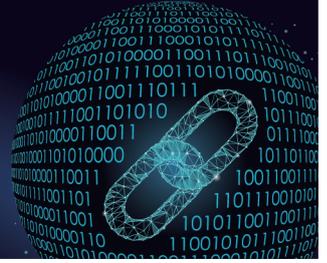




中国科技期刊卓越行动计划项目入选期刊

# 控制与决策

CONTROL AND DECISION



## 多策略混合的改进沙猫群优化算法及其应用

回立川, 于千皓

引用本文:

回立川,于千皓. 多策略混合的改进沙猫群优化算法及其应用[J]. *控制与决策*, 2024, 39(10): 3216–3224.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.1095>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### [基于自适应正态云模型的灰狼优化算法](#)

Grey wolf optimization algorithm based on adaptive normal cloud model

*控制与决策*. 2021, 36(10): 2562–2568 <https://doi.org/10.13195/j.kzyjc.2020.0233>

#### [嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法](#)

Whale optimization algorithm for embedded Circle mapping and one–dimensional oppositional learning based small hole imaging

*控制与决策*. 2021, 36(5): 1173–1180 <https://doi.org/10.13195/j.kzyjc.2019.1362>

#### [一种自适应拟牛顿–状态转移混合智能优化算法及应用](#)

A hybrid state transition optimization algorithm based on adaptive quasi–newton method and its application

*控制与决策*. 2021, 36(10): 2451–2458 <https://doi.org/10.13195/j.kzyjc.2020.0214>

#### [具有重组学习和混合变异的动态多种群粒子群优化算法](#)

Dynamic multi–population particle swarm optimization algorithm with recombined learning and hybrid mutation

*控制与决策*. 2021, 36(12): 2871–2880 <https://doi.org/10.13195/j.kzyjc.2020.0898>

#### [混合柯西变异和均匀分布的蝗虫优化算法](#)

Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm

*控制与决策*. 2021, 36(7): 1558–1568 <https://doi.org/10.13195/j.kzyjc.2019.1609>

# 多策略混合的改进沙猫群优化算法及其应用

回立川<sup>†</sup>, 于千皓

(辽宁工程技术大学 电气与控制工程学院, 辽宁 葫芦岛 125105)

**摘要:** 为了增加基本沙猫群算法的搜索效率、收敛效率, 增加种群的多样性, 增强全局搜索能力, 提出一种多策略混合的改进沙猫群算法 (improved sand cat swarm optimization algorithm, IMSCSO). 利用 Sine 映射进行初始化以获得分布更加均匀的种群; 在攻击行为中根据个体适应度的大小对攻击区间进行区域划分, 减小攻击范围, 增加搜索效率; 将向量的线性变换引入到搜寻行为中, 通过对系数矩阵的设计增加收敛效率; 利用聚集度圆增加算法跳出局部最优的能力; 通过引入生存机制的黄金正弦策略增强其局部开发的能力. 使用 12 个基本测试函数对改进后的算法进行测试, 经过 Wilcoxon 秩和检测、时间复杂度分析、李亚普诺夫稳定性分析验证改进算法的优越性. 最后使用改进沙猫群算法对 SVM 参数进行优化, 并应用于轴承故障检测, 证明算法在实际应用中的有效性.

**关键词:** 沙猫群算法; Sine 映射; 区域划分; 线性变换; 聚集度圆; 黄金正弦策略; 生存机制

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2023.1095

引用格式: 回立川, 于千皓. 多策略混合的改进沙猫群优化算法及其应用 [J]. 控制与决策, 2024, 39(10): 3216-3224.

## Improved sand cat swarm optimization algorithm based on multi-strategy mixing and its application

HUI Li-chuan<sup>†</sup>, YU Qian-hao

(Faculty of Electrical and Control Engineering, Liaoning Technical University, Huludao 125105, China)

**Abstract:** In order to improve the search efficiency and convergence efficiency of the basic sand cat swarm algorithm, increase the diversity of the population, and enhance global search capabilities, an improved sand cat swarm optimization algorithm (IMSCSO) with multi-strategy mixing is proposed. Sine mapping is used for initialization to obtain a more evenly distributed population. In the attack behavior, the attack interval is divided according to the size of the individual adaptability to reduce the attack range and increase the search efficiency. The linear transformation of vectors is introduced into the search behavior, and the convergence efficiency is increased by the design of the coefficient matrix. The aggregation circle is used to increase the ability of the algorithm to jump out of the local optimum. Its ability of developing locally is enhanced with golden sinusoidal strategies which cite survival strategies. The improved algorithm is tested by 12 basic test functions, and the superiority of the improved strategy is verified by Wilcoxon rank sum detection, time complexity analysis, and Lyapunov stability analysis. Finally, the improved sand cat swarm algorithm is used to optimize the SVM parameters and applied to bearing fault detection, which proves the effectiveness of the algorithm in practical applications.

**Keywords:** sand cat swarm algorithm; Sine mapping; regional division; linear transformation; aggregation circle; golden sinusoidal strategy; survival mechanisms

## 0 引言

在工业、工程及科学领域, 经常使用智能优化算法计算最优解, 例如对支持向量机的惩罚因子和核参数进行优化, 增加其分类的准确性. 在过去几十年里, 发展了大量受自然过程启发的元启发式算法, 如粒子群算法、灰狼算法、麻雀算法、蚁群算法等. 这些算法

已被应用于许多大规模优化问题.

大多数智能优化算法都存在依赖初始解、迭代后期种群多样性降低、容易陷入局部最优等问题. 国内外学者对其进行了相关改进研究, 通过引入混沌映射、种群精英化、反向学习等方法解决初始解的问题, 引入莱维飞行策略增加种群的多样性, 引入正余弦策

收稿日期: 2023-08-02; 录用日期: 2023-10-20.

基金项目: 辽宁省教育厅辽宁省高等学校基本科研项目 (LJ2017QL009, LJKMZ20220675).

责任编辑: 陈家伟.

<sup>†</sup>通讯作者. E-mail: 104675117@qq.com.

略增加寻优能力等. 文献[1]将相似度与聚集度的概念加入到算法中,当聚集度偏高时会对个体进行重新赋值,以此增加种群的多样性. 文献[2]将黄金正弦策略引入尊海鞘算法领导者的位置更新中,加强算法的全局探索能力,防止算法发生过早收敛的现象. 文献[3]使用 Sine 混沌映射初始化种群,通过混沌的遍历性和随机性解决种群分布不均匀、搜索空间不足等问题.

沙猫群优化算法由 Seyyedabbasi 等<sup>[4]</sup>于 2023 年提出,算法攻击行为中的攻击方向由  $0^\circ \sim 360^\circ$  之间的一个随机角度进行确定,然而大范围的随机角度具有一定的盲目性,会大大降低寻优效率,在其搜寻行为中,以整体适应度作为个体好坏的衡量标准不能完全正确反映出个体在空间中位置的优劣关系. 另一方面,对于在某一维度已收敛于最优位置的坐标,搜寻行为会使其在最优解附近发生震荡,导致收敛效率降低,并且基本算法也同时存在着依赖初始解、迭代后期种群多样性降低等问题. 文献[5]在搜寻行为中加入三角游走策略,增加沙猫群算法的搜索范围,在攻击行为中引入莱维飞行,增加种群的多样性,使沙猫能找到更好的位置,提高了算法的优化性能. 文献[6]建立了伪对立和伪反射机制,以此增加算法的优化效率,提高全局搜索能力,同时提出了一种非周期调整机制,均衡算法的开发能力,提高其收敛效率. 但是迭代过程中的振荡现象依然存在,过快的收敛速度也会导致搜索空间迅速减小,使算法更容易陷入局部最优.

本文通过引入 Sine 映射增加初始种群的随机性,使其分布更加均匀. 根据个体适应度对攻击区间进行分块处理,增加攻击效率,并通过线性变换实现个体位置更新. 在增加基本沙猫群算法收敛性能的同时,利用对系数矩阵的设计消除迭代过程中不同维度之间的影响. 对于个体中已经收敛于当前最优位置的维度,在迭代过程中使其保持不变,消除震荡现象,其他维度进行正常迭代. 通过聚集度圆对搜索空间的缩小进程进行人为干预,增加空间利用率,提高其跳出局部最优值的能力. 对于生存率较低的个体使用黄金正弦策略进行迭代,增强其局部开发的能力. 采用测试函数测试、李雅普诺夫稳定性分析、Wilcoxon 秩和检测来分析改进算法在时间复杂度不变前提下的收敛性能和寻优性能.

### 1 基本沙猫群算法

沙猫群算法是一种模仿自然界中沙猫生存行为的智能优化算法,主要引用沙猫的攻击行为和搜寻行

为,算法参数设置如下:

$$r_G = S_M - \left( \frac{S_M \times t}{t_{\max}} \right), \quad (1)$$

$$R = 2 \times r_G \times \text{rand} - r_G, \quad (2)$$

$$r = r_G \times \text{rand}. \quad (3)$$

其中:  $S_M$  取 2,  $t$  为当前迭代次数,  $t_{\max}$  为最大迭代次数,  $r_G$  随着迭代次数的增加由 2 线性递减至 0,  $R$  为选择使用搜索行为或攻击行为的参数,  $r$  为每只沙猫的灵敏度范围,  $\text{rand}$  为  $0 \sim 1$  之间的随机数.

攻击行为的数学模型如下所示:

$$P_r = |\text{rand} \times P_{bc}^t - P_c^t|, \quad (4)$$

$$P_c^{t+1} = P_{bc}^t - r \times P_r \times \cos(\theta). \quad (5)$$

其中:  $P_r$  为最优个体附近的一个随机个体,  $P_{bc}^t$  为当前最优个体,  $P_c^t$  为当前迭代个体,  $\theta$  为  $0^\circ \sim 360^\circ$  之间的随机角度.

搜寻行为的数学模型如下所示:

$$P_c^{t+1} = r \times (P_{bc}^t - \text{rand} \times P_c^t). \quad (6)$$

式(6)为搜寻行为的迭代公式. 新位置在最优个体与当前迭代个体之间,其中灵敏度  $r$  保证了搜寻行为的收敛效果和复杂度.

#### 1.1 Sine 混沌映射

通过伪随机函数难以在变量空间内生成均匀分布的种群,为了增加种群的多样性,使其分布更加均匀,引入 Sine 混沌映射,有

$$P_c^0(i+1) = \mu \sin[\pi \times P_c^0(i)]. \quad (7)$$

其中:  $\mu$  取 0.99,  $i$  为当前迭代个体序号.

图 1 给出了两维变量分别经过 Sine 映射初始化

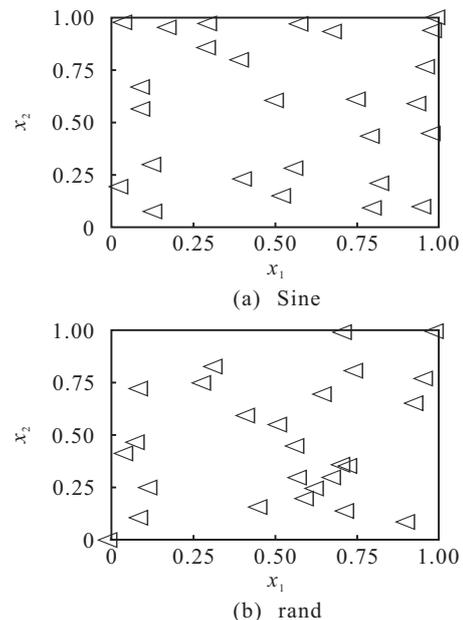


图 1 种群分布图

和rand函数初始化后种群分布情况.由图1可知,经Sine映射初始化后的种群没有出现重叠现象,分布更加均匀,种群多样性得到了提升.

## 1.2 攻击区域划分

在基本的沙猫群算法中,假设沙猫的攻击范围是一个圆,这样攻击方向便可以通过圆上的角度 $\theta$ 确定,其中 $\theta$ 为 $0^\circ \sim 360^\circ$ 之间的随机角度.为了增加攻击行为的搜索效率,对圆形攻击区域进行分块处理.本文以全部初始种群中的个体为基点,如图2所示,在其左右各生成一个区间. $P_{c1}$ 是初始种群中的个体,每个个体都对应一个攻击区间,每次迭代在其中一个区间内选择随机角度执行攻击行为,区间大小由当前个体适应度与当前最优适应度之间的差值决定.以图中个体为例, $P_{c1}(5)$ 的适应度较低,差值较大,因此攻击范围较大,以此增强全局搜索能力. $P_{c1}(6)$ 适应度较高,差值较小,攻击范围变小,局部搜索能力得到增强.

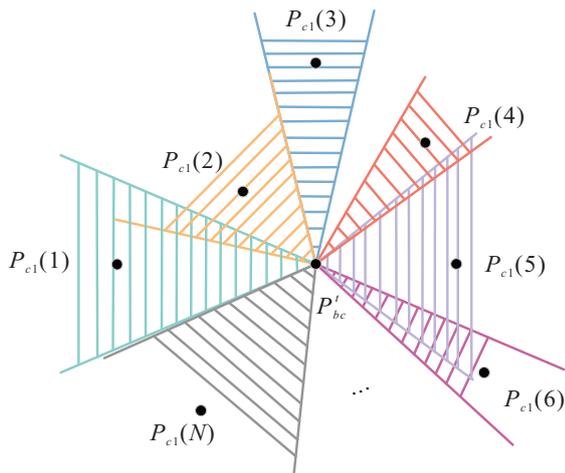


图2 区间分配示意图

具体步骤如下所示.

step 1: 依照适应度的高低对初始种群中的个体进行排序,适应度由高到低依次排列,并生成对应种群 $P_{c1}$ .

step 2: 在 $1 \sim N$ 之间生成随机整数 $s$ ,使用 $s$ 对排序后的个体进行调用,选择最终迭代角度, $N$ 为最大种群个数.

step 3: 将 $(P_r - P_{bc}^t)$ 设为基向量,利用下式给每个个体赋予角度值:

$$\theta_1(s) = a \cos \frac{(P_r - P_{bc}^t) \times [P_{c1}(s) - P_{bc}^t]}{|P_r - P_{bc}^t| \times |P_{c1}(s) - P_{bc}^t|}. \quad (8)$$

step 4: 利用下式计算攻击角度增益 $\theta_2$ :

$$\theta_2(s) = (\text{rand} - 0.5) \times 2 \times \pi \times z(s). \quad (9)$$

增益的最大范围由 $z$ 决定,有

$$z(s) = \frac{\text{fitness}(s) - \text{fitness}_{\text{best}}}{\sum_{i=1}^N [\text{fitness}(i) - \text{fitness}_{\text{best}}]}, \quad (10)$$

其中 $\text{fitness}_{\text{best}}$ 为当前最优适应度.适应度差值越小,最大增益越小.最后由下式确定最终迭代角度 $\theta$ :

$$\theta(s) = \theta_1(s) + \theta_2(s). \quad (11)$$

step 5: 将 $\theta$ 代入式(5)进行位置更新.

## 1.3 改进的搜寻行为

### 1.3.1 线性变换

在基本算法的搜寻机制中,若某一维度的适应度在整体适应度中所占比例较大,则该维度上坐标的优劣很大程度上决定了个体的优劣.这会导致对个体位置好坏的误判,并且原有搜索机制产生的振荡现象会导致搜寻行为的收敛效率降低.为此,将向量线性变换的过程引入到算法的搜索机制中,并对系数矩阵进行设计.改进步骤如下所示.

step 1: 对当前迭代个体 $P_c^t$ 和当前最差个体 $P_{wc}^t$ 的坐标进行处理,保留当前迭代维度的位置坐标,其他维度的坐标分别用最优个体 $P_{bc}^t$ 对应维度上的坐标代替.经坐标变换后生成新个体 $P'_{bc}, P'_c, P'_{wc}$ .

step 2: 计算处理后个体的适应度,并根据适应度的大小重新排出 $P'_{bc}, P'_c, P'_{wc}$ .

step 3: 计算 $P'_{bc}, P'_c$ 在迭代维度上的权值 $w_1, w_2$ ,有

$$e_1 = f(P'_{wc}) - f(P'_{bc}), \quad (12)$$

$$e_2 = f(P'_{wc}) - f(P'_c), \quad (13)$$

$$w_1 = \frac{e_1}{e_1 + e_2}, \quad (14)$$

$$w_2 = 1 - w_1. \quad (15)$$

step 4: 通过下式进行位置更新:

$$P_c^{t+1}(i, j) = w_1 \times P'_{bc}(j) + w_2 \times P'_c(j), \quad (16)$$

其中 $j$ 为当前迭代维度.

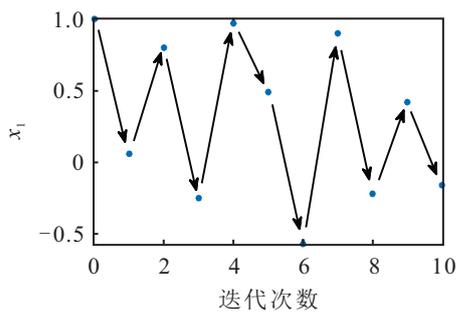
下面对改进前后的搜寻行为进行测试.设 $F(x) = 100(x_1 - 1)^2 + (x_2 - 1)^2$ ,求解其最小值,其中 $x_1, x_2 \in [-1, 3]$ .易知全局最优位置为 $(1, 1)$ ,全局最差位置为 $(-1, -1)$ 、 $(-1, 3)$ 、 $(3, -1)$ 和 $(3, 3)$ .这里设 $(1, 1)$ 为当前最优位置, $(-1, -1)$ 为当前最差位置, $(1, -0.9)$ 为当前迭代个体的坐标,迭代次数为10.此时个体一维坐标 $x_1$ 已经收敛于当前最优位置.表1、表2为改进前后搜寻行为的迭代结果,其中 $l$ 为迭代个体与最优个体之间的欧氏距离.由表1可知,第5次和第9次迭代后,迭代个体与当前最优个体之间的距离分别为0.98和0.82.显然在空间中第9次迭代结果更加接

表1 基本搜寻行为

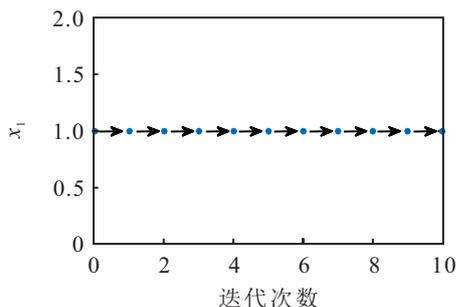
迭代次数	函数值	一维坐标	二维坐标	$l$
0	3.61	1	-0.9	1.9
1	88.58	0.059	0.70	0.99
2	4.22	0.80	0.77	0.30
3	158	-0.25	-0.12	1.68
4	0.11	0.97	1.03	0.042
5	27.19	0.49	0.16	0.98
6	249	-0.5	-0.61	2.25
7	0.96	0.90	0.82	0.21
8	149	-0.22	-0.22	1.73
9	34.45	0.42	0.42	0.82
10	135	-0.16	-0.20	1.67

表2 改进搜寻行为

迭代次数	函数值	一维坐标	二维坐标	$l$
0	3.61	1	-0.9	1.9
1	0.028	1	0.83	0.17
2	0.0071	1	0.916	0.084
3	0.0018	1	0.958	0.042
4	4.41e-04	1	0.979	0.021
5	1.10e-04	1	0.9895	0.0105
6	2.76e-05	1	0.99475	0.00525
7	6.89e-06	1	0.997375	0.002625
8	1.72e-06	1	0.998688	0.001312
9	4.31e-07	1	0.9993438	0.0006562
10	1.08e-07	1	0.9996719	0.0003281



(a) 表1对应个体



(b) 表2对应个体

图3 一维坐标变化曲线

近最优位置. 但第5次迭代的函数值更小, 适应度更大. 这是因为第1维的函数值在整体函数值中所占比例较大, 过多地影响了整体适应度的比较. 改进后的搜寻行为通过坐标变换将单个维度的坐标剥离出来进行比较计算, 使迭代结果更加精准. 图3(a)、(b)分别为表1、表2个体使用改进前后搜寻行为进行迭代时, 一维坐标的变化情况. 由图3(a)可知, 已收敛于最优位置的坐标随着迭代的进行在当前最优位置附近震荡. 而图3(b)改进后的搜寻行为消除了振荡现象, 同时其他维度继续搜寻, 实现全局最优的高效搜索.

1.3.2 聚集度圆

搜索空间过快减小会使种群的多样性降低, 容易发生聚集现象, 导致算法陷入局部最优. 本文将聚集度圆引入算法的搜索机制中, 并受逐维小孔成像反向学习策略<sup>[7]</sup>启发, 以聚集度圆的圆周为基点生成迭代个体的反向点, 最后在个体及其反向点中选择适应度较高的点作为迭代结果, 以此来增加空间利用率, 减小算法陷入局部最优的概率, 加强其跳出局部最优的能力. 聚集度公式<sup>[8]</sup>如下:

$$G(i) = \frac{1}{N} \sum_{i=1}^N \left[ 1 - \frac{d(i)}{d_{\max}} \right], \quad (17)$$

$$d(i) = \|P_c^t(i) - P_{bc}^t\|_2, \quad (18)$$

其中 $d(i)$ 为当前迭代个体与当前最优个体之间的欧氏距离.

为了更好地体现出种群与空间的相对关系, 将式

(17)中的 $d_{\max}$ 替换为 $\sqrt{\sum_{j=1}^{\dim} [\text{ub}(j) - \text{lb}(j)]^2}$ . 其中:  $\dim$ 为种群维度,  $\text{ub}(j)$ 和 $\text{lb}(j)$ 为函数在第 $j$ 维上的上下限. 由此可以得出改进后聚集度公式为

$$G = \frac{1}{N} \sum_{i=1}^N \left\{ 1 - \frac{d(i)}{\sqrt{\sum_{j=1}^{\dim} [\text{ub}(j) - \text{lb}(j)]^2}} \right\}. \quad (19)$$

设 $\bar{d}$ 为全部个体到最优个体之间的平均距离,  $\bar{d}$ 与 $G$ 的关系由改进后的聚集度公式推出, 如下所示:

$$\bar{d} = (1 - G) \sqrt{\sum_{j=1}^{\dim} [\text{bu}(j) - \text{lb}(j)]^2}. \quad (20)$$

将以 $P_{bc}^t$ 为圆心 $\bar{d}$ 为半径的圆称为聚集度圆, 通过对聚集度 $G$ 的控制影响聚集度圆的半径 $\bar{d}$ , 进而对搜索空间的大小进行调节. 最后利用下式生成个体 $P_{c2}$ :

$$P_{c2} = \frac{[2\bar{d} - d(i)][P_c^t(i) - P_{bc}^t]}{d(i)} + P_{bc}^t. \quad (21)$$

比较 $P_c^t$ 与 $P_{c2}$ 的适应度, 选择适应度更高的个体作为本次迭代的结果, 以此增加空间利用率.

图4为式(21)示意图.  $P_c^t(1)$ 、 $P_c^t(2)$ 为当前迭代个体;  $d(1)$ 、 $d(2)$ 分别为 $P_c^t(1)$ 、 $P_c^t(2)$ 到当前最优个体之间的距离,  $P_{c2}(1)$ 、 $P_{c2}(2)$ 为通过式(21)生成的对称点. 比较迭代个体与其对应对称点的适应度, 选择适应度更好的点作为迭代结果. 经测试, 将 $G$ 设置为分段函数, 有

$$G = \begin{cases} 0.2, & 0 \leq t < t_1; \\ 0.4, & t_1 \leq t < t_2; \\ 0.6, & t_2 \leq t < t_3; \\ 0.8, & t_3 \leq t < t_4; \\ 1, & t_4 \leq t \leq t_{\max}. \end{cases} \quad (22)$$

其中:  $t_{\max}$ 为最大迭代次数,  $t_1 \sim t_4$ 为 $0 \sim t_{\max}$ 之间的数.

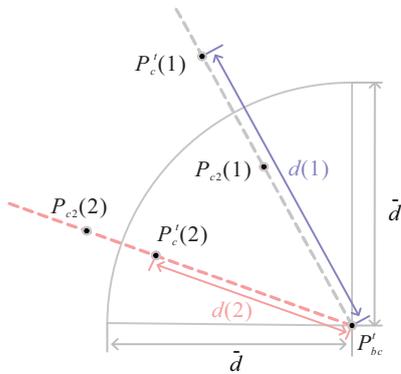


图4 式(21)示意图

### 1.4 引入生存机制的黄金正弦策略

#### 1.4.1 黄金正弦策略

黄金正弦算法<sup>[9]</sup>是2017年提出的一种新型智能优化算法, 其灵感来自于正弦函数单位圆内扫描, 并通过黄金分割率缩小搜索空间以逼近算法的最优解. 黄金算法具有结构简单, 寻优能力强等特点, 参数 $c_1$ 、 $c_2$ 设置为

$$c_1 = a(1 - \varphi) + b\varphi, \quad (23)$$

$$c_2 = a\varphi + b(1 - \varphi). \quad (24)$$

其中:  $a$ 取 $-\pi$ ;  $b$ 取 $\pi$ ;  $\varphi$ 为黄金分割数, 取值为0.618.

黄金正弦策略迭代公式如下:

$$P_c^{t+1} = P_c^t |\sin(r_1)| - r_2 \sin(r_1) \times |c_1 P_{bc}^t - c_2 P_c^t|. \quad (25)$$

其中:  $r_1$ 为 $[0, 2\pi]$ 之间的随机数,  $r_2$ 是 $[0, 2]$ 之间的随机数.

#### 1.4.2 生存机制

为了使较差的个体尽快向最优个体靠近, 将非洲野狗算法的生存机制引入到算法中, 生存率不足的个体需要向最优个体周围移动, 以便提高自己的生存概

率. 同时为了减少参与黄金正弦策略的个体数量, 在尽量不影响迭代效果的前提下减少计算量.  $P$ 为自适应参数, 生存率小于 $P$ 的个体 $P_c^t$ 将执行黄金正弦策略, 生存率公式如下:

$$sr = \frac{|f(P_{bc}^t) - f(P_c^t)|}{|f(P_{bc}^t) - f(P_{wc}^t) + \varepsilon|}, \quad (26)$$

$$P = \frac{0.7 \times [t_{\max} - (t + 1)]}{t_{\max}}. \quad (27)$$

其中:  $sr$ 为个体生存率;  $P$ 为判断是否执行生存行为的参数;  $\varepsilon$ 为无穷小的数, 防止分母为零.

## 2 算法流程与分析

### 2.1 算法流程

改进后算法整体步骤如图5所示.

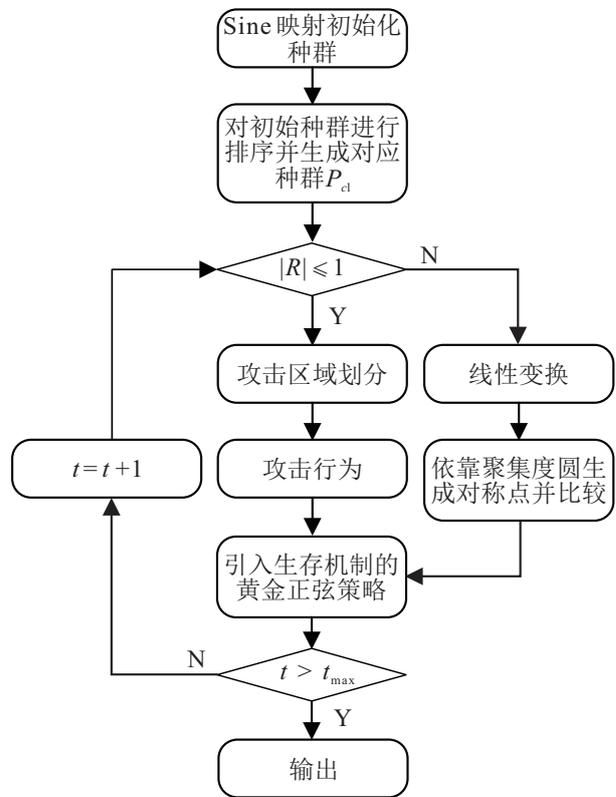


图5 整体流程

step 1: 设置种群规模 $N$ 、维度 $\dim$ 、最大迭代次数 $t_{\max}$ 、搜索空间上下限 $ub$ 和 $lb$ .

step 2: 利用Sine映射初始化种群, 并对初始种群排序, 对应生成排序后的种群 $P_{c1}$ .

step 3: 更新种群最优解、种群最差解、个体最优解, 计算算法参数 $r_G$ 、 $r$ 、 $R$ .

step 4: 利用式(18)计算全部个体到当前最优个体之间的距离.

step 5: 利用式(10)更新角度权值.

step 6: 当 $|R| > 1$ 时, 执行式(12)~(16)进行线性变换, 随后通过式(20)~(22)计算 $P_{c2}$ , 比较 $P_c^t$ 与 $P_{c2}$ 的适应度, 选择适应度更高的个体作为迭代结果; 当

$|R| \leq 1$  时, 通过式(4)生成随机个体, 生成随机数  $s$ , 计算随机个体到当前最优个体之间的距离, 最后执行式(8)、(9)、(11)、(5).

step 7: 利用式(26)计算个体生存率  $sr$ , 当  $sr$  小于  $P$  时, 执行式(23)~(25).

step 8: 重复 step 3~step 7, 直到得到最大迭代次数或算法收敛.

### 2.2 时间复杂度分析

设种群数量为  $N$ , 维度为  $\text{dim}$ , 最大迭代次数为  $t_{\max}$ . 其中计算参数的时间为  $t_{\max} \times 3$ , 更新个体位置的时间为  $2 \times t_{\max} \times N \times \text{dim}$ . 初始沙猫群算法的时间复杂度为

$$O[t_{\max} \times 3 + 2 \times t_{\max} \times N \times \text{dim}] = O(t_{\max} \times N \times \text{dim}).$$

Sine 映射的时间复杂度为

$$O[1 + N \times \text{dim}].$$

在攻击行为阶段, 假设对初始种群进行排序的时间为  $\tau_1$ , 生成随机数的时间为  $\tau_2$ ,  $N$  个个体到当前最优个体距离的时间为  $t_{\max} \times N$ , 角度权值的时间为  $t_{\max} \times N \times \text{dim}$ , 生成随机个体的时间为  $t_{\max} \times N \times \text{dim}$ , 沙猫位置更新的时间为  $2 \times t_{\max} \times N \times \text{dim}$ , 改进后攻击行为的时间复杂度为

$$O[\tau_1 + (4 + \tau_2) \times t_{\max} \times N \times \text{dim} + t_{\max} \times N] = O(t_{\max} \times N \times \text{dim}).$$

在搜索行为阶段,  $\bar{d}$  时间为  $2 \times t_{\max}$ ,  $N$  个个体到最优个体之间距离的时间为  $2 \times t_{\max} \times N \times \text{dim}$ , 迭代维度权值的时间为  $\tau_3$ , 进行位置更新的时间为  $t_{\max} \times N \times \text{dim}$ , 设置聚集度圆半径的时间为  $2 \times t_{\max}$ , 个体到最优个体距离的时间为  $N \times t_{\max}$ , 生成  $P_{c2}$  的时间为  $N \times t_{\max}$ , 比较  $P_c^t$  与  $P_{c2}$  大小的时间为  $N \times t_{\max}$ . 改进搜寻行为的时间复杂度为

$$O[4 \times t_{\max} + (\tau_3 + 2) \times t_{\max} \times N \times \text{dim} + 3 \times N \times t_{\max}] = O(t_{\max} \times N \times \text{dim}).$$

引入生存机制的黄金正弦策略的时间复杂度为  $O[N \times t_{\max} \times 5]$ .

综上所述, 改进后沙猫群算法的时间复杂度为

$$O(1 + N \times \text{dim} + t_{\max} \times N \times \text{dim} + N \times t_{\max} \times 5) = O(t_{\max} \times N \times \text{dim}),$$

与改进之前算法的时间复杂度相同.

### 2.3 收敛性分析

在基本沙猫群算法中攻击行为的作用在于发现新猎物, 而搜寻行为的作用是向已经发现的猎物靠近, 因此这里不对攻击行为做收敛性分析.

现对搜寻行为的收敛性进行分析<sup>[10-12]</sup>. 除  $P_{bc}^t$  外, 每个维度的迭代相互独立, 故将算法简化到一维进行分析, 即将  $P_{bc}^t$  简化为一个常量  $p$ . 式(6)简化为

$$P_c(t + 1) = rp - rR_1P_c(t). \quad (28)$$

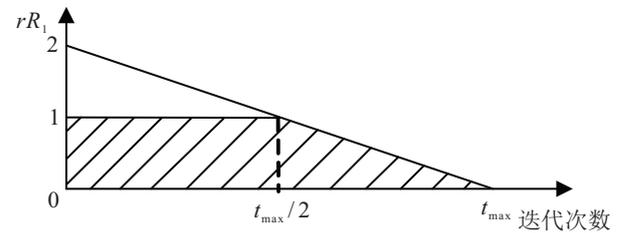
式(28)用矩阵形式表示为

$$y(t + 1) = Ay(t) + Bp. \quad (29)$$

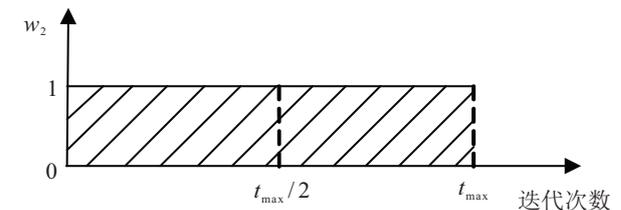
其中:  $y(t) = P_c(t)$ ;  $A = [-rR_1]$ ;  $B = [r]$ ;  $A$  为动态方程的系数;  $p$  为外部输入;  $B$  为输入矩阵, 通过外部输入对个体状态施加影响. 个体的状态决定于  $A$  的特征值, 矩阵  $A$  只有一个元素, 特征值为元素本身, 因此其特征值取决于  $-r \times R_1$ . 对于离散时间线性时不变自治系统, 李亚普诺夫意义下稳定的充分条件为: 全部特征值的幅值均等于或小于 1, 且等于 1 的特征值只能为最小多项式的单根. 原始沙猫群算法搜索行为系数矩阵  $A$  的特征值如图 6(a) 所示, 图中阴影部分包括边界线为收敛区域, 由图像可知迭代次数在 0 至  $t_{\max}/2$  之间时, 系统可能不收敛. 将改进搜寻行为的迭代公式简化为如下形式:

$$P_c(t + 1) = w_1p + w_2P_c(t)'. \quad (30)$$

改进后搜寻行为的系数矩阵  $A$  特征值的大小取决于  $w_2$ , 而  $w_2 = 1 - w_1$ ,  $w_1$  是  $0 \sim 1$  之间的数, 因此  $w_2$  的值如图 6(b) 所示, 改进后搜索行为系数矩阵  $A$  的特征值在全部迭代过程中始终小于等于 1, 因此在整个迭代过程中始终收敛, 与初始搜寻行为相比收敛性能得到了提升.



(a) 初始特征值



(b) 改进后特征值

图 6 特征值

### 3 测试仿真

#### 3.1 参数设置

选取沙猫群算法(SCSO)、粒子群算法(PSO)、灰狼算法(GWO)、鲸鱼算法(WOA)与改进的沙猫群算法(IMSCSO)进行比较. 基本参数设置为: 种群数量  $N = 30$ 、维度  $\text{dim} = 30$ 、最大迭代次数  $t_{\max} = 500$ . 具体算法参数设置如表3所示.

表3 算法参数设置

算法名称	基本参数
SCSO	$S_M = 2$
PSO	$C_1 = C_2 = 1.5, W = 0.7$
GWO	-
WOA	$b = 1$
IMSCSO	$S_M = 2, t_1 = 100, t_2 = 200, t_3 = 300,$ $t_4 = 400, t_{\max} = 500$

#### 3.2 测试函数介绍

采用12个基本测试函数进行测试, 其中  $F_1 \sim F_5$  为单峰函数,  $F_6 \sim F_{12}$  为多峰函数. 测试函数具体信息如表4所示.

表4 测试函数

序号	函数名	定义域	最优值
$F_1$	Sphere function	$[-100, 100]$	0
$F_2$	Schwefel's problem 2.22	$[-10, 10]$	0
$F_3$	Schwefel's problem 1.2	$[-100, 100]$	0
$F_4$	Schwefel's problem 2.21	$[-100, 100]$	0
$F_5$	Matyas function(20)	$[-10, 10]$	0
$F_6$	Generalized Schwefel's problem 2.26	$[-500, 500]$	-12 569.5
$F_7$	Generalized Rastrigin's function	$[-5.12, 5.12]$	0
$F_8$	Ackley's function	$[-32, 32]$	0
$F_9$	Generalized Griewank's function	$[-600, 600]$	0
$F_{10}$	Shekel's family1	$[0, 10]$	-10
$F_{11}$	Shekel's family2	$[0, 10]$	-10
$F_{12}$	Shekel's family3	$[0, 10]$	-10

#### 3.3 Wilcoxon秩和检测

Wilcoxon秩和检验是一种非参数替代双样本  $t$  检验, 这种检验方法可以被应用于比较两个独立的数组之间的差异性. 本文引入 Wilcoxon秩和检验以验证改进后的沙猫群算法与其他4种算法是否具有显著差异性. 每个算法进行30次独立运算并计算  $p$  值进行对比分析.  $p < 0.05$  表明两种算法之间具有显著差异性, N/A表示计算结果相近. 通过表5数据可知, 在函数  $F_7 \sim F_9$  中, IMSCSO与SCSO性能相当. 在函数  $F_5$  中, IMSCSO与PSO性能相当. 其余的  $p$  值均小于0.05, 表明改进后的沙猫群算法具有更优越的性能.

表5 秩和检测

函数序号	SCSO	PSO	GWO	WOA
$F_1$	1.21e-12	4.57e-12	1.21e-12	1.21e-12
$F_2$	3.02e-11	3.02e-11	3.02e-11	3.02e-11
$F_3$	3.00e-11	3.00e-11	3.00e-11	3.00e-11
$F_4$	5.57e-10	5.22e-09	5.57e-10	5.57e-10
$F_5$	1.21e-12	N/A	1.21e-12	1.21e-12
$F_6$	3.02e-11	3.02e-11	3.02e-11	3.02e-11
$F_7$	N/A	1.21e-12	4.52e-12	1.21e-12
$F_8$	N/A	4.52e-12	1.03e-12	1.21e-12
$F_9$	N/A	4.57e-12	6.62e-04	5.36e-06
$F_{10}$	1.10e-06	6.7e-03	3.99e-04	3.99e-04
$F_{11}$	8.29e-06	6.7e-03	7.60e-07	3.99e-04
$F_{12}$	3.99e-04	5.69e-05	2.03e-09	3.99e-04
+/-/-	9/3/0	11/1/0	12/0/0	12/0/0

#### 3.4 算法优化性能测试

每个测试函数分别进行50次独立运算, 测试结果如表6所示. 在测试函数  $F_7 \sim F_9$  中, 改进后的沙猫群算法与初始沙猫群算法寻优能力相当. 在与其他算法的对比中, 改进后的沙猫群算法在搜索能力和稳定性方面都具有明显优势.

表6 实验结果

函数序号	IMSCSO		SCSO		PSO		GWO		WOA	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
$F_1$	0.00e+00	0.00e+00	7.93e-98	5.55e-87	6.40e-02	3.76e-02	1.23e-29	1.59e-27	1.30e-19	2.24e-19
$F_2$	1.05e-161	9.69e-162	5.02e-49	3.50e-48	7.63e-01	3.21e-01	1.19e-16	1.19e-16	3.30e-13	2.62e-13
$F_3$	4.16e-321	0.00e+00	5.17e-61	3.62e-60	1.73e-01	1.14e-01	1.10e-05	2.39e-05	7.25e-03	2.87e-02
$F_4$	5.51e-163	0.00e+00	7.42e-30	5.14e-29	3.38e-01	2.16e-01	7.42e-07	7.96e-09	2.04e-04	1.99e-04
$F_5$	0.00e+00	0.00e+00	5.66e-220	0.00e+00	3.68e-31	1.83e-30	1.96e-101	1.15e-100	5.4e-115	3.18e-114
$F_6$	-12 569.413	1.00e-01	-6.50e+03	9.34e+02	-5.27e+03	1.47e+03	-5.89e+03	1.04e+03	-6.90e+03	5.75e+02
$F_7$	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.24e+02	2.99e+01	2.54e+00	3.48e+00	5.15e+00	5.53e+00
$F_8$	4.44e-16	0.00e+00	4.44e-16	0.00e+00	3.24e-01	1.50e-01	1.01e-13	2.12e-14	4.02e+00	8.04e+00
$F_9$	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.76e-03	2.91e-03	4.58e-03	7.96e-03	1.31e-02	1.60e-02
$F_{10}$	-1.015e+01	9.11e-03	-5.52e+00	2.11e+00	-8.53e+00	3.47e+00	-9.20e+00	2.04e+00	-6.73e+00	3.34e+00
$F_{11}$	-1.04e+01	3.94e-02	-5.77e+00	2.18e+00	-6.26e+00	3.24e+00	-1.02e+01	1.04e+00	-6.66e+00	3.03e+00
$F_{12}$	-1.05e+00	8.3e-03	-6.01e+00	2.83e+00	-5.39e+00	2.91e+00	-1.01e+01	1.86e+00	-6.13e+00	2.77e+00

### 3.5 轴承故障检测

为验证IMSCSO在实际应用中的有效性,利用不同优化算法对SVM的核参数和惩罚因子进行优化并建立轴承故障诊断模型.本文采用美国凯斯西储大学轴承数据集进行测试,参数设置如表7所示.各算法预测结果如图7~图11所示.

表7 实验参数说明

状态	训练样本	测试样本	特征值
内圈故障	70	30	1
外圈故障	70	30	2
滚动轴承故障	70	30	3
正常运行	70	30	4

由分类结果可知,IMSCSO-SVM、SCSO-SVM、GWO-SVM、PSO-SVM、WOA-SVM的准确率分别为98.3333%、94.1667%、95%、89.1667%和91.1667%,结果表明IMSCSO-SVM分类的准确率更高,具有优越性.

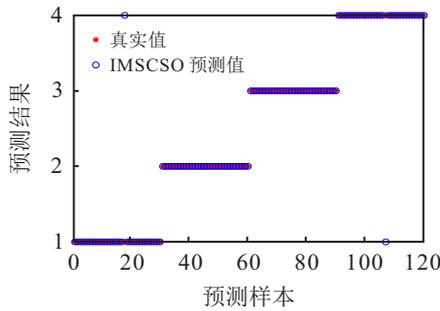


图7 IMSCSO-SVM预测结果

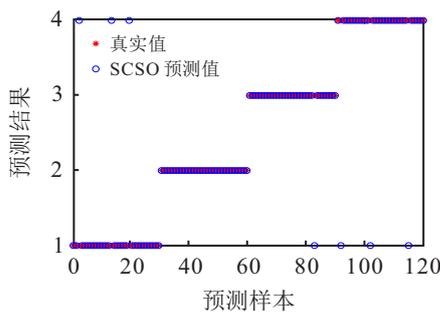


图8 SCSO-SVM预测结果

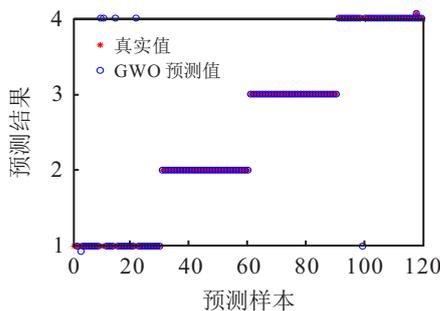


图9 GWO-SVM预测结果

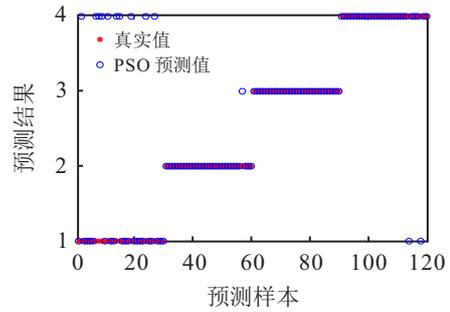


图10 PSO-SVM预测结果

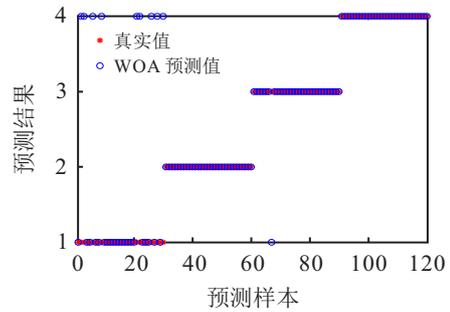


图11 WOA-SVM预测结果

## 4 结论

本文通过对沙猫群算法攻击行为和搜寻行为的分析发现了算法的不足,并针对缺点进行改进.在搜寻过程中消除不同维度之间的影响,规避迭代过程中的振荡现象,加强其跳出局部最优的能力和收敛效率,提升算法的局部开发能力.通过对12个基本测试函数的测试表明,改进后的沙猫群算法具有更好的寻优能力和稳定性,与基本沙猫群算法相比在时间复杂度不变的前提下,算法的寻优性能得到了显著提升,算法的收敛性能得到了加强.在轴承故障检测中,经IMSCSO优化后的故障诊断模型分类准确率更高,表明其在实际应用中也具有一定的优越性.

### 参考文献(References)

- [1] 刘建华, 樊晓平, 瞿志华. 一种基于相似度的新型粒子群算法[J]. 控制与决策, 2007, 22(10): 1155-1159. (Liu J H, Fan X P, Qu Z H. A new particle swarm optimization algorithm based on similarity[J]. Control and Decision, 2007, 22(10): 1155-1159.)
- [2] 周新, 邹海. 融合黄金正弦混合变异的自适应樽海鞘群算法[J]. 计算机工程与应用, 2021, 57(12): 75-85. (Zhou X, Zou H. Adaptive salp swarm algorithm with golden sine algorithm and hybrid mutation[J]. Computer Engineering and Applications, 2021, 57(12): 75-85.)
- [3] 回立川, 陈雪莲, 孟嗣博. 多策略混合的改进麻雀搜索算法[J]. 计算机工程与应用, 2022, 58(16): 71-83. (Hui L C, Chen X L, Meng S B. Improved sparrow search algorithm based on multi-strategy mixing[J]. Computer Engineering and Applications, 2022, 58(16): 71-83.)
- [4] Seyyedabbasi A, Kiani F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization

- problems[J]. *Engineering with Computers*, 2023, 39(4): 2627-2651.
- [5] Wu D, Rao H H, Wen C S, et al. Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems[J]. *Mathematics*, 2022, 10(22): 4350.
- [6] Li Y M, Wang G C. Sand cat swarm optimization based on stochastic variation with elite collaboration[J]. *IEEE Access*, 2022, 10: 89989-90003.
- [7] 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 one-dimensional oppositional learning based small hole imaging[J]. *Control and Decision*, 2021, 36(5): 1173-1180.)
- [8] 梁田, 曹德欣. 基于莱维飞行的改进简化粒子群算法[J]. *计算机工程与应用*, 2021, 57(20): 188-196.  
(Liang T, Cao D X. Improved and simplified particle swarm optimization algorithm based on levy flight[J]. *Computer Engineering and Applications*, 2021, 57(20): 188-196.)
- [9] Tanyildizi E, Demir G. Golden sine algorithm: A novel math inspired algorithm[J]. *Advances in Electrical and Computer Engineering*, 2017, 17(2): 71-78.
- [10] Clerc M, Kennedy J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1): 58-73.
- [11] 张丽平. 粒子群优化算法的理论及实践[D]. 杭州: 浙江大学, 2005: 25-30.  
(Zhang L P. Theory and practice of particle swarm optimization algorithm[D]. Hangzhou: Zhejiang University, 2005: 25-30.)
- [12] 郑大钟. 线性系统理论[M]. 北京: 清华大学出版社, 1990: 248-251.  
(Zheng D Z. Linear system theory[M]. Beijing: Tsinghua University Press, 1990: 248-251.)
- [13] Peraza-Vázquez H, Peña-Delgado A F, Echavarría-Castillo G, et al. A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies[J]. *Mathematical Problems in Engineering*, 2021, 2021: 1-19.
- [14] 孙丽君, 冯斌斌, 陈天飞. 基于鞅论的灰狼优化算法全局收敛性分析[J]. *控制与决策*, 2022, 37(11): 2839-2848.  
(Sun L J, Feng B B, Chen T F. Global convergence analysis of grey wolf optimization algorithm based on martingale theory[J]. *Control and Decision*, 2022, 37(11): 2839-2848.)
- [15] 何庆, 罗仕杭. 混合改进策略的黑猩猩优化算法及其机械应用[J]. *控制与决策*, 2023, 38(2): 354-364.  
(He Q, Luo S H. Chimp optimization algorithm based on hybrid improvement strategy and its mechanical application[J]. *Control and Decision*, 2023, 38(2): 354-364.)
- [16] 刘威, 郭直清, 王东, 等. 改进鲸鱼算法及其在浅层神经网络搜索中的权值阈值优化[J]. *控制与决策*, 2023, 38(4): 1144-1152.  
(Liu W, Guo Z Q, Wang D, et al. Improved whale optimization algorithm and its weights and thresholds optimization in shallow neural architecture search[J]. *Control and Decision*, 2023, 38(4): 1144-1152.)
- [17] 王凌, 吴昊, 唐芳, 等. 混合量子遗传算法及其性能分析[J]. *控制与决策*, 2005, 20(2): 156-160.  
(Wang L, Wu H, Tang F, et al. Hybrid quantum genetic algorithms and performance analysis[J]. *Control and Decision*, 2005, 20(2): 156-160.)
- [18] 付华, 许桐, 邵靖宇. 基于水波进化和动态莱维飞行的爬行动物搜索算法[J]. *控制与决策*, 2024, 39(1): 59-68.  
(Fu H, Xu T, Shao J Y. Reptile search algorithm based on water wave evolution and dynamic Levy flight[J]. *Control and Decision*, 2024, 39(1): 59-68.)
- [19] 付华, 刘尚霖, 管智峰, 等. 阶段化改进的海洋捕食者算法及其应用[J]. *控制与决策*, 2023, 38(4): 902-910.  
(Fu H, Liu S L, Guan Z F, et al. Phased-improvement marine predators algorithm and its application[J]. *Control and Decision*, 2023, 38(4): 902-910.)
- [20] 陈晟宗, 张纪会, 于守水, 等. 求解旅行商问题的波动温控模拟退火算法[J]. *控制与决策*, 2023, 38(4): 911-920.  
(Chen S Z, Zhang J H, Yu S S, et al. A simulated annealing algorithm with wave temperature control for the traveling salesman problem[J]. *Control and Decision*, 2023, 38(4): 911-920.)

### 作者简介

回立川(1980—), 男, 副教授, 硕士生导师, 博士, 从事智能优化算法及应用的研究, E-mail: 104675117@qq.com;

于千皓(1999—), 男, 硕士生, 从事智能优化算法及应用的研究, E-mail: 396271262@qq.com.