

文章编号: 1001-0920(2005)04-0377-05

增强型微粒群优化算法及其在软测量中的应用

陈国初, 俞金寿

(华东理工大学 自动化研究所, 上海 200237)

摘要: 对微粒群优化算法(PSO)进行分析, 提出一种增强型微粒群优化算法(EP SO). 用 EP SO 和 PSO 对几种常用函数的优化问题进行测试比较, 结果表明 EP SO 比 PSO 更容易找到全局最优解, 优化效率和优化性能明显提高. 将 EP SO 用于催化裂化装置主分馏塔粗汽油干点软测量, 建立了基于 EP SO 算法的粗汽油干点神经网络软测量模型. 研究表明, 基于 EPSONN 的软测量模型比基于 BPNN 的软测量模型具有更高的精度和更好的性能.

关键词: 微粒群优化; 增强型微粒群优化; 神经网络; 软测量

中图分类号: TP273

文献标识码: A

Enhanced particle swarm optimization and its application in soft-sensor

CHEN Guo-chu, YU Jin-shou

(Research Institute of Automation, East China University of Science and Technology, Shanghai 200237, China
Correspondent: CHEN Guo-chu, E-mail: chgcsh@sohu.com)

Abstract: An enhanced particle swarm optimization algorithm (EP SO) is proposed based on the analysis of PSO. Both EP SO and PSO are used to resolve several well-known and widely used test function optimization problems. Results show that EP SO has greater efficiency, better performance and more advantages in many aspects than PSO. Then, EP SO is applied to train artificial neural network (NN) to construct a practical soft-sensor of gasoline endpoint of main fractionator of fluid catalytic cracking unit. The obtained results show that the proposed method is feasible and effective in soft-sensor of gasoline endpoint.

Key words: PSO; EP SO; neural network; soft-sensor

1 引言

微粒群优化算法(PSO)^[1,2]是一种新的进化计算算法. PSO 简单且具有许多良好的优化性能, 因而得到广泛的应用^[3-6], 但存在容易陷入局部极值的缺陷. 本文通过分析 PSO 算法参数对其优化性能的影响, 提出一种增强型微粒群优化算法(EP SO), 并用 EP SO 和 PSO 对几种函数的优化问题进行测试和比较.

产品质量指标软测量的研究一直是一个热点问题^[7,8]. 催化裂化装置(FCCU)在石油二次加工中具有重要的经济意义, 其操作状况直接关系到炼油装

置轻质油品的收率, 从而影响整个炼油厂的经济效益^[9]. 本文将 EP SO 用于 FCCU 主分馏塔粗汽油干点软测量, 建立了基于 EP SO 的粗汽油干点神经网络软测量模型, 并与基于 BP 算法的粗汽油干点神经网络软测量模型进行比较. 结果表明, 基于 EPSONN 的软测量模型比基于 BPNN 的软测量模型具有更好的性能.

2 增强型微粒群优化算法

2.1 标准微粒群优化算法

假设在一个 D 维搜索空间中, 有 m 个微粒组成一微粒群, 其中第 i 个微粒的空间位置为 $x_i = (x_{i1},$

收稿日期: 2004-06-01; 修回日期: 2004-09-14

基金项目: 教育部博士点专项基金项目(20030251003).

作者简介: 陈国初(1971—), 男, 江西都昌人, 博士生, 从事化工过程建模、仿真与智能算法的研究; 俞金寿(1939—), 男, 浙江海宁人, 教授, 博士生导师, 从事工业过程建模、仿真、优化与控制等研究.

x_{i2}, \dots, x_{iD}), 第 i 个微粒所经历的最好位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 每个微粒的飞行速度为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, $i = 1, 2, \dots, m$. 在整个微粒群中, 所有微粒经历过的最好位置为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$. 每一代微粒的第 d 维 ($1 \leq d \leq D$) 根据如下方程变化:

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}), \quad (1)$$

$$x_{id} = x_{id} + v_{id}. \quad (2)$$

其中: ω 为惯性权值; c_1 和 c_2 为加速系数, 它们都是正的常数; r_1 和 r_2 是两个在 $[0, 1]$ 范围内变化的随机数. 搜索时, 各微粒每一步的位置、速度都被最大位置、最小位置和最大速度、最小速度所限制

2.2 EPSSO 的主要改进措施

(1) 对 ω 的改进

在 PSO 中, ω 对算法能否收敛具有重要作用, 它使微粒保持运动惯性, 使其有扩展搜索空间的趋势, 有能力探索新的区域. ω 值大有利于全局搜索, 收敛速度快, 但不易得到精确的解; ω 值小有利于局部搜索, 能得到更为精确的解, 但收敛速度慢. 最初版本的 PSO 中, ω 为常数; 后来有研究表明^[3, 6]: 为使算法具有较强的全局搜索能力, 开始 ω 应大些, 然后逐步减小 ω , 以便获得更精确的解. 一般地, 开始时 $\omega = 1.4$, 然后逐步线性减小到 $\omega = 0.35$.

由 PSO 微粒的搜索特征不难发现, 线性减小 ω 使其保持较大值和较小值的时间都很短, 不能满足开始搜索速度快些、搜索后期速度慢些的要求. 为此, 本文提出惯性权值 ω 随着搜索的进行, 不是线性减小而是按余弦规律减小. 这样, 开始搜索时 ω 能较长时间保持较大值以提高搜索效率, 在搜索后期 ω 又能较长时间保持较小值以提高搜索精度. ω 的修正公式为

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{1 + \cos((\text{iter} - 1)\pi / (\text{MaxStep} - 1))}{2}. \quad (3)$$

其中: ω_{\max} 为搜索开始时最大的 ω , ω_{\min} 为搜索结束时最小的 ω ; iter 为迭代所进行的步数, MaxStep 为允许最大迭代步数

(2) 对最好微粒和最坏微粒的处理

在 PSO 中, 各微粒的飞行都是随机的, 当最优微粒飞行一步后, 如果适应值变差, 下一步就以变差的位置继续飞行, 没能发挥最优微粒应有的优势. 对于每一步最优微粒的飞行, 本文采用试探法使其朝着适应值变优的方向飞行, 即最优微粒飞行一步后, 如果适应值变好, 则以新的位置继续飞行; 如果适应值变差, 则返回原位置重新搜索. 对于每一步的最差

微粒, 本文用该步的历史全局最优微粒取代

(3) 防止陷入局部极值和稳定性变差

在 PSO 中, 搜索陷入局部极值往往表现为微粒几乎停止不动. 为防止微粒陷入局部极值, 本文约定: 如果某微粒连续 5 步近似不动, 则重新初始化该微粒的飞行速度. 在 PSO 中, 算法稳定性变差往往表现为微粒一直以很快的速度飞行. 为改善 PSO 的搜索稳定性, 本文约定: 如果某微粒连续 5 步飞行速度超过某极限值而适应值又得不到改善, 则重新初始化该微粒的飞行速度.

2.3 EPSSO 算法流程

1) 初始化设置群体的规模、参数维数、惯性权值、加速系数、最大允许迭代次数或适应值限、各微粒的初始位置和初始速度等

2) 按预定准则评价各微粒的初始适应值

3) 根据式 (3) 计算惯性权值

4) 根据式 (1) 计算各微粒新的速度

5) 判断各微粒的速度, 如有必要, 则重新初始化微粒速度

6) 对各微粒新的速度进行限幅处理

7) 根据式 (2) 计算各微粒新的位置, 并对各微粒新的位置进行限幅处理

8) 比较每个微粒的当前适应值和历史最好适应值, 若当前适应值更优, 则令当前适应值为该微粒历史最好适应值, 并保存该微粒的位置为其个体历史最好位置; 比较群体所有微粒的当前适应值和群体历史最好适应值, 若当前适应值更优, 则令当前适应值为历史全局最好适应值, 并保存历史全局最优位置

9) 对每一步的最好微粒和最差微粒进行处理

10) 如果满足停止条件 (适应值误差小于设定的适应值误差限或迭代次数超过最大允许迭代次数), 则搜索停止, 输出历史全局最优位置和历史全局最优适应值即为所求结果; 否则, 返回 3) 继续搜索

3 测试函数的优化

3.1 所用测试函数

1) F_1 函数

$$f(x_1, x_2) = 21.5 + x_1 \sin(47\pi x_1) + x_2 \sin(207\pi x_2),$$

$$-3.0 \leq x_1 \leq 12.1, -4.1 \leq x_2 \leq 5.8 \quad (4)$$

该函数只有一个全局最大点 (11.625 5, 5.725), 最大值为 38.850 3. 它很容易陷入局部极大值 38.732 8, 38.619 9, 38.429 1, 33.007 7 等, 是一个很难的多峰值优化函数

2) F_2 函数

$$f(x_1, x_2) = \frac{\sin\left(\sqrt{(x_1 - 50)^2 + (x_2 - 50)^2} + e\right)}{\sqrt{(x_1 - 50)^2 + (x_2 - 50)^2} + e} + 1, \quad (5)$$

该函数在 (50, - 50) 处取得全局最大值 1. 151 1, 其第 2 极大值为 1. 128 37, 是一个很难的多峰值优化函数

3) Schaffer F_6 函数

$$f(x_1, x_2) = 0.5 - \frac{(\sin\sqrt{x_1^2 + x_2^2} - 0.5)}{(1 + 0.001(x_1^2 + x_2^2))^2}, \quad (6)$$

该函数有无数个极大点, 其中只有一个 (0, 0) 为全局最大点, 最大值为 1. 最大值峰周围有一圈脊, 它们的取值均为 0. 990 283, 因此很容易陷入此处的局部极大点, 是一个很难的多峰值优化函数

4) Shubert 函数

$$f(x, y) = \left\{ \prod_{i=1}^5 \cos[(i+1)x_i + i] \right\} \times \left\{ \prod_{i=1}^5 \cos[(i+1)x_2 + i] \right\} + 0.5[(x_1 + 1.42513)^2 +$$

$$(x_2 + 0.80032)^2], \quad (7)$$

该函数有 760 个局部极小点, 其中只有一个 (- 1. 425 13, - 0. 800 32) 为全局最小点, 最小值为 - 186 730 9, 很容易陷入局部极小点 - 186 34

5) Griewangk 函数 (只讨论二元情况)

$$f(x) = \frac{1}{4000} \prod_{i=1}^n \frac{x_i^2}{i} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (8)$$

该函数有一个全局最小点 (0, 0), 最小值为 0; 局部极小点为 (3. 14, 4. 438 4), 局部极小值为 0. 007 4

3.2 优化结果

本文同时用 PSO 和 EPSO 对以上函数进行优化寻优时, PSO 和 EPSO 的微粒个数均为 80, 最大迭代步数均为 3 000, 惯性权值都从 1. 8 减小到 0. 08; 迭代时, PSO 和 EPSO 的随机数 r_1 和 r_2 都相同, 加速系数 c_1 和 c_2 均为 2, 误差限均为 10^{-6} . 由于 PSO 和 EPSO 都是随机搜索, 每次搜索结果可能不同

本文对每一测试函数都进行 100 次独立测试, 测试结果的统计对比如表 1 所示

表 1 测试结果统计对比

序号	测试函数	PSO 算法				EPSO 算法			
		达最优 概率 /%	适应值 平均值	适应值 最大值	适应值 最小值	达最优 概率 /%	适应值 平均值	适应值 最大值	适应值 最小值
1	F_1 函数	2	38 735	38 850	38 733	71	38 815	38 850	38 733
2	F_2 函数	25	1. 134 3	1. 151 1	1. 128 4	90	1. 148 8	1. 151 1	1. 128 4
3	Schaffer F_6	84	0. 998 4	1	0. 990 3	100	1	1	1
4	Shubert	66	- 186 59	- 186 73	- 186 34	100	- 186 73	- 186 73	- 186 73
5	Griewangk	93	0. 000 3	0. 007 4	0	100	0	0	0

由表 1 可见, 在算法参数相同的情况下, EPSO 获得最优解的概率明显高于 PSO, EPSO 比 PSO 更容易找到全局最优解, 优化效率和优化性能明显提高

4 基于 EPSO 的神经网络

人工神经网络 (NN) 能根据训练样本的输入输出数据, 自动拟合其中复杂的非线性关系, 其信息的分布式存储使它具有较强的抗干扰能力, 因而在众多领域得到广泛的应用^[10]. 本文将 EPSO 用于三层前向网络连接权值和阈值的训练, 构造了基于 EPSO 算法的三层前向网络, 用于 FCCU 主分馏塔粗汽油干点软测量

4.1 EPSONN 的结构

当采用前向网络时, EPSONN 的网络结构与

BPNN 的网络结构相同, 所不同的只是学习算法. 三层前向网络的结构如图 1 所示. 在 EPSONN 中, 先以 NN 的各连接权值和各阈值构成每一微粒的位置参数, 根据目标函数计算各微粒的适应值; 再用 EPSO 算法搜索 NN 的最佳权值和最优阈值, 一旦搜索完毕, 最优微粒的位置就是 NN 的最佳连接权值和最优阈值的组合.

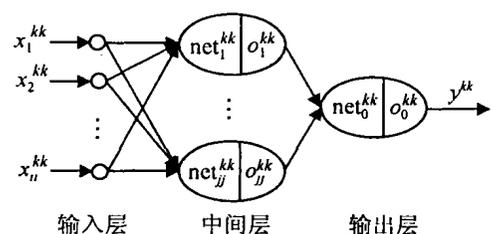


图 1 三层前向网络结构

图1中, x 为神经网络的输入, net 为神经元的输入权重和, o 为神经元的输出, y 为神经网络的输出, 上标 kk 为样本号. 假如输入层节点数为 ii , 中间层节点数为 jj , 输出层节点数为 1, 输入层节点到中间层节点的连接权值为 $w_1(ii, jj)$, 中间层节点神经元的阈值为 $b_1(jj)$, 中间层节点到输出层节点的连接权值为 $w_2(jj)$, 输出层节点神经元的阈值为 b_2 , 则微粒的维数为 $(ii + 2)jj + 1$.

4.2 EPSONN 学习算法流程

1) 初始化设置神经网络的结构、传递函数、目标函数等

2) 初始化设置微粒群的规模、参数维数、惯性权值、加速系数、最大允许迭代次数或适应值误差限、各微粒的初始位置和初始速度等

3) 前向计算神经网络直至输出, 并按预定准则评价各微粒的初始适应值

4) 根据式(3) 计算惯性权值

5) 根据式(1) 计算各微粒新的速度

6) 判断各微粒的速度, 如有必要, 则重新初始化微粒速度

7) 对各微粒新的速度进行限幅处理

8) 根据式(2) 计算各微粒新的位置, 并对各微粒新的位置进行限幅处理

9) 前向计算神经网络直至输出, 并按预定准则重新评价各微粒的适应值

10) 比较每个微粒的当前适应值和历史最好适应值, 若当前适应值更优, 则令当前适应值为该微粒历史最好适应值, 并保存该微粒的位置为其个体历史最好位置; 比较群体所有微粒的当前适应值和群体历史最好适应值, 若当前适应值更优, 则令当前适应值为历史全局最好适应值, 并保存历史全局最优位置

11) 对每一步的最好微粒和最差微粒进行处理

12) 若满足停止条件, 则搜索停止, 输出历史全局最优位置即为所求神经网络的最佳权值和最优阈值; 否则, 返回4) 继续搜索

5 基于 EPSONN 的粗汽油干点软测量

5.1 FCCU 主分馏塔工艺简介

本文以某 80 万吨/年的催化裂化装置为背景, 建立其主分馏塔粗汽油干点软测量模型. 主分馏塔是 FCCU 的主要分离设备, 塔的进料直接来自催化裂化反应器, 其中包括干气、液化气、汽油、柴油、循环油以及油浆等. 这些反应产物进入分馏塔底部, 与上部回流的循环油浆逆流接触, 被冷却洗涤后进入第一层塔板. 经过塔顶循环、一中循环、二中循环以

及塔底循环, 最终分离得到富气、粗汽油、柴油、回炼油及油浆. 其工艺流程示意如图 2 所示.

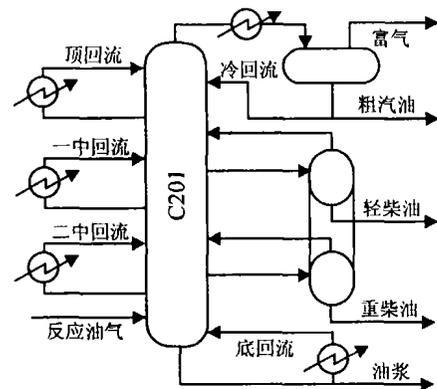


图2 FCCU 主分馏塔工艺流程示意

5.2 基于 EPSONN 的粗汽油干点软测量模型

将 EPSONN 用于 FCCU 主分馏塔粗汽油干点的软测量建模, 构建出基于 EPSONN 的主分馏塔粗汽油干点软测量模型. 通过对样本数据的主元分析, 对主分馏塔的工艺机理分析以及多次测试, 最终确定本文的软测量模型为: 由塔顶出口压力、塔顶油气出口温度、塔顶回流温度、顶回流抽出温度、18 层温度、9 层温度、一中返塔温度、油气进料温度、一中回流流量共 9 个输入信号, 中间层为 20 个节点, 输出信号为粗汽油干点, 构成的结构为 9-20-1 的神经网络. 神经元的传递函数均选用双曲正切函数, 模型的目标函数为

$$\min E = \frac{1}{2} \sum_{kk=1}^{n_p} (t^{kk} - y^{kk})^2 \quad (9)$$

其中: t 为粗汽油干点的实际值, y 为粗汽油干点的模型计算值, kk 为样本号, n_p 为样本总数.

在用 EPSO 寻找神经网络的最佳权值和最优阈值时, 各参数确定如下: 微粒个数为 100, 最大允许迭代次数为 20 000, 误差限为 0.2, 参数维数为 221, 惯性权值从 1.8 按余弦规律衰变到 0.06, 加速系数均为 2.

为了便于比较, 本文还用 BP 算法来训练神经网络. BPNN 模型结构、神经元传递函数等与 EPSONN 模型完全相同. BP 算法的学习速率为 0.016, 动量因子为 0.012.

6 研究结果与分析比较

6.1 训练过程与学习结果分析

采用经过检错、滤波、归一化预处理的 77 组训练样本, 对 EPSONN 模型和 BPNN 模型进行训练. 两种模型学习结果的对比如表 2 所示.

由表 2 可以看出, 两种模型的预测值与化验值的拟合程度较好, 训练过程满足要求, 并且

表 2 两种模型学习结果对比

模型指标	EPSONN 模型	BPNN 模型
$ e < 1$ 所占百分比	69.2%	64.9%
$ e < 2$ 所占百分比	100%	96.1%
均方差/	0.9248	1.0626
绝对误差平均值/	0.7671	0.8483

EPSONN 模型的拟合性能优于 BPNN 模型

6.2 检验结果与分析

采用同样经检错、滤波、归一化预处理的 50 组未参加训练的检验样本, 对 EPSONN 模型和 BPNN 模型进行检验。两种模型检验结果的对比如表 3 所示, 检验结果与实际化验结果的对比如图 3 所示。

表 3 两种模型检验结果对比

模型指标	EPSONN 模型	BPNN 模型
$ e < 1$ 所占百分比	56%	54%
$ e < 2$ 所占百分比	89%	86%
均方差/	1.3487	1.4425
绝对误差平均值/	1.1014	1.1284

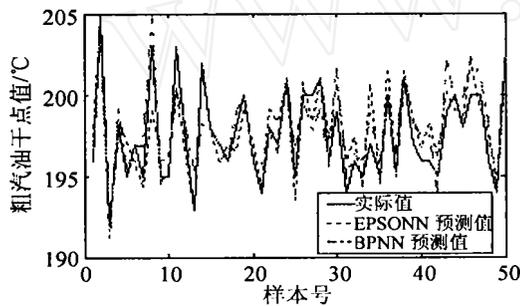


图 3 两种模型检验结果对比

由表 3 和图 3 可以看出, 两种模型的外推性较好, 模型具有较高的预测精度, 并且 EPSONN 模型的外推性能优于 BPNN 模型。

6.3 两种模型性能比较

1) BP 算法在寻找最优权值和最优阈值时, 单一初始值以梯度下降模式进行; 而 EPSONN 是以一群微粒即很多初始值, 采用依据自身经验并结合群体经验的有导向的随机搜索模式进行。对复杂的炼油装置建立产品质量软测量模型时, EPSONN 算法陷入局部极小的概率大大降低, 比 BP 算法更容易找到全局最优解。

2) 如果模型结构、传递函数和样本数据相同, 基于 EPSONN 的模型和基于 BPNN 的模型都能找到最优参数, 则两种模型的性能(拟合性能和外推性能)相当。

3) BP 算法在训练 NN 时, 反向调节权值采用

梯度下降模式, 要求出神经元传递函数和目标函数的导数, 导致它对目标函数和传递函数都有要求, 隐含层数也不宜过多; 而 EPSONN 在搜索时采用依据自身经验并结合群体经验的随机搜索模式, 无需求导计算, 对神经网络的层数、模型的目标函数和传递函数等没有限制, 因而应用更为灵活方便, 并且范围大为拓宽。

7 结 语

本文提出的 EPSONN 是一种改进的 PSO, 它能有效地搜索到全局最优解, 优化效率和优化性能较 PSO 有明显的提高。将 EPSONN 用于粗汽油干点软测量, 得到的基于 EPSONN 的软测量模型比基于 BPNN 的软测量模型具有更高的精度和更好的性能, 为实现炼油装置产品质量的直接控制, 进而实现综合控制提供了有利的依据。

参考文献 (References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. *Proc IEEE Int Conf on Neural Networks* [C]. Perth, 1995: 1942-1948.
- [2] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory [A]. *Proc the Sixth Int Symp on Micro Machine and Human Science* [C]. Nagoya, 1995: 39-43.
- [3] Eberhart R C, Shi Y. Particle swarm optimization: Developments, applications and resources [A]. *Proc 2001 Congress on Evolutionary Computation* [C]. Seoul, 2001: 81-86.
- [4] Parsopoulos K E, Vrahatis M N. Recent approaches to global optimization problems through particle swarm optimization [J]. *Natural Computing*, 2002: 235-306.
- [5] Claudia O Ourique, Evaristo C Bisciaia, Jr Jose Carlos Pinto. The use of particle swarm optimization for dynamical analysis in chemical processes [J]. *Computers and Chemical Engineering*, 2002, 26: 1783-1793.
- [6] Shi Y, Eberhart R C. A modified particle swarm optimizer [A]. *Proc IEEE Int Conf on Evolutionary Computation* [C]. Anchorage, 1998: 69-73.
- [7] 俞金寿, 等. *软测量技术及其在石油化工中的应用* [M]. 北京: 化学工业出版社, 2000.
- [8] Thompson M L, Kramer M A. Modeling chemical process using prior knowledge and neural networks [J]. *AIChE J*, 1994, 40(8): 1328-1340.
- [9] 林世雄. *石油炼制工程* [M]. 第 2 版. 北京: 石油工业出版社, 1988.
- [10] 焦李成. *神经网络系统理论* [M]. 西安: 西安电子科技大学出版社, 1995.