混合策略改进的正弦余弦算法*

郭园园，袁 杰†，杨炳媛

(新疆大学 电气工程学院，新疆 乌鲁木齐 830017)

摘 要： 针对正弦余弦算法在求解函数优化问题时存在的求解速度慢和收敛精度低等缺陷，提出了一种混合策略改进的正弦余弦算法. 首先，采用余弦拟合度策略，通过旋转变换算子与轴向变换算子来更新种群位置信息，进而能够有效保持种群的多样性和提高求解精度. 其次，利用单纯形法对种群中适应度较差的个体更新位置信息，增强算法的勘探能力和开采能力. 最后，采用自适应参数调整策略，来均衡全局的搜索性能和局部的开采性能. 增加动态惯性权重策略，使寻优过程充分利用个体的位置信息，提高局部的开采性能和加快算法的收敛速度. 采用8个基准测试函数来评估改进算法的性能，并与其它改进算法用于压力容器设计中. 实验结果表明：改进算法在求解速度和收敛精度方面均具有实质性的优势.

关键词： 正弦余弦算法；余弦拟合度；动态惯性权重；单纯形法

DOI： 10.13568/j.cnki.651094.651316. 2022.04.27.0001

中图分类号： TP301.6 **文献标识码：** A **文章编号：** 2096-7675(2023)01-0114-08

引文格式： 郭园园, 袁杰, 杨炳媛. 混合策略改进的正弦余弦算法[J]. 新疆大学学报(自然科学版)(中英文), 2023, 40(1): 114-121.

英文引文格式： GUO Yuanyuan, YUAN Jie, YANG Bingyuan. Improved sine cosine algorithm with hybrid strategy[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English), 2023, 40(1): 114-121.

Improved Sine Cosine Algorithm with Hybrid Strategy

GUO Yuanyuan, YUAN Jie, YANG Bingyuan

(*School of Electrical Engineering, Xinjiang University, Urumqi Xinjiang 830017, China*)

Abstract： Aiming at the slow solution speed and low convergence accuracy of the sine-cosine algorithm in solving function optimization problems, a hybrid strategy-improved sine-cosine algorithm is proposed. Firstly, the cosine fit strategy is adopted to update the population position information through the rotation transformation operator and the axial transformation operator, which can effectively maintain the diversity of the population and improve the accuracy of the solution. Secondly, the simplex method is used to update the position information of the individuals with poor fitness in the population to enhance the exploration and mining capabilities of the algorithm. Finally, an adaptive parameter adjustment strategy is adopted to balance the global search performance and the local mining performance. The dynamic inertia weight strategy is taken, and the individual location information is used in the optimization process, improving local mining performance and speeding up the convergence speed of the algorithm. Eight benchmark test functions are used to evaluate the performance of the improved algorithm, and used with other improved algorithms in the design of pressure vessels. Experimental results show that the improved algorithm has substantial advantages in aspect of solution speed and convergence accuracy.

Key words： sine cosine algorithm; cosine fit; dynamic inertia weight; simplex method

0 引 言

正弦余弦算法（SCA）[1]是Mirjalili通过观察正弦函数和余弦函数的数学模型，于2016年提出的一种元启发式优化算法. 常见的元启发式算法包括海鸥算法[2]、乌鸦搜索算法[3]以及较新的斑点鬣狗优化算法[4]和被囊群算法[5]，都具有算法原理和结构较为简单、程序编写更易实现并且需要控制的参数数目较少等优点. 文献[1]研

* **收稿日期：** 2022-04-27

基金项目： 国家自然科学基金“非结构环境下机器人羽流寻源自主演进策略研究”（62263031），“机器人化单分子病毒可控侵染细胞及原位定量表征方法研究”（62073227）；新疆维吾尔自治区自然科学基金“非结构环境下机器人建图与主动安全方法研究”（2022D01C53）.

作者简介： 郭园园（1996-），男，硕士生，从事机器人路径规划算法的研究，E-mail: 1097396076@qq.com.

† **通讯作者：** 袁杰（1975-），男，博士，教授，主要从事计算机应用的研究，E-mail: yuanjie222@163.com.

究表明，在求解函数优化问题时，相比粒子群算法、蝙蝠算法和萤火虫算法等部分经典优化算法，正弦余弦算法在求解精度和收敛速度方面具有更好的性能. 但与其它元启发式优化算法相类似，在迭代后期依然存在求解精度和收敛速度表现不佳的问题.

为了解决这些问题，近年来相关领域的学者做了许多研究工作，提出了多种有效的改进思路，包括以下三类：参数改进、策略改进、算法融合. 正弦余弦算法的控制参数r_1用来均衡全局搜索和局部搜索，因此如何调整这个参数成为算法改进的一个重要方向. 同时寻优策略的优化和融入其它优化算法的算子也是目前改善算法性能的重要手段. 文献[6]引入正切函数来优化控制参数，在全局搜索与局部搜索之间实现了动态平衡. 文献[7]引入随机差分变异策略以扩大在种群空间内的搜索范围，提高种群的遍历性和多样性. 文献[8]将人工蜂群算法的采蜜蜂和侦察蜂作为优化算子融入到标准正弦余弦算法中，加快算法的收敛速度和提高求解精度. 文献[9]提出了一种基于拉普拉斯和高斯分布的变异策略，增强全局搜索性能帮助算法跳出局部最优. 文献[10]引入Lévy飞行策略扩大对最优解位置的搜索领域，防止算法过早收敛. 文献[11]将精英个体的混沌搜索策略融入到算法中提高了局部开发能力，引入的反向学习策略能够增强算法的全局探索能力. 文献[12]引入了精英反向学习策略提高解的质量，并利用个体的反思学习能力避免算法过早收敛.

正弦余弦算法收敛速度慢和跳出局部极值能力弱导致了算法在有限时间内的收敛精度相比一些算法仍然较差，针对上述问题，本文提出了一种混合策略改进的正弦余弦算法（Improved Sine Cosine Algorithm with Hybrid Strategy, ISCA）. 采用余弦拟合度策略增强个体的分布质量，提高求解精度；控制参数r_1随着迭代过程的动态调整在全局搜索与局部搜索之间实现了动态平衡；动态惯性权重策略使种群的个体信息充分发挥，提高局部搜索能力和加快算法收敛速度；单纯形法更新适应度较差的个体信息，进一步提高算法的勘探能力和开发能力. 最后通过测试函数来求解改进算法的各项评价指标并应用于压力容器设计中以检验改进策略的有效性.

1　正弦余弦算法

求解优化问题时，搜寻的空间维度是R，整个种群的规模为N，在进行到第t次迭代时，个体h的位置信息为$p^{h,t}=\left[p_1^{h,t},p_2^{h,t},\cdots,p_R^{h,t}\right]$，其中：$h=1,2,\cdots,N$，$t=1,2,\cdots,Max\_iter$，$Max_iter$为迭代次数的最大上限. 在最开始的迭代中，种群没有先验知识，就把随机生成的位置作为种群的初始位置，再根据适应度函数求解每个个体的目标函数值，最后将当前迭代中适应度值最小的个体信息保留下来，每一代中个体的位置更新公式如式(1) 所示：

$$p^{h,t+1}=\begin{cases} p^{h,t}+r_1\sin(r_2)\left|r_3p^{g,t}-p^{h,t}\right|, & r_4<0.5 \\ p^{h,t}+r_1\cos(r_2)\left|r_3p^{g,t}-p^{h,t}\right|, & r_4\geq 0.5 \end{cases} \tag{1}$$

$$r_1=a-at/T \tag{2}$$

其中：$p^{g,t}$是第t次迭代时最优个体的位置信息；r_1是正弦函数和余弦函数的振幅因子，其表达式如式(2)所示，a是一个常数，T为最大迭代次数；r_2是处在0到2π之间的随机数；r_3是处在0到2之间的随机数；r_4是处在0到1之间的随机数.

2　混合策略改进的正弦余弦算法

2.1　余弦拟合度策略

在算法迭代搜索末期，其它个体会聚集在当前最优个体的周围，甚至移动轨迹都会出现交叠的情况，降低了解的分布质量，不利于维系种群多样性. 为了改善这种情况，引入了余弦拟合度来评估最优个体和附近个体的位置分布情形，具体的数学描述如式 (3) 所示：

首先构造α向量和β向量：

$$\begin{cases} \alpha=p^{h,t}-p^{g,t} \\ \beta=p^{k,t}-p^{g,t} \end{cases} \tag{3}$$

其中：$p^{h,t}$、$p^{k,t}$和$p^{g,t}$分别反映的是第t次迭代时个体h、除当前个体h之外的随机个体k和最优个体g在解空间内的位置分布情况.

定义$\cos(\alpha,\beta)$为两个向量之间的余弦拟合度，在-1到1之间取值，拟合度的数学描述如式（4）所示：

$$\cos(\alpha,\beta)=\frac{\alpha\cdot\beta}{|\alpha||\beta|} \tag{4}$$

其中：分子部分为两个向量的点积；分母部分为两个向量的模的乘积. 当拟合度为1时，表明两个向量共线，有交叠；当拟合度为-1时，表明两个向量虽然共线，但方向相反. 拟合度越大，表明个体h和个体k在方向上越趋近于交叠. 设定拟合阈值C_value，依次计算当前个体h和其它随机个体k的余弦拟合度，将拟合度大于拟合阈值的个体进行过滤，引入文献[13]中的旋转变换和轴向变换，通过比较当前个体与过滤后个体的适应度值，将适应度较差的个体经过旋转变换和轴向变换后更新位置信息，数学描述如式（5）所示：

$$\begin{cases} p^{k,t}=p^{g,t}+\frac{\lambda R_1 p^{g,t}}{n\left\|p^{k,t}\right\|_2},\ f(p^{k,t})\geq f(p^{h,t}) \\ p^{h,t}=p^{h,t}+\gamma R_2 p^{h,t},\ f(p^{k,t})<f(p^{h,t}) \end{cases} \tag{5}$$

其中：$f(p^{h,t})$和$f(p^{k,t})$分别是个体h和个体k的适应度值；λ是旋转系数；n为空间维度；R_1是服从-1到1之间均匀分布的随机矩阵；γ是轴向系数；R_2是一个稀疏随机对角矩阵，其内部元素值服从高斯分布.

2.2 单纯形法

单纯形法是一种用于求解多维度无约束条件的函数优化问题的搜索方法，核心内容包括：一是梯度评估，它是从最差顶点到其它所有顶点的方向向量m；二是对最差顶点采用镜像、延伸、外压缩和内压缩操作搜寻最优顶点，每一种操作都会生成更好的顶点来代替最差顶点. 单纯形法的搜索过程如图1所示.

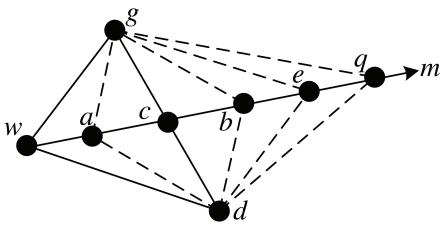

图 1 单纯形法

图1中，g是最优点，d是次优点，w是最差点，a是内压缩点，c是中心点，b是外压缩点，e是镜像点，q是延伸点，具体操作如下：

1）计算中心点位置，记作p^c.

$$p^c=\frac{p^g+p^d}{2} \tag{6}$$

2）执行镜像操作.

$$p^e=p^c+\eta(p^c-p^w) \tag{7}$$

其中：p^e是镜像点位置；η是镜像因子.

3）若$f(p^e)<f(p^g)$，执行延伸操作.

$$p^q=p^c+\tau(p^e-p^c) \tag{8}$$

其中：p^q是延伸点位置；τ是延伸因子. 若$f(p^q)<f(p^g)$，就用p^q代替p^w，否则用p^e代替p^w.

4）若$f(p^e)>f(p^w)$，执行外压缩操作.

$$p^b=p^c+\kappa(p^w-p^c) \tag{9}$$

其中：p^b是外压缩点位置；κ是外压缩因子. 若$f(p^b)<f(p^w)$，就用p^b代替p^w.

5）若$f(p^w)>f(p^e)>f(p^g)$，执行内压缩操作.

$$p^a=p^c-\zeta(p^w-p^c) \tag{10}$$

其中：p^a是内压缩点位置；ζ是内压缩因子. 若$f(p^a)<f(p^w)$，就用p^a代替p^w.

由单纯形法的4种操作可知，镜像操作可以使最差点找到解空间内的所有有效解；延伸操作可以使最优点规避局部最优，向远离最差点的方向勘探；内压缩和外压缩操作可以使最差点搜索到更优的位置，提高算法的局部开采性能.

2.3　控制参数自适应调整策略

由正弦余弦算法基本原理可知，振幅因子r_1关乎着算法的局部与全局搜索性能. 由式（1）可知，振幅因子小于1时，下一次迭代的位置处在当前个体h的位置$p^{h,t}$和当前最优个体g的位置$p^{g,t}$之间，体现算法的局部开采性能；振幅因子大于1时，下一次迭代的位置处在当前个体h的位置$p^{h,t}$和当前最优个体g的位置$p^{g,t}$之外，体现算法的全局勘探性能.

但在正弦余弦算法中振幅因子被设定为线性递减，并不能充分体现算法的性能. 因此在迭代早期，参数值应维持较大的初值，具有较好的种群多样性与全局搜索性能，伴随迭代进程慢慢减小，使迭代后期表现出较强的局部搜索性能，直到趋近于预设最小值. 结合上述分析，提出一种新的控制参数调整策略，数学描述如式（11）所示：

$$r_1 = \frac{r_{\max} - r_{\min}}{1 + e^{20t/Max\_iter - 10}} + \varepsilon gamrnd(x,y) \tag{11}$$

其中：$r_{\max}$和$r_{\min}$分别是振幅因子的最大值和最小值；Max_iter是最大迭代次数；ε是修正因子；$gamrnd(x,y)$是产生服从伽马分布参数为x,y的随机数.

2.4　动态惯性权重策略

惯性权重反映的是先前个体位置多大程度影响更新位置，如果按照式（1）更新位置，个体在迭代后期很容易发生“震摆”. 为了进一步均衡全局勘探和局部开采的性能，提出了动态惯性权重策略，数学描述如式（12）所示：

$$\omega = \mu\left(1 - \frac{\log(|f(p^{g,t})|) + |f(p^{g,t})|}{\log(|f(p^{h,t})|) + |f(p^{h,t})|}\right) + Q \tag{12}$$

$$Q = 1 - \frac{1}{1 + e^{-(\pi e\sqrt{\pi}t - \pi\sqrt{2e}Max\_iter)/Max\_iter}} \tag{13}$$

其中：μ是影响因子；$f(p^{g,t})$是第t次迭代最优个体g的适应度值；Q是改进的sigmoid函数；加入动态惯性权重策略后的位置更新公式如式（14）所示：

$$p^{h,t+1} = \begin{cases} \omega p^{h,t} + r_1 \sin(r_2)\,|r_3 p^{g,t} - p^{h,t}|, & r_4 < 0.5 \\ \omega p^{h,t} + r_1 \cos(r_2)\,|r_3 p^{g,t} - p^{h,t}|, & r_4 \geq 0.5 \end{cases} \tag{14}$$

当前个体h的适应度值逼近最优个体g的适应度值时，动态惯性权重ω取最小值，小的惯性权重有助于算法进行局部开采；当前个体h的适应度值偏离最优个体g的适应度值时，动态惯性权重ω取最大值，大的惯性权重能够更多继承上代个体的位置信息，有助于算法进行全局勘探.

2.5　混合策略改进的正弦余弦算法流程

在标准正弦余弦算法的基础上，提出了混合策略改进的正弦余弦算法，其流程如算法1所示.

算法 1 混合策略改进的正弦余弦算法

输入 种群规模N；最大迭代次数Max_iter；旋转系数λ；轴向系数γ；镜像因子η；延伸因子τ；外压缩因子κ；内压缩因子ζ；振幅因子最大值（最小值）$r_{\max}(r_{\min})$；修正因子ε；影响因子μ.

输出 所有个体中的最好位置$P*$.

步骤1. 初始化. 初始化ISCA算法的相关参数并随机生成种群的初始位置.

步骤2. 评估目标函数. 计算每个个体对应的适应度函数值.

步骤3. 将余弦拟合度大于阈值的个体按式（5）更新位置信息.

步骤4. 采用单纯形法更新最差个体的位置信息.

步骤5. 按式（12）更新惯性权重，按式（11）更新振幅因子，按式（14）更新个体的位置信息.

步骤6. 计算每个个体的适应度值，更新搜索到的最优个体的位置信息.

步骤7. 迭代结束判断. 当迭代过程仍在继续时，重复步骤3~6直至完成所有迭代. 当达到迭代次数的最大上限时，输出与最优适应度函数值相对应的位置信息.

3 算法性能测试及分析

3.1 算法性能测试函数

选取了经常使用到的8个标准测试函数对算法性能进行验证，这些函数来自文献[14]. 其中：$f_1(x)\sim f_3(x)$是单峰测试函数，在自变量的整个定义范围内有且仅有一个全局最小值，可以用来测试算法搜寻最优解期间的速度；$f_4(x)\sim f_8(x)$是多峰函数，在定义上下限范围内有多个局部极小值和一个全局最小值，可以用来测试算法全局搜索和避免过早收敛的能力. 测试函数的详细信息见表1.

表 1 测试函数

函数表达式	定义域	维度	最小值
$f_1(x)=\sum_{i=1}^{D}x_i^2$	[-100,100]	D	0
$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[-10,10]	D	0
$f_3(x)=\sum_{i=1}^{D}ix_i^4+rand[0,1)$	[-1.28,1.28]	D	0
$f_4(x)=-\sum_{i=1}^{D}\left[x_i\sin(\sqrt{\lvert x_i\rvert})\right]$	[-500,500]	D	-418.982 9D
$f_5(x)=\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[-5.12,5.12]	D	0
$f_6(x)=-20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}+20+e-e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	[-32,32]	D	0
$f_7(x)=\frac{1}{4\,000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	D	0
$f_8(x)=\frac{\pi}{D}\{10\sin^2(\pi y_1)+(y_D-1)^2+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]\}+\sum_{i=1}^{D}\mu(x_i,10,100,4)$	[-50,50]	D	0

3.2 性能测试结果及分析

将所提出的改进ISCA与WOA、TLBO[15]、GSA[16]、MVO[17]和SCA在8个测试函数上进行对比，为了避免实验的偶然性，种群的大小均设置为30，迭代次数最大上限均设置为1 000，空间维度均设置成30，将上述6种算法分别在测试函数$f_1(x)\sim f_8(x)$上独立运行30次. 为检验算法的综合能力，评价指标包括最优值、最差值、平均值和标准差. ISCA中设置参数λ=1；γ=1；η=1；τ=2；κ=0.5；ζ=0.5；$r_{\max}$=2；$r_{\min}$=0.01； ε=0.1；μ=1. 实验硬件为Inter Core i7-4800H、16 GB内存、2.9 GHz的计算机，软件为Matlab R2016a，实验结果如表2所示.

表 2 WOA、TLBO、GSA、MVO、SCA、ISCA函数测试结果对比

函数	评价指标	WOA	TLBO	GSA	MVO	SCA	ISCA
$f_1(x)$	最优值	6.88×10^{-168}	2.04×10^{-173}	6.42×10^{-17}	1.42×10^{-1}	2.00×10^{-5}	0
	最差值	8.06×10^{-147}	3.24×10^{-169}	2.24×10^{-16}	4.62×10^{-1}	5.13×10^{-1}	0
	平均值	3.57×10^{-148}	2.47×10^{-170}	1.15×10^{-16}	3.04×10^{-1}	3.96×10^{-2}	0
	标准差	1.50×10^{-147}	0	4.23×10^{-17}	7.49×10^{-2}	1.03×10^{-1}	0
$f_2(x)$	最优值	1.78×10^{-105}	1.50×10^{-87}	3.78×10^{-8}	2.78×10^{-1}	3.80×10^{-8}	0
	最差值	2.79×10^{-91}	5.52×10^{-85}	8.84×10^{-8}	1.02×10^{0}	4.47×10^{-4}	0
	平均值	1.10×10^{-92}	1.12×10^{-85}	5.43×10^{-8}	4.33×10^{-1}	3.02×10^{-5}	0
	标准差	5.14×10^{-92}	1.48×10^{-85}	1.32×10^{-8}	1.70×10^{-1}	8.30×10^{-5}	0
$f_3(x)$	最优值	1.02×10^{-4}	1.05×10^{-4}	2.84×10^{-2}	9.26×10^{-3}	2.94×10^{-3}	2.48×10^{-6}
	最差值	7.58×10^{-3}	1.16×10^{-3}	1.11×10^{-1}	3.77×10^{-2}	1.70×10^{-1}	3.24×10^{-4}
	平均值	1.56×10^{-3}	5.49×10^{-4}	5.61×10^{-2}	2.10×10^{-2}	3.44×10^{-2}	7.18×10^{-5}
	标准差	1.79×10^{-3}	2.48×10^{-4}	1.96×10^{-2}	6.90×10^{-3}	3.85×10^{-2}	7.49×10^{-5}
$f_4(x)$	最优值	-9.64×10^{3}	-4.59×10^{3}	-3.87×10^{3}	-9.38×10^{3}	-4.70×10^{3}	-1.26×10^{4}
	最差值	-6.72×10^{3}	-3.31×10^{3}	-2.00×10^{3}	-6.08×10^{3}	-3.44×10^{3}	-7.92×10^{3}
	平均值	-7.95×10^{3}	-3.77×10^{3}	-2.70×10^{3}	-7.95×10^{3}	-3.87×10^{3}	-1.13×10^{4}
	标准差	7.60×10^{2}	3.03×10^{2}	4.99×10^{2}	7.36×10^{2}	3.02×10^{2}	1.57×10^{3}

续表 2

函数	评价指标	WOA	TLBO	GSA	MVO	SCA	ISCA
$f_5(x)$	最优值	0	0	1.39×10^{1}	6.29×10^{1}	3.39×10^{-6}	0
	最差值	5.68×10^{-14}	2.79×10^{1}	4.38×10^{1}	1.63×10^{2}	6.19×10^{1}	0
	平均值	1.89×10^{-15}	1.24×10^{1}	2.60×10^{1}	1.11×10^{2}	1.26×10^{1}	0
	标准差	1.04×10^{-14}	7.45×10^{0}	6.49×10^{0}	2.94×10^{1}	1.84×10^{1}	0
$f_6(x)$	最优值	8.88×10^{-16}	4.44×10^{-15}	4.66×10^{-9}	1.63×10^{-1}	7.63×10^{-5}	8.88×10^{-16}
	最差值	7.99×10^{-15}	4.44×10^{-15}	1.07×10^{-8}	2.16×10^{0}	2.03×10^{1}	8.88×10^{-16}
	平均值	3.26×10^{-15}	4.44×10^{-15}	7.47×10^{-9}	1.22×10^{0}	1.08×10^{1}	8.88×10^{-16}
	标准差	2.35×10^{-15}	1.60×10^{-30}	1.23×10^{-9}	5.45×10^{-1}	9.76×10^{0}	0
$f_7(x)$	最优值	0	0	3.20×10^{0}	4.24×10^{-1}	7.39×10^{-6}	0
	最差值	0	6.88×10^{-7}	1.57×10^{1}	7.83×10^{-1}	7.95×10^{-1}	0
	平均值	0	2.29×10^{-8}	8.39×10^{0}	5.80×10^{-1}	1.94×10^{-1}	0
	标准差	0	1.26×10^{-7}	2.92×10^{0}	1.09×10^{-1}	2.37×10^{-1}	0
$f_8(x)$	最优值	1.39×10^{-3}	4.40×10^{-1}	3.85×10^{-19}	1.08×10^{-1}	4.49×10^{-1}	9.35×10^{-19}
	最差值	1.97×10^{-2}	8.21×10^{-1}	1.31×10^{0}	4.47×10^{0}	8.72×10^{2}	3.34×10^{-14}
	平均值	6.62×10^{-3}	6.10×10^{-1}	1.19×10^{-1}	1.60×10^{0}	4.10×10^{1}	2.45×10^{-15}
	标准差	5.42×10^{-3}	1.04×10^{-1}	2.62×10^{-1}	1.22×10^{0}	1.64×10^{2}	6.55×10^{-15}

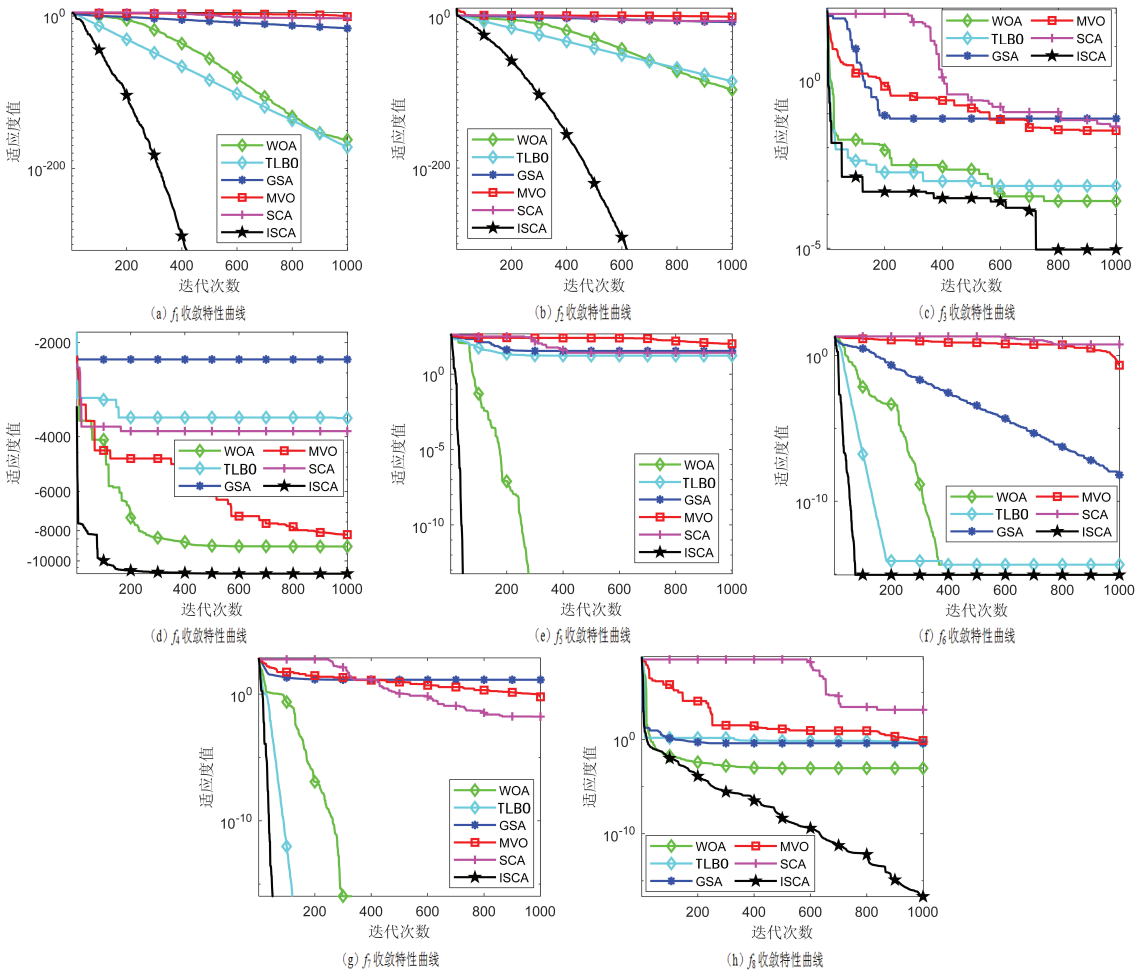

图 2　六种算法迭代寻优对比曲线

从表2中可以看出：SCA在所有函数上的平均值均差于WOA和TLBO，在部分函数上的平均值优于GSA和MVO，而ISCA在8个测试函数上的平均值均优于其它5种算法，说明改进算法在30次独立实验中能够取得一

个相对满意的求解水准并具有一定的优势. 其中：在函数$f_5(x)$和$f_6(x)$上WOA虽然和ISCA取得了一样的最优值，但是在平均值和标准差上还是差于ISCA. 在函数$f_1(x)$上，虽然TLBO取得了和ISCA一样的标准差，但在最优值和最差值指标上，ISCA均收敛到了全局最优. 在函数$f_8(x)$上，虽然ISCA的最优值差于GSA，但在最差值和平均值上均优于GSA十几个数量级，这表明了ISCA即使处于最不好的寻优条件也仍然能够找到较优的收敛精度，有助于求解未知的工程优化问题. 从标准差这个指标来看，ISCA在函数$f_1(x)$、$f_2(x)$、$f_5(x)\sim f_7(x)$上的标准差直接降为0，这表明了ISCA具有较好的寻优稳定性和鲁棒性，而且其最优值指标和最差值指标之间的极小数值差异更加印证了这一分析. ISCA在3个单峰函数和5个多峰函数的求解过程中都可以体现出其能够搜寻到更高精度的最优解，证明了ISCA具有较强的局部开采性能和较优的求解精度，尤其是对5组多峰函数仍维系着较优的目标函数值，验证了ISCA具有较优的全局搜索性能和较强的局部极值逃逸能力.

为了更加明了地查看各种算法的寻优情况，图2给出了ISCA和其它5种对比算法在30维下求解8个测试函数的收敛特性曲线. 从图2（a~h）可以明显看出，本文提出的ISCA在求解每一个测试函数迭代早期的求解质量就明显优于WOA、TLBO、GSA、MVO和SCA，而且ISCA曲线的下降趋势并没有随着迭代次数的增加就停滞下来，而是具有持续求解的能力. 图2（d~h）为多峰函数的收敛曲线，可以看出ISCA具有良好的求解速度与收敛精度，并且在迭代过程中除$f_3(x)$外并没有出现明显的拐点，由此说明ISCA具有较强的求解能力，可以快速逃逸局部最优约束向全局最优解靠近.

3.3 与其它改进的正弦余弦算法在工程优化方面的对比分析

在函数优化的基础上，探讨ISCA在工程优化方面的适用性，并选取文献[11]中的COSCA、文献[18]中的SCASL和文献[19]中的ASCA作为对比算法，选择文献[20]中压力容器的设计作为将要进行求解的工程优化问题，采用上述4种算法分别进行求解，得出实验结果，从而比较4种算法的性能，进一步评估提出的ISCA的有效性和可行性.

设计压力容器时，尽量降低对所需原材料的需求从而可以更大幅度减少生产成本，压力容器设计如图3所示，上下两侧都有封盖，头部的形状是半球形，L是去掉半球形的头部后剩余部分的高度，R是圆柱体部分的内侧半径，T_s和T_h分别表示圆柱体部分和头部的外侧半径和内侧半径的差值，其数学模型描述如下：

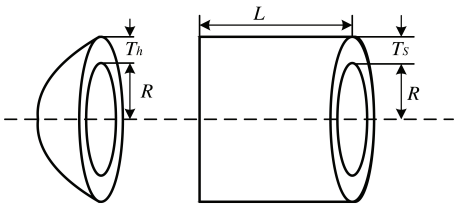

图 3 压力容器设计问题图解

$x=[x_1,x_2,x_3,x_4]=[T_s\ T_h\ R\ L]$；$\min f(x)=0.622\ 4x_1x_3x_4+1.778\ 1x_2x_3^2+3.166\ 1x_1^2x_4+19.84x_1^2x_3$；s.t. $g_1(x)=-x_1+0.019\ 3x_3\le 0$；$g_2(x)=-x_2+0.009\ 54x_3\le 0$；$g_3(x)=-\pi x_3^2-4/3\pi x_3^3+1\ 296\ 000\le 0$；$g_4(x)=x_4-240\le 0$；$0\le x_i\le 100, i=1,2$；$10\le x_i\le 200, i=3,4$.

COSCA、SCASL和ASCA的实验参数均按照文献中的最优值进行设置，为了让实验更公平，同时将4种改进算法都含有的调节参数设置成一样，即：种群大小均设置成N=30，迭代次数的上限均设置成Max_iter=1 000，每组实验均独立运行30次，4种改进算法对该压力容器设计问题的求解结果如表3所示，收敛曲线如图4所示.

表 3 实验结果对比

指标	ASCA	SCASL	COSCA	ISCA
最优值	8.50×10^3	8.66×10^3	1.01×10^4	8.14×10^3
最差值	1.36×10^4	1.28×10^4	1.66×10^4	1.00×10^4
平均值	1.03×10^4	1.01×10^4	1.26×10^4	8.61×10^3
标准差	1.08×10^3	9.73×10^2	1.77×10^3	3.63×10^2

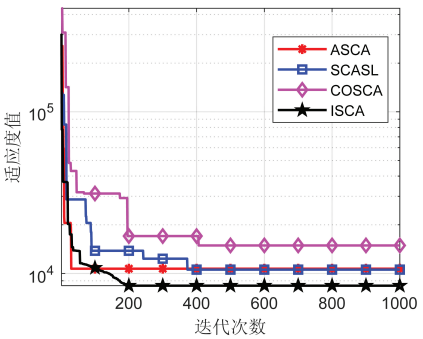

图 4 求解压力容器设计问题对比曲线

由表3可知，ISCA的各项指标均优于ASCA、SCASL和COSCA，平均值越小表明ISCA搜寻到的最优值的精度越高；标准差越小表明ISCA具有更好的寻优稳定性和鲁棒性，因此本文提出的混合策略改进的正弦余弦算法在求解工程优化问题方面是可行的和有效的.

4　结 论

为了改善正弦余弦算法存在的搜寻速度慢和寻优精度低等问题，本文提出了一种混合策略改进的正弦余弦算法. 采用余弦拟合度策略增强了个体的分布质量，提高了算法搜寻到的最优解的精度. 单纯形法通过更新适应度较差的个体信息，进一步提高了算法的勘探能力和开发能力. 通过对振幅因子的自适应调整均衡了全局勘探性能与局部开采性能. 动态惯性权重策略使种群的个体信息得到充分发挥，提高了局部搜索能力、加快了算法收敛速度. 通过测试函数和压力容器设计问题均检验了改进策略是有效的. 在未来的计划中，如何利用ISCA求解组合优化问题以及如何将算法扩展到其它的应用领域是接下来的研究重点.

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] DHIMAN G, KUMAR V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems[J]. Knowledge-Based Systems, 2018, 165: 169-196.

[3] ASKARZADEH A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm[J]. Computers and Structures, 2016, 169: 1-12.

[4] 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.

[5] KAUR S, AWASTHI L K, SANGAL A L, et al. Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization[J]. Engineering Applications of Artificial Intelligence, 2020, 90: 103541.

[6] 林杰, 何庆. 混合策略改进正弦余弦算法[J]. 计算机应用研究, 2020, 37(12): 3612-3617.

[7] 徐明, 焦建军, 龙文. 基于Logistic模型和随机差分变异的正弦余弦算法[J]. 计算机科学, 2020, 47(2): 206-212.

[8] 王联国, 刘小娟. 基于采蜜机制的正弦余弦算法及其在机械优化设计中的应用[J]. 中国机械工程, 2021, 32(21): 2577-2589.

[9] 何庆, 徐钦帅, 魏康园. 基于改进正弦余弦算法的无线传感器节点部署优化[J]. 计算机应用, 2019, 39(7): 2035-2043.

[10] 杨晓倩, 李琦, 李豪欣, 等. 基于改进正弦余弦算法的MIMO雷达相位编码信号集设计[J]. 电光与控制, 2021, 28(6): 90-94.

[11] 郭文艳, 王远, 戴芳, 等. 基于精英混沌搜索策略的交替正余弦算法[J]. 控制与决策, 2019, 34(8): 1654-1662.

[12] 方旭阳, 武相军, 游大涛. 具有学习机制的正弦余弦算法[J]. 计算机应用研究, 2020, 37(3): 809-813.

[13] 周晓君, 阳春华, 桂卫华. 状态转移算法原理与应用[J]. 自动化学报, 2020, 46(11): 2260-2274.

[14] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.

[15] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems[J]. Information Sciences, 2012, 183(1): 1-15.

[16] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitational search algorithm[J]. Information Sciences, 2009, 179(13): 2232-2248.

[17] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-verse optimizer: a nature-inspired algorithm for global optimization[J]. Neural Computing and Applications, 2016, 27(2): 495-513.

[18] 李银通, 韩统, 赵辉, 等. 自学习策略和Lévy飞行的正弦余弦优化算法[J]. 重庆大学学报, 2019, 42(9): 56-66.

[19] 牛培峰, 吴志良, 马云鹏, 等. 基于正弦余弦算法的汽轮机热耗率预测[J]. 动力工程学报, 2018, 38(2): 85-91.

[20] 肖子雅, 刘升, 韩斐斐, 等. 正弦余弦指引的乌鸦搜索算法研究[J]. 计算机工程与应用, 2019, 55(21): 52-59.

责任编辑：张自强