DOI:10.19651/j.cnki.emt.2313377

多策略SMA-BP神经网络的空气质量指数预测*

文昌俊[1,2]　陈洋洋[1,2]　何永豪[1,2]　陈　凡[1,2]
(1.湖北工业大学机械工程学院 武汉 430068; 2.湖北省现代制造质量工程重点实验室 武汉 430068)

摘　要：针对BP神经网络预测精度不佳、预测结果不稳定的问题，提出改进黏菌算法(ISMA)优化BP神经网络的预测模型，引入Tent混沌映射克服初始种群分布不均的缺点，针对黏菌算法位置更新的随机性和后期容易陷入局部最优等问题引入领导者策略和莱维飞行策略，利用自适应反向学习策略扩大搜索空间并用23组基准函数加以测试。随后利用ISMA算法优化BP网络模型的初始权值和阈值，构建ISMA-BP空气质量指数预测模型，最后将收集到的779组空气质量指数数据代入预测模型中进行测试分析，实验结果表明，与BP神经网络模型、GWO-BP、SMA-BP模型相比，ISMA-BP模型对AQI的预测具有更高的精度，其预测的均方误差为3.840 2，平均绝对误差分别为1.507 8。

关键词：黏菌算法；Tent混沌映射；反向学习策略；BP神经网络；灰色关联；度空气质量预测
中图分类号：TP393　　**文献标识码**：A　　**国家标准学科分类代码**：520.2060

Air quality index prediction by multi-strategy SMA-BP neural network

Wen Changjun[1,2]　Chen Yangyang[1,2]　He Yonghao[1,2]　Chen Fan[1,2]
(1. School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China;
2. Hubei Province Key Laboratory of Modern Manufacturing Quality Engineering, Wuhan 430068, China)

Abstract: Aiming at the problems of poor prediction accuracy and unstable prediction results of BP neural network, an improved slime mold algorithm (ISMA) is proposed to optimize the prediction model of BP neural network, and Tent chaotic mapping is introduced to overcome the shortcomings of uneven initial population distribution. The leader strategy and Levy flight strategy are introduced to solve the randomness of the position update and the problem of falling into local optimality. The adaptive reverse learning strategy is used to expand the search space and 23 groups of benchmark functions are tested. Then the ISMA algorithm was used to optimize the initial weights and thresholds of the BP network model, and the ISMA-BP Air quality index prediction model was constructed. At last, 779 sets of AQI data were collected and put into the prediction model for testing and analysis. The experimental results showed that, Compared with BP neural network model, GWO-BP model and SMA-BP model, ISMA-BP model has higher accuracy in predicting AQI. The mean square error of ISMA-BP model is 3.840 2, and the mean absolute error is 1.507 8 respectively.

Keywords: slime mold algorithm; Tent chaotic mapping; reverse learning strategy; BP neural network; grey correlation degree; AQI prediction

0　引　　言

随着我国工业化以及城市化进程的不断加快，导致了越来越严重的环境污染问题。空气质量下降已经成为了制约我国城市发展和影响城市居民健康的重要环境因素[1]。空气质量指数(air quality index，AQI)是国际上普遍采用的定量描述空气质量状况的重要指标[2]，通过其大小不仅可以反映空气污染的水平，还可以为安全生产和空气质量预警提供可靠保障，因此，科学准确的AQI预测模型对城市环境治理和抑制空气污染有着重要意义[3]。

传统的空气质量预测模型包括数理统计模型、灰色预测模型等[4]。郑瑶等[5]使用灰色关联度、多项式回归残差计算并修正GM(1,1)模型，利用粒子群算法优化GM(1,1)模型的预测关联度，构建灰色关联组合模型对PM10、PM2.5的浓度进行预测；史凯赫等[6]对现有的灰色预测模型进行改进，使用Hausdoff累加方式构建新灰色预测多变

收稿日期：2023-04-15
* 基金项目：国家自然科学基金(51875180)项目资助

量模型 FHGM(1,N)用于石家庄空气质量预测,结果表明模型具有 jiaohao1 的预测性能和泛化性;但是实际影响空气质量指数的各因素间有着较强的非线性关系,利用传统的统计预测模型预测得到的 AQI 精度较低。由于 BP 神经网络较强的非线性映射能力,使其被广泛应用于数据预测方面,Zhao 等[7]利用 BP 神经网络模型非线性映射特点,来预测空气质量的演变,取得了较为准确的结果。Su 等[8]利用粒子群算法优化 BP 神经网络的初始权值和阈值,并对空气质量进行了预测。但由于粒子群算法易陷入局部最优,模型的预测精度不高。

综上所述,单一 BP 神经网络模型 AQI 预测效果较差,即使利用算法优化了 BP 神经网络,但算法后期容易陷入局部最优,影响预测效果,故本文利用具有任意逼近非线性函数能力的 BP 神经网络结合具有优良的全局寻优能力的黏菌算法(slime mould algorithm,SMA),同时引入 Tent 混沌映射初始化种群,克服黏菌算法初始种群初始化不均匀的问题;引入领导者指导策略,避免黏菌位置更新过于随机,提高黏菌算法的全局寻优能力;迭代后期,引入 Levy 飞行机制,增加黏菌算法跳出局部极值的概率;再引入动态反向学习策略,扩大黏菌搜索空间,并用 23 组基准函数验证 ISMA 算法的有效性。最后利用改进黏菌算法建立预测模型(ISMA-BP)对空气质量指数进行预测,并利用武汉市空气质量数据进行仿真验证,同时与 BP、WOA-BP、SMA-BP 网络预测模型的性能进行比较。

1　BP 神经网络

基本的 BP 神经网络模型如图 1 所示。由于 BP 神经网络(back propagation neural network)具有良好的非线性映射能力以及数据处理能力,在缺少数学模型时,也能进行较精准预测,因此在数据预测领域,具有较为广泛的应用[9]。

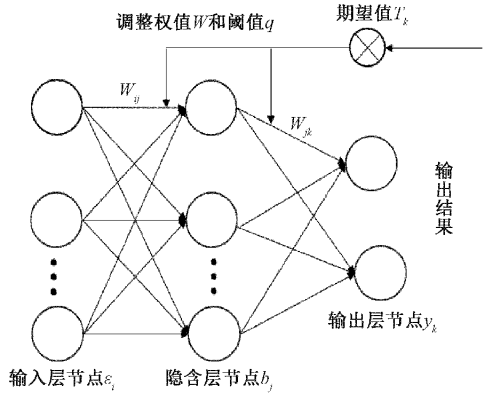

图 1　BP 神经网络拓扑结构

输出层节点的输出 y_k 为:

$$y_k = f_2\left(\sum_{j=1}^{l} w_{jk} f_1\left(\sum_{i=1}^{s} w_{ij}\varepsilon_i - q_j\right) - q_k\right) \tag{1}$$

式中:f_2 为输出层的激活函数,f_1 为隐含层的激活函数,w_{ij} 为第 i 个输入层节点和第 j 个隐含层节点之间的权值,w_{jk} 为第 j 个隐含层节点和第 k 个输出层节点之间的权值,q_j 和 q_k 分别为隐含层和输出层的阈值,s 和 l 分别为输入层节点个数和隐含层节点个数。

误差函数 e 的计算公式为:

$$e = \sum_{k=1}^{m}(T_k - y_k)^2 \tag{2}$$

式中:T_k 是输出层节点 k 的期望输出值,y_k 是实际输出值,m 为输出层节点个数。

BP 神经网络的初始权值与阈值都是随机产生的,基于梯度下降思想进行不断调整,因此 BP 神经网络性能的好坏受到初始权值的影响很大,这也导致了 BP 神经网络预测精度的不稳定性。针对 BP 神经网络存在的不足,使用改进黏菌算法来优化神经网络的初始权值和阈值,从而提高 BP 神经网络的预测精度。

2　改进黏菌算法

2.1　标准黏菌算法

黏菌优化算法是 Li 等[10]于 2020 年提出的一种全新的群智能算法,相比于传统优化算法,其拥有更优秀的搜索性能。SMA 算法利用特殊权值 W 来模拟生物振荡器的正负反馈过程,进而模拟觅食形态。黏菌觅食过程中的位置更新公式为:

$$X(t+1) = \begin{cases} rand.(UB - LB) + LB, & r < z \\ X_b(t) + v_b(W.X_A(t) - X_B(t)), & z \leqslant r < p \\ v_c.X(t), & r \geqslant p \end{cases} \tag{3}$$

式中:$X_b(t)$为当前发现食物浓度最高的位置,也叫最优解位置,v_b 的取值范围为$[-a,a]$;参数 v_c 从 1 到 0 线性递减;r 为$[0,1]$的随机数;UB、LB 分别为搜索区间的上界和下界;$X_A(t)$和 $X_B(t)$表示从第 t 代黏菌中随机选取的两个个体;p 为选择黏菌位置更新开关,可以用以下公式表示:

$$p = \tanh|S(i) - F_D| \tag{4}$$

式中:$S(i)$为排序后第 i 个黏菌个体的适应度;F_D 为所有迭代中最佳适应度。参数 a 的计算公式为:

$$a = \operatorname{arctanh}(-(t/T) + 1) \tag{5}$$

其中,t 为当前迭代的次数,T 表示最大迭代次数。

式(3)中,W 表示权重系数,当区域附近浓度较高时,该区域的权重为增大,否则就会寻找其他区域搜索,其满足:

$$W(i) = \begin{cases} 1 + r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & i < N/2 \\ 1 - r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & i \geqslant N/2 \end{cases} \tag{6}$$

式中:bF 和 wF 分别表示当前迭代中的最优适应度和最差适应度;$W(i)$是个体适应度按升序进行排序后的个体权重

系数，r 表示[0,1]间的随机数，N 表示黏菌种群个数。

2.2　改进黏菌算法

1)Tent 混沌映射初始种群

对于大多数算法而言，初始种群分布得越均匀，种群的多样性越丰富，获得最优解的机会也越大[11]。在传统黏菌算法中，初始种群的生成是基于随机策略的，这种随机策略是通过设定范围的上下界确定，无法保证黏菌初始位置的遍历性，使部分个体远离最优解。为了充分利用解的空间信息，增强初始解搜索空间的勘测能力，利用混沌映射方法初始化种群，增强种群多样性。

混沌映射具有非线性、随机性和遍历性的特点，目前常用的混沌映射包括 Logistic 映射[12]、Tent 映射[13]等，图 2 和 3 分别为 Logistic 和 Tent 混沌序列直方图。

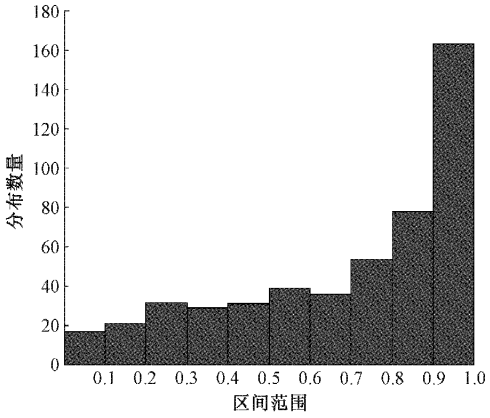

图 2　Logistic 映射

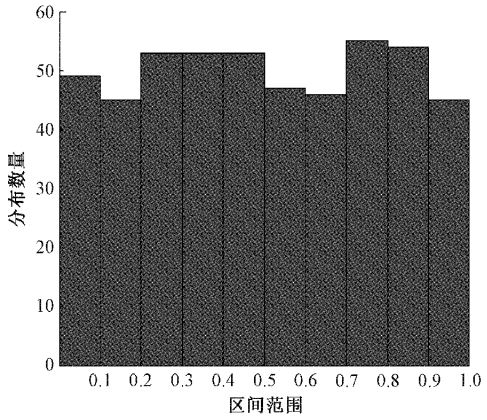

图 3　Tent 映射

图中可以看出 Logsitic 映射在[0.9,1]区间内的分布数量远远超过其他区间内的分布数量，相比之下 Tent 映射较为均匀，Tent 混沌映射的表达式如式(7)所示。

$$x_{t+1}=\begin{cases}2x_t, & 0<x_t\leqslant\dfrac{1}{2}\\ 2(1-x_t), & \dfrac{1}{2}<x_t\leqslant 1\end{cases}\tag{7}$$

式中：x_{t+1} 表示第 $t+1$ 次迭代时的混沌数值；x_t 为第 t 次迭代时的混沌数值；式中生成的 Tent 映射序列存在小周期和不稳定周期点。为了避免这种情况，当 $x_t=0,0.25,0.5,0.75$ 或 $x_t=x_{t+k}$，$k=\{1,2,3\}$（当序列陷入不动点或陷入周期时），引入下式使 Tent 序列重新进入混沌状态。

$$x_{t+1}=\begin{cases}2(x_t+0.1+rand()), & 0<x_t\leqslant\dfrac{1}{2}\\ 2(1-(x_t+0.1+rand())), & \dfrac{1}{2}<x_t\leqslant 1\end{cases}\tag{8}$$

2)领导者指导策略

传统 SMA 位置更新公式依赖于最优个体位置和种群中 2 个随机个体位置，导致搜索黏菌的移动过于随机，当两个随机解远离最优解区域，就需要更长的时间才能达到最优解。GWO 算法利用 3 个迄今为止最好的候选解（α、β 和 γ）来更新狼的搜索位置，受此启发，将迄今为止最好的 3 个候选解作为领导者来指导黏菌位置更新。此时的黏菌位置更新方式为：

$$X(t+1)=X_b(t)+v_b(W.X_{L1}-X_B(t))+v_b(W.X_{L2}-X_C(t))z<r<p\tag{9}$$

式中：X_{L1} 和 X_{L2} 分别为适应度第二优和第三优的个体位置；$X_B(t)$ 和 $X_C(t)$ 为适应度排序前 1/3 的个体中随机抽取的 2 个个体的位置。

3)莱维飞行策略

在标准 SMA 算法中，随着迭代次数的增加，最优个体易陷入局部极值，导致黏菌收敛至同一位置并停滞不前，影响最终精度。基于上述不足，引入 Levy 扰动策略对 SMA 算法后期迭代进行优化。

Levy 飞行[14]是一种步长具有概率分布的随机游走，它的运动特征是搜索范围远近交替地随机飞行，莱维飞行步长 L 的计算方式如下：

$$L=u/|v|^{1/\beta}\tag{10}$$

u 和 v 服从均值为 0、方差分别为 σ^u、σ^v 的正态分布：

$$\begin{cases}u\sim N(0,\sigma_u^2)\\ v\sim N(0,\sigma_v^2)\end{cases}\tag{11}$$

$$\begin{cases}\sigma_u=\left[\dfrac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right]\\ \sigma_v=1\end{cases}\tag{12}$$

其中，$\beta=1.5$。

将莱维飞行引入黏菌算法迭代后期的位置更新公式中，以增加黏菌个体在搜索过程中的活跃性，从而提高跳出局部极值的概率，有效改善 SMA 算法在迭代后期陷入局部最优的问题。改进后的黏菌位置更新公式为：

$$X(t+1)=X_b(t)+v_b(W.X_b(t)-X(t))+\rho.L\tag{13}$$

式中：ρ 为步长控制因子；L 为莱维飞行步长。

为了平衡 SMA 算法的探索和勘测能力，优化求解精度，将位置更新过程划分为 2 个阶段。

阶段 1:

$$X(t+1)=\begin{cases}rand.(UB-LB)+LB & r<z\\ X_b(t)+v_b((W.X_{L1}-X_B(t))+\\ W.X_{L2}-X_C(t)) & z,<r<p\\ v_c.X(t) & p<r\end{cases}\tag{14}$$

式中:$t\leqslant 2t_{\max}/3$。

阶段 2:

$$X(t+1)=\begin{cases}rand.(UB-LB)+LB & r<z\\ X_b(t)+v_b(WX_b(t)-X(t))+\rho.L & r\geqslant z\end{cases}\tag{15}$$

式中:$t>2t_{\max}/3$。

4)动态反向学习策略

反向学习的核心是利用当前解计算出反向解,然后同时评估当前解和反向解的适应度,选择适应度更优的个体用于下次的迭代计算。反向解的生成表达式为:

$$\overline{X_{ij}}=UB_{ij}+LB_{ij}-X_{ij}\tag{16}$$

式中:$i=\{1,2,\cdots,N\}$;$j=\{1,2,\cdots,D\}$;X_{ij} 和 $-X_{ij}$ 分别为第 i 个体第 j 维的当前解和反向解;UB_{ij} 和 UB_{ij} 分别表示第 i 个体第 j 维变量的上限和下限。

在传统反向学习中,UB_{ij} 和 UB_{ij} 是一个固定值,可能导致生成的解质量较差。因此引入随自适应的 UB_{ij} 和 UB_{ij},来提高反向解的质量。改进后的反向解生成表达式如下:

$$\begin{cases}X_{ij}(t)=UB_{ij}(t)+LB_{ij}(t)-X_{ij}\\ UB_{ij}(t)=\max(UB_j(t))\\ LB_{ij}(t)=\min(LB_j(t))\end{cases}\tag{17}$$

式中:$UB_{ij}(t)$和$UB_{ij}(t)$分别为第 t 代中第 i 个体 j 维变量的上限和下限,它随着迭代的变化而变化,迭代到后期,种群较为集中,此时反向解的上下边界也逐渐减小。将自适应反向学习引入 SMA 算法迭代中,扩大搜索空间,提高黏菌种群的解质量。

2.3　ISMA 算法测试

使用 23 个经典测试函数对本文你的 ISMA 算法进行验证。其中 f_1-f_7 为多维的单模态,只有一个全局最优解,f_8-f_{13} 为多维的多峰函数,存在多个局部极小值,$f_{14}-f_{23}$ 是固定维的多峰函数,局部极小值数量较少,用于检测算法探索和开发能力之间的平衡性能。

为了更好验证改进黏菌算法(ISMA)的有效性,将 ISMA 与传统黏菌算法(SMA)、麻雀搜索算法(SSA)、鲸鱼算法(WOA)、灰狼算法(GWO)以及人工蜂鸟算法(AHA)进行比较。6 种算法种群规模和最大迭代次数均为 30 和 500 代,算法所涉及的基本参数设置如表所示。

每个算法独立运行 30 次,选取均值(Mean)、标准差(Std)作为评价指标,记录 30 次的运行结果,如表 2 所示。

从表 2 中可以看出,对于单峰函数 $f_1\sim f_4$,ISMA 算法能取到理论最优值且标准差最小;对于函数 f_5,ISMA 的求解精度和标准差都要优于其他算法;对于 f_6 函数,ISMA 的求解精度和标准差仅次于 SSA 算法;对于 f_7 函数,ISMA 的求解精度最高,且标准差最小,超过其他算法;从 $f_1\sim f_7$ 整体来看,ISMA 的收敛精度是均优于 SMA 的,表明对 SMA 的改进策略的有效性。对于多峰函数 $f_8\sim f_{13}$,ISMA 的求解结果均值最优,尤其是对函数 f_{12} 和 f_{13} 的求解上,ISMA 的求解精度和标准差均优于 SMA 算法,表明 ISMA 能多次跳出局部极值;在 f_8、f_9 和 f_{11} 的求解都能达到理论最优值;对于 f_{14},ISMA 的求解标准差要优于其他算法;对于 f_{15} 和 f_{19},ISMA 的求解结果仅次于 AHA;对于 f_{16} 和 f_{18},各算法的求解均值一样,但是 ISMA 的求解结果标准差都要优于其他算法;对于 f_{17},ISMA 的求解标准差仅次于 SSA;对于 f_{21}、f_{22} 和 f_{23},ISMA 的求解结果均值最优,且标准差最小。

表 1　算法基本参数表

算法	参数设置
ISMA	$z=0.03$, $\rho=0.00002$
SMA	$z=0.03$
SSA	$c_1\in[0,1]$, $c_2\in[0,1]$
WOA	$a_1\in[0,2]$, $a_2\in[-2,-1]$,b=1
GWO	$a\in[0,2]$

图 4 给出了部分测试函数平均收敛曲线,对于函数 f_1、f_2 和 f_3,ISMA 的收敛速度都优于其他算法,能够快达到函数的理论最优值;对于单峰函数 f_5 和 f_7,ISMA 的收敛精度最高,且不断跳出局部极值点。对于多峰函数 f_8、f_{10}、f_{14} 和 f_{21},ISMA 都取到了最优值,并且跟其他取到最优值的算法相比,ISMA 的收敛速度更快一些。

2.4　ISMA-BP 网络

ISMA 算法优化 BP 神经网络的本质是利用 ISMA 算法优秀的全局寻优能力来寻找最优的初始权值和阈值,提高 BP 神经网络预测的准确性。其基本流程图如图 5 所示,

主要实施步骤为:

步骤 1)确定神经网络输入层、隐含层和输出层的节点数,以及神经网络输入层与隐含层之间的传递函数、隐含层和输出层之间的传递函数。

步骤 2)根据 BP 神经网络权值和阈值数目确定单个解的维度 D;确定种群的规模 N,最终迭代次数 M,以及其他参数。

步骤 3)计算初始种群的适应度值,将均方误差(MSE)作为 ISMA 算法的适应度函数,确定初始种群中每个个体的适应度值,并确定全局最优解。

$$MSE=\sum_{i=1}^{n}\sum_{j=1}^{m}(y_i(j)-\overline{y_i}(j))^2\tag{18}$$

其中,$y_i(j)$是第 i 个样本网络输出层第 j 个节点的期望输出值,$\overline{y_i}(j)$是第 i 个样本网络输出层第 j 个节点的预测输出值,n 是训练集样本数,m 是输出层节点数。

步骤 4)利用式(18)计算每个个体的适应度值,然后根据当前迭代数 t 选择式(14)、(15)更新黏菌的位置。

表 2　6 种预测模型性能对比

函数	指标	ISMA	SMA	SSA	WOA	GWO	AHA
f_1	Avg	**0.00×10^{+00}**	2.44×10^{-308}	4.38×10^{-08}	4.13×10^{-76}	2.47×10^{-27}	6.16×10^{-151}
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	4.98×10^{07}	2.92×10^{76}	3.75×10^{27}	2.66×10^{152}
f_2	Avg	**0.00×10^{+00}**	5.48×10^{153}	1.26×10^{+00}	1.08×10^{54}	9.36×10^{17}	1.00×10^{71}
	Std	**0.00×10^{+00}**	3.00×10^{152}	1.89×10^{+00}	7.55×10^{54}	6.21×10^{17}	3.72×10^{71}
f_3	Avg	**0.00×10^{+00}**	1.29×10^{304}	1.52×10^{+03}	4.76×10^{+03}	1.02×10^{05}	1.20×10^{127}
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	9.26×10^{+02}	1.91×10^{+04}	3.62×10^{-05}	5.84×10^{-127}
f_4	Avg	**0.00×10^{+00}**	6.10×10^{-142}	1.17×10^{+01}	5.11×10^{+01}	8.01×10^{-07}	5.74×10^{-62}
	Std	**0.00×10^{+00}**	3.34×10^{-141}	4.00×10^{+00}	2.81×10^{+01}	8.73×10^{-07}	2.28×10^{-61}
f_5	Avg	**4.88×10^{-02}**	2.80×10^{+01}	1.50×10^{+03}	2.79×10^{+01}	2.62×10^{+01}	2.67×10^{+01}
	Std	**4.64×10^{+00}**	1.16×10^{101}	1.95×10^{102}	4.31×10^{-01}	6.26×10^{-01}	4.16×10^{-01}
f_6	Avg	1.91×10^{-03}	5.43×10^{-03}	**1.26×10^{-07}**	3.77×10^{-01}	7.17×10^{-01}	4.86×10^{-02}
	Std	3.24×10^{-03}	3.87×10^{-03}	**1.34×10^{-07}**	2.35×10^{-01}	2.85×10^{-01}	8.59×10^{-02}
f_7	Avg	**6.55×10^{-05}**	1.81×10^{-04}	9.34×10^{-02}	4.17×10^{-03}	1.50×10^{-03}	2.66×10^{-04}
	Std	**6.03×10^{-05}**	2.35×10^{-04}	4.90×10^{-02}	4.17×10^{-03}	1.34×10^{-03}	1.74×10^{-04}
f_8	Avg	**−1.26×10^{+04}**	**−1.26×10^{+04}**	−9.51×10^{+03}	−6.44×10^{+03}	−7.13×10^{+03}	−1.11×10^{+04}
	Std	**3.17×10^{-02}**	4.73×10^{-01}	3.54×10^{+02}	1.90×10^{+03}	6.90×10^{+02}	3.75×10^{+02}
f_9	Avg	**0.00×10^{+00}**	**0.00×10^{+00}**	5.50×10^{+01}	1.89×10^{-15}	3.71×10^{+00}	**0.00×10^{+00}**
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	1.73×10^{+01}	1.04×10^{-14}	6.59×10^{+00}	**0.00×10^{+00}**
f_{10}	Avg	**8.88×10^{-16}**	**8.88×10^{-16}**	3.84×10^{+00}	4.09×10^{-15}	1.08×10^{-13}	**8.88×10^{-16}**
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	9.30×10^{-01}	2.70×10^{-15}	1.78×10^{-14}	**0.00×10^{+00}**
f_{11}	Avg	**0.00×10^{+00}**	**0.00×10^{+00}**	1.58×10^{-02}	1.58×10^{-02}	1.67×10^{-03}	**0.00×10^{+00}**
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	1.66×10^{-02}	4.82×10^{-02}	5.19×10^{-03}	**0.00×10^{+00}**
f_{12}	Avg	**1.45×10^{-05}**	3.47×10^{-03}	6.63×10^{100}	5.68×10^{-02}	4.68×10^{-02}	2.31×10^{-04}
	Std	**4.58×10^{-05}**	5.38×10^{03}	4.13×10^{+00}	1.57×10^{02}	2.74×10^{02}	1.25×10^{03}
f_{13}	Avg	**4.14×10^{-04}**	7.93×10^{03}	1.80×10^{+01}	4.74×10^{01}	6.17×10^{01}	1.87×10^{+00}
	Std	**3.12×10^{-04}**	9.12×10^{-03}	1.26×10^{+01}	2.05×10^{-01}	2.15×10^{-01}	5.88×10^{-01}
f_{14}	Avg	**9.98×10^{-01}**	**9.98×10^{-01}**	**9.98×10^{-01}**	**9.98×10^{-01}**	**9.98×10^{-01}**	**9.98×10^{-01}**
	Std	**2.17×10^{-16}**	8.17×10^{-13}	7.87×10^{-01}	3.23×10^{+00}	4.05×10^{+00}	6.05×10^{-01}
f_{15}	Avg	4.30×10^{-04}	6.33×10^{-04}	2.83×10^{-03}	6.64×10^{-04}	6.40×10^{-03}	3.10×10^{-04}
	Std	2.03×10^{-04}	2.72×10^{-04}	6.00×10^{-03}	3.99×10^{-04}	9.30×10^{-03}	1.05×10^{-05}
f_{16}	Avg	**−1.03×10^{+00}**	**−1.03×10^{+00}**	**−1.03×10^{+00}**	**−1.03×10^{+00}**	**−1.03×10^{+00}**	**−1.03×10^{+00}**
	Std	**5.13×10^{-16}**	9.81×10^{-10}	1.91×10^{-14}	3.78×10^{-09}	2.28×10^{-08}	5.76×10^{-16}
f_{17}	Avg	**3.98×10^{-01}**	**3.98×10^{-01}**	**3.98×10^{-01}**	3.98×10^{-01}	**3.98×10^{-01}**	**3.98×10^{-01}**
	Std	2.63×10^{-13}	5.42×10^{-08}	**2.02×10^{-14}**	8.14×10^{-06}	3.92×10^{-06}	1.05×10^{-09}
f_{18}	Avg	**3.00×10^{+00}**	**3.00×10^{+00}**	**3.00×10^{+00}**	**3.00×10^{+00}**	**3.00×10^{+00}**	**3.00×10^{+00}**
	Std	**3.57×10^{-14}**	8.27×10^{-11}	1.00×10^{-13}	1.57×10^{-04}	4.39×10^{-05}	1.43×10^{-14}
f_{19}	Avg	**−3.86×10^{+00}**	**−3.86×10^{+00}**	**−3.86×10^{+00}**	**−3.86×10^{+00}**	**−3.86×10^{+00}**	**−3.86×10^{+00}**
	Std	1.69×10^{-15}	6.65×10^{-07}	5.73×10^{-11}	8.01×10^{-03}	2.50×10^{-03}	2.70×10^{-15}
f_{20}	Avg	−3.21×10^{100}	−3.25×10^{100}	−3.23×10^{100}	−3.19×10^{100}	**−3.26×10^{+00}**	−3.31×10^{100}
	Std	**2.18×10^{-02}**	5.95×10^{02}	5.51×10^{02}	1.56×10^{01}	7.55×10^{02}	4.11×10^{02}
f_{21}	Avg	**−1.02×10^{+01}**	**−1.02×10^{+01}**	−2.63×10^{+00}	−5.05×10^{+00}	−1.02×10^{+01}	**−1.02×10^{+01}**
	Std	**4.98×10^{-13}**	2.36×10^{-04}	3.28×10^{+00}	2.91×10^{+00}	2.11×10^{+00}	4.51×10^{-04}
f_{22}	Avg	**−1.04×10^{+01}**	−1.04×10^{+01}	−8.17×10^{+00}	−7.80×10^{-04}	−1.00×10^{+01}	−1.04×10^{+01}
	Std	**4.68×10^{-13}**	3.17×10^{-04}	3.29×10^{+00}	2.92×10^{-04}	1.34×10^{+00}	9.10×10^{+02}
f_{23}	Avg	**−1.05×10^{+01}**	**−1.05×10^{+01}**	−8.53×10^{+00}	−8.16×10^{+00}	−1.00×10^{+01}	−1.04×10^{+01}
	Std	**5.91×10^{-13}**	3.16×10^{04}	3.17×10^{+00}	2.99×10^{+00}	2.06×10^{+00}	9.87×10^{01}

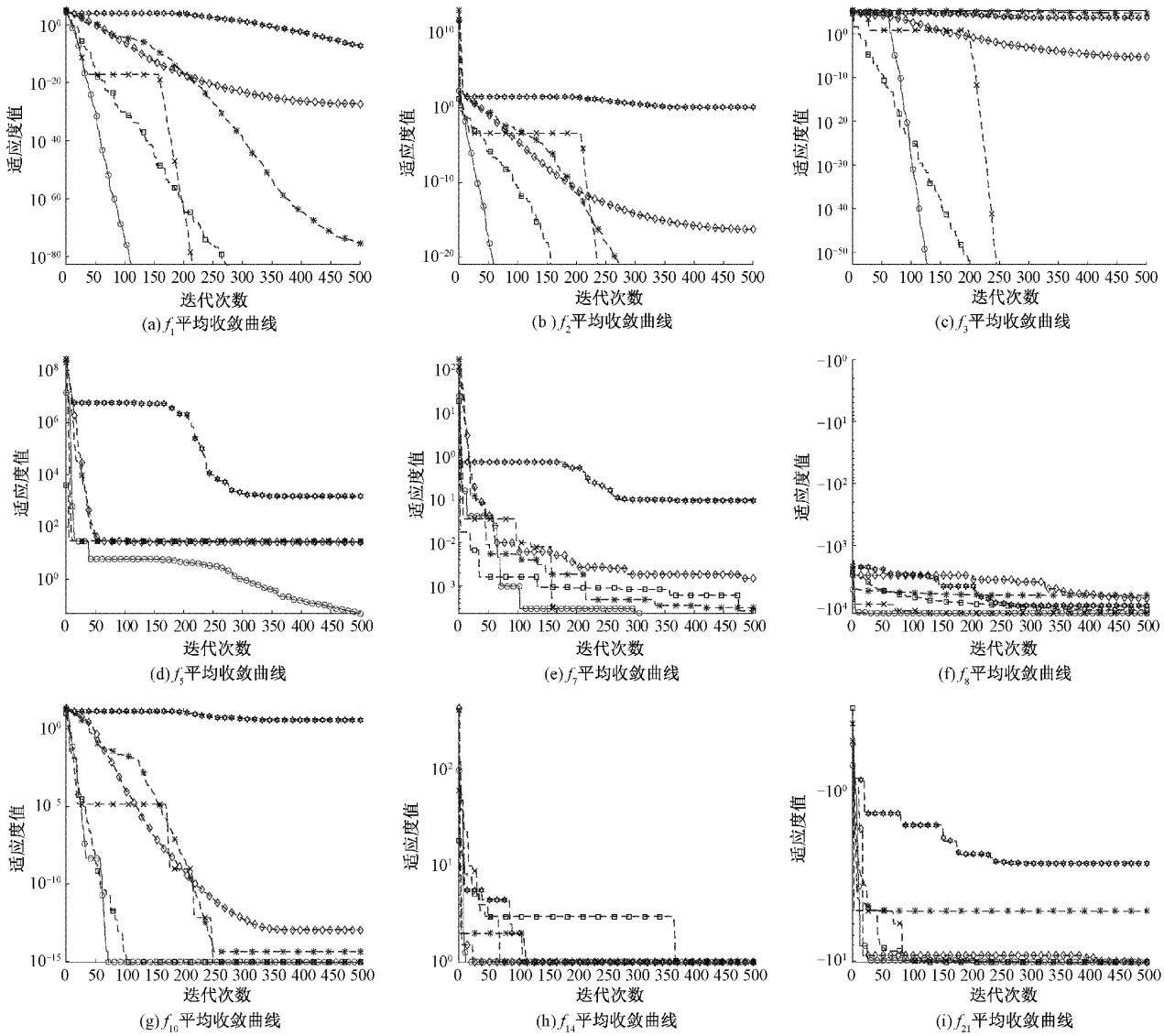

图 4　部分基准函数平均收敛曲线

步骤 5)利用式(17)求出更新后新解的反向解,计算反向解的适应度值,并两两比较选出适应度值更优的解参与下一步。

步骤 6)比较更新后黏菌适应度值与全局最优适应度值并更新全局最优解和适应度值。

步骤 7)若最优解适应度值或者迭代次数达到要求,则退出算法迭代,输出全局最优解,执行下一步骤,否则返回第 4 步继续求解寻优。

步骤 8)以 ISMA 算法的搜索结果作为 BP 神经网络的最佳始权值和阈值,对空气质量进行预测。

3　仿真验证

3.1　空气质量数据集构建

为了仿真验证本文提出的算法对空气质量指数预测的准确性,本文统计了空气质量在线监测分析平台实时监测的武汉市空气质量数据,数据采样时间为 2021 年 1 月 1 日～2023 年 2 月 1 日,共计 779 组数据,其中影响 AQI 的因素为 SO_2、NO_2、CO、O_3、PM10、PM2.5。采用灰色关联度法[15]确定各影响因素与 AQI 间的关联程度,结果如表所示。

从表中可以看出 6 个因素的灰色关联度均大于 0.7,关联度较高,即将这六个因素作为 AQI 预测模型的主要影响因素。训练集随机选取 749 组数据,剩下的 30 组数据为测试集。即将 749 组样本的 6 类影响因素数据作为输入,AQI 作为输出对模型进行训练,再将剩下的 30 组样本的 6 类影响数据代入训练好的改进 ISMA-BP 模型,将其对应输出作为 AQI 预测值,并与实际数据进行误差分析。为了避免由于数据单位或数量级的差异,而对数据模型产生影

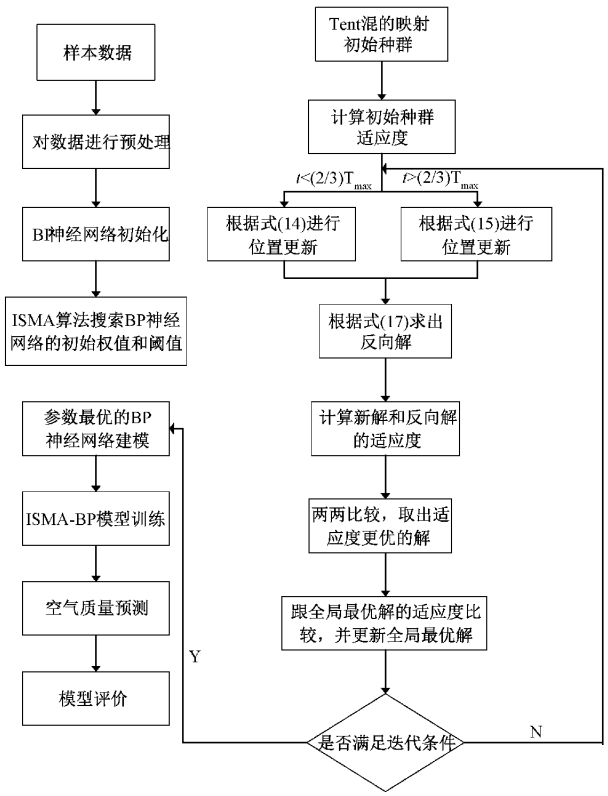

图 5　ISMA-BP 模型流程图

表 3　各因素灰色关联度

影响因素类型	灰色关联度
SO_2	0.824 5
NO_2	0.785 9
CO	0.834 6
O_3	0.774 4
PM10	0.826 3
PM2.5	0.782 0

响。利用式对样本数据进行归一化处理。

$$x' = \frac{(y_{max} - y_{min})x(x - x_{min})}{x_{max} - x_{min}} + y_{min} \tag{19}$$

式中：x_{min} 为样本数据同一类特征中的最小值，x_{max} 为样本数据同一类特征中的最大值，y_{max} 和 y_{min} 分别为指定归一化处理所到区间的最大值和最小值，x 为样本数据同一类特征中任一数值，x' 为归一化之后的数值。归一化[−1,1]区间之后的部分数据如表所示。

3.2　多模型预测性能比较

根据表 3 影响空气质量指数的因素输入个数与 AQI 的输出，可确定网络的输入层节点数为 6，输出层节点数为 1，隐含层节点数通过式(20)选取。

$$l = \sqrt{s+m} + \varphi \tag{20}$$

表 4　各影响因素及相应的空气质量指数

序号	影响空气质量指数的因素						AQI
	SO_2	NO_2	CO	O_3	PM10	PM2.5	
1	−0.466 7	0.010 3	−0.625 0	−0.633 5	−0.522 5	−0.587 0	−0.518 5
2	0.200 0	0.195 9	−0.125 0	−0.673 3	−0.261 3	−0.347 8	−0.398 1
3	−0.200 0	0.195 9	−0.500 0	−0.824 7	−0.243 2	−0.271 7	−0.324 1
4	0.333 3	0.278 4	0.125 0	−0.665 3	0.126 1	0.206 5	0.185 2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
777	−0.066 7	−0.195 9	−0.500 0	−0.243 0	−0.324 3	−0.337 0	−0.388 9
778	0.066 7	−0.051 5	−0.500 0	−0.243 0	−0.549 5	−0.608 7	−0.555 6
779	0.466 7	0.628 9	−0.125 0	−0.681 3	−0.225 2	−0.337 0	−0.231 5

其中，l 为隐含层节点数；s 为输入层节点数；m 为输出层节点数；φ 为介于 1～10 的常数。隐含层节点具体数目可利用多次试验及训练误差比较得出。根据图 6 不同隐藏层节点数的训练样本均方误差分析可知隐含层节点数为 10 时均方误差较小、故本文的 BP 神经网络结构选取 10 作为隐含层最佳节点数。

综上可得 BP 神经网络拓扑结构为 6-10-1，其余网络参数设置如下：训练次数为 1 000 次，训练目标误差设为 0.000 1，学习率设定为为 0.1，输入层与隐含层之间的传递函数为 tansig 双曲线正切函数，隐含层与输出层之间的传递函数为 purelin 线性函数。

由 6-10-1 网络拓扑结构确定 ISMA 算法每个个体的维数 D=6×10+10+10×1+1=81。

将 ISMA-BP 预测模型与 WOA-BP、SMA-BP 预测模型进行寻优适应度值比较，其适应度值迭代曲线如图 7 所示，预测结果如图 8 所示。

从图中可以看出，ISMA 算法的初始种群适应度比 WOA 和 SMA 算法要好，表明了 Tent 混沌映射初始种群的有效性。SMA 算法的最终适应度值为 0.071 4，WOA 算法的最终适应度值为 0.412 0，ISMA 的最终适应度值为 0.042 9。相较之下，ISMA 的收敛精度更高，表明 ISMA 算法具有更佳的寻优性能

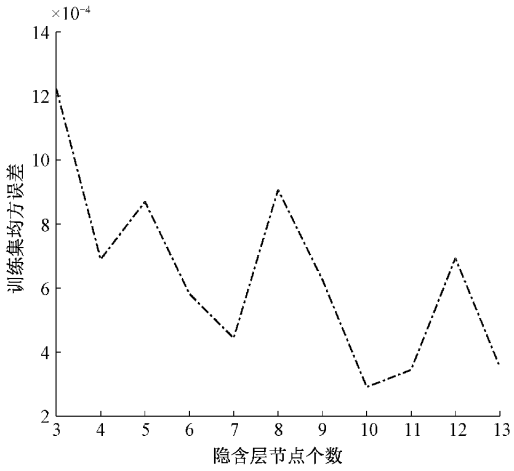

图 6　不同隐含层节点的均方根误差

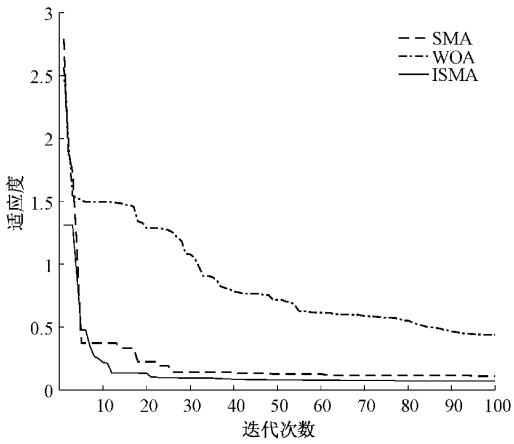

图 7　算法迭代适应度值

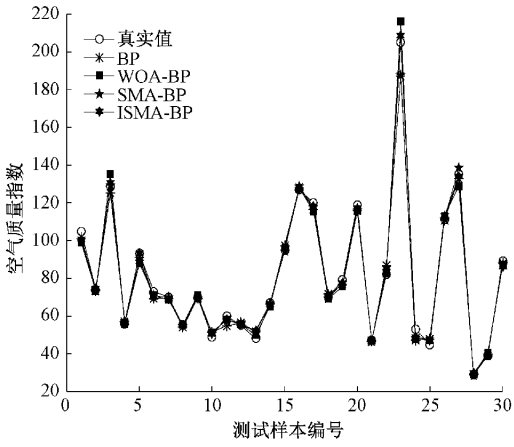

图 8　五种网络模型预测值

为了进一步比较 AQI 的预测模型的性能,选取均方根误差(MSE)、平均绝对误差(MAE)、平均绝对百分比误差(MAPE)作为评价指标,其计算公式为:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\overline{y_i} - y_i)^2 \tag{21}$$

$$MAE = \frac{\sum_{i=1}^{n}|\overline{y_i} - y_i|}{n} \tag{22}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\overline{y_i} - y_i}{y_i}\right| \tag{23}$$

式中:$\overline{y_i}$ 表示第 i 个测试样本的预测值,y_i 表示第 i 个测试样本的真实值,n 表示预测样本的个数。

利用式对上述 3 种预测模型的预测结果进行计算分析,得到 3 组指标数据,其中 MSE 用于验证模型误差平方的期望值,MAE 用于验证模型的预测准确度,MAPE 用于验证模型的平均偏离程度,其结果如表 4 所示。

表 5　4 种预测模型性能对比

预测模型	MSE	MAE	MAPE
BP	17.928 2	2.483 6	0.032 3
WOA-BP	12.537 3	2.539 3	0.028 8
SMA-BP	14.325 5	2.192 9	0.025 5
ISMA-BP	3.840 2	1.507 8	0.021 1

由表可知,ISMA-BP 的模型预测精度优于 BP、WOA 预测模型和 SMA 预测模型,MSE 分别从 17.928 2、12.537 3 和 14.325 5 降到了 3.840 2,MAE 的值分别降低了 0.975 8、1.031 8 和 0.685 1,MAPE 的值分别降低了 0.021 1、0.007 7 和 0.004 4,表明 ISMA-BP 模型对 AQI 预测的高准确性。

4　结　　论

BP 神经网络由于其随机选择的机制决定其预测性能不稳定性,而 SMA 算法存在黏菌移动过于随机且后期容易陷入局部最优解而导致其稳定性不高。针对 SMA 算法存在的之处,分别采用 Tent 混沌映射初始种群,克服其初始种群分布不均的缺陷,接着引入领导者策略和莱维飞行策略引导黏菌位置的更新,提高其全局搜索能力以及跳出局部极值的概率,最后运用动态的反向学习策略扩大黏菌搜索的空间。改进后的 ISMA-BP 模型,无论是预测精度还是稳定性方面,均优于其他模型。综上所述,ISMA-BP 模型能够对 AQI 进行较高精度与稳定性的预测。

参考文献

[1]　LIU H, SHI Y, CHEN C, ZHU D. Data multi-scale decomposition strategies for air pollution forecasting: A comprehensive review [J]. Journal of Cleaner Production, 2020, 277.

[2]　SU M S, LIU H, YU C Q, et al. A novel AQI forecasting method based on fusing temporal correlation forecasting with spatial correlation forecasting [J]. Atmospheric Pollution Research, 2023, 14(4).

[3]　李志刚,秦林林,付多民,等. 基于 CRQA-DBN-ELM

空气质量数据预测模型[J]. 电子测量技术，2022，45(19)：76-82.

[4] 石晓文，蒋洪迅. 面向高精度与强鲁棒的空气质量预测 LSTM 模型研究[J]. 统计与决策，2019，35(16)：49-53.

[5] 郑瑶，邢昱，郭悦嵩，等. 基于灰色关联模型优化的空气质量分析与可视化[J]. 能源与环保，2022，44(8)：50-56，63.

[6] 史凯赫，丁日佳，吴利丰，等. 预测空气质量的新型灰色系统多变量模型构建以石家庄市为例[J]. 系统科学学报，2023，31(2)：75-81.

[7] ZHAO X，MEI S C，AN Q L，et al. Data-driven temporal-spatial model for the prediction of AQI in nanjing[J]. Journal of Artificial Intelligence and Soft Computing Research，2020，10(4).

[8] SU Y，XIE H Y. Prediction of AQI by BP neural network based on genetic algorithm[C]. 2020 5th International Conference on Automation，Control and Robotics Engineering(CACRE)，2020：625-629.

[9] 江兵，杨春，杨雨亭，等. 基于 ACO 优化 BP 神经网络的变压器热点温度预测[J]. 电子测量与仪器学报，2022，36(10)：235-242.

[10] LI S M，CHEN H L，WANG M J，et al. Slime mould algorithm：A new method for stochastic optimization[J]. Future Generation Computer Systems，2020，111.

[11] 张玉杰，王帆. 基于改进麻雀搜索算法的照明控制优化[J]. 计算机应用，2023，43(3)：835-841.

[12] 刘公致，吴琼，王光义，等. 改进型 Logistic 混沌映射及其在图像加密与隐藏中的应用[J]. 电子与信息学报，2022，44(10)：3602-3609.

[13] 马晓宁，李笑含. 基于 Tent 混沌映射的可复制的鲸鱼算法[J]. 计算机仿真，2022，39(8)：363-368.

[14] 徐大也，胡立坤，王小勇，等. 基于概率路线图法的窄道采样与轨迹优化[J]. 国外电子测量技术，2023，42(2)：1-8.

[15] 陈岚，文斌，贺南，等. 基于融合模型动态权值的气温预测[J]. 电子测量技术，2022，45(15)：68-74.

作者简介

文昌俊，博士，教授，研究生导师，主要研究方向为质量管理与可靠性技术。

E-mail：794411947@qq.com

陈洋洋(通信作者)，硕士研究生，主要研究方向为质量管理与可靠性分析。

E-mail：1293401250@qq.com

何永豪，硕士研究生，主要研究方向为质量管理与可靠性分析。

陈凡，硕士研究生，主要研究方向为质量管理与可靠性分析。