



# 多策略融合的黄金正弦樽海鞘群算法

## 摘要

针对樽海鞘群算法(Salp Swarm Algorithm, SSA)收敛性能差、容易陷入局部最优等问题,提出了多策略融合的黄金正弦樽海鞘群算法(Golden sine Salp Swarm Algorithm with Multi-strategy, MGSSA)。首先采用选择反向学习策略对种群中完全偏离最优个体寻优方向的个体计算选择反向解,改善种群质量;然后在跟随者位置更新阶段加入最优个体和精英均值个体引导,以加快算法收敛速度;最后根据概率选择黄金正弦算法变异策略,进一步改善解的质量,同时便于算法后期跳出局部最优。本研究在14个基准测试函数上进行实验,与其他群智能优化算法和其他改进樽海鞘群算法对比,将其应用于拉压弹簧设计问题测试解决工程优化问题的性能。结果表明:MGSSA具有较高的收敛精度和稳定性,在求解工程问题时性能良好。

## 关键词

樽海鞘群算法;选择反向学习;精英均值;黄金正弦算法

中图分类号 TP18

文献标志码 A

收稿日期 2022-12-15

资助项目 国家自然科学基金(61364015)

作者简介

丁美芳,女,硕士生,研究方向为智能优化算法.1207263694@qq.com

吴克晴(通信作者),男,博士,副教授,研究方向为泛函分析、运筹学、智能优化.wkq622@126.com

## 0 引言

优化问题在人工智能领域和现实生活中普遍存在,一直以来备受学者关注。元启发式优化算法机制灵活,仅需改变部分参数或搜索策略便可运用于各类优化问题,兼容性很高。近几十年来,学者们提出了许多元启发式优化算法,例如粒子群优化算法(Particle Swarm Optimization, PSO)<sup>[1]</sup>、鲸鱼优化算法(Whale Optimization Algorithm, WOA)<sup>[2]</sup>、郊狼优化算法(Coyote Optimization Algorithm, COA)<sup>[3]</sup>、哈里斯鹰优化算法(Harris Hawks Optimization, HHO)<sup>[4]</sup>、黑猩猩优化算法(Chimp Optimization Algorithm, ChOA)<sup>[5]</sup>等。

樽海鞘群算法(Salp Swarm Algorithm, SSA)是一种新型元启发式优化算法,由Mirjalili等<sup>[6]</sup>于2017年首次提出。相较于其他群智能优化算法,樽海鞘群算法具有结构简单、参数少、易于实现等优点。自提出以来,该算法被广泛应用于无源时差定位<sup>[7]</sup>、PMSM参数辨识<sup>[8]</sup>、图像分割<sup>[9]</sup>、特征选择<sup>[10]</sup>、训练神经网络<sup>[11]</sup>、网页排名<sup>[12]</sup>等多个领域。

然而,同其他群智能优化算法一样,樽海鞘群算法仍存在收敛速度慢、容易陷入局部最优<sup>[13]</sup>等问题。为此,国内外学者做了一些改进:文献[13]采用折射反向学习产生动态候选解,同时在跟随者阶段引入非线性递减的自适应控制因子,提升算法开采能力;文献[14]将Sigmoid函数的扰动参数应用于领导者位置更新的概率参数中,增强种群多样性,提升算法的勘探能力;文献[15]在领导者位置更新阶段引入疯狂算子,并通过一定概率随机扰动食物源的位置,从而维持领导者个体的多样性;文献[16]在领导者位置更新阶段引入控制搜索范围的非线性衰减因子,有利于算法在前期跳出局部最优;文献[17]采用平方指数和高斯模型相结合的方法,将平方指数策略引入领导者位置更新,增强了算法的收敛性能;文献[18]提出一种基于惯性权重和领导者-跟随者自适应调节方法,实现了对全局搜索和局部搜索的优化,从而大大提高了寻优精度。

上述改进算法虽然都在一定程度上改善了基本SSA的性能,但是对一些复杂度较高、规模较大的问题,算法的收敛速度和稳定性仍有改进的空间。因此,本文提出一种多策略融合的黄金正弦樽海鞘群算法(Golden sine Salp Swarm Algorithm with Multi-strategy, MGSSA)。首先,在初始阶段引入选择反向学习策略,丰富种群多样性,改善初始

1 江西理工大学 理学院,赣州,341000

解的质量.其次,将最优个体和精英均值个体引入跟随者位置更新阶段,增强整个种群之间的信息交互能力,使跟随者位置在迭代初期更侧重于跟随全局最优个体,加快收敛速度,在迭代后期更侧重于相邻个体之间的交流,提升局部开采能力.该策略很好地平衡了算法的全局勘探和局部开采能力.最后,融合黄金正弦算子和判定参数对个体位置进行变异,强化算法跳出局部极值的能力.理论分析证明了本文提出的算法并未增加算法的时间复杂度,同时选取14个基准测试函数进行数值比较实验,结果表明,在SSA中融合上述改进策略提升了SSA的收敛精度和稳定性,从而提高了算法性能.将MGSSA应用于拉压弹簧工程问题,取得了较好的结果.

## 1 樽海鞘群算法

樽海鞘群算法<sup>[6]</sup>模拟了自然界中樽海鞘群以链式结构进行捕食的过程.该算法分为领导者位置更新和跟随者位置更新两个部分.算法将每一代的最优个体设为领导者,其余个体均为跟随者.

与其他群智能优化算法类似,在求解问题时,樽海鞘群个体在 $D \times N$ 的空间中展开搜索,其中, $D$ 为给定问题的变量数, $N$ 为种群数量.这样,樽海鞘群个体位置就被存储在一个称为 $X$ 的二维矩阵中,空间中樽海鞘群的位置用 $X_n = [X_{n1}, X_{n2}, \dots, X_{nD}]^T$ 表示,在搜索空间中有一个名为 $F$ 的食物源位置用 $F_n = [F_{n1}, F_{n2}, \dots, F_{nD}]^T$ 表示.

$$X_j^1 = \begin{cases} F_j + c_1((b_u - b_l)c_2 + b_l), & c_3 \geq 0.5, \\ F_j - c_1((b_u - b_l)c_2 + b_l), & c_3 < 0.5, \end{cases} \quad (1)$$

其中: $X_j^1$ 为当前樽海鞘群链中最优个体的位置; $F_j$ 为食物源的位置; $b_u$ 和 $b_l$ 分别对应的是上下界; $c_2$ 和 $c_3$ 为(0,1)之间的随机数, $c_1$ 为2到0非线性递减的控制因子,用于平衡领导者的勘探和开采能力,计算公式如下:

$$c_1 = 2e^{-\left(\frac{t}{T}\right)^2}, \quad (2)$$

其中: $t$ 为当前迭代次数; $T$ 为最大迭代次数.

在基本的樽海鞘群算法中,每个跟随者更新的位置只与其历史位置以及前一个个体的位置有关.其位置更新公式如下:

$$X_j^{i'} = \frac{X_j^i + X_j^{i-1}}{2}, \quad (3)$$

其中: $X_j^{i'}$ 为新一代 $i$ 个体位置; $X_j^i$ 为上一代 $i$ 个体位置; $X_j^{i-1}$ 为上一代 $i$ 个体前一个体的位置.

## 2 改进樽海鞘群算法(MGSSA)

在基本SSA中,每次迭代通常取上一次迭代的最优个体作为整个种群的领导者,其余个体均为跟随者.由于领导者数目单一,种群进化方向趋同,整个算法的全局勘探能力较弱,在求解复杂问题时,容易陷入局部最优.同时,基本SSA中跟随者的位置更新仅与和它相邻的个体位置有关,这在一定程度上限制了算法的收敛速度以及搜索方向.为改善基本SSA算法存在的问题,本文提出以下改进策略:

1) 提出一种选择反向学习策略,提高种群质量,同时协助种群在搜索后期跳出局部最优;

2) 针对算法收敛速度慢的问题,在跟随者位置更新阶段引入最优个体和精英均值个体引导,吸收更多优质个体的有利信息;

3) 根据不同的变异概率,选取不同的黄金正弦变异策略,使算法在前期以较大概率向群体均值方向前进,加快算法寻优速度,后期侧重挑选随机个体进行周边搜索,便于跳出局部最优.

### 2.1 选择反向学习策略

反向学习(Opposition-Based Learning, OBL)的基本概念最初是由Tizhoosh<sup>[19]</sup>提出的,其主要思想是通过同时比较当前位置的目标函数值及其相反位置的目标函数值,保留两者中的适应度值更优的点,从而扩大搜索范围.目前已经有许多研究将反向学习策略融入到智能优化算法中,算法的求解性能均有所增强<sup>[20]</sup>.

选择反向学习(Selective Opposition-Based Learning, SOBL)是一种由反向学习启发的新型算法改进策略,其主要思想是通过将两个个体间各维度位置的距离与给定的阈值进行比较,在远距离维数大于近距离维数的情况下,选择距离大于阈值的维度进行反向学习,从而提高个体解的质量.SOBL近年来被应用于改进灰狼优化算法<sup>[21]</sup>,且经过实验证实对算法性能的改善有正向促进作用.本文将应用于樽海鞘群算法改善搜索性能.SOBL基本原理如图1所示.

在图1中,假设有一个三维空间, $(a, b, c)$ 是食物源的位置,五角星的位置为问题最优解的位置, $(a', b', c')$ 是 $(a, b, c)$ 的完全相反点.最优解位置与 $(a, b, c)$ 和 $(a', b', c')$ 的距离分别为 $d_1$ 和 $d_3$ ,同时假设 $(a', b', c)$ 是 $(a, b, c)$ 仅前二维相反的点(选择性相反),其与解的距离为 $d_2$ .从图1中可以看出, $d_2 < d_1 < d_3$ ,即在这种情况下,完全反向个体位置更偏

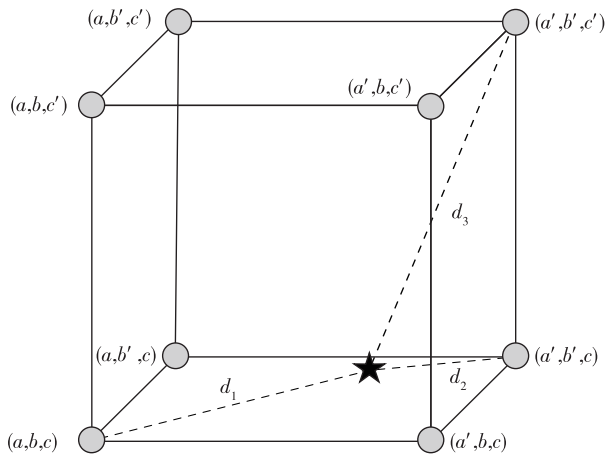


图1 SOBL原理示意图  
Fig. 1 Principle diagram of SOBL

离最优解,而选择性反向个体则给出了更好的结果.

在介绍选择反向的具体实现步骤前,需要先了解斯皮尔曼相关系数.

### 2.1.1 斯皮尔曼系数

斯皮尔曼相关性分析是一种查找两个序列之间的统计相关性的方法(此处为两个寻优个体的位置).假设有两个序列  $k_1, k_2, k_3, \dots, k_n$  和  $l_1, l_2, l_3, \dots, l_n$ , 令  $d_i = k_i - l_i, i = 1, 2, \dots, n$ , 则两者间的斯皮尔曼相关系数<sup>[22]</sup> 为

$$r = 1 - \frac{6 \sum_i (d_i)^2}{n(n^2 - 1)}, \quad (4)$$

其中,  $n$  为序列维度.在每次迭代之后,根据食物源的位置计算每个个体的斯皮尔曼系数.若相关系数为负,则表明该个体位置与食物源位置相反,继续朝该方向开采获得更有益信息的概率相对较小,进行反向学习可以加大寻找到有益信息的概率;反之,若相关系数为正,则表明该个体位置与食物源位置一致,继续朝该方向搜索有利于挖掘更多有益信息,因而无需进行反向学习.总体而言,使用斯皮尔曼系数有助于选择需要反向学习的个体,减少不必要的个体反向操作.

### 2.1.2 选择反向学习机制

对每个斯皮尔曼系数小于零的个体,给定一个阈值  $a$ , 其计算公式如下:

$$a = 2 - \frac{2t}{T}, \quad (5)$$

其中:  $t$  为算法当前迭代次数;  $T$  为算法最大迭代次数.阈值  $a$  是随迭代次数线性减少的,该值确定该个体位置是否接近食物源.对于当前个体和食物源的

每个维度  $j$ , 使用式(6) 计算它们之间的差值:

$$D(j) = |X(j) - F(j)|, \quad (6)$$

其中:  $X(j)$  为当前个体第  $j$  维位置;  $F(j)$  为食物源第  $j$  维位置.如果差值  $D(j)$  大于阈值  $a$ , 则差值较大的维数  $g_{no}$  将增加.设总维数为  $n$ .若  $(n - g_{no}) < g_{no}$ , 则对个体  $X$  和  $F(j)$  差值大于阈值的维度进行反向学习策略:

$$\tilde{X}(j) = b_u(j) + b_l(j) - X(j), \quad (7)$$

其中:  $\tilde{X}(j)$  为反向个体  $X$  第  $j$  维位置;  $b_u(j)$  为第  $j$  维的上界;  $b_l(j)$  为第  $j$  维的下界.如果适应度值更优,则用反向个体替换原个体,否则,保留原个体.

选择反向学习与传统反向学习相比,优点在于:用斯皮尔曼系数筛选与食物源位置相反的个体进行反向,保留了相对优质个体的同时减少计算量,通过阈值决定反向的维度,细化反向的方向,使个体搜索更靠近最优方向.

## 2.2 引入最优个体和精英均值个体的跟随者位置更新

在基本的樽海鞘群算法中,除了一个领导者之外,其余的都是跟随者.跟随者的搜索方向仅和自身位置及与它相邻的个体位置有关,这在一定程度上影响了算法的收敛速度及搜索方向.为了使跟随者能更快地搜索到有益信息,本文设计了一种受种群最优个体和精英群均值个体影响的跟随者搜索策略.

文献[23]提出一种自适应领导者结构,本文在此基础上提出由 1 自适应增加到  $N/2$  的线性精英群.每次迭代的精英种群数量如下:

$$m = \lceil (t/T) \times (N/2) \rceil, \quad (8)$$

其中:  $t$  为算法当前迭代次数;  $T$  为算法最大迭代次数;  $N$  为种群规模;  $\lceil \cdot \rceil$  表示向上取整.将该精英群数目记为  $m$ , 每个精英个体记为  $E_j^i$ .通过对精英群求解平均值,可以得到精英均值个体:

$$E_j^{\text{mean}} = \text{avg} \sum_{i=1}^m E_j^i. \quad (9)$$

引入种群最优个体和精英均值个体位置更新公式如下:

$$X_1 = F_j - X_j^i, \quad X_2 = E_j^{\text{mean}} - X_j^i,$$

$$X_j^i = \frac{1}{4} \times \left( X_j^i + X_j^{i-1} + \frac{c_1}{2} \times \text{rand}() \times (X_1 + X_2) \right), \quad (10)$$

其中,  $c_1$  更新参照式(2).

在迭代初期,个体搜索比较盲目,此时跟随者的

位置更新融合了种群最优个体和精英群均值个体位置,可以使跟随者在前期开发出更多更好的解.而在算法后期,随着参数  $c_1$  的逐渐减小,精英个体对跟随者的影响也会更小,而且此时种群搜索范围也更小,即使是跟随者,蕴含的有益信息量也很大,此时削弱精英个体的影响,对于整个算法而言,可以拓宽算法的勘探能力,这种机制有助于算法在迭代后期跳出局部最优.

### 2.3 黄金正弦算子变异策略

黄金正弦算法是2017年由Tanyildizi等<sup>[24]</sup>提出的新型智能算法.受正弦函数启发,黄金正弦算法取黄金分割数使粒子在更优的搜索范围进行寻优,加快了算法的收敛速度.选取的黄金正弦算子具体公式如下:

$$X_i(t+1) = X_i(t) | \sin R_1 | + R_2 \sin(R_1) | aX_{\text{best}} - bX_i(t) |, \quad (11)$$

其中:  $R_1$  为0到  $2\pi$  的随机数,决定个体  $i$  下一代位置更新的移动步长;  $R_2$  为0到  $\pi$  的随机数,决定个体  $i$  下次更新的方向;  $a$  和  $b$  为黄金分割系数,影响着粒子的搜索空间,对搜索空间进行了一个更优的选择,引领粒子趋近于更优值.计算公式如下:

$$\begin{aligned} a &= -\pi(1-\tau) + \pi\tau, \\ b &= -\pi\tau + \pi(1-\tau), \end{aligned} \quad (12)$$

其中,  $\tau$  为黄金分割比,值为  $(\sqrt{5}-1)/2$ .

本文利用黄金正弦算子的优势,对樽海鞘群算法个体进行变异,打开更优的搜索空间,向更优解靠拢.为避免陷入局部最优,分概率采用均值策略和随机策略选取粒子进行变异操作.具体公式如下:

$$X_j^i = \begin{cases} E_j^{\text{mean}} | \sin R_1 | + R_2 \sin(R_1) | aF_j - bE_j^{\text{mean}} |, & p \leq c_1/2, \\ X_j^{\text{index}} | \sin R_1 | + R_2 \sin(R_1) | aF_j - bX_j^{\text{index}} |, & p > c_1/2, \end{cases} \quad (13)$$

其中:  $E_j^{\text{mean}}$  为精英均值个体位置;  $X_j^{\text{index}}$  为随机个体位置;  $p$  是一个(0,1)之间的随机数.

为了防止过早收敛并达到全局最优,需要在勘探和开采之间保持良好的平衡,这种平衡就是通过参数  $p$  实现的.在算法迭代前期,种群相对分散,精英均值个体蕴含较多有利信息,在周边搜寻可以加快收敛速度,而此时  $c_1$  较大,也使得在精英均值个体周围搜索的概率更大.随着种群不断进化,后期  $p > c_1/2$  的概率更大,算法更多在随机选择的个体周围

进行搜索,有利于跳出局部最优.

在进行黄金正弦变异后,将变异后的个体适应度值与原种群对应个体适应度值对比,保留适应度值小的,更新其位置.

### 2.4 MGSSA 算法伪代码

MGSSA 算法伪代码如下:

- 1) 设置种群规模  $N$ , 维数  $D$ , 最大迭代次数  $T$
- 2) 随机初始化种群  $\{x_i, i = 1, 2, \dots, N\}$
- 3) 计算初代种群适应度并排序, 记录当前最优个体位置 FoodPosition
- 4) While  $t \leq T$
- 5) 根据式(5)更新阈值  $a$
- 6) 筛选种群进行选择反向学习, 计算适应度, 比较学习后个体与之前的个体的适应度值大小, 选更小的个体作为新的位置
- 7) for  $i = 1$  to  $N$
- 8) if  $(i == 1)$
- 9) 根据式(1)更新领导者个体位置
- 10) else
- 11) 根据式(10)更新跟随者位置
- 12) end
- 13) 更新变异概率  $p$
- 14) if  $p \leq c_1/2$
- 15) 根据式(13)对每个个体  $x_i$  位置进行变异
- 16) else
- 17) 根据式(13)对每个个体  $x_i$  位置进行变异
- 18) end
- 19) end
- 20) 计算每个个体的适应度值, 更新 FoodPosition
- 21) end while
- 22) 输出最优位置 FoodPosition 及其适应度值 FoodFitness

### 2.5 时间复杂度分析

在基本的SSA中,假设樽海鞘群种群规模为  $N$ , 解空间为  $n$  维, 求解目标函数所需的执行时间为  $f(n)$ . 根据文献[18], SSA算法的时间复杂度为

$$T(n) = O(n + f(n)).$$

在MGSSA算法中,参数初始化执行时间  $\eta_0$ , 每维生成随机数时间  $\eta_1$ , 求解目标函数执行时间  $f(n)$  对适应度值排序时间为  $\eta_2$ , 则初始化阶段时间复杂度为

$$T_1 = O(\eta_0 + N(n\eta_1 + f(n) + \eta_2) = O(n + f(n)).$$

进行选择反向学习阶段,根据式(5) 计算阈值  $a$  时间为  $\eta_3$ ,根据式(6) 计算每个跟随者与食物源之间的距离所需时间为  $\eta_4$ ,根据式(4) 根计算对应斯皮尔曼系数时间为  $\eta_5$ ,假设斯皮尔曼系数为负的个体数为  $n_0(n_0 < N)$ ,根据式(7) 选择个体反向学习的时间为  $\eta_6$ ,则该阶段时间复杂度为

$$T_2 = O(\eta_3 + N(n\eta_4 + \eta_5) + n_0n\eta_6) = O(n),$$

在领导者位置更新时间复杂度为

$$T_3 = O(n).$$

在跟随者位置更新阶段,跟随者数目为  $N - 1$ ,设由式(10) 对跟随者每一维进行更新的时间为  $\eta_7$ ,则这一阶段的时间复杂度为

$$T_4 = O((N - 1)n\eta_7) = O(n).$$

在黄金正弦算子变异阶段,生成变异概率  $p$  时间为  $\eta_8$ ,根据式(13) 对每个个体变异的时间为  $\eta_9$ ,则这一阶段时间复杂度为

$$T_5 = O(\eta_8 + Nn\eta_9) = O(n).$$

在边界处理和更新食物源位置阶段,时间复杂度与基本 SSA 一致,为

$$T_6 = O(n + f(n)).$$

综上可得 MGSSA 算法的总时间复杂度为

$$T(n) = T_1 + T(T_2 + T_3 + T_4 + T_5 + T_6) = O(n + f(n)).$$

由此可知, MGSSA 与基本 SSA 相比,并未增加时间复杂度.

### 3 实验仿真及结果分析

#### 3.1 仿真实验设计及实验环境

为了评估 MGSSA 的性能,实验从文献[25] 中选取了 14 个基准测试函数,包括 7 个单峰函数  $f_1 \sim f_7$  和 7 个多峰函数  $f_8 \sim f_{14}$ ,在表 1 中列出了各函数的名称及其他相关属性.将 MGSSA 与 14 种算法进

表 1 基准测试函数

Table 1 Benchmark test functions

函数公式	函数名称	维度	范围	最优值
$f_1(x) = \sum_{i=1}^n x_i^2$	Sphere	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^n  x_i ^{i+1}$	Powellsum	30	$[-1, 1]$	0
$f_3(x) = \sum_{i=1}^n  x_i $	Schwefel's 2.20	30	$[-10, 10]$	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	Schwefel's 2.21	30	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{n-1}  x_i + 0.5 ^2$	Step	30	$[-100, 100]$	0
$f_6(x) = \sum_{i=1}^{n-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$	Brown	30	$[-1, 4]$	0
$f_7(x) = \sum_{i=1}^n ix_i^2$	Sum Squares	30	$[-10, 10]$	0
$f_8(x) = 418.9829n - \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	Schwefel's 2.26	30	$[-500, 500]$	0
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	Rastigin	30	$[-5.12, 5.12]$	0
$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	Ackley	30	$[-32, 32]$	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	Griewank	30	$[-600, 600]$	0
$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	Penalized	30	$[-50, 50]$	0
$f_{13}(x) = 1 - \cos\left(2\pi\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) + 0.1\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}$	Salomon	30	$[-100, 100]$	0
$f_{14}(x) = \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	Styblinski-Tang	30	$[-5, 5]$	-1 174.985

行比较,包括3种单策略改进算法、6种最近几年提出的SSA改进算法和5种群智能优化算法.最后将改进算法应用于工程约束优化问题探究其求解效果.为了验证数值的有效性,对14个基准函数运行30次的MGSSA和其他比较算法的结果进行非参数Wilcoxon秩和检验,根据每次运行获得的均值,对实验结果进行统计分析,帮助判断各算法与最优算法之间是否具有显著性差异,从而更好地评判算法的性能.其中显著性水平校正为0.05,将均值最小的算法视为最优算法,记作NA.若均值相同,则认为标准差越小的算法越好.若 $p$ 值小于0.05,则认为该算法与最优算法存在显著性差异,若大于0.05,则该算法与最优算法无显著性差异.符号“+”表示提出的算法得到的结果优于比较算法,而“-”表示提出的算法

得到的结果相较于比较算法更差,符号“=”表明彼此间没有显著差异.因此表中“+/-/-”行给出算法在14个基准函数上优/等/劣的个数.所有实验都是在操作系统为Windows 11的PC机上用Matlab2019a软件实现的,PC机配置AMD R5 3500U处理器,主频2.10 GHz CPU,内存8 GB.

### 3.2 改进策略分析

为探究不同策略对算法改进的影响程度,对仅添加精英均值个体策略的改进算法ESSA,仅融合黄金正弦变异的算法GSSA和加入选择反向学习策略的算法SOSSA与MGSSA进行对比实验.

4种算法独立运行30次的统计结果如表2所示,部分函数收敛曲线如图2所示.可以看出,在14

表2 不同改进策略实验测试结果

Table 2 Experimental test results of different improvement strategies

函数	指标	ESSA	GSSA	SOSSA	MGSSA	函数	指标	ESSA	GSSA	SOSSA	MGSSA
$f_1$	最优值	2.84E-28	0.00E+00	4.51E+00	0.00E+00	$f_8$	最优值	1.28E+02	1.28E-05	1.33E+02	1.27E-05
	均值	1.11E-27	0.00E+00	3.20E+01	0.00E+00		均值	1.70E+02	3.06E-05	1.80E+02	1.27E-05
	标准差	8.60E-28	0.00E+00	2.79E+01	0.00E+00		标准差	2.19E+01	2.93E-05	2.36E+01	2.36E-10
	$p$ -value	1.21E-12	NA	1.21E-12	NA		$p$ -value	3.02E-11	3.02E-11	3.02E-11	NA
$f_2$	最优值	5.28E-39	0.00E+00	1.28E-08	0.00E+00	$f_9$	最优值	0.00E+00	0.00E+00	2.29E+01	0.00E+00
	均值	3.07E-35	1.34E-21	3.35E-06	0.00E+00		均值	0.00E+00	0.00E+00	4.63E+01	0.00E+00
	标准差	5.93E-35	7.35E-21	3.43E-06	0.00E+00		标准差	0.00E+00	0.00E+00	1.48E+01	0.00E+00
	$p$ -value	1.72E-12	2.98E-11	1.72E-12	NA		$p$ -value	NA	NA	1.21E-12	NA
$f_3$	最优值	7.25E-14	0.00E+00	3.49E+01	0.00E+00	$f_{10}$	最优值	2.66E-15	-8.88E-16	3.21E+00	-8.88E-16
	均值	1.58E-13	0.00E+00	6.96E+01	0.00E+00		均值	7.64E-15	-8.88E-16	6.34E+00	-8.88E-16
	标准差	4.65E-14	0.00E+00	1.97E+01	0.00E+00		标准差	3.44E-15	0.00E+00	1.36E+00	0.00E+00
	$p$ -value	1.21E-12	NA	1.21E-12	NA		$p$ -value	1.21E-12	NA	1.21E-12	NA
$f_4$	最优值	5.42E-15	0.00E+00	6.67E+00	0.00E+00	$f_{11}$	最优值	0.00E+00	0.00E+00	1.28E-01	0.00E+00
	均值	1.27E-14	0.00E+00	1.32E+01	0.00E+00		均值	0.00E+00	0.00E+00	7.21E-01	0.00E+00
	标准差	5.54E-15	0.00E+00	2.82E+00	0.00E+00		标准差	0.00E+00	0.00E+00	2.20E-01	0.00E+00
	$p$ -value	1.21E-12	NA	1.21E-12	NA		$p$ -value	NA	NA	1.21E-12	NA
$f_5$	最优值	7.31E-09	1.25E-11	3.84E+00	5.99E-09	$f_{12}$	最优值	2.82E-11	2.54E-12	3.15E+00	2.48E-11
	均值	1.02E-08	1.54E-07	2.60E+01	1.00E-08		均值	5.20E-11	4.09E-09	8.60E+00	4.42E-11
	标准差	1.73E-09	7.06E-07	1.57E+01	2.22E-09		标准差	1.62E-11	7.25E-09	4.50E+00	1.20E-11
	$p$ -value	4.20E-04	6.05E-07	3.02E-11	NA		$p$ -value	7.24E-02	3.32E-06	3.02E-11	NA
$f_6$	最优值	6.95E-31	0.00E+00	7.18E-02	0.00E+00	$f_{13}$	最优值	1.66E-15	0.00E+00	2.70E+00	0.00E+00
	均值	2.85E-30	0.00E+00	2.33E+00	0.00E+00		均值	3.75E-15	0.00E+00	3.63E+00	0.00E+00
	标准差	1.90E-30	0.00E+00	5.66E+00	0.00E+00		标准差	1.21E-15	0.00E+00	6.06E-01	0.00E+00
	$p$ -value	1.21E-12	NA	1.21E-12	NA		$p$ -value	1.21E-12	NA	1.21E-12	NA
$f_7$	最优值	4.29E-29	0.00E+00	2.11E+00	0.00E+00	$f_{14}$	最优值	-1.12E+03	-1.17E+03	-1.05E+03	-1.17E+03
	均值	1.91E-28	0.00E+00	2.06E+01	0.00E+00		均值	-1.01E+03	-1.14E+03	-9.84E+02	-1.17E+03
	标准差	1.13E-28	0.00E+00	1.30E+01	0.00E+00		标准差	5.37E+01	8.08E+01	3.97E+01	9.44E-11
	$p$ -value	1.21E-12	NA	1.21E-12	NA		$p$ -value	3.02E-11	3.02E-11	3.02E-11	NA

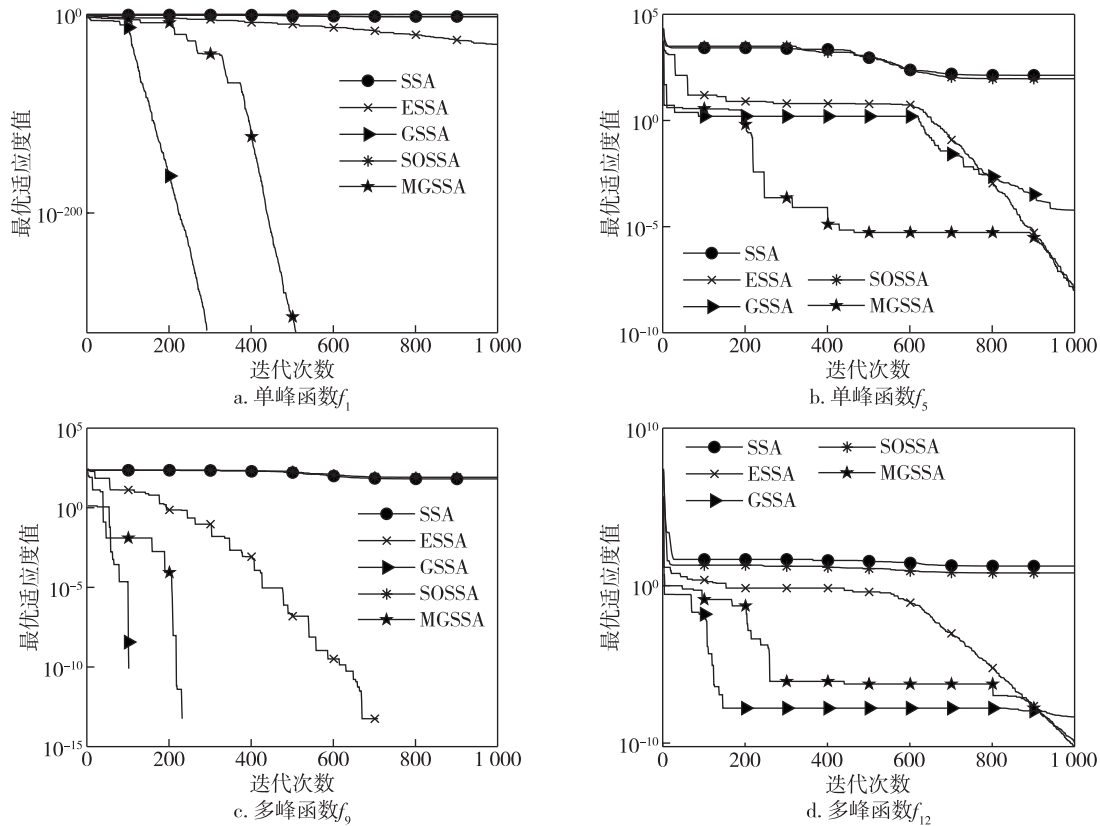


图2 不同策略函数收敛曲线

Fig. 2 Convergence curves of different strategy functions

个函数中,ESSA、GSSA 和 MGSSA 分别在 3、9、14 个函数中寻到了最优,而 SOSSA 在 14 个函数的求解效果均差于其他 3 种算法.这表明黄金正弦融合策略与精英均值个体对标准 SSA 的改进起较大作用,选择反向学习策略对标准 SSA 的改进帮助较弱.通过图 2 也可以发现,与标准 SSA 相比,选择反向学习策略在求解单峰函数时,虽然收敛效果方面的提升并不明显,但在一定程度上改善了算法的求解精度.因此,将其用于算法开始,改善种群多样性十分合适.同时可以看出在函数  $f_{12}$  收敛曲线中,GSSA 在 200 代左右就陷入了局部停滞,而 MGSSA 在 800 代左右仍然可以跳出局部最优,进一步寻找更优解,其搜索性能明显更佳.综合表 2 数据,MGSSA 在 14 个函数上表现均优其他 3 种单策略改进算法,这表明将 3 种策略融合求解问题可以有效提升算法的求解性能.

### 3.3 MGSSA 与其他群智能算法比较

实验选取 5 个群智能算法进行比较,分别为 SSA<sup>[6]</sup>、GoldSA<sup>[24]</sup>、HHO<sup>[4]</sup>、WOA<sup>[2]</sup>、ChOA<sup>[5]</sup>,为保证实验的公平性,相应算法的参数根据对应参考文献进行设置.本文的 MGSSA 与这 5 种算法比较的结

果数据如表 3 所示.

根据表 3 的结果,MGSSA 可以在 7 个单峰函数中找到 6 个函数的精确最优值,表明该算法具有较强的鲁棒性和较高的收敛精度.对于唯一一个未寻到精确最优值的函数  $f_5$ ,其最优值、均值和标准差均优于其他 5 种算法.因此,MGSSA 在求解单峰函数时具有良好的稳定性和较强的开采能力.对于 7 个多峰函数,MGSSA 寻到了 4 个精确最优值,在另外 3 个函数上也找到了接近全局最优值,表明 MGSSA 有较强的摆脱局部最优的能力.对于函数  $f_8$  和函数  $f_{14}$ ,GoldSA 和 HHO 均寻到相似和相同最优值,但 MGSSA 的均值和标准差优于 GoldSA 和 HHO,表明 MGSSA 的稳定性更强.从优/等/劣的结果中也可以看出,MGSSA 分别在 14、5、11、13、14 个函数上结果优于 SSA、GoldSA、HHO、WOA、ChOA,在 6 种算法中总体求解性能最佳.结果表明 MGSSA 在求解单峰函数中有较强的开采能力,在求解多峰函数时可以有效避免局部最优,有较强的勘探能力.图 3 是各算法部分函数收敛曲线,可以更直观地观察算法的求解性能.

表 3 其他群智能算法比较实验结果

Table 3 Experimental result comparison with other swarm intelligence algorithms

函数	指标	MGSSA	SSA	GoldSA	HHO	WOA	ChOA
$f_1$	最优值	0.00E+00	6.61E+00	0.00E+00	7.00E-265	5.01E-225	3.03E-37
	均值	0.00E+00	4.75E+01	0.00E+00	8.23E-224	1.23E-201	6.34E-23
	标准差	0.00E+00	2.88E+01	0.00E+00	0.00E+00	0.00E+00	1.53E-22
	$p$ -value	NA	1.21E-12	NA	1.21E-12	1.21E-12	1.21E-12
$f_2$	最优值	0.00E+00	1.91E-07	0.00E+00	0.00E+00	0.00E+00	1.58E-79
	均值	0.00E+00	3.90E-06	0.00E+00	2.15E-296	7.03E-296	1.79E-53
	标准差	0.00E+00	5.40E-06	0.00E+00	0.00E+00	0.00E+00	9.67E-53
	$p$ -value	NA	1.21E-12	NA	1.21E-12	1.21E-12	1.21E-12
$f_3$	最优值	0.00E+00	3.49E+01	0.00E+00	2.98E-134	1.74E-146	2.89E-25
	均值	0.00E+00	7.31E+01	0.00E+00	1.12E-120	8.42E-135	8.17E-14
	标准差	0.00E+00	2.45E+01	0.00E+00	5.12E-120	3.52E-134	3.53E-13
	$p$ -value	NA	1.21E-12	NA	1.21E-12	1.21E-12	1.21E-12
$f_4$	最优值	0.00E+00	7.50E+00	0.00E+00	3.50E-131	5.27E-04	3.87E-09
	均值	0.00E+00	1.34E+01	1.6948e-316	4.96E-120	3.20E+01	3.00E-05
	标准差	0.00E+00	2.49E+00	0.00E+00	1.22E-119	3.34E+01	3.00E-05
	$p$ -value	NA	1.21E-12	1.61E-01	1.21E-12	1.21E-12	1.21E-12
$f_5$	最优值	1.49E-09	6.84E+00	7.16E-08	3.06E-09	1.69E-03	1.81E+00
	均值	9.61E-09	3.00E+01	3.38E-05	9.43E-06	5.16E-03	2.55E+00
	标准差	2.79E-09	2.24E+01	5.18E-05	1.44E-05	2.46E-03	3.47E-01
	$p$ -value	NA	3.02E-11	3.02E-11	5.07E-10	3.02E-11	3.02E-11
$f_6$	最优值	0.00E+00	1.12E-01	0.00E+00	1.04E-261	1.53E-227	8.82E-38
	均值	0.00E+00	1.05E+00	0.00E+00	9.87E-238	4.45E-208	5.43E-26
	标准差	0.00E+00	1.70E+00	0.00E+00	0.00E+00	0.00E+00	1.42E-25
	$p$ -value	NA	1.21E-12	NA	1.21E-12	1.21E-12	1.21E-12
$f_7$	最优值	0.00E+00	3.98E+00	0.00E+00	1.79E-264	8.73E-226	1.18E-36
	均值	0.00E+00	2.10E+01	0.00E+00	3.30E-234	6.60E-207	1.22E-22
	标准差	0.00E+00	1.15E+01	0.00E+00	0.00E+00	0.00E+00	4.62E-22
	$p$ -value	NA	1.21E-12	NA	1.21E-12	1.21E-12	1.21E-12
$f_8$	最优值	1.27E-05	1.41E+02	1.29E-05	1.34E-05	1.82E-03	2.86E+02
	均值	1.27E-05	1.83E+02	1.29E-03	2.39E-03	7.79E+01	3.03E+02
	标准差	2.05E-10	2.17E+01	1.76E-03	3.51E-03	5.99E+01	8.80E+00
	$p$ -value	NA	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
$f_9$	最优值	0.00E+00	1.58E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	均值	0.00E+00	4.42E+01	0.00E+00	0.00E+00	0.00E+00	2.00E+00
	标准差	0.00E+00	1.94E+01	0.00E+00	0.00E+00	0.00E+00	2.39E+00
	$p$ -value	NA	1.21E-12	NA	NA	NA	6.23E-10
$f_{10}$	最优值	-8.88E-16	4.43E+00	-8.88E-16	-8.88E-16	-8.88E-16	2.00E+01
	均值	-8.88E-16	6.83E+00	-8.88E-16	-8.88E-16	2.19E-15	2.00E+01
	标准差	0.00E+00	1.23E+00	0.00E+00	0.00E+00	2.23E-15	1.23E-03
	$p$ -value	NA	1.21E-12	NA	NA	9.16E-09	1.21E-12
$f_{11}$	最优值	0.00E+00	7.87E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	均值	0.00E+00	3.09E-01	0.00E+00	0.00E+00	1.69E-03	1.12E-02
	标准差	0.00E+00	1.64E-01	0.00E+00	0.00E+00	9.27E-03	1.97E-02
	$p$ -value	NA	1.21E-12	NA	NA	3.34E-01	3.13E-04



续表3

函数	指标	MGSSA	SSA	GoldSA	HHO	WOA	ChOA
$f_{12}$	最优值	2.96E-11	2.99E+00	3.53E-09	3.03E-09	1.13E-04	1.54E-01
	均值	5.05E-11	7.56E+00	7.48E-07	9.34E-07	1.64E-03	3.69E-01
	标准差	1.51E-11	2.84E+00	2.00E-06	1.70E-06	2.69E-03	2.09E-01
	p-value	NA	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
$f_{13}$	最优值	0.00E+00	2.40E+00	0.00E+00	1.62E-131	2.78E-103	9.99E-02
	均值	0.00E+00	3.34E+00	0.00E+00	2.15E-119	1.47E-01	1.09E-01
	标准差	0.00E+00	5.77E-01	0.00E+00	8.91E-119	7.30E-02	2.82E-02
	p-value	NA	1.20E-12	NA	1.21E-12	1.14E-12	1.21E-12
$f_{14}$	最优值	-1.17E+03	-1.06E+03	-1.17E+03	-1.17E+03	-1.17E+03	-7.52E+02
	均值	-1.17E+03	-9.89E+02	-1.17E+03	-1.17E+03	-1.16E+03	-6.65E+02
	标准差	6.71E-11	3.87E+01	6.42E-04	1.56E-03	4.49E+01	3.89E+01
	p-value	NA	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
+/-/+			14/0/0	5/9/0	11/3/0	13/1/0	14/0/0

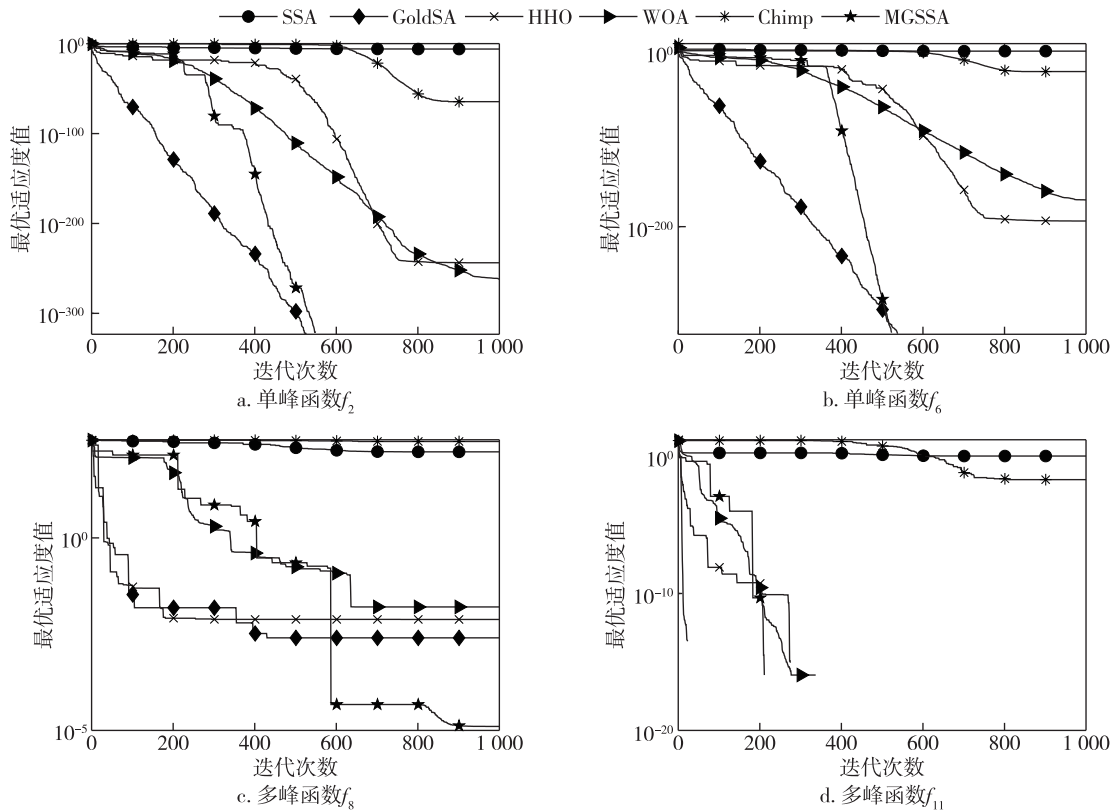


图3 群智能优化算法函数收敛曲线

Fig. 3 Convergence curves of functions for swarm intelligence optimization algorithms

MGSSA 可以有更好的测试结果,主要是均值个体策略优化了跟随者的搜索方向,从而改善算法的开采能力,在算法后期又有黄金正弦扰动帮助其跳出局部最优.SOBL 优化了整体种群位置,加快种群的收敛速度同时也改善解的质量.

### 3.4 MGSSA 与其他改进 SSA 算法

为进一步测试 MGSSA 的性能,使用 6 种其他的

SSA 改进算法与 MGSSA 对比,包括 CASSA<sup>[15]</sup>、RDSSA<sup>[16]</sup>、RCSSA<sup>[13]</sup>、ALSSA<sup>[18]</sup>、AIWSSA<sup>[26]</sup> 和 AGHSSA<sup>[27]</sup>.所有算法相关参数均按相应参考文献设置,在 14 个基准函数上进行测试,各算法独立运行 30 次,在  $\alpha=0.05$  显著水平下通过 Wilcoxon 秩和检验获得的实验结果如表 4 所示.

表4 SSA改进算法实验测试结果

Table 4 Experimental test results of improved Salp Swarm Algorithms

函数	统计结果	MGSSA	RDSSA	RCSSA	CASSA	ALSSA	AIWSSA	AGHSSA
$f_1$	最优值	0.00E+00	1.24E-65	1.57E-311	7.02E-95	2.26E-131	0.00E+00	0.00E+00
	均值	0.00E+00	1.12E-62	0.00E+00	1.13E-94	2.66E-122	0.00E+00	0.00E+00
	标准差	0.00E+00	3.74E+01	2.46E-62	0.00E+00	1.54E-95	1.21E-121	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_2$	最优值	0.00E+00	7.09E-78	1.68E-321	1.08E-96	2.63E-198	0.00E+00	0.00E+00
	均值	0.00E+00	1.45E-71	8.97E-317	3.51E-96	7.05E-184	0.00E+00	0.00E+00
	标准差	0.00E+00	4.99E-71	0.00E+00	1.91E-96	0.00E+00	0.00E+00	0.00E+00
	p-value	NA	2.37E-12	1.24E-09	2.37E-12	2.37E-12	NA	NA
$f_3$	最优值	0.00E+00	2.57E-32	1.89E-156	4.72E-47	5.36E-67	0.00E+00	0.00E+00
	均值	0.00E+00	2.07E-31	3.85E-154	5.36E-47	2.00E-63	0.00E+00	0.00E+00
	标准差	0.00E+00	1.69E-31	3.10E-154	3.34E-48	3.53E-63	0.00E+00	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_4$	最优值	0.00E+00	2.40E-33	2.35E-157	2.22E-48	2.08E-69	0.00E+00	0.00E+00
	均值	0.00E+00	2.20E-32	1.44E-155	2.95E-48	4.33E-63	0.00E+00	0.00E+00
	标准差	0.00E+00	2.93E-32	1.05E-155	4.89E-49	1.79E-62	0.00E+00	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_5$	最优值	1.49E-09	4.93E+00	5.84E-09	6.67E-07	6.27E+00	7.89E-09	1.52E-02
	均值	9.05E-09	5.95E+00	8.61E-09	9.21E-07	7.02E+00	1.05E-08	5.68E-01
	标准差	2.98E-09	4.54E-01	1.81E-09	2.15E-07	3.64E-01	1.43E-09	4.56E-01
	p-value	1.41E-01	3.02E-11	NA	3.02E-11	3.02E-11	1.04E-04	3.02E-11
$f_6$	最优值	0.00E+00	1.92E-68	3.85E-303	1.58E-94	5.35E-135	0.00E+00	0.00E+00
	均值	0.00E+00	3.42E-65	7.70E-303	2.00E-94	4.33E-125	0.00E+00	0.00E+00
	标准差	0.00E+00	7.97E-65	0.00E+00	2.16E-95	1.41E-124	0.00E+00	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_7$	最优值	0.00E+00	2.92E-66	1.56E-312	1.36E-93	1.45E-133	0.00E+00	0.00E+00
	均值	0.00E+00	1.07E-63	0.00E+00	1.60E-93	3.21E-124	0.00E+00	0.00E+00
	标准差	0.00E+00	2.55E-63	0.00E+00	1.45E-94	1.73E-123	0.00E+00	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_8$	最优值	1.27E-05	3.00E+02	1.08E+02	1.22E+02	2.83E+02	7.24E+00	2.17E+01
	均值	1.27E-05	3.33E+02	1.65E+02	1.71E+02	3.24E+02	1.66E+02	1.67E+02
	标准差	2.05E-10	1.37E+01	3.60E+01	2.57E+01	1.67E+01	6.07E+01	5.35E+01
	p-value	NA	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
$f_9$	最优值	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.59E-11	0.00E+00	0.00E+00
	均值	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.33E+00	0.00E+00	0.00E+00
	标准差	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.90E+00	0.00E+00	0.00E+00
	p-value	NA	NA	NA	NA	1.21E-12	NA	NA
$f_{10}$	最优值	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16
	均值	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16	-8.88E-16
	标准差	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	p-value	NA	NA	NA	NA	NA	NA	NA
$f_{11}$	最优值	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	均值	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	p-value	NA	NA	NA	NA	NA	NA	NA

续表4

函数	统计结果	MGSSA	RDSSA	RCSSA	CASSA	ALSSA	AIWSSA	AGHSSA
$f_{12}$	最优值	2.05E-11	3.76E-01	2.58E-11	1.32E-08	9.18E-01	3.07E-11	2.04E-05
	均值	4.27E-11	7.51E-01	4.56E-11	2.44E-08	1.52E+00	4.73E-11	6.09E-03
	标准差	1.28E-11	1.70E-01	1.66E-11	8.93E-09	1.56E-01	1.13E-11	8.69E-03
	p-value	NA	3.02E-11	7.17E-01	3.02E-11	3.02E-11	3.33E-01	3.02E-11
$f_{13}$	最优值	0.00E+00	3.60E-34	1.25E-157	9.12E-49	9.99E-02	0.00E+00	0.00E+00
	均值	0.00E+00	1.20E-32	8.66E-156	1.03E-48	9.99E-02	0.00E+00	0.00E+00
	标准差	0.00E+00	1.54E-32	6.05E-156	7.53E-50	3.80E-06	0.00E+00	0.00E+00
	p-value	NA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NA	NA
$f_{14}$	最优值	-1 174.985	-647.3527	-1 174.985	-1 104.3014	-707.0761	-1 174.985	-1 174.985
	均值	-1 174.985	-581.7705	-1 025.1357	-1 036.9163	-646.523	-1 059.0639	-1 174.9812
	标准差	6.71E-11	3.10E+01	4.32E+01	3.21E+01	4.16E+01	1.41E+02	1.06E-02
	p-value	NA	3.02E-11	5.57E-10	3.02E-11	3.02E-11	3.65E-08	3.02E-11
+/-/-			11/3/0	9/4/1	11/3/0	12/2/0	4/10/0	4/10/0

从表4可以看出:在14个基准测试函数中, MGSSA在6个单峰函数和7个多峰函数上具有最佳的平均值;针对未寻到精确最优值的函数 $f_5$ ,也优于除RCSSA以外的5个改进算法,总体求解效果最优.同时可以观察到,与同样融合了黄金正弦算法的AGHSSA相比, MGSSA在1个单峰函数( $f_5$ )和3个多峰函数( $f_8, f_{12}, f_{14}$ )上寻到了更优,而对比RDSSA、RCSSA、CASSA、ALSSA和AIWSSA,也分别在11、9、

11、12和4个函数上寻到了更优解,表明精英均值个体策略与选择反向学习策略在协助算法求解多峰函数时发挥了重要作用.因此,3种策略同时使用可有效提升SSA求解多峰函数的性能.

为了直观地观察7种SSA改进算法的收敛性能,本文选取几个具有代表性的函数绘制寻优收敛曲线如图4所示.

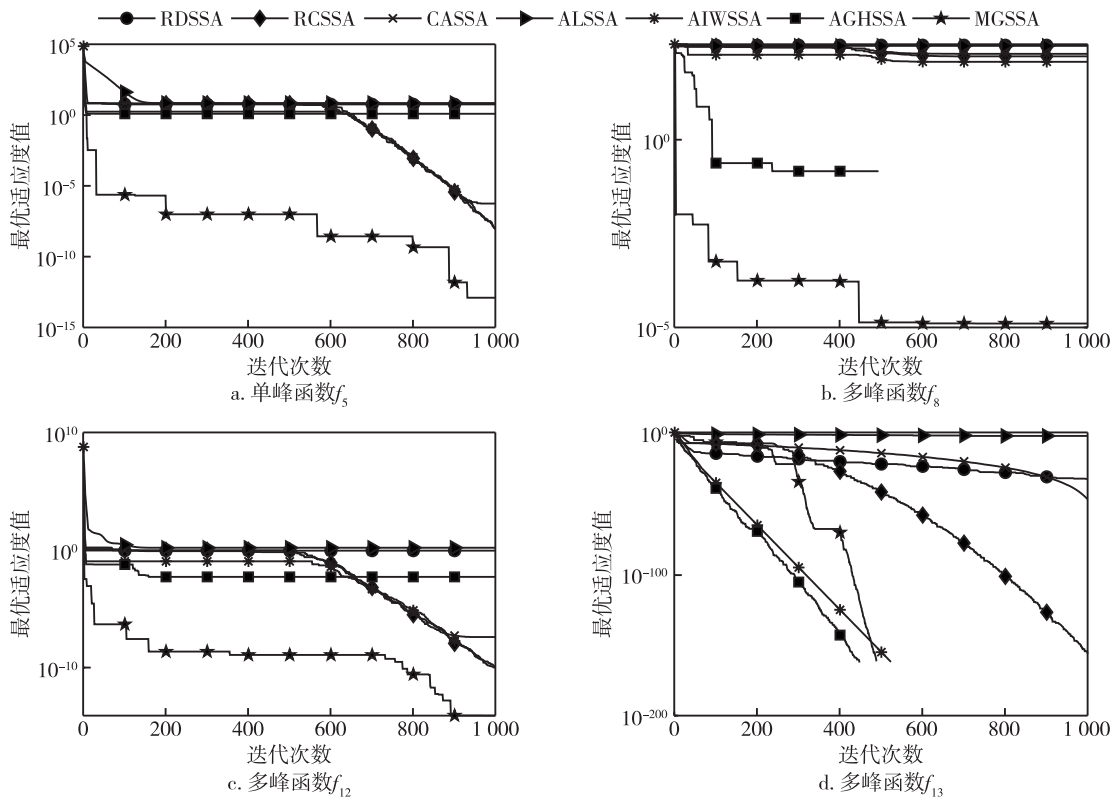


图4 改进SSA函数收敛曲线

Fig. 4 Convergence curves of functions for improved Salp Swarm Algorithms

在单峰函数  $f_5$  和多峰函数  $f_{12}$  上,在 CASSA、ALSSA、AGHSSA 和 RDSSA 陷入局部最优的情况下, MGSSA 能很好地跳出局部最优,找寻更优解,其勘探能力较强.同时在  $f_8, f_{13}$  两个函数上,也可以观察到 MGSSA 在搜索过程中收敛速度和跳出局部最优的能力十分显著.

### 3.5 工程约束优化问题求解

为了更好地验证 MGSSA 在求解工程约束问题时的性能,本文将 MGSSA 与 CASSA<sup>[15]</sup>、RDSSA<sup>[16]</sup>、RCSSA<sup>[13]</sup>、ALSSA<sup>[18]</sup>、AIWSSA<sup>[26]</sup> 和 AGHSSA<sup>[27]</sup> 6 种改进 SSA 算法和基本 SSA 应用于拉压弹簧设计问题中的求解.同时利用罚函数法,在目标函数中加入惩罚项,将约束优化问题转化为无约束优化问题.将种群大小设为 40,最大迭代次数设置为 500,每个算法独立运行 30 次取最优求解结果比较,各算法参数设置同对应参考文献,比较结果如表 5 所示.

表 5 8 种算法求解拉压弹簧设计问题最优结果

Table 5 Optimal results of 8 algorithms for solving tension/compression spring design problem

算法	$x_1$	$x_2$	$x_3$	最小质量
SSA	0.052 438	0.375 003	11.096 5	0.013 505
RCSSA	0.05	0.313 888	14.507 3	0.012 954
CASSA	0.05	0.311 513	14.841 7	0.013 116
ALSSA	0.061 133	0.619 380	5.156 4	0.016 566
AIWSSA	0.05	0.310 535	14.982 4	0.013 184
AGHSSA	0.067 769	0.840 170	3.136 8	0.019 821
MGSSA	0.05	0.316 409	14.163 4	0.012 786
RDSSA	0.066 244	0.809 880	3.575 5	0.019 815

拉压弹簧设计问题的目的是最小化拉伸或压缩弹簧的质量,如图 5 所示.此问题受最小挠度、剪切应力、浪涌频率、外径限制和设计变量的限制.设计变量是平均线圈直径  $D = x_1 (0.05 \leq x_1 \leq 2)$ 、导线直径  $d = x_2 (0.25 \leq x_2 \leq 1.3)$  和活动线圈数量  $N = x_3 (2 \leq x_3 \leq 15)$ .问题可以表述为

目标函数:

$$f(X) = (x_3 + 2)x_2x_1^2.$$

约束函数:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71\,785x_1^4} \leq 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12\,566(x_2x_1^3 - x_1^4)} + \frac{1}{5\,108x_1^2} - 1 \leq 0,$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

从表 5 结果可以看出,在 8 种算法中, MGSSA 求解结果最佳, RCSSA 次之, RDSSA、ALSSA 与 AGHSSA 在求解工程约束问题时效果相比于其他几个改进算法效果差.

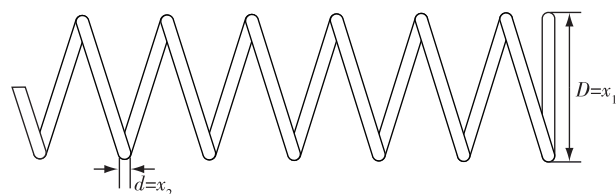


图 5 拉压弹簧模型

Fig. 5 Model of tension/compression spring

## 4 结束语

本研究引入的选择反向学习策略,相较于传统的反向学习策略,减少了计算维度,同时保留了种群的多样性,改善了每代种群解的质量.在跟随者位置更新阶段引入的精英均值策略,通过均值个体和最优个体的有效引导,使算法的寻优速度加快,改善了标准 SSA 的收敛速度慢的问题,同时收敛精度大幅提升.黄金正弦变异策略的加入,使算法在前期在均值个体周围寻找更优,后期随机选择个体在周围进行搜索,既有利于前期加快收敛速度,又有利于后期跳出局部最优.

通过在 14 个基准测试函数上进行对比实验,验证了 MGSSA 相较于 RDSSA、RCSSA、CASSA、ALSSA 和 AIWSSA 的稳定性和先进性.通过拉压弹簧设计问题可以观察到 MGSSA 在求解工程约束优化问题时表现良好.在接下来的研究中,将进一步改进樽海鞘群算法并将其应用到现实问题的求解.

## 参考文献

### References

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C]//International Conference on Neural Networks. November 27 - December 1, 1995, Perth, WA, Australia. IEEE, 2002:1942-1948
- [2] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67
- [3] Pierezan J, Coelho L D S. Coyote optimization algorithm: a new metaheuristic for global optimization problems [C]//2018 IEEE Congress on Evolutionary Computation (CEC). July 8 - 13, 2018, Rio de Janeiro, Brazil. IEEE, 2018: 1-8

- [ 4 ] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: algorithm and applications [ J ]. *Future Generation Computer Systems*, 2019, 97: 849-872
- [ 5 ] Khishe M, Mosavi M R. Chimp optimization algorithm [ J ]. *Expert Systems with Applications*, 2020, 149: 113338
- [ 6 ] 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
- [ 7 ] 陈涛, 王梦馨, 黄湘松. 基于樽海鞘群算法的无源时差定位 [ J ]. *电子与信息学报*, 2018, 40(7): 1591-1597  
CHEN Tao, WANG Mengxin, HUANG Xiangsong. Time difference of arrival passive location based on salp swarm algorithm [ J ]. *Journal of Electronics & Information Technology*, 2018, 40(7): 1591-1597
- [ 8 ] 王梦秋, 王艳, 纪志成. 基于改进樽海鞘群算法的PMSM多参数辨识 [ J ]. *系统仿真学报*, 2018, 30(11): 4284-4291, 4297  
WANG Mengqiu, WANG Yan, JI Zhicheng. Permanent magnet synchronous motor multi-parameter identification based on improved salp swarm algorithm [ J ]. *Journal of System Simulation*, 2018, 30(11): 4284-4291, 4297
- [ 9 ] Ibrahim A, Mohammed S, Ali H A, et al. Breast cancer segmentation from thermal images based on chaotic salp swarm algorithm [ J ]. *IEEE Access*, 2020, 8: 122121-122134
- [ 10 ] Tubishat M, Idris N, Shuib L, et al. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection [ J ]. *Expert Systems with Applications*, 2020, 145: 113122
- [ 11 ] Panda N, Majhi S K. Improved salp swarm algorithm with space transformation search for training neural network [ J ]. *Arabian Journal for Science and Engineering*, 2020, 45(4): 2743-2761
- [ 12 ] Manohar E, Banu E A, Punithavathani D S. Composite analysis of web pages in adaptive environment through modified salp swarm algorithm to rank the web pages [ J ]. *Journal of Ambient Intelligence and Humanized Computing*, 2022, 13(5): 2585-2600
- [ 13 ] 范千, 陈振健, 夏樟华. 一种基于折射反向学习机制与自适应控制因子的改进樽海鞘群算法 [ J ]. *哈尔滨工业大学学报*, 2020, 52(10): 183-191  
FAN Qian, CHEN Zhenjian, XIA Zhanghua. A modified salp swarm algorithm based on refracted opposition-based learning mechanism and adaptive control factor [ J ]. *Journal of Harbin Institute of Technology*, 2020, 52(10): 183-191
- [ 14 ] Kumar S, Kumari R, Nayyar A. Sigmoidal salp swarm algorithm [ C ] // 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD 2020). August 6-7, 2020, Durban, South Africa. IEEE, 2020: 1-7
- [ 15 ] 张达敏, 陈忠云, 辛梓芸, 等. 基于疯狂自适应的樽海鞘群算法 [ J ]. *控制与决策*, 2020, 35(9): 2112-2120  
ZHANG Damin, CHEN Zhongyun, XIN Ziyun, et al. Salp swarm algorithm based on craziness and adaptive [ J ]. *Control and Decision*, 2020, 35(9): 2112-2120
- [ 16 ] 陈雷, 蔺悦, 康志龙. 基于衰减因子和动态学习的改进樽海鞘群算法 [ J ]. *控制理论与应用*, 2020, 37(8): 1766-1780  
CHEN Lei, LIN Yue, KANG Zhilong. Improved salp swarm algorithm based on reduction factor and dynamic learning [ J ]. *Control Theory & Applications*, 2020, 37(8): 1766-1780
- [ 17 ] Qais M H, Hasanien H M, Alghuwainem S. Enhanced salp swarm algorithm: application to variable speed wind generators [ J ]. *Engineering Applications of Artificial Intelligence*, 2019, 80: 82-96
- [ 18 ] 刘景森, 袁蒙蒙, 左方. 面向全局搜索的自适应领导者樽海鞘群算法 [ J ]. *控制与决策*, 2021, 36(9): 2152-2160  
LIU Jingsen, YUAN Mengmeng, ZUO Fang. Global search-oriented adaptive leader salp swarm algorithm [ J ]. *Control and Decision*, 2021, 36(9): 2152-2160
- [ 19 ] 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 (CIM-CA-IAWTIC'06). November 28-30, 2005, Vienna, Austria. IEEE, 2005: 695-701
- [ 20 ] Nasser A B, Zamli K Z, Hujainah F, et al. An adaptive opposition-based learning selection: the case for jaya algorithm [ J ]. *IEEE Access*, 2021, 9: 55581-55594
- [ 21 ] Dhargupta S, Ghosh M, Mirjalili S, et al. Selective opposition based grey wolf optimization [ J ]. *Expert Systems with Applications*, 2020, 151: 113389
- [ 22 ] Alsaqr A M. Remarks on the use of Pearson's and Spearman's correlation coefficients in assessing relationships in ophthalmic data [ J ]. *African Vision and Eye Health*, 2021, 80(1): 10
- [ 23 ] 王秋萍, 王彦军, 戴芳. 多策略集成的樽海鞘群算法的机器人路径规划 [ J ]. *电子学报*, 2020, 48(11): 2101-2113  
WANG Qiuping, WANG Yanjun, DAI Fang. Multi-strategy ensemble salp swarm algorithm for robot path planning [ J ]. *Acta Electronica Sinica*, 2020, 48(11): 2101-2113
- [ 24 ] Tanyildizi E, Demir G. Golden sine algorithm: a novel math-inspired algorithm [ J ]. *Advances in Electrical and Computer Engineering*, 2017, 17(2): 71-78
- [ 25 ] Askari Q, Saeed M, Younas I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization [ J ]. *Expert Systems with Applications*, 2020, 161: 113702
- [ 26 ] 白钰, 彭珍瑞. 基于自适应惯性权重的樽海鞘群算法 [ J ]. *控制与决策*, 2022, 37(1): 237-246  
BAI Yu, PENG Zhenrui. Salp swarm algorithm based on adaptive inertia weight [ J ]. *Control and Decision*, 2022, 37(1): 237-246
- [ 27 ] 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 Xin, ZOU Hai. Adaptive salp swarm algorithm with golden sine algorithm and hybrid mutation [ J ]. *Computer Engineering and Applications*, 2021, 57(12): 75-85

## Golden sine salp swarm algorithm with multi-strategy

DING Meifang<sup>1</sup> WU Keqing<sup>1</sup> XIAO Peng<sup>1</sup>

<sup>1</sup> School of Science, Jiangxi University of Science and Technology, Ganzhou 341000, China

**Abstract** To improve the poor convergence performance and escape from local optimum of Salp Swarm Algorithm (SSA), a Golden sine SSA with Multi-strategy (MGSSA) is proposed. First, the Selective Opposition-Based Learning (SOBL) strategy is used to improve the population quality by calculating selective opposite solutions for individuals in the population that completely deviate from the optimal individual search direction. Then the optimal individual and elite mean individual are added in the follower position update phase to speed up the convergence of the algorithm. Finally, the golden sine algorithm variation strategy is selected based on the probability to further improve the quality of the solution, and facilitate the algorithm to jump out of the local optimum later. In this study, experiments are conducted on 14 benchmark test functions to compare with other swarm intelligence optimization algorithms and novel improved SSA, and then the proposed approach is applied to test the solution of engineering optimization problems in tension/compression spring design. The results show that the proposed MGSSA has high convergence accuracy and stability, and performs well in solving engineering problems.

**Key words** salp swarm algorithm (SSA); selective opposition-based learning (SOBL); elite mean; golden sine algorithm