刘勇, 赵启巍, 陈茵理. 多策略算术优化算法[J]. 智能计算机与应用,2025,15(6):32-42. DOI:10.20169/j.issn.2095-2163.250606

多策略算术优化算法

刘 勇, 赵启巍, 陈茵理
(上海理工大学 管理学院, 上海 200093)

摘 要: 针对算术优化算法寻优精度低和计算速度慢等缺点,提出一种多策略算术优化算法。首先,结合一种动态莱维飞行优化策略对当前最优解进行动态更新,并用于算法所有的解更新方程;其次,采用 Sigmoid 函数设计非线性的数学优化加速器函数,实现全局探索和局部开发两种优化阶段的选择;最后,引入基于正余弦函数的动态权重,较大的权重使算法在优化过程前期具有较强的全局探索能力,而较小的权重使算法在优化过程后期具有较强的局部开发能力。通过实验验证了 3 种策略组合的有效性。采用 30 维、500 维和 1 000 维的函数进行数值实验,并将新算法与基本算术优化算法、灰狼优化算法、鲸鱼优化算法、麻雀搜索算法和正余弦优化算法进行对比,并进行 Wilcoxon 秩和检验。实验结果表明新算法具有更好的优化性能。

关键词: 算术优化算法; 动态莱维飞行优化策略; 数学优化加速器函数; 动态权重
中图分类号: TP301 **文献标志码**: A **文章编号**: 2095-2163(2025)06-0032-11

Multi-strategy arithmetic optimization algorithm

LIU Yong, ZHAO Qiwei, CHEN Yinli
(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Aiming at the shortcomings of arithmetic optimization algorithm, such as low optimization accuracy and slow computation speed, a multi-strategy arithmetic optimization algorithm is proposed. Firstly, a dynamic Lévy flight optimization strategy is adopted to dynamically update the current best solution, which is used in all solutions update equations. Secondly, the Sigmoid function is used to design the nonlinear mathematical optimization accelerator function, which realizes the choice of the two optimization phases, including global exploration and local exploitation. Finally, the dynamic weighting coefficient based on the sine cosine function is introduced into the algorithm. In the early stage of the optimization process, the larger weights give the algorithm a strong global exploration capability, while in the later stage of the optimization process, the smaller weights give the algorithm a strong local exploitation capability. The effectiveness of the three strategy combinations is verified by experiments. Thirty, five hundred and one thousand dimensional functions are used in the numerical experiments. The proposed algorithm is compared with arithmetic optimization algorithm, grey wolf optimizer, whale optimization algorithm, sparrow search algorithm, sine cosine algorithm. The Wilcoxon rank-sum test is performed. The experimental results show that the presented algorithm has better optimization performance.

Key words: arithmetic optimization algorithm; dynamic Levy flight optimization strategy; mathematical optimization accelerator function; dynamic weight

0 引 言

2021 年,Abualigah 等学者[1]在研究数学四则运算时,发现乘除运算可以实现智能优化算法的全局探索操作,而加减运算可以实现局部开发操作,并设计出了一种新型元启发式算法-算术优化算法(Arithmetic Optimization Algorithm, AOA)。现有的元启发式算法优化原理可阐述为:

(1)主要基于生物规律,如遗传算法[2]、粒子群优化算法[3]和鲸鱼优化算法[4]等;

(2)主要基于物理学规律,如模拟退火算法[5]、引力搜索算法[6]和光学优化算法[7]等。

基金项目: 教育部人文社会科学研究青年基金项目(21YJC630087); 上海市哲学社会科学规划科课题资助项目(2019BGL014)。

作者简介: 刘 勇(1982—),男,博士(后),副教授, 硕士生导师,主要研究方向:智能优化,服务网络设计与优化,系统工程。Email:1323483815@qq.com; 赵启巍(1998—),女,硕士研究生,主要研究方向:智能优化,系统工程; 陈茵理(2001—),女,硕士研究生,主要研究方向:智能优化,系统工程。

收稿日期: 2023-11-04

这些元启发式方法为许多问题的求解提供了算法支撑，但是也受到这些方法自身优化机制的限制，在求解越来越复杂的难题时暴露出优化精度低等问题[8]。算术优化算法 AOA 是基于四则运算的一种元启发式方法，为优化算法的研究提供了新颖的算法设计思路。

目前，算术优化算法 AOA 的研究还处于起步阶段。已有国内外学者主要针对基本算法收敛性精度低、收敛速度慢等不足提出改进方法。例如郑婷婷等学者[9]通过引入自适应分布变异策略来提高算法的收敛速度，并引入余弦控制因子的动态边界来平衡全局探索和局部开发。兰周新等学者[10]通过 Sobol 初始化序列来增加种群的多样性，重构数学优化加速函数和混沌精英变异策略来改善算法的优化性能。Agushaka 等学者[11]利用 Beta 分布初始化种群，使用自然对数和指数算子生成的高密度值来增强算法的探索能力。Yang 等学者[12]提出一种基于对立学习和螺旋建模的新型算术优化算法。此外，也有学者将算术优化算法应用到实际优化工程问题中。如 Danilo 等学者[13]将算法优化算法与涡旋搜索算法相结合，并应用到光伏发电系统的成本优化问题。

上述研究对基本算术优化算法 AOA 的性能有一定的提升，但涉及算法核心优化机制的研究较少，算法仍容易陷入局部最优，算法的全局探索和局部开发能力有待进一步提升[10-12]。因此，本文提出一种多策略算术优化算法（Multi Strategy Arithmetic Optimization Algorithm，MSAOA）。MSAOA 算法通过动态莱维飞行优化策略实现对当前最优解的更新；利用 Sigmoid 函数构建非线性数学优化加速器函数，用于全局探索和局部开发两种优化阶段的选择操作；采用正余弦函数设计动态权重，在优化过程前期，较大的权重使算法具有较强的全局探索能力，而在优化过程后期，较小的权重使算法具有较强的局部开发能力。采用低维和高维的测试函数进行数值实验。一方面验证 3 种改进策略的有效性；另一方面通过与基本算术优化算法、灰狼优化算法、鲸鱼优化算法、麻雀搜索算法和正余弦优化算法的比较，结果表明所提出的 MSAOA 算法具有更好的计算性能。

1　算术优化算法

算术优化算法 AOA 利用四则运算的 4 种算子，包括乘法 M、除法 D、减法 S 和加法 A。乘法和除法算子用于设计全局探索迭代方程，加法和减法算子用于设计局部开发迭代方程[1]。在算法中，通过数学优化加速器（Math Optimizer Accelerated，MOA）选择进行全局探索、还是局部开发。AOA 算法主要原理如下。

1.1　初始化

AOA 在初始化中按下式生成一组 $N \times n$ 维矩阵：

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,n} \end{bmatrix} \tag{1}$$

其中，N 表示群体规模；n 表示问题空间维数；$x_{i,j}$ 表示第 i 个解在第 j 维的分量。

1.2　基于 MOA 的探索和开发阶段选择

在 AOA 算法优化过程中，通过下式计算得到数学优化加速器 MOA 的函数值，选择进行全局探索、还是局部开发：

$$MOA(C\_Iter) = \mathrm{Min} + C\_Iter \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{M\_Iter}\right) \tag{2}$$

其中，C_Iter 表示当前迭代次数；M_Iter 表示最大迭代次数；Min 和 Max 分别表示数学优化加速器 MOA 函数值的最小值与最大值。

首先生成随机数 $r_1 \in [0,1]$，然后将 r_1 与数学优化加速器 MOA 函数值进行比较。如果 $r_1 > MOA$，算法进行全局探索，否则进行局部开发。

1.3　全局探索阶段

AOA 算法在探索阶段使用乘法 M 算子和除法 D 算子，因这 2 个算子具有较高的分散性，有利于算法进行大范围进行优化搜索，用于设计全局探索迭代方程[1]。首先生成随机数 $r_2 \in [0,1]$，如果 $r_2 < 0.5$，算法使用除法算子执行全局探索操作，否则采用乘法算子执行全局探索操作。迭代方程如下所示：

$$x_{i,j}(C\_Iter + 1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j), & r_2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{3}$$

其中，$x_{i,j}(C\_Iter + 1)$ 表示下一次迭代的第 i 个解在第 j 维的分量；$best(x_j)$ 表示当前最优解在第 j 维的位置；ε 表示取值极小的常数，防止分母为 0；μ 表示调整搜索过程的控制参数；UB_j 和 LB_j 分别表示在第 j 维变量的上界值和下界值；MOP 表示数学优化器系数，数学定义公式如下：

$$MOP = 1 - \frac{C\_Iter^{\frac{1}{\alpha}}}{M\_Iter^{\frac{1}{\alpha}}} \tag{4}$$

其中，α 表示敏感参数。

1.4 局部开发阶段

AOA 算法在开发阶段使用加法 A 算子和减法 S 算子，因这 2 个算子具有低分散性，有利于进行小范围精细搜索，用于设计局部开发迭代方程[1]。首先生成随机数 $r_3 \in [0,1]$，如果 $r_3<0.5$，算法使用减法算子进行局部开发，否则利用加法算子进行局部开发。更新方程为：

$$x_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), r_3 < 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j), \text{otherwise} \end{cases} \tag{5}$$

2 多策略算术优化算法

在算术优化算法 AOA 中，首先进行全局探索的更新方程式(3)和进行局部开发的更新方程式(5)都使用当前最优解。因此，当前最优解直接影响算法优化性能。如果能够提高当前最好解的质量，则可以提高算法的搜索性能。其次，数学优化加速器 MOA 函数决定算法进行全局探索、还是局部开发两个阶段的选择。根据 AOA 算法的设计，在优化过程的早期，MOA 函数值较小，算法进行全局探索；优化过程的后期，MOA 函数值较大，算法进行局部开发。因此，MOA 函数是随迭代次数单调递增函数。在基本算术优化算法 AOA 中，式(2)采用线性单调递增函数设计 MOA 函数，不能体现优化过程 MOA 函数变化的非线性特征，可以采用非线性函数构造 MOA 函数。最后，在全局探索的迭代方程式(3)和局部开发的迭代方程式(5)中，可以引入动态权重系数。在搜索过程的前期，较大的权重使算法具备比较强的全局探索性能，而在搜索过程后期，较小的权重使算法具备比较强的局部开发能力。

因此，本文针对上述 3 种情况，基于动态莱维飞行优化策略对当前最优解进行动态更新；采用 Sigmoid 函数设计非线性的数学优化加速器函数 MOA；引入基于正余弦函数的动态权重。在此基础上，提出多策略算术优化算法。

2.1 动态莱维飞行优化策略

在 AOA 算法中，全局探索的迭代方程式(3)和局部开发的迭代方程式(5)中当前最优解直接决定 AOA 算法的计算精度和寻优速度，通过对当前最优解的优化提升算法性能。本文采用动态莱维飞行策略来优化当前最优解。莱维飞行本质上是一种服从莱维分布的随机游走策略[14]，其运动轨迹如图 1 所示。因其行走步长长短相间的特性，使得搜索群体能够在不同范围内进行寻优，常被用于设计优化策略[15-16]。

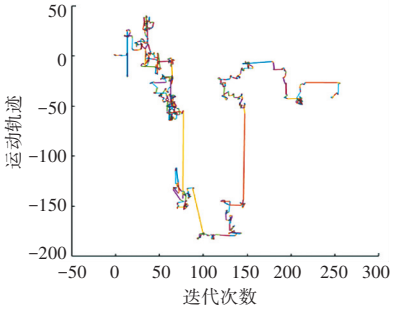


图 1 莱维飞行运动轨迹

Fig. 1 Levy's flight trajectory

莱维分布数学表达式如下：

$$Levy(\beta) = \frac{u}{|v|^{\frac{1}{\beta}}} \tag{6}$$

$$u \sim (0,\sigma_u^2), v \sim (0,\sigma_v^2) \tag{7}$$

其中，$\sigma_u = 1$；$\sigma_v = \left\{ \frac{\Gamma(1+\beta)\sin(\pi/2)}{\Gamma((1+\beta)/2)2^{(\beta-1)/2}\beta} \right\}^{1/\beta}$；$\beta$ 表示常数，且其取值范围为 $0 < \beta < 2$。

莱维飞行的概率密度函数表达式如下：

$$L(X) = \frac{1}{\pi}\int_0^{\infty} \cos(r \cdot x) \cdot e^{-\gamma r^{\beta}} \mathrm{d}r \tag{8}$$

其中，γ 表示比例因子，且 $\gamma > 0$。

对当前最优解采用动态莱维飞行优化策略，其更新公式具体如下：

$$\begin{cases} newx_{\text{best}}^j = x_{\text{best}}^j + Levy(\beta_1) \times x_{\text{best}}^j, & r > \dfrac{C\_Iter}{M\_Iter} \\ newx_{\text{best}}^j = x_{\text{best}}^j + Levy(\beta_2) \times x_{\text{best}}^j, & r \leqslant \dfrac{C\_Iter}{M\_Iter} \end{cases} \tag{9}$$

其中，$newx_{\text{best}}^j$ 表示莱维飞行优化后的最优解在第 j 维的分量；x_{best}^j 表示莱维飞行优化前的最优解在第 j 维的分量；r 表示 0 到 1 之间服从均匀分布的随机数。

参数 β_1 和 β_2 使其随算法迭代次数变化而变化，其数学表达式为：

$$\begin{cases} \beta_1 = \varphi_1 + \varphi_2 \cdot \sin\left[\dfrac{\pi}{2} \cdot \left(\dfrac{C\_Iter}{M\_Iter}\right)^2\right] \\ \beta_2 = \delta_1 + \delta_2 \cdot \cos\left[\dfrac{\pi}{2} \cdot \left(\dfrac{C\_Iter}{M\_Iter}\right)^2 + \dfrac{\pi}{2}\right] \end{cases} \tag{10}$$

其中，$\delta_1, \delta_2, \varphi_1, \varphi_2$ 为常数。

在算法中，β_1 取值比 β_2 取值小，β_1 和 β_2 随迭代次数变化曲线如图 2 所示。

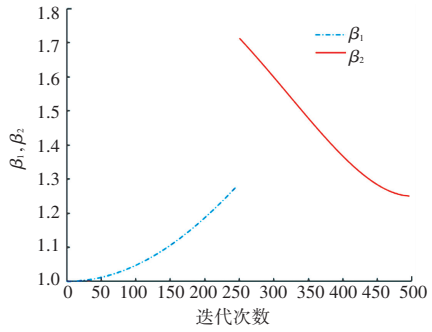


图 2　β_1 和 β_2 随迭代次数变化图

Fig. 2　Variation of β_1 and β_2 with number of iterations

相比较而言，在优化过程前期的最优解质量较差，需要对其进行大幅度的调整；在优化过程后期的最优解质量较好，需要对其进行小幅度的调整。由文献[14-17]可知，当 β 越小，$Levy(\beta)$ 值越大；β 越大，$Levy(\beta)$ 值越小。在算法寻优过程的前期，迭代次数 C_Iter 和最大迭代次数 M_Iter 相差较大，主要利用值较大的 $Levy(\beta_1)$ 对当前最优解进行幅度较大的更新；在寻优过程的后期，最优解质量逐渐变好，而迭代次数 C_Iter 越来越接近最大迭代次数 M_Iter，主要利用值较小的 $Levy(\beta_2)$ 对当前最优解进行幅度较小的优化。

2.2　基于 Sigmoid 函数的数学优化加速器 MOA

AOA 算法进行全局探索和局部开发由数学优化加速器 MOA 函数确定。较小的 MOA 函数值让算法选择全局探索操作，较大的MOA函数值让算法选择局部开发操作。基本算术优化算法AOA采用线性函数设计 MOA 函数，其取值随迭代次数的增加而线性增加。本文考虑采用 Sigmoid 函数来构造一种非线性的 MOA 函数，更加准确表达优化过程。新的数学优化加速器 *IMOA* 函数表达式为：

$$IMOA(C\_Iter)=\frac{1}{2}\mathrm{Min}+\frac{\mathrm{Max}-\mathrm{Min}}{1+e^{\left(-\left(\frac{C\_Iter-\frac{M\_Iter}{2}}{M\_Iter}\right)\times 10\right)}} \tag{11}$$

其中，Min 和 Max 分别表示新的数学优化加速器 IMOA 预设的最小值与最大值；C_Iter 和 M_Iter 分别表示算法的当前迭代次数和最大迭代次数。

2.3　基于正余弦函数的动态权重

研究表明，进一步增强算术优化算法 AOA 的全局探索和局部开发能力可以提高算法的搜索性能[9-12]。在微粒群优化算法的解更新方程中引入动态权重后，可以提高算法的全局探索和局部开发性能[18]。受此启发，在全局探索对应的解更新方程式(3)和局部开发对应的解更新方程式(5)中，本文将基于正余弦函数的动态权重 $\omega(C\_Iter)$ 引入到位置更新公式中。该权重随着迭代次数的变化而进行动态调整，其数学公式为：

$$\omega(C\_Iter)=k\cos\left(\frac{2}{\pi}\cdot\sin\left(\sqrt{\frac{C\_Iter}{M\_Iter}}\right)\right) \tag{12}$$

其中，k 为权重系数。

引入动态权重 $\omega(C\_Iter)$ 后，全局探索对应的位置更新方程和局部开发对应的位置更新方程具体如下：

$$x_{i,j}(C\_Iter+1)=\begin{cases}\omega(C\_Iter)best(x_j)\div(MOP+\varepsilon)\times((UB_j-LB_j)\times\mu+LB_j), & r_2<0.5\\ \omega(C\_Iter)best(x_j)\times MOP\times((UB_j-LB_j)\times\mu+LB_j), & \text{otherwise}\end{cases} \tag{13}$$

$$x_{i,j}(C\_Iter+1)=\begin{cases}\omega(C\_Iter)best(x_j)-MOP\times((UB_j-LB_j)\times\mu+LB_j), & r_3<0.5\\ \omega(C\_Iter)best(x_j)+MOP\times((UB_j-LB_j)\times\mu+LB_j), & \text{otherwise}\end{cases} \tag{14}$$

在算法寻优过程的前期，动态权重 $\omega(C\_Iter)$ 值相对较大，有利于扩大搜索范围，提高算法的全局搜索能力；在算法寻优过程的后期，动态权重 $\omega(C\_Iter)$ 值相对较小，有利于缩小搜索区域，提高算法的局部开发能使得算法不断逼近最优解。

综上所述，本文提出的多策略算术优化算法主要计算步骤如下。

步骤 1　算法参数初始化，随机生成初始搜索群体。

步骤 2　采用式(9)对当前最优个体进行更新。

步骤 3　如果 $r_1 > IMOA$ 函数值，采用式(13)进行全局探索操作；否则，采用式(14)进行局部开发操作。

步骤 4　判断算法是否满足结束条件。满足结束条件，算法停止并输出最好结果；否则转步骤 2。

3　数值实验

为测试多策略算术优化算法 MSAOA 的优化性能，采用 20 个基本测试函数进行数值实验。首先，将所提出的 MSAOA 算法和鲸鱼优化算法(WOA)[4]、灰狼优化算法(GWO)[19-20]、麻雀搜索算法(SSA)[21]、正余弦优化算法(SCA)[22]与算术优化算法(AOA)对求解 30 维的函数优化计算结果进行比较；然后，分析多策略算术优化算法 MSAOA 中 3 种优化策略组合的有效性；最后，将这 6 种算法用于求解 500 维和 1 000 维的函数优化问题，进一步测试 MSAOA 算法的搜索

性能。

3.1 实验环境及测试函数

本实验硬件为 Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz,16.0 GB 内存,软件为 64 位 Windows 11 操作系统,编程工具为 Matlab R2018b。

在实验中,选取了 20 个不同特点的基准测试函数进行对比实验,具体函数信息见表 1。

表 1 测试函数

Table 1 Test functions

函数名	函数	范围	最优值
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	[-100,100]	0
Quartic Function	$f_2 = \sum_{i=1}^{D} i \times x_i^4$	[-1.28,1.28]	0
Schwefel 1.2	$f_3 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
Rastrigin Problem	$f_4 = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
Griewank	$f_5 = \frac{1}{4\,000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
Cigar	$f_6 = x_i^2 + 10^6 \sum_{i=2}^{D} x_i^2$	[-1,1]	0
Bohachevsky	$f_7 = \sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$	[-15,15]	0
AMGM	$f_8 = \left(\frac{\sum_{i=1}^{D} x_i}{D}\right)^2 + (\prod_{i=1}^{D} x_i)^{\frac{2}{D}}$	[0,10]	0
Alpine	$f_9 = \sum_{i=1}^{D} \lvert x_i \sin x_i + 0.1x_i \rvert$	[-10,10]	0
Ackley	$f_{10} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0
Zero Sum	$f_{11} = 1 + \sqrt{1\,000\left(\lvert \sum_{i=1}^{D} x_i \rvert\right)}$	[-1,1]	0
Sodp	$f_{12} = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	[-100,100]	0
Wavy	$f_{13} = 1 - \frac{\sum_{i=1}^{D} \left(\cos(10x_i) \times \exp\left(\frac{-x_i^2}{2}\right)\right)}{D}$	[-10,10]	0
Chung Reynolds	$f_{14} = \left(\sum_{i=1}^{D} x_i^2\right)^2$	[-100,100]	0
Schumer Steiglitz	$f_{15} = \sum_{i=1}^{D} x_i^4$	[-100,100]	0
Brown	$f_{16} = \sum_{i=1}^{D-1} (x_i^{2(x_{i+1}^2+1)} + x_{i+1}^{2(x_i^2+1)})$	[-1,4]	0
Drop Wave	$f_{17} = -\frac{\left(1 + \cos\left(12\sqrt{\sum_{i=1}^{D} x_i^2}\right)\right)}{2 + 0.5\sum_{i=1}^{D} x_i^2}$	[-5.12,5.12]	-1
Infinity	$f_{18} = \sum_{i=1}^{D} (x_i^6 \times 6\sin\left(\frac{1}{x_i}\right) + 2)$	[-1,1]	0
Zacharov	$f_{19} = \sum_{i=1}^{D} x_i^2 + \left(0.5\sum_{i=1}^{D} i \times x_i\right)^2 + \left(0.5\sum_{i=1}^{D} i \times x_i\right)^4$	[-100,100]	0
Schwefel 2.23	$f_{20} = \sum_{i=1}^{D} x_i^{10}$	[-100,100]	0

3.2　低维函数优化实验结果与分析

MSAOA 算法、鲸鱼优化算法(WOA)[4]、灰狼优化算法(GWO)[19-20]、麻雀搜索算法(SSA)[21]、正余弦优化算法(SCA)[22]和算术优化算法(AOA)这 6 种算法在优化 30 维测试函数的实验结果进行比较。所有算法群体规模为 30,最大迭代次数为 500。WOA、GWO、SSA 和 SCA 根据对应文献设置参数。AOA 中的参数设置为 Min = 0.2,Max = 1; α = 5, μ = 0.499。MSAOA 算法和 AOA 算法相同参数设置相同,其它参数取值为 k = 0.2, φ_1 = 1, φ_2 = 0.75, δ_1 = 2, δ_2 = 0.75。对每一个测试函数,每种算法都独立运行 30 次,计算平均值和标准差统计指标,具体实验结果见表 2。

表 2　低维函数计算结果

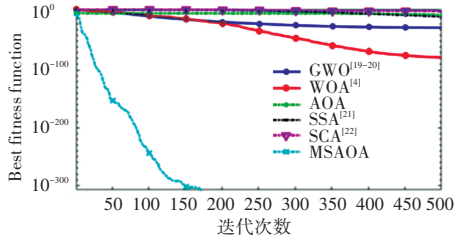
Table 2　Calculation results of low-dimensional functions

函数	指标	WOA[4]	GWO[19-20]	SSA[21]	SCA[22]	AOA	MSAOA
f_1	平均值	7.58E-76	1.15E-27	3.37E-07	1.53E+01	4.61E-06	**0.00E+00**
	标准差	2.12E-75	1.83E-27	3.87E-07	2.48E+01	1.97E-06	**0.00E+00**
f_2	平均值	1.64E-102	4.51E-40	3.66E+07	2.33E+07	4.27E-06	**0.00E+00**
	标准差	6.34E-102	1.11E-39	6.41E+07	3.94E+07	7.48E-06	**0.00E+00**
f_3	平均值	4.60E+04	4.52E-06	1.19E+03	9.22E+03	1.43E-03	**0.00E+00**
	标准差	1.90E+04	6.03E-06	5.35E+02	4.31E+03	1.04E-03	**0.00E+00**
f_4	平均值	3.47E-15	3.01E+00	5.68E+01	3.19E+01	1.97E-06	**0.00E+00**
	标准差	1.33E-14	3.29E+00	2.08E+01	3.64E+01	1.99E-06	**0.00E+00**
f_5	平均值	2.04E-02	6.90E-03	2.33E-02	9.81E-01	1.83E-03	**0.00E+00**
	标准差	5.31E-02	1.13E-02	1.72E-02	5.19E-01	5.70E-03	**0.00E+00**
f_6	平均值	6.72E-71	1.03E-25	8.93E-02	6.95E+02	1.28E-04	**0.00E+00**
	标准差	2.34E-70	1.53E-25	1.12E-01	8.67E+02	2.06E-04	**0.00E+00**
f_7	平均值	**0.00E+00**	**0.00E+00**	1.43E+01	2.27E+00	4.40E-06	**0.00E+00**
	标准差	**0.00E+00**	**0.00E+00**	3.19E+00	2.87E+00	3.48E-06	**0.00E+00**
f_8	平均值	**0.00E+00**	3.97E-42	7.38E-03	**0.00E+00**	9.70E-05	**0.00E+00**
	标准差	**0.00E+00**	9.20E-42	1.25E-02	**0.00E+00**	6.26E-05	**0.00E+00**
f_9	平均值	1.49E-31	5.29E-04	4.00E+00	9.39E-01	9.89E-05	**1.38E-237**
	标准差	8.09E-31	6.77E-04	1.39E+00	1.54E+00	1.22E-04	**0.00E+00**
f_{10}	平均值	4.69E-15	1.14E-13	2.43E+00	1.59E+01	5.39E-04	**8.88E-16**
	标准差	2.96E-15	3.58E-14	5.36E-01	7.64E+00	1.15E-04	**0.00E+00**
f_{11}	平均值	1.04E+00	1.30E+00	1.00E+00	2.54E+00	1.00E+00	**0.00E+00**
	标准差	1.95E-02	1.09E-01	1.77E-03	7.93E-01	2.15E-03	**0.00E+00**
f_{12}	平均值	3.74E-94	2.68E-76	1.35E+20	9.59E+17	8.42E-25	**0.00E+00**
	标准差	9.87E-94	5.89E-76	3.76E+20	2.92E+18	1.56E-24	**0.00E+00**
f_{13}	平均值	**0.00E+00**	1.18E-01	5.66E-01	1.13E-01	7.52E-08	**0.00E+00**
	标准差	**0.00E+00**	9.07E-02	1.73E-01	1.03E-01	4.02E-08	**0.00E+00**
f_{14}	平均值	7.59E-150	2.01E-59	1.62E-18	4.84E-02	1.02E-16	**0.00E+00**
	标准差	2.31E-149	3.75E-59	4.75E-18	1.69E-01	1.18E-16	**0.00E+00**
f_{15}	平均值	8.24E-113	1.73E-45	8.02E-09	1.11E+05	8.66E-12	**0.00E+00**
	标准差	1.78E-112	2.77E-45	9.59E-09	2.29E+05	6.08E-12	**0.00E+00**
f_{16}	平均值	8.93E-78	8.12E-31	6.05E-10	2.89E-03	6.44E-04	**2.31E-177**
	标准差	2.03E-77	7.83E-31	5.76E-10	3.03E-03	4.36E-04	**0.00E+00**
f_{17}	平均值	-9.06E-01	-8.46E-01	-2.58E-01	-6.67E-01	**-1.00E+00**	**-1.00E+00**
	标准差	6.12E-02	7.50E-02	5.79E-02	1.09E-01	**0.00E+00**	**0.00E+00**
f_{18}	平均值	5.29E-120	3.78E-66	3.31E-08	1.73E-03	1.17E-28	**0.00E+00**
	标准差	8.74E-120	7.44E-66	4.69E-08	2.45E-03	1.66E-28	**0.00E+00**
f_{19}	平均值	1.25E+05	3.13E+02	1.43E+04	1.98E+04	4.06E-04	**0.00E+00**
	标准差	3.04E+04	3.58E+02	1.08E+04	5.63E+03	1.15E-04	**0.00E+00**
f_{20}	平均值	6.73E-198	2.86E-104	5.06E-17	1.43E-04	5.05E-48	**0.00E+00**
	标准差	0.00E+00	3.04E-104	1.02E-16	2.80E-04	5.57E-48	**0.00E+00**

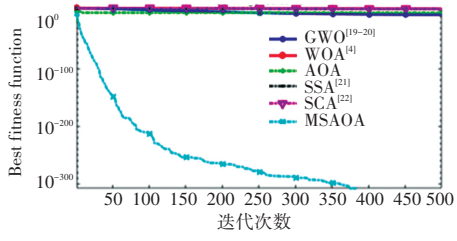
由表 2 可知,在计算这 20 个测试函数时,本文的 MSAOA 算法无论在单峰测试函数、还是多峰测试函数上的寻优性能都是最好的。WOA 算法[4]仅对测试函数 f_7、f_8、f_{13} 能找到全局最优值,但是对 f_3 以及 f_{19} 的寻优效果较差;GWO 算法[19-20]只能对测试函数 f_7 能够找到全局最优值;SSA 算法[21]对 20 个测试函数均无法找到全局最优解,特别是对函数 f_2、f_4 以及 f_{12} 的寻优效果较差;SCA 算法[22]仅对 f_8

能找到全局最优值;对其余 19 个测试函数结果均较差;AOA 算法仅对 f_{17} 能够找到全局最优值;MSAOA 对 f_1、f_2、f_3、f_4、f_5、f_6、f_7、f_8、f_{11}、f_{12}、f_{13}、f_{14}、f_{15}、f_{17}、f_{18}、f_{20} 都能找到最优解且标准差最小,整体寻优效果均优于其余 5 种算法。

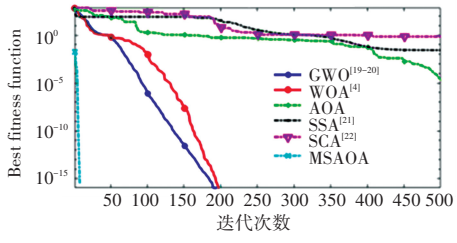
为更直观比较这 6 种算法的优化性能,图 3 给出这 6 种算法在求解部分函数时的寻优曲线。由图 3 可知,本文算法 MSAOA 在寻优精度和速度上均明显优于 AOA、WOA[4]、GWO[19-20]、SSA[21] 以及 SCA[22] 这 5 种方法。综上可知:本文提出的动态莱维飞行优化策略,引用 Sigmoid 函数构造数学优化加速器 MOA 函数以及基于正余弦函数设计动态权重的这 3 种方法能够显著提升基本 AOA 的优化性能。



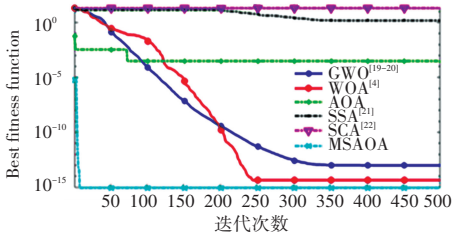
(a) 情况 1



(b) 情况 2



(c) 情况 3



(d) 情况 4

图 3 测试函数寻优曲线

Fig. 3 Optimization curves for test functions

为了进一步比较各算法的寻优性能,对各算法基于表 2 的测试结果进行显著性水平为 5%的 Wilcoxon 秩和检验。表 3 为 MSAOA 分别与 WOA[4]、GWO[19-20]、SSA[21]、SCA[22] 以及 AOA 五个算法的寻优结果秩和检验的 P 值。当 P 值小于 0.05 时,表明 2 个算法的寻优效果有差异,反之则没有。

表 3 30 维函数的 Wilcoxon 秩和检验结果

Table 3 Wilcoxon rank-sum test results of 30-dimensional functions

函数	WOA	GWO	SSA	SCA	AOA
f_1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_4	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_5	4.19E-02	1.37E-03	1.21E-12	1.21E-12	1.21E-12
f_6	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_7	-	-	1.21E-12	1.21E-12	1.21E-12
f_8	-	1.21E-12	1.21E-12	-	1.21E-12
f_9	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{10}	1.30E-08	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{11}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{12}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{13}	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{14}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{15}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{16}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{17}	3.12E-13	1.21E-12	8.28E-13	8.28E-13	-
f_{18}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{19}	3.02E-11	3.02E-11	2.70E-11	3.02E-11	3.02E-11
f_{20}	3.02E-11	3.02E-11	2.70E-11	3.02E-11	3.02E-11

从表 3 中可以看出,对于检验中出现“-”表明两算法性能没有明显差异,即均找到了全局最优值,其余的测试结果表明对比的 2 个算法寻优结果均有明显差异,进一步验证了改进后的算法性能得到了显著提升。

3.3 3 种优化策略组合的有效性分析

在基本 AOA 算法基础上,引入动态莱维飞行优化策略,利用 Sigmoid 函数构造数学优化加速器 MOA 函数以及基于正余弦函数设计动态权重,提出本文的多策略算术优化算法 MSAOA。将通过实验验证这 3 种策略对算法优化性能的有效性。将仅采用第 1、2、3 种策略的 AOA 算法分别记为 AOA-1、AOA-2、AOA-3;将采用第 1、2 种、第 1、3 种和第 2、3 种策略的 AOA 算法分别记为 AOA-4,AOA-5,AOA-6。因篇幅限制,选取 20 个测试函数中的 15

个函数进行展示。算法参数同上，计算结果见表 4。

从表 4 可以发现，采用单一策略的 3 种算法 AOA-1，AOA-2，AOA-3 的优化性能均优于基本算法 AOA。进一步比较可以发现，采用 2 种策略的算法 AOA-4，AOA-5，AOA-6 的结果均优于单一策略的 3 种算法。采用 3 种策略的 MSAOA 算法结果均优于 2 种策略结合的算法结果，进一步验证了 3 种策略组合的有效性。

表 4　策略组合优化结果

Table 4　Results of three optimization strategies

函数	指标	AOA	AOA-1	AOA-2	AOA-3	AOA-4	AOA-5	AOA-6	MSAOA
f_1	平均值	4.61E-06	2.45E-99	8.90E-84	2.68E-59	4.06E-203	5.72E-172	2.63E-233	0.00E+00
	标准差	1.97E-06	5.99E-99	4.49E-83	1.36E-59	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	平均值	4.27E-06	2.86E-214	0.00E+00	3.26E-114	0.00E+00	8.24E-294	0.00E+00	0.00E+00
	标准差	7.48E-06	0.00E+00	0.00E+00	1.51E-114	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_3	平均值	1.43E-03	4.56E-98	2.29E-88	4.26E-59	2.14E-198	4.01E-142	6.75E-221	0.00E+00
	标准差	1.04E-03	1.02E-97	1.25E-87	1.37E-59	0.00E+00	1.06E-141	0.00E+00	0.00E+00
f_4	平均值	1.97E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	1.99E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_5	平均值	1.83E-03	0.00E+00	1.33E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	5.70E-03	0.00E+00	1.38E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_6	平均值	1.28E-04	1.38E-102	0.00E+00	1.29E-57	0.00E+00	8.24E-163	0.00E+00	0.00E+00
	标准差	2.06E-04	3.07E-102	0.00E+00	1.22E-57	0.00E+00	2.22E-162	0.00E+00	0.00E+00
f_7	平均值	4.40E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	3.48E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_8	平均值	9.70E-05	6.22E-63	1.08E-05	0.00E+00	3.76E-66	0.00E+00	0.00E+00	0.00E+00
	标准差	6.26E-05	1.52E-62	1.00E-05	0.00E+00	6.36E-66	0.00E+00	0.00E+00	0.00E+00
f_9	平均值	9.89E-05	7.16E-53	3.51E-128	1.37E-31	6.60E-193	3.82E-85	4.33E-210	1.38E-237
	标准差	1.22E-04	2.18E-52	1.07E-127	1.00E-31	0.00E+00	1.01E-84	0.00E+00	0.00E+00
f_{10}	平均值	5.39E-04	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	标准差	1.15E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	平均值	8.42E-25	5.39E-123	9.58E-156	7.16E-92	7.66E-252	1.85E-197	1.06E-308	0.00E+00
	标准差	1.56E-24	1.04E-122	2.53E-155	1.90E-91	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{14}	平均值	1.02E-16	2.86E-235	0.00E+00	4.41E-123	0.00E+00	6.31E-299	0.00E+00	0.00E+00
	标准差	1.18E-16	0.00E+00	0.00E+00	1.60E-124	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{15}	平均值	8.66E-12	9.14E-228	5.64E-188	2.20E-119	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	6.08E-12	0.00E+00	0.00E+00	9.02E-120	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{18}	平均值	1.17E-28	2.90E-308	0.00E+00	1.64E-191	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	标准差	1.66E-28	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{19}	平均值	4.06E-04	1.46E-96	3.74E-79	4.10E-59	1.78E-154	3.32E-157	1.43E-197	0.00E+00
	标准差	1.15E-04	3.27E-96	8.36E-79	1.96E-59	4.06E-203	6.85E-157	0.00E+00	0.00E+00

3.4　高维函数优化实验结果与分析

上述实验结果表明 MSAOA 算法求解低维函数优化具有较好的计算性能。为测试算法求解高维函数优化的性能，采用 MSAOA 与 WOA[4]、GWO[19-20]、SSA[21]、SCA[22] 以及 AOA 这 6 种算法分别求解 500 维和 1 000 维的函数。参数设置保持不变，每种算法独立运行 30 次。计算结果见表 5。由实验结果可知，MSAOA 算法求解高维函数的寻优效果仍然较好，特别是对于测试函数 f_2、f_3、f_4、f_5、f_6、f_7、f_8、f_{12}、f_{13}、f_{17}、f_{18}、f_{19} 以及 f_{20} 在 500 维和 1 000 维时均能找到最优解，对于其余函数的测试结果也有较明显的提升，进一步说明了 MSAOA 算法的优越性。

表 5 高维函数计算结果

Table 5 Calculation results of high-dimensional functions

函数	算法	D = 500		D = 1 000	
		平均值	标准差	平均值	标准差
f_1	WOA	4.00E-72	8.91E-72	3.16E-72	6.28E-72
	GWO	1.75E-03	6.28E-04	2.95E-01	7.80E-02
	SSA	9.36E+04	8.24E+03	2.38E+05	6.88E+03
	SCA	2.18E+05	7.38E+04	3.82E+05	9.22E+04
	AOA	5.32E-01	4.28E-02	1.48E+00	6.46E-02
	MSAAOA	**3.30E-108**	**1.47E-107**	**7.69E-107**	**4.19E-106**
f_2	WOA	2.50E-94	5.08E-94	2.57E-99	5.23E-99
	GWO	1.84E-03	1.42E-03	2.32E+02	1.00E+02
	SSA	6.64E+09	1.57E+09	4.46E+11	1.59E+10
	SCA	6.65E+11	2.08E+11	2.33E+13	5.43E+12
	AOA	1.56E-01	2.67E-01	7.03E-03	1.30E-03
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_3	WOA	3.48E+07	1.03E+07	1.53E+08	5.92E+07
	GWO	3.02E+05	5.43E+04	1.26E+06	2.23E+05
	SSA	1.79E+06	6.60E+05	4.26E+06	4.41E+05
	SCA	6.56E+06	5.62E+05	2.91E+07	7.99E+06
	AOA	6.32E+00	3.09E+00	3.54E+01	3.22E+00
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_4	WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GWO	7.22E+01	2.62E+01	2.05E+02	3.08E+01
	SSA	3.15E+03	1.18E+02	7.79E+03	1.66E+02
	SCA	1.44E+03	6.87E+02	2.27E+03	1.02E+03
	AOA	1.03E-02	1.28E-03	3.78E-02	1.52E-03
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_5	WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GWO	1.75E-02	3.52E-02	1.80E-02	5.04E-03
	SSA	8.27E+02	5.07E+01	2.10E+03	1.53E+02
	SCA	2.57E+03	5.61E+02	5.53E+03	1.12E+03
	AOA	1.33E+03	3.44E+02	1.33E+04	1.81E+03
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_6	WOA	2.73E-65	4.24E-65	1.84E-71	2.41E-71
	GWO	2.27E-01	6.99E-02	2.51E+01	5.03E+00
	SSA	9.61E+06	5.79E+05	2.29E+07	7.18E+05
	SCA	1.77E+07	2.11E+06	5.75E+07	1.71E+07
	AOA	6.43E-01	3.20E-02	1.94E+00	8.03E-02
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GWO	1.29E-03	4.82E-04	2.75E+00	1.51E+00
	SSA	6.91E+03	4.48E+02	1.63E+04	8.81E+02
	SCA	1.40E+04	1.37E+03	3.62E+04	3.16E+03
	AOA	7.57E-02	7.76E-03	2.72E-01	1.99E-02
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_8	WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GWO	1.24E-10	4.30E-11	6.77E-10	3.87E-10
	SSA	7.29E+00	4.33E-01	1.13E+01	2.50E-01
	SCA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	AOA	1.97E+00	4.76E-01	1.57E-01	2.87E-02
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	WOA	2.18E-51	4.38E-51	2.88E-49	5.80E-49
	GWO	5.90E-02	1.11E-02	2.27E+00	2.48E+00
	SSA	3.13E+02	6.14E+00	7.23E+02	1.54E+01
	SCA	1.44E+02	5.89E+01	2.96E+02	1.35E+02
	AOA	5.22E-02	9.36E-03	1.55E-01	5.00E-03
	MSAAOA	**6.06E-58**	**3.25E-57**	**1.80E-54**	**8.38E-54**
f_{10}	WOA	4.44E-15	3.23E-15	4.44E-15	2.29E-15
	GWO	1.74E-03	2.50E-04	1.63E-02	1.88E-03
	SSA	1.43E+01	2.34E-01	1.44E+01	1.42E-01
	SCA	1.70E+01	4.73E+00	1.88E+01	4.07E+00
	AOA	2.71E-02	9.13E-04	3.34E-02	5.32E-04
	MSAAOA	**8.88E-16**	**0.00E+00**	**8.88E-16**	**0.00E+00**
f_{11}	WOA	1.31E+00	1.88E-01	1.17E+00	8.79E-02
	GWO	2.01E+00	1.54E-01	1.63E+00	1.65E-01
	SSA	1.01E+00	2.84E-03	1.01E+00	3.06E-03
	SCA	3.92E+00	7.69E-01	3.86E+00	1.29E+00
	AOA	1.04E+00	1.50E-02	1.07E+00	2.91E-02
	MSAAOA	**7.00E-01**	**4.66E-01**	**8.00E-01**	**4.07E-01**

续表 5

Table 5

函数	算法	D = 500		D = 1 000	
		平均值	标准差	平均值	标准差
f_{12}	WOA	7.41E-98	1.51E-97	7.21E-103	1.47E-102
	GWO	2.06E-31	4.18E-31	6.29E-50	1.28E-49
	SSA	Inf	NaN	Inf	NaN
	SCA	Inf	NaN	Inf	NaN
	AOA	1.23E-24	2.49E-24	2.80E-24	5.65E-24
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{13}	WOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GWO	1.21E-01	3.05E-02	1.32E-01	2.82E-02
	SSA	8.23E-01	2.12E-02	8.47E-01	1.80E-02
	SCA	2.85E-01	1.10E-01	1.52E-01	1.05E-01
	AOA	4.56E-05	2.83E-06	7.94E-05	1.89E-06
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{14}	WOA	8.02E-145	1.63E-144	1.39E-140	2.83E-140
	GWO	1.06E-11	7.63E-12	5.97E-07	3.90E-07
	SSA	5.77E+04	7.53E+03	3.64E+05	3.56E+04
	SCA	2.31E+05	1.58E+05	1.46E+06	6.35E+05
	AOA	2.82E-09	2.54E-10	3.23E-08	2.01E-09
	MSAAOA	**1.60E-226**	**0.00E+00**	**2.82E-231**	**0.00E+00**
f_{15}	WOA	5.59E-109	1.14E-108	3.48E-102	7.00E-102
	GWO	2.57E-04	1.88E-04	3.21E-01	1.63E-01
	SSA	4.42E+07	3.43E+06	1.35E+08	5.31E+06
	SCA	2.06E+09	5.65E+08	5.79E+09	7.27E+08
	AOA	2.75E-03	1.46E-03	1.13E-02	5.72E-04
	MSAAOA	**5.84E-206**	**0.00E+00**	**8.78E-223**	**0.00E+00**
f_{16}	WOA	3.66E-74	6.97E-74	2.02E-68	4.11E-68
	GWO	1.21E-04	7.73E-05	1.26E+02	2.22E+02
	SSA	3.28E+04	7.69E+03	1.12E+05	1.62E+04
	SCA	4.16E+13	5.45E+13	1.21E+16	1.05E+16
	AOA	5.95E+13	1.11E+14	3.02E+17	1.74E+17
	MSAAOA	**9.69E-110**	**3.34E-109**	**8.38E-108**	**1.10E-107**
f_{17}	WOA	-8.43E-01	1.28E-01	-8.76E-01	7.50E-02
	GWO	-2.89E-01	1.52E-10	-1.09E-01	1.14E-02
	SSA	-1.02E-02	5.66E-04	-2.03E-01	4.02E-01
	SCA	-5.06E-03	9.33E-04	-2.53E-03	1.01E-03
	AOA	-9.98E-01	8.76E-05	-7.95E-01	4.03E-01
	MSAAOA	**-1.00E+00**	**0.00E+00**	**-1.00E+00**	**0.00E+00**
f_{18}	WOA	1.40E-121	2.85E-121	5.70E-116	1.16E-115
	GWO	3.04E-13	5.40E-13	1.67E-08	1.91E-08
	SSA	6.67E-02	1.19E-02	2.83E-01	2.37E-02
	SCA	4.42E+01	3.47E+00	1.11E+02	9.50E+00
	AOA	2.21E-21	7.38E-22	1.32E-20	6.31E-21
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{19}	WOA	1.65E+06	4.62E+04	3.35E+06	7.01E+04
	GWO	9.98E+05	1.26E+05	2.06E+06	5.03E+04
	SSA	1.59E+06	5.29E+05	3.52E+06	1.29E+06
	SCA	1.98E+06	8.73E+05	9.83E+06	1.04E+07
	AOA	9.67E+01	1.94E+02	2.38E+00	2.99E-01
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{20}	WOA	3.60E-176	0.00E+00	1.70E-179	0.00E+00
	GWO	2.11E-16	2.41E-16	2.29E-10	2.09E-10
	SSA	2.45E-04	5.66E-05	1.63E-03	3.71E-04
	SCA	1.65E+01	1.81E+00	4.74E+01	2.18E+00
	AOA	3.53E-38	8.75E-39	3.99E-37	1.27E-37
	MSAAOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

仍然采用显著性水平为 5%的 Wilcoxon 秩和检验对各算法基于表 5 的计算结果进行统计检验。表 6 和表 7 分别表示维数在 500 维和 1000 维下 MSAOA 与 WOA[4]、GWO[19-20]、SSA[21]、SCA[22]以及 AOA 五个算法的秩和检验结果。从表 6 和表 7 的实验结果可知，在求解高维函数优化问题时，MSAOA 算法和其余 5 种对比算法之间具有显著性

的差异，MSAOA 算法搜索性能更好。

表 6　500 维函数的 Wilcoxon 秩和检验结果

Table 6　Wilcoxon rank－sum test results of 500－dimensional functions

函数	WOA[4]	GWO[19-20]	SSA[21]	SCA[22]	AOA
f_1	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_4	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_5	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_6	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_7	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_8	-	1.21E-12	1.21E-12	-	1.21E-12
f_9	9.79E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{10}	1.06E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{11}	8.99E-12	8.99E-12	8.99E-12	8.99E-12	8.99E-12
f_{12}	1.21E-12	1.21E-12	1.69E-14	1.69E-14	1.21E-12
f_{13}	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{14}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{15}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{16}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{17}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{18}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{19}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{20}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12

表 7　1 000 维 Wilcoxon 秩和检验结果

Table 7　Wilcoxon rank－sum test results of 1 000－dimensional functions

函数	WOA[4]	GWO[19-20]	SSA[21]	SCA[22]	AOA
f_1	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_2	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_4	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_5	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_6	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_7	-	-	1.21E-12	1.21E-12	1.21E-12
f_8	-	1.21E-12	1.21E-12	-	1.21E-12
f_9	6.77E-05	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{10}	1.07E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{11}	6.30E-12	6.30E-12	6.30E-12	6.30E-12	6.30E-12
f_{12}	1.21E-12	1.21E-12	1.69E-14	1.69E-14	1.21E-12
f_{13}	-	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{14}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{15}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{16}	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
f_{17}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{18}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{19}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
f_{20}	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12

4　结束语

为改进基本算术优化算法的搜索性能，本文提出一种多策略算术优化算法 MSAOA，主要包括 3 种优化策略：

（1）采用动态莱维飞行优化策略对当前最优解质量做进一步提升，并应用到算法的所有迭代方程中，促进优化信息的共享。

（2）基于 Sigmoid 函数设计随迭代次数单调递增的非线性数学优化加速器函数，替代现有的线性数学优化加速器函数。

（3）在全局探索和局部开发的解更新方程中，引入基于正余弦函数的动态权重，较大权重保证算法在寻优过程前期具有较强全局探索能力，而较小权重保证算法在后期具有较强局部开发能力。通过采用 1 种、2 种和 3 种优化策略的对比实验，验证了这 3 种优化策略组合的有效性。采用低维和高维函数进行数值实验，并将 MSAOA 算法和 WOA[4]、GWO[19-20]、SSA[21]、SCA[22]以及 AOA 五种算法进行比较，Wilcoxon 秩和检验结果表明 MSAOA 算法良好的寻优性能在统计检验上是显著的。将多策略算术优化算法 MSAOA 应用到多目标应急资源调度问题是进一步的研究方向。

参考文献

[1] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 113609.

[2] HOLLAND J H. Genetic algorithms [J]. Scientific American, 1992, 267 (1): 66-72.

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

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

[5] KIRKPATRICK S, GELATT C D, VECCHI M P. Optimization by simulated annealing [J]. Science, 1983, 220 (4598): 671-680.

[6] 韩维，崔荣伟，苏析超，等. 基于双种群模糊引力搜索算法的舰载机甲板作业调度[J]. 控制与决策，2021，36(11)：2751-2759.

[7] 王金叶，马良，刘勇. 量子光学优化算法[J]. 计算机应用研究，2018，35(3)：654-657.

[8] 刘勇，马良，张惠珍，等. 智能优化算法[M]. 上海：上海人民出版社，2019.

[9] 郑婷婷，刘升，叶旭. 自适应 t 分布与动态边界策略改进的算术优化算法[J]. 计算机应用研究，2022，39(5)：1410-1414.

[10] 兰周新，何庆. 多策略融合算术优化算法及其工程优化[J]. 计算机应用研究，2022，39(3)：758-763.

[11] AGUSHAKA J O, EZUGWU A E. Advanced arithmetic optimization algorithm for solving mechanical engineering design

problems [J]. Plos One, 2021,16(8): 0255703.
[12] YANG Yang, QIAN Chen, LI Haomiao, et al. An efficient DBSCAN optimized by arithmetic optimization algorithm with Opposition-based learning [J]. The Journal of Supercomputing 2022,78(18):19566-19604.
[13] DANILO M O, GIRAL RDA, HERNáNDEZ J C. Efficient integration of PV sources in distribution networks to reduce annual investment and operating costs using the modified arithmetic optimization algorithm[J]. Electronics, 2022,11(11):1680.
[14] GIOVANNI I, VLADEMIR C D, VINICIUS V D M. An improved Jaya optimization algorithm with Levy flight [J]. Expert Systems With Applications,2021,165:113902.
[15] GAO Shuzhi, GAO Yue, ZHANG Yimin, et al. Adaptive cuckoo algorithm with multiple search strategies [J]. Applied Soft Computing,2021,106:107181.
[16] 梁静. 融合自适应权重与 Levy 飞行的拉丁超立方体海鸥优化算法及应用[J]. 智能计算机与应用,2022,12(11):216-223.
[17] LAN K T, LAN C H. Notes on the distinction of Gaussian and Cauchy mutations [C]//Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications. Piscataway,NJ:IEEE, 2008: 272-277.
[18] HINDRIYANTO D P, WEE P M. Particle swarm optimization with adaptive selection of inertia weight strategy[J]. International Journal of Computational Science and Engineering, 2016, 13 (1): 38-47.
[19] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer [J]. Advances in Engineering Software,2014,69:46-61.
[20] ELGHAMRAWY S M, HASSANIEN A E. A hybrid genetic-grey wolf optimization algorithm for optimizing Takagi-Sugeno-Kang fuzzy systems [J]. Neural Computing and Applications, 2022, 34(19):17051-17069.
[21] HUANG Zuwei, ZHU Dongli, LIU Yujia, et al. Multi-strategy sparrow search algorithm with non-uniform mutation[J]. Systems Science & Control Engineering,2022,10(1):936-954.
[22] MOOKIAH S, PARASURAMAN K, KUMAR C S. Color image segmentation based on improved sine cosine optimization algorithm [J]. Soft Computing, 2022, 26(23):13193-13203.