

一种新的混合智能极限学习机

林梅金^{1,2}, 罗飞², 苏彩红¹, 许玉格²

(1. 佛山科学技术学院 机械与电气工程学院, 广东 佛山 528000;

2. 华南理工大学 自动化科学与工程学院, 广州 510640)

摘要: 提出一种基于差分进化(DE)和粒子群优化(PSO)的混合智能方法——DEPSO算法, 并通过对10个典型函数进行测试, 表明DEPSO算法具有良好的寻优性能. 针对单隐层前向神经网络(SLFNs)提出一种改进的学习算法——DEPSO-ELM算法, 即应用DEPSO算法优化SLFNs的隐层节点参数, 采用极限学习算法(ELM)求取SLFNs的输出权值. 将DEPSO-ELM算法应用于6个典型真实数据集的回归计算, 并与DE-ELM、SaE-ELM算法相比, 获得了更精确的计算结果. 最后, 将DEPSO-ELM算法应用于数控机床热误差的建模预测, 获得了良好的预测效果.

关键词: 粒子群优化算法; 差分进化算法; 极限学习机; 混合

中图分类号: TP18

文献标志码: A

An improved hybrid intelligent extreme learning machine

LIN Mei-jin^{1,2}, LUO Fei², SU Cai-hong¹, XU Yu-ge²

(1. Mechanical and Electrical Engineering College, Foshan University, Foshan 528000, China; 2. School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China. Correspondent: LIN Mei-jin, E-mail: lmjfans@163.com)

Abstract: An improved hybrid intelligent algorithm based on differential evolution(DE) and particle swarm optimization(PSO) is proposed. The performance of DEPSO algorithm is verified by simulations on 10 benchmark functions. Then, an improved learning algorithm named DEPSO extreme learning machine(DEPSO-ELM) algorithm for single hidden layer feedforward networks(SLFNs) is proposed. In DEPSO-ELM, DEPSO is used to optimize the network hidden node parameters, and ELM is used to analytically determine the output weights. Simulation results of 6 real world datasets regression problems show that the DEPSO-ELM algorithm performs better than DE-ELM and SaE-ELM. Finally, the effectiveness of the DEPSO-ELM algorithm is verified in the prediction of NC machine tool thermal errors.

Key words: particle swarm optimization; differential evolution; extreme learning machine; hybrid

0 引言

神经网络是对生物神经元细胞网络的抽象模拟, 具有自学习、并行处理等能力, 可以对大量数据进行学习, 从而实现函数逼近、系统辨识等功能, 在过去的几十年里得到了广泛的研究^[1]. 近年来, 一种新的人工神经网络模型学习方法——极限学习机(ELM)被提出^[2]. 这种学习方法在保证网络具有良好泛化性能的同时, 极大地提高了神经网络的学习速度, 并且避免了基于梯度下降学习方法的许多问题^[3], 如陷入局部极小、收敛速度慢等问题. 然而ELM在应用时需要先设定隐层节点数, 随机产生隐层节点参数(输入节点与隐层节点的连接权值和隐层节点的阈

值), 这些参数在网络训练阶段保持不变. 随机产生隐层节点参数往往会导致网络的一些隐层节点所起的作用非常小^[4]. 文献[2]指出, 为了保证ELM的性能, 常常需要设定一个较大的隐层节点数, 这使得网络在应用阶段将耗费大量的计算时间.

近年来, 采用智能优化算法结合ELM算法以提高SLFNs的性能已成为一个研究热点. 如: 文献[5]提出一种进化极限学习机(E-ELM), 结合差分进化与ELM两种算法的作用提高了SLFNs的性能; 文献[4]提出了SaE-ELM, 采用自适应DE算法优化SLFNs的隐层节点参数, 然后采用Moore-Penrose(MP)广义逆获得网络的输出连接权值; 文献[6]提出

收稿日期: 2014-03-09; 修回日期: 2014-06-19.

基金项目: 广东省自然科学基金项目(S2011010001153); 中央高校基本科研业务费专项重点项目(2014ZZ0037).

作者简介: 林梅金(1981—), 女, 讲师, 博士生, 从事智能控制与优化的研究; 罗飞(1958—), 教授, 博士生导师, 从事复杂系统控制等研究.

了IPSO-ELM,采用改进的粒子群算法优化SLFNs的隐层节点参数,利用ELM算法求得SLFNs的输出权值,从而提高了SLFNs的性能.这些基于智能算法的极限学习机应用在回归计算和分类上具有良好的性能^[7].

本文在上述工作的基础上,首先研究两种典型的智能优化算法:差分进化算法和粒子群优化算法,借鉴蛙跳算法(SFLA)^[8]的模因进化机制,提出一种新的混合智能优化算法(DEPSO),并选择10个典型函数进行测试,仿真结果验证了DEPSO算法比基本的DE、PSO和SFLA算法具有更好的寻优性能.然后,将DEPSO算法用于优化SLFNs的隐层节点参数,采用ELM算法求得SLFNs的输出权值,所得到的DEPSO-ELM算法应用在6种典型真实数据集的回归计算上,与文献[4]中的DE-ELM和SaE-ELM两种算法相比,获得了更精确的计算结果.最后,将DEPSO-ELM算法应用在数控机床热误差的建模预测上,获得了良好的预测效果.

1 差分粒子群混合智能优化算法(DEPSO)

1.1 差分进化算法

差分进化算法(DE)是由Storn等^[9]提出的一种基于群体演化的优化算法.它采用实数编码形式产生候选解,通过差分变异、二元交叉、贪心选择这3种操作进行最优解的探索过程,是一种全局优化性能和数值稳定性都非常出色的算法^[10].DE算法的计算过程可描述如下.

首先,根据实际问题,DE算法产生 N_P 个 D 维的解种群,其中解个体 $X_{i,G} = (x_{i1,G}, x_{i2,G}, \dots, x_{iD,G})$, $i \in \{1, 2, \dots, N_P\}$, G 表示迭代的代数.然后,对于每个解向量 $X_{i,G}$,应用差分变异策略产生对应的变异向量 $V_{i,G}$.在DE算法中,常用的差分变异策略有:DE/rand/1、DE/best/1、DE/best/2、DE/rand/2等^[10].本文采用DE/rand/2差分变异策略,即

$$V_{i,G} = X_{r_1^i,G} + F \cdot (X_{r_2^i,G} - X_{r_3^i,G}) + F \cdot (X_{r_4^i,G} - X_{r_5^i,G}). \quad (1)$$

其中:下标 $r_1^i, r_2^i, r_3^i, r_4^i, r_5^i$ 是从集合 $\{1, 2, \dots, N_P\}$ 中随机取出的两两不相同的随机数,且与 i 不同;缩放因子 F 是一个正的控制参数,用来控制差异向量的长度.

采用二元交叉操作作用于变异向量 $V_{i,G}$ 和对应的原解向量 $X_{i,G}$,产生最终的试探解 $U_{i,G} = (u_{i1,G}, \dots, u_{iD,G})$,有

$$u_{ij,G} = \begin{cases} v_{ij,G}, & \text{rand}() \leq C_R \text{ or } j = j_{\text{rand}}; \\ x_{ij,G}, & \text{rand}() > C_R \text{ and } j \neq j_{\text{rand}}. \end{cases} \quad (2)$$

其中: $j \in \{1, 2, \dots, D\}$; $u_{ij,G}, v_{ij,G}, x_{ij,G}$ 分别是试探

解、变异解和原解的第 j 个量;随机数 $\text{rand}() \in [0, 1]$; C_R 是交叉概率; j_{rand} 是从集合 $\{1, 2, \dots, D\}$ 中选择的随机数,以确保试探解与原解的不同.

最后,对试探解 $U_{i,G}$ 和原解 $X_{i,G}$ 进行选择算子操作,两者中较优者将作为新解 $X_{i,G+1}$ 保留到DE算法的下一代.

1.2 粒子群优化算法

粒子群优化算法(PSO)是Kennedy等^[11]于1995年提出的一种基于群体演化的优化算法.在PSO算法中,粒子群由 N_P 个粒子组成,粒子 $i \in \{1, 2, \dots, N_P\}$ 具有位置信息 X_i 和飞行速度信息 V_i .在算法迭代过程中,每个粒子在 D 维问题空间通过飞行方式来搜索信息,对所经过的位置进行评价.典型的PSO算法迭代方程为

$$V_{i,G+1} = w \times V_{i,G} + c_1 \times \text{rand}() \times (P_{i,G} - X_{i,G}) + c_2 \times \text{rand}() \times (P_{g,G} - X_{i,G}), \quad (3)$$

$$X_{i,G+1} = X_{i,G} + V_{i,G+1}. \quad (4)$$

其中: G 表示PSO算法迭代的代数, $P_{i,G}$ 表示第 i 个粒子在前 G 代迭代过程中自身发现的具有最优适应值的位置信息, $P_{g,G}$ 表示在前 G 代迭代过程中种群发现的具有最优适应值的位置信息,参数 w 称为惯性权重, c_1, c_2 称为学习因子.

1.3 DEPSO算法

近20年来,很多学者通过综合DE和PSO两个算法的优点,取长补短,以提高混合算法的寻优性能^[10].如:文献[12]的DEPSO算法在进化过程中每个个体根据预设的概率选择DE或者PSO来更新当前位置;文献[13]的DEPSO算法通过预定义概率 C_R 决定每个候选解的每一维分量的产生方式为DE或者PSO;文献[14]的自适应DEPSO算法通过进化过程中在线依据DE和PSO在统计周期内的相对成功率来确定个体的进化方法;文献[15]的DEPSO算法将PSO作为进化的主流程,在给定的周期内由DE算法改变粒子的位置;文献[16]的DEPSO算法其主体算法为PSO,采用DE算法在线调整PSO算法的3个参数 w, c_1 和 c_2 .

蛙跳算法的迭代过程为:分模因组-不同模因组独立进化-混合模因组,如此反复迭代进化.受此计算机制的启发,结合前人的工作,本文提出一种以DE和PSO算法为进化计算方法的具有蛙跳算法中模因进化能力的混合智能优化算法,以期利用两种算法的互补优势提高混合算法的性能.该算法描述如下.

Step 1: 在解空间随机产生某优化问题的 N_P 个解作为初始种群 $P = \{X_j\}_{j=1}^{N_P}$,初始化总进化代数 I_{itermax} ,子种群进化代数 $I_{\text{iter}}, C_n = I_{\text{itermax}}/I_{\text{iter}}$.

Step 2: 将种群 P 随机平均分割成 N_k 个子种群.

Step 3: 随机选择 k ($1 < k < N_k$) 个子种群, 分别应用 DE 算法迭代计算 I_{iter} 代; 剩下 $N_k - k$ 个子种群, 分别应用 PSO 算法迭代计算 I_{iter} 代; 记录迭代过程整个种群的最优值变化情况.

Step 4: 混合 N_k 个子种群获得新种群 P ; 判断子种群混合次数是否达到 C_n 次, 如果否, 则转 Step 2, 如果是, 则转 Step 5.

Step 5: 算法停止.

1.4 DEPSO 算法的仿真实验

本文选取 10 个典型函数^[3], 如表 1 所示, 用来验证智能优化算法的寻优性能. 每个函数中解的维数均设置为 $D = 30$; 解的取值范围设置为 F_{n6} : $x_i \in [-100, 100]$, F_{n9} : $x_i \in [-500, 500]$, 其他 8 个函数均为 $x_i \in [-30, 30]$.

表 1 典型寻优函数

Function name	Equations
Sphere	$F_{n1} = \sum_{i=1}^D x_i^2$
Quadric	$F_{n2} = \left(\sum_{i=1}^D x_i \right)^2$
Griewank	$F_{n3} = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Ackley	$F_{n4} = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Rosenbrock	$F_{n5} = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
Rastrigin	$F_{n6} = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Schwefel's 1.2	$F_{n7} = \left(\sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2 \right) (1 + 0.4 N(0, 1))$
Griewank with noise	$F_{n8} = \sum_{i=1}^D \frac{(x_i - N(0, 1))^2}{4000}$
Schwefel	$F_{n9} = 418.9829D - \sum_{i=1}^D x_i \sin(x_i ^{1/2})$
Schwefel's 2.22	$F_{n10} = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $

为了便于对比 DEPSO 算法性能, 本文采用 4 种算法进行仿真实验: 1) DE 算法^[3], 其变异策略采用“DE/rand/2”, 算法参数取值为 $F = 0.5$, $C_R = 0.3$; 2) PSO 算法^[11]的参数取值为 $w = 0.729$, $c_1 = 1.495$, $c_2 = 1.495$; 3) SFLA 算法^[8]的参数设置: 模因组数为 5, 模因进化代数为 8, $C = 1.2$; 4) 本文提出的 DEPSO 算法参数的设置与上述相同. 在仿真实验中不同算法采用相同的种群规模, 即 $N_P = 40$; 算法迭代代数 $I_{itermax} = 2000$ 次; 每种算法对每个典型函数寻优计算 50 次, 通过求平均值以获得随机算法的平均寻优结果.

对于 DEPSO 算法, 子种群数设置为 $N_k = 5$. 本文仿真了 $I_{iter} = 10$ 时 ($k = 1, \dots, 4$) 4 种不同组合情况, 获得在 $k = 4$ (即 5 个子种群随机选择 4 个子种群采用 DE 算法迭代计算, 余下 1 个子种群采用 PSO 算法迭代计算) 时, 算法寻优精度最佳. 并且仿真了 $k = 4$ 时, $I_{iter} = 100, 80, 50, 40, 20, 10, 5$ 的 7 种取值的优化结果, 发现随着 $I_{iter} = 100$ 下降到 $I_{iter} = 10$ 的过程, 算法寻优精度呈现变好趋势. 然而, 当 $I_{iter} = 5$ 时算法寻优精度变差. 因此, 在 DEPSO 算法对比仿真实验中 I_{iter} 和 k 分别设置为 $I_{iter} = 10, k = 4$.

表 2 列出了 DE 算法、PSO 算法、SFLA 算法和 DEPSO 算法在 10 个典型函数上进行 50 次寻优计算的平均最优值和均方差. 图 1 显示了 4 种算法在 10 个典型函数上寻优过程平均最优值随代数的变化情况 (为了便于曲线显示, 将平均最优值加 0.001 后取 $\log 10$).

表 2 典型函数寻优结果

函数	性能	PSO	DE	SFLA	DEPSO
F_{n1}	均值	1.249 4 e+003	1.766 7 e+002	1.195 1 e+002	1.200 8 e-009
	方差	2.578 7 e+002	7.343 8 e+001	4.127 0 e+001	7.454 8 e-010
F_{n2}	均值	1.249 4 e+000	0.000 0 e+000	0.000 0 e+000	4.352 8 e-007
	方差	1.905 9 e-008	0.000 0 e+000	0.000 0 e+000	8.203 4 e-007
F_{n3}	均值	1.292 8 e+000	9.859 0 e-001	1.001 1 e+000	8.401 1 e-004
	方差	6.050 0 e-002	9.260 0 e-002	3.590 0 e-002	1.500 0 e-003
F_{n4}	均值	1.671 1 e+001	9.158 9 e+000	7.819 0 e-000	8.648 0 e-004
	方差	7.478 0 e-001	1.473 0 e+000	1.127 9 e-000	7.059 3 e-006
F_{n5}	均值	1.325 5 e+007	2.976 4 e+005	1.286 6 e+005	2.667 8 e+001
	方差	6.235 6 e+006	1.989 5 e+005	7.429 7 e+004	9.377 2 e+000
F_{n6}	均值	1.613 7 e+003	3.677 0 e+002	8.155 5 e+002	1.464 4 e+002
	方差	3.350 5 e+002	7.399 9 e+001	1.228 5 e+001	1.223 5 e+001
F_{n7}	均值	3.705 4 e+005	2.398 1 e+004	7.784 2 e+002	2.331 9 e+002
	方差	9.582 6 e+004	9.379 5 e+003	3.792 1 e+002	1.405 2 e+001
F_{n8}	均值	3.119 0 e-001	3.840 0 e-002	1.460 9 e-003	8.948 9 e-004
	方差	6.670 0 e-002	1.340 0 e-002	5.251 6 e-002	1.361 6 e-004
F_{n9}	均值	7.451 3 e+003	6.845 6 e+003	7.509 7 e+003	4.497 4 e+003
	方差	6.206 1 e+002	1.446 5 e+003	6.131 6 e+002	3.677 1 e+002
F_{n10}	均值	2.036 6 e+002	5.586 4 e+001	8.823 8 e+002	1.027 6 e-004
	方差	2.349 7 e+001	1.232 2 e+001	4.006 8 e+001	3.792 1 e-005

从表 2 和图 1 的仿真结果可知, 对 F_{n2} Quadric 函数进行寻优实验, 本文仿真的 4 种算法均可以达到良好的寻优效果. 其他 9 个典型函数的仿真结果可总结为两点: 1) 在算法的寻优精度上, DEPSO 算法对其他 9 个典型函数的寻优结果较 DE、PSO 和 SFLA 算法均有较大的提高, 均获得了更优秀的解. 2) 在算法的跳出极值能力上, PSO 算法很快就陷入极小值点,

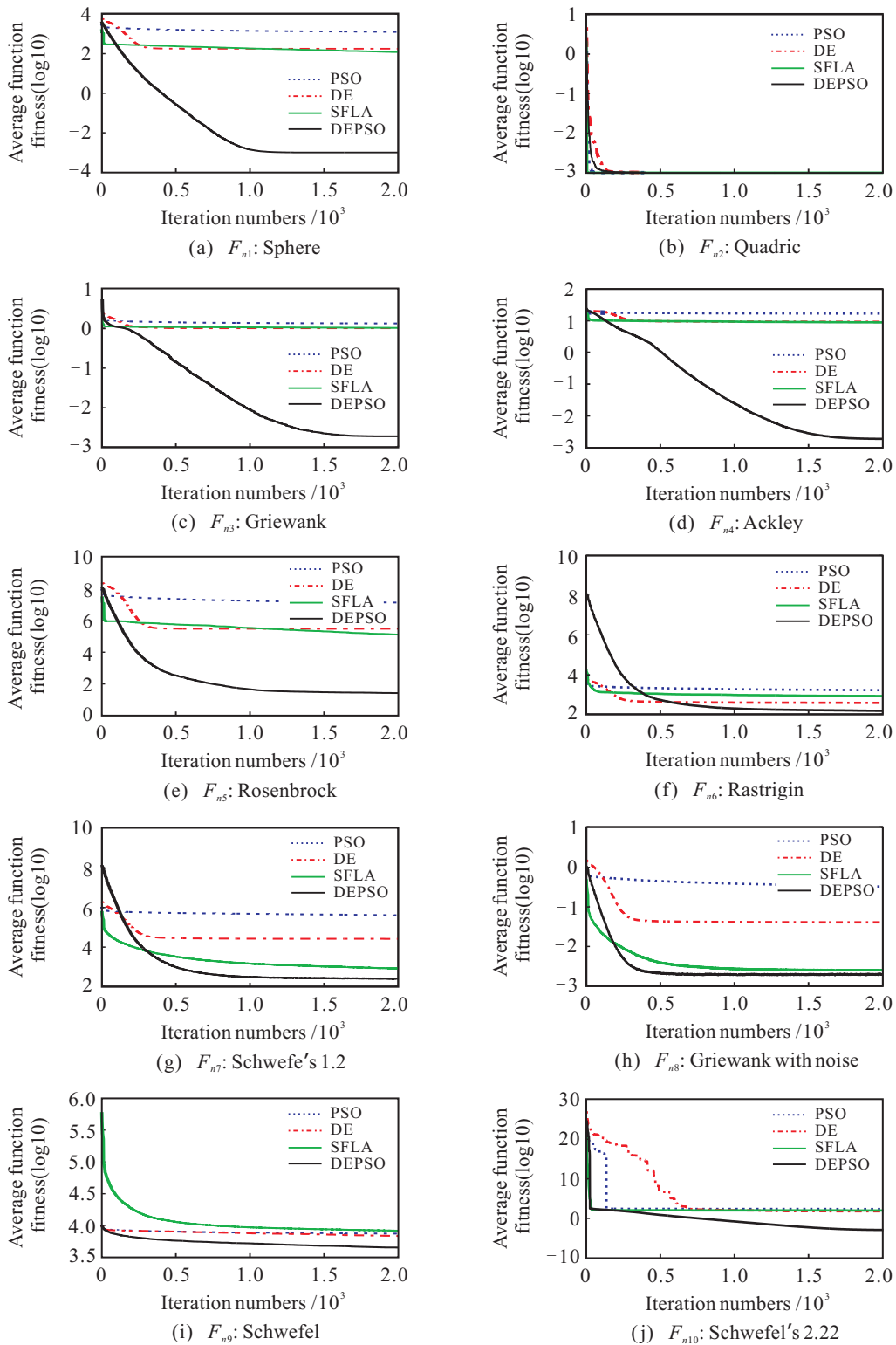


图 1 典型函数仿真曲线

在算法寻优期间所起作用的时域非常短; DE 和 SFLA 算法在寻优的初期有一定的搜索能力, 跳出局部极小值的能力比 PSO 算法强, 可以收敛到更精确的解; DEPSO 算法在迭代过程能不断地跳出局部极小值, 寻找到更优秀的解, 具有很好的搜索能力. 总之, 本文提出的 DEPSO 算法可以平衡智能优化算法的寻优精度与收敛速度两要素, 相比 DE、PSO 和 SFLA 算法, 在寻优性能上有明显的提高.

2 混合智能极限学习机

2.1 极限学习机 (ELM)

极限学习机是近几年提出的一种新的单隐层前向神经网络的学习方法^[2]. 这种学习方法在保证神经网络具有良好泛化性能的同时, 极大地提高了前向神经网络的学习速度.

在 ELM 中, 对于 N 个不同的学习样本 $(I_j, O_j)_{j=1}^N$

(输入向量 $I_j \in R^d$, 目标向量 $O_j \in R^m$), 具有 \tilde{N} 个隐层节点的 SLFNs 模型可表示为

$$\tilde{O}_j = \sum_{i=1}^{\tilde{N}} \beta_i g(a_i, b_i, I_j), \quad j = 1, 2, \dots, N. \quad (5)$$

其中: $a_i \in R^d$ 是输入节点与第 i 个隐层节点的连接权值向量, $b_i \in R$ 是第 i 个隐层节点的阈值, $\beta_i \in R^m$ 是第 i 个隐层节点与输出节点的连接权值向量, $g(a_i, b_i, I_j)$ 是第 i 个隐层节点关于学习样本 I_j 的输出. 式 (5) 可以简记为

$$H\beta = \tilde{O}. \quad (6)$$

其中

$$H_{N \times \tilde{N}} = \begin{bmatrix} g(a_1, b_1, I_1) & \cdots & g(a_{\tilde{N}}, b_{\tilde{N}}, I_1) \\ \vdots & \ddots & \vdots \\ g(a_1, b_1, I_N) & \cdots & g(a_{\tilde{N}}, b_{\tilde{N}}, I_N) \end{bmatrix},$$

$$\beta_{\tilde{N} \times m} = [\beta_1^T \cdots \beta_{\tilde{N}}^T]^T,$$

$$\tilde{O}_{N \times m} = [\tilde{O}_1^T \cdots \tilde{O}_N^T]^T,$$

H 称为神经网络的隐层节点输出矩阵.

ELM 算法首先根据先验知识设定网络隐层节点数; 然后随机设定网络隐层节点参数, 即输入节点与隐层节点的连接权值及隐层节点的阈值, 将网络输入输出关系变成一个线性方程组形式 $H\beta = O$; 最后通过如下线性方程组求解获得网络输出连接权值:

$$\hat{\beta} = H^\dagger O, \quad (7)$$

其中 H^\dagger 为隐层输出矩阵 H 的 Moore-Penrose (MP) 广义逆.

2.2 DEPSO-ELM 算法的仿真实验

ELM 是一种非常简单且快速的神经网络学习方法, 然而由于随机设定网络隐层节点参数, 常常导致网络中某些隐层节点的输出对网络的输出的影响非常小^[4]. 为了提高 ELM 性能, 本文将 DEPSO 算法用于优化 SLFNs 的隐层节点参数, 采用 ELM 算法求得 SLFNs 的输出权值, 获得一种针对 SLFNs 的新学习方法, 称为 DEPSO-ELM 算法. 该算法可以应用在分类、函数拟合、回归计算等方面. 本文通过仿真实验测试 DEPSO-ELM 算法在回归计算上的能力. 实验中采用 6 个典型真实数据集, 分别是: Servo、Wisconsin Breast Cancer (简称 Cancer)、Autompg、Auto Machine CPU (简称 CPU)、Price、Boston Housing (简称 Housing). 这些典型真实数据集是从加州大学厄文分校的机器学习库中获得的 (<http://www.ics.uci.edu/mllearn/MLRepository.html>). 仿真过程中将每种真实数据集分为两部分: 训练数据集和测试数据集, 如表 3 所示.

DEPSO-ELM 算法应用在上述 6 个真实数据集的回归计算的步骤如下.

表 3 典型真实数据集

数据集名称	训练数据	测试数据
Servo	80	87
Cancer	100	94
Autompg	220	178
CPU	100	109
Price	80	79
Housing	250	256

Step 1: 选择一种真实数据集, 将 d 维的输入数据分别标准化到 $[-1, 1]$, 将 m 维的目标数据分别标准化到 $[0, 1]$. 根据表 3 将真实数据集随机分为两部分: 训练数据集和测试数据集.

Step 2: 初始化 DEPSO 算法参数: N_P 、 N_k 、 k 、 I_{itermax} 、 I_{iter} 、 $C_n = I_{\text{itermax}}/I_{\text{iter}}$ 、 F 、 C_R 、 w 、 c_1 、 c_2 等.

Step 3: 初始化 DEPSO 算法的种群 $P = \{X_j\}_{j=1}^{N_P}$, 其中 $X_j \in R^{d \cdot \tilde{N} + \tilde{N}}$ 代表种群中第 j 个 SLFNs 网络的隐层节点参数, SLFNs 的输入节点与隐层节点的连接权值取值范围 $\{X_{ij}\}_{i=1}^{d \cdot \tilde{N}} \in [-1, 1]$, 隐层节点的阈值取值范围 $\{X_{ij}\}_{i=d \cdot \tilde{N} + 1}^{d \cdot \tilde{N} + \tilde{N}} \in [0, 1]$.

Step 4: 设定种群 $P = \{X_j\}_{j=1}^{N_P}$ 的适应度值 $J = \{J_j\}_{j=1}^{N_P}$ 的计算过程如下: 首先以训练数据集为基础, 由 X_j 确定 SLFNs 中隐层节点参数, 应用 ELM 算法计算 SLFNs 隐层节点到输出节点连接权值矩阵, 即根据式 (7) 计算 $\hat{\beta}_j$; 然后根据式 (5) 计算 SLFNs 的输出 \tilde{O}_j ; 最后依据下式计算 X_j 的适应度值:

$$J_j = \sqrt{\frac{\sum_{i=1}^N \|\tilde{O}_{j,i} - O_{j,i}\|}{m \times N}}. \quad (8)$$

由式 (8) 可知, 具有越小的适应度值, 该个体越优秀.

Step 5: 将种群 P 随机平均分割成 N_k 个子种群.

Step 6: 随机选择 k ($1 < k < N_k$) 个子种群, 分别应用 DE 算法迭代计算 I_{iter} 代; 剩下 $N_k - k$ 个子种群, 分别应用 PSO 算法迭代计算 I_{iter} 代.

Step 7: 混合 N_k 个子种群获得新种群 P ; 判断子种群混合次数是否达到 C_n 次, 如果否, 则转 Step 5, 如果是, 则转 Step 8.

Step 8: 算法停止, 输出最优解 X_{best} .

Step 9: 以测试数据集为基础, 首先由 X_{best} 确定 SLFNs 中隐层节点参数; 然后根据式 (7) 计算出 $\hat{\beta}_j$, 由 $H\hat{\beta} = \tilde{O}$ 计算 SLFNs 的输出 \tilde{O} ; 最后根据式 (8) 获得该真实数据集回归计算的结果.

在本节 DEPSO-ELM 算法的回归计算仿真中, 部分参数设定为: $I_{\text{iter}} = 10$, $I_{\text{itermax}} = 1000$, $\tilde{N} = 20$, 其他参数的取值与 1.4 节相同. 表 4 列出了 DEPSO-ELM 算法在 6 个典型真实数据集上进行 50 次回归计算的平均最优值和均方差. 通过与文献 [4] 中的两种算法

DE-ELM 和 SaE-ELM 的计算结果对比可知, 本文所提出的 DEPSO-ELM 算法比其他两种算法的计算结果更精确。

表 4 典型数据集回归计算结果

数据集	性能	DE-ELM	SaE-ELM	DEPSO-ELM
Servo	均值	0.105 1	0.088 4	0.073 5
	方差	0.022 6	0.016 1	0.010 9
Cancer	均值	0.266 3	0.256 1	0.225 4
	方差	0.020 8	0.021 7	0.011 6
Autompg	均值	0.069 1	0.063 1	0.059 8
	方差	0.012 9	0.010 0	0.004 4
CPU	均值	0.052 8	0.039 6	0.003 5
	方差	0.023 8	0.008 3	0.000 4
Price	均值	0.049 4	0.043 7	0.043 2
	方差	0.009 4	0.009 4	0.006 8
Housing	均值	0.096 5	0.090 4	0.076 7
	方差	0.013 7	0.013 2	0.008 7

3 基于 DEPSO-ELM 的数控机床热误差预测研究

精密加工技术已成为现代机械制造中重要的组成部分。在机床的各种误差源中, 热误差占机床总误差的 70% 左右^[17]。应用设计和制造技术改进机床自身结构可减小误差, 但是运用误差补偿技术来提高机床加工精度却是一种有效且经济的途径。要对机床进行有效的热误差补偿必须解决两个前提问题, 即热误差的检测及热误差预测模型的建立。

3.1 机床关键点温度及热误差的检测

将 DEPSO-ELM 预测模型应用于图 2 所示的数控机床工作的热误差补偿。

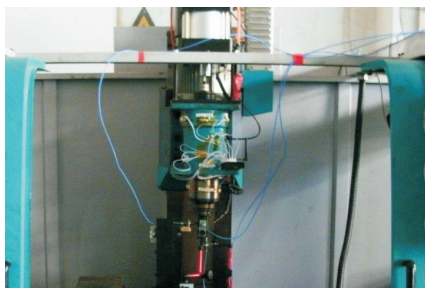
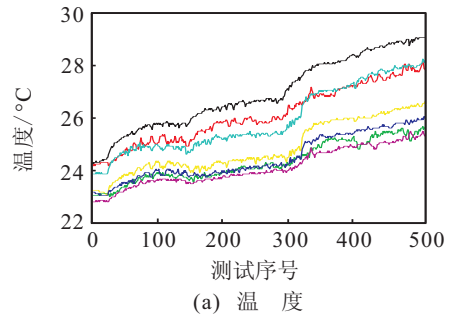
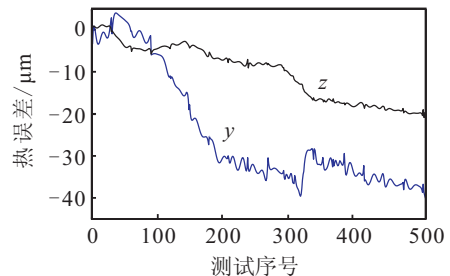


图 2 数据检测实验图

为了描述热误差相关的机床温度场分布, 选择机床主轴上 4 个温度传感器测量值, 滚珠丝杠上的 2 个温度传感器测量值, 冷却箱体外壳的 1 个温度传感器测量值共 7 个温度值。采用两个位移传感器测量机床刀具的 y 向和 z 向的热误差。本文完成了机床运行在 1000、1300、1600、1800 不同转速下机床空载的实验数据测量, 通过数据处理获得了 512 组实验数据。图 3 显示了机床关键点温度值及热误差变化情况。



(a) 温度

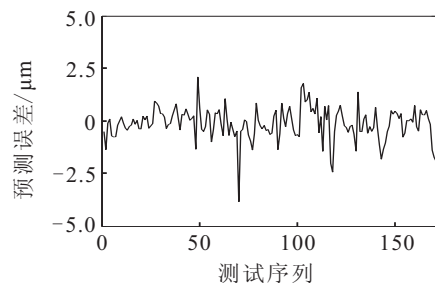


(b) 热误差

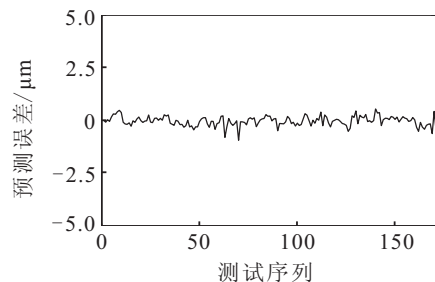
图 3 机床关键点温度及热误差变化图

3.2 基于 DEPSO-ELM 的机床热误差预测分析

基于 ELM 的智能模型具有很强的学习能力和泛化能力, 可以应用于实际生产过程中对关键变量的预报上^[18]。本文研究了 DEPSO-ELM 算法, 提高了 SLFNs 的性能, 将其应用于数控机床热误差补偿建模中, 将提高机床的加工精度。本文选择 7 个关键点温度数据、转速和 y 、 z 向热误差测量数据, 共 9 个变量作为 DEPSO-ELM 的输入数据, 将当前测量点 1min 后所测量的 y 、 z 向热误差数据作为 DEPSO-ELM 的目标数据。在 512 组的实验数据中随机选择 341 组作为训练数据, 获得数控机床热误差输入输出关系预测模型, 剩下的 171 组作为测试数据。DEPSO-ELM 算法运用在机床热误差建模实验的参数设置与 2.2 节相同。



(a) y 方向热误差



(b) z 方向热误差

图 4 y 、 z 方向热误差的预测误差曲线

图4显示了 y 、 z 方向的实测值与预测模型的预测值之间的偏差曲线. 在171个预测点中, y 向预测与实测的偏差最大为 $3.8232\ \mu\text{m}$,绝对平均偏差为 $0.5589\ \mu\text{m}$. z 向预测与实测的偏差最大为 $1.9081\ \mu\text{m}$,绝对平均偏差为 $0.3795\ \mu\text{m}$.实验结果表明,将DEPSO-ELM算法应用于数控机床热误差建模预测可以获得良好的预测精度.

4 结 论

本文研究了基于群体演化的两种具有代表性的优化算法:差分进化算法和粒子群优化算法,提出了一种新的混合智能优化算法,通过10个典型函数的寻优仿真实验,表明了所提出的算法比DE、PSO、SFLA算法具有更好的计算速度与精度.然后,将所提出的DEPSO算法应用于优化SLFNs的隐层节点参数,采用ELM算法求得SLFNs的输出权值,并将所提出的DEPSO-ELM算法应用于6种来自真实世界的典型数据集上进行回归计算,获得了比DE-ELM、SaE-ELM两种算法更好的计算结果.将DEPSO-ELM算法应用于数控机床热误差的建模预测上也获得了良好的预测效果.

参考文献(References)

- [1] 刘德荣,李宏亮,王鼎.基于数据的自学习优化控制:研究进展与展望[J].自动化学报,2013,39(11):1858-1870. (Liu D R, Li H L, Wang D. Data-based self-learning optimal control: Research progress and prospects[J]. Acta Automatica Sinica, 2013, 39(11): 1858-1870.)
- [2] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.
- [3] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 398-417.
- [4] Cao J, Lin Z, Huang G B. Self-adaptive evolutionary extreme learning machine[J]. Neural Processing Letters, 2012, 36(3): 285-305.
- [5] Zhu Q Y, Qin A K, Suganthan P N, et al. Evolutionary extreme learning machine[J]. Pattern Recognition, 2005, 38(10): 1759-1763.
- [6] Han F, Yao H F, Ling Q H. An improved evolutionary extreme learning machine based on particle swarm optimization[J]. Neurocomputing, 2013, 116: 87-93.
- [7] Miche Y, Sorjamaa A, Bas P, et al. OP-ELM: Optimally pruned extreme learning machine[J]. IEEE Trans on Neural Networks, 2010, 21(1): 158-162.
- [8] 崔文华,刘晓冰,王伟,等.混合蛙跳算法研究综述[J].控制与决策,2012,27(4):481-486. (Cui W H, Liu X B, Wang W, et al. Survey on shuffled frog leaping algorithm[J]. Control and Decision, 2012, 27(4): 481-486.)
- [9] Storn R, Price K. Differential evolution — A simple and efficient heuristics for global optimization over continuous spaces[J]. J of Global Optimization, 1997, 11(4): 341-359.
- [10] 辛斌,陈杰.粒子群优化与差分进化混合算法的综述与分类[J].系统科学与数学,2012,31(9):1130-1150. (Xin B, Chen J. Particle swarm optimization and differential evolution: State of the art[J]. J of Systems Science and Mathematical Sciences, 2012, 31(9): 1130-1150.)
- [11] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [12] Kim P, Lee J. An integrated method of particle swarm optimization and differential evolution[J]. J of Mechanical Science and Technology, 2009, 23(2): 426-434.
- [13] Hao Z F, Guo G H, Huang H. A particle swarm optimization algorithm with differential evolution[C]. Proc of the 6th Int Conf on Machine Learning and Cybernetics. Hong Kong, 2007: 1031-1035.
- [14] Xin B, Chen J, Peng Z H, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization[J]. Science China Information Sciences, 2010, 53(5): 980-989.
- [15] Zhang C S, Ning J X, Lu S, et al. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization[J]. Operations Research Letters, 2009, 37(2): 117-122.
- [16] Kannan S, Slochanal S M R, Subbaraj P, et al. Application of particle swarm optimization technique and its variants to generation expansion planning[J]. Electric Power Systems Research, 2004, 70(3): 203-210.
- [17] 李永祥.数控机床热误差建模新方法及其应用研究[D].上海:上海交通大学机械与动力工程学院,2007. (Li Y X. New methods of thermal error modeling for NC machine tools and their researchment and application[D]. Shanghai: School of Mechanical & Power Engineering, Shanghai Jiaotong University, 2007.)
- [18] 田慧欣,毛志忠.基于Bagging的多模型钢水温度预报[J].控制与决策,2009,24(5):687-691. (Tian H X, Mao Z Z. Multi-model prediction of molten steel temperature based on bagging[J]. Control and Decision, 2009, 24(5): 687-691.)

(责任编辑:李君玲)