相同的功能：

$$\boldsymbol{v}_i \leftarrow K[\boldsymbol{v}_i + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i) + \phi_2(\boldsymbol{h}_i - \boldsymbol{x}_i)],$$

其中K被称为压缩系数（如果$K=\omega$且上式中的Φ_1和Φ_2分别由Φ_1/K和Φ_2/K替换，则等于前一页的惯性权重公式）。

用 t 表示代数，则：

$$\begin{aligned}\boldsymbol{v}_i(t+1) &= K\left[\boldsymbol{v}_i(t) + \underline{(\phi_1+\phi_2)}\left(\underline{\frac{\phi_1\boldsymbol{b}_i(t)+\phi_2\boldsymbol{h}_i(t)}{\phi_1+\phi_2}} - \boldsymbol{x}_i(t)\right)\right] \\ &= K[\boldsymbol{v}_i(t) + \underline{\phi_T}(\underline{\boldsymbol{p}_i(t)} - \boldsymbol{x}_i(t))],\end{aligned}$$

惯性权重与压缩系数

- 压缩系数
  - 假设$\boldsymbol{p}_i(t)$不随时间改变，且定义：$\boldsymbol{y}_i(t)=\boldsymbol{p}_i(t)-\boldsymbol{x}_i(t)$.

可得：

$$\left.\begin{aligned}
\boldsymbol{v}_i(t+1) &= K\boldsymbol{v}_i(t)+K\phi_T\boldsymbol{y}_i(t),\\
\boldsymbol{y}_i(t+1) &= \boldsymbol{p}_i-\boldsymbol{x}_i(t+1)\\
&= \boldsymbol{p}_i-\boldsymbol{x}_i(t)-\boldsymbol{v}_i(t+1)\\
&= \boldsymbol{y}_i(t)-K\boldsymbol{v}_i(t)-K\phi_T\boldsymbol{y}_i(t)\\
&= -K\boldsymbol{v}_i(t)+(1-K\phi_T)\boldsymbol{y}_i(t).
\end{aligned}\right\}$$

即：

$$\begin{bmatrix}\boldsymbol{v}_i(t+1)\\ \boldsymbol{y}_i(t+1)\end{bmatrix}=\begin{bmatrix}K & K\phi_T\\ -K & 1-K\phi_T\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_i(t)\\ \boldsymbol{y}_i(t)\end{bmatrix}$$

惯性权重与压缩系数

- 压缩系数
  - 上式右边的矩阵支配着系统的稳定性，它的特征值为：

$$\lambda = \frac{1}{2}\left[1 - K(\phi_T - 1) \pm \sqrt{1 + K^2(\phi_T - 1)^2 - 2K(\phi_T + 1)}\right]$$
$$= \frac{1}{2}\left[1 - K(\phi_T - 1) \pm \sqrt{\Delta}\right].$$

其中判别式△由上式定义。记特征值为λ_1和λ_2，则：

$$\lambda_1 = \frac{1}{2}\left[1 - K(\phi_T - 1) + \sqrt{\Delta}\right], \quad \lambda_2 = \frac{1}{2}\left[1 - K(\phi_T - 1) - \sqrt{\Delta}\right].$$

如果$|\lambda_1|<1$且$|\lambda_2|<1$，则系统是稳定的。

可以进一步分析决定粒子群优化算法稳定性的λ_1和λ_2将随着压缩系数K如何变化。

惯性权重与压缩系数

- 压缩系数
  - 定理：如果在下式的速度更新中$\boldsymbol{b}_i$和$\boldsymbol{h}_i$为常数，并且$\Phi_T = \Phi_1 + \Phi_2 > 4$

$$\boldsymbol{v}_i \leftarrow K[\boldsymbol{v}_i + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i) + \phi_2(\boldsymbol{h}_i - \boldsymbol{x}_i)],$$

  则对于$K < 2/(\Phi_T - 2)$，粒子群优化是稳定的。

  - 对于稳定的粒子群优化算法，K可以写成：

  $K = 2\alpha/(\Phi_T - 2)$，这里$\Phi_T = \Phi_{1,max} + \Phi_{2,max}$，取$\alpha \in (0,1)$，α指示在粒子群优化算法变得不稳定之前，压缩系数K与其理论上的最大值的接近程度（较大的α允许更多的探索，较小的α更强调开发）

  - 在粒子群优化实际应用时，通常建议将Φ_T设置为比4略大，并将Φ_T近似等分给$\Phi_{1,max}$和$\Phi_{2,max}$，例如$\Phi_{1,max} = \Phi_{2,max} = 2.05$.

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. **全局速度更新**
6. 完全知情的粒子群
7. 从错误中学习
8. 总结

全局速度更新

- 速度更新公式的一般化
  - 将速度更新公式进行一般化，得到：

$$\boldsymbol{v}_i \leftarrow K[\boldsymbol{v}_i + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i) + \phi_2(\boldsymbol{h}_i - \boldsymbol{x}_i) + \phi_3(\boldsymbol{g} - \boldsymbol{x}_i)],$$

其中***g***是从第一代开始到目前为止找到的最好个体。

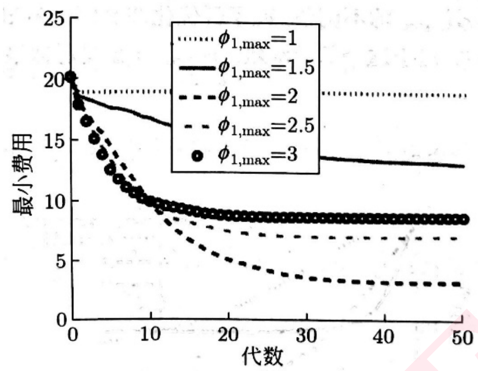
  - 例：使用粒子群优化算法优化20维Ackley函数，以上式进行速度更新，种群规模为50，精英参数为2，邻域规模为σ=4，采用以下标称值：

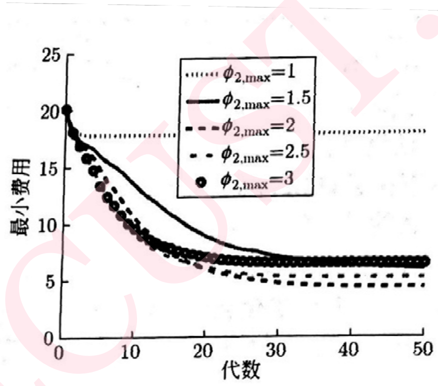
$\Phi_{1,max} = \Phi_{2,max} = \Phi_{3,max} = 2.1$， $\Phi_T = \Phi_{1,max} + \Phi_{2,max} + \Phi_{3,max}$

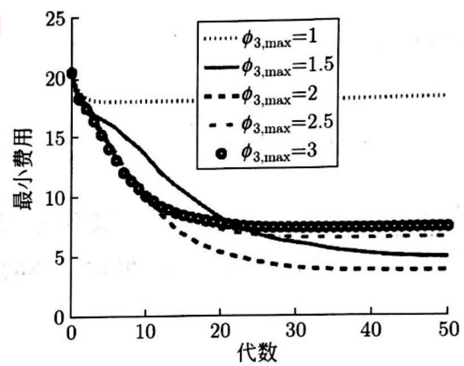
α=0.9，$K = 2\alpha/(\Phi_T - 2)$

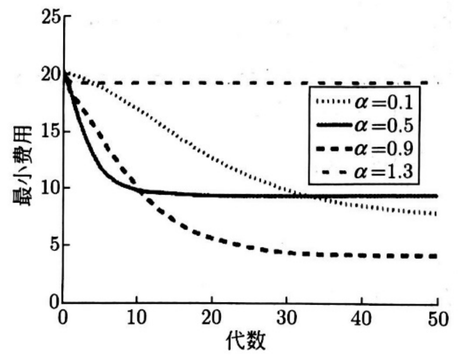
全局速度更新

- 速度更新公式的一般化
  - 例：使用粒子群优化算法优化20维Ackley函数，以上式进行速度更新，种群规模为50，精英参数为2，邻域规模为σ=4。考察$\Phi_{1,max}$，$\Phi_{2,max}$，$\Phi_{3,max}$和α取不同的值时，粒子群优化的性能。









这三幅图分别表示：$\Phi_{1,max}$=2，$\Phi_{2,max}$=2，$\Phi_{3,max}$=2均为近似最优。

K=0.9K_{max}近似最优

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. **完全知情的粒子群**
7. 从错误中学习
8. 总结

完全知情的粒子群

- 速度更新公式的一般化
  - 上一章得到了最一般的速度更新形式：

$$\boldsymbol{v}_i(t+1) = K[\boldsymbol{v}_i(t) + \phi_T(\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t))],$$

$$\phi_T = \phi_{1,\max} + \phi_{2,\max} + \phi_{3,\max},$$

$$\boldsymbol{p}_i(t) = \frac{\phi_1 \boldsymbol{b}_i(t) + \phi_2 \boldsymbol{h}_i(t) + \phi_3 \boldsymbol{g}(t)}{\phi_1 + \phi_2 + \phi_3}.$$

用3个粒子的位置构造速度更新：1）当前个体到目前为止最好的位置**b**$_i$(t)，2）当前的邻域到目前为止的最好位置**h**$_i$(t)，3）到目前为止种群的最好位置**g**(t)。

问：为什么不让种群中的每一个个体都为速度更新做出贡献？

完全知情的粒子群

- 完全知情的粒子群
  - 因此，上式可写成：

$$\boldsymbol{v}_i(t+1) = K[\boldsymbol{v}_i(t) + \phi_T(\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t))],$$

$$\phi_T = \frac{1}{N}\sum_{j=1}^{N}\phi_{j,\max},$$

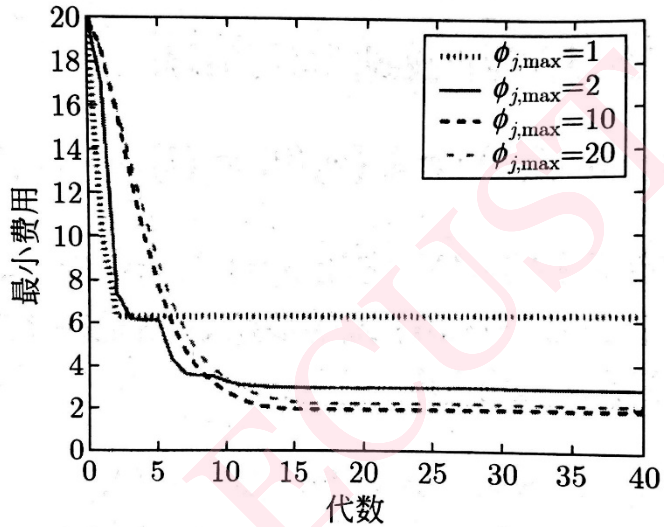
$$\boldsymbol{p}_i(t) = \frac{\sum_{j=1}^{N} w_{ij}\phi_j \boldsymbol{b}_j(t)}{\sum_{j=1}^{N} w_{ij}\phi_j},$$

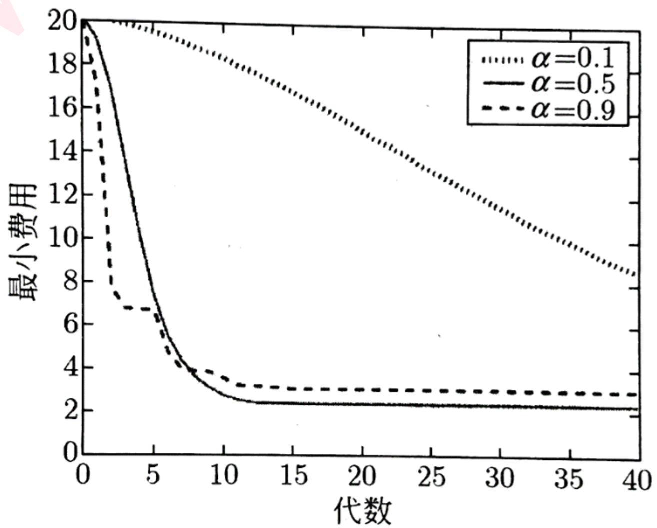
其中，**b**$_j$(t)是由第j个粒子到目前为止找到的最好的解——完全知情的粒子群

上式中的权重ω_{ij}是确定性因子，描述第j个粒子对第i个粒子的速度的影响；有时候，对于所有的j，取ω_{ij} = 常数；另一些时候，想让ω_{ij}对较好的粒子**x**$_j$的j取更大的值，对离**x**$_i$较近的粒子**x**$_j$的j也取更大的值。

完全知情的粒子群

- 完全知情的粒子群
  - 例：用完全知情的粒子群，对20维Ackley函数进行优化，种群规模为40，精英参数为2， $\Phi_{j,max}=\Phi_{max}=2$, $j\in[1,20]$, $\alpha=0.9$, $K=2\alpha/(3\Phi_{max}-2)$





当Φ_{max}较小时，群体会很快收敛，但是收敛到较差的解；当Φ_{max}增大时，最初的收敛会变慢，但最后收敛到更好的解

对于较小的α值，收敛非常慢，当α=0.9时，收敛最快，但是当α=0.5时，最后的解更好

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. **从错误中学习**
8. 总结

从错误中学习

- 基本思想
  - 粒子群优化的基本思路在于生物体倾向于重复过去成功的策略，包括生物体本身使用过的有利的策略，也包括在别的地方观察到的有利的策略，如前面章节得到的速度更新基本公式：

$$\boldsymbol{v}_i \leftarrow K[\boldsymbol{v}_i + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i) + \phi_2(\boldsymbol{h}_i - \boldsymbol{x}_i) + \phi_3(\boldsymbol{g} - \boldsymbol{x}_i)],$$

其中$\mathbf{x}_i$和$\mathbf{v}_i$是第i个粒子的位置和速度，$\mathbf{b}_i$是第i个粒子以前最好的位置，$\mathbf{h}_i$是第i个邻域当前最好的位置，$\mathbf{g}$是整个种群以前最好的位置；Φ_1,Φ_2和Φ_3分别为在$(0, \Phi_{1,max})$, $(0, \Phi_{2,max})$和$(0, \Phi_{3,max})$上均匀分布的随机数。

  - 生物体不仅从成功中学习，也会从错误中学习

从错误中学习

- 负强化的粒子群优化
  - negative reinforcement particle swarm optimization（NPSO）
  - 每一个粒子不仅移向自己及其邻居的最好位置，而且会远离自己及其邻居的最差位置：

$$
\begin{aligned}
\boldsymbol{v}_i \leftarrow K[&\boldsymbol{v}_i + \phi_1(\boldsymbol{b}_i - \boldsymbol{x}_i) + \phi_2(\boldsymbol{h}_i - \boldsymbol{x}_i) + \phi_3(\boldsymbol{g} - \boldsymbol{x}_i) - \\
&\phi_4(\bar{\boldsymbol{b}}_i - \boldsymbol{x}_i) - \phi_5(\bar{\boldsymbol{h}}_i - \boldsymbol{x}_i) - \phi_6(\bar{\boldsymbol{g}} - \boldsymbol{x}_i)],
\end{aligned}
$$

其中，$\bar{\boldsymbol{b}}_i$ 是第i个粒子以前最差的位置；$\bar{\boldsymbol{h}}_i$是第i个邻域当前最差的位置；$\bar{\boldsymbol{g}}$是整个种群以前最差的位置；每一个是在$(0,\Phi_{j,max})$上均匀分布的随机数；并且每个$\Phi_{j,max}$都是可调参数并为正数。

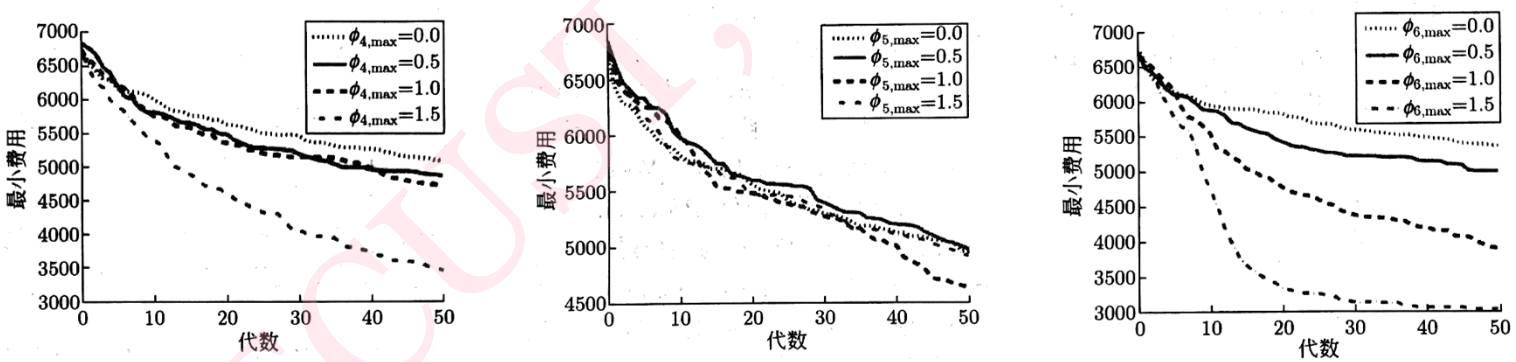
从错误中学习

- 负强化的粒子群优化
  - negative reinforcement particle swarm optimization（NPSO）
  - 标准粒子群优化的速度调整朝向有利的解，在NPSO的速度调整中增加了远离有害的解
  - 平衡：朝向有利的解、远离有害的解
  - 应该花多大力气聚焦在成功上并努力赶上？应该花多大力气聚焦失败并极力避免？
  - 大多数人认可正强化比负强化更有效，但大多数人也同意这两种强化对学习来说都很重要

从错误中学习

- 负强化的粒子群优化

例：用NPSO优化20维Schwefel正弦函数，取种群规模为20，精英参数为2，$\Phi_{1,max} = \Phi_{2,max} = \Phi_{3,max} = 2$， $\Phi_{4,max} = \Phi_{5,max} = \Phi_{6,max} = 0$，$\alpha$=0.9，$K = 2\alpha/(\Phi_{1,max} + \Phi_{2,max} + \Phi_{3,max} - 2)$



20次蒙特卡洛仿真上的平均：避开自己以前最差位置的粒子性能显著提升

本章内容

1. 背景知识
2. 基本粒子群优化算法
3. 速度限制
4. 惯性权重与压缩系数
5. 全局速度更新
6. 完全知情的粒子群
7. 从错误中学习
8. **总结**

总结

- 粒子群优化
  - 基本粒子群算法
  - 速度限制
  - 惯性权重与压缩系数
  - 全局速度更新
  - 完全知情的粒子群
  - 从错误中学习
  - 群体智能、进化算法

结束