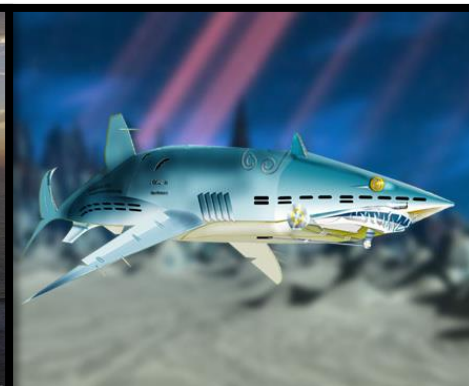


机器人智能控制

易建强 蒲志强 袁如意

中国科学院自动化研究所

2020年秋季



第三讲 智能优化方法之 粒子群优化

蒲志强 高级工程师

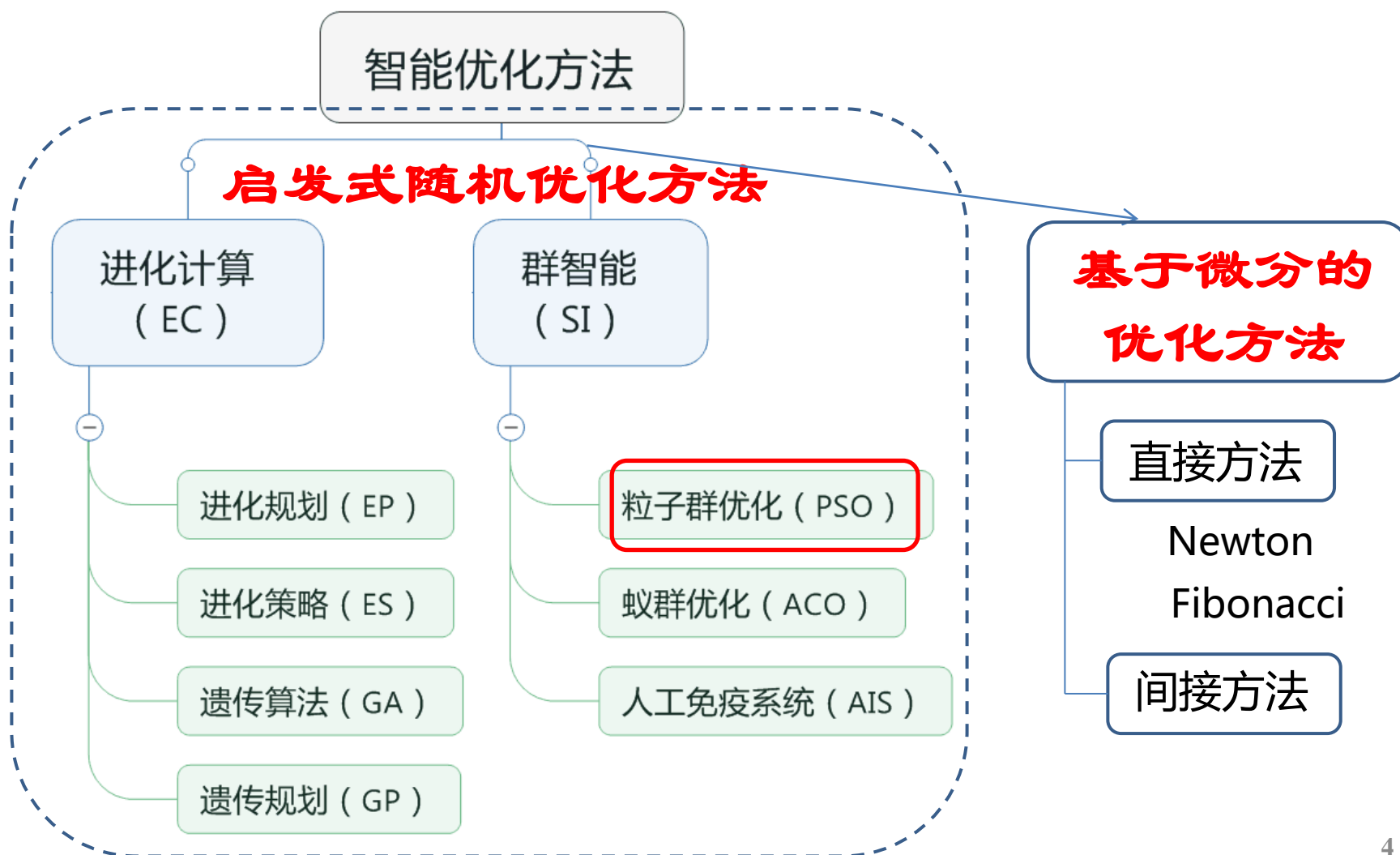
zhiqiang.pu@ia.ac.cn



本讲的主要内容

- 一、产生和发展
- 二、基本粒子群优化算法
- 三、改进和研究方向
- 四、应用实例

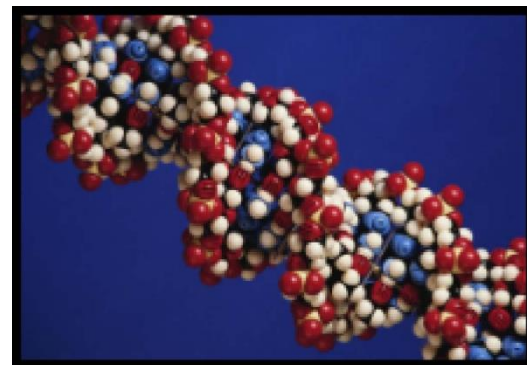
一、产生和发展



■ 向大自然学习

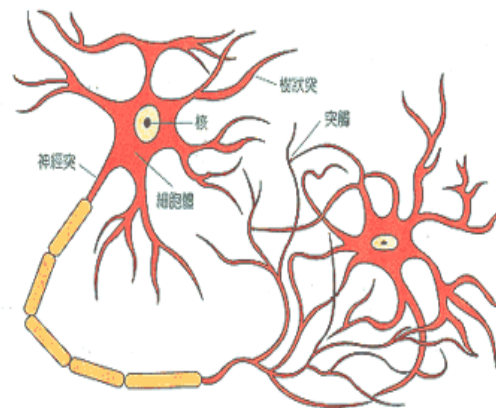
□ 遗传算法 (GA)

物竞天择，设计染色体编码，根据适应值函数进行染色体选择、交叉和变异操作，优化求解



□ 人工神经网络算法 (ANN)

模仿生物神经元，透过神经元的信息传递、训练学习、联想，优化求解



□ 模拟退火算法 (SA)

模模仿金属物质退火过程

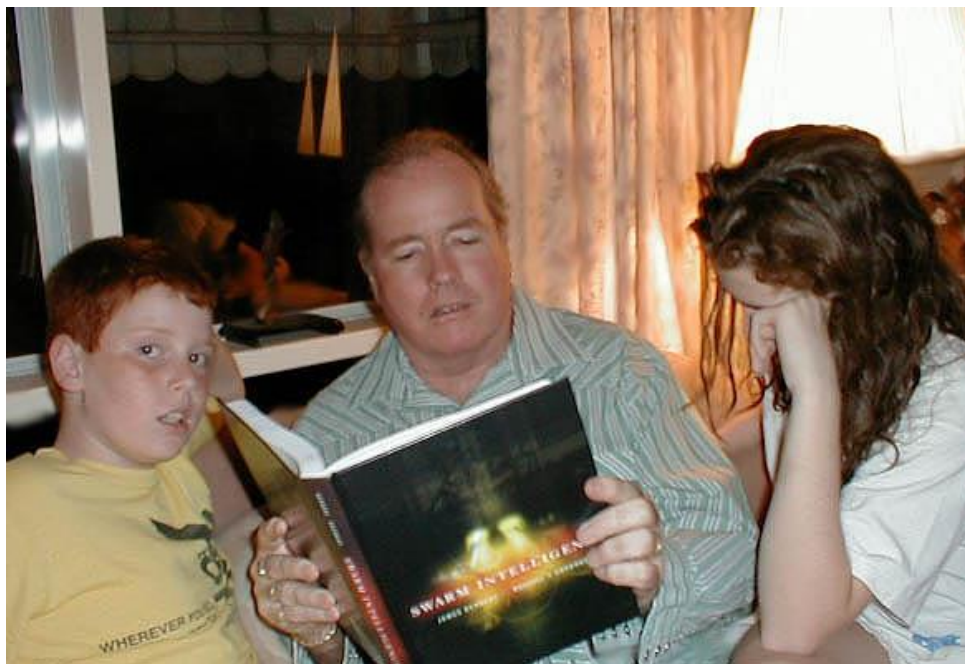


■ 产生背景

- 20世纪80年代，社会性动物(如蚁群、蜂群、鸟群)的自组织行为引起了人们的广泛关注。许多学者对其进行数学建模和计算机仿真。产生了所谓的群集智能Swarm Intelligence (SI)。
- 1995年，由美国学者Kennedy和Eberhart提出，基本概念源于对鸟群、鱼群的捕食行为的研究。
- 粒子群优化 Particle Swarm Optimization (PSO)。

■ 发明人

James Kennedy



Russell Eberhart



■ 基本思想

设想这样一个场景：一群鸟在随机搜索食物

已知 { 在这块区域里只有一块食物
所有的鸟都不知道食物在哪里
但它们能感受到当前的位置离食物还有多远

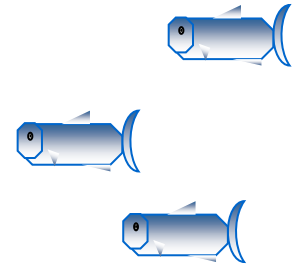
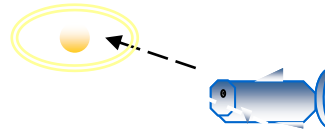
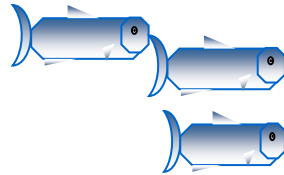
那么：找到食物的最优策略是什么呢？

搜寻目前离食物最近的鸟的周围区域
根据自己飞行的经验判断食物的所在

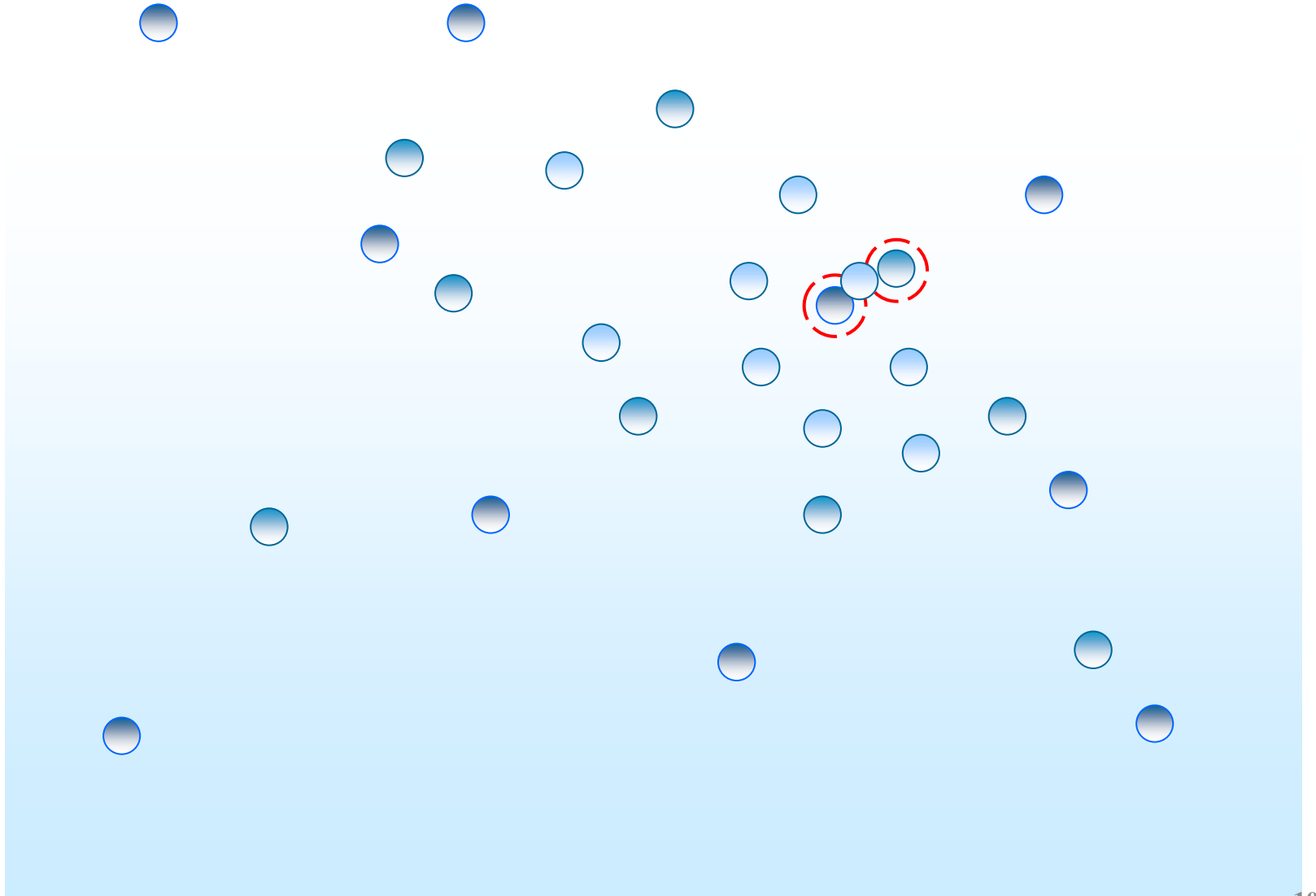
PSO正是从这种模型中得到了启发

PSO的基础：信息的社会共享

■ 基本思想



■ 粒子群特性



二、基本粒子群优化算法

1. 算法流程
2. 工具箱
3. 参数分析
4. 优缺点分析

■ 基本思想

- 每个寻优的问题解都被想像成一只鸟，称为“粒子”。所有粒子都在一个D维空间进行搜索。
- 所有的粒子都由一个fitness function 确定适应值以判断目前的位置好坏。
- 每一个粒子必须赋予记忆功能，能记住所搜寻到的最佳位置。
- 每一个粒子还有一个速度以决定飞行的距离和方向。这个速度根据它本身的飞行经验以及同伴的飞行经验进行动态调整。

■ 算法流程

- 1) 初始化
- 2) 计算适应度
- 3) 搜索位置的最优解
- 4) 位置和速度的更新
- 5) 满足终止条件，结束，否则，返回2

■ 1-初始化

- D 维的目标搜索空间
- m 个粒子组成的群落
- 第 i 个粒子在 D 维的搜索空间中的位置

$$\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD}), i = 1, 2, \dots, m$$

- 第 i 个粒子在 D 维的搜索空间中的飞翔速度

$$\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD}), i = 1, 2, \dots, m$$

■ 2-计算适应度

- 每个粒子的位置就是一个潜在的解，将其代入一个**目标函数**就可以计算出其适应值，根据适应值的大小衡量解的优劣。
- p_{best} : 单个粒子的最优适应度
- g_{best} : 整个群落中粒子的最优适应度
- p_{id} : p_{best} 所对应的粒子的位置
- p_{gd} : g_{best} 所对应的粒子的位置

■ 3-最优解

- 第 i 个粒子的适应度与前一时刻的适应度比较，得出最优适应度 p_{best} 和最优位置 p_{id}
- 整个粒子群的适应度比较，得出最优适应度 g_{best} 和最优位置 p_{gd}

■ 4-更新

- 最初方法
 - 粒子*i*的速度更新

粒子*i*当前位置与自己
最优位置之间的距离

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})$$

粒子*i*的
速度项

粒子*i*的位置更新

粒子*i*当前位置与群体
最优位置之间的距离

$$x_{id} = x_{id} + v_{id}$$

学习因子 c_1, c_2 : 非负

r_1, r_2 为随机数, $\in [0, 1]$

$v_{id} \in [-v_{\max}, v_{\max}]$, 最大飞行速度 v_{\max} : 常数

■ 4-更新

- 目前常用方法

- 粒子 i 的速度更新

惯性权
重因子

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id})$$

- 粒子 i 的位置更新

$$x_{id} = x_{id} + v_{id}$$

- 发明人：Yunhui Shi 和
Russell Eberhart, 1998



■ 4-更新

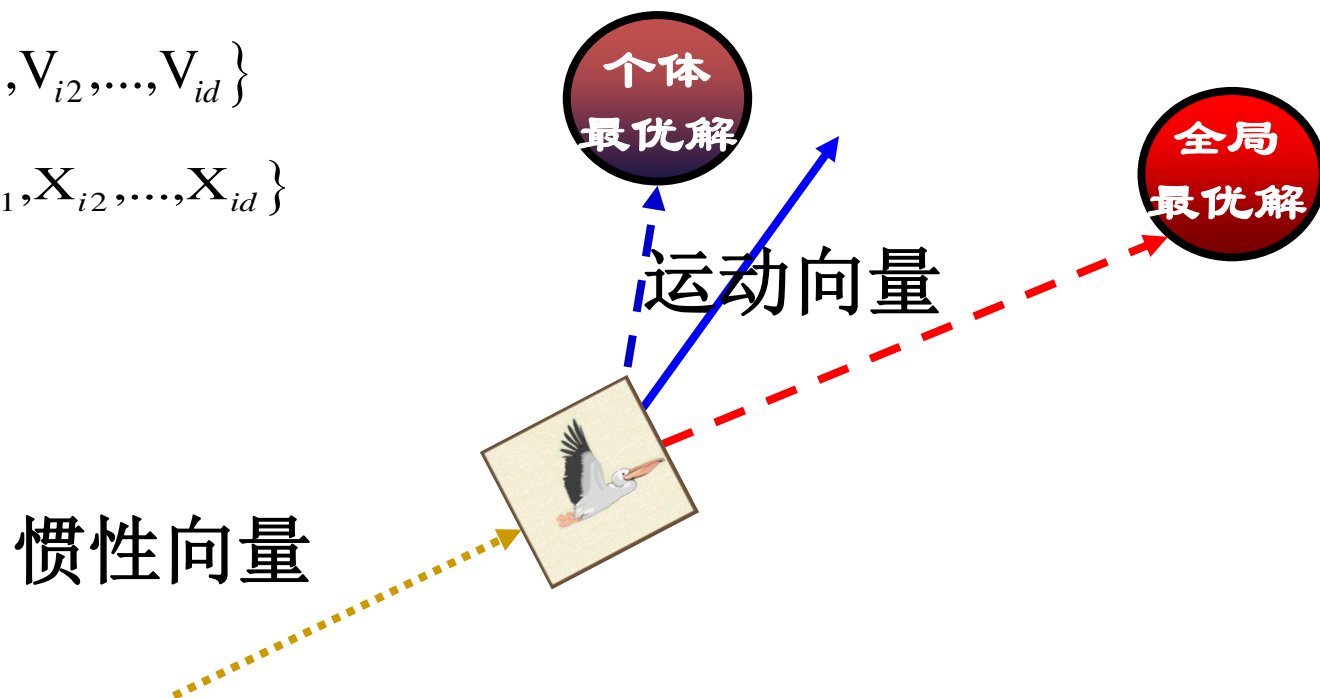
学习因子

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

$$x_i(t+1) = x_i(t) + v_{id}(t+1)$$

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

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



■ 5-终止条件

- 如果满足：
 - 最大迭代次数
 - 最优位置满足预定适应度最小阈值
- 则：
 - 结束，显示结果
- 否则：
 - 返回计算适应度部分

■ 函数寻优

- Shaffer's F6 函数

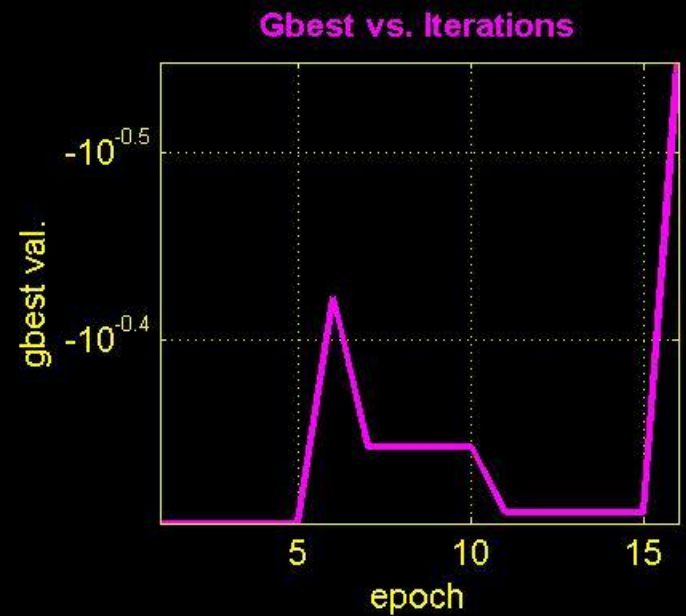
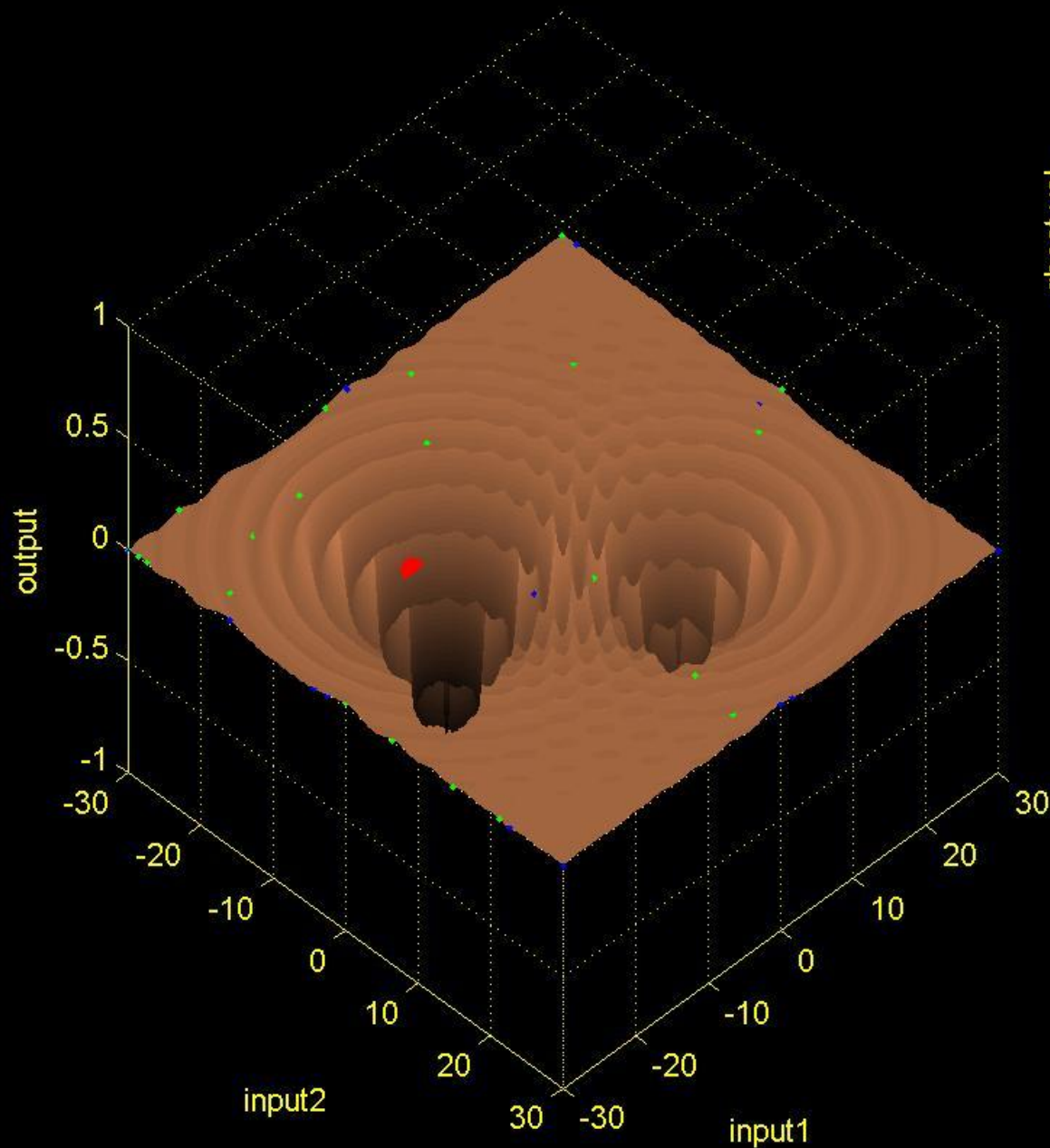
$$f = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{\left(1 + 0.001(x^2 + y^2)\right)^2}$$

- 在(0,0)有全局最大值1
- 最大值周围有一个圈脊，取值为0.990283，易在此停滞

- Particle Swarm Optimization toolbox

- <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7506>
- 作者：Brian Birge
- 静态环境、动态环境

-0.282598 = f6_bubbles_dyn([-6.24068, -14.4153])



Common PSO

Dimensions : 2

of particles : 30

Minimize to : Unconstrained

Function : $f6_bubbles_dyn$

Inertia Weight : 0.85758

Environment Change

Red = Global Best

Green = Personal Bests

Blue = Current Positions

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id})$$

- 惯性权重因子: w

- 控制速度对当前速度的影响



迭代次数

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

- 开始时, $w = w_{\max}$, 速度大, 有利于搜索更大的空间, 发现新的解。

- 后期, w 减小, 逐渐接近 w_{\min} , 速度变小, 有利于在当前的解空间里挖掘更好的解。

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id})$$

- 学习因子: c_1, c_2

- 调整粒子的自身经验与社会(群体)经验在其运动中所起作用的权重。
- $c_1=0$, 粒子没有自身经验, 只有社会经验, 收敛速度可能很快, 但在问题程度复杂时, 容易陷入局部最优解。
- $c_2=0$, 粒子没有群体共享信息, 只有自身经验, 因为个体间没有交互, 得到最优解的几率小。

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id})$$

- 最大飞行速度： v_{max}
 - 决定搜索空间的分辨率和精细程度
 - 越高，容易错过最优解
 - 越低，解空间的搜索能力不强，易陷入局部极值点

■ 常用参数

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id})$$

- 惯性权重因子 w : 0.9->0.4
- 学习因子 (c_1, c_2) : (2.0, 2.0)
- 最大飞行速度 v_{max} : 解空间范围的10~20%
- 种群粒子数 : 20~50

- 优点：
 - 参数不多，操作简单，使用方便
 - 收敛速度快
- 缺点：
 - 易陷入局部极小点
 - 不能保证收敛到最优解

■ 与GA比较

- 相似点：
 - 随机初始化种群
 - 适应度来评价个体或粒子，并指导搜索
 - 不能保证找到全局最优解

■ 与GA比较

- 区别：
 - PSO中没有交叉和变异，而根据速度搜索。
 - PSO具有记忆。
 - 信息共享：GA的染色体之间共享信息，整个种群比较均匀地向最优区域移动；PSO中，只有 g_{best} 的信息传递给其他粒子，是单向的信息流动，整个搜索过程是跟踪当前最优解的过程，可能更快收敛到最优解。

三、改进和研究方向

1. 收缩因子算法
2. 粒子拓扑结构
3. PSO的研究方向

1999年, Clerc引入收缩因子以保证算法的收敛性。

速度更新公式为

$$v_{id} = K[v_{id} + \varphi_1 r_1 (pbest_{id} - x_{id}) + \varphi_2 r_2 (gbest_d - x_{id})]$$

其中, 收缩因子 K 为受 φ_1 φ_2 限制的 w 。 φ_1 φ_2 是需要预先设定的模型参数

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \varphi = \varphi_1 + \varphi_2, \varphi > 4$$

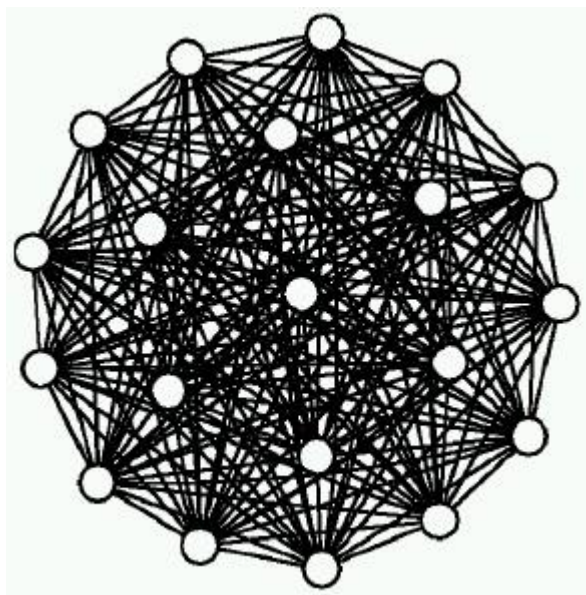
收缩因子法控制系统行为最终收敛, 且可以有效搜索不同区域, 该法能得到较高质量的解。

- 全局算法
 - 与全部粒子比较，得出适应度最优值
 - g_{best} : 全部粒子的最优适应度
 - p_{gd} : g_{best} 所对应的粒子的位置
- 局部算法
 - 与邻域粒子比较，得出局部适应度最优值
 - l_{best} : 临近粒子的最优适应度
 - p_{ld} : l_{best} 所对应的粒子的位置

邻域拓扑结构 决定 群体历史最优位置

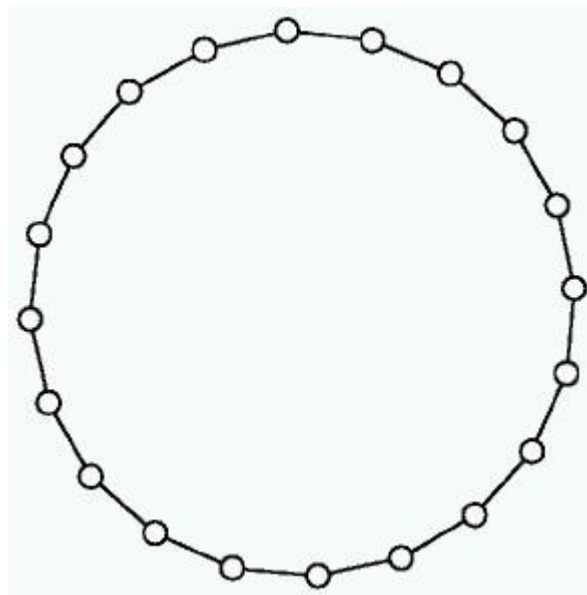
■ 典型拓扑结构

全连接



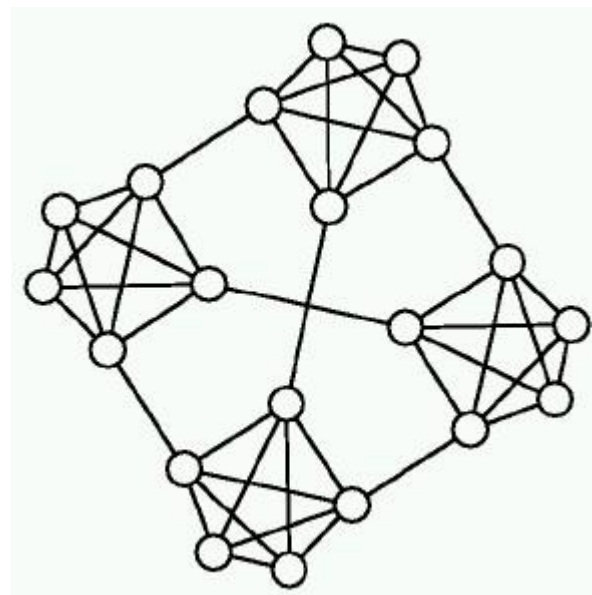
All

环形



Ring

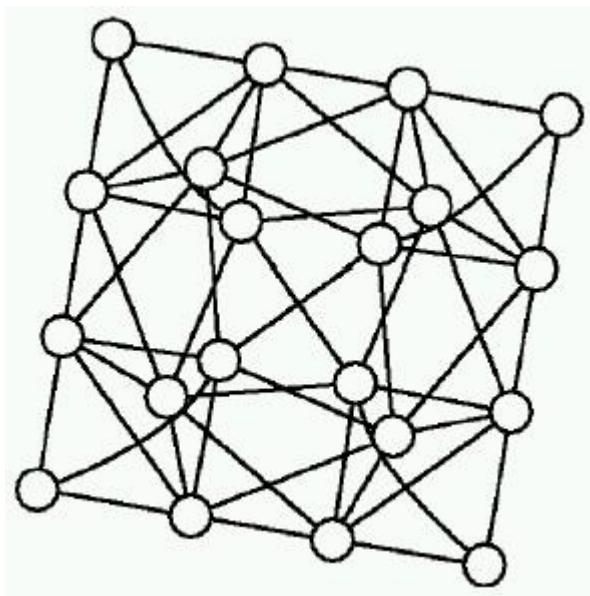
四组合



FourClusters

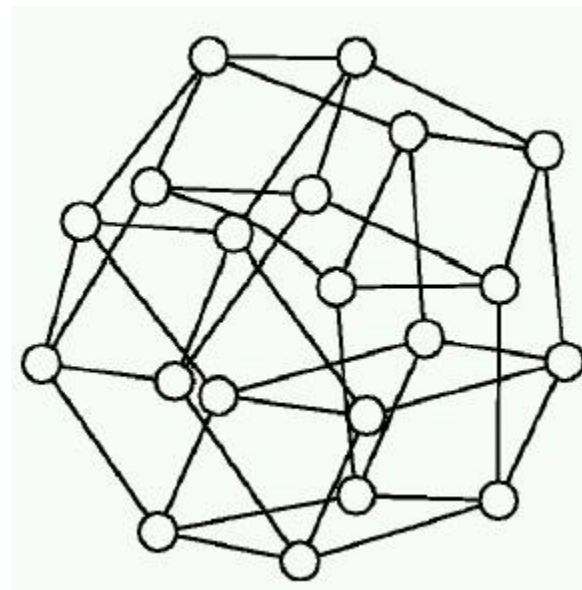
■ 典型拓扑结构

金字塔



Pyramid

四面体



Square

- 算法收敛问题

- 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。

- 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。

- 参数选择与优化

- 参数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，对算法性能改进有着重要的意义。

- 与其他计算智能算法的融合

- 如何将其他计算智能方法的优点和PSO的优点相结合，构造出性能好、有特色的混合算法，是当前算法改进的一个重要方向。

- 算法应用

- 算法的有效性必须在应用中才能体现，广泛开拓PSO的应用领域，对深化研究PSO算法非常有意义（为什么用？必要性。怎么用？问题表达和建模）。

- 继续向大自然学习

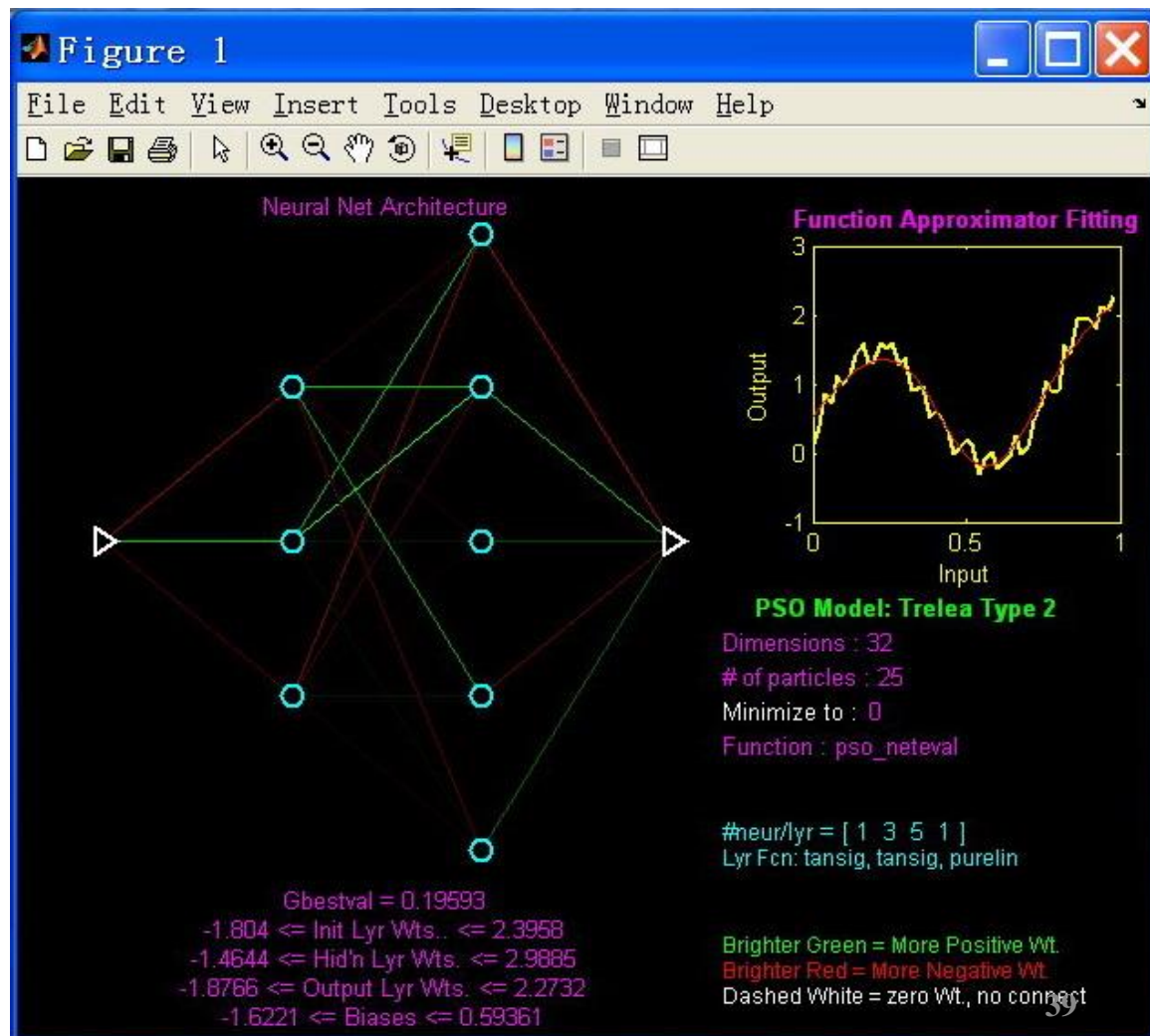
- PSO算法来源于对自然现象的模拟，继续深化对自然中群体行为的认知，挖掘自然的宝藏，或将获取意想不到的巨大突破。

四、应用实例

1. 神经网络训练
2. PID参数优化

神经网络训练

- 梯度下降误差反传方法
 - Traingd
 - Trainlm
- 粒子群算法
 - TrainPSO



■ 神经网络训练 —— 帕金森症状的诊断



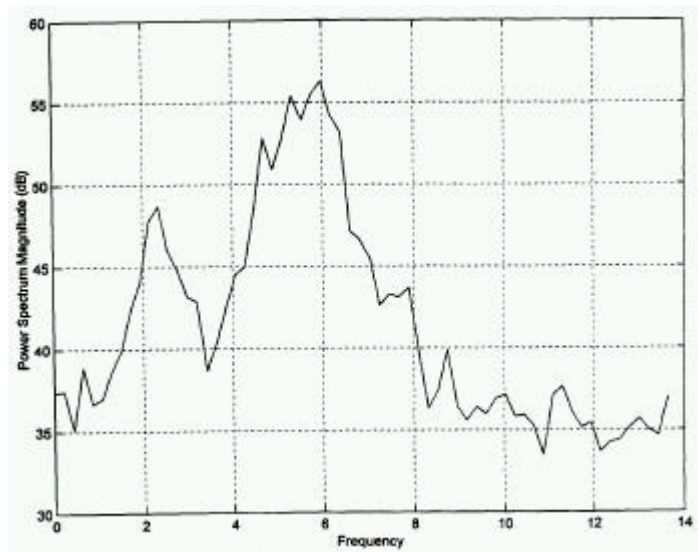
神经网络

帕金森
症状？

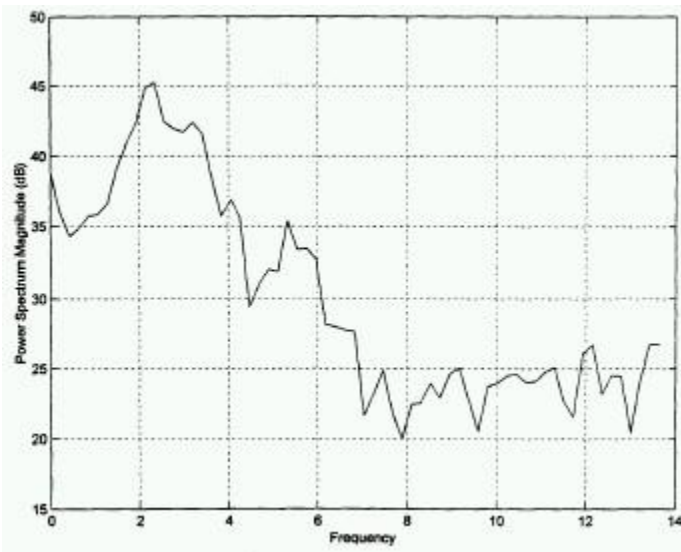
■ 神经网络训练 —— 帕金森症状的诊断

• 数据处理

- 功率谱：64个点的频率值，去掉最大2个点，最小2个点，共60个点作为输入。



帕金森症状



正常

■ 神经网络训练 —— 帕金森症状的诊断

- 神经网络结构：
 - 三层前向网络：输入层60个节点，隐含层12个节点，输出层2个节点
- 训练数据：
 - 12个帕金森症状病人，10个正常人
- 训练：
 - 隐含层节点的权重

■ 神经网络训练 —— 帕金森症状的诊断

- PSO参数

- 粒子个数：30个
- 学习因子 c_1, c_2 : 2.0, 2.0
- 最大飞行速度 v_{max} : 2.0
- 惯性权重因子 w : 0.9~0.4
- 迭代次数：2000

■ 神经网络训练 —— 帕金森症状的诊断

• 结果

- 迭代38次收敛
- $\text{Output1} > 0.8$, 患病
- $\text{Output1} < 0.2$, 正常
- 准确率100%

Classification	Output 1	Output 2
Normal	0.053	0.948
Normal	0.027	0.973
Tremor	0.917	0.088
Tremor	0.981	0.019
Normal	0.181	0.813
Normal	0.025	0.975
Normal	0.038	0.962
Tremor	0.982	0.020
Tremor	0.932	0.067
Tremor	0.948	0.051
Tremor	0.968	0.036
Tremor	0.982	0.019
Normal	0.048	0.953
Tremor	0.986	0.015
Normal	0.066	0.935
Normal	0.070	0.930
Tremor	0.842	0.157
Tremor	0.944	0.058
Normal	0.028	0.972
Tremor	0.955	0.049
Tremor	0.990	0.011
Normal	0.038	0.961

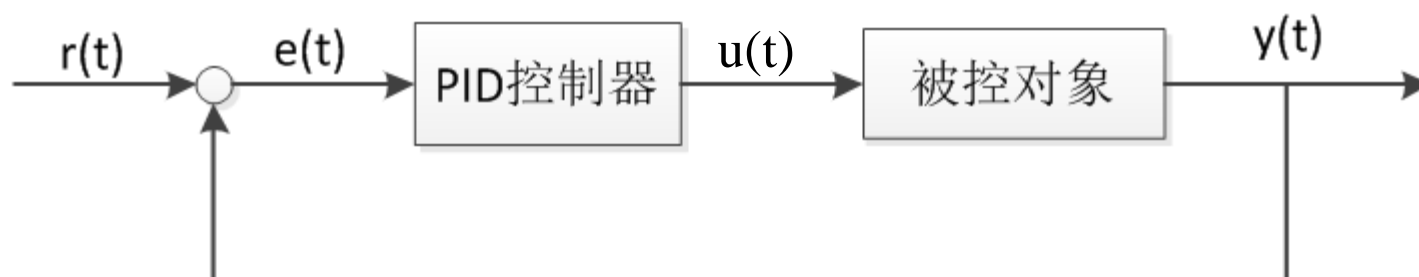
■ 神经网络训练 —— 帕金森症状的诊断

- 与误差反传 (BP) 训练算法比较
 - 收敛速度要快，训练数据较少的情况下，消耗时间大约是误差反传算法的六分之一
 - 全局收敛性要好，误差更小

- PID形式

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

- 控制器结构



- 被控对象

$$G(s) = \frac{s+2}{s^4 + 8s^3 + 4s^2 - s + 0.4}$$

- 优化目标——PID控制参数

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

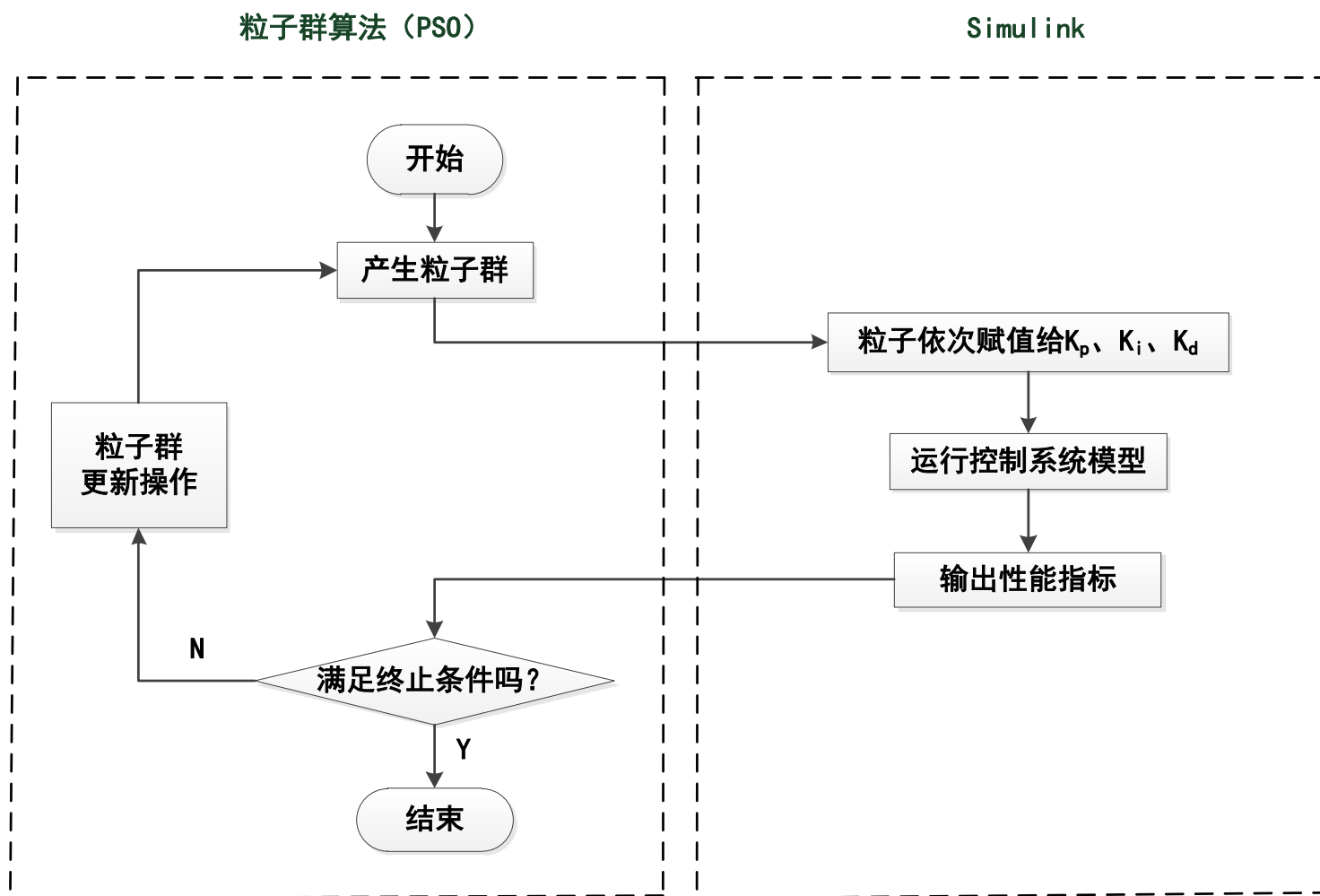
- 问题：如何评价参数好还是不好？

- 评价指标

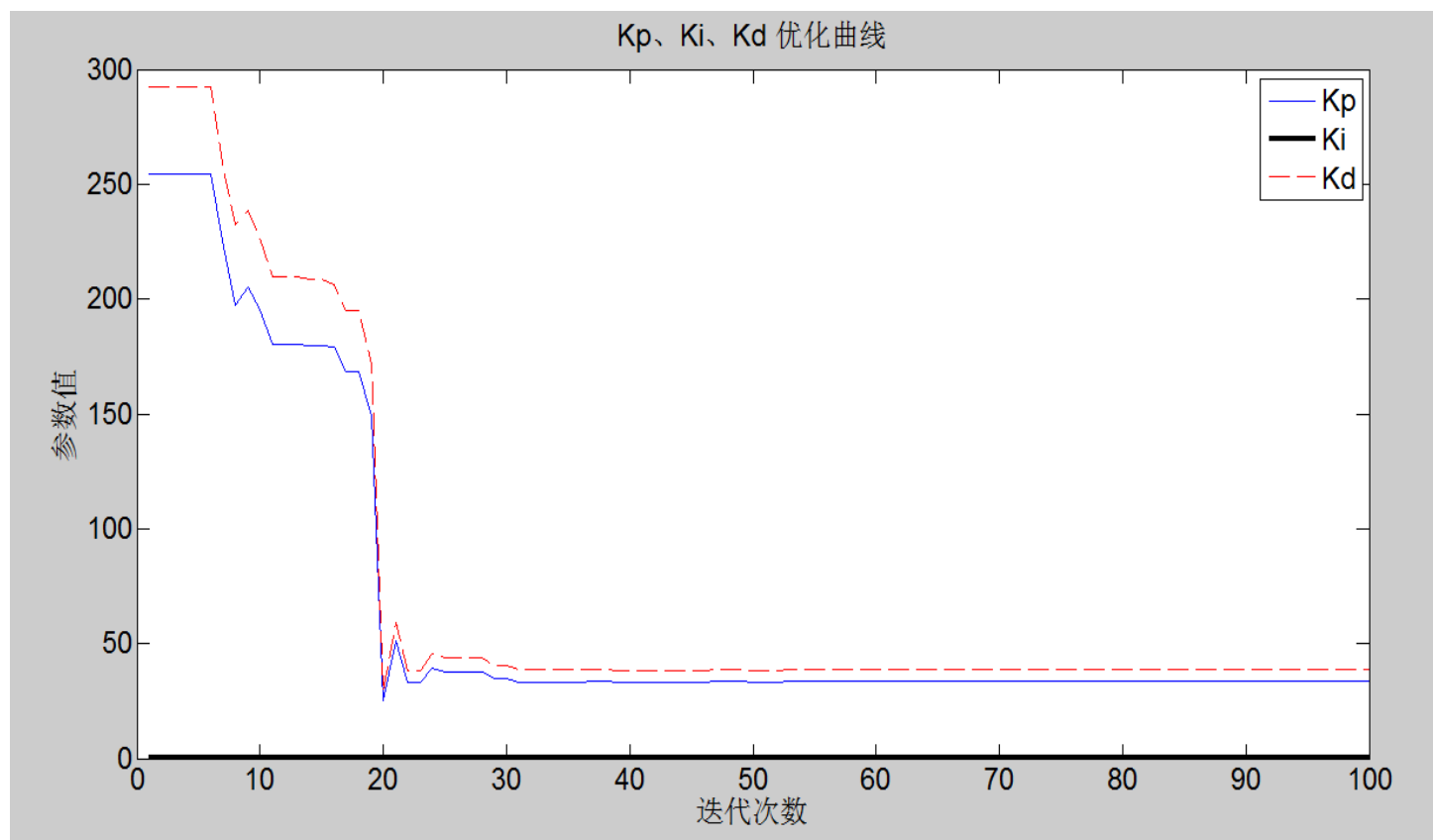
- PID控制器的优化问题就是确定一组合适的参数 K_p 、 K_i 、 K_d ,使得指标达到最优。常用的误差性能指标包括ISE、IAE、ITAE等，这里选用ITAE指标，其定义为

$$J = \int_0^{\infty} t |e(t)| dt$$

- 算法流程

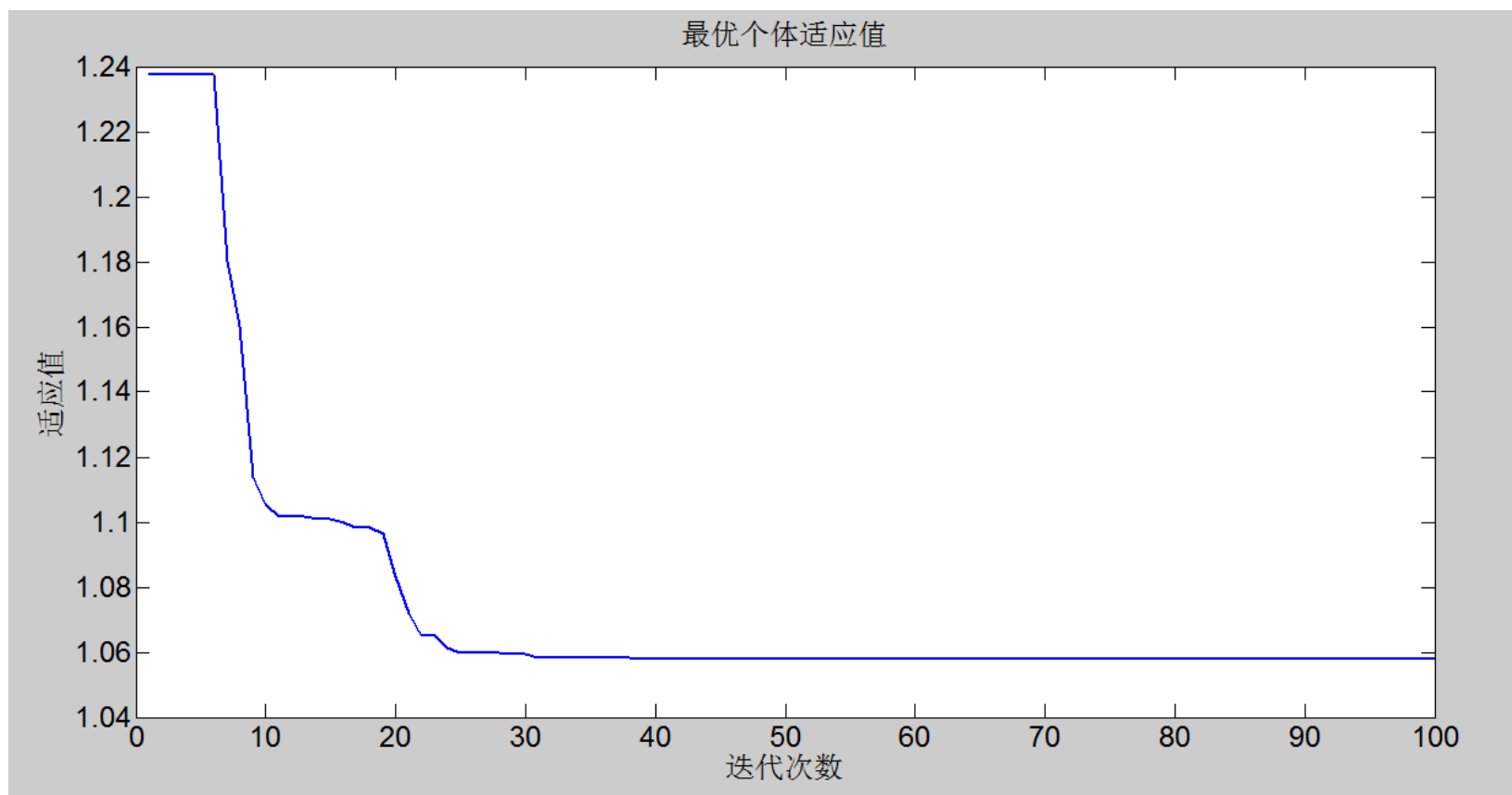


- 优化结果



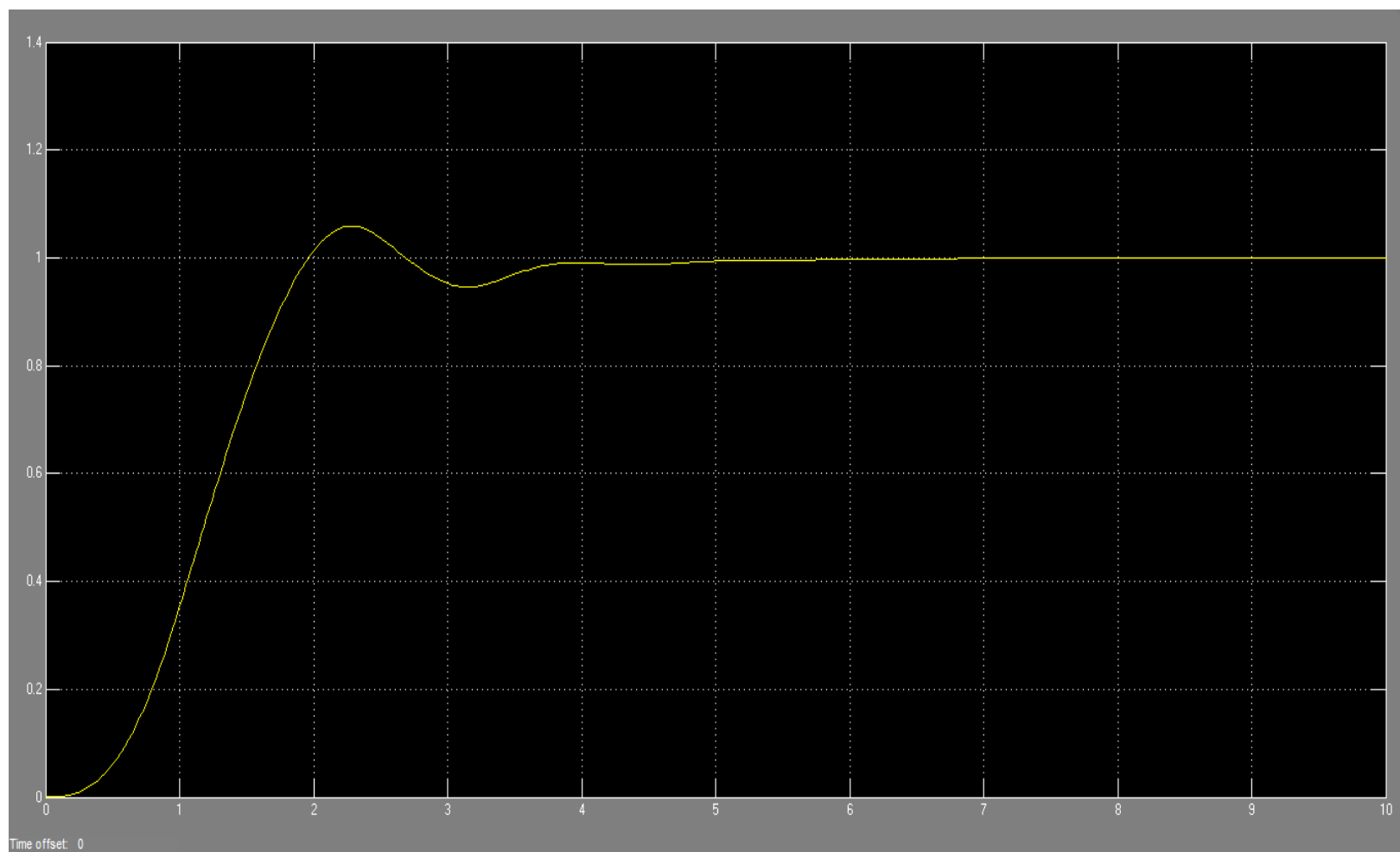
$$\begin{aligned} K_p &= 33.6469, & K_i &= 0.1662, \\ K_d &= 38.7990, & ITAE &= 1.0850 \end{aligned}$$

- 优化结果



$$\begin{aligned} K_p &= 33.6469, & K_i &= 0.1662, \\ K_d &= 38.7990, & ITAE &= 1.0850 \end{aligned}$$

- 最优参数控制曲线



■ 复习 —— PSO算法流程

- 1) 初始化
- 2) 计算适应度
- 3) 搜索位置的最优解
- 4) 位置和速度的更新
- 5) 满足终止条件，结束，否则，返回2