

doi:10.11835/j.issn.1000-582X.2019.09.007

# 自学习策略和 Lévy 飞行的正弦余弦优化算法

李银通, 韩统, 赵辉, 王骁飞

(空军工程大学航空工程学院, 西安 710038)

**摘要:**针对正弦余弦算法(SCA, sine cosine algorithm)局部搜索能力差的缺陷,提出自学习策略和 Lévy 飞行的正弦余弦优化算法(SCASL, sine cosine optimization algorithm with self-learning strategy and Lévy flight)。首先,提出正弦余弦算法自学习策略和非线性权重因子,使搜索个体记忆自身历史最优位置,在寻优过程中指导搜索个体更新位置,提高 SCA 的局部搜索能力;算法寻优后期,当搜索陷入局部最优时,采用基于 Lévy 飞行的停滞扰动策略使算法跳出局部最优,提高 SCA 的局部最优规避能力。基于 13 个经典基准测试函数对算法性能进行测试的实验结果表明,SCASL 相比标准 SCA 和较新的优化算法 SSA, VCS, WOA, GSA, 具有更高的计算效率,收敛精度以及更强的局部最优规避能力。求解无人作战飞机航迹规划的仿真结果表明,在有 6 个敌方威胁源的战场环境中,相比 SCA, SCASL 求解得到的飞行航迹具有更小的航迹代价。综上,所提出的 SCASL 具有较强的寻优能力。

**关键词:**优化算法;正弦余弦优化算法;自学习策略;Lévy 飞行

**中图分类号:**TP301;V247 **文献标志码:**A **文章编号:**1000-582X(2019)09-055-11

## An improved sine cosine optimization algorithm with self-learning strategy and Lévy flight

LI Yintong, HAN Tong, ZHAO Hui, WANG Xiaofei

(School of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, P.R.China)

**Abstract:** In order to improve the performance of sine cosine algorithm (SCA) with poor local search ability, an sine cosine optimization algorithm with self-learning strategy and Lévy flight (SCASL) was proposed. Firstly, the self-learning strategy and nonlinear weight factor of sine cosine algorithm was proposed, so that the search individual could remember its historical optimal position, which guided the individual to update its location in the optimization process, thus improving the local search ability of SCA. When the search was stagnant, the stagnation perturbation strategy based on Lévy flight was adopted to jump out of local optimum so as to improve the local optimum avoidance ability. Based on 13 classic benchmark functions, the numerical simulation was conducted and the results show that SCASL has higher computational efficiency, convergence accuracy and stronger local optimum avoidance ability compared with standard SCA and other state-of-the-art optimization algorithms such as SSA, VCS, WOA and GSA. The

**收稿日期:**2019-01-12

**基金项目:**中国航空科学基金(20175196019);陕西省自然科学基金(2017JM6078)。

Supported by Aeronautical Science Foundation of China(20175196019) and Shaanxi Nature Science Foundation of China (2017JM6078).

**作者简介:**李银通,男,主要从事智能优化算法方向研究,(E-mail) yintongli0007@163.com。

**通信作者:**韩统,男,副教授,主要从事最优控制与武器应用研究,(E-mail)hantonghuokong@163.com

simulation results of unmanned combat aircraft flight path plan show that for the battlefield environment with six enemy threat sources, SCASL can stably obtain less expensive flight path than the standard SCA. Therefore, the proposed SCASL has better optimization performance.

**Keywords:** optimization algorithm; sine cosine optimization algorithm; self-learning strategy; Lévy flight

正弦余弦优化算法<sup>[1]</sup>(SCA, sine cosine algorithm)是澳大利亚学者 Mirjalili 于 2016 年提出的一种基于种群的智能优化算法。与大多数智能优化算法相比,SCA 具有架构简单,控制参数少,计算效率高等优点。Mirjalili<sup>[1]</sup>已经证明 SCA 在整体优化性能上优于萤火虫算法(FA, firefly algorithm)、花朵授粉算法(FPA, flower pollination algorithm)、粒子群算法(PSO, particle swarm optimization)以及遗传算法(GA, genetic algorithms)等。基于此,SCA 已经被广泛应用于求解工程实际问题<sup>[2-4]</sup>。SCA 主要依赖正弦函数和余弦函数的震荡特性进行寻优,由于种群更新对当前最优解的依赖过强,算法迭代后期种群多样性迅速降低,导致算法易于陷入局部最优。

由于 SCA 架构简单,因此具有较大的改进潜力。目前,国内外学者对 SCA 改进策略的研究主要分为 2 类。一类是与其他智能优化算法进行杂交融合以提高算法的优化性能:王蕾等<sup>[5]</sup>将 SCA 与花授粉算法(FPA)进行融合,利用 SCA 的正弦余弦震荡特性,提高融合算法的局部最优规避能力与寻优精度;Chegini 等<sup>[6]</sup>将 SCA 与粒子群算法(PSO)进行融合,提高算法的全局搜索能力和局部最优规避能力;Nenavath 等<sup>[7]</sup>将 SCA 与差分进化算法(DE, differential evolution)进行融合,提高算法的局部最优规避能力和收敛速度,并将融合后的算法应用于求解目标跟踪问题;Issa 等<sup>[8]</sup>利用 SCA 较好的全局搜索特性以及 PSO 良好的局部搜索能力提出的基于粒子群算法的自适应正弦余弦优化算法,寻优收敛精度较好。另一类是借鉴其他优化算法的搜索策略,针对性地提高 SCA 的局部搜索能力或全局搜索能力:Rizk-Allah<sup>[9]</sup>将多正交搜索策略引入 SCA,充分利用多正交搜索策略的局部搜索优势,提高算法的寻优精度与局部最优规避能力;Gupta 等<sup>[10]</sup>借鉴粒子群算法中的交叉操作,将交叉策略应用到 SCA 种群更新模型中,提高算法的局部搜索能力;Long 等<sup>[11]</sup>分别引入基于高斯分布的非线性权重因子和惯性权重,提高算法的收敛速度和避免局部最优能力;Gupta 等<sup>[12]</sup>基于扰动率将反向学习策略应用于 SCA,提高算法跳出局部最优的能力。

但是,将 SCA 与其他智能算法进行杂交融合不仅增加了 SCA 算法的复杂度,而且算法的局部搜索与全局搜索难以有效平衡;目前将其他策略引进 SCA 的研究成果,虽然在一定程度上提高了算法的局部搜索能力或全局搜索能力,但能整体上同时提高算法的收敛精度、计算效率以及规避局部最优能力的研究还相对较少。鉴于此,为提高算法的收敛精度和局部最优规避能力,对 SCA 做以下 2 个方面改进:1)受粒子群算法搜索机制启发,引入自学习策略,记录每个搜索个体搜索到的历史最优位置,减少种群更新对当前最优解的依赖,提高算法的局部搜索能力,并引入非线性权重因子平衡算法的局部搜索和全局搜索;2)当算法迭代后期搜索停滞时,采用 Lévy 飞行策略施加扰动量,使算法跳出局部最优。基于 13 个经典基准测试函数及航迹规划问题对提出的算法进行测试,验证算法的寻优性能。

## 1 相关知识

正弦余弦优化算法利用正弦函数和余弦函数的震荡特性进行寻优,随着迭代次数的增加,最终收敛于最优解或最优解附近。已知非约束  $n$  维最小化优化问题

$$\begin{aligned} [\min] f(x) &= \min f(x_1, x_2, x_3, \dots, x_n), \\ \text{s.t. } L_i &\leq x_i \leq U_i, i = 1, 2, 3, \dots, n. \end{aligned}$$

其中: $x_i$  为第  $i$  个待优化变量; $L_i$  为  $x_i$  的下边界; $U_i$  为  $x_i$  的上边界。

SCA 求解该优化问题的基本流程为:首先在  $n$  维搜索空间中随机产生  $N$  个搜索个体  $X_1, X_2, X_3, \dots, X_N$ , 每个搜索个体为待优化问题的 1 个候选解,第  $i$  个个体的位置为  $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ , 依据适应度函数计算每个个体的适应度值  $f(X_i)$ , 并将最好适应度值对应的搜索个体记为当前最优个体  $X^*$ , 寻优过程中搜索个体位置更新模型为

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t + r_1 \cdot \sin(r_2) \cdot |r_3 \cdot X_j^* - X_{ij}^t|, r_4 < 0.5, \\ X_{ij}^t + r_1 \cdot \cos(r_2) \cdot |r_3 \cdot X_j^* - X_{ij}^t|, r_4 \geq 0.5, \end{cases} \quad (1)$$

$$r_1 = a \cdot (1 - t/t_{\max}), \quad (2)$$

式中:  $X_{ij}^t$  为第  $t$  代种群的第  $i$  个搜索个体的第  $j$  维位置;  $X_j^*$  为当前最优个体的第  $j$  维位置;  $a$  为大于 1 的常数;  $t$  为当前迭代次数;  $t_{\max}$  为最大迭代次数。  $r_1$  为控制参数, 用于平衡算法的全局搜索与局部搜索, 当  $|r_1 \cdot \sin(r_2)| \geq 1$  时, 算法执行全局搜索, 当  $|r_1 \cdot \sin(r_2)| < 1$  时, 算法执行局部搜索;  $r_2 \in (0, 2\pi)$  为服从均匀分布的随机数, 用于控制位置更新的步长;  $r_3 \in (0, 2)$  为服从均匀分布的随机权重, 用于控制当前最优解对搜索个体位置更新的影响程度;  $r_4 \in (0, 1)$  为服从均匀分布的随机数, 用于选择个体位置更新策略。

## 2 改进的正弦余弦优化算法

### 2.1 自学习策略

通过对 SCA 种群更新模型分析, 算法在整个寻优迭代过程中, 搜索个体的位置更新对于搜索个体本身位置的依赖程度始终不变, 导致算法在寻优前期的全局搜索能力不强; 算法种群在整个迭代过程中主要受当前最优解的指导, 致使迭代后期种群多样性急剧下降, 易于陷入局部最优, 出现早熟收敛现象。鉴于此, 受粒子群算法中惯性权重控制参数启发, 引入非线性调整参数  $\omega$  用于调节种群迭代过程中搜索个体对当前自身位置信息的依赖, 以提高算法的全局搜索能力。迭代前期, 搜索个体的位置更新应低程度地依赖自身位置信息, 以便于搜索更大的空间, 提高算法的全局搜索能力; 迭代后期应高程度地依赖自身位置信息与当前最优解位置信息, 以提高算法的收敛速度, 得到全局最优解。非线性调整参数  $\omega$  更新公式为

$$\omega = (t/t_{\max})^2, \quad (3)$$

此外, 针对种群迭代过程中只依赖于当前种群最优解易于陷入局部最优的缺陷, 对比粒子群算法中“自学习环节”和“社会学习环节”, SCA 种群更新模型中只包含“社会学习环节”, 研究在每次迭代过程中将每个个体搜索到的历史最优解保存下来, 引入“自学习环节”提高种群多样性, 避免陷入局部最优, 改进算法的搜索性能。引入非线性权重因子和“自学习环节”后的搜索个体位置更新公式为

$$X_i^{t+1} = \begin{cases} \omega \cdot X_i^t + r_1 \cdot \sin(r_2) \cdot (|X_i^* - X_i^t| + |X_i^* - X_i^t|), r_4 < 0.5, \\ \omega \cdot X_i^t + r_1 \cdot \cos(r_2) \cdot (|X_i^* - X_i^t| + |X_i^* - X_i^t|), r_4 \geq 0.5, \end{cases} \quad (4)$$

式中  $X_i^*$  为第  $i$  个搜索个体搜索到的历史最优位置。

### 2.2 停滞扰动策略

Lévy 过程是一个连续时间的随机过程, 该理论最早是由法国科学家 Paul Lévy 提出。研究人员通过对 Lévy 过程的研究发现, 自然界中很多动物的活动规律均与 Lévy 过程的特点吻合<sup>[13]</sup>, 反过来, 研究人员通过对多种动物基于 Lévy 过程的觅食活动特点研究, 提出了 Lévy 飞行觅食理论。Lévy 飞行的特点为长时间以较小步长随机游走, 偶尔以较大步长进行方向突变跳跃, 与智能优化算法中的全局搜索和局部搜索特征相似。因此, Lévy 飞行被研究人员广泛应用在各种优化算法中, 用于产生随机步长, 在种群搜索个体更新过程中施加一个扰动量, 丰富种群多样性, 提高算法的搜索能力<sup>[14-15]</sup>。Lévy 飞行的随机游走步长服从一个重尾的概率分布, 称之为 Lévy 分布, 其幂律分布的形式为

$$L(s) \sim |s|^{-1-\beta}, \beta \in (0, 2), \quad (5)$$

式中:  $s$  为随机步长;  $\beta$  为指数参数, 决定 Lévy 分布的形状,  $\beta$  值越大, 生成的随机步长越小。该分布形式直接使用简单的 Matlab 程序语言难以实现。因此, 采用 Mantegna 所提出的生成 Lévy 飞行随机搜索路径  $L(\lambda)$  的方法生成 Lévy 飞行随机步长<sup>[16]</sup>, 其模型为

$$s = u / |v|^{\frac{1}{\beta}}, \quad (6)$$

$$\begin{cases} \sigma_u = \left[ \frac{\Gamma(1+\beta) \cdot \sin(\pi \cdot \beta/2)}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right]^{\frac{1}{\beta}}, \\ \sigma_v = 1, \end{cases} \quad (7)$$

式中:  $s$  为随机游走步长,  $u$  和  $v$  均为服从正态分布的参数, 即  $u \sim N(0, \sigma_u^2)$ ,  $v \sim N(0, \sigma_v^2)$ ,  $\Gamma(x)$  为 gamma 函数。

易于陷入局部最优,出现早熟收敛现象是大多数智能优化算法所面临的问题,SCA 由于其架构特点更是如此。随着寻优过程中迭代次数的增加,当种群所有个体搜索到的历史最优个体的适应度值均值连续 5 次迭代不再变化,则认为搜索陷入停滞,此时采用 Lévy 飞行随机游走策略更新种群搜索个体的位置,提高种群多样性,使算法跳出局部最优。基于 Lévy 飞行的停滞扰动策略模型为

$$X_i^{t+1} = X^* + \text{randn} \cdot \text{Levy}(X_i) + \text{randn} \cdot |X_i^* - X_i^t|, \quad (8)$$

$$\text{Levy}(X_i) = \alpha \cdot s \cdot (X^* - X_i^t), \quad (9)$$

式中:randn 为服从正态分布的随机量; $\alpha \in [-1, 1]$ 为比例因子。

### 2.3 SCASL 寻优步骤

**算法:**自学习策略和 Lévy 飞行的正弦余弦算法

**Step 1:**设置算法的基本参数:种群数目  $N$ ,最大迭代次数  $t_{\max}$ ,问题维度  $D$ ,比例因子  $\alpha$ ,Lévy 飞行指数参数  $\beta$ ;

**Step 2:**随机初始化初始种群搜索个体在搜索空间的位置信息,并进行边界控制;

**Step 3:**计算每个个体的适应度值,更新当前搜索到的最优个体位置信息以及每个搜索个体搜索到的历史最优位置信息;

**Step 4:**判断搜索是否陷入停滞,如果停滞,执行 Step 5;否则,执行 Step 6;

**Step 5:**采用停滞扰动策略更新种群搜索个体的位置,执行 Step 7;

**Step 6:**更新控制参数  $r_1, r_2, r_4, \omega$ ,采用式(4)更新种群搜索个体的位置;

**Step 7:**对新个体位置进行边界控制,对越界值进行随机初始化;

**Step 8:**计算每个个体的适应度值,更新当前搜索到的最优个体位置信息以及每个搜索个体搜索到的历史最优位置信息;

**Step 9:**判断是否满足算法结束条件,不满足,则执行 Step 4;否则,输出得到的最优值,算法寻优结束。

### 2.4 SCASL 收敛性证明

采用文献[17]提出的随机搜索算法收敛准则分析 SCASL 的收敛性。设最小化优化问题为  $\langle f, s \rangle$ ,  $f$  为适应度函数,  $s$  为可行解空间。利用随机搜索算法  $W$  进行寻优,第  $t$  次迭代输出结果为  $X_t^b$ ,第  $t+1$  次迭代输出的结果为  $X_{t+1}^b = W(X_t^b, \xi)$ ,其中  $\xi$  为算法搜索到的位置。

**条件 1:**  $f(W(X_t^b, \xi)) \leq f(X_t^b)$ ,若  $\xi \in s$ ,则有  $f(W(X_t^b, \xi)) \leq f(\xi)$ 。

**条件 2:**若  $\forall D \in s$ ,有  $v(D) > 0$ ,则满足  $\prod_{t=1}^{\infty} (1 - u_t(D)) = 0$ 。其中,  $v(D)$  为  $D$  的勒贝格测度,  $u_t(D)$  为算法  $W$  第  $t$  次迭代后解的概率密度。

**定理 1:**假设  $f$  可测,  $X^*$  为全局最优点集合,可测空间  $s$  为  $R^n$  的可测度子集,算法  $W$  满足条件 1 和条件 2,则有  $\lim_{t \rightarrow \infty} (X_t^b \in X^*) = 1$ ,即算法  $W$  以概率 1 收敛于全局最优。

**定理 2:**SCASL 具有全局收敛性。

**证明:**1)依据 SCASL 基本原理,显然 SCASL 满足条件 1;2)设  $M_t$  为 SCASL 在第  $t$  次迭代过程的候选解集,随着迭代次数的增加,有  $M_t \xrightarrow{t \rightarrow \infty} X_b(t)$ ,其中  $X_b(t)$  为迭代至第  $t$  代搜索到的最优解,  $v(M_t)$  逐渐减小,存在非空解集  $P \in s$ ,使得  $v(M_t \cap P) < v(P)$ ,当 SCASL 搜索陷入停滞,采用停滞扰动策略随机更新种群,有  $P \subseteq M_t$ ,即  $\forall D \in s$ ,有  $0 < u_t(D) \leq 1$ ,因此有  $\prod_{t=1}^{\infty} (1 - u_t(D)) = 0$ ,即 SCASL 满足条件 2,因此 SCASL 以概率 1 收敛于全局最优解。

## 3 仿真结果及分析

### 3.1 经典基准测试函数实验

文献[7]使用的 23 个经典测试函数在智能优化算法领域被广泛用来测试算法性能,具有权威性。为了检验所提出 SCASL 的性能,选取其中 13 个具有代表性的测试函数对 SCASL 进行性能测试,并将结果与基本版本 SCA 和 5 个性能较优的智能优化算法 SSA<sup>[18]</sup>,VCS<sup>[19]</sup>,WOA<sup>[20]</sup>,GSA<sup>[21]</sup>进行对比。选取的测试函

数包括 7 个单峰测试函数和 6 个多峰测试函数,具体描述如表 1。所有仿真实验均基于 Intel(R)Core(TM) i7-4770K CPU@3.50GHz 8GB RAM 计算机平台上的 MATLAB2013a。

表 1 测试所用 13 个经典测试函数

Table 1 Thirteen benchmark functions used in the experiment

函数名	函数编号	问题维度(D)	搜索范围	全局最优解
Sphere	F1	30	$[-100, 100]$	0
Schwefel2.22	F2	30	$[-10, 10]$	0
Schwefel1.2	F3	30	$[-100, 100]$	0
Schwefel2.21	F4	30	$[-100, 100]$	0
Rosenbrock	F5	30	$[-30, 30]$	0
Sterp	F6	30	$[-100, 100]$	0
Quartic	F7	30	$[-1.28, 1.28]$	0
Schwefel2.26	F8	30	$[-500, 500]$	-418.9829D
Rastrigin	F9	30	$[-5.12, 5.12]$	0
Ackley	F10	30	$[-32, 32]$	0
Griewank	F11	30	$[-600, 600]$	0
Penalized	F12	30	$[-50, 50]$	0
Penalized2	F13	30	$[-50, 50]$	0

为了实验的公平性,在实验平台相同的基础上,所有算法最大迭代次数  $t_{\max}$  均设为 500,算法的种群规模  $N$  除 VCS 外均设置为 30,由于 VCS 每次迭代对种群进行 3 次评价,为保证所有算法最大评价次数相同,VCS 种群规模  $N$  设为 10。算法其余参数设置与原文献保持一致,具体如下为:SCASL: $\alpha=0.05, a=2, \beta=0.5$ ; SCA: $r_1=2-2t/t_{\max}$ ; SSA: $c_1=2 \cdot e^{(-4t/t_{\max})^2}$ ; VCS: $\lambda=N/2, \sigma=0.3$ ; WOA: $b=1$ ; GSA: $G_0=100, \alpha=20$ 。

### 3.1.1.1 统计结果分析

使用 SCASL 以及 5 个对比算法求解 13 个经典基准测试函数,表 2 中列出了各算法独立运行 30 次所输出结果的统计量:平均值(Mean)和标准差(SD)。经典测试函数集中  $F1 \sim F7$  为单峰测试函数,这类函数只有一个全局最优解,无局部最优解。因此,这类函数常用来测试启发式搜索算法的收敛速度。对表 2 中求解单峰测试函数结果的统计均值分析知:SCASL 在求解  $F1 \sim F4$  时,均能稳定收敛到全局最优解;求解  $F5$  时虽未收敛到最优值,但收敛精度明显优于对比算法;求解  $F6$  时精度差于 GSA,但优于另外 4 种对比算法;求解  $F7$  时,收敛精度与 VCS 相当,优于另外 4 种对比算法。值得说明的是  $F5$  为一个极复杂的病态测试函数,全局最优值位置在一条极其狭窄的峡谷上,谷中曲面上的最速下降方向与到达全局最优值的方向近似垂直,大部分智能优化算法很难找到全局最优解,而 SLSCA 在求解  $F5$  时明显优于其他对比算法。综上,说明 SCASL 在处理单峰函数问题上的寻优性能优于另外 5 种对比算法。

经典测试函数集中  $F8 \sim F13$  为多峰测试函数,这类函数有多个局部最优解,寻优算法在求解时易于陷入局部最优。因此多峰测试函数常用来表征启发式搜索算法的局部最优规避能力。对表 2 中求解多峰函数结果的均值分析发现:SCASL 在求解  $F8, F13$  时,优于所有对比算法;求解  $F12$  时,仅次于 VCS,优于另外 4

种算法;SCASL 和 VCS 在求解  $F_9, F_{10}$  和  $F_{11}$  时性能相当,其中在  $F_9, F_{11}$  上均收敛到全局最优值,优于其他 4 种对比算法。因此,可以说明 SCASL 具有较强的局部最优规避能力。

表 2 经典测试函数实验结果

Table 2 Experimental results based on classical test functions

函数编号	结果	SCASL	SCA	SSA	VCS	WOA	GSA
$F_1$	Mean	0.00E+00	2.01E+01	1.33E-07	4.62E-269	3.91E-73	4.51E-02
	(SD)	(0.00E+00)	(3.65E+01)	(1.29E-07)	(0.00E+00)	(1.78E-72)	(2.10E-01)
	R	NaN	—	—	—	—	—
$F_2$	Mean	0.00E+00	3.52E-02	2.62E+00	1.08E-136	1.82E-50	2.38E-02
	(SD)	(0.00E+00)	(4.07E-02)	(1.83E+00)	(4.16E-136)	(9.48E-50)	(8.41E-02)
	R	NaN	—	—	—	—	—
$F_3$	Mean	0.00E+00	8.29E+03	1.43E+03	1.50E-257	4.60E+04	9.99E+02
	(SD)	(0.00E+00)	(5.87E+03)	(8.82E+02)	(0.00E+00)	(1.31E+04)	(3.48E+02)
	R	NaN	—	—	—	—	—
$F_4$	Mean	0.00E+00	3.61E+01	1.18E+01	8.36E-134	4.44E+01	7.10E+00
	(SD)	(0.00E+00)	(1.19E+01)	(3.62E+00)	(4.57E-133)	(2.83E+01)	(2.29E+00)
	R	NaN	—	—	—	—	—
$F_5$	Mean	6.39E-02	1.47E+05	2.74E+02	2.39E+01	2.80E+01	4.39E+01
	(SD)	(3.44E-02)	(6.14E+05)	(3.19E+02)	(7.73E-01)	(4.63E-01)	(3.11E+01)
	R	NaN	—	—	—	—	—
$F_6$	Mean	2.49E-04	1.40E+01	2.11E-07	6.75E-06	4.84E-01	2.55E-16
	(SD)	(4.37E-04)	(1.32E+01)	(2.05E-07)	(1.89E-05)	(2.51E-01)	(1.04E-16)
	R	NaN	—	+	+	—	+
$F_7$	Mean	2.66E-04	9.37E-02	1.87E-01	2.66E-04	2.54E-03	7.69E-02
	(SD)	(2.76E-04)	(7.42E-02)	(9.62E-02)	(1.83E-04)	(3.29E-03)	(2.84E-02)
	R	NaN	—	—	=	—	—
$F_8$	Mean	-1.26E+04	-3.78E+03	-7.57E+03	-1.08E+04	-1.05E+04	-2.50E+03
	(SD)	(2.41E+00)	(3.04E+02)	(7.71E+02)	(5.64E+02)	(2.04E+03)	(4.73E+02)
	R	NaN	—	—	—	—	—
$F_9$	Mean	0.00E+00	4.38E+01	6.11E+01	0.00E+00	1.89E-15	2.99E+01
	(SD)	(0.00E+00)	(3.17E+01)	(1.82E+01)	(0.00E+00)	(1.04E-14)	(6.19E+00)
	R	NaN	—	—	=	=	—
$F_{10}$	Mean	8.88E-16	1.50E+01	2.73E+00	8.88E-16	4.20E-15	4.63E-02
	(SD)	(0.00E+00)	(8.59E+00)	(6.61E-01)	(0.00E+00)	(2.79E-15)	(2.54E-01)
	R	NaN	—	—	=	—	—
$F_{11}$	Mean	0.00E+00	9.86E-01	1.73E-02	0.00E+00	1.17E-02	2.69E+01
	(SD)	(0.00E+00)	(4.44E-01)	(1.09E-02)	(0.00E+00)	(3.74E-02)	(6.01E+00)
	R	NaN	—	—	=	=	—

续表

函数编号	结果	SCASL	SCA	SSA	VCS	WOA	GSA
F12	Mean	2.30E-05	2.80E+05	5.33E+00	1.06E-06	2.15E-02	2.06E+00
	(SD)	(4.63E-05)	(1.08E+06)	(1.60E+00)	(5.66E-06)	(1.74E-02)	(9.96E-01)
	R	NaN	-	-	+	-	-
F13	Mean	1.22E-04	2.47E+05	1.88E+01	1.66E+00	5.98E-01	9.99E+00
	(SD)	(4.02E-04)	(8.62E+05)	(1.55E+01)	(1.16E+00)	(2.89E-01)	(8.25E+00)
	R	NaN	-	-	-	-	-
B/E/W(+/=/-)		NaN	0/0/13	1/0/12	2/4/7	0/2/11	1/0/12

图 1 为 6 种算法独立运行 30 次求解 6 个具有代表性的经典基准测试函数所得值的箱式图。由图知, SCASL 在求解 F1, F2, F4, F5, F8, F11 时, 均无异常点出现。并且 SCASL 在求解 6 个测试函数时, 收敛值的分布相相对比算法整体上更加集中, 明显优于其他算法, 说明本文提出的 SCASL 求解问题时的鲁棒性更强。

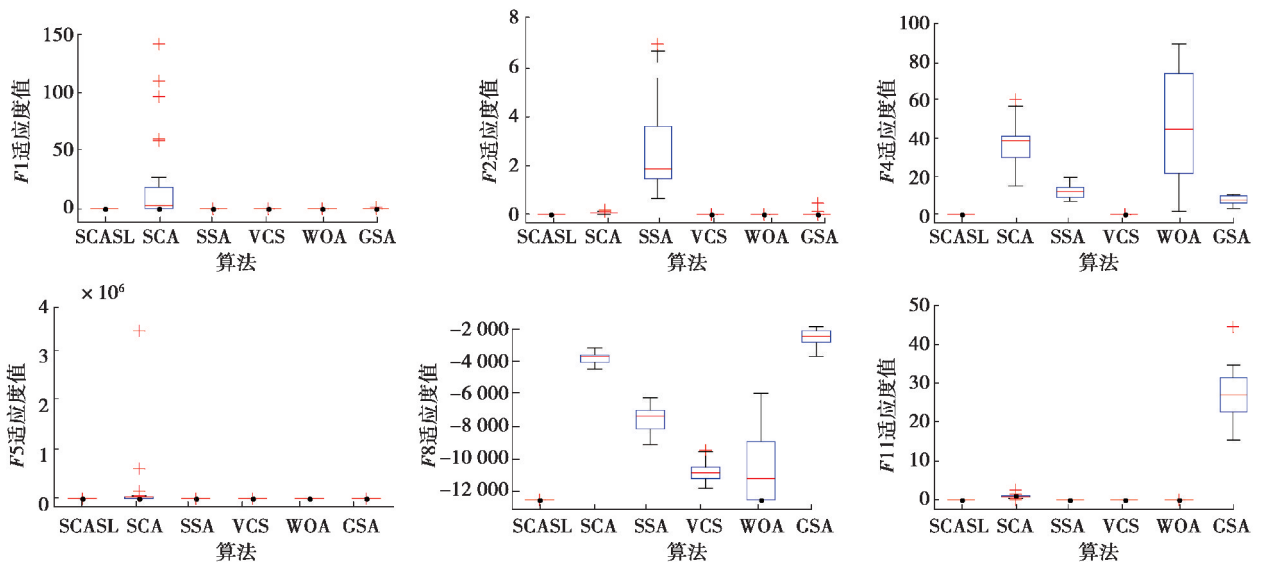


图 1 不同算法在 6 个基准测试函数的收敛箱式图

Fig. 1 Convergence box diagrams of different algorithms on 6 benchmark functions

### 3.1.2 Wilcoxon 统计检验分析

通过对 30 次独立运行结果的平均值和标准差进行分析, 比较算法的性能, 并不能精确分析每次运行的结果。因此, 虽然独立运行 30 次在一定程度上能避免偶然性, 但仍有一定的概率出现偶然优势, 致使算法在均值上具有较好的表现。为了比较每次运行的结果并判断整体结果的显著性差别, 采用 Wilcoxon 统计检验对结果进行分析。将 6 种算法求解 13 个经典基准测试函数独立运行 30 次得到的结果作为样本, 在置信度为 0.05 的条件下进行检验, 判断对比算法所得结果与 SCASL 所得结果的显著性差别。

表 1 中列出 Wilcoxon 统计检验结果(R), 最后一行为检验结果的统计数量(B/E/W)。符号‘+’表示该算法性能优于 SCASL; 符号‘=’表示该算法与 SCASL 性能相当, 无明显差异; 符号‘-’表示该算法性能明显劣于 SCASL。检验结果表明: 在 13 个测试函数中, SCASL 表现出的性能在所有函数上明显优于 SCA, 在 1 个函数上劣于 SSA 和 GSA, 在 2 个函数上劣于 WOA, 在 2 个函数上劣于 VCS, 在 4 个函数上与 VCS 无明显差别。综上, SCASL 整体寻优性能优于 5 种对比算法。

### 3.1.3 收敛性能分析

通过分析可以发现, SCASL 的收敛精度整体上优于另外 5 种对比算法。为进一步验证 SCASL 在寻优过程中的收敛特性, 图 2 列出了算法在求解 6 个具有代表性的测试函数时的收敛曲线。对图 2 分析可得:

SLASCA 在求解单峰测试函数  $F1, F2, F3, F5$ , 多峰函数  $F8$  和  $F13$  时, 收敛速度优于另外 5 种对比算法; 在求解函数  $F5, F13$  时, SCASL 迭代前期收敛速度较快, 迭代后期收敛速度较慢, 主要是由于  $F5$  的最优值很难得到, 而  $F13$  为多峰测试函数, 存在大量局部最优值, 相对比较复杂, SCASL 为了规避局部最优值, 避免早熟收敛, 致使迭代后期收敛速度相对较慢。综上, 整体上 SCASL 收敛速度优于另外 5 种对比算法, 具有相对较好的收敛速度与收敛精度。

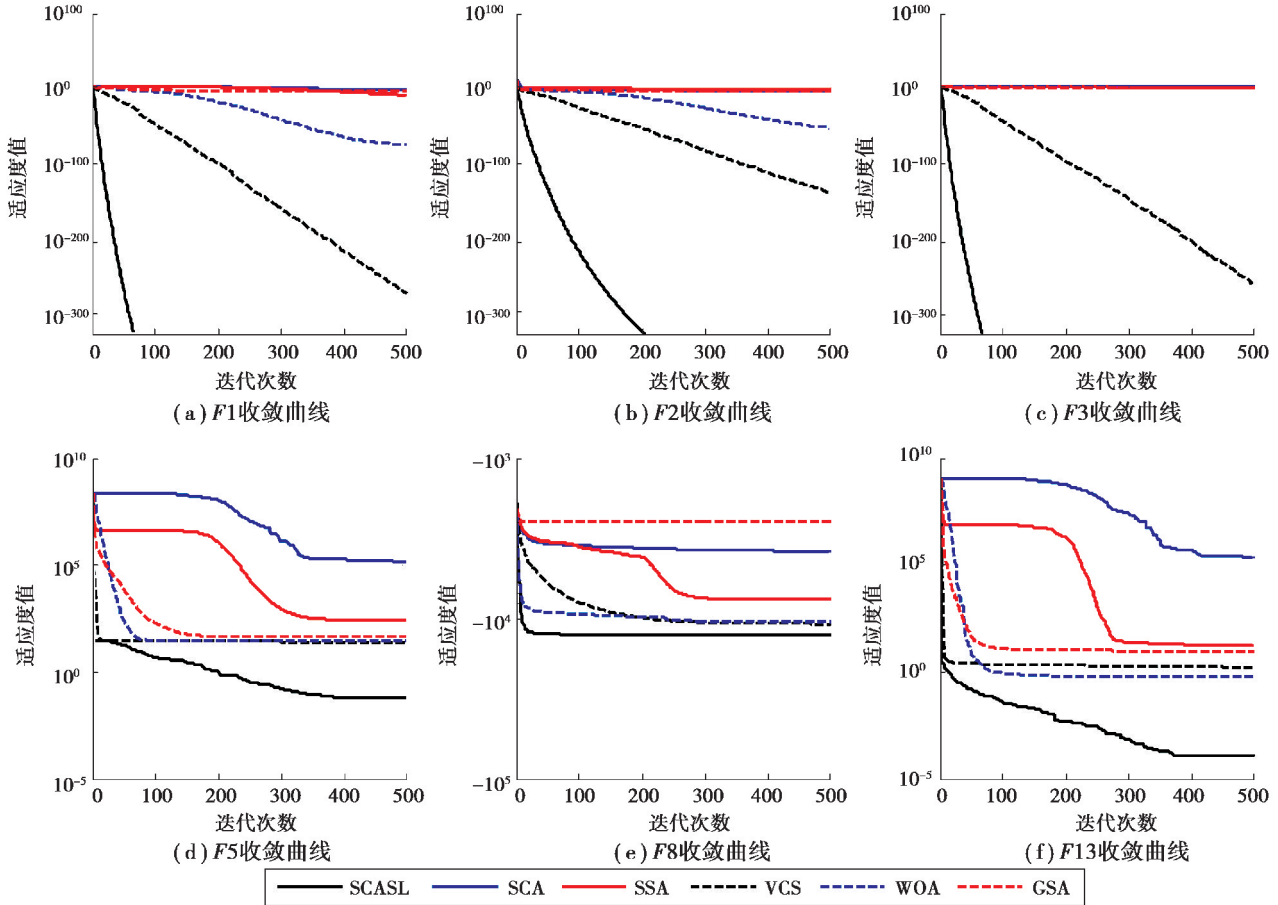


图 2 算法收敛曲线

Fig. 2 Algorithm convergence curves

### 3.1.4 复杂度分析

智能优化算法在对实时性有一定要求的工程问题上也有广泛的应用, 例如使用智能优化算法对无人作战飞机进行实时航迹规划和任务分配等。因此, 基于算法求解测试函数的 CPU 运行时间对算法复杂度进行分析具有必要性。

表 3 列出了 SCASL 和 SCA 求解 13 个测试函数独立运行 30 次的 CPU 平均运行时间。通过表 1 发现, 所提出的 SCASL 虽然增加了策略, 但是运行时间相比 SCA 却有所减少, 说明算法的计算效率要高于 SCA, 即算法的复杂度小于 SCA。算法计算效率提高的主要原因是增加策略改进算法的同时对算法架构与编程方式进行了优化。

### 3.2 航迹规划问题实验

为测试 SCASL 求解带约束条件的工程优化问题的性能, 选择求解无人作战飞机航迹规划问题对算法性能做进一步分析。目前, 智能优化算法在航迹规划问题领域的应用, 一般将规划空间离散化, 寻优得到一系列航迹点, 进而得到一条从起点到终点满足约束条件的可行航迹。文献[22]中航迹规划模型的适应度函数为

$$J = \lambda \cdot J_t + (1 - \lambda) \cdot J_L, \quad (10)$$

式中:  $\lambda \in (0, 1)$  为威胁权重因子;  $J_t$  为威胁代价;  $J_L$  为油耗代价。



假设威胁源均为敌方防空导弹阵地,当无人作战飞机进入其威胁半径时易于被击毁,所以认为威胁区为禁飞区。假设无人机飞行距离与油耗成正比,所以将  $J_L$  等效为航迹长度。简化后的航迹规划模型适应度函数为

$$J = \begin{cases} J_L, & \text{未经过禁飞区;} \\ +\infty, & \text{经过禁飞区,} \end{cases} \quad (11)$$

实验条件:算法种群规模设置为 30,最大迭代次数设置为 100,搜索维度设置为 10,无人作战飞机起点坐标为(15 km,10 km),目标点坐标为(95 km,95 km),威胁源参数如表 4。SCASL 及 SCA 独立运行 5 次得到的结果如表 5,5 次结果中适应度值最小的航迹图和收敛曲线如图 3 和图 4。

表 3 算法运行时间

Table 3 Algorithm running time

$t/s$	SCASL	SCA
F1	8.60E-02	1.11E-01
F2	9.87E-02	1.20E-01
F3	5.32E-01	5.58E-01
F4	1.08E-01	1.33E-01
F5	1.22E-01	1.58E-01
F6	1.13E-01	1.45E-01
F7	1.72E-01	2.07E-01
F8	1.27E-01	1.57E-01
F9	1.27E-01	1.52E-01
F10	1.37E-01	1.66E-01
F11	1.47E-01	1.73E-01
F12	3.13E-01	3.46E-01
F13	3.08E-01	3.44E-01

表 4 威胁源参数

Table 4 Threat source parameters

威胁源	威胁中心/km	威胁半径/km
威胁源 1	(25,35)	10
威胁源 2	(40,70)	10
威胁源 3	(50,30)	8
威胁源 4	(60,55)	8
威胁源 5	(80,75)	7
威胁源 6	(90,40)	8

表 5 独立运行 5 次结果

Table 5 Results of running 5 times independently

适应度值/km	SCASL	SCA
1	125.359 5	168.672 7
2	153.856 2	204.371 4
3	124.492 7	195.930 7
4	126.696 3	170.589 2
5	119.847 6	170.167 8
适应度均值	130.050 5	181.946 4

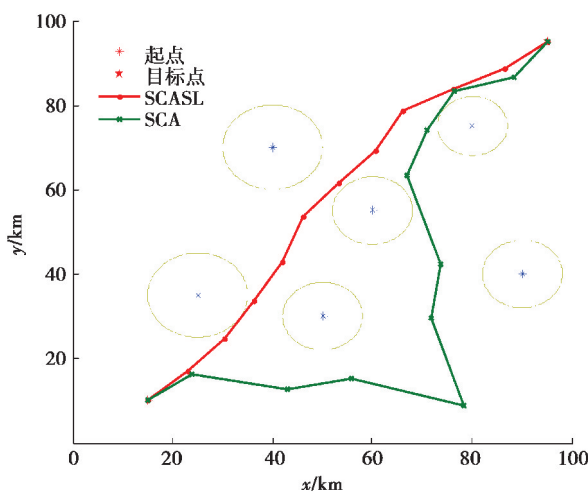


图 3 SCASL 和 SCA 所得航迹图

Fig. 3 Flight paths obtained by SCASL and SCA

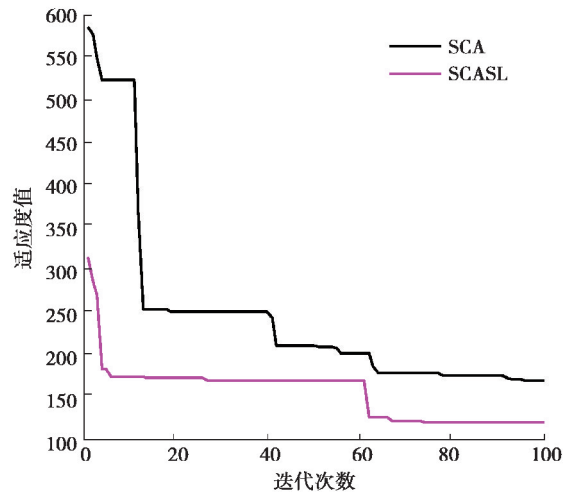


图 4 适应度值收敛曲线

Fig. 4 Fitness value convergence curves

由表 5 知, SCASL 独立运行 5 次的适应度均值为 130.0505, 远小于 SCA 所得到的 181.9464, 并且 SCASL 所得结果均小于 SCA。由图 3 和图 4 易知, SCASL 得到了近似最优航迹, 而 SCA 明显陷入了局部最优。以上结果证明 SCASL 求解带约束条件的工程优化问题时具有更强的局部最优规避能力和更高收敛精度。

## 4 结 论

针对 SCA 易于陷入局部最优和局部搜索能力差的缺陷, 提出了一种自学习策略和 Lévy 飞行的正弦余弦优化算法, 既增强了算法的局部搜索能力提高了收敛速度, 又增强了算法规避局部最优的能力。基于 13 个经典基准测试函数的结果表明, SCASL 的局部最优规避能力和收敛精度整体上优于 SCA, SSA, VCS, WOA 和 GSA; 航迹规划的仿真结果证明 SCASL 求解带约束条件的优化问题时同样具有更强的局部最优规避能力。因此, 所提出的改进策略具有有效性, SCASL 具有较强的寻优能力。

### 参考文献:

- [1] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.
- [2] 张哲. 基于正弦余弦算法的城市土地利用空间的优化配置[J]. 山东农业大学学报(自然科学版), 2016, 47(5): 701-704.  
ZHANG Zhe, Optimal allocation of urban land use space based on sine cosine algorithm[J]. Journal of Shandong Agricultural University(Natural Science Edition), 2016, 47(5): 701-704.(in Chinese)
- [3] 朱静, 何玉珠, 崔唯佳. 正弦余弦算法优化的 SVM 模拟电路故障诊断[J]. 导航与控制, 2018(4): 33-40.  
ZHU Jing, HE Yuzhu, CUI Weijia. Analog circuit fault diagnosis based on SVM optimized by SCA[J]. Navigation and Control, 2018(4): 33-40.(in Chinese)
- [4] Li S, Fang H J, Liu X Y. Parameter optimization of support vector regression based on sine cosine algorithm[J]. Expert Systems with Applications, 2018, 91: 63-77.
- [5] 王蕾, 丁正生. 融合正弦余弦算法和精英算子的花授粉算法[J/OL]. 计算机工程与应用. <http://kns.cnki.net/kcms/detail/11.2127.TP.20190225.1639.024.html>.  
WANG Lei, DING Zhengsheng. Improved flower pollination algorithm combining sine cosine algorithm and elite operator [J/OL]. Computer Engineering and Applications. <http://kns.cnki.net/kcms/detail/11.2127.TP.20190225.1639.024.html>. (in Chinese).

- [6] Chegini S N, Bagheri A, Najafi F. Psoscalf: a new hybrid PSO based on sine cosine algorithm and levy flight for solving optimization problems[J]. *Applied Soft Computing*, 2018, 73: 697-726.
- [7] Nenavath H, Jatoth R K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking[J]. *Applied Soft Computing*, 2018, 62: 1019-1043.
- [8] Issa M, Hassanien A E, Oliva D, et al. ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment[J]. *Expert Systems with Applications*, 2018, 99: 56-70.
- [9] Rizk-Allah R M. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems [J]. *Journal of Computational Design and Engineering*, 2018, 5(2): 249-273.
- [10] Gupta S, Deep K. Improved sine cosine algorithm with crossover scheme for global optimization[J]. *Knowledge-Based Systems*, 2019, 165: 374-406.
- [11] Long W, Wu T B, Liang X M, et al. Solving high-dimensional global optimization problems using an improved sine cosine algorithm[J]. *Expert Systems with Applications*, 2019, 123: 108-126.
- [12] Gupta S, Deep K. A hybrid self-adaptive sine cosine algorithm with opposition based learning[J]. *Expert Systems with Applications*, 2019, 119: 210-230.
- [13] Reynolds A. Liberating Lévy walk research from the shackles of optimal foraging[J]. *Physics of Life Reviews*, 2015, 14: 59-83.
- [14] 景坤雷, 赵小国, 张新雨, 等. 具有 Levy 变异和精英自适应竞争机制的蚁狮优化算法[J]. *智能系统学报*, 2018, 13(2): 236-242.  
JING Kunlei, ZHAO Xiaoguo, ZHANG Xinyu, et al. Ant lion optimizer with levy variation and adaptive elite competition mechanism[J]. *CAAI Transactions on Intelligent Systems*, 2018, 13(2): 236-242.(in Chinese)
- [15] Wang X F, Zhao H, Han T, et al. A gaussian estimation of distribution algorithm with random walk strategies and its application in optimal missile guidance handover for multi-UCAV in over-the-horizon air combat[J]. *IEEE Access*, 2019, 7: 43298-43317.
- [16] Mantegna R N. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes[J]. *Physical Review E*, 1994, 49(5): 4677.
- [17] Solis F J, Wets R J B. Minimization by random search techniques[J]. *Mathematics of Operations Research*, 1981, 6(1): 19-30.
- [18] 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.
- [19] Li M D, Zhao H, Weng X W, et al. A novel nature-inspired algorithm for optimization: virus colony search[J]. *Advances in Engineering Software*, 2016, 92: 65-88.
- [20] Mirjalili S, Lewis A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [21] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm[J]. *Information Sciences*, 2009, 179 (13): 2232-2248.
- [22] Zhang S, Zhou Y Q, Li Z M, et al. Grey wolf optimizer for unmanned combat aerial vehicle path planning[J]. *Advances in Engineering Software*, 2016, 99:121-136.