

# 基于改进型鲸鱼优化算法和最小二乘支持向量机的炼钢终点预测模型研究

郑威迪<sup>1</sup>, 李志刚<sup>1</sup>, 贾涵中<sup>2</sup>, 高 闯<sup>1</sup>

(1. 辽宁科技大学电子与信息工程学院, 辽宁鞍山 114051; 2. 国网辽宁省电力有限公司信息通信分公司, 辽宁鞍山 114051)

**摘 要:** 终点碳含量是决定钢质量的关键因素, 是转炉炼钢过程中需要控制的核心变量之一. 本文建立了一种基于莱维飞行的鲸鱼优化算法 (Levy Whale Optimization Algorithm, LWOA) 和最小二乘向量机 (Least Squares Support Vector Machine, LSSVM) 的钢水终点碳含量综合预测模型. 通过莱维飞行代替了传统鲸鱼优化算法 (Whale Optimization Algorithm, WOA) 参数的随机选择, 优化了鲸鱼算法中跳出局部最优的能力; 借助改变鲸鱼算法的系数向量收敛方式明显提高了鲸鱼优化算法的泛化能力、预测精度和收敛速度. 数据仿真结果表明, 所提出的 LWOA-LSSVM 预测模型, 不仅能够克服局部寻优获取全局最优解, 而且具有快速的收敛速度和更高的预测精度, 得出预测结果的均方根误差、平均绝对误差和平均绝对百分比误差与遗传算法 BP 神经网络、遗传算法最小二乘支持向量机和传统鲸鱼算法最小二乘支持向量机相比均有着明显提高. 同时, 通过调整目标命中率 and 训练输入样本量验证了预测模型具有更好的鲁棒性.

**关键词:** 炼钢; 碳含量; 鲸鱼优化算法; 最小二乘法; 支持向量机; 莱维飞行

**中图分类号:** TP18

**文献标识码:** A

**文章编号:** 0372-2112 (2019)03-0700-07

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2019.03.026

## Research on Prediction Model of Steelmaking End Point Based on LWOA and LSSVM

ZHENG Wei-di<sup>1</sup>, LI Zhi-gang<sup>1</sup>, JIA Han-zhong<sup>2</sup>, GAO Chuang<sup>1</sup>

(1. College of Electronics and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, China;

2. Information Communication Branch, State Grid Liaoning Electric Power Co. Ltd, Anshan, Liaoning 114051, China)

**Abstract:** The final carbon content is the key factor in determining the quality of steel, and is one of the core variables to be controlled in the process of converter steel-making. Based on the Levy whale optimization algorithm (LWOA) and least squares support vector machine (LSSVM), a comprehensive prediction model of carbon content at the end of the steel-making process is established. When the random selection of the parameters of the traditional whale optimization algorithm (WOA) is replaced with the Levy flight algorithm, the ability to jump out of the local optimum is optimized. Changing the method of coefficient vector convergence results in improvements to the generalization ability, prediction precision and convergence speed of the WOA. Data simulation results show that the proposed LWOA-LSSVM forecasting model not only overcomes the local optimization to obtain the global optimal solution, but also achieves faster convergence speed and higher prediction accuracy. Prediction results of the model, concerning root mean square error, mean absolute error, and mean absolute percentage error, show noticeable improvements when compared to those of the genetic algorithm and back propagation (BP) neural network, the genetic algorithm and LSSVM, and the traditional WOA and LSSVM. At the same time, through adjustments of the target hit ratio and the number of training sample entries, the prediction model is proven to be more robust than the aforementioned algorithms.

**Key words:** steel-making; carbon content; whale optimization algorithm (WOA); least squares method; support vector machine; Levy flight

## 1 引言

炼钢终点预测是目前转炉炼钢行业中影响性能指标的重要一环,由于钢水终点含碳量受温度、铁水含碳量、铁水量等较多因素影响,是一个非常复杂的非线性问题,以数学推导的方式难以提出精准的数学模型,最初转炉炼钢过程中普遍采用的方法是经验预测法<sup>[1]</sup>,不仅难以保证预测精度,而且由于炼钢单次过程存在较大产量,一旦出错会造成很大的经济损失.为解决这个问题,先后出现了基于红外激光的光电探测法<sup>[2]</sup>,以及基于辐射结合神经网络的光学探测法<sup>[3]</sup>,但由于其需要在较近环境采集图像信息以维持其精确性,导致炼钢中产生的高温对设备的寿命有很大影响.之后出现的基于神经网络或多任务学习的炼钢终点预测等方法则各自有其局限性,无法以更高的预测精度完成对炼钢的终点预测<sup>[4,5]</sup>.

鲸鱼优化算法(Whale Optimization Algorithm, WOA)是通过模仿鲸鱼狩猎捕食的行为进而提出的一种新的优化算法,由格里菲斯大学的 Mirjalili 与 Lewis 等人于 2016 年完成研究<sup>[11]</sup>,与粒子群优化算法(Particle Swarm Optimization, PSO)<sup>[12]</sup>、蚁群算法(Ant Colony Optimization, ACO)<sup>[13]</sup>、人工蜂群算法(Artificial Bee Colony, ABC)<sup>[14]</sup>等通过模拟自然界动物行为而得出的群体智能优化算法相比各有优劣.以鲸鱼优化算法而言,其优势在于调节参数简单,运算速度较快,但收敛精度以及跳出局部最优的能力有待提高,有鉴于此,笔者通过改进的鲸鱼优化算法(Levy Whale Optimization Algorithm, LWOA)与最小二乘支持向量机(Least Squares Support Vector Machine, LSSVM)结合,以终点碳含量为预测目标,为转炉炼钢终点预测提出了一种基于 LWOA-LSSVM 的炼钢终点预测综合建模的方法.

## 2 支持向量机

支持向量机(Support Vector Machines, SVM)是一种基于统计学习理论的机器学习方法,其基本思想是将低维空间的数据向量映射到高维空间,利用风险最小化原理构建决策函数,用来解决样本数量少、维数高的非线性问题.最小二乘支持向量机在原基础上进行简化,以等式作为其约束条件,并采用了不同的决策函数,从而在原基础上节约了运算成本. LSSVM 将非线性向量  $\Phi(x)$  映射到高维空间,将其转变为线性回归问题,如式(1)所示:

$$y = \mathbf{w}^T \Phi(x) + b \quad (1)$$

其中  $\mathbf{w}$  是  $n$  维的权值向量,  $b$  是偏差.

考虑结构风险最小化原则,可将线性回归问题转

为求解最优问题:

$$\min J(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{1}{2} J \sum_{i=1}^m \xi_i^2 \quad (2)$$

$$y_i = \mathbf{w}^T \Phi(x_i) + b + \xi_i \quad (3)$$

其中  $J$  表示正规化函数,  $\xi_i$  表示误差,  $i$  表示该空间向量第  $i$  个维度,  $\mathbf{w}$  表示权重矢量,由于  $\mathbf{w}$  属于高维空间,不能直接求解,故引入核函数:

$$y = \sum_{i=1}^n a_i K(x, x_i) + b \quad (4)$$

为保证运算效率,本文采用 RBF 作为核函数,其函数表示如下:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right) \quad (5)$$

通过以上过程可知,对 LSSVM 模型影响能力最大的参数为  $J$  与  $\sigma^2$ ,为了获取最优的  $J$  与  $\sigma^2$ ,本文采用改进的鲸鱼优化算法寻找最优解.

## 3 鲸鱼优化算法

### 3.1 传统鲸鱼优化算法

鲸鱼优化算法是一种通过模拟鲸鱼捕食的行为进行建模得到的新型优化算法.鲸鱼通过猎物的气味寻找位置并进行包围,假设气味所反应的猎物位置为当前最优位置或接近最优位置,定义一定数量虚拟的座头鲸作为搜索代理,通过对比各种搜索代理的可行解寻找最优解,作为座头鲸下一个位置向量,同时其他搜索代理更新自身位置,以此完成寻找最优解的策略.其狩猎行为如图 1 所示.



图1 座头鲸气泡网络捕食策略

座头鲸在狩猎时会根据猎物的位置不断调整自身位置,为了描述这种捕食策略,采用如下数学模型:

$$D_1 = |CX^*(t) - X(t)| \quad (6)$$

$$X(t+1) = X^*(t) - AD_1 \quad (7)$$

式中,  $t$  为当前迭代次数,  $X(t)$  为当前座头鲸的坐标向量,  $X(t+1)$  表示下一次迭代后的目标坐标向量,  $X^*(t)$  为到目前得到最佳解的坐标向量,并且如果有更好的可行解,应立即更新  $X^*(t)$ ,  $A$  和  $C$  是系数,由式(8)、式(9)得出:

$$A = 2ar_1 - a \quad (8)$$

$$C = 2r_2 \quad (9)$$

其中,  $r_1, r_2$  为 0 到 1 之间的随机数,  $a$  由式(10)得出:

$$a = 2 - 2t/T_{\max} \quad (10)$$

座头鲸为寻找更优的目标位置以靠近猎物, 会随机采用任意鲸鱼坐标向量代替座头鲸产生下一次迭代的鲸鱼坐标向量, 从而达到偏离猎物的目的, 以此来避免陷入局部最优. 该描述采用如下数学模型:

$$X(t+1) = \begin{cases} X^*(t) - AD_1, & |A| < 1, p < 0.5 \\ X_{\text{rand}}(t) - AD_2, & |A| \geq 1, p < 0.5 \\ X^*(t) + D_3 e^{bl} \cos(2\pi l), & p \geq 0.5 \end{cases} \quad (11)$$

$$D_1 = |CX^*(t) - X(t)| \quad (12)$$

$$D_2 = |CX_{\text{rand}} - X(t)|$$

$$D_3 = |X^*(t) - X(t)|$$

其中  $t$  表示当前迭代次数,  $X^*(t)$  表示到目前为止最优位置向量,  $X_{\text{rand}}$  为随机鲸鱼位置向量,  $X(t)$  表示当前鲸鱼位置向量,  $b$  是一个常数默认为 1, 用来控制围猎路径形状,  $l$  采用下式得出:

$$l = (a_2 - 1)r_3 + 1 \quad (13)$$

$$a_2 = -1 - t/T_{\max} \quad (14)$$

其中  $r_3$  为 0 到 1 间的随机数,  $t$  表示当前的迭代次数,  $T_{\max}$  表示最大迭代次数.

当  $t \geq \frac{T_{\max}}{2}$  时, 即当迭代次数达到设定值的一半时,

$A$  恒小于 1, 此时鲸鱼进入攻击模式, 不再随机采用搜索代理进行偏离猎物的行为.

### 3.2 改进鲸鱼优化算法

WOA 算法优势在于其调节参数较少, 使用方便, 有较快的收敛速度, 有一定的跳出局部最优能力. 但是, 由于其算法本身单纯采用随机系统进行勘探, 对随机的过度依赖限制了 WOA 算法的搜索速度, 因此对 WOA 算法还可以进一步加快收敛速度并提高收敛精度; 另外, 由于系数向量  $A$  的限制, WOA 算法中迭代次数达到最大设定迭代次数的一半时会失去跳出局部最优的能力, 因此 WOA 算法有一定陷入局部最优的风险, 导致算法预测结果不精确.

为解决 WOA 算法的上述缺陷, 笔者提出了一种基于莱维飞行的改进型鲸鱼优化算法. 改进后的算法相比较 WOA 算法而言具有更快的收敛速度和更高的收敛精度, 同时具有更好的跳出局部最优的能力.

(1) 采用莱维飞行进行最佳解的寻找, 能够对 WOA 算法进行更快的收敛并达到更高的收敛精度.

莱维飞行是一种基于莱维分布的随机搜索, 近些年在优化领域中被多次引用, 常用来改进布谷鸟、粒子群等算法<sup>[9]</sup>, 是一种小范围搜索、大范围跳变的随机方式.

在鲸鱼优化算法中采用莱维飞行进行改进. 式(8)由以下式替换:

$$A = 2a\text{Levy}(\lambda) - a \quad (15)$$

其中  $\text{Levy}(\lambda)$  表示其服从参数为  $\lambda$  的莱维分布, 即:

$$\text{Levy} \sim u = t^{-\lambda} \quad (16)$$

由于莱维飞行比较复杂, 因此采用 Mantegna 算法对其进行模拟, 其数学表述如下:

$$s = \frac{\mu}{|\nu|^{\frac{1}{\beta}}} \quad (17)$$

其中  $u, v$  服从参数为  $\sigma_\mu, \sigma_\nu$  的正态分布:

$$\mu \sim N(0, \sigma_\mu^2) \quad (18)$$

$$\nu \sim N(0, \sigma_\nu^2) \quad (19)$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \beta 2^{\frac{\beta-1}{2}}} \right\} \quad (20)$$

$$\sigma_\nu = 1 \quad (21)$$

为减少整个算法的运算开销, 对  $\beta$  取常数 1.5, 此时  $\sigma_\mu$  为常数 0.6966.

(2) WOA 算法中系数向量  $A$  采用线性方式收敛, 具有一定的局限性, 为使算法具有更高的跳出局部最优的能力, 式(10)采用下式替换:

$$a = 2e^{0.15(-\log(\frac{10t}{T_{\max}}))^4} \quad (22)$$

其中  $t$  表示当前的迭代次数,  $T_{\max}$  表示最大迭代次数.

式中  $a$  的值在迭代前期随迭代次数增加会缓慢下降, 有利于跳出局部最优进行全局搜索, 而迭代后期  $a$  的值呈指数形势迅速下降, 以提高快速进行局部搜索的能力.

## 4 LWOA 算法性能测试

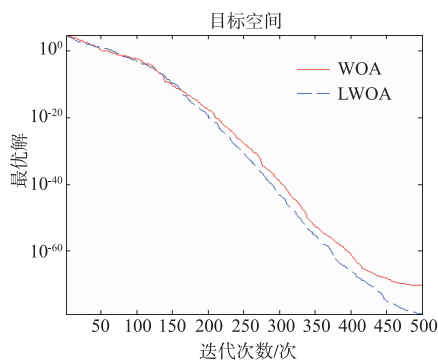
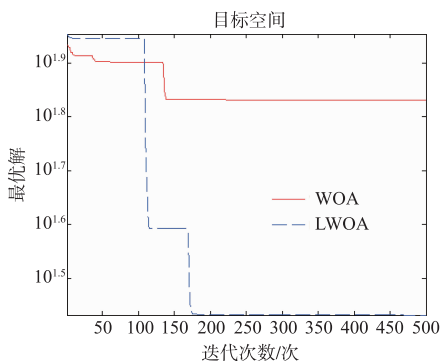
为对比 WOA 算法和 LWOA 算法运行效果, 采用如表 1 所示的 9 个基准函数进行性能测试. 为保证算法测试的公平性, WOA 算法和 LWOA 算法都采用 30 个种群规模以及 500 迭代次数. 测试结果如表 1 所示:

由表中数据可以看出, 对于单峰基准函数 ( $F_1$  到  $F_6$ ) LWOA 算法能更快地寻找到符合条件的目标; 而对于多峰基准函数 ( $F_7$  到  $F_9$ ) LWOA 算法相比与 WOA 算法具有更快的跳出局部最优的能力.

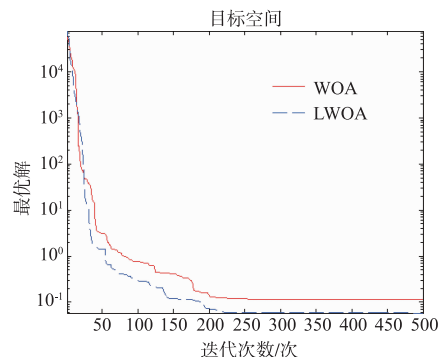
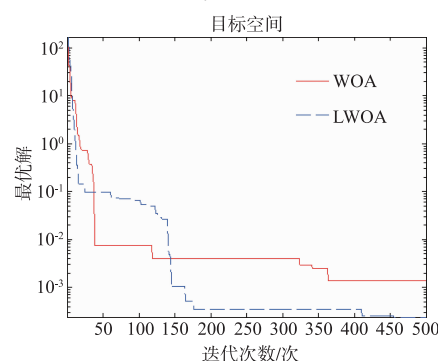
为更为明显地对比两种算法的优劣, 笔者给出  $F_1$ 、 $F_5$ 、 $F_6$  和  $F_7$  这 4 种基准函数的仿真结果, 如图 2~5 所示. 由于 LWOA 采用更有针对性的随机方式和更科学的收敛方式, 该算法更难陷入局部最优, 并具有更高的收敛速度, LWOA 算法相较于 WOA 算法约提高 5%~10% 的收敛精度.

表 1 9 个基准测试函数

Item	Function	WOA	LWOA
$F_1$	$F_1(x) = \sum_{i=1}^n x_i^2$	0	0
$F_2$	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	0	0
$F_3$	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2 \sigma_v = 1$	53410	47393
$F_4$	$F_4(x) = \max_i \{  x_i , 1 \leq i \leq n \}$	67.865	27.012
$F_5$	$F_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	28.744	27.437
$F_6$	$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	0.1172	0.0582
$F_7$	$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	0.0014	0.0002
$F_8$	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	-11674	-12346
$F_9$	$F_9(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	0	0

图2 两种算法在 $F_1$ 测试函数下的性能比较图3 两种算法在 $F_5$ 测试函数下的性能比较

为更加详细地测试 LWOA 与其他优化算法的优劣性,采用如表 2 所示的 9 种函数和 4 种优化算法进行大量的测试,为保证 4 种算法的可比较性与公平性,均采用 20~60 随机数量的搜索代理,进行最大迭代次数为 500 次的测试,独立运行 20 次,分别计算 4 种算法在 10 维、30 维和 50 维的 9 种测试函数下的优化能力,其测

图4 两种算法在 $F_6$ 测试函数下的性能比较图5 两种算法在 $F_7$ 测试函数下的性能比较

试结果如表 3 所示,可以比较明显地看出 LWOA 具有比其他算法更好的性能。

表 2 9 个基准测试函数

Item	Function
$F_1$	$F_1(x) = \sum_{i=1}^n x_i^2$
$F_2$	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $
$F_3$	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$
$F_4$	$F_4(x) = \max_i \{  x_i , 1 \leq i \leq n \}$
$F_8$	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$
$F_9$	$F_9(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$
$F_{10}$	$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$
$F_{11}$	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$
$F_{12}$	$F_{12}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$

表 3 4 种算法对基准测试函数的运行结果

Item	Dim	PSO		DE		WOA		LWOA	
		Mean	S. D.	Mean	S. D.	Mean	S. D.	Mean	S. D.
$F_1$	10	$5.667 \times 10^{-3}$	$1.582 \times 10^{-9}$	$1.641 \times 10^{-29}$	$1.539 \times 10^{-29}$	0	0	<b>0</b>	0
	30	$3.402 \times 10^{-2}$	$1.821 \times 10^{-2}$	$1.032 \times 10^{-9}$	$2.891 \times 10^{-9}$	0	0	<b>0</b>	0
	50	$2.925 \times 10^{-1}$	$9.021 \times 10^{-2}$	$6.618 \times 10^{-5}$	$4.039 \times 10^{-5}$	0	0	<b>0</b>	0
$F_2$	10	$2.061 \times 10^{-3}$	$3.205 \times 10^{-3}$	$1.372 \times 10^{-16}$	$2.114 \times 10^{-16}$	0	0	<b>0</b>	0
	30	$1.118 \times 10^1$	$5.761 \times 10^{-1}$	$2.952 \times 10^{-7}$	$1.481 \times 10^{-7}$	0	0	<b>0</b>	0
	50	$5.446 \times 10^1$	$1.891 \times 10^1$	$7.379 \times 10^{-4}$	$2.198 \times 10^{-4}$	0	0	<b>0</b>	0
$F_3$	10	$3.672 \times 10^{-6}$	$8.053 \times 10^{-6}$	$3.042 \times 10^1$	$1.374 \times 10^1$	$1.973 \times 10^4$	$1.806 \times 10^4$	$1.260 \times 10^4$	$1.587 \times 10^4$
	30	$1.921 \times 10^1$	$8.048 \times 10^{-1}$	$1.902 \times 10^4$	$2.518 \times 10^3$	$1.308 \times 10^4$	$6.465 \times 10^4$	$4.340 \times 10^4$	$4.007 \times 10^4$
	50	$4.182 \times 10^1$	$1.674 \times 10^1$	$6.612 \times 10^4$	$6.421 \times 10^3$	$9.296 \times 10^4$	$2.548 \times 10^4$	$7.372 \times 10^4$	$5.846 \times 10^4$
$F_4$	10	$7.208 \times 10^{-3}$	$6.632 \times 10^{-3}$	$2.308 \times 10^{-3}$	$1.288 \times 10^{-4}$	$3.239 \times 10^2$	$3.876 \times 10^3$	$2.121 \times 10^2$	$8.232 \times 10^1$
	30	$9.937 \times 10^{-1}$	$5.264 \times 10^{-1}$	$4.332 \times 10^1$	$2.965 \times 10^{-1}$	$5.287 \times 10^2$	$3.165 \times 10^2$	$3.425 \times 10^1$	$2.342 \times 10^2$
	50	$3.581 \times 10^1$	$9.328 \times 10^{-1}$	$2.197 \times 10^1$	$2.096 \times 10^1$	$7.469 \times 10^1$	$2.367 \times 10^2$	$4.864 \times 10^1$	$6.748 \times 10^1$
$F_8$	10	$-1.608 \times 10^3$	$1.731 \times 10^2$	$-4.189 \times 10^3$	$1.729 \times 10^{-12}$	$-3.897 \times 10^3$	$6.108 \times 10^2$	-3396.1	$5.548 \times 10^2$
	30	$-2.811 \times 10^3$	$3.466 \times 10^2$	$-1.246 \times 10^4$	$2.535 \times 10^1$	$-1.236 \times 10^4$	$1.768 \times 10^3$	<b><math>-1.328 \times 10^4</math></b>	$1.604 \times 10^3$
	50	$-3.780 \times 10^3$	$5.446 \times 10^2$	$-1.949 \times 10^4$	$1.649 \times 10^2$	$-1.917 \times 10^4$	$1.218 \times 10^3$	<b><math>-1.951 \times 10^4</math></b>	$2.811 \times 10^3$
$F_9$	10	$9.502 \times 10^1$	$4.038 \times 10^1$	$9.832 \times 10^{-2}$	$2.212 \times 10^{-2}$	0	0	<b>0</b>	0
	30	$3.242 \times 10^1$	$8.795 \times 10^1$	$8.031 \times 10^{-2}$	$2.494 \times 10^{-2}$	0	0	<b>0</b>	0
	50	$5.665 \times 10^1$	$1.159 \times 10^1$	$7.461 \times 10^{-1}$	$2.804 \times 10^{-1}$	0	0	<b>0</b>	0
$F_{10}$	10	$1.922 \times 10^1$	$1.009 \times 10^1$	$2.558 \times 10^{-14}$	$3.235 \times 10^{-14}$	$2.134 \times 10^{-15}$	$1.452 \times 10^{-15}$	<b>0</b>	0
	30	$4.052 \times 10^1$	$8.892 \times 10^{-1}$	$4.208 \times 10^{-6}$	$2.369 \times 10^{-6}$	$3.675 \times 10^{-15}$	$2.281 \times 10^{-15}$	<b>0</b>	0
	50	$5.446 \times 10^1$	$8.172 \times 10^{-1}$	$1.546 \times 10^{-3}$	$4.467 \times 10^{-4}$	$3.329 \times 10^{-15}$	$1.368 \times 10^{-15}$	<b>0</b>	0
$F_{11}$	10	$4.749 \times 10^1$	$1.443 \times 10^1$	$1.021 \times 10^{-7}$	$3.708 \times 10^{-7}$	$1.831 \times 10^{-2}$	$7.632 \times 10^{-2}$	<b><math>8.773 \times 10^{-3}</math></b>	$3.212 \times 10^{-2}$
	30	$2.606 \times 10^1$	$3.232 \times 10^1$	$2.558 \times 10^{-8}$	$5.211 \times 10^{-8}$	$3.537 \times 10^{-3}$	$1.296 \times 10^{-3}$	<b><math>2.928 \times 10^{-3}</math></b>	$1.731 \times 10^{-3}$
	50	$4.954 \times 10^1$	$5.056 \times 10^1$	$2.908 \times 10^{-4}$	$1.235 \times 10^{-4}$	$1.326 \times 10^{-3}$	$2.312 \times 10^{-3}$	<b><math>9.215 \times 10^{-4}</math></b>	$6.121 \times 10^{-3}$
$F_{12}$	10	$1.097 \times 10^1$	$3.061 \times 10^{-1}$	$9.980 \times 10^{-1}$	$1.365 \times 10^{-16}$	$1.564 \times 10^1$	$1.926 \times 10^1$	<b><math>1.494 \times 10^1</math></b>	$0.859 \times 10^1$

## 5 炼钢终点预测模型

### 5.1 模型建立及参数优化

本文以某炼钢厂实际采集的转炉冶炼数据作为研究对象,通过 200 组实验数据进行采样分析,以终点碳含量为预测目标,150 组数据进行训练,50 组数据进行预测,采用 BP (Back Propagation, 反向传播算法)、GA (Genetic Algorithm, 遗传算法)、WOA 等算法进行同等条件的预测分析,以此进行 LSSVM-LWOA 算法与 BP-GA、LSSVM-GA、LSSVM-WOA 三种算法的优劣性比较。

### 5.2 模型性能分析

在预测模型分析中,终点碳含量满足下式视为预测结果命中:

$$C_s \times (100\% - k) < C_y < C_s \times (100\% + k) \quad (23)$$

其中  $C_s$  为终点碳含量实测结果,  $C_y$  为终点碳含量预测结果,  $k$  为命中精度,一般取 2%。

对于参与对比的四种不同的优化算法均采用 20 次迭代和 30 个代理,结果如下所示。其中图 6 为 LSSVM-LWOA 的训练结果,图 7 为四种优化算法的预测结果对比。

由图 6、7 可以看出,采用 LSSVM-WOA 算法对训练样本进行预测得到的结果能做到很好的拟合效果,同时对预测样本的预测精度优于其他优化算法。

为更详细地比较四种算法的优劣,笔者采用三种不同的性能指标来综合评价四种优化算法的预测精度,分别为 RMSE (Root Mean Square Error, 均方根误差)、MAE (Mean Absolute Error, 平均绝对误差) 和 MAPE (Mean Absolute Percentage Error, 平均绝对百分

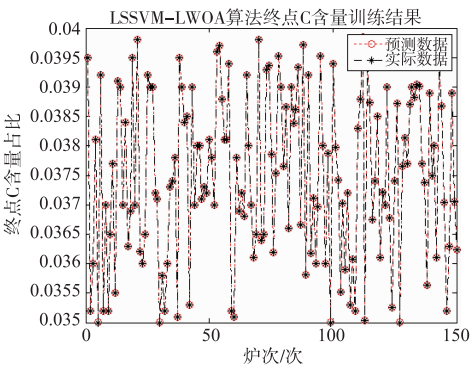


图6 训练样本预测结果对比

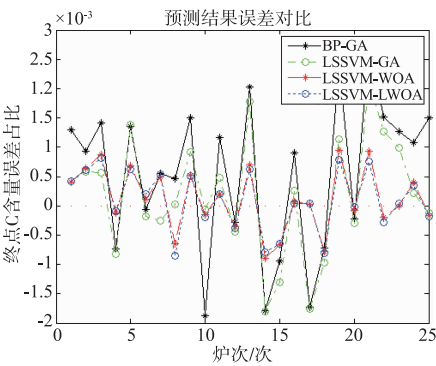


图7 预测样本预测结果误差对比

比误差),如表 4 所示. 可以通过误差数据明显看出, LSSVM-LWOA 具有更高的预测精度,其误差数据明显低于其他三种算法. 笔者认为,该结果的原因在于 LSSVM-LWOA 算法综合了各算法的优点,将 LSSVM 更快的搜索速度、Levy flight 更有目的性的随机搜索范围和 WOA 更高的收敛精度相结合,具有更强的跳出局部最优的能力,因此其实验结果的误差更小.

表 4 测试数据准确度对比

Model	RMSE	MAE	MAPE
BP-GA	$1.266 \times 10^{-3}$	$1.218 \times 10^{-3}$	$3.288 \times 10^{-2}$
LSSVM-GA	$9.979 \times 10^{-4}$	$8.066 \times 10^{-4}$	$2.168 \times 10^{-2}$
LSSVM-WOA	$5.266 \times 10^{-4}$	$4.418 \times 10^{-4}$	$1.191 \times 10^{-2}$
LSSVM-LWOA	<b><math>5.073 \times 10^{-4}</math></b>	<b><math>4.324 \times 10^{-4}</math></b>	<b><math>1.164 \times 10^{-2}</math></b>

5.3 模型鲁棒性分析

为验证算法的鲁棒性,获取了四种算法在不同条件下的性能数据. 分别取三种不同的目标命中率和四种不同大小的训练样本集作为优化条件对四种算法进行训练,其训练后的预测命中结果分别如表 5 和表 6 所示. 从结果中可以看出,改变样本集和命中概率阈值对四种模型均存在一定的影响,但相较而言 LSSVM-LWOA 算法收到的影响相对最小,可以保证在不同环境下的预测精度较为稳定,具有更好的鲁棒性.

表 5 四种算法在不同概率下的性能比较

Model	0.02	0.025	0.03
BP-GA	28%	36%	44%
LSSVM-GA	52%	60%	68%
LSSVM-WOA	76%	92%	100%
LSSVM-LWOA	<b>80%</b>	<b>100%</b>	<b>100%</b>

表 6 四种算法在不同训练样本下的性能比较

Training sample	20	50	100	150
BP-GA	26%	34%	46%	48%
LSSVM-GA	38%	40%	66%	70%
LSSVM-WOA	<b>44%</b>	60%	90%	100%
LSSVM-LWOA	40%	<b>72%</b>	<b>92%</b>	<b>100%</b>

6 结论

本文建立了一种基于 LWOA 和 LSSVM 的综合预测模型,可对炼钢转炉冶炼中的终点碳含量进行准确预测. 实验结果表明,LWOA 算法具有更高的预测精度与收敛速度,相较于 WOA 算法约提高 5%~10% 的收敛精度;同时采用 LSSVM-LWOA 等四种模型对大量转炉数据进行预测分析,证明 LSSVM-LWOA 模型具有更好的收敛精度且预测误差最小,更适合用于转炉冶炼方向.

参考文献

[1] 冯捷,张红文. 转炉炼钢生产[M]. 北京:冶金工业出版社,2006. 332-338.

[2] Daniel M. Mass spectrometry for oxygen steelmaking control[J]. Steel Times,1997,225(11):439-440.

[3] 温宏愿,赵琦,等. 基于炉口辐射和改进神经网络的转炉终点预测模型[J]. 光学学报,2008,28(11):2131-2135.

Wen Hong-yuan,Zhao Qi,et al. Basic-oxygen-furnace endpoint forecasting model based on radiation and modified neural network[J]. Acta Optica Sinica,2008,28(11):2131-2135. (in Chinese)

[4] 谢书明,陶钧,柴天佑. 基于神经网络的转炉炼钢终点控制[J]. 控制理论与应用,2003,20(6):903-907.

Xie Shu-ming,Tao Jun,Chai Tian-you. BOF steelmaking endpoint control based on neural network[J]. Control Theory & Applications,2003,20(6):903-907. (in Chinese)

[5] 程进,王坚. 基于多任务学习的炼钢终点预测方法[J]. 计算机应用,2017,37(3):889-895.

Cheng Jin,Wang Jian. Endpoint prediction method for steelmaking based on multi-task learning[J]. Journal of Computer Applications,2017,37(3):889-895. (in Chi-



nese)

- [6] 吴虎胜,张凤鸣,吴庐山. 一种新的群体智能算法——狼群算法[J]. 系统工程与电子技术,2013,35(11):2430-2438.  
Wu Hu-sheng, Zhang Feng-ming, Wu Lu-shan. New swarm intelligence algorithm—wolf pack algorithm[J]. Systems Engineering and Electronics, 2013, 35(11): 2430-2438. (in Chinese)
- [7] 严良涛,李鸣,杨大勇. 基于 GA-KPLSR 的转炉终点碳含量的预测研究[J]. 控制工程,2017,24(5):923-926.  
Yan Liang-tao, Li Ming, Yang Da-yong. Prediction of carbon content at end point based on GA-KPLSR in converters[J]. Control Engineering of China, 2017, 24(5): 923-926. (in Chinese)
- [8] 杨普,杨杰. 转炉炼钢终点控制技术现状研究[J]. 山西冶金,2017,24(4):53-54,57.  
Yang Pu, Yang Jie. Research on end-point control technology of converter steelmaking[J]. Shanxi Metallurgy, 2017, 24(4): 53-54, 57. (in Chinese)
- [9] 王庆喜,郭晓波. 基于莱维飞行的粒子群优化算法[J]. 计算机应用研究,2016,33(9):2588-2591.  
Wang Qing-xi, Guo Xiao-bo. Particle swarm optimization algorithm based on Levy flight[J]. Application Research of Computers, 2016, 33(9): 2588-2591. (in Chinese)
- [10] 牛培峰,吴志良,马云鹏,等. 基于鲸鱼优化算法的汽轮机热耗率模型预测[J]. 化工学报,2017,68(3):1049-1057.  
Niu Pei-feng, Wu Zhi-liang, Ma Yun-peng, et al. Prediction of steam turbine heat consumption rate based on whale optimization algorithm[J]. Ciesc Journal, 2017, 68(3): 1049-1057. (in Chinese)
- [11] Seyedali M, Andrew L. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95(5): 51-67.
- [12] Kennedy J, Eberhart R. Particle swarm optimization[A]. Proceedings of IEEE International Conference on Neural Networks[C]. Perth, Australia: IEEE, 1995. 1942-1948.
- [13] Dorigo M, Maniezzo V. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics, 1996, 26(1): 29-41.
- [14] Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization[R]. Kayseri Turkey: Erciyes University, 2005. 1-10.

## 作者简介



**郑威迪** 男,1994年5月生于辽宁铁岭.现为辽宁科技大学硕士研究生.主要研究方向为神经网络算法.  
E-mail:574011074@qq.com



**李志刚 (通信作者)** 男,1972年5月生于辽宁鞍山.副教授.信息系统项目管理师、一级建造师.主要研究方向为人工智能和模式识别.  
E-mail:li7275@163.com



**贾涵中** 男,1989年5月生于辽宁鞍山.2014年7月毕业于东北大学,硕士.研究方向:智能控制与无线通信技术.  
E-mail:0801jchz@163.com



**高 闯** 男,1982年5月生于辽宁鞍山.2007年1月毕业于英国伦敦大学国王学院,硕士.主要研究方向为智能控制.  
E-mail:13500422153@163.com