

For citation: Cao Lican, Xu Qiao. A multi strategy improved sand cat swarm optimization algorithm // Grand Altai Research & Education — Issue 1 (24)'2025 (DOI: 10.25712/ASTU.2410-485X.2025.01) — EDN: <https://elibrary.ru/SEDEZT>

UDK 004.021

A MULTI STRATEGY IMPROVED SAND CAT SWARM OPTIMIZATION ALGORITHM

Cao Lican¹, Xu Qiao¹

1 Hubei Digital Textile Equipment Key Laboratory, Wuhan Textile University, Wuhan, 430073, China
E-mail: 351669242@qq.com ; 327778240@qq.com

Abstract: In order to overcome the shortcomings of the development capability of the standard sand cat swarm optimization algorithm and avoid its tendency to fall into local optima, this paper proposes an improved sand cat swarm optimization algorithm (MSISCSO) that integrates multiple strategies. Firstly, use Sin Tent Cos chaotic mapping to initialize the initial individuals of the sand cat population and increase the diversity of the initial population; Secondly, the first stage global development strategy of the Raccoon Optimization Algorithm is introduced to update the position of the sand cat, enhancing its convergence speed during the development stage; Furthermore, introducing a crossover strategy instead of the position update formula during the exploration phase of the sand cat group enhances its ability to escape from local optima; Finally, the sand cat group is subjected to adaptive t-distribution perturbation to further enhance its global exploration ability. Comparative tests were conducted on six test functions using MSISCSO and five other algorithms, and the results showed that the MSISCSO algorithm had faster convergence speed and accuracy on all six test functions. Finally, the MSISCSO algorithm was applied to solve spring design problems, demonstrating that the improved algorithm can solve complex engineering optimization problems.

Key words: Sand cat swarm optimization algorithm; Chaotic mapping; Vertical and horizontal intersections; Adaptive t-distribution

一种多策略改进的沙猫群优化算法

曹李灿¹, 徐巧¹

1 湖北省数字化纺织装备重点实验室, 武汉纺织大学, 中国, 武汉, 430073
E-mail: 351669242@qq.com ; 327778240@qq.com

摘要: 为了克服标准沙猫群优化算法开发能力短板, 避免其容易陷入局部最优的缺陷, 本文提出了一种融合多种策略的改进沙猫群优化算法 (MSISCSO). 首先, 使用 Sin-Tent-Cos 混沌映射初始化沙猫群初始个体, 增加沙猫初始种群多样性; 其次, 引入浣熊优化算法第一阶段全局开发策略进行沙猫位置更新, 增强其开发阶段收敛速度; 再者, 引入纵横交叉策略代替沙猫群探索阶段的位置更新公式, 增强其跳出局部最优的能力; 最后对沙猫群进行自适应 t 分布扰动, 进一步增强其全局探索能力. 在6个测试函数使用 MSISCSO 与其它5种算法上进行对比测试, 结果表明 MSISCSO 算法在6个测试函数上均具有更快的收敛速度和收敛精度. 最后将 MSISCSO 算法用于解决弹簧设计问题, 表明改进后算法可以解决复杂的工程优化问题.

关键词: 沙猫群优化算法; 混沌映射; 纵横交叉; 自适应 t-分布

0引言

优化算法是一种在优化问题中, 寻找到最优解或近似最优解的方法, 广泛用于在各类优化问题中寻找最优方案. 常见的优化算法有遗传算法 [1], 粒子群算

法 [2], 模拟退火算法 [3] 等. 这些优化算法在生产调度, 交通运输, 人工智能等领域展现出了巨大的应用价值.

沙猫群优化算法 (Sand cat swam optimization, SCSO) 是一种模拟沙猫在沙漠中捕食与生存行为的智能优化算法, 该算法于 2022 年提出 [4]. 该算法一经提出, 便受到了许多学者的关注. Kiani 等人提出了一种增强的 PSCSO, 用于解决复杂的工程优化问题 [5]. 贾鹤鸣等人使用互利共生机制与 Levy 飞行策略对沙猫群优化算法进行了改进, 并将改进后的算法用于解决机器人路径规划问题并取得了不错的效果 [6]. 王雷等人将高斯随机游走策略与麻雀警戒机制用于改进沙猫群优化算法, 并使用 cubic 混沌映射初始化种群, 并将改进后的沙猫群优化算法用于无刷电机直流控制 [7]. 总的来看, 沙猫群优化算法仍然存在种群多样性低, 开发与探索能力不足的缺点, 因此本文提出了一种融合多种策略改进的沙猫群优化算法 (MSISCSO). 首先, 使用 Sin-Tent-Cos 混沌映射用于初始化种群提高原算法初始种群的多样性与随机性; 其次本文引入浣熊优化算法第一阶段全局开发策略进行沙猫群位置更新; 再者, 本文引入纵横交叉策略来代替沙猫群的后期探索阶段进行位置更新; 最后对沙猫群进行自适应 t 分布扰动, 进一步提高原算法跳出局部最优的能力. 并对改进后的沙猫群优化算法进行基准函数测试, 结果显示改进后的算法相比于其它优化算法, 表现出更好的优化性能.

1 沙猫群优化算法

原始沙猫群优化算法可分为两个阶段: 开发阶段和探索阶段.

1.1 探索阶段

沙猫在搜寻猎物时, 主要是对低频段的感知. 沙猫的听觉范围可用式 (1) 进行表示.

$$\vec{r}_G = S_M - \left(\frac{2 \times S_M \times t}{T_{max}} \right) \quad (1)$$

式中, S_m – 为常数, t – 为迭代次数, T_{max} – 为最大迭代次数.

沙猫群优化算法使用随机向量进行探索阶段与开发阶段的划分. 随机向量 R 可用式 (2) 表示.

$$\vec{R} = 2 \times \vec{r}_G \times rand(0,1) - \vec{r}_G \quad (2)$$

当 $|\vec{R}| > 1$ 时, 沙猫群优化算法进入探索阶段, 沙猫在探索阶段的位置更新公式可以用式 (3) 表示.

$$\vec{pos}(t+1) = \vec{r} \times (\vec{pos}_b(t) - rand(0,1) \times \vec{pos}(t)) \quad (3)$$

式中, $\vec{pos}(t+1)$ 表示迭代后沙猫的位置, $\vec{pos}_b(t)$ 表示当代种群中最优沙猫个体, $\vec{pos}(t)$ 表示当前沙猫个体, $\vec{r}$ 为参数向量, 可用式 (4) 表示.

$$\vec{r} = \vec{r}_G \times rand(0,1) \quad (4)$$

1.2 开发阶段

当 $|R| \leq$ 时, 沙猫群优化算法进入开发阶段, 在开发阶段沙猫基于式 (5) 进行位置更新.

$$\overrightarrow{pos}(t+1) = \overrightarrow{pos}_b(t) - \vec{r} \times \overrightarrow{pos}_{ind} \times \cos(\theta) \quad (5)$$

式中 θ 为一随机角度, $\theta \in [0, 360]$, $\overrightarrow{pos}_{ind}$ 表示随机一只沙猫的位置, 具体可用式 (6) 计算.

$$\overrightarrow{pos}_{ind} = |rand(0,1) \times \overrightarrow{pos}_b(t) - \overrightarrow{pos}(t)| \quad (6)$$

2 改进沙猫群优化算法

2.1 Sin-Tent-Cos 混沌映射

原始沙猫群优化算法的初始种群是随机生成的, 这可能会导致种群的分布均匀性变差, 因此本文采用 Sin-Tent-Cos 混沌映射进行种群初始化. Sin-Tent-Cos 混沌映射是一种复合的混沌映射, 该种混沌映射由基本的 Sine 混沌映射, Tent 混沌映射, Cosine 混沌映射组合而成, 相比于单一的混沌映射, Sin-Tent-Cos 混沌映射在遍历性等方面具有明显优势. 其公式为:

$$x(t+1) = \begin{cases} \cos(\pi(rs\sin(\pi x(t)) + 2(1-r)x(i) - 0.5)), & \text{if } x(t) < 0.5 \\ \cos(\pi(rs\sin(\pi x(t)) + 2(1-r)(1-x(i)) - 0.5)), & \text{else} \end{cases}, r \in [0,1] \quad (7)$$

2.2 融合浣熊优化算法

受浣熊优化算法 [8] 启发, 使用浣熊优化算法在开发阶段前一半个体的全局搜索策略来代替沙猫进行全局搜索行为. 浣熊优化算法的全局搜索策略可以解决原算法需要较多参数的问题, 同时可进一步加强沙猫的开发能力. 其位置更新公式为:

$$\overrightarrow{pos}_h(t+1) = \overrightarrow{pos}(t) + r_1 \times (\overrightarrow{pos}_b(t) - r_2 \times \overrightarrow{pos}(t)) \quad (8)$$

式中 $\overrightarrow{pos}_h(t+1)$ 表示经过浣熊优化算法更新策略后的沙猫位置, r_1 为 $[0,1]$ 之间的随机数, r_2 为 $[1,2]$ 之间的随机数. 经过浣熊优化算法更新策略后的沙猫个体在进行范围限制后和原沙猫个体进行贪婪选择, 保留更优的个体, 其公式为:

$$\overrightarrow{pos}(t+1) = \begin{cases} \overrightarrow{pos}_h(t+1), & \text{if } fit(\overrightarrow{pos}_h(t+1)) < fit(\overrightarrow{pos}(t)) \\ \overrightarrow{pos}(t), & \text{else} \end{cases} \quad (9)$$

2.3 纵横交叉策略

针对沙猫群优化算法在迭代后期容易陷入局部最优解的问题, 本文引入纵横交叉策略 [9] 用于沙猫群优化算法的探索阶段以提高算法跳出局部最优的能力.

2.3.1 横向交叉

不同的沙猫个体之间在同一维度上进行交叉运算, 使得沙猫之间可以利用对方的位置信息进行进一步的探索. 横向交叉的位置更新公式为:

$$New_{pos_{i,d}}^t = r_1 \times \overrightarrow{pos_{i,d}^t} + (1 - r_1) \times \overrightarrow{pos_{j,d}^t} + r_2 \times (\overrightarrow{pos_{i,d}^t} - \overrightarrow{pos_{j,d}^t}) \quad (10)$$

$$New_{pos_{j,d}}^t = r_3 \times \overrightarrow{pos_{j,d}^t} + (1 - r_3) \times \overrightarrow{pos_{i,d}^t} + r_4 \times (\overrightarrow{pos_{j,d}^t} - \overrightarrow{pos_{i,d}^t}) \quad (11)$$

式中 $New_{pos_{i,d}}^t, New_{pos_{j,d}}^t$ 分别为沙猫在 d 维度横向交叉后的个体 r_1, r_3 为 0 到 1 之间的随机数 r_2, r_4 为 -1 到 1 之间的随机数, 将横向交叉后的新沙猫个体与原沙猫个体进行贪婪选择, 保留更优的沙猫个体.

2.3.2 纵向交叉

不同维度上的同一沙猫个体进行交叉运算, 使得沙猫可以避免在某一维度上陷入局部最优. 纵向交叉的位置更新公式为:

$$New'_{pos_{i,d}}^t = r_5 \times \overrightarrow{pos_{i,d1}^t} + (1 - r_5) \times \overrightarrow{pos_{i,d2}^t} \quad (12)$$

式中 $New'_{pos_{i,d}}^t$ 为沙猫在 $d1, d2$ 与维度交叉后的新个体, r_5 为 0 到 1 之间的随机数, 同样地, 对纵向交叉后的新沙猫与原始沙猫进行贪婪选择, 保留更优的沙猫个体.

2.4 自适应 t 分布扰动

针对沙猫群优化算法后期探索能力不够的问题, 本文引入自适应分布扰动 [10] 对进行探索行为之后的沙猫群进行扰动. 分布概率密度函数可用 (13) 表示:

$$f(t) = \frac{Gam(\frac{n+1}{2})}{(\pi n)^{\frac{1}{2}} Gam(\frac{n}{2})} \times \left(1 + \frac{t^2}{n}\right)^{-\frac{n+1}{2}} \quad (13)$$

式中 $Gam(x)$ 为伽马函数, 其自由度参数 n 决定了 t 分布的最终形态: 当自由度参数 $n=1$ 时, t 分布可以看为柯西分布, 当自由度参数 n 趋向无穷时, t 分布收敛于高斯分布. 本文用沙猫群优化算法的迭代次数作为 t 分布的自由度参数, 实现 t 分布开发与探索的平衡. 在算法的前期, 迭代次数较小, 因此可以增强沙猫的全局勘探能力, 同时在算法后期, 迭代次数较大, 可增强沙猫跳出局部最优的能力, 此时沙猫的位置更新公式为:

$$\overrightarrow{pos}(t+1) = \overrightarrow{pos}(t) + \overrightarrow{pos}(t) \times trnd(t) \quad (14)$$

式中, $\overrightarrow{pos}(t+1)$ 为经过扰动后的沙猫个体, $trnd(t)$ 为参数为沙猫群优化算法迭代次数的分布。

2.5 改进沙猫群优化算法流程

MSISCSO 的流程如下:

Step1: 根据式 (7) 进行混沌初始化种群, 得到初始沙猫群位置。

Step2: 计算每一只沙猫的适应度, 并根据适应度选出当前种群中最优沙猫个体并记录其适应度。

Step3: 根据式 (1) 和式 (2) 计算随机向量 R。

Step4: 当 $|R| > 1$ 时, 使用式 (8) 和式 (9) 进行沙猫群位置更新。

Step5: 当 $|R| \leq 1$ 时, 使用式 (10), (11), (12) 进行沙猫群位置进行交叉。

Step6: 根据式 (13) 和式 (14) 对沙猫群进行分布扰动。

Step7: 判断是否达到最大迭代次数, 如果是, 则输出最优沙猫位置, 最优沙猫适应度, 适应度迭代曲线; 否则返回步骤2。

改进沙猫群优化算法流程图见图1。

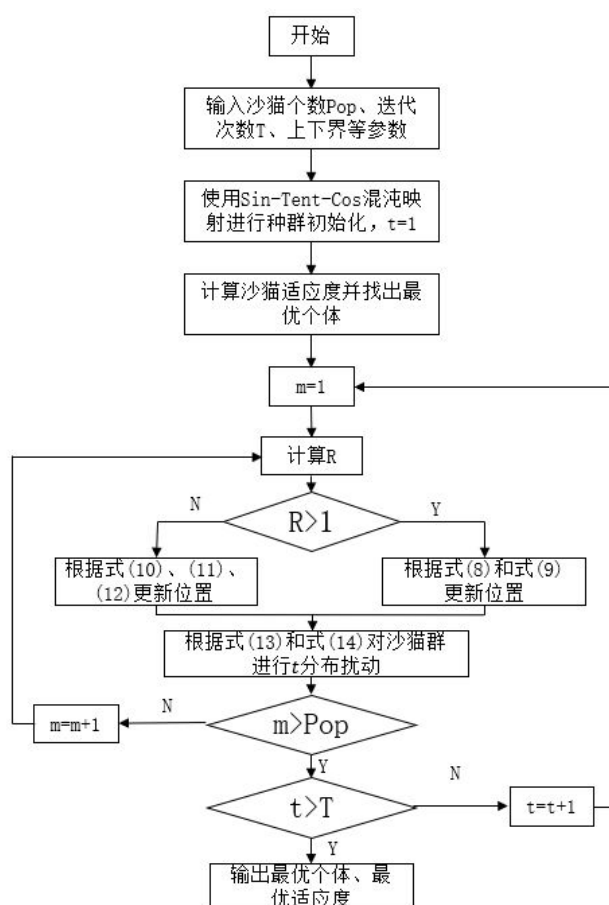


图1 MSISCSO 算法流程图
Fig. 1 MSISCSO algorithm flowchart

3 仿真验证与结果分析

3.1 测试函数以及测试指标

为了验证本文所提出的 MSISCSO 算法的性能, 本文选取了 6 个测试函数来进行性能测试, 其具体信息见表1. 在这 6 个测试函数中 F1~F3 为单峰函数, 这类函数在整个搜索空间内仅存在一个最优值, 故用于测试算法的收敛速度和精度; F4~F5 为多峰函数, 其在搜索范围内分布着多个局部最优解, 同时存在一个全局最优解, 可用于测试算法跳出局部最优的能力; F6 为固定多峰函数, 其搜索空间构造较为复杂, 能有效测试算法的在复杂搜索空间中的搜索效能. 同时本文将 MSISCSO 算法与沙猫群优化算法 (SCSO), 粒子群优化算法 (PSO), 灰狼优化算法 (GWO), 蜣螂优化算法 (DBO), 霜冰优化算法 (RIME) 进行比较, 其中所有算法种群数设为 30, 迭代次数设为 500, 每个算法的运行次数设为 30 次. 平均值可以反映算法在多次运行时的总体效果, 而最优值可以反映算法在多次运行时的最佳效果, 标准差可用于体现算法多次运行的稳定性. 因此本文使用平均值, 最优值, 标准差来检验算法的综合性能.

表1 测试函数信息

Tab. 1 Test function information

函数编号	函数名称	搜索范围	维度	最优值
F1	Sphere	[-100,100]	30	0
F2	Schwefel2.22	[-10,10]	30	0
F3	Schwefel2.21	[-100,100]	30	0
F4	Rastrigin	[-5.12,5.12]	30	0
F5	Ackley	[-32,32]	30	0
F6	Shekel 5-m	[0,10]	4	-10.15

3.2 测试结果分析

表2 各算法测试结果

Tab. 2 Test results of various algorithms

函数	指标	MSISCSO	SCSO	PSO	GWO	DBO	RIME
F1	平均值	0	7.27e-111	0.0045	2.44e-27	1.00e-109	2.038
	标准差	0	3.98e-110	0.0033	4.60e-27	5.31e-109	0.660
	最优值	0	4.88e-125	2.15e-04	5.25e-29	4.78e-155	0.8639
F2	平均值	0	7.49e-60	1.6912	9.93e-17	3.05e-54	1.2246
	标准差	0	3.74e-59	4.6071	5.94e-17	1.67e-53	0.6984
	最优值	0	3.12e-65	0.0031	1.70e-17	2.08e-82	0.4611
F3	平均值	0	4.79e-50	7.2961	7.53e-07	2.11e-56	7.1399
	标准差	0	1.45e-49	1.5855	7.50e-07	1.13e-55	2.5275
	最优值	0	3.43e-55	4.0306	7.77e-08	4.44e-83	3.5173
F4	平均值	0	0	61.4946	2.0916	0.9291	65.5334
	标准差	0	0	19.8688	2.9257	3.9801	12.5956
	最优值	0	0	25.9063	0	0	44.9419
F5	平均值	8.88e-16	8.88e-16	0.5747	1.01e-13	8.88e-16	2.2750
	标准差	0	0	0.5553	1.82e-14	0	0.4952

函数	指标	MSISCSO	SCSO	PSO	GWO	DBO	RIME
F6	最优值	8.88e-16	8.88e-16	0.0096	7.55e-14	8.88e-16	1.2907
	平均值	-10.1532	-5.1003	-6.0510	-9.0577	-6.8525	-7.2887
	标准差	6.33e-15	1.6724	3.1174	2.2623	2.5912	2.7906
	最优值	-10.1532	-10.1531	-10.1531	-10.1530	-10.1532	-10.1532

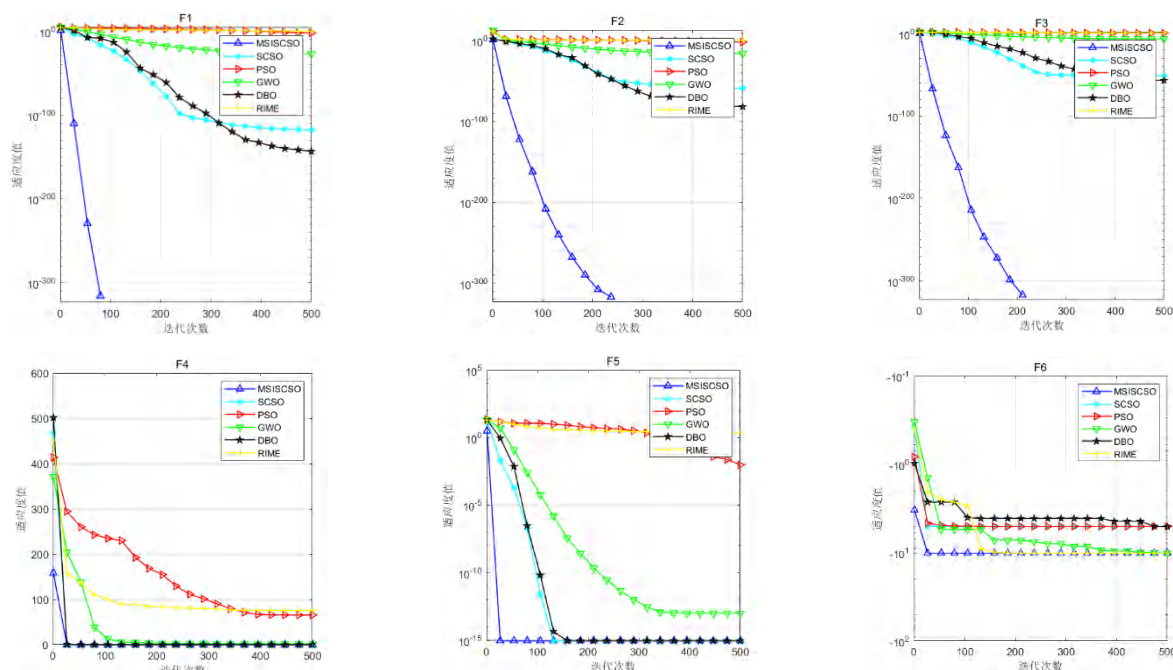


图2 各算法收敛曲线

Fig. 2 Convergence curves of various algorithms

从表2 中可以看出, 改进之后的 MSISCSO 算法在所有类型函数上均有优秀表现. 在单峰类型函数中, MSISCSO 找到了所有函数的理论最优值, 同时标准差为0, 说明 MSISCSO 算法具有极为优秀的鲁棒性和探索能力. 在多峰函数 F4 中, MSISCSO 与 SCSO 的各项评价指标均相同, 但相较于其它算法, 也体现出明显的优势. 在固定多峰函数 F6 中, MSISCSO 算法也找到了最优值, 说明 MSISCSO 在复杂的搜索空间中也可进行快速高效搜索. 从图2 中可以看出在 6 个测试函数上, MSISCSO 算法均表现出了极为优秀的收敛速度和收敛精度, 在算法迭代前期, MSISCSO 可以快速进行全局快速开发, 在算法迭代后期, MSISCSO 可以跳出局部最优值, 继续深入探索.

4 基于 MSISCSO 算法的工程应用

弹簧设计问题 [11] 是一个典型的工程优化问题, 其数学形式可由式 (15) 表示.

$$\begin{aligned}
X &= [x_1, x_2, x_3], f(X) = (x_1 + 2)x_2x_3^2 \\
s. t. \quad g_1(X) &= 1 - \frac{x_2^3x_1}{71785x_3^4} \leq 0 \\
g_2(X) &= \frac{4x_2^2 - x_2x_3}{12566(x_2x_3^3 - x_3^4)} + \frac{1}{5108x_3^2} - 1 \leq 0 \\
g_3(X) &= 1 - \frac{140.45x_3}{x_1x_2^2} \leq 0 \\
g_4(X) &= \frac{x_2 + x_3}{1.5} - 1 \leq 0 \\
2 \leq x_1 \leq 15, 0.25 \leq x_2 \leq 1.3, 0.05 \leq x_3 \leq 2
\end{aligned} \tag{15}$$

本文分别使用改进后的 MSISCSO 算法, SCSO 算法, PSO 算法, GWO 算法, DBO 算法, RIME 算法对该优化问题进行求解, 表3 列出了各个算法的最优解以及对应的最优值和排名.

表3弹簧设计问题算法排名

Tab. 3 Ranking of Algorithm for Spring Design Problems

算法	x_1	x_2	x_3	f	排名
MSISCSO	11.216598	0.357958	0.051741	0.012665	1
SCSO	14.029540	0.317416	0.050000	0.012720	4
PSO	9.162187	0.399485	0.053409	0.012720	5
GWO	9.063138	0.402261	0.053519	0.012747	6
DBO	13.204733	0.328037	0.050472	0.012706	3
RIME	11.313613	0.356575	0.051700	0.012689	2

从表3 中可以看出 MSISCSO 算法相对于其它算法, 可以找出更优值, 表明本文所提出的 MSISCSO 算法对于复杂的工程优化问题也可以进行快速高效求解.

5结语

针对传统的沙猫群优化算法存在收敛速度慢, 容易陷入局部最优等问题, 本文提出了一种融合多种策略的改进沙猫群优化算法 (MSISCSO). 首先, 通过 Sin-Tent-Cos 混沌映射代替原算法的随机初始化提高沙猫初始种群的多样性; 其次, 引入浣熊优化算法第一阶段的全局搜索策略加强沙猫的全局搜索能力; 再者, 通过引入纵横交叉策略解决原算法容易陷入局部最优的问题; 最后, 对沙猫群个体进行自适应 t 分布扰动, 进一步加强沙猫的后期探索能力. 使用测试函数对改进后沙猫群优化算法进行测试, 同时与其它五种优化算法进行对比, 测试结果表明改进的沙猫群优化算法具有优秀的优化性能, 同时通过弹簧设计问题来证明所提出算法的可以用于解决复杂的工程应用问题. 结果表明, 多策略改进的沙猫群优化算法具有较强的全局开发能力和探索能力, 同时对于复杂的工程优化问题也有较为优秀的解决能力.

参考文献

- [1] HOLLANDJH. Genetic algorithms [J]. Scientific American, 1992, 267(1):66-73.
- [2] Marini F., Walczak B. Particle swarm optimization (PSO).a tutorial [J]. Chemometrics and Intelligent Laboratory Systems, 2015, 149:153-165.

- [3] Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing [J]. Science, 1983, 220(4598): 671-680.
- [4] Seyyedabbasi A., Kiani F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems [J]. Engineering with Computers, 2022, 39(4):2627-2651.
- [5] Kiani F., Anka F.A., Erenel F. PSCSO: enhanced sand cat swarm optimization inspired by the political system to solve complex problems [J]. Advances in Engineering Software, 2023, 178: 103423.
- [6] 贾鹤鸣, 李永超, 游进华, 等. 改进沙猫群优化算法的机器人路径规划 [J]. 福建工程学院学报, 2023, 21(1):72-77.
- [7] 王雷, 崔玉鑫, 高亨, 等. 基于多策略融合算法的无刷直流电机控制研究 [J]. 农业装备与车辆工程, 2024, 62(08):118-123.
- [8] Dehghani M., Montazeri Z., Trojovská E., et al. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2023, 259: 110011.
- [9] 曹敬椿, 卢敏. 基于混合策略改进 SSA-SVM 的工业用水量预测 [J]. 水电能源科学, 2023, 41(09):28-31. DOI: 10.20040/j.cnki.1000-7709.2023.20222434.
- [10] 李红民, 马亚伟, 刘瑞玉, 等. 融合混沌映射和自适应 T 分布的蜣螂优化算法 [J]. 软件工程, 2024, 27(11):63-68. DOI: 10.19644/j.cnki.issn2096-1472.2024.011.013.
- [11] 盛斌, 张军. 多策略改进蜣螂优化算法及其应用 [J/OL]. 制造技术与机床, 1-15[2025-02-04] — URL: <http://kns.cnki.net/kcms/detail/11.3398.th.20241216.1455.046.html>.

References

- [1] HOLLANDJH. Genetic algorithms [J]. Scientific American, 1992, 267(1):66-73.
- [2] Marini F., Walczak B. Particle swarm optimization (PSO).a tutorial [J]. Chemometrics and Intelligent Laboratory Systems, 2015, 149:153-165.
- [3] Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing [J]. Science, 1983, 220(4598): 671-680.
- [4] Seyyedabbasi A., Kiani F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems [J]. Engineering with Computers, 2022, 39(4):2627-2651.
- [5] Kiani F., Anka F.A., Erenel F. PSCSO: enhanced sand cat swarm optimization inspired by the political system to solve complex problems [J]. Advances in Engineering Software, 2023, 178:103423.
- [6] Jia Heming, Li Yong Chao, You Jinhua, et c. Improving the Optimization Algorithm for Sand Cat Swarm in Robot Path Planning [J]. Journal of Fujian University of Engineering, 2023, 21(1):72-77.
- [7] Wang Lei, Cui Yuxin, Gao Heng, et c. Research on Brushless DC Motor Control Based on Multi Strategy Fusion Algorithm [J]. Agricultural Equipment and Vehicle Engineering, 2024, 62(08):118-123.
- [8] Dehghani M., Montazeri Z., Trojovská E., et al. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2023, 259:110011.
- [9] Cao Jingchun, Lu Min. Improved Industrial Water Consumption Prediction Based on Hybrid Strategy SSA-SVM [J]. Hydroelectric Energy Science, 2023, 41(09):28-31. DOI: 10.20040/j.cnki.1000-7709.2023.20222434
- [10] Li Hongmin, Ma Yawei, Liu Ruiyu, et c. Optimization algorithm for dung beetle integrating chaotic mapping and adaptive T-distribution [J]. Software Engineering, 2024, 27(11):63-68. DOI: 10.19644/j.cnki.issn2096-1472.2024.011.013

- [11] Sheng Bin, Zhang Jun Multi strategy Improved Beetle Optimization Algorithm and Its Application [J/OL]. Manufacturing Technology and Machine Tool,1-15 [2020-02-04] — URL: <http://kns.cnki.net/kcms/detail/11.3398.th.20241216.1455.046.html>.