

# 控制与决策

Control and Decision

## 基于自适应惯性权重的樽海鞘群算法

白钰, 彭珍瑞

引用本文:

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.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.0454>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 双重驱动的果蝇优化算法及其在PID控制器中的应用

Double drive fruit fly optimization algorithm and its application in PID controller  
控制与决策. 2021, 36(9): 2225–2233 <https://doi.org/10.13195/j.kzyjc.2020.0046>

### 基于混沌“微变异”自适应遗传算法

Adaptive genetic algorithm based on chaos “micro variation”  
控制与决策. 2021, 36(8): 2042–2048 <https://doi.org/10.13195/j.kzyjc.2021.0319>

### 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>

### 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>

# 基于自适应惯性权重的樽海鞘群算法

白 钰, 彭珍瑞<sup>†</sup>

(兰州交通大学 机电工程学院, 兰州 730070)

**摘要:** 针对标准樽海鞘群算法收敛精度低、收敛速度慢的问题, 提出一种基于自适应惯性权重的樽海鞘群算法 (AIWSSA)。首先, 在追随者位置更新公式中引入惯性权重因子评价个体之间的影响程度; 然后, 结合种群成功率与非线性递减函数对惯性权重因子进行自适应调整, 使算法的全局和局部搜索能力得到更好地平衡; 最后, 为防止算法陷入局部最优, 引入差分变异思想对非最优个体进行变异。对 12 个基准测试函数进行求解, 实验结果表明: AIWSSA 具有较高的收敛精度、收敛速度和鲁棒性; Wilcoxon 统计检验结果表明: 与标准樽海鞘群算法、改进的樽海鞘群算法、其他群体智能算法相比, AIWSSA 表现出较好的性能。通过将其应用于两种带约束的工程设计问题, 验证了 AIWSSA 的有效性。

**关键词:** 樽海鞘群算法; 自适应惯性权重; 非线性控制参数; 差分变异; 约束优化

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.0454

开放科学(资源服务)标识码(OSID):



引用格式: 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.

## Salp swarm algorithm based on adaptive inertia weight

BAI Yu, PENG Zhen-rui<sup>†</sup>

(School of Mechatronic Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:** The standard salp swarm algorithm (SSA) has low convergence accuracy and slow convergence speed. In order to solve these problems, a salp swarm algorithm based on adaptive inertia weight (AIWSSA) is proposed. Firstly, the inertia weight factor is introduced into the follower position update formula to evaluate the degree of influence between the individuals. Secondly, the combination of population successful rate and nonlinear decreasing function is applied to adjust the inertia weight factor adaptively to balance the exploration and exploitation abilities of the proposed algorithm. Finally, the differential mutation for the non-optimal individuals is used to avoid the algorithm of being trapped into local optimum. Then the experiments on the 12 benchmark test functions are conducted. The results show that the proposed AIWSSA has higher convergence accuracy, convergence speed and robustness, and the Wilcoxon statistical test results demonstrate that it has better performance compared with the standard salp swarm algorithm, the improved salp swarm algorithms and other swarm intelligence algorithms. Two constrained engineering design problems are applied to verify the effectiveness of the algorithm.

**Keywords:** salp swarm algorithm; adaptive inertia weight; nonlinear control parameter; differential mutation; constrained optimization

## 0 引言

优化算法在许多领域发挥着非常重要的作用。遗传算法(genetic algorithm, GA)<sup>[1]</sup>、差分进化算法(differential evolution algorithm, DEA)<sup>[2]</sup>等仿生进化算法以及粒子群优化(particle swarm optimization, PSO)算法<sup>[3]</sup>、萤火虫算法(firefly algorithm, FA)<sup>[4]</sup>、人工蜂群(artificial bee colony, ABC)算法<sup>[5]</sup>、蝙蝠算法(bat algorithm, BA)<sup>[6]</sup>、蝴蝶算法(butterfly optimization

algorithm, BOA)<sup>[7]</sup>等群体智能算法, 自提出以来就受到了广泛的关注。但是, 随着优化问题对象越来越复杂, 数据越来越庞大, 传统优化算法的收敛精度与收敛速度日渐难以满足要求。因此, 设计新的算法成为一个很好的应对途径。

樽海鞘群算法(salp swarm algorithm, SSA)<sup>[8]</sup>是 2017 年由 Mirjalili 等提出的一种基于随机种群的新型群体智能算法, 具有控制参数少、计算量小、易于实

收稿日期: 2020-04-21; 修回日期: 2020-11-03.

基金项目: 国家自然科学基金项目(51768035); 甘肃省高校协同创新团队项目(2018C-12).

责任编委: 刘德荣.

<sup>†</sup>通讯作者. E-mail: pzzui@163.com.

现等优点,在很多优化问题上体现出明显的优越性,自开发以来已成功应用于许多领域.但是,与其他群体智能算法一样,标准SSA仍然存在着收敛速度慢、求解精度低等缺陷.

研究人员采用了诸多方法对标准SSA进行了改进,如:文献[9]采用混沌映射调整领导者位置更新时的随机参数 $c_2$ ,提高了标准SSA在特征选择问题上的分类精度;文献[10]采用折射反向学习机制和自适应控制因子提高标准SSA的全局和局部搜索性能;文献[11]采用变异算子(DE/rand/1)对非最优个体进行变异,在配电系统无功功率补偿应用中取得了较好的效果;文献[12]结合标准SSA与PSO算法更新种群位置,在特征选择问题中能够删除冗余或混淆的特征,同时保持较高的精度和效率;文献[13]添加常数惯性权重因子到食物源位置,并结合 $K$ 邻域分类器进行特征选择;文献[14]采用时变惯性权重因子更新追随者位置,并结合自适应突变策略提高了SSA算法的性能.

为解决标准SSA求解精度不高和收敛速度慢的问题,本文提出一种基于自适应惯性权重的樽海鞘群算法(salp swarm algorithm based on adaptive inertia weight, AIWSSA).首先,在追随者位置更新时引入基于非线性递减函数的惯性权重因子,更好地平衡全局和局部搜索能力;然后,引入成功率作为种群状态反馈参数,自适应调整惯性权重因子;最后,对非最优解对应的个体进行差分变异,提高种群多样性,防止种群陷入局部最优.通过求解12个基准测试函数的最优解,并与标准SSA和其他群体智能算法作性能对比,验证了AIWSSA的有效性和鲁棒性.

## 1 标准樽海鞘群算法

在樽海鞘链中,根据个体在链状结构中的位置可将种群分为两部分:领导者和追随者.领导者位于链的前端,引领整个种群在空间内搜索食物源;其余个体为追随者,追随领导者进行局部搜索.种群通过个体之间的相互协作寻找全局最优位置(即食物源).以最小化问题为例,SSA的具体过程如下:

1) 初始化.随机初始化 $i$ 只个体的位置 $x_j^i, i \in [1, 2, \dots, N], j \in [1, 2, \dots, D]$ .其中: $N$ 为种群大小, $D$ 为空间维数.计算所有个体适应度值,选择适应度值最小的个体位置作为初始全局最优位置 $F_j$ .

2) 种群位置更新.领导者的位置更新如下:

$$x_j^i(t) = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j), & c_3 \geq 0.5; \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j), & c_3 < 0.5. \end{cases} \quad (1)$$

其中: $i = 1; t$ 为当前迭代次数; $F_j$ 为食物源在第 $j$ 维

空间中的位置; $ub_j$ 和 $lb_j$ 分别为第 $j$ 维空间的上限和下限; $c_2$ 和 $c_3$ 为 $[0, 1]$ 内的随机数,前者决定领导者在第 $j$ 维空间搜索时移动的步长,后者根据是否大于0.5确定移动的正负方向,第 $j$ 维步长由 $c_2$ 随机指定的位置和参数 $c_1$ 共同决定.参数 $c_1$ 定义如下:

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

其中 $T_{\max}$ 为最大迭代次数.

追随者呈链状顺次跟随移动,其位置变化只和它的初始位置、加速度、速度相关,因此移动的距离 $R$ 可表示为

$$R = 0.5a\Delta t^2 + v_0\Delta t. \quad (3)$$

其中:时间 $\Delta t$ 为迭代差值,即 $\Delta t = 1$ ;每次迭代开始个体初速度均为0,即 $v_0 = 0$ ;一次迭代开始到该次迭代结束时加速度 $a = (v_{\text{final}} - v_0)/\Delta t$ ;追随者移动到前一个樽海鞘位置时的末速度 $v_{\text{final}} = (x_j^{i-1}(t) - x_j^i(t))/\Delta t$ .将已知量代入式(3)可得

$$R = 0.5(x_j^{i-1}(t) - x_j^i(t)). \quad (4)$$

因此,追随者的位置更新公式为

$$x_j^i(t+1) = x_j^i(t) + R = 0.5(x_j^{i-1}(t) + x_j^i(t)), \quad (5)$$

其中 $i \geq 2$ 为除领导者之外剩余追随者的数量.

3) 全局最优位置更新.最优适应度值小于上次迭代的最优适应度值,则替换该最优适应度值对应的个体位置为全局最优位置.

4) 若满足停止条件(如达到精度或迭代次数要求),则停止迭代,输出优化结果;否则跳到步骤2),直到满足迭代终止条件.

## 2 自适应惯性权重樽海鞘群算法

为平衡SSA的全局和局部搜索能力,首先在追随者位置更新时引入基于非线性递减函数的惯性权重因子;然后引入种群成功率作为反馈参数对惯性权重因子自适应调整.此外,为防止陷入局部最优,对非最优个体进行差分变异,增加种群的多样性.

### 2.1 惯性权重策略

文献[3]首次将惯性权重因子 $\omega$ 引入PSO算法并取得了较为理想的效果.随后的众多研究也验证了 $\omega$ 对平衡算法全局和局部搜索能力的重要性.

从式(5)可以看出,某次迭代时,当前位置根据第 $i-1$ 只与第 $i$ 只追随者历史位置的中点进行更新,这也是樽海鞘链追随特性的数学模型体现.但是,这是一种盲目跟从的行为,没有考虑到前者对后者的影响,只是单向地接受前者的位置信息来更新当前位置,这在一定程度上限制了算法的搜索效率<sup>[15]</sup>.本文引入基于非线性递减函数的惯性权重因子评价樽海鞘链中第 $i-1$ 只追随者对第 $i$ 只追随者的影响程度,

改进的追随者位置更新公式如下:

$$x_j^i(t+1) = 0.5(\omega(t)x_j^{i-1}(t) + x_j^i(t)), \quad (6)$$

$$\omega(t) = \omega_i + (\omega_f - \omega_i) \exp(-(\alpha t/T_{\max})^2). \quad (7)$$

其中:  $\omega_i$  为初始惯性权重,  $\omega_f$  为迭代结束时的最终惯性权重,  $\alpha$  为非线性控制参数.

## 2.2 种群成功率策略

由式(7)可以看出,  $\omega(t)$  按照关于迭代次数  $t$  的非线性函数递减, 因为不监测个体位置、适应度值等情况的变化, 忽略了种群所处的实际环境, 不能完全体现实际的优化搜索过程, 因此, 其本质上并不是自适应的. 本文在此基础上引入种群成功率<sup>[16]</sup>, 并以此为反馈参数对式(7)进一步改进, 使得  $\omega(t)$  根据种群状态进行自适应调整.

以最小化问题为例, 樽海鞘链的个体  $i$  在第  $t$  次迭代中的成功值  $S(i, t)$  定义为

$$S(i, t) = \begin{cases} 1, & \text{fit}(\text{pbest}^i(t)) < \text{fit}(\text{pbest}^i(t-1)), \\ 0, & \text{fit}(\text{pbest}^i(t)) = \text{fit}(\text{pbest}^i(t-1)). \end{cases} \quad (8)$$

其中:  $\text{pbest}^i(t)$  为迭代至第  $t$  次时, 个体  $i$  的历史最优位置;  $\text{fit}(\cdot)$  为适应度函数. 根据个体  $i$  的成功值  $S(i, t)$ , 整个种群的成功率  $P_s$  定义为

$$P_s(t) = \frac{1}{N} \sum_{i=1}^N S(i, t). \quad (9)$$

其中:  $N$  为种群大小;  $P_s(t) \in [0, 1]$  为当前迭代  $t$  结束时, 历史最优位置对应的适应度值降低的个体数目占种群大小  $N$  的比例.

迭代初期, 较多个体分散于搜索空间内, 适应度值降低的个体数目相对较多, 反馈在成功率上则表现为  $P_s(t)$  较高, 结合  $P_s(t)$  的  $\omega(t)$  具有较强的全局搜索能力; 迭代后期, 种群个体逐渐聚集,  $P_s(t)$  逐渐降低, 结合  $P_s(t)$  的  $\omega(t)$  具有较强的局部搜索能力. 因此, 以  $P_s(t)$  为反馈参数监测种群个体分布状态,  $\omega(t)$  的自适应调整公式为

$$\omega(t) = \omega_i + (\omega_f - \omega_i) P_s(t) \exp(-(\alpha t/T_{\max})^2). \quad (10)$$

## 2.3 差分变异策略

如标准 SSA 所示, 种群全局最优位置始终以适应度值最优的个体位置进行更新, 因此忽略了其他非最优个体的位置变化, 使得种群多样性随迭代次数的增加而降低, 可能会造成个体陷入局部最优. 对非最优的个体进行变异, 可以增加其在下次迭代中成为最优个体的可能性, 提高算法跳出局部最优的能力. 本文引入差分进化中个体变异的思想, 根据变异算子

DE/best/1 对非最优个体进行变异

$$x_j^i(t) = F_j + \beta(x_j^{r_1} - x_j^{r_2}), \quad i \neq r_1 \neq r_2. \quad (11)$$

其中:  $x_j^i$  为变异后非最优个体的位置;  $\beta \in [0, 2]$  为变异系数, 此处取值为 0.5;  $x_j^{r_1}$  和  $x_j^{r_2}$  为任意选择的两只个体对应的位置. 若变异后个体的适应度值大于变异前, 则保持变异前的位置不变.

## 2.4 算法步骤

为防止算法早熟收敛, 降低陷入局部最优的可能性, 文献[17]采用了多领导者机制, 使算法在迭代前期具有更好的全局搜索性和随机性. 但算法的随机性随领导者数量的增加而增强, 导致稳定性降低, 为了在随机性与稳定性之间取得权衡, 本文选取一半的樽海鞘个体作为领导者. AIWSSA 步骤如下所示:

### 算法1 AIWSSA 算法.

初始化种群大小  $N$ , 最大迭代次数  $T_{\max}$ , 惯性权重  $\omega_i$  和  $\omega_f$ , 个体位置  $x_j^i$ .

计算每只樽海鞘个体适应度值  $\text{fit}(x^i)$ .

选出最优个体作为食物源位置  $F_j$ .

while ( $t < T_{\max}$ ) do

    根据式(2)更新  $c_1$ ;

$P_s = 0$ ;

    for  $i = 1, 2, \dots, N$ , do

        if  $i \leq N/2$ , then

            根据式(1)更新领导者位置;

        else

            根据式(6)更新追随者位置;

        end

        根据式(8)更新成功值  $S$ ;

    end

    计算每只樽海鞘个体适应度值  $\text{fit}(x^i)$ .

    更新食物源位置  $F_j$ .

    根据式(9)更新成功率  $P_s$ .

    根据式(10)更新惯性权重值  $\omega(t)$ .

    根据式(11)变异非最优个体位置  $x_j^i$ .

$t = t + 1$ .

end

return  $F_j$

## 2.5 时间复杂度分析

假设优化问题的目标函数计算量为 Cof, 解空间维度为  $D$ , 樽海鞘种群大小为  $N$ , 迭代次数为  $T_{\max}$ . 根据标准 SSA 的描述及时间复杂度的运算规则, 标准 SSA 的时间复杂度为  $O(T_{\max}(N \cdot D + N \cdot \text{Cof}))$ <sup>[8]</sup>.

由算法1的步骤可看出, 成功值  $S$ 、成功率  $P_s$ 、惯性权重因子  $\omega$  的更新以及非最优个体位置  $x_j^i$  的变异是 AIWSSA 相对标准 SSA 增加的步骤. 根据时间复杂度运算规则, 在一次迭代中, 所增加步骤对应的时间

复杂度依次为 $O(N)$ 、 $O(1)$ 、 $O(1)$ 和 $O(N)$ . 因此,  $T_{\max}$ 次迭代中, AIWSSA增加的时间复杂度 $O(T_{\max}(2N + 2))$ , 略去低阶项后, AIWSSA的总时间复杂度仍为 $O(T_{\max}(N \cdot D + N \cdot \text{Cof}))$ , 与标准SSA的时间复杂度属于同一数量级, 说明AIWSSA没有增加过多的计算开销.

### 3 数值实验与结果分析

为验证本文提出的AIWSSA的有效性和鲁棒性, 进行4组实验测试: 1) 非线性控制参数 $\alpha$ 和惯性权重因子 $\omega$ 的敏感性分析; 2) 改进策略的有效性分析; 3) AIWSSA与其他策略改进的SSA性能对比; 4) AIWSSA与其他群体智能算法性能对比.

#### 3.1 基准测试函数及评价准则

为了比较的公平性, 4组实验测试均在12个典型基准测试函数上进行, 测试函数参照文献[8]. 所有测试函数都为最小化问题, 理论最优值 $f_{\min} = 0$ , 包含的特征分为单峰(unimodal)、多峰(multimodal)、可分(separable)、不可分(non-separable), 其中函数 $f_7$ 包含噪声. 函数的名称、维度、搜索区间、特征、精度 $e$ 如表1所示.

表1 基准测试函数

函数	函数名	维度	特征	搜索区间	精度 $e$
$f_1$	Sphere	30	US	[-100, 100]	1e-08
$f_2$	Schwefel 2.22	30	UN	[-10, 10]	1e-08
$f_3$	Schwefel 1.2	30	UN	[-100, 100]	1e-08
$f_4$	Schwefel 2.21	30	UN	[-100, 100]	1e-08
$f_5$	Rosenbrock	30	UN	[-30, 30]	1e-00
$f_6$	Step	30	US	[-100, 100]	1e-08
$f_7$	Quartic	30	US	[-1.28, 1.28]	1e-04
$f_8$	Schaffer N.2	2	MN	[-100, 100]	1e-08
$f_9$	Rastrigin	30	MS	[-5.12, 5.12]	1e-08
$f_{10}$	Ackley	30	MN	[-32, 32]	1e-08
$f_{11}$	Griewank	30	MN	[-600, 600]	1e-08
$f_{12}$	Penalized	30	MN	[-50, 50]	1e-02

算法性能根据以下准则进行评价: 1) 平均值(mean value, MV)反映算法的收敛精度和寻优能力; 2) 标准差(standard deviation, SD)反映算法的稳定性; 3) 成功率(successful rate, SR)是算法多次独立运行后, 收敛到问题最优解的实验次数占总运行次数的

百分比. 求解是否成功按下式进行判断:

$$|f - f_{\min}| < e. \tag{12}$$

若算法 $t$ 次迭代后求解的最优适应度值 $f$ 与理论最优值 $f_{\min}$ 的绝对差值小于设定的收敛精度 $e$ , 则认为该次实验成功收敛到问题的最优解. 其中, 预先设定的收敛精度值 $e$ 参照文献[18-20]的设定.

为避免单次运行的随机性造成结果的偏差, 实验测试均独立运行30次. 除了采集基准测试函数的均值、标准差和成功率之外, 采用非参数统计检验方法Wilcoxon秩和检验在5%显著性水平下对30次结果统计分析. 零假设是假定两个算法没有显著性差异, 当检验所计算的 $p$ 值(接受零假设的概率)小于0.05时, 认为两个算法具有统计学意义上的显著性差异, 否则拒绝零假设, 认为算法性能相近.

本文所有算法的种群大小 $N$ 均为30, 最大迭代次数 $T_{\max}$ 均为1000. 实验测试环境为: Intel(R) Core(TM)2-P7450处理器, 4G内存, 基于Matlab 2018b环境M语言编写算法.

#### 3.2 参数敏感性分析

惯性权重因子 $\omega$ 是影响全局和局部搜索能力的关键. 由式(10)可以看出, 惯性权重因子 $\omega$ 的变化又依赖于两组控制参数: 1) 初始值 $\omega_i$ 和最终值 $\omega_f$ ; 2) 非线性控制参数 $\alpha$ . 因此, 本节通过数值实验测试分析这两组参数对AIWSSA性能的影响.

选取 $\omega_i$ 和 $\omega_f$ 分别从0到1, 步长为0.1进行变化, 参数 $\alpha$ 保持不变, 为防止数据量级差异造成结果偏差, 首先将30次实验中, 12个基准测试函数的最优适应度值分别归一化到[0, 1], 然后对30组归一化后的最优适应度值求平均, 得到12个基准测试函数的收敛精度, 最后对12个收敛精度再次求平均得到AIWSSA求解的综合收敛精度. 表2列出了不同组合的 $\omega_i$ 和 $\omega_f$ 值对综合收敛精度的影响结果(限于文章篇幅, 在不影响判断变化趋势的情况下, 仅显示 $\omega_i \in [0, 0.5]$ 之间的结果). 选取 $\alpha$ 从1到10, 步长为1进行变化, 参数 $\omega$ 保持不变, 观察12个基准测试函数收敛精度的变化. 表3列出了 $\alpha$ 取值为1~10对AIWSSA收敛精度的影响结果.

表2 不同的 $\omega_i$ 和 $\omega_f$ 值对AIWSSA收敛精度的影响结果

$\omega_i$	$\omega_f$										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	0.0634	0.0634	0.0728	0.0674	0.0701	0.0582	0.0647	0.0607	0.0728	0.0662	<b>0.0515</b>
0.1	0.0542	0.0674	0.0635	0.0609	0.0754	0.0649	0.0608	0.0556	0.0515	<b>0.0501</b>	0.0583
0.2	0.0688	0.0674	0.0634	0.0701	0.0648	0.0622	0.0565	<b>0.0514</b>	0.0676	0.0769	0.0678
0.3	0.0661	0.0742	0.0687	0.0687	0.0702	0.0634	0.0673	0.0637	0.0668	<b>0.0620</b>	0.0673
0.4	0.0729	0.0689	0.0689	0.0621	0.0688	0.0688	<b>0.0647</b>	0.0836	0.0712	0.0685	0.0889
0.5	0.0663	0.0676	0.0664	0.0744	0.0719	0.0764	0.0800	0.1042	0.0884	0.1031	0.0961

表3 不同的 $\alpha$ 值对AIWSSA收敛精度的影响结果

$\alpha$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$
1	0	2.45e-264	0	8.01e-255	28.062 2	<b>1.22e-08</b>	3.63e-05	0	0	<b>8.88e-16</b>	0	1.38e-02
2	0	4.62e-298	0	3.67e-295	28.014 0	1.29e-08	4.23e-05	0	0	<b>8.88e-16</b>	0	6.91e-03
3	0	0	0	2.60e-307	28.021 9	1.27e-08	4.81e-05	0	0	<b>8.88e-16</b>	0	8.66e-11
4	0	3.99e-314	0	2.39e-312	28.015 4	1.25e-08	<b>3.35e-05</b>	0	0	<b>8.88e-16</b>	2.47e-04	3.46e-03
5	0	4.10e-315	0	3.38e-316	28.016 0	1.28e-08	3.99e-05	0	0	<b>8.88e-16</b>	0	6.91e-03
6	0	2.42e-319	0	1.03e-316	28.028 8	1.29e-08	5.42e-05	0	0	<b>8.88e-16</b>	0	3.46e-03
7	0	1.32e-319	0	3.05e-319	28.051 9	1.24e-08	5.35e-05	0	0	<b>8.88e-16</b>	3.29e-04	1.04e-02
8	0	7.55e-321	0	2.72e-320	28.029 6	1.37e-08	5.25e-05	0	0	<b>8.88e-16</b>	3.29e-04	6.91e-03
9	0	2.30e-322	0	<b>1.61e-321</b>	28.050 3	1.24e-08	4.97e-05	0	0	<b>8.88e-16</b>	0	3.46e-03
10	0	2.40e-322	0	2.11e-321	<b>28.008 4</b>	1.29e-08	5.61e-05	0	0	<b>8.88e-16</b>	0	<b>8.22e-11</b>

由表2可知,  $\omega_f$  不变时, 综合收敛精度随着  $\omega_i$  的增加而降低,  $\omega_i \in [0, 0.4]$  时精度较高, 而在  $\omega_i \in [0, 0.4]$  时, 不同的  $\omega_f$  取值对综合收敛精度影响不大, 总体而言,  $\omega_f \in [0.6, 1.0]$  时, AIWSSA 求解的综合收敛精度较高. 因此,  $\omega_i$  和  $\omega_f$  合理取值范围分别推荐为  $\omega_i \in [0, 0.4]$ ,  $\omega_f \in [0.6, 1.0]$ .

由表3可知, 除函数  $f_{11}$ ,  $f_{12}$  外, 在其余10个函数上收敛精度相差不大. 这充分说明在求解同一个函数时, AIWSSA 对控制参数  $\alpha$  取值不太敏感, 总体而言,  $\alpha = 9 \sim 10$  时算法总体寻优性能较好.

### 3.3 改进策略有效性分析

分析3种改进策略对算法性能的影响. 将AIWSSA与仅采用惯性权重策略的算法(ISSA)、仅采用种群成功率策略的算法(PSSA)、仅采用差分变异策略的算法(DSSA)进行比较. 其中: ISSA的参数根据参数敏感性分析后的推荐值进行设置, 取  $\omega_i = 0, \omega_f = 1, \alpha = 9$ ; PSSA和DSSA的参数分别根据式(9)和(11)进行设置. 4种算法独立运行30次的统计结果如表4所示.

表4 不同改进策略实验测试结果

函数	评价标准	ISSA	PSSA	DSSA	AIWSSA	函数	评价标准	ISSA	PSSA	DSSA	AIWSSA
$f_1$	MV	0	1.58e-26	8.33e-09	0	$f_7$	MV	5.43e-05	5.44e-05	7.50e-02	5.60e-05
	SD	0	1.62e-26	2.31e-09	0		SD	5.02e-05	5.61e-05	3.44e-02	4.83e-05
	SR/%	100	100	76.67	100		SR/%	80	90	0	90
	$p$	NA	6.06e-13	6.06e-13	NA		$p$	NA	1.42e-01	1.51e-11	2.60e-01
$f_2$	MV	1.00e-251	7.26e-15	2.92e+00	0	$f_8$	MV	0	0	3.26e-16	0
	SD	0	3.84e-15	2.55e+00	0		SD	0	0	6.05e-16	0
	SR/%	100	100	0	100		SR/%	100	100	100	100
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	NA	NA	7.21e-05	NA
$f_3$	MV	0	8.65e-35	2.69e+01	0	$f_9$	MV	0	0	4.89e+01	0
	SD	0	9.50e-35	2.08e+01	0		SD	0	0	1.78e+01	0
	SR/%	100	100	0	100		SR/%	100	100	0	100
	$p$	NA	6.06e-13	6.06e-13	NA		$p$	NA	NA	6.06e-13	NA
$f_4$	MV	9.47e-252	6.69e-13	9.22e+00	0	$f_{10}$	MV	8.88e-16	9.41e-15	2.30e+00	8.88e-16
	SD	0	1.62e-12	3.21e+00	0		SD	0	4.81e-15	8.91e-01	0
	SR/%	100	100	0.00	100		SR/%	100	100	0	100
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	NA	3.31e-13	6.06e-13	NA
$f_5$	MV	2.80e+01	2.80e+01	6.98e+01	2.80e+01	$f_{11}$	MV	0	0	1.17e-02	0
	SD	9.08e-02	7.57e-02	7.25e+01	8.09e-02		SD	0	0	1.12e-02	0
	SR/%	0	0	0	0		SR/%	100	100	0	100
	$p$	NA	2.46e-01	1.06e-01	2.78e-03		$p$	NA	NA	6.06e-13	NA
$f_6$	MV	1.28e-08	1.21e-08	8.52e-09	1.22e-08	$f_{12}$	MV	7.89e-11	8.75e-11	4.44e+00	7.53e-11
	SD	2.58e-09	2.95e-09	2.01e-09	3.01e-09		SD	2.24e-11	3.25e-11	2.00e+00	2.46e-11
	SR/%	13.33	20.00	83.33	23.33		SR/%	100	100	0	100
	$p$	8.49e-09	3.37e-06	NA	2.93e-06		$p$	4.68e-01	4.38e-01	1.51e-11	NA

本文定义均值最小的算法为最优算法, 表示为NA; 均值相同时, 标准差较小的算法性能较优. 某算法的 $p$ 值大于0.05, 认为该算法与最优算法性能无显著性差异. 由表4可知: 在求解精度方面, DSSA仅在1个函数上具有最高精度; 从统计学意义上讲, PSSA、ISSA和AIWSSA分别在6个、9个和10个函数上具有最高精度. 在求解成功率方面, DSSA在绝大多数函数上均为0%, 而ISSA、PSSA和AIWSSA在8个函数上具有100%的求解成功率. 由此可见: 仅采用差分

变异策略对改进标准SSA性能的帮助有限; 惯性权重策略对标准SSA性能的改进起主导作用; 种群成功率策略作为当前种群状态的反馈, 与惯性权重策略结合后进一步提高了标准SSA的收敛精度.

本文通过收敛曲线刻画收敛速度方面的性能, 算法迭代中, 根据每次迭代的平均最优适应度值绘制曲线图, 横坐标代表迭代次数, 纵坐标代表每次迭代的平均最优适应度值的对数. 图1给出了部分基准测试函数的收敛曲线.

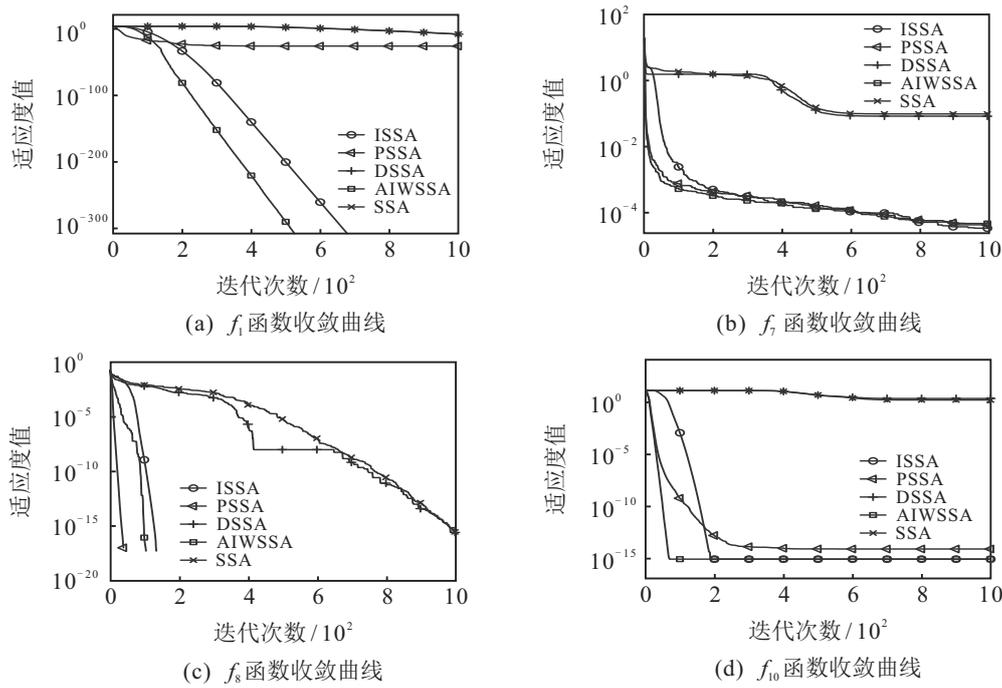


图1 基准测试函数  $f_1$ 、 $f_7$ 、 $f_8$ 、 $f_{10}$  收敛曲线

对 AIWSSA 的有效性进行分析: 迭代初期, 种群分散于整个搜索空间进行全局搜索, 较多个体的适应度值能够降低, 反映在图 1 中为一段时间内 PSSA 收敛曲线的快速下降 (如图 1(a)、1(b) 和 1(d) 中 PSSA 与 ISSA 收敛曲线交叉前 PSSA 的收敛曲线), 这说明种群成功率策略在此阶段起主导作用; 随着迭代逐渐向局部搜索过渡, 种群成功率趋于平稳, PSSA 可能表现出陷入局部最优的现象, 此时惯性权重策略结合种群成功率, 引导算法跳出局部最优, 反映在图 1 中

为 PSSA 与 ISSA 收敛曲线交叉后, ISSA 曲线收敛速度高于 PSSA; 两者结合后, AIWSSA 算法整体收敛速度得到了有效的提升。

### 3.4 改进樽海鞘群算法性能对比

将 AIWSSA 与标准樽海鞘群算法 (SSA)<sup>[8]</sup>、疯狂自适应樽海鞘群算法 (CASSA)<sup>[17]</sup>、突变樽海鞘群算法 (MSSA)<sup>[11]</sup> 的性能进行对比, 其他对比算法控制参数的具体设置请参见相应的参考文献。4 种算法独立运行 30 次的统计结果如表 5 所示。

表5 SSA改进算法实验测试结果

函数	评价标准	SSA	CASSA	MSSA	AIWSSA	函数	评价标准	SSA	CASSA	MSSA	AIWSSA
$f_1$	MV	1.10e-08	6.97e-74	0	0	$f_7$	MV	7.56e-02	6.80e-05	1.63e-02	3.93e-05
	SD	3.09e-09	8.75e-75	0	0		SD	2.86e-02	6.46e-05	1.93e-02	2.99e-05
	SR/%	40	100	100	100		SR/%	0	73.33	3.33	96.67
	$p$	6.06e-13	6.06e-13	NA	NA		$p$	1.51e-11	9.04e-02	1.91e-10	NA
$f_2$	MV	8.48e-01	1.34e-36	8.65e-02	0	$f_8$	MV	4.88e-16	0	0	0
	SD	7.87e-01	7.15e-38	3.04e-01	0		SD	5.96e-16	0	0	0
	SR/%	0	100	90	100		SR/%	100	100	100	100
	$p$	6.06e-13	6.06e-13	4.08e-02	NA		$p$	2.53e-09	NA	NA	NA
$f_3$	MV	2.30e+01	2.04e-71	0	0	$f_9$	MV	4.56e+01	0	3.32e-02	0
	SD	1.32e+01	3.67e-72	0	0		SD	1.51e+01	0	1.82e-01	0
	SR/%	0	100	100	100		SR/%	0	100	93.33	100
	$p$	6.06e-13	6.06e-13	NA	NA		$p$	6.06e-13	NA	2.10e-02	NA
$f_4$	MV	6.56e+00	8.35e-38	3.15e-03	0	$f_{10}$	MV	2.12e+00	8.88e-16	8.88e-16	8.88e-16
	SD	3.14e+00	9.74e-39	1.73e-02	0		SD	7.65e-01	0	0	0
	SR/%	0	100	96.67	100		SR/%	0	100	100	100
	$p$	6.06e-13	6.06e-13	1.67e-01	NA		$p$	6.06e-13	NA	NA	NA
$f_5$	MV	5.20e+01	2.80e+01	2.78e+01	2.80e+01	$f_{11}$	MV	1.16e-02	0	1.14e-09	0
	SD	7.57e+01	7.52e-02	9.34e-01	7.03e-02		SD	1.13e-02	0	6.27e-09	0
	SR/%	0	0	0	0		SR/%	0	100	96.67	100
	$p$	2.95e-01	3.70e-01	NA	3.46e-04		$p$	6.31e-12	NA	1.67e-01	NA
$f_6$	MV	1.10e-08	1.39e-06	1.30e-08	1.26e-08	$f_{12}$	MV	4.65e+00	4.59e-08	7.95e-03	7.98e-11
	SD	2.99e-09	5.70e-07	2.46e-09	2.73e-09		SD	3.21e+00	2.05e-08	2.66e-02	1.76e-11
	SR/%	43.33	0	10.00	16.67		SR/%	0	100	90	100
	$p$	NA	1.51e-11	1.68e-04	2.34e-02		$p$	1.51e-11	1.51e-11	2.10e-01	NA

由表5可知, AIWSSA在10个函数上具有最小平均值且其标准差很小, 反映了其收敛精度较高的同时还具有较强的鲁棒性. 在寻优成功率方面, 除了函数  $f_5 \sim f_7$  之外, AIWSSA在其他函数上的成功率均为100%, 而标准SSA仅在1个函数上达到100%, 其寻优能力有待提升. 与CASSA相比, AIWSSA在6个函数上具有显著性的优势, 在5个函数上与其收敛性能相当, 但在这5个函数上, AIWSSA成功率均不低于CASSA; 与MSSA相比, AIWSSA在3个函数上具

有显著性的优势, 在7个函数上与其收敛性能相当, 但AIWSSA在这7个函数上的成功率均为100%, 而MSSA在这7个函数中有3个函数仅能达到96.67%、96.67%、90%. 综上所述, 与标准SSA、CASSA、MSSA相比, AIWSSA性能具有显著性的优势.

为更加直观地研究4种樽海鞘群算法收敛性能, 图2给出了部分基准测试函数的收敛曲线. 由图2可知, 与标准SSA及其他樽海鞘群算法相比, AIWSSA具有较快的收敛速度和较高的收敛精度.

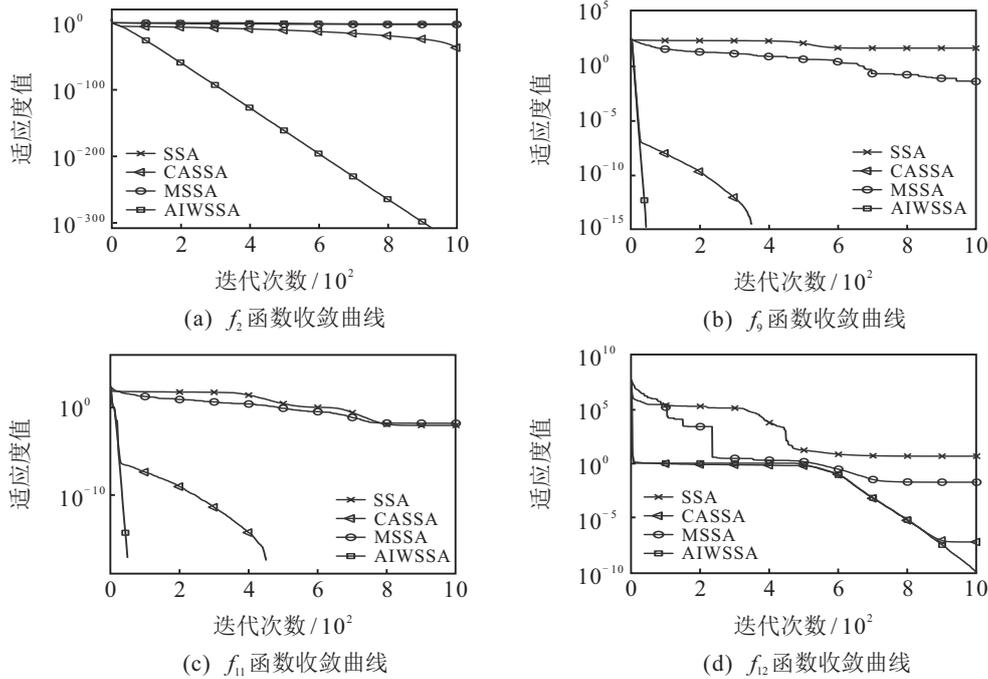


图2 基准测试函数  $f_2, f_9, f_{11}, f_{12}$  收敛曲线

### 3.5 群体智能算法性能对比

为进一步验证AIWSSA的有效性, 选择3种近年来较为流行且性能较为优越的群体智能算法进行性能对比. 对比算法有: 灰狼算法(grey wolf optimizer, GWO)<sup>[21]</sup>, 飞蛾扑火(moth-flame optimization, MFO)算法<sup>[22]</sup>、鲸鱼优化算法(whale optimization algorithm,

WOA)<sup>[23]</sup>. 由于各对比算法已经分别与经典的GA<sup>[1]</sup>、DE<sup>[2]</sup>、PSO<sup>[3]</sup>、FA<sup>[4]</sup>、BA<sup>[6]</sup>、BOA<sup>[7]</sup>算法进行了对比且性能上优于这几种算法, 本文不再将AIWSSA与这几种经典算法做对比. 3种群体智能算法控制参数的具体设置请参见相应的参考文献. 表6列出了4种算法独立运行30次的统计结果.

表6 其他群体智能算法实验测试结果

函数	评价标准	GWO	MFO	WOA	AIWSSA	函数	评价标准	GWO	MFO	WOA	AIWSSA
$f_1$	MV	2.35e-59	3.33e+03	8.72e-126	0	$f_7$	MV	8.69e-04	2.99e+00	1.19e-03	5.23e-05
	SD	3.20e-59	6.61e+03	4.44e-125	0		SD	3.35e-04	4.92e+00	1.53e-03	5.85e-05
	SR/%	100	0	100	100		SR/%	0	0	20	80
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	1.51e-11	1.51e-11	3.80e-07	NA
$f_2$	MV	9.15e-35	3.50e+01	3.32e-83	0	$f_8$	MV	0	2.37e-04	0	0
	SD	6.91e-35	2.18e+01	1.03e-82	0		SD	0	1.14e-03	0	0
	SR/%	100	0	100	100		SR/%	100	93.33	100	100
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	NA	8.04e-02	NA	NA
$f_3$	MV	6.40e-15	1.92e+04	2.50e+04	0	$f_9$	MV	5.75e-01	1.71e+02	0	0
	SD	2.29e-14	1.44e+04	1.33e+04	0		SD	1.78e+00	3.06e+01	0	0
	SR/%	100	0	0	100		SR/%	90	0	100	100
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	6.80e-04	6.06e-13	NA	NA
$f_4$	MV	1.53e-14	6.77e+01	1.98e+01	0	$f_{10}$	MV	1.60e-14	1.56e+01	4.56e-15	8.88e-16
	SD	3.90e-14	7.44e+00	2.33e+01	0		SD	2.79e-15	7.00e+00	2.87e-15	0
	SR/%	100	0	0	100		SR/%	100	0	100	100
	$p$	6.06e-13	6.06e-13	6.06e-13	NA		$p$	1.30e-13	6.06e-13	1.90e-08	NA

表6 (续)

函数	评价标准	GWO	MFO	WOA	AIWSSA	函数	评价标准	GWO	MFO	WOA	AIWSSA
$f_5$	MV	2.68e+01	8.01e+06	2.77e+01	2.80e+01	$f_{11}$	MV	2.73e-03	2.41e+01	1.14e-02	0
	SD	6.69e-01	2.44e+07	3.48e-01	6.09e-02		SD	6.33e-03	5.77e+01	3.73e-02	0
	SR/%	0	0	0	0		SR/%	80	0	90	100
	$p$	NA	2.79e-10	1.30e-05	2.49e-11		$p$	5.52e-03	6.06e-13	4.08e-02	NA
$f_6$	MV	7.14e-01	2.01e+03	3.73e-02	1.23e-08	$f_{12}$	MV	4.32e-02	1.71e+07	3.76e-03	7.98e-11
	SD	3.96e-01	4.08e+03	1.38e-02	1.80e-09		SD	2.33e-02	6.49e+07	1.99e-03	1.76e-11
	SR/%	0	0	0	10		SR/%	3.33	16.67	100	100
	$p$	1.51e-11	1.51e-11	1.51e-11	NA		$p$	3.56e-09	7.05e-10	2.79e-10	NA

由表6可知, AIWSSA在11个函数上都具有最小平均值和标准差, 反映了AIWSSA相比GWO、MFO、WOA具有较高的收敛精度和较高的鲁棒性. 在寻优

成功率方面, AIWSSA的成功率均不低于其他3种算法. 综上所述, 相比其他3种群体智能优化算法, AIWSSA性能具有显著性优势.

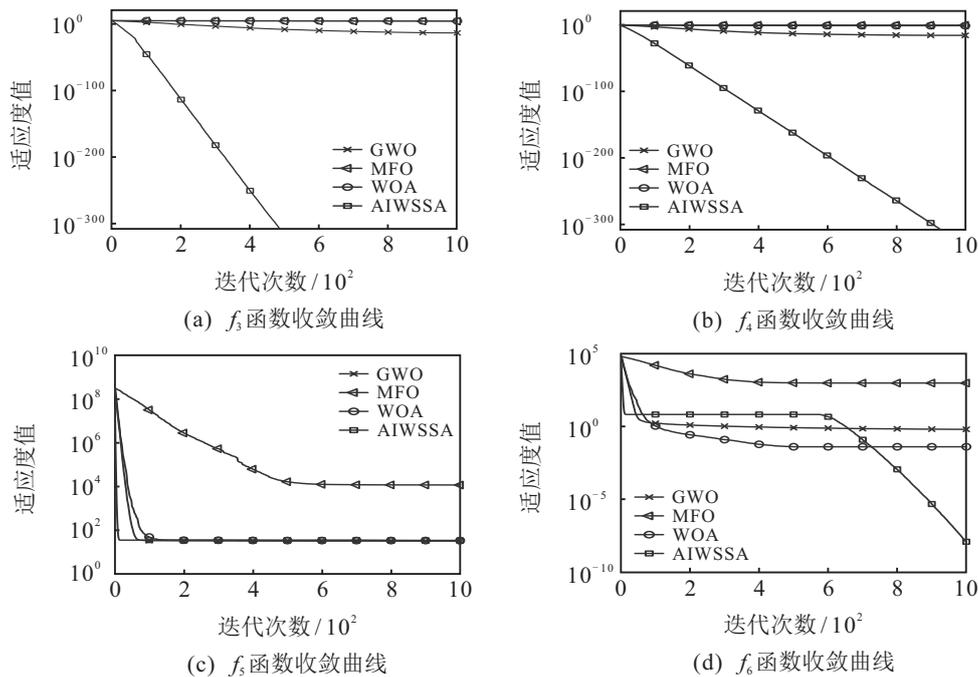


图3 基准测试函数  $f_3$ 、 $f_4$ 、 $f_5$ 、 $f_6$  收敛曲线

图3给出了部分基准测试函数的收敛曲线. 由图3可知, 与其他3种群体智能优化算法相比, AIWSSA具有较快的收敛速度和较高的收敛精度.

### 4 工程算例及结果分析

为进一步验证AIWSSA算法在工程应用中的有效性和可行性, 选用三杆桁架和焊接梁设计问题对工程算例进行分析.

本文采用文献[24]中的罚函数法进行约束处理, 对两个工程算例优化求解, 并分别将优化结果与其他算法优化的结果进行对比. 为公平起见, 对比算法的优化结果均是来自原文献中的结果.

#### 4.1 三杆桁架设计问题

该问题以如图4的桁架结构为研究对象, 优化目标是在满足压力的条件下使得桁架结构重量最小.

桁架共包含3个决策变量, 即杆1~杆3的截面积. 由于杆1与杆3对称, 只取杆1和杆2的截面积  $A_1$ 、 $A_2$ , 分别表示为  $x_1$ 、 $x_2$ ; 存在3个约束条件, 其具体的

数学模型表示如下:

$$\begin{aligned}
 \min f(X) &= (2\sqrt{2}x_1 + x_2)l. \\
 \text{s.t. } g_1(X) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0; \\
 g_2(X) &= \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0; \\
 g_3(X) &= \frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0; \\
 0 \leq x_i &\leq 1.0, \quad i = 1, 2.
 \end{aligned} \tag{13}$$

其中:  $l = 100 \text{ cm}$ ,  $P = 2 \text{ kN/cm}^2$ ,  $\sigma = 2 \text{ kN/cm}^2$ .

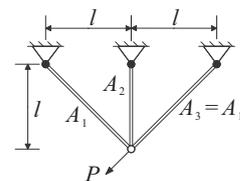


图4 三杆桁架模型

分别利用算法AIWSSA和标准SSA对上述优化

问题进行求解, 并与MFO<sup>[22]</sup>、ATMDE<sup>[25]</sup>(differential evolution with adaptive trade-off model)、GDAFA<sup>[26]</sup>(dynamically adaptive firefly algorithm with globally orientation)算法的最优求解结果进行比较, 对比结果如表7所示。

表7 5种不同算法求解三杆桁架设计问题的最优解对比

算法	$x_1$	$x_2$	$f(X)$
MFO	0.788 2	0.409 5	263.895 98
ATMDE	0.788 7	0.408 2	<b>263.895 84</b>
GDAFA	0.785 4	0.417 7	263.930 00
SSA	0.788 7	0.408 3	<b>263.895 84</b>
AIWSSA	0.788 7	0.408 3	<b>263.895 84</b>

由表7可知, AIWSSA对三杆桁架设计问题的优化效果优于或不差于其他4种算法。

### 4.2 焊接梁设计问题

该问题以如图5所示的焊接梁为研究对象, 优化目标是使总制造成本最低。焊接梁共包含4个决策变量: 焊缝宽度 $h$ 、长度 $l$ 、梁宽度 $d$ 、厚度 $b$ , 分别表示为 $x_1, x_2, x_3, x_4$ ; 包含7个约束条件: 剪切应力 $\tau$ 、横梁弯曲应力 $\sigma$ 、屈曲载荷 $P_c$ 、横梁挠度 $\delta$ 以及各设计变量之间的尺寸约束。其具体的数学模型表示如下:

$$\begin{aligned} \min f(X) &= 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2). \\ \text{s.t. } g_1(X) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} - \tau_{\max} \leq 0; \\ g_2(X) &= \frac{6PL}{x_3^2x_4} - \sigma_{\max} \leq 0; \\ g_3(X) &= x_1 - x_4 \leq 0; \\ g_4(X) &= \\ &0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0; \\ g_5(X) &= 0.125 - x_1 \leq 0; \\ g_6(X) &= \frac{4PL^3}{Ex_3^3x_4} - \delta_{\max} \leq 0; \\ g_7(X) &= P - \frac{4.013Ex_3x_4^3}{6L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \leq 0; \\ 0.1 \leq x_i \leq 2, \quad i &= 1, 4; \\ 0.1 \leq x_i \leq 10, \quad i &= 2, 3. \end{aligned} \tag{14}$$

其中

$$\begin{aligned} \tau' &= \frac{P}{\sqrt{2}x_1x_2}, \\ \tau'' &= \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right), \\ J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \\ R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad P = 6\,000 \text{ lb}, \\ L &= 14 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \\ \tau_{\max} &= 13\,600 \text{ psi}, \quad \sigma_{\max} = 30\,000 \text{ psi}, \end{aligned}$$

$$\delta_{\max} = 0.25 \text{ in.}$$

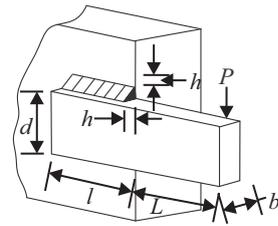


图5 焊接梁模型

分别利用算法AIWSSA和标准SSA对上述优化问题求解, 并与ATMDE<sup>[25]</sup>、 $V(\mu + 1)$ -ES<sup>[27]</sup>(variation of  $(\mu + 1)$  evolution strategies)、RO<sup>[28]</sup>(ray optimization)、BA<sup>[29]</sup>、TEO<sup>[30]</sup>(thermal exchange optimization)算法的最优结果进行比较, 对比结果如表8所示。

表8 7种不同算法求解焊接梁设计问题的最优解对比

算法	$x_1$	$x_2$	$x_3$	$x_4$	$f(X)$
ATMDE	0.244 4	6.217 5	8.291 5	0.244 4	2.381 0
$V(\mu + 1)$ -ES	0.199 7	3.612 1	9.037 5	0.206 1	1.737 3
RO	0.203 7	3.528 5	9.004 2	0.207 2	1.735 3
BA	0.201 5	3.562 0	9.041 4	0.205 7	1.731 2
TEO	0.205 7	3.472 0	9.035 1	0.205 8	1.725 3
SSA	0.205 7	3.471 4	9.036 6	0.205 7	1.724 9
AIWSSA	0.205 7	3.470 8	9.036 8	0.205 7	<b>1.724 8</b>

从表8可知, AIWSSA对焊接梁设计问题的优化效果优于其他6种算法。

综上所述, 本文算法AIWSSA在实际工程应用中是有效和可行的, 且能够表现出较好的寻优能力。

### 5 结论

本文提出了一种基于自适应惯性权重的樽海鞘群算法——AIWSSA。所提出的算法在追随者位置更新中引入惯性权重因子评价樽海鞘个体之间的影响程度; 结合种群成功率与非线性递减函数对惯性权重因子进行自适应调整, 使算法在迭代初期具有较高的全局搜索能力, 在迭代后期具有较高的局部搜索能力; 引入差分进化思想对非最优个体进行差分变异, 增加种群的多样性, 防止算法陷入局部最优。通过对12个基准测试函数的求解以及参数敏感性分析和改进策略有效性分析, 验证了AIWSSA在求解无约束优化问题时具有较高的收敛精度和求解鲁棒性; 与SSA、CASSA、MSSA、GWO、WOA、MFO等算法相比, AIWSSA具有较强的竞争力。通过对两个带约束工程设计问题的优化, 验证了AIWSSA在求解约束优化问题上的有效性。如何进一步提高算法收敛速度, 并将其应用于实际工程问题, 还需要进一步研究。

### 参考文献(References)

[1] Ren Q S, Ye Z X, Zeng J, et al. Analysis of genetic operators[J]. Acta Electronica Sinica, 2000, 28(5): 113-114.

- [2] Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [3] Shi Y H, Eberhart R, Chen Y. Evolutionary modular fuzzy system[C]. *IEEE World Congress on Computational Intelligence*. Anchorage: IEEE, 1998: 387-391.
- [4] Yang X S. Firefly algorithm, stochastic test functions and design optimisation[J]. *International Journal of Bio-Inspired Computation*, 2010, 2(2): 78-84.
- [5] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459-471.
- [6] Yang X S, Hossein Gandomi A H. Bat algorithm: A novel approach for global engineering optimization[J]. *Engineering Computations*, 2012, 29(5): 464-483.
- [7] Arora S, Singh S. Butterfly optimization algorithm: A novel approach for global optimization[J]. *Soft Computing*, 2019, 23(3): 715-734.
- [8] 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.
- [9] Sayed G I, Khoriba G, Haggag M H. A novel chaotic salp swarm algorithm for global optimization and feature selection[J]. *Applied Intelligence*, 2018, 48(10): 3462-3481.
- [10] Fan Q, Chen Z J, Xia Z H. 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.
- [11] Gholami K, Parvaneh M H. A mutated salp swarm algorithm for optimum allocation of active and reactive power sources in radial distribution systems[J]. *Applied Soft Computing*, 2019, 85: 105833.
- [12] Ibrahim R A, Ewees A A, Oliva D, et al. Improved salp swarm algorithm based on particle swarm optimization for feature selection[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2019, 10(8): 3155-3169.
- [13] Hegazy A E, Makhlof M A, El-Tawel G S. Improved salp swarm algorithm for feature selection[J]. *Journal of King Saud University — Computer and Information Sciences*, 2020, 32(3): 335-344.
- [14] Wu J, Nan R J, Chen L. Improved salp swarm algorithm based on weight factor and adaptive mutation[J]. *Journal of Experimental and Theoretical Artificial Intelligence*, 2019, 31(3): 493-515.
- [15] Wang M Q, Wang Y, Ji Z C. Permanent magnet synchronous motor multi-parameter identification based on improved salp swarm algorithm[J]. *Journal of System Simulation*, 2018, 30(11): 4284-4291.
- [16] Nickabadi A, Ebadzadeh M M, Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight[J]. *Applied Soft Computing*, 2011, 11(4): 3658-3670.
- [17] Zhang D M, Chen Z Y, Xin Z Y, et al. Salp swarm algorithm based on craziness and adaptive[J]. *Control and Decision*, 2020, 35(9): 2112-2120.
- [18] Kong Z, Yang Q F, Zhao J, et al. Adaptive adjustment of weights and search strategies-based whale optimization algorithm[J]. *Journal of Northeastern University: Natural Science*, 2020, 41(1): 35-43.
- [19] Wang J H, Zhang L, Shi C, et al. Whale optimization algorithm based on chaotic search strategy[J]. *Control and Decision*, 2019, 34(9): 1893-1900.
- [20] 顾清华, 李学现, 卢才武, 等. 求解高维复杂函数的遗传-灰狼混合算法[J]. *控制与决策*, 2020, 35(5): 1191-1198.  
(Gu Q H, Li X X, Lu C W, et al. Hybrid genetic grey wolf algorithm for high dimensional complex function optimization[J]. *Control and Decision*, 2020, 35(5): 1191-1198.)
- [21] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [22] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. *Knowledge-based Systems*, 2015, 89: 228-249.
- [23] Mirjalili S, Lewis A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [24] Yang X S. Optimization techniques and applications with examples[M]. New Jersey: John Wiley and Sons, 2018, 1-364.
- [25] Fu C M, Xu Y D, Jiang C, et al. Improved differential evolution with shrinking space technique for constrained optimization[J]. *Chinese Journal of Mechanical Engineering*, 2017, 30(3): 553-565.
- [26] Liu J S, Mao Y N, Liu X Z, et al. A dynamic adaptive firefly algorithm with globally orientation[J]. *Mathematics and Computers in Simulation*, 2020, 174: 76-101.
- [27] Mezura-Montes E, Coello C A C. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems[J]. *International Journal of General Systems*, 2008, 37(4): 443-473.
- [28] Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization[J]. *Computers and Structures*, 2012, 112/113: 283-294.
- [29] Gandomi A H, Yang X S, Alavi A H, et al. Bat algorithm for constrained optimization tasks[J]. *Neural Computing and Applications*, 2013, 22(6): 1239-1255.
- [30] Sahoo A, Jagannathan S. A novel meta-heuristic optimization algorithm: Thermal exchange optimization[J]. *Advances in Engineering Software*, 2017, 110: 69-84.

## 作者简介

白钰(1985—),男,博士生,从事智能优化、机电控制的研究, E-mail: baiyusd@163.com;

彭珍瑞(1972—),男,教授,博士生导师,从事智能优化、机电控制等研究, E-mail: pzrui@163.com.

(责任编辑: 闫妍)