

文章编号:1001-9081(2007)03-0730-03

多速粒子群优化算法及其在软测量中的应用

须文波,杜润龙

(江南大学 信息工程学院,江苏 无锡 214122)

(du_2000@163.com)

摘要:多速粒子群优化算法(MVPSO)是一种改进的粒子群优化算法,具有概念清晰、操作简单、易实现等优点,同时克服了标准 PSO 算法易陷入局部极值的不足。用 MVPSO 和 PSO 对几种典型多峰值函数优化问题进行测试,结果表明 MVPSO 优化算法更容易找到全局最优解,优化效率和优化性能明显提高。将 MVPSO 优化算法应用于青霉素发酵过程产物(青霉素)浓度软测量,建立基于 MVPSO 算法的青霉素发酵过程产物浓度软测量模型。实验表明,基于 MVPSONN 的软测量模型比基于 BPNN 的软测量模型具有更好的性能。

关键词:粒子群优化算法;多速;软测量;发酵过程

中图分类号: TP18;TP273 **文献标识码:** A

Multi-velocity particle swarm optimization and its application in soft sensor

XU Wen-bo, DU Run-long

(School of Information Technology, Southern Yangtze University, Wuxi Jiangsu 214122, China)

Abstract: Multi-velocity particle swarm optimization algorithm (MVPSO) is an improved PSO algorithm, which can overcome the shortcomings of easily getting in the local extremum. Besides, this improved algorithm is characterized by clear definition, simple operation and easy realization. Both MVPSO and PSO were used to resolve several typical test function optimization problems. The results indicate that MVPSO has higher efficiency, better performance and more advantages in many aspects than PSO. MVPSO was applied to train artificial neural network to construct a practical soft sensor of product concentration of penicillin fermentation process. The experimental results indicate that this method is feasible and effective in soft sensor application.

Key words: particle swarm optimization; multi-velocity; soft sensor; fermentation process

0 引言

粒子群优化算法(Particle Swarm Optimization, PSO)^[1,2]是一种新的进化计算算法,其概念和参数调整简单、易实现,既保持了传统进化算法的优点,又具有自己许多良好的优化性能,因而得到广泛应用。但由于其收敛的全局最优性问题在理论上并未得到证明^[3],存在易陷入局部极值问题,即“早熟”现象。

针对 PSO 算法易陷入局部极值问题,一些学者提出将“免疫”和“遗传”的概念引入 PSO 算法^[4]。也有一些学者提出增加群体多样性的方法^[5]。这些改进方法在一定程度上克服了“早熟”现象,但这些改进方法概念复杂、编程实现困难,失去了 PSO 算法固有的简单易用的优点。本文在对 PSO 算法机理进行分析的基础上,介绍一种新颖的多速粒子群优化算法(Multi Velocity PSO, MVPSO)^[6],用常用多峰值优化函数进行测试比较,最后将 MVPSO 优化算法应用于青霉素浓度神经网络软测量模型,并与基于 BP 算法的青霉素浓度神经网络软测量模型进行比较,结果表明,基于 MVPSO 的软测量模型具有良好性能。

1 多速粒子群优化算法

1.1 PSO 算法

PSO 算法是 1995 年由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同提出的,基本思想是模拟鸟类群体行为,并利用了生物学家所提出的生物群体模型。

假设在一个 D 维搜索空间中,有 m 个粒子组成一粒子群,其中第 i 个粒子的空间位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$,第 i 个粒子所经历的最好位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$,每个粒子的飞行速度为 $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$, $i = 1, \dots, m$ 。在整个粒子群中,所有粒子经历的最好位置为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 。每一代粒子的第 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,各粒子每一步的位置、速度都被最大位置、最小位置和最大速度、最小速度所限制。

1.2 多速粒子群优化算法

在标准粒子群算法中,决定每个粒子搜索效果的关键因素

收稿日期:2006-09-13;修订日期:2006-11-16

作者简介:须文波(1946-),男,江苏无锡人,教授,博士生导师,主要研究方向:智能控制、软测量技术;杜润龙(1979-),男,安徽阜南人,硕士研究生,主要研究方向:软测量技术。

是速度。粒子通过随机地改变速度,使自己在解空间中到达新的位置。通过比较不同位置的适应值,每个粒子都试图“飞”向最佳位置,从而得到最优解。由公式(1)可知,如果 x_{id} 非常接近 p_{id} 和 p_{gd} ,同时惯性系数 w 近似为 0,粒子在 p_{gd} 邻域内将会停止移动,失去搜索解空间的能力,从而产生“早熟”现象。粒子的速度也不应该太快,否则将会“飞”出解空间。

因此,粒子群优化算法优化性能的关键问题是如何处理速度问题。使粒子既能以较大的速度在整个解空间上搜索,又能以较小的速度作局部搜索,以得到精度较高的全局最优解。MVPSO 优化算法的基本原理是:粒子在同一个方向上以不同的速度到达不同的位置,选取两个极值更新粒子的速度,逐步得到全局最优解。较大的速度能够使粒子在整个解空间上进行全局性搜索,避免“早熟”;较小的速度能够使粒子作细微的局部搜索,提高解的精度。

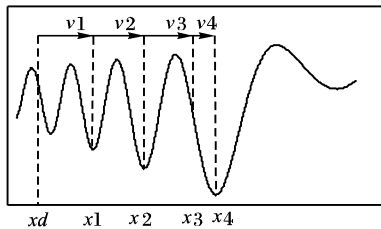


图 1 粒子多速搜索示例

如图 1 所示,粒子在同一个方向上,以 4 种不同的速度 v_1, v_2, v_3, v_4 到达 4 个位置 x_1, x_2, x_3, x_4 ,通过比较得到优化值 x_4 而非 x_1 ,避免了“早熟”现象。

MVPSO 算法中每一代粒子的第 d 维 ($1 \leq d \leq D$) 根据如下方程变化:

$$\begin{cases} v_{id}^0 = w \times v_{id}^0 + c_1 \times r_1 \times (G_P_{id} - x_{id}^0) + c_2 \times r_2 \times (G_P_{gd} - x_{id}^0) \\ x_{id}^0 = x_{id}^0 + v_{id}^0 \\ v_{id}^1 = k \times a(1) \times v_{id}^0 \\ v_{id}^2 = k \times a(2) \times v_{id}^0 \\ \vdots \\ v_{id}^j = k \times a(j) \times v_{id}^0 \\ x_{id}^1 = x_{id}^0 + v_{id}^1 \\ x_{id}^2 = x_{id}^0 + v_{id}^2 \\ \vdots \\ x_{id}^j = x_{id}^0 + v_{id}^j \end{cases} \quad (3)$$

其中, v_{id}^0 为参考速度的第 d 维, $v_{id}^m, m = 1, 2, \dots, j$, 是搜索速度的第 d 维。参考速度决定搜索速度,搜索速度决定粒子的位置; G_P_{id} 为第 i 个粒子在不同速度上所经历最好位置时的第 d 维; G_P_{gd} 为群中所有粒子在不同速度上所经历最好位置时的第 d 维; k 为速度系数,用于把速度限定在一定的区域上; $a(m), m = 1, 2, \dots, j$ 为速度变量系数,用于设定参考速度与搜索速度之间的关系,当参考速度过大, $a(m)$ 应该使搜索速度降低;当参考速度过小, $a(m)$ 应该使搜索速度增加;当参考速度适中, $a(m)$ 应该在一定范围内调整搜索速度。这样才会使粒子具有较强的全局搜索和局部细微搜索能力。 $a(m)$ 的公式如下:

$$a(m) = \begin{cases} m, & M_1 < v_{id}^0 < M_2 \text{ and } m \geq j/2 \\ m/j + 1, & v_{id}^0 < M_1 \\ 1 - m/j, & M_1 < v_{id} < M_2 \text{ and } m < j/2 \\ m/j, & v_{id}^0 \geq M_2 \end{cases} \quad (4)$$

每个粒子的速度都限定在一定的范围内:

$$v_{id}^m = \begin{cases} v_{\max}, & v_{id}^m > v_{\max} \\ v_{\min}, & v_{id}^m < v_{\min} \end{cases}, m = 0, 1, \dots, j \quad (5)$$

1.3 MVPSO 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误差限、各粒子的初始位置(随机)和初始速度(随机)等;
- 2) 根据公式(3)~(5)计算每个粒子的不同速度和位置;
- 3) 计算每个粒子对应于不同速度下的适应值;
- 4) 比较每个粒子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粒子历史最好适应值,并保存该粒子的位置为其个体历史最好位置;
- 5) 比较群体所有粒子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;
- 6) 检查是否满足停止条件(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),如满足,则搜索停止,输出历史全局最优位置和全局最优适应值;否则,返回 2) 继续搜索。

2 测试函数的优化

2.1 所用测试函数

1) F_1 函数

$$f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2), -3.0 \leq x_1 \leq 12.1, 4.1 \leq x_2 \leq 5.8 \quad (6)$$

该函数只有一个全局最大点 (11.6255, 5.725), 最大值为 38.8503。它很容易陷入局部极大值 38.7328, 38.6199, 38.4291, 33.0077 等, 是一个很难的多峰值优化函数。

2) Shubert 函数

$$f(x_1, x_2) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \right\} \times \left\{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \right\} + 0.5[(x_1 + 1.42513)^2 + (x_2 + 0.80032)^2], -10 \leq x_1, x_2 \leq 10 \quad (7)$$

该函数有 760 个局部极小点, 其中只有一个 (-1.42513, -0.80032) 为全局极小点, 最小值为 -187.7309, 很容易陷入局部极小点 -186.34。

3) Schaffer F_6 函数

$$f(x_1, x_2) = 0.5 - \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}, -10 \leq x_1, x_2 \leq 10 \quad (8)$$

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, 取值均为 0.990283, 因此很容易陷入此处的局部极大值, 是一个很难的多峰值优化函数。

2.2 优化结果

用标准 PSO 和 MVPSO 对以上函数进行优化, 粒子个

数均为80,最大迭代步数均为3000,惯性权值从1.8递减到0.08;迭代时,加速系数 c_1 和 c_2 均为2,误差限均为0.000001。

由于 PSO 和 MVPSO 都是随机搜索,每次搜索结果可能

不同,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。可见,在算法参数相同的情况下,MVPSO 比标准 PSO 算法更容易获得全局最优解,优化效率和优化性能明显提高。

表 1 测试结果统计对比

序号	测试函数	PSO 算法			MVPSO 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	14	38.749	38.850	38.733
2	Shubert	66	-186.59	-186.73	-186.34	83	-186.66	-186.73	-166.99
3	Schaffer F ₆	84	0.9984	1.000	0.9903	100	1	1	1

3 基于 MVPSO 的神经网络

人工神经网络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,因此在众多领域得到广泛的应用^[7]。本文将 MVPSO 应用于三层前向网络权值和阈值的训练,构建基于 MVPSO 的三层前向神经网络模型,用于青霉素发酵过程青霉素浓度预估软测量。

MVPSO 神经网络(MVPSO-NN)与 BP 神经网络的网络结构相同,所不同的只是学习算法。最常见的三层前向网络结构如图 2 所示。

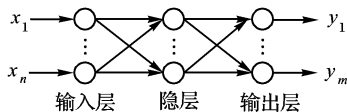


图 2 三层前向神经网络结构

其中 x_1, x_2, \dots, x_n 为输入变量, y_1, y_2, \dots, y_m 为输出变量,假如输入层节点数为 n ,输出层节点数为 m ,隐层节点数为 k ,输入层节点到隐层节点的连接权值为 $w_{1(i,j)}$,阈值为 $b_{1(j)}$ ($1 \leq i \leq n, 1 \leq j \leq k$),隐层节点到输出层节点的连接权值为 $w_{2(i,j)}$,阈值为 $b_{2(j)}$, ($1 \leq i \leq k, 1 \leq j \leq m$),则每个粒子的维数为 $nk + k + km + m$ 。

以神经网络的各连接权值和阈值构成每一粒子的位置参数,根据目标函数计算各粒子的适应值,再用 MVPSO 算法搜索最优权值和阈值,全局历史最优粒子的位置参数就是优化了的权值和阈值的组合。

4 基于 MVPSO-NN 的青霉素浓度软测量

青霉素的生产方法是利用产黄青霉进行分批发酵,用葡萄糖、玉米浆为培养基。调节温度、pH 等变量来控制发酵过程^[8]。通过对青霉素发酵过程分析,选取温度 $T(k)$ 、pH (k) 、溶氧值 $DO(k)$ 、通风量 $Air(k)$ 、菌体浓度 $X(k)$ 、青霉素浓度 $c_p(k)$ 、以及发酵时间 k 等 7 个变量为模型的输入。模型输出为青霉素浓度 $c_p(k+1)$ 。模型描述为:

$$c_p(k+1) = f(T(k), pH(k), DO(k), Air(k), X(k), c_p(k), k) \quad (9)$$

其中 $f(\cdot)$ 表示输入输出间的非线性关系。

输入层神经元 7 个,隐层神经元 9 个,单隐层结构,构成 7-9-1 结构的神经网络。隐层神经元激活函数为 logsig,输出层神经元激活函数为 purelin,目标函数为:

$$\min E = \frac{1}{2} \sum_{kk=1}^{n_p} (c_p^{kk} - y^{kk})^2 \quad (10)$$

其中: c_p 为青霉素浓度实际值, y 为青霉素浓度模型预估

值, kk 为样本号, n_p 为样本总数。

用 MVPSO 优化权值和阈值时的各参数确定如下:粒子个数 40,最大允许迭代次数 3 000,惯性系数 w 从 1.2 线性衰减到 0.2,误差限为 0.000 1,加速度系数均为 2,速度系数 $k = 0.5$,速度变量系数 $a(m)$ 中的 $m = 3$,每个粒子的维数为 82。粒子初始化时的随机位置的每一维限定在 $[-10, 10]$ 范围内。BPNN 的参数设定为:学习速率为 0.016,动量因子为 0.012,学习误差为 0.000 1,最大训练次数为 3 000。

5 实验结果与分析

采用生物发酵实验室青霉素发酵小试过程数据进行建模仿真,发酵过程约持续 208 h,每 4 h 采样 1 次,发酵初始状态数据 1 组,每批次有 53 组数据,由于要预测下一时刻青霉素浓度,将 k 时刻的温度 $T(k)$ 、pH (k) 、溶氧值 $DO(k)$ 、通风量 $Air(k)$ 、菌体浓度 $X(k)$ 、青霉素浓度 $c_p(k)$ 、以及发酵时间 k 作为模型输入, $k+1$ 时刻青霉素浓度作为模型输出。第 53 组数据因没有下一时刻的实际浓度而不能参加训练,每批有 52 组数据参加训练网络。共 10 批生产数据,每批数据都表示一个完整的发酵过程,其中 9 批数据(468 个样本)用来训练模型,1 批数据用来检验模型的泛化能力。为了提高模型的精确度,对样本数据进行归一化处理,归一化公式选为:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (11)$$

式中: x' 为归一化后的数据, x 为原始样本数据, x_{\max} 、 x_{\min} 为样本数据的最大值、最小值。归一化后样本数据在 $[0, 1]$ 之间。训练结束后,统计结果见表 2。

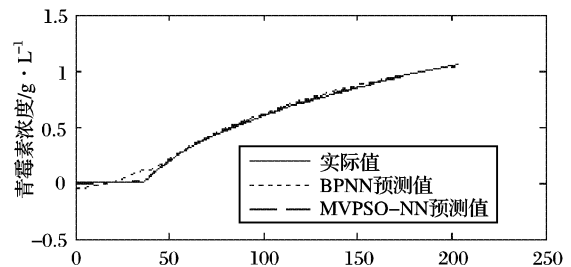


图 3 两种模型的青霉素浓度预估曲线

表 2 两种模型学习性能对比

性能指标	MVPSO-NN	BPNN
$ e < 0.01$ (g/L) 所占比例	70.1%	63.2%
$ e < 0.02$ (g/L) 所占比例	96.4%	90.2%
均方差	0.005 75	0.024 20
绝对误差平均值	0.004 23	0.020 20

动延迟的影响,以文件最近 K 次访问情况为基础建立缓存效用作为替换依据,每次替换缓存效用最小的文件的媒体段。SCU-K 算法可以根据文件的访问情况动态调整文件的缓存大小,让文件的缓存大小适应媒体流行度、字节有用性和已缓存部分大小的变化,使越靠近起始部分的媒体段具有相对越高的缓存效用,降低了文件前缀部分被替换的概率,避免 LRU 和 LFU 算法中出现的媒体文件被连续替换而最终被完全替换出缓存的问题。在与 LRU,LFU,LRU-K 算法的对比实验中,SCU-K 算法在提高空间利用率,缓存字节命中率 and 降低启动延迟方面具有更大的优势。

参考文献:

- [1] 钟玉琢, 向哲, 沈洪, 等. 流媒体和视频服务器 [M]. 北京: 清华大学出版社, 2003. 117 - 158.
- [2] SEN S, REXFORD J, TOWSLEY D. Proxy prefix caching for multimedia streams [A]. Proceedings of IEEE Infocom [C]. New York, 1999. 1310 - 1319.
- [3] LIM E, PARK SH, HONG HO, *et al.* A proxy caching scheme for continuous media streams on the Internet [A]. The 15th International Conference on Information Networking (ICOIN'01) [C]. Beppu City, Oita, Japan, 2001. 720 - 725.
- [4] WU K, YU PS, WOLF J. Segment-based proxy caching of multimedia streams [A]. Proceedings of WWW [C]. Hongkong, China, 2001. 56 - 60.
- [5] CHEN S, SHEN B, WEE S, *et al.* Adaptive and lazy segmentation based proxy caching for streaming media delivery [A]. Proceedings of ACN NOSSDAV [C]. Monterey, CA, 2003. 694 - 703.
- [6] OTOO E, OLKEN F, SHOSHANI A. Disk cache replacement algorithm for storage resource managers in data grids [A]. Proceedings of IEEE / ACM Conference on Supercomputing [C]. Baltimore,

Maryland, USA, 2002. 1 - 15.

- [7] ROBINSON JT, DEVARAKONDA MV. Data cache management using frequency-based replacement [A]. Proceedings of SIGMETRICS on Measuring and Modeling of Computer Systems [C]. Boulder, Colorado, USA, 1990. 134 - 142.
- [8] SMARAGDAKIS Y, KAPLAN S, WILSON P. The EELRU adaptive replacement algorithm [J]. Elsevier Science Performance Evaluation, 2003, 53(2): 93 - 123.
- [9] SMARAGDAKIS Y, KAPLAN S, WILSON P. EELRU: Simple and effective adaptive page replacement [A]. Proceedings of SIGMETRICS on the Measurement and Modeling of Computer Systems [C]. Atlanta, Georgia, USA, 1999. 122 - 133.
- [10] 罗彬, 王宏远, 盛东晖, 等. 代理服务器视频对象缓存方法的研究 [J]. 武汉大学学报, 2005, 38(5): 141 - 145.
- [11] NEIL EJO, NEIL PEO, WEIKUM G. The LRU-K page replacement algorithm for database disk buffering [A]. Proceedings of International Conference on Management of Data [C]. Washington, DC, USA, 1993. 297 - 306.
- [12] NEIL EJO, NEIL PEO, WEIKUM G. An optimality proof of the LRU-K page replacement algorithm [J]. Journal of the ACM, 1999, 46(1): 92 - 112.
- [13] DAN A, SITARAM D. A generalized interval caching policy for mixed interactive and long video environments [A]. Proceedings of MMCN [C]. San Jose, CA, USA, 1996. 344 - 351.
- [14] 肖明忠, 李晓明, 刘翰宇, 等. 基于流媒体文件字节有用性的代理服务器缓存替代策略 [J]. 计算机学报, 2004, 27(12): 1633 - 1641.
- [15] SCHEUERMANN P, SHIN J, VINGRALEK R. WATCHMAN: A data warehouse intelligent cache manager [A]. Proceedings of VLDB [C]. Bombay, India, 1996. 51 - 62.

(上接第 732 页)

由表 2 可以看出, MVPSO-NN 模型的拟合性能优于 BPNN 模型。

用没有参加训练的第 10 批数据对两种模型进行检验, 结果如图 3 所示, 统计结果如表 3 所示。

表 3 两种模型泛化性能比较

性能指标	MVPSO-NN	BPNN
$ e < 0.01$ (g/L) 所占比例	66.3%	56.2%
$ e < 0.02$ (g/L) 所占比例	88.2%	85.2%
均方差	0.0065	0.0289
绝对误差平均值	0.0058	0.0218

由表 3 及图 3 可以看出, 两种模型的泛化性能较好, 具有较高的预测精度, 并且 MVPSO-NN 模型优于 BPNN 模型。

6 结语

多速粒子群算法 (MVPSO) 是一种改进的 PSO, 优化效率和优化性能较标准 PSO 有明显提高。本文将 MVPSO 算法应用于前向神经网络权值和阈值的优化, 试验结果显示, MVPSO 算法陷入局部极小值的概率显著降低, 比 BP 算法更容易找到全局最优解, 而且在应用上灵活方便, 目标函数与传递函数没有必须可导的限制, 应用范围大为拓宽。将基于 MVPSO 算法的神经网络应用于青霉素发酵过程产物 (青霉素) 浓度软测量, 仿真实验表明, 该模型训练误差小、学习速

度快、泛化能力强, 可以满足青霉素发酵产物浓度预估要求, 为实现发酵过程优化控制提供了有利依据。

参考文献:

- [1] KENNEDY J, EBERHART RC. Particle swarm optimization [A]. Proceedings of IEEE International Conference on Neural Networks [C]. Perth, 1995. 1942 - 1948.
- [2] EBERHART RC, KENNEDY J. A new optimizer using particle swarm theory [A]. Proceedings of the 6th International Symposium on Micro Machine and Human Science [C]. Nagoya, 1995. 39 - 43.
- [3] VAN DEN BERGH F. An Analysis of Particle Swarm Optimizers [D]. Department of Natural and Agricultural Science, University of Pretoria, Pretoria, South Africa, 2001.
- [4] YANG W. Survey on Particle Swarm Optimization Algorithm [J]. Engineering Science, 2004, 6(5): 87 - 94.
- [5] XIE XF, ZHANG WJ. A Dissipative Particle Swarm Optimization [A]. Proceedings of the Congress on Evolutionary Computation [C]. Honolulu, USA, 2002, 2. 1456 - 1461.
- [6] WANG B, DONG YL, WANG CL. A New Particle Optimizer Algorithm and Application [A]. 1st International Symposium on Systems and Control in Aerospace and Astronautics [C]. 2006, 1. 312 - 315.
- [7] 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, 1988.
- [8] 常玉清, 邹伟, 王福利. 基于支持向量机的软测量方法研究 [J]. 控制与决策, 2005, 20(11): 1307 - 1310.