

http://bhxb.buaa.edu.cn jbuua@buaa.edu.cn

DOI: 10.13700/j.bh.1001-5965.2021.0629

基于混合策略的麻雀搜索算法改进及应用

宋立钦¹, 陈文杰^{1,*}, 陈伟海², 林岩², 孙先涛¹

(1. 安徽大学 电气工程与自动化学院, 合肥 230601; 2. 北京航空航天大学 自动化科学与电气工程学院, 北京 100191)

摘要: 针对麻雀搜索算法 (SSA) 搜索精度不高、全局搜索能力不强、收敛速度慢和易于陷入局部最优等问题, 提出了一种基于混合策略的麻雀搜索算法 (HSSA)。采用改进的 Circle 混沌映射初始化种群, 提高种群多样性; 结合樽海鞘群算法改进发现者的搜索公式, 提高算法迭代前期的全局搜索能力和范围; 在加入者的搜索公式中引入自适应步长因子, 提高算法的局部搜索能力和收敛速度; 通过镜像选择机制, 提升每次迭代后的个体质量, 提高算法的寻优精度和寻优速度; 在位置更新处加入模拟退火机制, 帮助算法跳出局部最优。利用 8 种测试函数进行测试, 结果表明, 改进算法比 SSA 有更好的寻优性能。将改进前后算法与极限学习机结合进行实验, 人体表面肌电信号数据集的分类预测精度从 80.17% 提高到 90.87%, 证实了改进算法的可行性和良好性能。

关键词: 麻雀搜索算法; Circle 混沌映射; 樽海鞘群算法; 镜像选择; 自适应步长因子; 模拟退火机制; 极限学习机

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-5965(2023)08-2187-13

麻雀搜索算法^[1](sparrow search algorithm, SSA) 是 2020 年由薛建凯等提出的一种新型智能优化算法, 与粒子群优化^[2](particle swarm optimization, PSO) 算法、人工蜂群^[3](artificial bee colony, ABC) 算法等同属于群体智能优化算法。SSA 算法通过不断更新个体位置, 模拟麻雀的捕食行为来寻找最优食源。相比传统算法, SSA 算法原理简单、易于实现且寻优能力相对较强, 在单峰、多峰等测试函数上的表现优于 PSO、ABC 等传统算法。但由于 SSA 算法在种群初始化、位置更新策略等方面存在漏洞, 导致其存在全局搜索能力和局部搜索能力不协调、收敛速度慢和易于陷入局部最优等问题。针对这些问题, 学界已经有了广泛的研究。吕鑫等^[4]根据鸟群算法原理改进了算法发现者和加入者的个体更新方式, 增强了算法全局搜索能力。汤安迪等^[5]融合了正弦余弦算法和反向学习策略, 提高了算法

的开发和探索能力。黄敬宇^[6]引进了自适应 t 分布和 Tent 混沌映射, 提高了算法的收敛速度和跳出局部最优的能力。虽然这些改进算法一定程度上提升了 SSA 算法的性能, 但是在增强算法全局和局部搜索能力、提高算法收敛速度及帮助算法跳出局部最优等方面还有待更深一步的研究。

为此, 本文提出了一种基于混合策略的麻雀搜索算法 (hybrid strategy-based sparrow search algorithm, HSSA), 依次引进了 Circle 混沌映射、樽海鞘群算法、自适应步长因子、镜像选择和模拟退火 5 种策略, 对 SSA 算法进行改进。通过比较发现, 改进算法的全局和局部搜索能力、收敛速度和跳出局部最优的能力有了较大的提升。

目前, SSA 等群体智能优化算法被广泛运用到了工程问题上。例如, 徐健等^[7]运用改进的 ABC 算法优化 BP 神经网络对脑电信号进行分类识别。

收稿日期: 2021-10-23; 录用日期: 2021-12-10; 网络出版时间: 2022-01-25 17:50

网络出版地址: kns.cnki.net/kcms/detail/11.2625.V.20220121.1826.001.html

基金项目: 国家自然科学基金 (51975002)

*通信作者. E-mail: wjchen@ahu.edu.cn

引用格式: 宋立钦, 陈文杰, 陈伟海, 等. 基于混合策略的麻雀搜索算法改进及应用 [J]. 北京航空航天大学学报, 2023, 49 (8): 2187-2199.
SONG L Q, CHEN W J, CHEN W H, et al. Improvement and application of hybrid strategy-based sparrow search algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics, 2023, 49 (8): 2187-2199 (in Chinese).

刘栋等^[8]运用 SSA 算法结合极限学习机 (extreme learning machine, ELM) 对短期风电功率进行预测。蒋艳会和李峰^[9]结合改进的 PSO 算法与多阈值法来实现图像的分割。通过结果发现,这些群体智能优化算法使研究目标的精度和效率有了较大的提升。本文实验运用改进算法结合 ELM^[10]对人体表面肌电信号数据集进行分类预测,表明了改进算法在工程问题上的实际性能。

1 麻雀搜索算法的改进

1.1 麻雀搜索算法

在 SSA 算法中,个体分为 3 种,分别为发现者、加入者和警戒者,个体的位置代表候选解。SSA 算法设定发现者和警戒者的比例各为 10%~20%,发现者和加入者的身份可以互换,并且两者比例保持不变,其有各自的职责,发现者为种群提供觅食区域和方向,加入者跟随发现者进行觅食,警戒者负责食源区域周围的监视。在觅食过程中,三者位置不断更新,进而获取最优食源,最优食源的位置即为寻得的最优解。为了提高 SSA 算法的寻优性能,改善算法中存在的问题,对 SSA 算法进行了改进。

1.2 麻雀搜索算法的改进策略

1.2.1 改进的 Circle 混沌映射初始化种群

由于麻雀个体各个维度的大小在初始化阶段是随机生成的,导致初始解易出现聚集,在解空间里覆盖率不高,个体之间差异性较低,而混沌映射初始化种群可以有效改善该问题。目前, Tent 混沌映射^[11]、logistic 混沌映射^[12]和 Circle 混沌映射^[13]较为常见。通过文献 [14] 可知, Circle 混沌映射比较稳定且混沌值的覆盖率较高。但 Circle 混沌映射混沌值依旧分布不均匀,在 [0.2,0.6] 之间的取值较为密集,因此将 Circle 混沌映射公式进行改进,使其混沌值分布更加均匀。

原 Circle 混沌映射表达式为

$$x_{n+1} = \text{mod} \left(x_n + 0.2 - \frac{0.5}{2\pi} \sin(2\pi x_n), 1 \right) \quad (1)$$

改进后的 Circle 混沌映射表达式为

$$x_{n+1} = \text{mod} \left(3.85x_n + 0.4 - \frac{0.7}{3.85\pi} \sin(3.85\pi x_n), 1 \right) \quad (2)$$

式中: n 为解的维度。为直观清晰地显示改进效果,取 n 为 2000。

改进前后两者的初始解维度分布和直方图如图 1~图 4 所示。

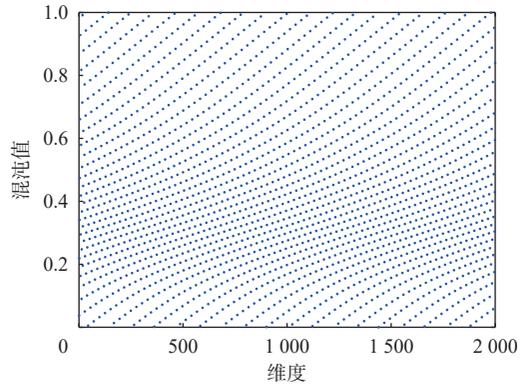


图 1 Circle 混沌映射分布

Fig. 1 Distribution of Circle chaotic mapping

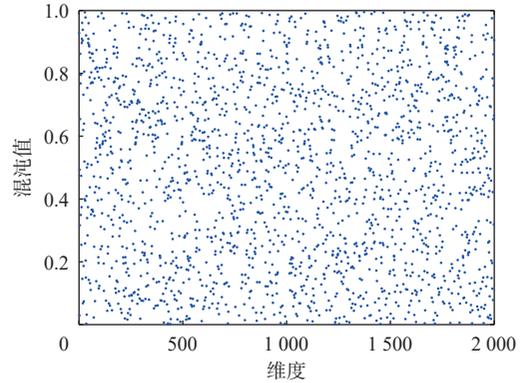


图 2 改进 Circle 混沌映射分布

Fig. 2 Distribution of improved Circle chaotic mapping

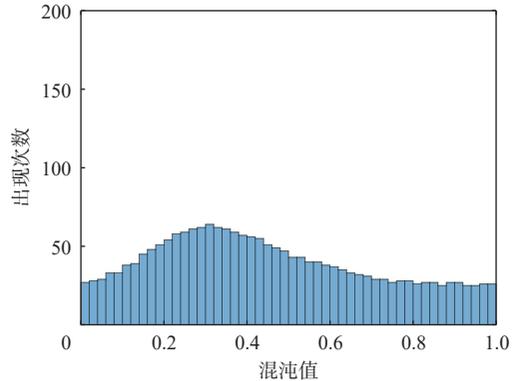


图 3 Circle 混沌映射分布直方图

Fig. 3 Distribution histogram of Circle chaotic mapping

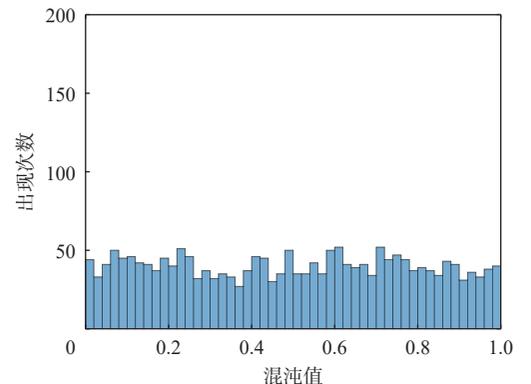


图 4 改进 Circle 混沌映射分布直方图

Fig. 4 Distribution histogram of improved Circle chaotic mapping

将图 1 和图 2 进行对比、图 3 和图 4 进行对比, 可以发现, 改进的 Circle 混沌映射混沌值分布更为均匀。因此, 采用改进的 Circle 混沌映射初始化种群, 增强种群的多样性, 进而增强算法的寻优能力。

1.2.2 发现者位置更新策略的改进

SSA 算法中发现者的位置更新公式如下:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \exp\left(-\frac{i}{\alpha \cdot i_{\max}}\right) & R_2 < N_{ST} \\ x_{i,j}^t + QL & R_2 \geq N_{ST} \end{cases} \quad (3)$$

式中: α 为 $(0,1]$ 之间的随机数; i_{\max} 为总迭代次数; Q 为服从正态分布的随机数; R_2 为预警值, 取 $[0,1]$ 之间的随机数; ST 为安全值, 本文取 0.8。

当 $R_2 < ST$ 时, SSA 算法的个体收敛于最优解的方式是向零点靠近, 每次迭代后个体的位置都在变小, 在零点附近局部搜索能力较强, 导致 SSA 算法前期搜索范围不足, 全局搜索能力不强, 易漏掉非零处的最优解。为了解决上述问题, 本文借鉴了樽海鞘群算法^[15] 领导者的更新策略:

$$x_d^1 = \begin{cases} F_d + c_1[(S_{ub} - S_{lb})c_2 + S_{lb}] & c_3 \geq 0.5 \\ F_d - c_1[(S_{ub} - S_{lb})c_2 + S_{lb}] & c_3 < 0.5 \end{cases} \quad (4)$$

式中: x_d^1 和 F_d 分别为第 1 个个体在 d 维的位置和食源在 d 维的位置; S_{ub} 和 S_{lb} 分别为解空间的上下限值; c_2 和 c_3 为 $[0,1]$ 之间的随机数; c_1 在算法的全局和局部搜索中发挥着重要作用, 其表达式为

$$c_1 = 2e^{-(4t/T_{\max})^2} \quad (5)$$

其中: t 为当前迭代次数; T_{\max} 为最大迭代次数。

由于表达式 $c_1[(S_{ub} - S_{lb})c_2 + S_{lb}]$ 可以协调樽海鞘群的全局和局部搜索能力, 提高算法前期搜索范围, 将其应用于 SSA 算法发现者的位置更新策略中。但直接加入 SSA 算法会导致前期搜索范围过广, 降低搜索精度和效率, 因此对表达式进行修改, 满足提高 SSA 算法前期全局搜索能力和范围的要求。又因为 $R_2 \geq N_{ST}$ 时, 式 (3) 左右两侧代表个体在某维度的位置, 为具体的数值, 所以 L 应略去。最终发现者的位置更新新表达式如下:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \frac{c_1[(S_{ub} - S_{lb})c_2 + S_{lb}]}{(1 + c_3)S_{ub}} & R_2 < N_{ST} \\ x_{i,j}^t + Q & R_2 \geq N_{ST} \end{cases} \quad (6)$$

分别对改进前后的发现者位置更新策略做仿真, 结果如图 5 和图 6 所示。

根据 2 个更新策略的仿真可知, 改进后的 SSA 算法发现者在迭代前期不一定每维都在变小, 增加了每维向正负方向变大的可能性, 提高了麻雀种群的搜索范围和全局搜索能力, 同时也兼顾了算法迭

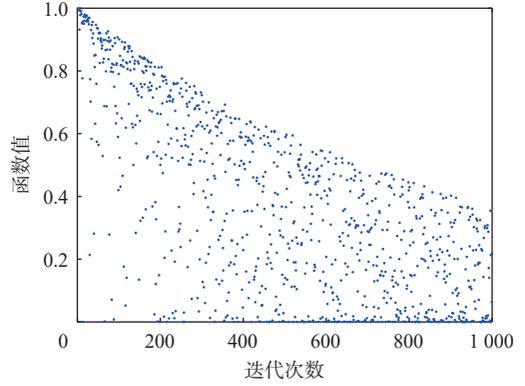


图 5 改进前发现者位置更新

Fig. 5 Location update of discoverer before improvement

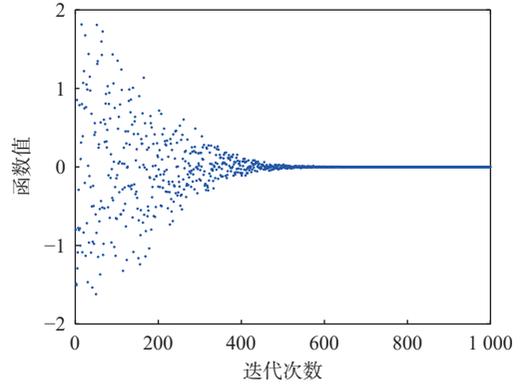


图 6 改进后发现者位置更新

Fig. 6 Location update of discoverer after improvement

代后期的收敛速度和局部搜索能力。因此, 改进后的发现者位置更新策略有效协调了算法的全局和局部的搜索能力。

1.2.3 加入者位置更新策略的改进

加入者的位置更新公式如下:

$$x_{i,j}^{t+1} = \begin{cases} Q \exp\left(\frac{(x_{w,j}^t - x_{i,j}^t)}{i^2}\right) & i > n/2 \\ x_{p,j}^{t+1} + \frac{1}{D} \sum_{j=1}^D (\text{rand}\{-1, 1\} |x_{i,j}^t - x_{p,j}^{t+1}|) & \text{其他} \end{cases} \quad (7)$$

式中: $x_{w,j}^t$ 为第 t 次迭代后种群最差个体在第 j 维的位置; $x_{p,j}^{t+1}$ 为第 $t+1$ 次迭代时发现者最优个体在第 j 维的位置。

由加入者更新公式可知, 加入者会以一定概率向发现者的最优位置靠近, 但是加入者向发现者最优位置移动的距离随机。位置更新后, 加入者不宜距离发现者最优位置过大, 如果距离过大, 容易导致算法收敛速度慢, 最优位置处的局部搜索能力较差。因此, 引进自适应步长因子, 控制加入者和发现者各维度之间的距离。当加入者和发现者最优位置较远时, 步长较大, 增加算法收敛速度, 反之, 减小步长, 增加算法的局部搜索能力。改进后加入

者的位置更新表达式如下:

$$x_{i,j}^{t+1} = \begin{cases} Q \exp((x_{w,j}^t - x_{i,j}^t) / i^2) & i > n/2 \\ x_{p,j}^{t+1} + H_{i,j} \frac{1}{D} \sum_{j=1}^D (\text{rand}\{-1,1\}(x_{i,j}^t - x_{p,j}^{t+1})) & \text{其他} \end{cases} \quad (8)$$

$$H_{i,j} = s_1 + (s_2 - s_1) |x_{i,j}^t - x_{p,j}^{t+1}| / M_j \quad (9)$$

式中: $H_{i,j}$ 为第 i 个加入者在第 j 维的步长因子; s_1 取 0.001, s_2 取 1; M_j 为当前加入者和发现者最优位置之间在第 j 维的最大距离。

1.2.4 镜像选择

镜像选择^[16]基于反向学习^[17]的思想,对每次迭代后的种群进行变异操作。变异公式为

$$X_i' = \mathbf{u} + \mathbf{l} - X_i \quad (10)$$

式中: X_i 为迭代后的第 i 个个体; X_i' 为通过变异产生的镜像个体; \mathbf{u} 和 \mathbf{l} 分别为个体取值的上下界。

新生成的 N 个镜像个体和原种群的 N 个个体合并,根据适应度值排序,由优胜劣汰原则选出 N 个较为优秀的个体进行最优位置更新,进行下次循环。该方法通过种群变异增加了种群多样性,通过对最优个体的选择,提升了每次迭代后个体的质量,提高了算法的寻优效率和收敛速度,同时也提高了算法的寻优精度。为此,选取 30 个麻雀个体对式 (11) 进行一次迭代寻优实验,观察镜像选择前后的种群位置对比(见图 7、图 8)。图中,全局最优位置坐标为 (0,0,0)。

$$f(x) = \sum_{i=1}^n x_i^2 \quad (11)$$

由镜像选择前后的种群位置可知,镜像选择后生成的新种群较原种群的位置更加优秀,进而提高了算法的寻优速度和寻优精度。变异后得到的新种群对于算法跳出局部最优也有一定的帮助。

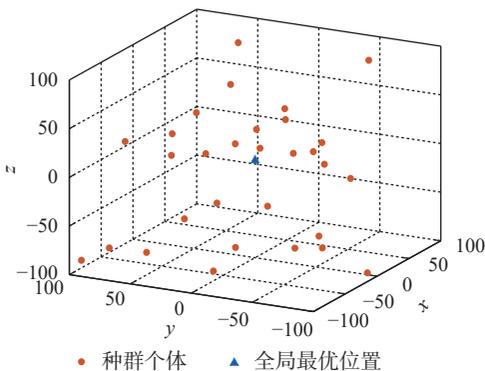


图 7 镜像选择前种群个体位置

Fig. 7 Individual position of population before mirror selection

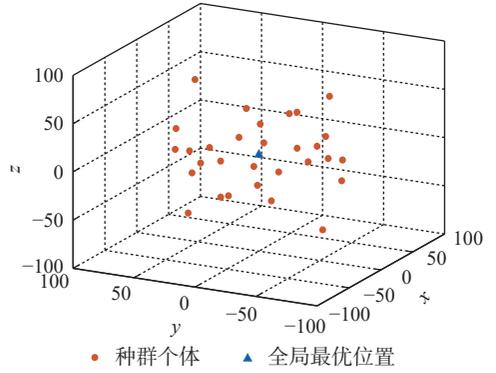


图 8 镜像选择后种群个体位置

Fig. 8 Individual position of population after mirror selection

1.2.5 模拟退火机制

为了更加高效地提升 SSA 算法跳出局部最优的能力,引进了模拟退火^[18]机制。即当新位置的适应度值大于当前全局最优位置的适应度值时,仍然有一定的概率接受该新位置。其中,退火温度 T 决定了麻雀种群接收新位置的概率, T 越高,麻雀种群接收新位置的概率越高,反之则越低。退火温度 T 随着迭代次数的增加而递减,退火温度 T 的表达式为

$$T(t+1) = \varepsilon T(t) \quad (12)$$

式中: t 为迭代次数; ε 为退火系数,取值为 [0.9,1]。

进行全局最优位置更新时,将新位置的适应度值 F_n 和原位置的适应度值 F_0 作差:

$$\Delta F = F_n - F_0 \quad (13)$$

当 $\Delta F \leq 0$ 时,直接接受新位置。当 $\Delta F > 0$ 时,根据式 (14) 进行判断:

$$\exp(-\Delta F/T) \geq \text{rand}(0,1) \quad (14)$$

如果式 (14) 成立,则接受新位置,反之不接受。

1.3 HSSA 算法整体流程

步骤 1 初始化参数,包括种群数量、发现者和警戒者的比例、警戒阈值和安全阈值等。

步骤 2 利用改进的 Circle 混沌映射初始化种群。

步骤 3 计算麻雀个体的适应度值,并对其排序,找出最优和最差适应度值的个体及其位置。

步骤 4 根据式 (6) 进行发现者位置的更新。

步骤 5 根据式 (8) 进行加入者位置的更新。

步骤 6 根据 SSA 算法的警戒者更新公式进行警戒者的位置更新。

步骤 7 对更新后的麻雀个体的位置进行镜像选择,选出适应度值较好的前 N 个个体组成新的麻

雀种群。

步骤 8 将新种群的适应度值与原种群的适应度值进行比较, 进行个体位置的更新。

步骤 9 由模拟退火机制进行全局最优位置的更新。

步骤 10 判断是否达到迭代循环结束条件, 若达到, 则进行步骤 11, 否则, 跳转到步骤 3。

步骤 11 算法执行结束, 输出并记录最优结果。

2 性能测试

2.1 基准测试函数

为了测试 HSSA 算法的寻优性能, 选取了 8 种基准测试函数^[19]用于测试。为了保证算法的可靠性, 这些函数包含了单峰和多峰函数, $f_1 \sim f_5$ 为单峰基准测试函数, $f_6 \sim f_8$ 为多峰基准测试函数。基准测试函数如表 1 所示, 其中, D 表示维度。

表 1 基准测试函数

Table 1 Benchmark function

函数	公式	维度	搜索范围	最优值
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
Schwefel 1.2	$f_2(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[-100,100]	0
Schwefel 2.22	$f_3(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
Rosenbrock	$f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
Quartic	$f_5(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	30	[-1.28,1.28]	0
Schwefel 2.26	$f_6(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	-418.982 9D
Rastrigin	$f_7(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0

2.2 HSSA 算法性能比较和分析

实验采用的 PC 设备参数为: i5-8300H CPU, 2.30 GHz, 运行内存 16 GB, Windows10 系统。在 MATLAB2020b 上进行仿真实验。依次将 HSSA 算法和不同种类的智能优化算法、已有的改进 SSA 算法做对比, 验证 HSSA 算法的优越性。同时, 将不同策略进行对比, 验证引进策略的有效性。

2.2.1 不同种类智能优化算法比较

将 HSSA 算法同 SSA 算法、PSO 算法、ABC 算法和鲸鱼优化算法^[20](whale optimization algorithm, WOA) 在 8 个基准测试函数上进行寻优对比。各算法种群数均设为 30, 迭代次数为 500, 收敛曲线的维度和搜索上下界范围依照上述 8 种基准测试函数设定。由于算法的寻优性能会随求解维度的改变而改变, 维度越高, 对于函数的求解复杂度会增加, 函数的求解维度分为 30 和 80, 以验证不同维度对各算法的影响。其余参数如表 2 所示。5 种优化算法在 8 种基准测试函数上的收敛曲线如图 9 所示。为了降低实验的偶然性, 增加实验结果的说服力, 各算法分别在 8 种基准测试函数上独立运行 50 次, 求得函数的最优值、平均值和标准差。实验

表 2 算法参数设置

Table 2 Algorithm parameter setting

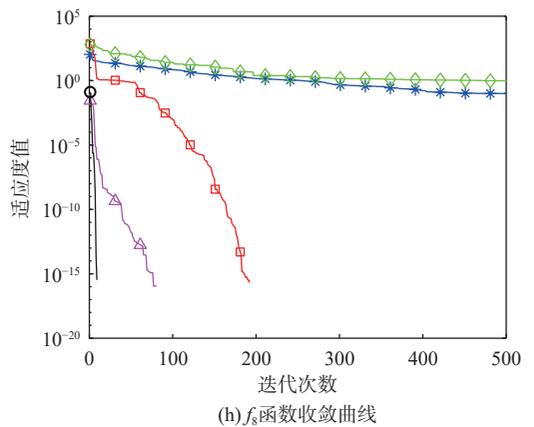
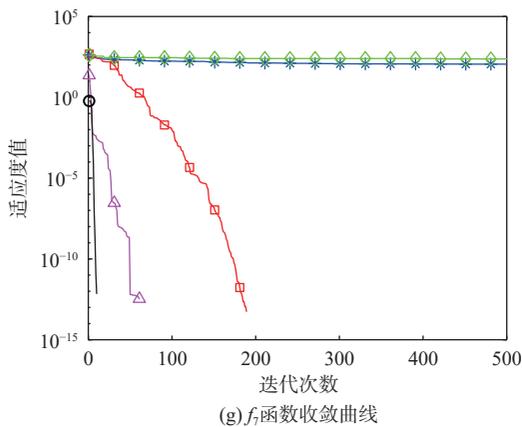
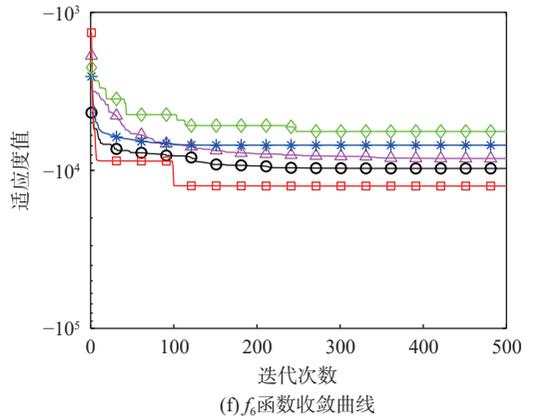
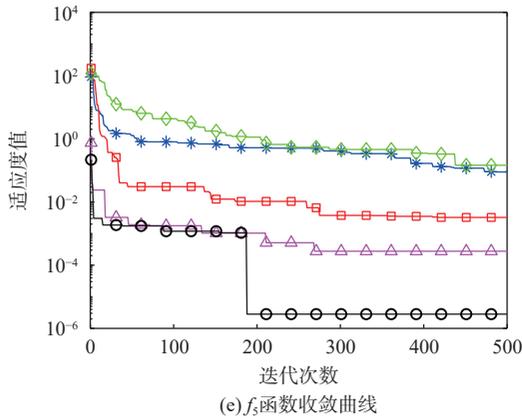
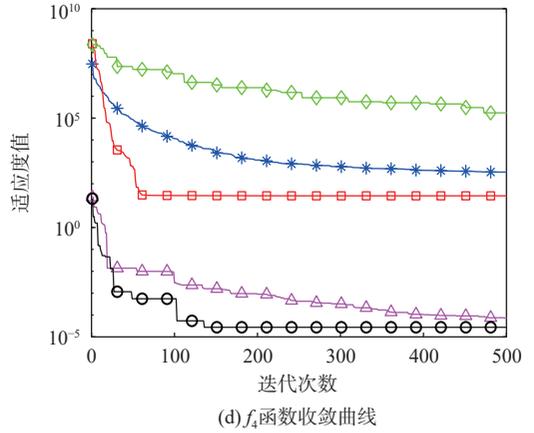
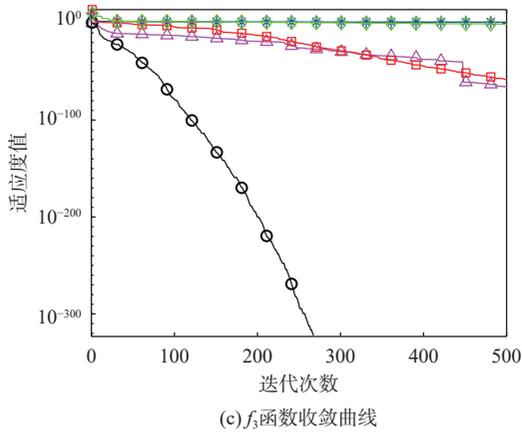
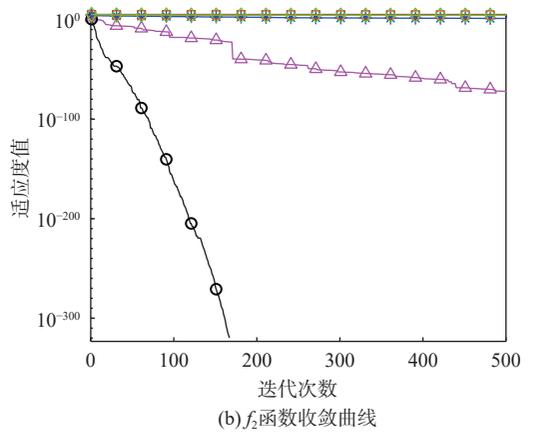
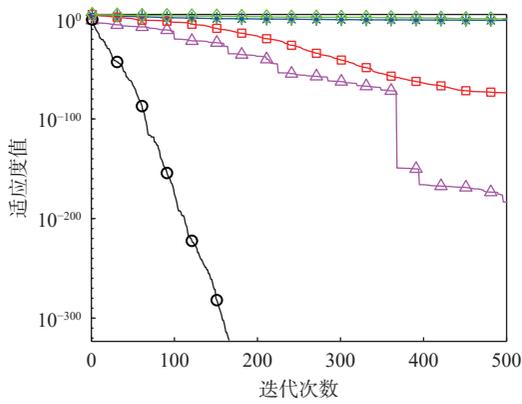
算法	参数设置
PSO	$w=0.9, b_1=1.494 45, b_2=1.494 45$
WOA	$a \in [0,2]$, 并从 2 线性下降
ABC	$\text{limit}=\text{round}(0.6 \dim \cdot \text{SN}), \alpha=1$
SSA	$N_{\text{PD}}=0.2N_{\text{pop}}, N_{\text{SD}}=0.2N_{\text{pop}}, N_{\text{ST}}=0.8$
HSSA	$N_{\text{PD}}=0.2N_{\text{pop}}, N_{\text{SD}}=0.2N_{\text{pop}}, N_{\text{ST}}=0.8$

注: w 为速度惯帧因子, b_1 为自我学习因子, b_2 为群体学习因子, a 为系数向量参数, α 为加速系数最大值, N_{PD} 为发现群体数量, N_{SD} 为警戒者群体数量, N_{pop} 为麻雀种群总体数量, N_{ST} 为安全值。

测得数据如表 3 所示。

由函数收敛曲线和测得实验数据可以直观看出, HSSA 算法在 2 个维度下的寻优性能均表现优秀。其他算法随着维度的增加, 寻优性能皆有不同程度的下降。HSSA 算法在 f_6 函数上不如 WOA 的寻优精度高, 但比其他优化算法的寻优精度高, 收敛速度快。在 f_7 和 f_8 函数上, HSSA 算法寻优性能和 SSA 算法相当, 但是比 SSA 算法的收敛速度快。在其余函数上, HSSA 算法和其他算法相比有较大的优越性。

为了进一步验证 HSSA 算法的性能优势, 实验



—△— SSA —○— HSSA —*— PSO —□— WOA —◇— ABC

图9 函数收敛曲线

Fig. 9 Function convergence curves

采用 Wilcoxon 秩和检验^[21]来展示其与其他4种优化算法的显著性差异。即在显著性水平 $P=5\%$ 的情况下,比较算法之间的性能,结果如表4所示。

由表4可知,通过 HSSA 和表中4中算法的比较,可以发现 P 值中大部分数据小于 5% 。HSSA 算法在8种基准测试函数不同维度的寻优性能皆优于 PSO 和 ABC 算法。WOA 算法在 f_6 函数上和

HSSA 算法存在显著性差异,但其寻优性能强于 HSSA 算法。在其余7种基准测试函数上,HSSA 算法优于 WOA 算法,-表示显著性差异不明显,表示2种算法的寻优性能相当。因此,在 f_7 和 f_8 函数上,HSSA 算法和 SSA 算法的寻优性能相当,在其余6种基准测试函数上,HSSA 算法的寻优性能优于 SSA 算法。

表4 Wilcoxon 秩和检验 P 值
Table 4 P values for Wilcoxon rank-sum test

函数	解维度为30时				解维度为80时			
	PSO	WOA	ABC	SSA	PSO	WOA	ABC	SSA
f_1	1.2118×10^{-12}							
f_2	1.2118×10^{-12}							
f_3	1.2118×10^{-12}							
f_4	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	4.0772×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	6.0658×10^{-11}
f_5	3.0199×10^{-11}							
f_6	3.0199×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	7.3891×10^{-11}	3.3384×10^{-11}	3.0199×10^{-11}	3.0199×10^{-11}	5.5329×10^{-8}
f_7	1.2118×10^{-12}	3.2801×10^{-7}	1.2118×10^{-12}	-	1.2118×10^{-12}	1.4000×10^{-4}	1.2118×10^{-12}	-
f_8	1.2118×10^{-12}	1.9457×10^{-9}	1.2118×10^{-12}	-	1.2118×10^{-12}	1.4552×10^{-4}	1.2118×10^{-12}	-

2.2.2 不同改进算法比较

目前,SSA 算法研究领域已有几种改进算法,本文选取了近期改进效果较好的改进型麻雀算法(improved sparrow search algorithm, ISSA)^[22]的寻优数据进行对比。为保证实验公平性,种群数设为50,最大迭代次数设为500,其余各参数参见文献[23]。在8种基准测试函数上分别进行30次寻优实验,

对比2种算法寻优得到的最优值、平均值和标准差,结果如表5所示。

与 ISSA 算法比较可知,对于 $f_1 \sim f_3$ 函数,HSSA 算法较 ISSA 算法的寻优性能有所提升,并且在 $f_1 \sim f_3$ 函数上的寻优性能提升较大,在 f_6 函数上的寻优性能不如 ISSA 算法,对于 $f_7 \sim f_8$ 函数,HSSA 算法和 ISSA 算法的寻优性能相当。

表5 不同改进 SSA 算法性能比较
Table 5 Performance comparison of different improved SSA

算法	f_1			f_2			f_3			f_4		
	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差
ISSA	0	1.13×10^{-81}	4.42×10^{-81}	0	6.28×10^{-93}	1.49×10^{-92}	0	9.25×10^{-49}	3.20×10^{-48}	7.85×10^{-7}	9.16×10^{-6}	1.44×10^{-5}
HSSA	0	0	0	0	0	0	0	0	0	3.17×10^{-11}	4.37×10^{-6}	6.98×10^{-6}

算法	f_5			f_6			f_7			f_8		
	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差
ISSA	2.11×10^{-5}	3.20×10^{-4}	2.13×10^{-4}	-1.26×10^4	-1.26×10^4	2.70×10^{-1}	0	0	0	0	0	0
HSSA	2.26×10^{-6}	5.04×10^{-5}	4.45×10^{-5}	-1.14×10^4	-8.87×10^3	7.01×10^2	0	0	0	0	0	0

2.2.3 不同改进策略的有效性分析

为了证明上述策略对 SSA 算法的性能提升均有一定的作用,选取单一策略改进的 SSA 算法进行对比,即只引进改进的发现者位置更新策略的 FSSA 算法、只引进改进的加入者位置更新策略的 JSSA 算法、只引进镜像选择策略的 ZSSA 算法和

只引进模拟退火机制的 MSSA 算法。将4种改进算法在8种基准测试函数上的寻优数据 SSA、HSSA 算法做对比,验证各策略的有效性和 HSSA 算法的优势。每个函数测试50次,维度设为30,种群数设为30,迭代次数为500。其余各参数同表2,测得数据如表6所示。

表 6 不同改进策略性能比较
Table 6 Performance comparison of different improvement strategies

算法	f_1			f_2		
	最优值	平均值	标准差	最优值	平均值	标准差
SSA	0	$7.522\ 9 \times 10^{-64}$	$4.120\ 5 \times 10^{-63}$	0	$4.528\ 3 \times 10^{-53}$	$1.826\ 1 \times 10^{-52}$
FSSA	0	0	0	0	0	0
JSSA	0	$1.096\ 8 \times 10^{-69}$	$6.007\ 2 \times 10^{-69}$	$1.865\ 5 \times 10^{-232}$	$5.615\ 2 \times 10^{-73}$	$3.075\ 4 \times 10^{-72}$
ZSSA	0	$1.246\ 9 \times 10^{-75}$	$6.829\ 6 \times 10^{-75}$	0	$7.156\ 2 \times 10^{-61}$	$3.919\ 6 \times 10^{-62}$
MSSA	0	$2.103\ 3 \times 10^{-74}$	$1.152\ 0 \times 10^{-73}$	0	$3.605\ 3 \times 10^{-53}$	$1.974\ 7 \times 10^{-52}$
HSSA	0	0	0	0	0	0

算法	f_3			f_4		
	最优值	平均值	标准差	最优值	平均值	标准差
SSA	0	$2.231\ 4 \times 10^{-34}$	$1.222\ 1 \times 10^{-33}$	$2.728\ 7 \times 10^{-7}$	$1.224\ 3 \times 10^{-4}$	$2.791\ 5 \times 10^{-4}$
FSSA	0	0	0	$6.625\ 7 \times 10^{-8}$	$2.216\ 2 \times 10^{-4}$	$4.181\ 2 \times 10^{-4}$
JSSA	$4.014\ 3 \times 10^{-205}$	$9.687\ 6 \times 10^{-35}$	$5.211\ 4 \times 10^{-33}$	$4.437\ 5 \times 10^{-8}$	$1.154\ 2 \times 10^{-4}$	$2.348\ 0 \times 10^{-4}$
ZSSA	0	$1.369\ 5 \times 10^{-36}$	$7.560\ 0 \times 10^{-36}$	$3.808\ 7 \times 10^{-8}$	$1.502\ 7 \times 10^{-5}$	$3.787\ 4 \times 10^{-5}$
MSSA	0	$1.219\ 6 \times 10^{-39}$	$6.679\ 0 \times 10^{-39}$	$3.269\ 7 \times 10^{-8}$	$7.891\ 0 \times 10^{-5}$	$1.598\ 6 \times 10^{-4}$
HSSA	0	0	0	$3.090\ 3 \times 10^{-9}$	$1.827\ 9 \times 10^{-5}$	$2.874\ 7 \times 10^{-5}$

算法	f_5			f_6		
	最优值	平均值	标准差	最优值	平均值	标准差
SSA	$2.922\ 4 \times 10^{-5}$	$6.817\ 2 \times 10^{-4}$	$5.277\ 4 \times 10^{-4}$	$-8.929\ 0 \times 10^3$	$-7.856\ 3 \times 10^3$	$6.222\ 0 \times 10^2$
FSSA	$1.627\ 1 \times 10^{-5}$	$2.474\ 1 \times 10^{-4}$	$2.211\ 8 \times 10^{-4}$	$-1.021\ 4 \times 10^4$	$-8.565\ 9 \times 10^3$	$7.770\ 4 \times 10^2$
JSSA	$1.756\ 8 \times 10^{-5}$	$4.313\ 1 \times 10^{-4}$	$4.537\ 6 \times 10^{-4}$	$-1.055\ 5 \times 10^4$	$-8.625\ 0 \times 10^3$	$6.170\ 8 \times 10^2$
ZSSA	$8.687\ 5 \times 10^{-6}$	$3.050\ 8 \times 10^{-4}$	$3.168\ 3 \times 10^{-4}$	$-9.504\ 5 \times 10^3$	$-8.498\ 3 \times 10^3$	$8.148\ 2 \times 10^2$
MSSA	$4.384\ 4 \times 10^{-6}$	$4.154\ 1 \times 10^{-4}$	$4.265\ 1 \times 10^{-4}$	$-9.706\ 3 \times 10^3$	$-8.404\ 0 \times 10^3$	$7.898\ 0 \times 10^2$
HSSA	$1.631\ 2 \times 10^{-6}$	$7.270\ 1 \times 10^{-5}$	$6.070\ 1 \times 10^{-5}$	$-1.146\ 3 \times 10^4$	$-8.694\ 6 \times 10^3$	$9.477\ 7 \times 10^2$

算法	f_7			f_8		
	最优值	平均值	标准差	最优值	平均值	标准差
SSA	0	0	0	0	0	0
FSSA	0	0	0	0	0	0
JSSA	0	0	0	0	0	0
ZSSA	0	0	0	0	0	0
MSSA	0	0	0	0	0	0
HSSA	0	0	0	0	0	0

由表 6 数据可知, 在 $f_1 \sim f_3$ 函数的寻优过程中, FSSA 算法中改进的发现者更新策略起决定性作用, 其他策略对于函数的寻优也有一定的辅助作用。在 $f_4 \sim f_6$ 函数的寻优过程中, 所有改进策略均不同程度提高了算法的寻优性能, 其中, 镜像选择

策略在 f_4 函数上的寻优贡献比其他策略大, 改进的加入者更新策略在 f_6 函数上的寻优贡献比其他策略大, 但是多策略融合起来对于算法的寻优性能提升最大。在 $f_7 \sim f_8$ 函数的寻优过程中, 各种改进策略的算法均能找到函数最优值。通过上述数据的对

比可以发现,各改进策略对于算法寻优性能的提升均提供了不同程度的帮助,验证了其有效性。

2.3 HSSA 算法时间复杂度分析

为了验证 HSSA 算法的性能优势不是以时间为代价获得的,采用计算算法时间复杂度^[23]的方式将 HSSA 算法和 SSA 算法进行对比。由文献 [24-25] 可知,SSA 算法的时间复杂度 T 为

$$T = O(D + f(D)) \quad (15)$$

式中: D 为维度; $f(D)$ 为求解目标函数所需要的时间。

假设麻雀的种群数为 N , 设 HSSA 算法种群初始化参数的时间为 ε_1 , 每一维生成 Circle 混沌映射混沌值的时间为 ε_2 , 则 HSSA 算法初始化所需时间为

$$T_1 = O(\varepsilon_1 + N(f(D) + \varepsilon_2)D) \quad (16)$$

SSA 算法发现者比例设为 r_1 , 其中,有 3 个随机数,生成所需时间分别为 ε_3 、 ε_4 、 ε_5 , 每一维位置更新所需时间为 ε_6 。因此,发现者阶段所需时间复杂度为

$$T_2 = O(r_1 N(\varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6)D) \quad (17)$$

SSA 算法加入者比例设为 r_2 , 其中,有 2 个随机数,生成所需时间分别为 ε_7 、 ε_8 , 自适应步长因子 $H_{i,j}$ 的生成所需时间为 ε_9 , 每一维位置更新所需时间为 ε_{10} 。因此,加入者阶段所需时间复杂度为

$$T_3 = O(r_2 N(\varepsilon_7 + \varepsilon_8 + \varepsilon_9 + \varepsilon_{10})D) \quad (18)$$

SSA 算法警戒者的位置更新公式没有改变,故时间复杂度为 T_4 , 与原算法相同,每一维生成镜像维度值所需时间为 ε_{11} 。因此,镜像选择阶段所需时间复杂度为

$$T_5 = O(N\varepsilon_{11}D) \quad (19)$$

模拟退火机制中,随机数生成所需时间为 ε_{12} , 退火温度的更新所需时间为 ε_{13} 。因此,模拟退火阶段所需时间复杂度为

$$T_6 = O(\varepsilon_{12} + \varepsilon_{13}) \quad (20)$$

因此,HSSA 算法总的时间复杂度为

$$T' = T_1 + i_{\max}(T_2 + T_3 + T_4 + T_5 + T_6) = O(D + f(D)) \quad (21)$$

因为 $T' = T$, 所以 HSSA 算法和 SSA 算法的时间复杂度相同。由此证明 HSSA 算法的性能提升并不是通过牺牲时间来实现的。

3 应用分析

ELM 是一种结构简单、参数设置少、泛化能力强且学习速度快的机器学习算法,做数据集的分类预测效果相对较好。但是,由于 ELM 的输入权值

和隐层偏差是随机赋值的,造成 ELM 的分类预测性能也会有随机性,且面对复杂样本和多维度的数据分类预测效果较差。因此,实验使用 HSSA 算法和各种优化算法对 ELM 的输入权值和隐层偏差进行参数优化,使其寻得最优值,进而提高 ELM 的分类预测性能。

为比较 HSSA 算法和其他算法对 ELM 的优化效果,实验截取了 UCI 数据库^[26]中的部分人体表面肌电信号^[27]数据集作为训练集和测试集进行分类预测。该数据集中动作类别分为 5 类,每类动作包含 130 个样本数据,动作类别标签分组如表 7 所示。

表 7 动作类别标签

类别标签	人体动作
1	跳跃
2	跑步
3	下蹲
4	站立
5	行走

目前,肌电信号在助力和康复外骨骼^[28]领域的研究价值重大。肌电信号的产生超前于人体运动动作,通过对肌电信号的预测识别可以判断出人体的运动意图,进而控制外骨骼做出相应动作来实现助力或帮助病人完成康复训练。因此,此项应用对肌电数据的分类预测精度要求很高。但是由于从人体表面肌电信号提取的肌电数据复杂性较高、维数较大且不稳定,ELM 对其分类预测难度较大,精度较低。面对该复杂数据,通过 HSSA 等算法对 ELM 进行优化,来提高对此数据集的分类预测精度。通过对比各种算法优化 ELM 的实验结果更能体现出 HSSA 算法的优越性。实验设置训练集样本为 500 个,测试集样本为 150 个,每个样本的特征维数为 8 维,ELM 隐藏层数为 20 层。HSSA 算法种群数设为 30,解空间上下限设为 $[-100,100]$,迭代次数为 500 次。每个算法分别进行 10 次实验,分类预测精度结果如图 10 所示。

分类预测结果由表 8 可知,ELM 对于该数据集的分类预测结果并不理想,加入优化算法后分类预测精度提升明显。其中,经 HSSA 算法优化后的 ELM 在此数据集上的分类预测精度提升了将近 48%,明显高于其他算法。通过此应用说明 HSSA 算法具有实际应用能力,提高了对此肌电数据集的分类预测精度,也验证了 HSSA 算法良好的寻优性能。

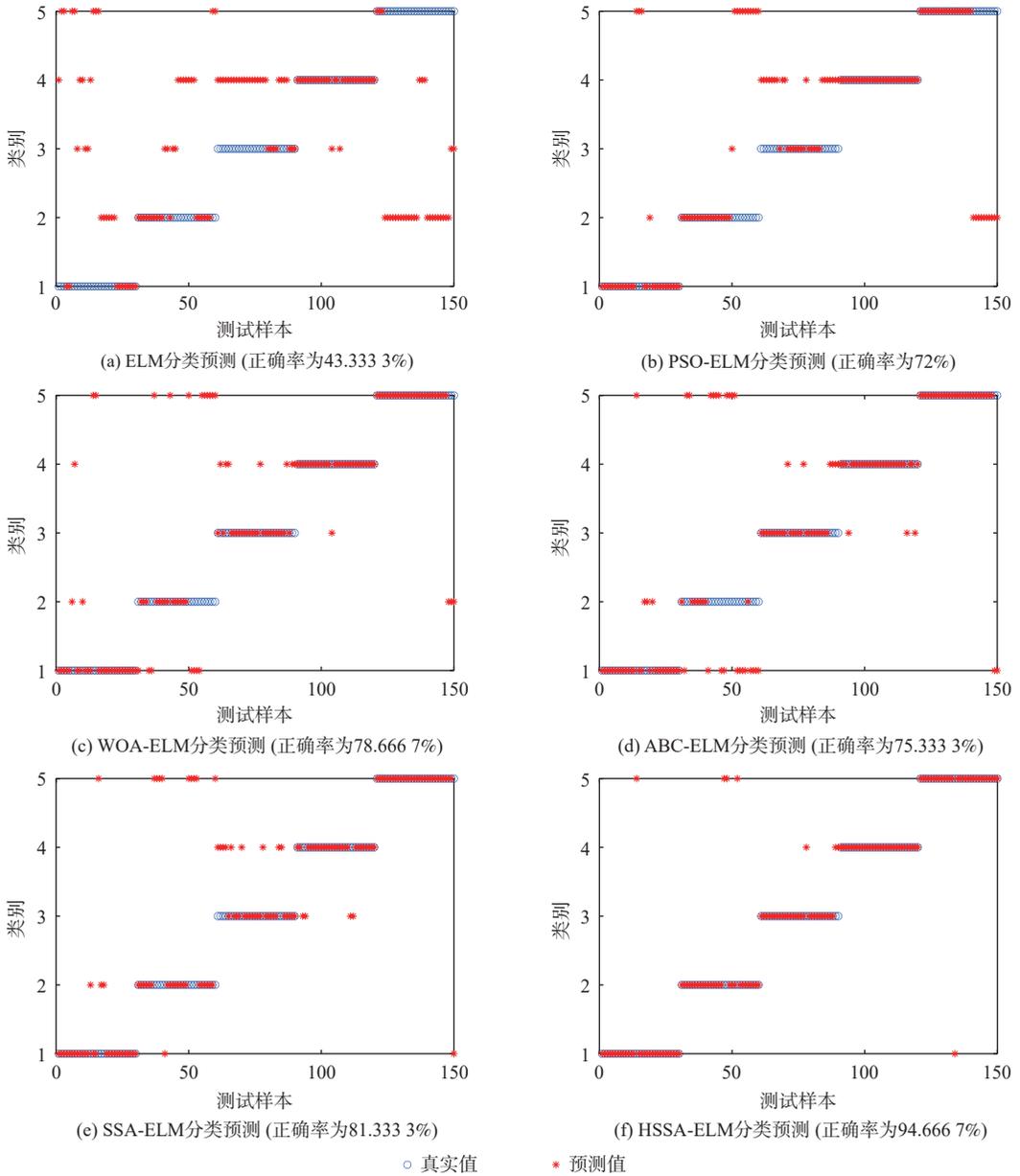


图 10 分类预测结果

Fig. 10 Classification and prediction results

表 8 算法分类预测精度比较
 Table 8 Comparison of classification and prediction accuracy of algorithms %

算法类别	最优预测精度	平均预测精度
ELM	54.67	42.67
PSO-ELM	78.00	70.89
WOA-ELM	82.33	77.56
ABC-ELM	76.67	72.42
SSA-ELM	84.00	80.17
HSSA-ELM	96.00	90.87

4 结 论

1) 通过实验仿真测试, 在 8 种基准测试函数上对各个算法进行对比, 可以发现 HSSA 算法的寻优

精度和收敛速度相较于 SSA 算法有较大提升, 与其他智能优化算法相比同样优势明显。

2) 通过对改进策略的分析和测得数据的对比可知, 各个策略均一定程度上提升了 SSA 算法的性能, 所有策略融合后, 其性能可以达到最大化, 并且各个策略均没有增加算法的时间复杂度。

3) 通过对 ELM 的优化结果进行对比可知, 经 HSSA 算法优化的 ELM 性能提升明显, 提高了对肌电数据集的分类预测精度, 证明了 HSSA 算法具有良好的实际应用能力。

之后工作将把 HSSA 算法应用于模式识别、路径规划和图像分割等领域, 进一步验证其工程应用性能。

参考文献 (References)

- [1] XUE J K, SHEN B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22-34.
- [2] WANG D S, TAN D P, LIU L. Particle swarm optimization algorithm: An overview[J]. *Soft Computing*, 2018, 22(2): 387-408.
- [3] KARABOGA D, OZTURK C. A novel clustering approach: Artificial bee colony (ABC) algorithm[J]. *Applied Soft Computing*, 2011, 11(1): 652-657.
- [4] 吕鑫, 慕晓冬, 张钧. 基于改进麻雀搜索算法的多阈值图像分割[J]. *系统工程与电子技术*, 2021, 43(2): 318-327.
- LV X, MU X D, ZHANG J. Multi-threshold image segmentation based on improved sparrow search algorithm[J]. *Systems Engineering and Electronics*, 2021, 43(2): 318-327(in Chinese).
- [5] 汤安迪, 韩统, 徐登武, 等. 基于混沌麻雀搜索算法的无人机航迹规划方法[J]. *计算机应用*, 2021, 41(7): 2128-2136.
- TANG A D, HAN T, XU D W, et al. Path planning method of unmanned aerial vehicle based on chaos sparrow search algorithm[J]. *Journal of Computer Applications*, 2021, 41(7): 2128-2136(in Chinese).
- [6] 黄敬宇. 融合t分布和Tent混沌映射的麻雀搜索算法研究[D]. 兰州: 兰州大学, 2021.
- HUANG J Y. Research on sparrow search algorithm based on fusion of t distribution and tent chaotic mapping[D]. Lanzhou: Lanzhou University, 2021(in Chinese).
- [7] 徐健, 陈倩倩, 刘秀平. 基于交叉运算的人工蜂群优化BP神经网络的脑电信号分类[J]. *激光与光电子学进展*, 2020, 57(21): 244-253.
- XU J, CHEN Q Q, LIU X P. Classification of electroencephalography based on BP neural network optimized by crossover operation of artificial bee colonies[J]. *Laser & Optoelectronics Progress*, 2020, 57(21): 244-253(in Chinese).
- [8] 刘栋, 魏霞, 王维庆, 等. 基于SSA-ELM的短期风电功率预测[J]. *智慧电力*, 2021, 49(6): 53-59.
- LIU D, WEI X, WANG W Q, et al. Short-term wind power prediction based on SSA-ELM[J]. *Smart Power*, 2021, 49(6): 53-59(in Chinese).
- [9] 蒋艳会, 李峰. 基于混沌粒子群算法的多阈值图像分割[J]. *计算机工程与应用*, 2010, 46(10): 175-176.
- JIANG Y H, LI F. Multi-threshold method of image segmentation based on chaotic particle swarm optimization algorithm[J]. *Computer Engineering and Applications*, 2010, 46(10): 175-176(in Chinese).
- [10] HUANG G B, WANG D H, LAN Y. Extreme learning machines: a survey[J]. *International Journal of Machine Learning and Cybernetics*, 2011, 2(2): 107-122.
- [11] 吕鑫, 慕晓冬, 张钧, 等. 混沌麻雀搜索优化算法[J]. *北京航空航天大学学报*, 2021, 47(8): 1712-1720.
- LYU X, MU X D, ZHANG J, et al. Chaos sparrow search optimization algorithm[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2021, 47(8): 1712-1720(in Chinese).
- [12] HERBADJI D, DEROUICHE N, BELMEGUENAI A, et al. A tweakable image encryption algorithm using an improved logistic chaotic map[J]. *Traitement Du Signal*, 2019, 36(5): 407-417.
- [13] ARORA S, ANAND P. Chaotic grasshopper optimization algorithm for global optimization[J]. *Neural Computing and Applications*, 2019, 31(8): 4385-4405.
- [14] 张达敏, 徐航, 王依柔, 等. 嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法[J]. *控制与决策*, 2021, 36(5): 1173-1180.
- ZHANG D M, XU H, WANG Y R, et al. Whale optimization algorithm for embedded Circle mapping and onedimensional oppositional learning based small hole imaging[J]. *Control and Decision*, 2021, 36(5): 1173-1180(in Chinese).
- [15] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems[J]. *Advances in Engineering Software*, 2017, 114: 163-191.
- [16] LI J N, LE M L. Improved whale optimization algorithm based on mirror selection[J]. *Transactions of Nanjing University of Aeronautics and Astronautics*, 2020, 37(S): 115-123.
- [17] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence[C]//International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Piscataway: IEEE Press, 2006: 695-701.
- [18] SUMAN B, KUMAR P. A survey of simulated annealing as a tool for single and multiobjective optimization[J]. *Journal of the Operational Research Society*, 2006, 57(10): 1143-1160.
- [19] SHADRAVAN S, NAJI H R, BARDSIRI V K. The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems[J]. *Engineering Applications of Artificial Intelligence*, 2019, 80: 20-34.
- [20] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [21] 黄海燕, 彭虎, 邓长寿, 等. 均匀局部搜索和高斯变异的布谷鸟搜索算法[J]. *小型微型计算机系统*, 2018, 39(7): 1451-1458.
- HUANG H Y, PENG H, DENG C S, et al. Cuckoo search algorithm of uniform local search and Gauss mutation[J]. *Journal of Chinese Computer Systems*, 2018, 39(7): 1451-1458(in Chinese).
- [22] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用[J]. *控制与决策*, 2022, 37(1): 87-96.
- FU H, LIU H. Improved sparrow search algorithm with multi-strategy integration and its application[J]. *Control and Decision*, 2022, 37(1): 87-96(in Chinese).
- [23] 刘小娟, 王联国. 一种基于差分进化的正弦余弦算法[J]. *工程科学学报*, 2020, 42(12): 1674-1684.
- LIU X J, WANG L G. A sine cosine algorithm based on differential evolution[J]. *Chinese Journal of Engineering*, 2020, 42(12): 1674-1684(in Chinese).
- [24] 唐延强, 李成海, 宋亚飞, 等. 自适应变异麻雀搜索优化算法[J]. *北京航空航天大学学报*, 2023, 49(3): 681-692.
- TANG Y Q, LI C H, SOMG Y F, et al. Adaptive mutation sparrow search optimization algorithm[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2023, 49(3): 681-692(in Chinese).
- [25] 毛清华, 张强. 融合柯西变异和反向学习的改进麻雀算法[J]. *计算机科学与探索*, 2021, 15(6): 1155-1164.
- MAO Q H, ZHANG Q. Improved sparrow algorithm combining Cauchy mutation and opposition-based learning[J]. *Journal of Fron-*

- tiers of Computer Science and Technology, 2021, 15(6): 1155-1164 (in Chinese).
- [26] 张强, 李盼池, 王梅. 基于自适应进化策略的人工蜂群优化算法[J]. 电子科技大学学报, 2019, 48(4): 560-566.
ZHANG Q, LI P C, WANG M. Artificial bee colony optimization algorithm based on adaptive evolution strategy[J]. Journal of University of Electronic Science and Technology of China, 2019, 48(4): 560-566(in Chinese).
- [27] 佟丽娜, 侯增广, 彭亮, 等. 基于多路sEMG时序分析的人体运动模式识别方法[J]. 自动化学报, 2014, 40(5): 810-821.
TONG L N, HOU Z G, PENG L, et al. Multi-channel sEMG time series analysis based human motion recognition method[J]. Acta Automatica Sinica, 2014, 40(5): 810-821(in Chinese).
- [28] 刘冰, 李宁, 于鹏, 等. 上肢康复外骨骼机器人控制方法进展研究[J]. 电子科技大学学报, 2020, 49(5): 643-651.
LIU B, LI N, YU P, et al. Research on the control methods of upper limb rehabilitation exoskeleton robot[J]. Journal of University of Electronic Science and Technology of China, 2020, 49(5): 643-651(in Chinese).

Improvement and application of hybrid strategy-based sparrow search algorithm

SONG Liqin¹, CHEN Wenjie^{1,*}, CHEN Weihai², LIN Yan², SUN Xiantao¹

(1. School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China;

2. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China)

Abstract: Aiming at solving the problems in the original sparrow search algorithm (SSA), such as low search accuracy, weak global search ability, slow convergence speed and easy tendency to fall into local optimum, a hybrid strategy-based sparrow search algorithm (HSSA) is proposed. First, an improved Circle chaotic map was used to initialize the population and increase the diversity of the population. Then, the salp swarm algorithm was integrated into the search formula of the discoverers to enhance its global search ability and scope in the early stage of iteration, and an adaptive step size factor was introduced into the search formula of the participants to improve the local search ability and convergence speed of the algorithm. Next, the mirror selection mechanism was applied to boost the individual quality after each iteration, thereby improving the search accuracy and speed of the algorithm. Finally, a simulated annealing mechanism was added to the location update, thus enabling the algorithm effectively to jump out of local optimum. The test results of eight functions show that the HSSA has better optimization performance than SSA. By combining the improved algorithm and the extreme learning machine, the classification and prediction accuracy of human surface electromyogram signal data increased from 80.17% to 90.87%, which proves the feasibility and good performance of the improved algorithm.

Keywords: sparrow search algorithm; Circle chaotic mapping; salp swarm algorithm; mirror selection; adaptive step factor; simulated annealing mechanism; extreme learning machine