

基于改进麻雀搜索算法的优化型极限学习机

张恩辅, 段冰冰, 刘津平, 马云鹏, 金音

(天津商业大学信息工程学院, 天津 300134)

✉ enfuzhang@163.com; 18298884708@163.com; 2947469890@qq.com; mayunpeng@tjcu.edu.cn; 385739020@qq.com



摘要:为提升极限学习机的性能,文章利用改进的麻雀搜索算法对极限学习机的参数进行优化。首先,提出一种菱形分组机制用于增加算法种群多样性和融合模拟退火思想改善算法陷入局部极值点的缺陷。其次,通过10个基准函数进行仿真测试,实验结果表明,改进的麻雀搜索算法在大部分测试函数上表现出更好的性能。最后,将改进的算法用于优化极限学习机的输入权阈值,通过基准数据集仿真测试,优化后的极限学习机在建模精度上平均提高了7.4%。

关键词:极限学习机;麻雀搜索算法;分组机制;模拟退火

中图分类号:TP181 **文献标志码:**A

An Optimized Extreme Learning Machine Based on Improved Sparrow Search Algorithm

ZHANG Enfu, DUAN Bingbing, LIU Jinping, MA Yunpeng, JIN Yin

(College of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China)

✉ enfuzhang@163.com; 18298884708@163.com; 2947469890@qq.com; mayunpeng@tjcu.edu.cn; 385739020@qq.com

Abstract: In order to improve the performance of the Extreme Learning Machine (ELM), this paper proposes to optimize the parameters of the ELM with an Improved Sparrow Search Algorithm (ISSA). Firstly, a rhombus grouping mechanism is proposed to increase the diversity of the algorithm population and to improve the defect of the algorithm trapped in local extreme points by incorporating the Simulated Annealing idea. Then, the simulation test is carried out through 10 benchmark functions, and the experimental results show that ISAA has better performance on most test functions. Finally, ISSA is used to optimize the input weight threshold of the ELM. Through the simulation test of the benchmark data set, the optimized ELM has an average improvement of 7.4% in modeling accuracy.

Key words: Extreme Learning Machine; Sparrow Search Algorithm; grouping mechanism; Simulated Annealing

0 引言(Introduction)

极限学习机^[1](Extreme Learning Machine, ELM)是黄广斌教授于2004年提出的一种单隐层前馈神经网络模型,该模型简单、运算速度快,具有较好的回归预测能力^[2]。然而,ELM存在随机的权阈值,可能导致其出现稳定性差和泛化性能不高等问题,而群体智能优化算法可以解决以上问题。

XUE等^[3]于2020年提出麻雀搜索算法(Sparrow Search Algorithm, SSA),但该算法仍存在种群多样性不足和易陷入局部最优等缺点,为进一步提高SSA的性能,SUN等^[4]将混沌映射机制与柯西变异扰动策略结合,提升了原始算法的种群多样性,增强了全局搜索能力。LIU等^[5]首先利用教与学的策略扩大算法的搜索范围,其次结合滑动窗口技术对ELM的参数进行优化,最后应用于预测赤铁矿的研磨粒度,经过改进的

ELM拥有更快的收敛速度和预测准确度,但算法的稳定性仍有待提高。

为解决SSA存在的问题,本文提出一种改进的麻雀搜索算法(Improved Sparrow Search Algorithm, ISSA)。首先在麻雀的位置更新策略中引入一种菱形结构的分组机制,可以有效地提升算法的种群多样性,其次在更新种群的最优值时,引入模拟退火的思想,增加种群的全局搜索范围和获取最优解的概率,最后对ISSA的收敛精度和稳定性等进行测试,并应用于优化ELM的模型参数。实验结果表明:ISSA有更高的收敛精度和更好的稳定性,并且可有效地提高ELM的模型精度。

1 知识简介(Knowledge introduction)

1.1 极限学习机

ELM的核心是将单隐层神经网络的训练学习过程转化为求解线性最小二乘问题,再通过求解广义逆矩阵计算隐藏层与输出层连接权值。

设ELM中目标函数的输出为公式(1):

$$\mathbf{T} = \mathbf{H}\boldsymbol{\beta} \quad (1)$$

公式(1)中, \mathbf{H} 表示隐藏层的输出矩阵, $\boldsymbol{\beta}$ 表示隐含层与输出层的连接权值, \mathbf{T} 表示期望输出矩阵,具体表示如下:

$$\mathbf{H}(\mathbf{W}, \mathbf{b}, \mathbf{X}) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_l \cdot \mathbf{x}_1 + b_l) \\ \vdots & & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_l \cdot \mathbf{x}_N + b_l) \end{bmatrix}_{N \times l} \quad (2)$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_l]_{m \times l}^T \quad (3)$$

$$\mathbf{T} = [t_1, t_2, \dots, t_N]_{m \times N}^T \quad (4)$$

其中, N 为样本个数, m 为输出层神经元个数, l 代表隐藏层神经元个数。

连接权值 $\boldsymbol{\beta}$ 可以通过最小二乘法求解公式(5):

$$\operatorname{argmin}_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| \quad (5)$$

获得解如下:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^+ \mathbf{T} \quad (6)$$

公式(6)中的 \mathbf{H}^+ 为矩阵 \mathbf{H} 的广义逆矩阵。

1.2 麻雀搜索算法

在麻雀搜索算法中,将麻雀主要分为探索者、追随者。探索者的任务主要是探索食物,而追随者的任务是跟随探索者觅食,这两类麻雀的角色在它们的竞争过程中可能发生互换,并且均会对周围环境保持警惕,以防止天敌的到来。

在该模型中规定,适应度值较优的麻雀角色为探索者,在捕获食物的过程中拥有优先权且有更宽阔的搜索范围,因此探索者位置的更新公式由公式(7)表示:

$$\mathbf{X}_{i,j}^{t+1} = \begin{cases} \mathbf{X}_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right), & R_2 < ST \\ \mathbf{X}_{i,j}^t + Q \cdot \mathbf{L}, & R_2 \geq ST \end{cases} \quad (7)$$

由于麻雀种群中的追随者会根据探索者的能量高低(即适应度值的优劣)确定寻觅的位置,因此追随者位置的更新公式由公式(8)表示:

$$\mathbf{X}_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{\mathbf{X}_{\text{worst}}^t - \mathbf{X}_{i,j}^t}{i^2}\right), & i > \frac{n}{2} \\ \mathbf{X}_p^{t+1} + |\mathbf{X}_{i,j}^t - \mathbf{X}_p^{t+1}| \cdot \mathbf{A}^+ \cdot \mathbf{L}, & \text{其他} \end{cases} \quad (8)$$

在麻雀种群位置不断更新的过程中,假设有10%~20%的麻雀会意识到有危险发生,它们的初始位置将随机产生,并

由公式(9)进行位置的更新。

$$\mathbf{X}_{i,j}^{t+1} = \begin{cases} \mathbf{X}_{\text{best}}^t + \beta \cdot |\mathbf{X}_{i,j}^t - \mathbf{X}_{\text{best}}^t|, & f_i > f_g \\ \mathbf{X}_{i,j}^t + K \cdot \frac{|\mathbf{X}_{i,j}^t - \mathbf{X}_{\text{worst}}^t|}{(f_i - f_w) + \epsilon}, & f_i = f_g \end{cases} \quad (9)$$

2 改进的麻雀搜索算法(Improved Sparrow Search Algorithm)

2.1 分组机制

为增加麻雀种群的多样性和让种群分布更加均匀,本文将采用一种分组机制对麻雀追随者的位置进行更新。首先将所有的追随者按照适应度值从优到劣进行排序,其次根据混合蛙跳算法中的分组思想,将所有的追随者分为若干个大组^[6]。我国城市流动人口网络是以菱形结构为核心的,这样的结构使得我国流动人口的变化相对稳定,也在一定程度上验证菱形结构的稳定性较强,最后将每一大组的麻雀根据菱形结构划分为三个小组,如图1所示,菱形结构上部20%的麻雀为高级麻雀,菱形结构下部20%的麻雀为低级麻雀,中间部分60%的麻雀作为中级麻雀,级别越高麻雀的其适应度值更优^[7]。在三类麻雀中,由于高级麻雀的学习能力较强,因此将其与探索者中最优的麻雀进行学习。中级麻雀的学习能力一般,因此将其与组内最优的麻雀进行学习,并利用一个非线性的动态权重值,让其学习速度随着迭代次数的增加而增大。低级麻雀的学习能力一般,因此利用它进行全局的搜索,可防止麻雀后期陷入局部最优。追随者的位置由公式(10)和公式(11)进行更新。

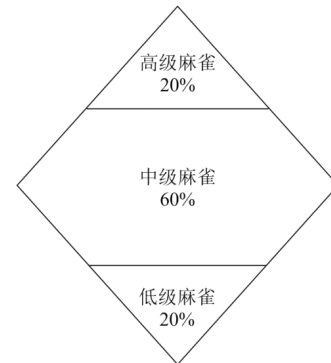


图1 麻雀数量分布图

Fig. 1 Distribution map of sparrow population

$$\mathbf{X}_{pos}^{t+1} = \begin{cases} \mathbf{X}_p^{t+1} + |\mathbf{X}_{pos}^t - \mathbf{X}_p^{t+1}| \cdot \mathbf{A}^+ \cdot \mathbf{L}, & j < \frac{t \cdot \text{Num}}{5} \\ Q \cdot \exp\left(\frac{\mathbf{X}_{\text{worst}}^t - \mathbf{X}_{pos}^t}{pos^2}\right), & j > \frac{4 \times t \cdot \text{Num}}{5} \\ \mathbf{txBest} + |\mathbf{X}_{pos}^t - \mathbf{txBest}| \times \text{randn} \times \omega, & \text{其他} \end{cases} \quad (10)$$

$$\omega = \cos\left(\frac{\pi}{2} \times \frac{t}{\text{iter}_{\max}}\right) \quad (11)$$

其中, $t \cdot \text{Num}$ 为每大组的麻雀数量, \mathbf{txBest} 为第 i 组最优麻雀的位置, randn 是一个均值为0、方差为1的标准正态分布随机数, t 为当前的迭代次数, iter_{\max} 为最大的迭代次数。 $pos = \text{num} + g \cdot \text{Cnt} \times (j - 1) + i$, num 是探索者的数量, $g \cdot \text{Cnt}$ 是划分的大组数量, i 表示麻雀所在的大组序号, j 表示麻雀在大组内的位置编号。

2.2 模拟退火思想

麻雀搜索算法的种群更新机制依赖于每一次的最优适应度

值,并以贪婪的方式对最优位置和最优适应度值进行更换,因此在算法的后期容易产生陷入局部最优、全局的搜索范围变窄等现象。模拟退火算法^[8]可以解决上述问题。模拟退火算法的全局搜索能力较强,它会根据算法的迭代次数评估算法获取最优解的可靠性,并以一定的概率接受“较差”的结果,有助于扩大麻雀种群在后期的整体搜索范围,获取到更高精度的解。因此,根据公式(12)对麻雀最优的适应度值及其位置进行更新。

$$\begin{cases} best\mathbf{X}=\mathbf{X}_i^t, \\ f_{best}=f_i^t, \end{cases} \quad f_i^t < f_{best} \text{ 或 } e^{-\frac{f_i^t-f_{best}}{T}} > rand \quad (12)$$

其中, $best\mathbf{X}$ 表示当前麻雀最优适应度值的位置, \mathbf{X}_i^t 表示第 i 个麻雀在迭代第 t 次时的位置。 f_{best} 表示当前最优的适应度值, f_i^t 表示第 i 个麻雀在迭代第 t 次时的适应度值。 T 表示当前退火的温度值, $rand$ 表示一个 $(0,1)$ 的随机数。

当 $f_i^t < f_{best}$ 时,可以根据贪心的策略对麻雀最优的适应度值及位置进行更新,有利于种群在后期的迭代过程中获取更优的适应度值。相反,当 $f_i^t \geq f_{best}$ 时,可以在满足 $e^{-(f_i^t-f_{best})/T} > rand$ 的条件下,同样对麻雀最优的适应度值及位置进行更新,从而在一定程度上提升了种群的多样性,避免了麻雀在后期的搜索过程中陷入局部最优解,并且平衡了算法的全局搜索和局部搜索能力。

为了使 ISSA 的流程更加的简洁明了,给出其流程图如图 2 所示。

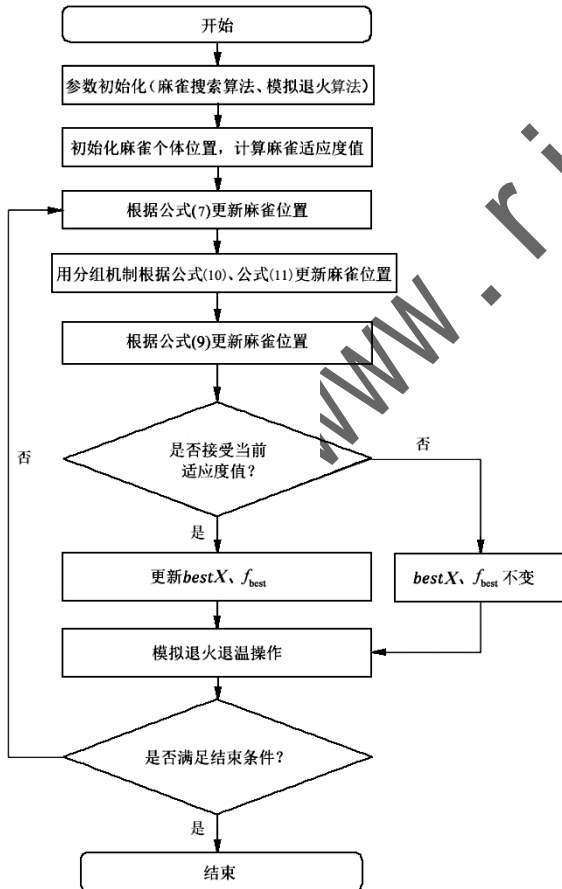


图 2 ISSA 流程图

Fig. 2 ISSA flowchart

2.3 基准函数集测试

为了更好地展现 ISSA 算法的改进效果,将 ISSA 分别与 SSA^[3]、PSO^[9]、GWO^[10]、BA^[11] 等经典群体智能优化算法进行对比。为了体现实验的公平性,此次测试设置相同的基础参数。所有群体智能优化算法的最大迭代次数 $iter_{max}$ 均为 1 000,优化的问题变量维数 dim 相同。为了测试算法在不同维度下函数的性能,分别将 dim 设为 10、30 和 50,并观察算法改进后的效果。在 SSA 和 ISSA 中,麻雀个体数量 n 均为 100,探索者的占比为 20%,追随者的占比为 80%,并随机从麻雀种群中寻找 20% 的个体作为可以意识到危险会发生的物种,安全阈值 ST 均为 0.8。模拟退火算法的初始温度 T 为 20 000 °C,温度变化率 θ 为 0.2。本次实验采用的 CPU 型号是 Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz,并在 MATLAB 2020b 仿真软件上对代码进行编译和运行。

由于每一次单独的实验具有偶然性,因此本实验将对基准函数集(表 1)的 10 个函数进行测试,使算法对每个函数独立运行 30 次,并取平均收敛值和标准差作为实验结果。这两个性能指标越小,意味着算法的收敛精度越高、稳定性越好。实验最终结果记录在表 2、表 3 和表 4 中,最优性能参数以加粗形式呈现。

表 1 基准函数集

Tab.1 Benchmark function set

函数	函数表达式	迭代次数/次	取值范围	理论最优值
F1	$f(x) = \sum_{i=1}^n (x_i)^2$	1 000	$[-100, 100]$	0
F2	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	1 000	$[-10, 10]$	0
F3	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	1 000	$[-100, 100]$	0
F4	$f(x) = \max_{1 \leq i \leq n} x_i $	1 000	$[-100, 100]$	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i^2 - 1)]$	1 000	$[-30, 30]$	0
F6	$f(x) = \sum_{i=1}^n (ix_i)^4 - \text{random}[0, 1]$	1 000	$[-1.28, 1.28]$	0
F7	$f(x) = \sum_{i=1}^n (-x_i \sin \sqrt{ x_i })$	1 000	$[-500, 500]$	$-418.9829 \times n$
F8	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	1 000	$[-5, 12, 5, 12]$	0
F9	$f(x) = -20 \exp \left\{ -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n \cos(x_i^2)} \right\} - \exp \left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + 20 + e$	1 000	$[-32, 32]$	0
F10	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	1 000	$[-600, 600]$	0

通过表 2 可以发现,ISSA 除了在函数 F_8 、 F_9 和 F_{10} 的平均收敛值与 SSA 相等,剩余函数的平均收敛值均优于 SSA。此外,尽管 SSA 在函数 F_7 的标准差略优于 ISSA,但 ISSA 的标准差在函数 F_4 、 F_5 和 F_6 均小于 SSA。表 3 中,ISSA 除了在函数 F_8 、 F_9 和 F_{10} 的平均收敛值与 SSA 相等,剩余函数的平均收敛值均优于 SSA。尽管 SSA 在函数 F_6 和 F_7 的标准差略优于 ISSA,但除了二者标准差在函数 F_8 、 F_9 和 F_{10} 相等,ISSA 的标准差均小于 SSA。表 4 中,除了函数 F_7 的标准差,ISSA 在剩余函数的平均收敛值和标准差为全局最优。此外,在三个维度下,ISSA 在大部分的函数性能均优于 GWO、PSO 和 BA。

表 2 实验结果对比 ($dim = 10$)

Tab.2 Comparison of experimental results ($dim = 10$)

函数	性能	ISSA	SSA	GWO	PSO	BA
F1	Mean	0	6.887×10^{-240}	5.238×10^{-178}	5.268×10^{-56}	5.985×10^{-5}
	Std	0	0	0	2.737×10^{-55}	1.488×10^{-5}
F2	Mean	0	1.478×10^{-210}	1.137×10^{-87}	4.445×10^{-19}	7.688×10^{-5}
	Std	0	0	6.067×10^{-87}	1.373×10^{-18}	1.487×10^{-5}
F3	Mean	0	1.242×10^{-221}	1.033×10^{-87}	1.80×10^{-18}	7.107×10^{-5}
	Std	0	0	3.291×10^{-87}	4.425×10^{-18}	2.083×10^{-5}
F4	Mean	8.552×10^{-211}	2.874×10^{-108}	1.073×10^{-56}	1.155×10^{-16}	0.004
	Std	0	1.574×10^{-107}	4.574×10^{-56}	3.397×10^{-15}	6.486×10^{-5}
F5	Mean	9.197×10^{-8}	2.786×10^{-6}	5.930	2.663	0.928 7
	Std	3.407×10^{-7}	8.610×10^{-6}	0.537	1.179	1.799
F6	Mean	4.649×10^{-5}	9.023×10^{-5}	9.836×10^{-5}	0.001 7	0.004
	Std	4.432×10^{-5}	6.203×10^{-5}	5.634×10^{-5}	0.001	0.002
F7	Mean	-3 123.008	-2 852.946	-3 067.301	-2 701.956	-Inf
	Std	512.882	367.787	329.439	284.496	NaN
F8	Mean	0	0	0	1.758	8.537
	Std	0	0	0	1.188	4.020
F9	Mean	8.882×10^{-16}	8.882×10^{-16}	4.559×10^{-15}	4.441×10^{-15}	0.750
	Std	0	0	6.486×10^{-16}	0	0.792
F10	Mean	0	0	0.011	0.131	7.176×10^{-6}
	Std	0	0	0.015	0.071	1.650×10^{-6}

表 3 实验结果对比 ($dim = 30$)

Tab.3 Comparison of experimental results ($dim = 30$)

函数	性能	ISSA	SSA	GWO	PSO	BA
F1	Mean	0	8.442×10^{-138}	1.995×10^{-85}	1.030×10^{-15}	0.001
	Std	0	4.624×10^{-137}	5.253×10^{-85}	1.428×10^{-15}	1.627×10^{-4}
F2	Mean	3.140×10^{-204}	5.286×10^{-67}	3.133×10^{-49}	1.114×10^{-7}	0.172
	Std	0	2.895×10^{-66}	4.001×10^{-49}	2.457×10^{-7}	0.099
F3	Mean	3.733×10^{-315}	2.342×10^{-162}	3.736×10^{-27}	3.033	0.003
	Std	0	1.277×10^{-161}	8.400×10^{-27}	1.683	7.086×10^{-4}
F4	Mean	1.174×10^{-202}	2.340×10^{-67}	8.590×10^{-22}	0.213	0.015
	Std	0	1.282×10^{-66}	9.427×10^{-22}	0.076	0.001
F5	Mean	1.777×10^{-8}	2.584×10^{-7}	26.075	40.552	27.440
	Std	5.798×10^{-8}	4.396×10^{-7}	0.669	33.136	1.205
F6	Mean	1.174×10^{-4}	1.280×10^{-4}	2.687×10^{-4}	0.029	0.011
	Std	1.494×10^{-4}	1.479×10^{-4}	1.771×10^{-4}	0.0108	0.004
F7	Mean	-9 723.438	-7 602.913	-6 571.52	-7 108.245	-Inf
	Std	1 647.696	806.022	571.950	683.796	NaN
F8	Mean	0	0	0.145	30.931	25.973
	Std	0	0	0.795	7.687	7.351
F9	Mean	8.882×10^{-16}	8.882×10^{-16}	1.095×10^{-14}	2.023×10^{-8}	0.603
	Std	0	0	3.106×10^{-15}	2.547×10^{-8}	0.655
F10	Mean	0	0	0.001	0.010	6.105×10^{-5}
	Std	0	0	0.003	0.010	9.776×10^{-6}

表 4 实验结果对比 ($dim = 50$)

Tab.4 Comparison of experimental results ($dim = 50$)

函数	性能	ISSA	SSA	GWO	PSO	BA
F1	Mean	0	2.282×10^{-150}	3.072×10^{-62}	3.668×10^{-7}	0.003 9
	Std	0	1.250×10^{-149}	5.836×10^{-62}	5.597×10^{-7}	3.128×10^{-4}
F2	Mean	5.87×10^{-199}	3.120×10^{-54}	1.751×10^{-36}	0.003	0.486
	Std	0	1.709×10^{-53}	9.579×10^{-37}	0.005	0.189
F3	Mean	4.31×10^{-273}	1.862×10^{-130}	8.031×10^{-15}	203.796	0.027
	Std	0	1.020×10^{-129}	2.291×10^{-14}	51.563	0.005
F4	Mean	2.800×10^{-153}	1.600×10^{-78}	3.984×10^{-15}	1.539	0.042
	Std	1.532×10^{-152}	8.763×10^{-78}	4.810×10^{-15}	0.217	0.035
F5	Mean	6.596×10^{-8}	1.922×10^{-6}	46.308	111.956	48.246
	Std	1.258×10^{-7}	2.857×10^{-6}	0.765	73.811	0.726 6
F6	Mean	1.268×10^{-4}	1.326×10^{-4}	5.238×10^{-4}	0.201	0.021
	Std	1.369×10^{-4}	1.590×10^{-4}	1.971×10^{-4}	0.045	0.006
F7	Mean	-15 811.222	-12 553.053	-9 495.657	-10 619.818	-Inf
	Std	2 579.039	936.145	1 487.914	1 187.730	NaN
F8	Mean	0	0	0.397	96.028	41.273
	Std	0	0	1.561	16.490	11.453
F9	Mean	8.882×10^{-16}	8.882×10^{-16}	2.019×10^{-14}	0.002	0.791
	Std	0	0	4.444×10^{-15}	0.005	0.665
F10	Mean	0	0	2.609×10^{-4}	0.002	1.389×10^{-4}
	Std	0	0	0.001	0.004	1.678×10^{-5}

图3至图8为仿真实验的部分适应度值对比图,其中实线表示SSA的适应度值变化过程,虚线和圆点构成的曲线为ISSA的适应度值变化过程,其他曲线为GWO、PSO和BA的适应度值变化过程。从图3至图8可知,无论是在单峰函数还是多峰函数,无论是在低维还是高维条件下,ISSA的收敛速度均比SSA的收敛速度快。

综上所述,ISSA在大部分的函数收敛精度更高、稳定性更好,有更快的收敛速度,并且更适用于求解高维函数的最优值问题。因此,ISSA存在巨大的发展空间。

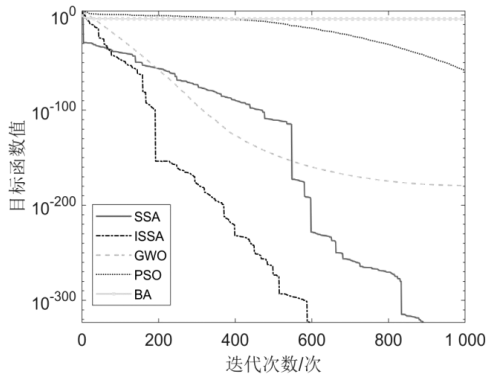


图3 函数F1(dim=10)适应度值对比

Fig. 3 Comparison of fitness values for function F1 (dim=10)

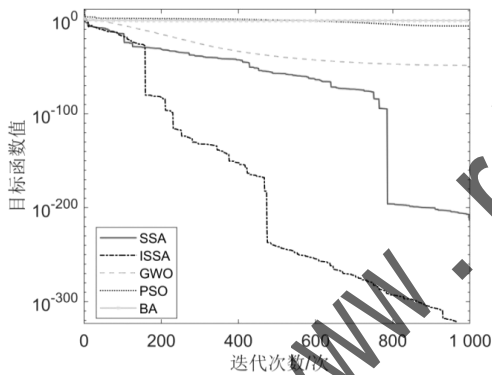


图4 函数F2(dim=30)适应度值对比

Fig. 4 Comparison of fitness values for function F2 (dim=30)

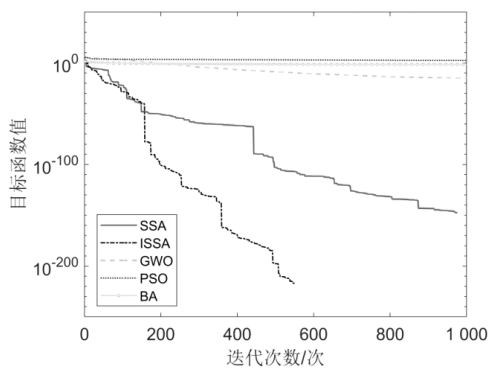


图5 函数F3(dim=50)适应度值对比

Fig. 5 Comparison of fitness values for function F3 (dim=50)

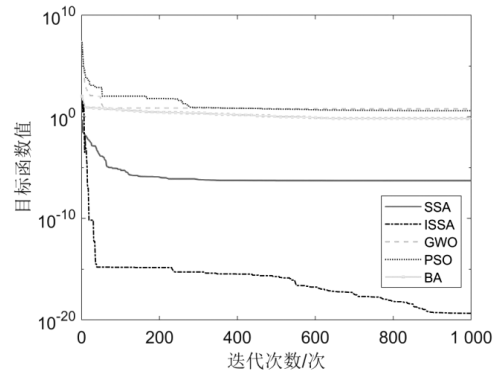


图6 函数F5(dim=10)适应度值对比

Fig. 6 Comparison of fitness values for function F5 (dim=10)

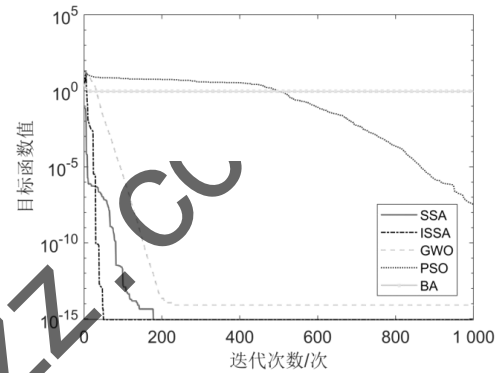


图7 函数F9(dim=30)适应度值对比

Fig. 7 Comparison of fitness values for function F9 (dim=30)

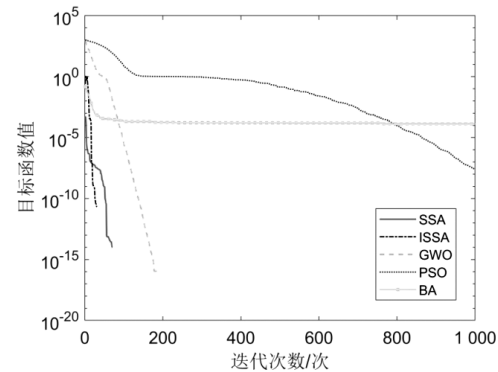


图8 函数F10(dim=50)适应度值对比

Fig. 8 Comparison of fitness values for function F10 (dim=50)

3 优化型极限学习机 (Optimized Extreme Learning Machine)

3.1 整定参数思路

由于极限学习机中输入层权重 $W_{l \times n}$ 和隐藏层神经元阈值 $b_{l \times 1}$ 为随机值,这意味着一共存在 $l \times n + l$ 个随机数,而大量的随机数容易导致算法的稳定性较差,因此可以考虑对输入层权重和隐藏层神经元阈值进行优化。整定参数的步骤如下。

- (1)数据集的选定、数据预处理工作和训练集、测试集的划分。
- (2)初始化 ELM 的参数,确定隐藏层的神经元的个数 l 、

激活函数 $g(x)$ 。

(3)初始化 ISSA 的参数,麻雀总数量为 n ,用麻雀的位置信息表示 \mathbf{W}, \mathbf{b} ,每只麻雀的位置维度 $dim=l \times n+l$ 。

(4)设 $f(x)$ 为 ELM 模型在训练过程中的均方根误差 (RMSE),利用 ISSA 寻找在 $f(x)$ 最小值时的连接权值 $\hat{\beta}$,即公式(13):

$$f(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (T_i - Y_i)^2} \quad (13)$$

其中, n 为样本总数, T_i 为预测输出的结果, Y_i 为实际输出的结果。

(5)返回 ELM 模型最优的 \mathbf{W}, \mathbf{b} 和 $\hat{\beta}$ 。

(6)利用最优的 \mathbf{W}, \mathbf{b} 和 $\hat{\beta}$ 作为 ELM 的参数,建立 ELM 预测模型。

(7)通过决定系数 (R^2)、均方误差 (MSE) 和标准差 (STD) 三个性能指标对 ISSA-ELM 的性能进行评估。

3.2 模型预测效果对比

为了验证使用 SSA 优化 ELM 的有效性和可行性,以及 ISSA 的性能,本文采用 UCI(University of California, Irvine, 加州大学欧文分校)数据集进行仿真实验,此次实验的数据集简要信息如表 5 所示,表 5 中的 6 个数据集分别记录了基于近红外光谱的汽油辛烷值含量 (Spectra)、台湾地区某地的房地产价值 (Estate)、建筑的热负荷 (Efficiency)、美国的保险费 (Insurance)、混凝土抗压强度 (Concrete) 和意大利某城市的空气湿度值 (Air Quality)。

为了保证实验的公平性,SSA 和 ISSA 的基本参数均一致,并且 ELM 模型中隐含层的神经元个数均设定为 30 个,使用的激活函数为 Sigmoid。本文将每一个数据集进行 30 次的独立实验,并取决定系数 (R^2)、均方误差 (MSE) 和标准差 (STD) 的平均值作为结果,决定系数越大、均方误差和标准差越小,说明模型预测值的拟合度更高且预测效果更稳定,最终 ELM,SSA-ELM 和 ISSA-ELM 模型的预测性能指标对比如表 6 所示。

表 5 实验数据集描述

Tab.5 Description of experimental data set

数据集名称	属性个数/个	样本总数/个	训练集个数/个	测试集个数/个
Spectra	402	60	50	10
Estate	7	414	334	80
Efficiency	9	768	588	180
Insurance	7	1 000	800	200
Concrete	9	1 030	830	200
Air Quality	11	1 500	1 200	300

表 6 预测性能指标对比

Tab.6 Comparison of predicted performance indicators

数据集	决定系数 (R^2)			均方误差 (MSE)			标准差 (STD)		
	ELM	SSA-ELM	ISSA-ELM	ELM	SSA-ELM	ISSA-ELM	ELM	SSA-ELM	ISSA-ELM
Spectra	0.774	0.833	0.926	0.770	0.442	0.170	0.078	0.204	0.044
Estate	0.611	0.631	0.641	72.546	69.857	67.164	0.100	0.100	0.090
Efficiency	0.923	0.974	0.988	7.900	2.647	1.294	0.010	0.011	0.008

续表

数据集	决定系数 (R^2)			均方误差 (MSE)			标准差 (STD)		
	ELM	SSA-ELM	ISSA-ELM	ELM	SSA-ELM	ISSA-ELM	ELM	SSA-ELM	ISSA-ELM
Insurance	0.799	0.846	0.847	3.071×10^7	2.360×10^7	2.339×10^7	0.058	0.042	0.043
Concrete	0.690	0.814	0.832	85.510	52.302	45.854	0.040	0.030	0.030
Air Quality	0.924	0.930	0.933	96.022	90.190	85.218	0.026	0.023	0.024

通过对表 6 的数据进行分析可知,尽管在第四个和第六个数据集中,SSA-ELM 模型略优于 ISSA-ELM 模型的 STD,但 ISSA-ELM 模型在 6 个数据集的决定系数和均方误差都优于 SSA-ELM 模型,并且 ISSA-ELM 所有的性能参数均远优于原始的 ELM 模型。经过计算可知,ISSA-ELM 模型的预测精度平均值比 SSA-ELM 模型高 2.3%,比 ELM 模型高 7.4%。因此,ISSA 算法对 ELM 模型的参数调优效果更佳,更适合其他研究人员利用 ELM 模型进行相关实验。实验的部分仿真效果图如图 9 至图 12 所示,其中星形实线为数据集目标值,即图例中的 Predictive Test Set,圆圈实线为 ELM 模型的预测值,叉形实线为 SSA-ELM 模型的预测值,六角星实线为 ISSA-ELM 模型的预测值,观察图 9 至图 12 可以发现 ISSA-ELM 模型的拟合效果更好。

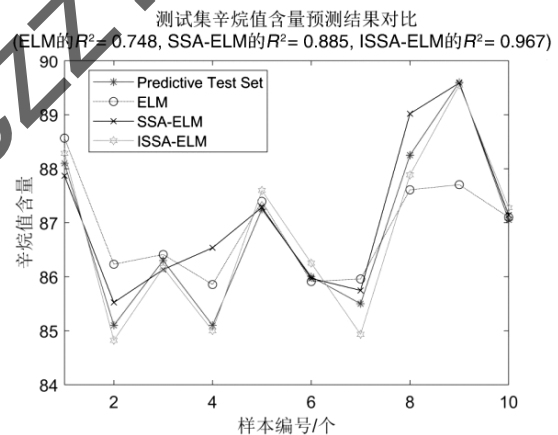


图 9 Spectra 仿真图

Fig.9 Spectra simulation diagram

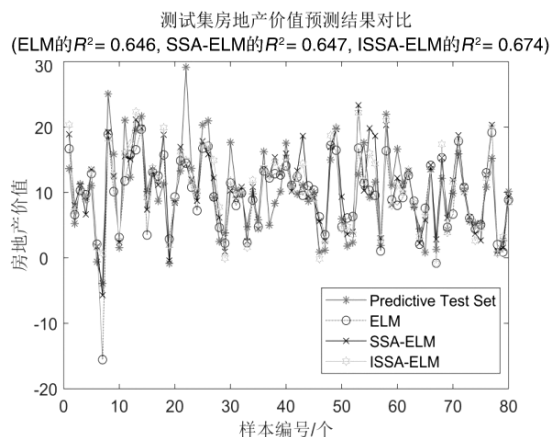


图 10 Estate 仿真图

Fig.10 Estate simulation diagram

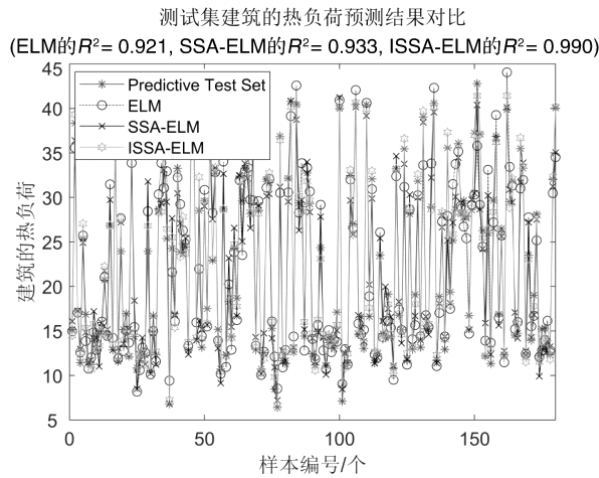


图 11 Efficiency 仿真图

Fig. 11 Efficiency simulation diagram

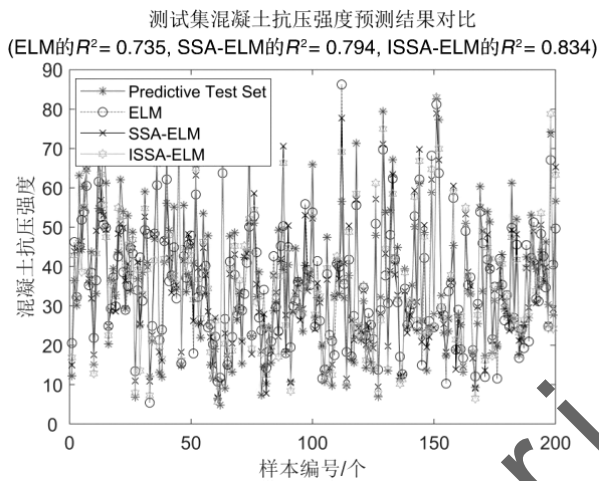


图 12 Concrete 仿真图

Fig. 12 Concrete simulation diagram

4 结论(Conclusion)

本文通过改进麻雀搜索算法的种群位置更新机制和种群寻优策略,提出了一种改进的麻雀搜索算法,并基于10个基准测试函数,将ISSA与4种传统的群体智能优化算法在3个不同的维度下进行仿真对比实验,结果表明:ISSA算法具有更高的收敛精度、更好的稳定性和更快的收敛速度,并且更适用于求解高维函数的最优值问题。此外,将ISSA用于整定极限学习机模型参数,提出了一种ISSA-ELM模型并通过UCI数据集将其与传统的ELM模型、SSA-ELM模型在模型精度和稳定性上进行对比,结果表明:ISSA-ELM具有更高的模型精度和更好的稳定性。综上,这种优化型极限学习机既具有理论意义,又具有实用价值。

参考文献(References)

[1] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: theory and applications [J]. Neurocomputing, 2006, 70(1-3): 489-501.

- [2] HUANG G B, ZHOU H M, DING X J, et al. Extreme learning machine for regression and multiclass classification [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(2): 513-529.
- [3] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm [J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.
- [4] SUN H Z, WANG J, CHEN C, et al. ISSA-ELM: a network security situation prediction model [J]. Electronics, 2023, 12(1): 25-45.
- [5] LIU H T, DAI J Y, CHEN X Y. A moving window double locally weighted extreme learning machine on an improved sparrow searching algorithm and its case study on a hematite grinding process [J]. Processes, 2023, 11(1): 169.
- [6] EUSUFF M, LANSEY K, PASHA F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization [J]. Engineering Optimization, 2006, 38(2): 129-154.
- [7] 施喻, 王士君, 王冬艳, 等. 中国市域间日常人口流动特征及影响因素 [J]. 地理科学, 2022, 42(11): 1889-1899.
- [8] HWANG C R. Simulated annealing: theory and applications [J]. Acta Applicandae Mathematica, 1988, 12(1): 108-111.
- [9] POLI R, KENNEDY J, BLACKWELL T. Particle swarm optimization [J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [10] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.
- [11] YANG X S. Bat algorithm for multi-objective optimisation [J]. International Journal of Bio-Inspired Computation, 2011, 3(5): 267-274.

作者简介:

张恩辅(2000-),男,本科生。研究领域:智能计算和机器学习。
段冰冰(2002-),女,本科生。研究领域:仿生智能计算,机器学习。

刘津平(2000-),男,本科生。研究领域:仿生智能计算,特征提取。

马云鹏(1989-),男,博士,副教授。研究领域:复杂过程控制,机器学习,仿生智能计算。本文通信作者。

金音(1985-),女,硕士,政工师。研究领域:仿生智能计算,机器学习。