一种改进的樽海鞘群算法 *

陈连兴 [a†]，牟永敏 [b]

(北京信息科技大学 a. 网络文化与数字传播北京市重点实验室; b. 计算机学院, 北京 100101)

摘 要：针对樽海鞘群算法在对函数优化问题求解上，出现的求解精度不高，收敛速度慢的缺点，提出了一种改进的群海鞘群算法。对于领导者引入加权重心，取代最优个体位置，防止过早聚集在最优个体附近。对于追随者引入自适应惯性权重，平衡算法的全局搜索和局部寻优能力。最后对于个体进行逐维随机差分变异，减少维间干扰，提高了种群的多样性。仿真实验结果表明改进的樽海鞘群算法在均值、标准差和收敛曲线优于标准樽海鞘群算法和其他改进算法。说明了改进后的算法提高了寻优性能，有较高的求解精度和较快的收敛速度。

关键词：樽海鞘群算法；加权重心；逐维变异；惯性权重；函数优化

中图分类号：TP301 doi: 10.19734/j.issn.1001-3695.2020.09.0242

Improved salp swarm algorithm

Chen Lianxing[a†], Mu Yongmin[b]

(*a. Beijing Key Laboratory of Internet Culture & Digital Dissemination Research*, *b. Computer School*, *Beijing Information Science & Technology University, Beijing* 100101, *China*)

Abstract: This paper proposed an improved salp swarm algorithm that aimed to solve the problem of low precision and slow convergence speed in solving function optimization problems. For leaders, this paper replaced the optimal individual position with weighted centroid to prevent premature aggregation near the optimal individual. This paper also introduced adaptive inertia weight to the followers to balance the global search and local optimization capabilities of the algorithm. Furthermore, this paper contributed to reducing the inter dimensional interference and improving the diversity of the population by dimensional random difference mutation. The simulation results illustrate that the improved algorithm is better than the standard tympanum group algorithm and other improved algorithms in terms of mean, standard deviation, and convergence curve. The results also show that the improved algorithm improves the performance of optimization, and has higher precision as well as faster convergence speed.

Key words: salp swarm algorithm; weighted centroid; dimension by dimension mutation; inertia weight; function optimization

0 引言

群体智能优化算法通过模拟自然群体的行为为解决复杂的优化问题提供了强大的工具，并已广泛应用于许多领域[1-2]。以粒子群优化算法[3]为代表的早期群体智能优化算法取得了良好的效果。近几年来，随着人们对自然界人类的不断的了解，出现了新的群体智能优化算法。例如，灰狼优化算法[4]，正弦余弦算法[5]，鲸鱼优化算法[6]，人工蜂群优化算法[7]，樽海鞘群算法(Salp Swarm Algorithm，SSA)[8]等算法。无论群体智能优化算法的结构如何，它们都分为两个主要阶段，探索和开发。在探索阶段，算法可以通过随机化有效的发现搜索空间。在开发阶段，算法汇聚到最优的区域。但是，在大多数情况下，这两个阶段之间没有明确的界限，这使得群智能优化算法陷入局部最优。

在这之中，樽海鞘群算法自从 2017 年被提出以来，就开始被广泛使用[9-10]。而且有很多学者从不同的角度进行改进。例如：文献[11]提出了固定惯性权重的 SSA，能够使收敛速度加快；文献[12]提出了基于混沌理论的 SSA；文献[13]提出了基于疯狂自适应的 SSA，在食物源位置上引入疯狂算子，在追随者位置更新公式中引入自适应惯性权重；文献[14]提出了混沌精英质心拉伸机制的 SSA，选择最优个体进行精英重心拉伸机制，增强全局搜索能力；文献[15]对领导者个体执行精英反向学习策略，引入差分策略来更新追随者位置，在搜索过程中对食物位置进行 Gauss 变异以避免陷入局部最优；文献[16]利用莱维飞行策略对领导者位置进行随机更新，对追随者位置更新公式进行修改，使追随者不再盲目跟随；文献[17]将粒子群优化算法的随机惯性权重引入到追随者位置更新公式中，用差分进化算法的变异操作替代领导者位置更新操作；文献[18]在领导者更新公式中添加衰减因子，在追随者更新公式引入动态学习策略；文献[19]面向全局搜索的自适应领导者樽海鞘群算法，在领导者公式中引入上一代樽海鞘位置和惯性权重，并且自适应调整领导者和追随者的数量。上述算法在不同方面，针对樽海鞘群算法的缺点进行弥补，但樽海鞘群算法仍然没有完全解决平衡全局搜索与局部搜索，和易陷入局部最优解的问题。

受上述文献启发，提出了一种改进的樽海鞘群算法(Modified Salp Swarm Algorithm，MSSA)。a)将加权重心引入领导者更新公式，可以充分利用种群中的信息，同时引入非线性参数来限制个体对自身位置的依赖；b)将自适应惯性权重引入到追随者更新公式，搜索前期权重较大，算法全局搜索能力强，搜索后期权重较小，算法局部寻优能力强；c)对于更新完成的个体，使用逐维随机差分变异，减少维间干扰，增加种群多样性。改进后的算法提高了算法收敛速度，有效平衡了算法全局搜索能了和局部搜索能力，增强了跳出局部

收稿日期：2020-09-22；**修回日期：**2020-11-16 **基金项目：**北京市自然科学基金资助项目(Z160002)；网络文化与数字传播北京市重点实验室开放课题(5221935409)

作者简介：陈连兴(1996-)，男(通信作者)，河北沧州人，硕士研究生，主要研究方向为智能计算、软件理论与应用(ischenlx@163.com)；牟永敏(1961-)，男，山东烟台人，教授，博士，主要研究方向为软件理论与应用.

最优的能力。对 10 个标准函数进行了仿真对比实验，实验结果证明了改进算法在收敛速度和收敛精度上优于标准樽海鞘群算法和改进的樽海鞘群算法，说明了改进的有效性。

1　樽海鞘群算法

樽海鞘群算法是 Mirjalili 等人于 2017 年提出的一种全新的群体智能优化算法。该算法的主要思想来源于樽海鞘的群聚行为。在 SSA 算法中将樽海鞘群分为两组，领导者和追随者。在樽海鞘链中领导者处于前面，其余的个体为追随者。和其他的群体智能优化算法不同在于，领导者不会影响整个群体的移动，追随者会根据前一个个体的位置来更新自己的位置，依此类推形成了樽海鞘链。领导者对于排在后面的追随者的领导作用会越来越弱，在后面的追随者不会一味的向着领导者移动，保持了种群的多样性。

在樽海鞘群算法中，设樽海鞘群 X 在 D 维空间搜索，X 由 N 个维度为 D 的樽海鞘个体组成，搜索空间上限为 $ub=[ub_1 \quad ub_2 \quad \cdots \quad ub_D]$，搜索空间下限为 $lb=[lb_1 \quad lb_2 \quad \cdots \quad lb_D]$。樽海鞘种群矩阵如下所示。

$$\boldsymbol{X}=\begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_D^N \end{pmatrix} \tag{1}$$

根据文献[8]，基本樽海鞘群算法主要分为三个步骤。

1)种群初始化

与其他群体智能优化算法类似，SSA 算法通过生成随机数的方式进行种群的初始化，即

$$X=lb+rand(N,D)\times(ub-lb) \tag{2}$$

2)领导者位置更新

领导者负责搜索食物来领导整个群体的移动方向，领导者位置更新公式为

$$x_j^1=\begin{cases} F_j+c_1((ub_j-lb_j)c_2+lb_j) & c3\geq 0.5 \\ F_j-c_1((ub_j-lb_j)c_2+lb_j) & c3<0.5 \end{cases} \tag{3}$$

其中，x_j^1 为作为领导者的第 1 只樽海鞘个体在第 j 维上的位置；ub_j 和 lb_j 分别是第 j 维的上下限；F_j 为食物在第 j 维上的位置；c_2 和 c_3 是在 [0,1] 范围内均匀生成的两个随机数，c_1 是负责平衡探索和开发的收敛因子，公式如下：

$$c_1=2e^{-(\frac{4t}{T})^2} \tag{4}$$

其中：t 为当前的迭代次数，T 为最大迭代次数。根据式(3)中所示的运动策略，领导者以食物位置为中心并在整个搜索区域持续震荡。在初始阶段，c_1 的值比较大，接近 2，领导者在比较大的范围内去搜索。在后期阶段，c_1 的值接近于 0，领导者接近食物。因此，领导者使得整个算法在开始阶段不易陷入局部最优，后期提高收敛的速度。

3)追随者位置更新

追随者根据牛顿运动定律更新位置，公式如下：

$$x_j^i=\frac{1}{2}at^2+v_0t \tag{5}$$

其中，x_j^i 为第 i 个追随者在第 j 维上的位置，$i\geq 2$，t 是时间，v_0 为初速度，加速度 $a=\frac{v_{final}-v_0}{\Delta t}$，$v_{final}=\frac{(x_j^{i-1}-x_j^i)}{\Delta t}$，因为在算法中迭代次数为时间，每次迭代之间的差值为 1，$v_0=0$，式(5)可表示为

$$x_j^i=\frac{1}{2}(x_j^i+x_j^{i-1}) \tag{6}$$

由于在实际的迭代过程中，目标(食物)的位置并不明确，因此在迭代的过程中，计算所有樽海鞘个体的适应度值，并将最优适应度值的樽海鞘的位置设置为当前食物位置。

2　改进的樽海鞘群算法

2.1　加权重心学习策略

标准樽海鞘群算法中，所有领导者都向食物源进行学习，忽略了种群内其他优秀的个体。文献[20]提出重心反向学习，重心中包含了群体搜索经验，但是重心中所有个体的贡献相同。本文提出了加权重心，根据个体的优劣状况使用不同的权重来计算重心，可以在不忽略较差个体的同时，向更多优秀个体学习，合理利用了种群的信息，同时避免了只向最优个体学习陷入早熟。重心定义如下：

定义 1　重心设 $(X^1,\cdots, X^N)$ 是 D 维空间中的 n 个点，则整体重心为

$$M=\frac{x^1+\cdots+x^n}{n} \tag{7}$$

加权重心公式如下：

$$M_W=\frac{\sum_{i=1}^{n}x^i(\frac{f_{bad}-f_{X^i}}{\alpha})+F(\frac{f_{bad}-f_F}{\alpha})}{n+1} \tag{8}$$

$$\alpha=\sum_{i=1}^{n}(f_{bad}-f_{X^i})+(f_{bad}-f_F) \tag{9}$$

其中：M_W 为加权重心，f_{bad} 为当前种群最差个体适应度值，f_F 为食物源适应度值，f_{x^i} 为当前种群 x^i 个体的适应度值，F 为食物源位置，$(f_{bad}-f)/\alpha$ 是个体的权重，越优秀的个体其权重越大，在加权重心中占比也就越大，同时在加权重心引入食物源的位置和权重，可以更加充分利用全局种群信息。

修改后的领导者公式为

$$x^i=\omega x^i+rand*(M_W-x^i) \tag{10}$$

$$\omega=(l/l_{\max})^2 \tag{11}$$

修改后的公式可以在向最优解逐渐学习的同时，没有过多丢失种群多样性。ω 用来调节个体在搜索过程中对自身位置的依赖。搜索前期，较低的 ω 可以降低个体对自身的依赖，从而增大了搜索的范围，算法的全局搜索能力得到增强。搜索后期，ω 逐渐向 1 靠拢，不影响算法寻优。

2.2　自适应惯性权重

自适应惯性权重在很多群体智能优化算法中被使用，搜索前期权重较大，可以增强全局搜索能力，搜索后期自适应权重较小，可以增强局部寻优能力。使用的自适应惯性权重公式如下：

$$\omega=\lambda^{(-\varphi\times t/T)}\times\cos(\pi\times t) \tag{12}$$

将式(12)应用到(6)中，新的追随者更新公式如下：

$$x_j^i=\frac{1}{2}(x_j^i+\omega\times x_j^{i-1}) \tag{13}$$

λ=2，φ=4/3 时，式(12)的值为从-1 和 1 逐渐向-0.4 和 0.4 收敛，文献[13]中所提出自适应惯性权重从 0.9 线性递减到 0.4，相较于文献[13]中的自适应惯性权重，式(12)可以再不提高计算复杂度的同时，提高算法前期的搜索范围，增加种群多样性。

2.3　逐维随机差分变异

在基本樽海鞘群算法中，领导者受到食物源的位置来更新自己的位置，追随者追随上一个个体进行更新。如式(3)(6)所示，领导者 X 的更新是在食物源头 F 的附近产生新的个体，追随者再根据领导者的位置进行位置更新。因此食物源的位置将直接影响整个种群的搜索方向，如果食物源陷入局部最优，那么种群会在食物源附近不断搜索，无法跳出局部最优，降低了种群的多样性，从而算法只能得到局部最优解。

为了解决这一问题，常用的方法为加入变异操作，对更新完的个体进行变异，增强了种群的多样性，从而跳出局部最优。常使用的变异算子有高斯变异、柯西变异等。通常变

异均是对所有维度同时进行变异，再根据目标函数来评判，这样对于高维函数，维度间会相互干扰，造成有些维度经过变异得到了更好的解，但是由于其他维度经过变异，变异效果不好，且变异效果差的维度覆盖了变异效果好的维度，使个体最后变异效果不佳，进而影响了算法的收敛速度和精度。使用逐维变异，可以避免高维函数之间每个维度相互干扰，从而提高变异解的质量。

使用随机差分变异[21]进行逐维变异，通过该变异得到一个新的个体的维度，具体公式为

$$x_j^i = r_1 \times (F_j - x_j^i) + r_2 \times (x_j^{'} - x_j^i) \tag{14}$$

其中：x_j^i 为樽海鞘群中的第 i 个个体的第 j 维，F_j 为食物源位置的第 j 维，$x_j^{'}$ 为种群中随机的一个个体的第 j 维，r_1 和 r_2 为 [0,1] 的随机数。在种群位置更新完成后，使用逐维随机

差分变异对个体的每个维度进行变异，某一维度进行变异后，对其进行评价，如果优秀，则保留变异后的解，如果变异后，评价结果变差，则舍弃较差的维度信息，减少了各个维度间的干扰，同时增大了搜索的范围。

由于变异操作具有一定的盲目性，把所有个体都进行逐维随机差分变异势必会导致算法的搜索效率下降和计算量大幅的增加，因此仅挑选种群中最优秀和最差个体进行变异，对最优个体变异，可以提高搜索效率，对最差个体变异，可以提高搜索范围，跳出局部最优解。

2.4 改进后的算法

为了算法在迭代的前期能够有较强的全局搜索能力，选取种群中前一半的个体作为领导者，增多领导者，可以增强算法的随机性。算法具体步骤如下：

a) 初始化种群和参数。初始化种群个体数量 N，最大迭代次数 T。使用式(2)生成初始种群。

b) 计算种群每个个体的适应度，最优的个体作为食物位置。

c) 根据位置更新公式更新个体位置。前一半个体为领导者使用式(10)进行更新，后一半个体为追随者使用式(13)进行更新。

d) 对更新完的个体，选择最优和最差个体，对其通过式(14)进行逐维随机差分变异，将更新的维度与其余维度组成新的个体，比较变异前后个体适应度值的变化，如果好则保留。

e) 找出最优个体适应度值，更新食物位置

f) 判断是否满足迭代次数要求或精度要求，若是进行步骤 g)，否则返回步骤 c)。

g) 输出最优个体的适应度值。

MSSA 算法主要由初始化、领导者位置更新、追随者位置更新和逐维随机差分变异四部分组成。假设算法迭代次 T，种群规模为 N，维度 D。

a) 种群初始化时间复杂度为 $O(ND)$。

b) 计算加权重心的时间复杂度为 $O(T)$。

c) 领导者更新的时间复杂度为 $O(NDT)$。

d) 追随者更新的时间复杂度为 $O(NDT)$。

e) 逐维变异的时间复杂度为 $O(kDT)$，其中 k 是变异的个体数量；

f) 更新种群适应度值时间复杂度为 $O(NT)$

算法整体复杂度为

$$O(ND)+O(T)+O(2NDT)+O(kDT)+O(NT)=O(CNDT)$$

其中：C 是常数。

3 实验仿真

3.1 参数设置和实验环境

实验中所有算法均设置参数为种群规模为 30，最大迭代次数为 500。实验环境为：Windows10 64bit 系统，CPU 3.7GHz，16GB 内存，实验仿真软件为 MATLAB R2016a，使用 M 语言编写。

3.2 标准测试函数

为验证改进的樽海鞘群算法是有效的，选取了 10 个维度从 10-100 的基准函数进行对比实验，如表 1 所示，其中 F1-F5 为单峰函数，主要用于检验算法的计算精度和收敛速度，F6-F10 为多峰函数，主要检验算法全局优化能力。

表 1　基准函数

Tab. 1　Benchmark functions

函数	维度	范围	最优值
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	50	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i}x_j)^2$	100	[-100,100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	50	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	10	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	50	[-5.12,5.12]	0
$F_7(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[-32,32]	0
$F_8(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	100	[-600,600]	0
$F_9(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n}u(x_i,10,100,4)$，$y_i=1+\frac{x_i+1}{4}$，$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\0 & -a<x_i<a\\k(-x_i-a)^m & x_i<-a\end{cases}$	10	[-50,50]	0
$F_{10}(x)=\sum_{i=1}^{n}\lvert x\sin(x)+0.1x\rvert$	10	[-100,100]	0

3.3 实验结果分析

3.3.1 各个策略对算法搜索影响

为了检验加权重心自学习、自适应惯性权重和逐维随机差分变异对樽海鞘群算法的改进效果，将自适应惯性权重和随机余弦变异分别应用到樽海鞘群算法上。实验中所有算法均设置参数为种群规模为 30，最大迭代次数为 500。使用 MATLAB 分别进行独立的 30 次仿真实验，对所得出的最优值进行求平均值和标准差，以避免因为算法的随机性而带来的偏差，从而保证实验的公平性与合理性。

通过表 2 发现，这三种改进策略的樽海鞘群算法在均值和标准差上均优于标准樽海鞘群算法。其中自适应惯性权重和加权重心学习策略的效果要优于逐维差分变异的，这说明自适应惯性权重的平衡开发和探索、加权重心学习策略的利用整个种群资源和平衡对自身的依赖都起到了作用。逐维随机差分变异的主要作用是跳出局部最优解，因此效果弱于其他两种策略，在多峰函数 F6 中，逐维随机差分变异和自适应惯性权重占算法优化的主导地位，这说明函数在寻优过程中多次陷入局部最优解，通过逐维变异跳出了局部最优。所以改进的樽海鞘群算法是以加权重心学习策略和自适应惯性权重为主导，逐维随机差分变异为辅助的改进算法。

3.3.2 改进算法与其他改进算法对比实验

选取了基本樽海鞘群算法(SSA)[8]、和 6 个改进的樽海鞘群算法，分别为文献[13]中的基于疯狂自适应的樽海鞘群算法(CASSA) 、文献[15]中的改进的樽海鞘群算法(ISSA)、文献[16]中的基于 Levy 飞行策略条件化更新的樽海鞘群算法(LECUSSA)、文献[17]中的集成随机惯性权重和差分变异操

作的樽海鞘群算法(iSSA)、文献[18] 基于衰减因子和动态学习的改进樽海鞘群算法(RDSSA)、文献[19] 面向全局搜索的自适应领导者樽海鞘群算法(ALSSA)和提出的改进的樽海鞘群算法(Modified salp swarm algorithm,MSSA)进行对比实验。实验中所有算法均设置参数为种群规模为 30，最大迭代次数为 500。使用 MATLAB 分别进行独立的 30 次仿真实验,其中 MSSA、CASSA 和 ISSA 单独设置的参数，如表 3 所示，对所得出的最优值进行求平均值和标准差。

表 2　不同策略影响

Tab. 2　Experiments affected by different strategies

函数	SSA		加权重心学习		逐维随机差分变异		自适惯性应权重	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F1	1.5466e-07	2.2648e-07	1.8501e-54	2.7851e-54	2.0791e-14	7.1546e-15	3.4669e-75	1.1472e-73
F2	9.861	3.1091	8.6231e-30	6.3949e-30	9.3722e-08	1.186e-08	1.3233e-39	1.6384e-37
F3	55438.4176	31178.3151	6.199e-51	7.2811e-51	3.9937e-12	4.6839e-12	2.4107e-73	1.6457e-71
F4	20.9363	2.2196	3.7585e-30	2.341e-30	7.5826e-08	5.7279e-08	1.3623e-38	1.6466e-37
F5	49.3291	88.3824	8.9328	0.056337	8.4304	0.06634	9.1699	0.13648
F6	85.6195	26.3002	0	0	2.8422e-14	9.9273e-14	0	0
F7	2.5955	0.86775	8.8818e-16	0	6.1107e-08	4.2364e-08	8.8818e-16	0
F8	13.3378	2.5416	0	0	1.5775e-13	2.3222e-13	0	0
F9	1.3558	2.5201	1.9262e-06	3.5769e-06	6.0996e-11	1.0616e-10	4.6392e-11	4.2973e-11
F10	0.001045	0.0016939	3.9205e-32	2.3782e-32	2.156e-10	1.5977e-10	1.4387e-41	4.8098e-40

表 3　算法参数设置

Tab. 3　Algorithm parameter setting

算法	主要参数
MSSA	λ=2 ， φ=4/3
CASSA	$\omega_s=0.9$ ， $\omega_e=0.9$ ， $P_{cr}=0.3$ ， $x_{craziness}=0.0001$
ISSSA	$S_{\max}=10$

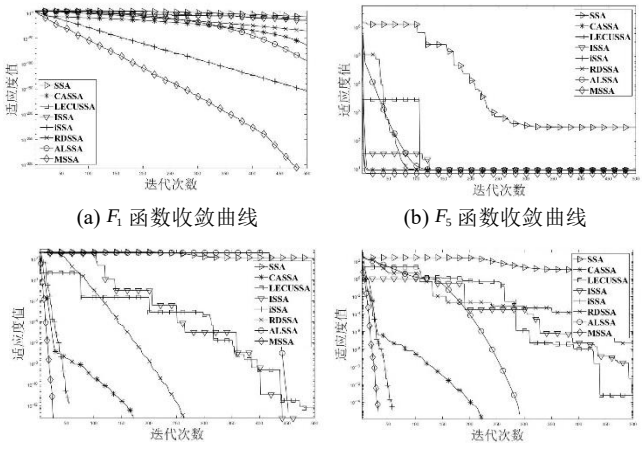
通过分析表 4，函数 F1-F5 和 F9-F10 改进的算法在均值上明显优于其他改进的算法和标准算法，其中 F1 和 F3 搜索到了最优值。函数 F6 和 F8 改进的算法和部分其他改进的算法能够搜索到最优值。函数 F7 改进的算法与 RDSSA、CASSA、iSSA 和 ALSSA 平均值相同，优于 SSA、ISSA 和 LECUSSA。因此整体上改进的樽海鞘群算法在平均值上是优于 SSA、ISSA、LECUSSA、CASSA、RDSSA、iSSA 和 ALSSA 的。平均值能够体现出算法的收敛性，这说明了改进的樽海鞘群算法比其他对比算法的收敛性好，有着更高的收敛精度。对于标准差，只有在 F5 中 LECUSSA、RDSSA、ALSSA 优于改进的算法标准差，但其均值明显劣于改进后的算法，其他函数中改进后的算法标准差优于其他算法或者均可稳定搜索到最优解。因此改进的樽海鞘群算法在标准差上是优于 SSA、ISSA、LECUSS、CASSA、RDSSA、iSSA 和 ALSSA 的。标准差可以反映出数据集的离散程度，标准差越小说明数据集的离散程度越低，实验结果的稳定性也就越好，这说明改进的樽海鞘群算法的比其他算法，其实验结果的数据集的离散程度低，稳定性好。

虽然改进的樽海鞘群算法在最优解的平均值和标准差上优于标准樽海鞘群算法和其他改进算法，但是根据表 4 可以看出，改进算法在多峰函数求解的精度低于单峰函数，说明算法在对于多峰函数求解上仍存在不足，这是进一步研究的方向。

图 1 为函数 F1、F5、F6 和 F8 的函数收敛曲线，发现 F1、F6 和 F8 的收敛曲线中，改进的算法的收敛速度明显优于其他算法，其中 F6 和 F8 在均在 50 次以内的迭代次数内就收敛到最优解，说明算法的搜索能力强。函数 F5 的收敛曲线，改进的算法在搜索前期收敛速度与 RDSSA 近似，优于其他算法，说明在求解精度固定的情况下，改进的算法可以在很少的迭代次数内搜索到结果。

通过 30 次独立运行的结果的均值和标准差进行分析，无法准确的分析每次的结果，为了验证 MSSA 的运行结果与其他算法运行结果的显著差异性，使用 Wilcoxon 秩和检验对结果进行分析，在 5%的置信度下进行。将每个其他算法与 MSSA 进行成对比较，实验结果如表 5 所示。其中 NaN 表示相应算法在秩和检验中没有数据与自身进行比较 ，当 p-value 的值小于 0.05 时，说明改进算法优于其他算法。

根据表 5 中的结果，MSSA 的 p-value 均小于 0.05，因此 MSSA 性能优于其他改进算法。



(a) F_1 函数收敛曲线　(b) F_5 函数收敛曲线

(c) F_6 函数收敛曲线　(d) F_8 函数收敛曲线

图 1　基准函数收敛曲线

Fig. 1　Benchmark functions convergence curve

平均绝对误差(Mean Absolute Error,MAE)是可以评估优化算法性能的指标[22],公式如下：

$$MAE=\frac{\sum_{i=1}^{N}|m_i-o_i|}{N} \tag{15}$$

其中：N 为被测函数个数，m_i 和 o_i 分别是第 i 个被测函数搜索最优平均值和理论最优值，实验结果如表 6 所示，表中 MSSA 的 MAE 值是最小的，表明 MSSA 优于其他改进算法。

基于对比实验的结果，表明了改进的樽海鞘群算法在收敛精度和算法稳定性上优于其他算法，收敛速度优于大多数的改进算法。

3.3.3 改进的樽海鞘群算法运行时间分析

智能优化算法在实际工程问题上有广泛的应用，因此测试算法的运行时间具有必要性，由于受到标准樽海鞘群算法收敛精度的限制，故只选择了部分基准函数，且修改维度的情况下进行测试。

表 7 列出了 SSA 和 MSSA 在求解相同精度的情况下，独立运行 30 次所消耗的时间，因为将加权重心引入领导者

更新公式，可以充分利用种群中的信息，使得在算法搜索前期向优秀个体附近靠拢，且没有大幅度降低种群多样性，非线性参数来限制个体对自身信息的依赖，提高了算法全局搜索能力；将自适应惯性权重引入到追随者更新公式，平衡了算法的全局探索和局部开发能力；对于更新完成的个体，使用逐维随机差分变异，减少维间干扰，生成更好的变异解，避免陷入局部最优，防止出现早熟。所以消耗的时间均短于标准 SSA，说明 MSSA 的搜索效率是高于标准 SSA 的。

表 4　基准函数优化结果

Tab. 4　Results of benchmark functions

函数	指标	SSA	LECUSSA	ISSA	RDSSA	CASSA	iSSA	ALSSA	MSSA
F1	Mean	1.5466e-07	6.7507e-13	9.781e-15	1.0742e-35	1.3797e-64	4.979e-152	4.1924e-96	0
	Std	2.2648e-07	1.1576e-12	4.2634e-14	3.0152e-36	2.4917e-65	2.0486e-151	4.2279e-96	0
F2	Mean	9.861	5.3301e-07	4.3987e-09	2.4828e-18	7.3246e-32	2.161e-77	5.2419e-48	3.6117e-189
	Std	3.1091	4.5624e-07	1.0416e-08	2.2521e-19	2.8009e-33	3.7252e-77	3.1479e-48	6.2295e-189
F3	Mean	55438.4176	1.4221e-08	1.1998e-16	3.8862e-34	7.8496e-61	5.157e-148	1.1212e-85	0
	Std	31178.3151	3.4103e-08	2.9149e-16	2.5521e-34	1.4722e-61	1.2991e-147	1.6953e-85	0
F4	Mean	20.9363	9.5024e-08	3.3191e-10	1.3463e-18	4.8636e-33	3.7578e-60	7.1889e-46	1.4048e-145
	Std	2.2196	7.483e-08	8.6529e-10	2.1771e-19	7.2301e-34	1.623e-59	4.3664e-46	2.5637e-145
F5	Mean	49.3291	8.9996	8.1271	8.9633	8.3829	6.2302	8.9707	5.8508
	Std	88.3824	0.0013384	0.18251	0.014866	0.092198	4.0783	0.027583	0.06298
F6	Mean	85.6195	2.0464e-13	0	0	0	0	0	0
	Std	26.3002	6.0516e-13	0	0	0	0	0	0
F7	Mean	2.5955	0.99834	9.0981e-10	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16
	Std	0.86775	4.3517	3.2348e-09	0	0	0	0	0
F8	Mean	13.3378	1.5451e-12	3.0531e-16	4.9965e-08	0	0	0	0
	Std	2.5416	2.6785e-12	1.3308e-15	9.1573e-08	0	0	0	0
F9	Mean	1.3558	8.8171e-07	2.932e-11	0.57779	5.6431e-09	8.3053e-08	1.9746	2.3725e-16
	Std	2.5201	1.3997e-06	2.3214e-11	0.25847	7.0668e-09	4.6032e-08	0.86189	7.8371e-16
F10	Mean	0.001045	3.5861e-07	3.2965e-11	4.6775e-21	1.4626e-33	1.798e-05	6.8373e-18	1.509e-147
	Std	0.0016939	1.5575e-06	1.2212e-10	1.1883e-21	1.0103e-34	1.5877e-05	2.9803e-17	3.0764e-147

表 5　MSSA 与其他 6 种算法统计检验结果

Tab. 5　Statistical test results of MSSA and other six algorithms

函数	SSA p-value	LECUSSA p-value	ISSA p-value	RDSSA p-value	CASSA p-value	iSSA p-value	ALSSA p-value
F1	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09
F2	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08
F3	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09	8.0065e-09
F4	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08
F5	1.0992e-05	6.7478e-08	6.7478e-08	6.7478e-08	6.7478e-08	0.0315	6.7478e-08
F6	8.0065e-09	0.00015638	NaN	NaN	NaN	NaN	NaN
F7	8.0065e-09	8.0065e-09	1.0536e-07	NaN	NaN	NaN	NaN
F8	8.0065e-09	8.0065e-09	7.8609e-09	8.0065e-09	NaN	NaN	NaN
F9	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08
F10	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08	6.7956e-08

表 6　MAE 算法排名

Tab. 6　MAE algorithm ranking

算法	MAE	排名
MSSA	0.5851	1
SSA	5.5621e+03	8
iSSA	0.7524	3
ISSA	0.6230	2
LECUSSA	0.9998	6
CASSA	0.8383	4
RDSSA	0.9541	5
ALSSA	1.0945	7

4　结束语

针对 SSA 所存在的缺点，在基础算法之上，提出了一种改进的樽海鞘群算法，改进的算法引入了加权重心学习策略、自适应惯性权重和逐维随机差分变异，选取了 10 个基准函数进行实验。实验结果表明，三个改进的策略均是有效的，且改进后收敛精度和算法稳定性上优于基本樽海鞘群算法和其他改进的算法，收敛速度优于大多数的改进的算法。

表 7　算法运行时间

Tab. 7　Algorithm running time

函数	维度	精度	SSA/s	MSSA/s
F1	30	10^{-5}	0.103118	0.010837
F2	10	10^{-5}	0.100599	0.009315
F3	10	10^{-5}	0.151823	0.018133
F4	10	10^{-5}	0.097266	0.007673
F6	10	10^{-1}	0.112655	0.010783
F7	10	10^{-5}	0.105522	0.014597
F9	10	10^{-5}	0.402745	0.200558
F10	10	10^{-1}	0.215859	0.034226

参考文献：

[1] A. Greco, A. Pluchino, F. Cannizzaro. An improved ant colony optimization algorithm and its applications to limit analysis of frame

structures [J]. Engineering Optimization, 2019, 51 (11): 1867-1883.

[2] 郝群茹，潘帅. 基于禁忌搜索算法的物流系统车辆路径优化 [J]. 科学技术与工程, 2019, 19 (34): 401-407. (Hao Qunru, Pan Shuai. Vehicle routing optimization of logistics system based on Tabu search algorithm [J]. Science Technology and Engineering, 2019, 19 (34): 401-407.)

[3] Kennedy J, Eberhart R C. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.

[4] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69 (3): 46−61.

[5] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2016, 96 (3): 120-133.

[6] Mirjalili S, Lewis A. The Whale Optimization Algorithm [J]. Advances in Engineering Software, 2016, 95 (5): 51-67.

[7] Gao W, Liu S, Huang L. A novel artifificial bee colony algorithm based on modifified search equation and orthogonal learning [J]. IEEE Transactions on Cybernetics, 2013, 43 (3): 1011−1024.

[8] Mirjalili S, Gandomi A H, Mirjalili S Z, *et al.* Salp swarm algorithm: a bio-inspired optimizer for engine-ering design problems [J]. Advances in Engineering Software, 2017, 114 (6): 163-191.

[9] 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 (07): 1591-1597. (Chen Tao, Wang Mengxin, Huang Xiangsong. Time Difference of Arrival Passive Location Based on Salp Swarm Algorithm [J]. Journal of Electronics and Information Technology, 2018, 40 (07): 1591-1597.)

[10] 杨博，钟林恩，朱德娜，等. 部分遮蔽下改进樽海鞘群算法的光伏系统最大功率跟踪 [J]. 控制理论与应用, 2019, 36 (03): 339-352. (Yang Bo, Zhong Linen, Zhu Dena, *et al*. Modified salp swarm algorithm based maximum power point tracking of power-voltage system under partial shading condition [J]. Control Theory and Applications, 2019, 36 (03): 339-352.)

[11] Hegazy A E, Makhlouf M A, Eltawel G S. Improved salp swarm algorithm for feature selection [J]. Journal of King Saud University–Computer and Information Sciences, 2018, 6 (3): 1-10.

[12] Gehad I S, Ghada K, Mohamed H H. A novel chaotic salp swarm algorithm for global optimization and feature selection [J]. Applied Intelligence, 2018, 48: 3462–3481.

[13] 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 (09): 2112-2120. (Zhang Damin, Chen Zhongyun, Xin Ziyun, *et al.* Salp Swarm Algorithm Based on Craziness and Adaptive [J]. Control and Decision, 2020, 35 (09): 2112-2120.)

[14] 陈忠云，张达敏，辛梓芸，等. 混沌精英质心拉伸机制的樽海鞘群算法 [J]. 计算机工程与应用, 2020, 56 (10): 44-50. (Chen Zhongyun, Zhang Damin, Xin Ziyun, *et al.* Salp Swarm Algorithm Using Chaotic and EliteCentroid Stretching Mechanism [J]. Computer Engineering and Applications, 2020, 56 (10): 44-50.)

[15] 王彦军，王秋萍，王晓峰. 改进的樽海鞘群算法及在焊接梁问题中的应用 [J]. 西安理工大学学报, 2019, 35 (04): 484-493. (Wang Yanjun, Wang Qiuping, Wang Xiaofeng. An improved salp swarm algorithm and its application to welding beam problem [J]. XI AN University of Technology, 2019, 35 (04): 484-493.)

[16] 张严，秦亮曦. 基于 Levy 飞行策略的改进樽海鞘群算法 [J]. 计算机科学, 2020, 47 (07): 154-160. (Zhang Yan, Qin Liangxi. Imporved Salp Swarm Algorithm Based on Levy Flight Strategy [J]. Computer Science, 2020, 47 (07): 154-160.)

[17] 张志强，鲁晓锋，隋连升，等. 集成随机惯性权重和差分 变异操作的樽海鞘群算法 [J]. 计算机科学, 2020, 47 (08): 297-301. (Zhang Zhiqiang, Lu Xiaofeng, Sui Liansheng, *et al.* Salp Swarm Algorithm with Random Inertia Weight and Differential Mutation Operator [J]. Computer Science, 2020, 47 (08): 297-301.)

[18] 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 (08): 1766-1780. (Chen Lei, Lin Yue, Kang Zhilong. Improved salp swarm algorithm based on reduction factor and dynamic learning [J]. Control Theory and Applications, 2020, 37 (08): 1766-1780.)

[19] 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/OL]. 控制与决策: 1-10 [2020-08-21]. https://doi. org/10. 13195/j. kzyjc. 2020. 0090. (Liu Jingsen, Yuan Mengmeng, Zuo Ffang. Global Search-oriented Adaptive Leader Salp Algorithm [J/OL]. Control and Decision: 1-10 [2020-08-21]. https://doi. org/10. 13195/j. kzyjc. 2020. 0090.)

[20] Rahnamayan S, Jesuthasan J, Bourennani F, *et al.* Computing opposition by involving entire population [C]// Evolutionary Computation. New York: IEEE, 2014. 1800-1807

[21] 覃溪，龙文. 基于随机差分变异的改进鲸鱼优化算法 [J]. 中国 科技论文, 2018, 13 (08): 937-942. (Tan Xi, Long Wen. An improved whale optimization algorithm based on stochastic differential mutation [J]. China Sciencepaper, 2018, 13 (08): 937-942.)

[22] Emad N. A modified flower pollination algorithm for global optimization [J]. Expert Systems with Applications, 2016, 57 (9): 192-203.