多策略融合的改进天鹰优化算法

张长胜,张健忠,钱 斌,胡 蓉
(昆明理工大学信息工程与自动化学院,云南昆明 650500)

摘 要: 为了解决天鹰优化算法(Aquila Optimization algorithm,AO)易陷入局部最优及收敛速度慢的问题,本文提出一种多策略融合的改进天鹰优化算法(Multi-Strategy Integration Aquila Optimization algorithm,MSIAO). 该算法采用结合Tent混沌映射的折射反向学习初始化种群以提高算法前期的搜索效率,根据种内互助及优化策略解决算法寻优停滞的缺陷,并通过基于Bernoulli混沌序列的自适应权重策略提高算法的收敛速度,引入了柯西-高斯变异算子增强算法迭代后期逃逸局部极值的能力. 本文对10个基准函数、部分CEC2014测试函数集进行实验,并将MSIAO用于2个工程设计优化问题. 结果表明,对于高维单峰、高维多峰以及固定维复杂多模态函数,MSIAO比AO具有更高的收敛精度和更快的收敛速度;MSIAO对压力容器与焊接梁优化设计的经济成本较AO分别节约4.62%、0.77%,验证了MSIAO对于处理机械工程问题的实用性和优越性.

关键词: 天鹰优化算法;折射反向学习;种内互助;Bernoulli序列;自适应权重;柯西-高斯变异
基金项目: 国家自然科学基金(No.51665025,No.61963022)
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112(2023)05-1245-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220205

Improved Aquila Optimization Based on Multi-Strategy Integration

ZHANG Chang-sheng, ZHANG Jian-zhong, QIAN Bin, HU Rong
(*Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan* 650500, *China*)

Abstract: In order to solve the problem that aquila optimization algorithm (AO) is easy to fall into local optimum and slow convergence, this paper proposes an improved aquila optimization algorithm with multi-strategy integration (MSIAO). In this algorithm, the refracted opposition-based learning combined with Tent chaotic map is used to initialize the population to improve the early search efficiency of the algorithm, and intraspecific and mutual assistance and optimization strategy are used to solve the problem of optimization stagnation of the algorithm. The convergence speed of the algorithm is improved by an adaptive weighting strategy based on Bernoulli chaotic sequences. Cauchy-Gaussian mutation operator is introduced to enhance the ability of the algorithm to escape local extremum in the later iteration. This paper conducts experiments on 10 benchmark functions and some CEC2014 test function sets, and the proposed MSIAO is applied to 2 engineering design optimization problems. The results show that MSIAO has higher convergence accuracy and faster convergence speed than AO for high-dimensional single-peak, high-dimensional multi-peak and fixed-dimensional complex multimode functions. Compared with AO, MSIAO saves 4.62% and 0.77% in economic cost of optimal design of pressure vessel and welding beam, which verifies the practicability and superiority of MSIAO in dealing with mechanical engineering problems.

Key words: aquila optimization; refracted opposition-based learning; intraspecific and mutual assistance; Bernoulli sequence; adaptive weight; Cauchy-Gaussian mutation
Foundation Item(s): National Natural Science Foundation of China (No.51665025, No.61963022)

1 引言

天鹰优化算法(Aquila Optimization algorithm,AO)是Abualigah等人[1]在2021年提出的一种新型元启发优化算法,其灵感源于天鹰种群的狩猎行为. 因其具有构造简便、易于实现、寻优效率较高等优势,目前,AO已成功运用于神经网络参数优化[2]、网络资源优化调度[3]、燃料电池系统设计[4]及石油产量预测[5]等领域.

在前期研究中,学者们发现,相较于传统元启发算

收稿日期:2022-03-01;修回日期:2022-07-19;责任编辑:覃怀银

法，如灰狼优化算法（Grey Wolf Optimization，GWO）[6]、樽海鞘群算法（Salp Swarm Algorithm，SSA）[7]、鲸鱼优化算法（Whale Optimization Algorithm，WOA）[8]、粒子群算法（Particle Swarm Optimization，PSO）[9]等，AO算法在基础性能测试方面具有更高的寻优精度，然而在寻优过程中仍存在算法收敛速率慢和易陷入局部极值等不足，对于存在较多局部极值、维度较高等优化问题尤为突出．为提高AO算法在实际应用中的优化能力，部分学者通过引入优秀搜索算子对其改进：文献[10]引入自适应权值算子替换全局搜索阶段的随机数，提升算法收敛速度机搜索，且在局部搜索阶段引入混沌算子代替随算子，避免算法陷入局部最优；文献[11]运用正弦混沌映射取代全局寻优中的随机搜索算子，避免算法早熟，并引入莱维飞行机制改进算法全局寻优能力；文献[12]引入对抗搜索算子提升种群质量，同时利用小波算子对全局搜索进行扰动，减少算法陷入局部最优概率．此外，还有学者结合其他优化算法对AO改进：文献[13]糅合了AO全局探索机制与HHO局部开发方式，充分发挥各算法优势，提升算法持续搜索能力；文献[14]采用WOA算法开发能力较强的螺旋更新策略和收缩包围机制改善AO空间探索能力及寻优灵活性，提升算法寻优精度．

上述方法有效改善了AO寻优性能，但均未解决其寻优机制导致算法易陷入局部最优的固有缺陷，甚至部分算法杂交耦合导致灵活性降低，时间复杂度增加．鉴于此，本文分别从种群初始化、全局搜索、局部开发方面对AO进行改进，同时优化各阶段狩猎行为，提出一种多策略融合的改进天鹰优化算法（Multi-Strategy Integration Aquila Optimization，MSIAO）．

2 天鹰优化算法仿生原理

AO算法的寻优机制主要模拟了天鹰在捕食过程中的四种狩猎方式[1]：①利用俯身高空翱翔选择搜索空间；②通过短滑翔飞行在等高线发散空间内探索猎物位置；③以慢下降攻击的低空飞行方式在收敛空间内进行小范围探索；④以步行突袭的方式靠近搜索区域内猎物所在位置并抓取猎物．

AO算法迭代寻优时，天鹰种群位置通过式(1)生成：

$$\begin{aligned}&\boldsymbol{X}_i=\mathrm{LB}_i^j+(\mathrm{UB}_i^j-\mathrm{LB}_i^j)\times\mathrm{rand},\\&i=1,2,\cdots,N;\ j=1,2,\cdots,\mathrm{Dim}\end{aligned}\tag{1}$$

式(1)中，$\boldsymbol{X}_i$为第i只天鹰的位置，LB_i^j和UB_i^j分别表示第i只天鹰在第j维度的最小值与最大值，rand为0到1之间的随机值．

狩猎方式一数学表达式为

$$\boldsymbol{X}_a(t+1)=\boldsymbol{X}_{\mathrm{best}}(t)\times(1-t/T)+\boldsymbol{X}_M(t)-\boldsymbol{X}_{\mathrm{best}}(t)\times\mathrm{rand}\tag{2}$$

$$\boldsymbol{X}_M(t)=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}_i(t)\tag{3}$$

式(2)中，$\boldsymbol{X}_{\mathrm{best}}(t)$是第$t$次迭代最优位置；(1-$t$/$T$)控制扩展搜索；$\boldsymbol{X}_i(t)$为第$t$次迭代时第$i$个个体当前所在位置．$\boldsymbol{X}_M(t)$为第$t$次迭代时所有个体当前位置的均值，$t$和$T$分别表示当前迭代和最大迭代次数．

狩猎方式二数学表达式为

$$\boldsymbol{X}_b(t+1)=\boldsymbol{X}_{\mathrm{best}}(t)\times\mathrm{Levy(Dim)}+\boldsymbol{X}_R(t)+(\boldsymbol{y}-\boldsymbol{x})\times\mathrm{rand}\tag{4}$$

式(4)中，$\boldsymbol{X}_R(t)$为种群范围[1，N]内的随机个体，Levy(Dim)为莱维飞行分布函数，由式(5)计算得到：

$$\mathrm{Levy(Dim)}=\frac{s_1\times u\times\varepsilon}{v^{1/\beta}},\varepsilon=\frac{\Gamma(1+\beta)\times\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\times\beta\times2^{\frac{\beta-1}{2}}}\tag{5}$$

式(5)中，s_1=0.01，β=1.5，u和v是0~1的随机数，且均服从正态随机分布．$\boldsymbol{y}$和$\boldsymbol{x}$模拟搜索轨迹中的螺旋形状，数学模型如式(6)所示：

$$\boldsymbol{y}=\boldsymbol{r}\times\cos(\boldsymbol{\theta}),\ \boldsymbol{x}=\boldsymbol{r}\times\sin(\boldsymbol{\theta})\tag{6}$$

$$\boldsymbol{r}=r_1+U\times\boldsymbol{D}_1,\ \boldsymbol{\theta}=-\omega\times\boldsymbol{D}_1+\theta_1\tag{7}$$

式(7)中，r_1为1~20的随机整数，$\boldsymbol{D}_1$为1到Dim的向量，ω=0.005，U=0.005 65，$\theta_1=(3\times\pi)/2$.

狩猎方式三数学表达式为

$$\begin{aligned}\boldsymbol{X}_c(t+1)=&[\boldsymbol{X}_{\mathrm{best}}(t)-\boldsymbol{X}_M(t)]\times\alpha-\mathrm{rand}\\&+[(\mathrm{UB}_i^j-\mathrm{LB}_i^j)\times\mathrm{rand}+\mathrm{LB}_i^j]\times\delta\end{aligned}\tag{8}$$

式(8)中，α和δ取0.1.

狩猎方式四数学表达式为

$$\begin{aligned}\boldsymbol{X}_d(t+1)=&Q(t)\times\boldsymbol{X}_{\mathrm{best}}(t)-G_1\times\boldsymbol{X}(t)\times\mathrm{rand}\\&-G_2\times\mathrm{Levy(Dim)}+G_1\times\mathrm{rand}\end{aligned}\tag{9}$$

$$Q(t)=t^{(2\times\mathrm{rand}-1)/(1-t)^2}\tag{10}$$

$$G_1=2\times\mathrm{rand}-1,G_2=2\times(1-t/T)\tag{11}$$

式(9)中，$Q(t)$为均衡搜索策略的质量函数．G_1表示捕猎过程中用来跟踪猎物的各种运动．G_2表示天鹰跟随猎物从第一个位置到最后一个位置的飞行斜率．

3 改进AO算法

3.1 混沌初始化

Logistic映射[15]是一种典型的混沌映射，但因其混沌序列分布边缘化，使用该映射初始化种群会导致天鹰种群分布不均，降低搜索速度．而Tent混沌序列分布更具普遍性[16]，使得天鹰种群空间分布更加均匀，避免种群单一化，从而提升搜索效率．Tent混沌映射数学模型为

$$x_{i+1}=\begin{cases}2x_i,0\le x_i\le0.5\\2(1-x_i),0.5<x_i\le1\end{cases}\tag{12}$$

Tent混沌映射在迭代过程中存在不稳周期点，因

此在其基础上引入一个随机因子 $\text{rand}\times\frac{1}{N}\times\lambda$($\lambda$=0.005),可以有效避免不稳周期点. 改进后Tent混沌映射公式为

$$x_{i+1}=\begin{cases}2x_i+\text{rand}\times\dfrac{1}{N}\times\lambda, 0\le x_i\le 0.5\\ 2(1-x_i)+\text{rand}\times\dfrac{1}{N}\times\lambda, 0.5<x_i\le 1\end{cases}\tag{13}$$

利用式(13)映射到式(1)可得AO的混沌初始解$\boldsymbol{X}_i'$,如式(14)所示:

$$\begin{aligned}&\boldsymbol{X}_i'=\text{LB}_i^j+(\text{UB}_i^j-\text{LB}_i^j)\times x_i,\\&i=1,2,\cdots,N;\ j=1,2,\cdots,\text{Dim}\end{aligned}\tag{14}$$

式(14)中,$\boldsymbol{X}_i'$为混沌种群,x_i为Tent混沌序列.

3.2　折射反向学习

折射反向学习根据光的折射原理[17]计算反向解,扩大搜索范围,增加更优解的选取概率,通过比较当前解与其折射反向解,选择两者中的较优解进行迭代寻优,如图1所示.

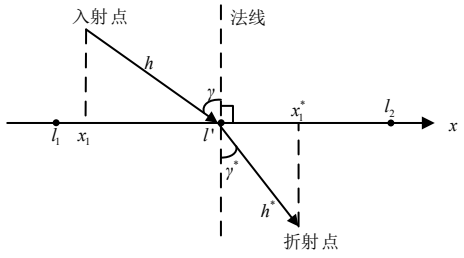


图1　折射反向学习原理图

其原理图1中,x轴上的寻优范围为$[l_1,l_2]$,x_1为当前个体位置,x_1^*为折射后位置,γ、γ^*分别表示入射角、折射角,h和h^*分别为入射、折射光线所对应的长度,l'为$[l_1,l_2]$的中点. 根据折射率可得:

$$n=\frac{\sin\gamma}{\sin\gamma^*}=\frac{(\frac{l_1+l_2}{2}-x_1)h^*}{(x_1^*-\frac{l_1+l_2}{2})h}\tag{15}$$

当n=1时,将式(15)拓展至Dim维空间后可得:

$$x_1^{j*}=\frac{l_1^j+l_2^j}{2}+\frac{l_1^j+l_2^j}{2m}+\frac{x_1^j}{m}\tag{16}$$

式(16)中,$m=h/h^*$,x_1^{j*}和x_1^j分别为x_1^*和x_1在第j维的分量($j=1,2,\cdots,\text{Dim}$),l_1^j、l_2^j分别为搜索空间内第j维的最小值和最大值.

3.3　种内互助及优化策略

在$\boldsymbol{X}_a$中,算法通过$(1-t/T)$来控制搜索空间,当迭代次数较大时,其值近似在(0,1),此时搜索趋于同一方向开发,种群逐渐被同化. 为解决此问题,将算子改进为$(1-2t/T)$,搜索范围可近似在(−1,1)内,可保证算法全方位开发,维持种群多样性;其次,个体都依赖于种群平均位置与最优位置进行更新,忽略了个体间位置的差异性,导致寻优具有一定盲目性,故对$\boldsymbol{X}_a$优化为

$$\begin{aligned}\boldsymbol{X}_a'(t+1)=&\boldsymbol{X}_{\text{best}}(t)\times(1-2t/T)\\&+[\frac{1}{N}\boldsymbol{X}_i(t)-\boldsymbol{X}_{\text{best}}(t)]\times\text{rand}\end{aligned}\tag{17}$$

式(17)充分发挥了个体位置信息,与原式(2)只根据种群平均位置和最优位置信息交流的方式相比,个体信息的引入更具针对性,故$\boldsymbol{X}_a'(t+1)$不再单一围绕种群进行迭代寻优,而是对前一时刻位置信息进行继承学习,使得个体均能被指引到猎物位置邻域,从而提高算法全局搜索性能.

在$\boldsymbol{X}_b$中,位置更新依托于精英个体和某一随机个体位置,缺乏对先前个体与其他个体位置关系的良好经验继承和学习机制,导致搜索效率降低. $\boldsymbol{X}_b$中随机解$\boldsymbol{X}_R$若为最优解,则容易导致算法陷入局部极值,寻优停滞. 为提高算法逃逸局部最优能力,提出一种种内互助策略对种群位置进行更新,如式(18)所示:

$$\boldsymbol{X}_b'(t+1)=\boldsymbol{X}_{\text{best}}(t)\times\text{Levy(Dim)}+(\boldsymbol{y}-\boldsymbol{x})\times\boldsymbol{H}\times R\tag{18}$$

式(18)中,$\boldsymbol{H}=\boldsymbol{X}_i(t)-\boldsymbol{X}_R'(t)$称为互助量,表示种群内第$i$只天鹰与其他个体的互助关系特征;$\boldsymbol{X}_R'(t)$表示除第$i$只天鹰外其余个体位置信息;$R$=rand,其作用解释如下:自然界中,同种生物个体之间通过分工协作捕获猎物以保证生存,当第i个体得到其余个体的所有信息量时,则可能获得足够的狩猎利益,但是种群内个体间位置信息复杂交错,因此个体间相互作用时可能只会获得部分利益并不一定能全部受益.

$\boldsymbol{X}_c$、$\boldsymbol{X}_d$在算法迭代后期进行开发,此时所有解已趋于最优解邻域范围内,$\boldsymbol{X}_c$利用UB_i^j与LB_i^j并不能对算法起到较好的收束作用,故对其优化,如式(19)所示. $\boldsymbol{X}_d$中$G_2\times\text{Levy(Dim)}$产生的步长大小不一,不利于算法集中开发,而$G_1\times\text{rand}$与$G_1\times\boldsymbol{X}(t)\times\text{rand}$因子重复,可合并以减少计算资源浪费,则$\boldsymbol{X}_d$优化后如式(20)所示.

$$\begin{aligned}\boldsymbol{X}_c'(t+1)=&(\boldsymbol{X}_{\text{best}}(t)-\boldsymbol{X}_M(t))\times\alpha\\&+((\text{ub}_i^j(t)-\text{lb}_i^j(t))\times\text{rand}+\text{lb}_i^j(t))\times\delta\end{aligned}\tag{19}$$

$$\boldsymbol{X}_d'(t+1)=Q(t)\times\boldsymbol{X}_{\text{best}}(t)-G_1\times\boldsymbol{X}_M(t)\times\text{rand}\tag{20}$$

式(19)中,$\text{ub}_i^j(t)$和$\text{lb}_i^j(t)$分别表示迭代到第t次时每个个体各维度上的最大、最小值,相比于UB_i^j与LB_i^j,其能够动态跟随种群位置变化,反应种群具体位置信息,使得算法具有较强开发能力. $\boldsymbol{X}_d'$相比于$\boldsymbol{X}_d$更容易集中开发且达到节约计算资源的目的.

3.4　自适应权重

为加快算法搜索速率,提出一种基于Bernoulli混沌序列扰动[18]的自适应权重[19]系数,公式为

$$w(t)=x_{t+1}\times\text{csch}(\frac{2\pi t}{T})\tag{21}$$

式(21)中,csch为双曲余割函数,x_{t+1}为Bernoulli序列,其定义如下:

$$x_{t+1}=\begin{cases}\dfrac{x_t}{1-\varphi}, 0<x_t\le 1-\varphi\\ \dfrac{x_t-1+\varphi}{\varphi}, 1-\varphi<x_t<1\end{cases} \tag{22}$$

式(22)中,φ取0.5.

采用自适应权重对全局搜索进行调节,公式为

$$\boldsymbol{X}_a^*=\boldsymbol{X}_a'\times w(t) \tag{23}$$

$$\boldsymbol{X}_b^*=\boldsymbol{X}_b'\times w(t) \tag{24}$$

式(23)、式(24)中,$\boldsymbol{X}_a'$,$\boldsymbol{X}_b'$为当前位置,$\boldsymbol{X}_a^*$、$\boldsymbol{X}_b^*$为自适应位置.

3.5 柯西-高斯变异

随着算法寻优推进,个体会被逐渐同化,使得算法陷入局部最优. 故引入柯西-高斯变异策略[20]以丰富种群多样性,减小陷入局部极值概率,提升算法全局寻优能力. 对当前最佳个体进行变异,选取变异前后较优个体参与位置更新. 变异算子如式(25)所示:

$$\begin{aligned}\text{Cauchy\_Gauss}(0,\sigma^2)=&[1-(\frac{t}{T})^2]\times\text{Cauchy}(0,\sigma^2)\\&+(\frac{t}{T})^2\times\text{Gauss}(0,\sigma^2)\end{aligned} \tag{25}$$

$$\sigma=\begin{cases}1, & f(\boldsymbol{X}_{\text{new}})<f(\boldsymbol{X}_i)\\ \exp(\frac{f(\boldsymbol{X}_{\text{new}})-f(\boldsymbol{X}_i)}{|f(\boldsymbol{X}_{\text{new}})|}), & f(\boldsymbol{X}_{\text{new}})\ge f(\boldsymbol{X}_i)\end{cases} \tag{26}$$

式(25)中,σ^2表示柯西-高斯变异的标准差,Cauchy$(0,\sigma^2)$和Gauss$(0,\sigma^2)$分别是服从柯西和高斯分布的随机因子,Cauchy_Gauss$(0,\sigma^2)$是具有柯西与高斯分布特点的因子. 通过柯西-高斯变异后,个体位置更新公式为

$$\boldsymbol{X}_{\text{best}}'=\boldsymbol{X}_{\text{best}}\times[\text{Cauchy\_Gauss}(0,\sigma^2)] \tag{27}$$

式(27)中,$\boldsymbol{X}_{\text{best}}$与$\boldsymbol{X}_{\text{best}}'$分别为变异前后个体位置.

3.6 MSIAO算法

本文将3.1~3.5节中的各改进策略用于优化基本AO算法后得到MSIAO算法,如算法1所示.

算法1 MSIAO算法

输入: 种群规模,问题维度,问题变量上界与下界,最大迭代次数及其他相关参数;

Step1: 利用融合Tent混沌映射的折射反向学习策略初始化种群,并记录种群搜索空间信息;

Step2: 计算每个个体适应度值,选出当前适应度值最优个体;

Step3: 根据式(27)对当前最佳位置进行柯西-高斯变异,并计算其适应度值,若变异个体适应度值优于原个体则取代原个体;否则,保留原个体信息;

Step4: 进入种群更新阶段,根据算法进程选取公式$\boldsymbol{X}_a^*$,$\boldsymbol{X}_b^*$,$\boldsymbol{X}_c'$,$\boldsymbol{X}_d'$进行迭代寻优,并利用式(21)更新自适应权重因子;

Step5: 计算更新后个体适应度值,并与更新前适应度值进行比较,选择较优个体保留下一次更新;

Step6: 若满足终止条件,停止算法,输出寻优结果;否则,返回Step3继续执行算法;

输出: 最佳位置.

3.7 MSIAO算法时间复杂度分析

在基本AO算法中,假设问题维度为Dim,种群大小为N,最大迭代次数为Max_Iter,则初始化阶段复杂度为$O(N\times\text{Dim})$,位置更新及适应度函数计算的时间复杂度为$O(N\times\text{Max_Iter}\times\text{Dim})+O(N\times\text{Max_Iter})$,AO算法时间复杂度为

$$\begin{aligned}f(T)&=O(N\times\text{Dim})+O(N\times\text{Max\_Iter}\times\text{Dim})\\&\quad+O(N\times\text{Max\_Iter})\\&=O(N\times\text{Max\_Iter}\times\text{Dim})\end{aligned}$$

在MSIAO算法中,假设引入Tent混沌序列所需时间为t_1,按照折射反向学习机制生成反向种群所需时间为t_2,则初始种群阶段的时间复杂度为

$$f(T_1)=O(N\times\text{Dim}+t_1+t_2)=O(N\times\text{Dim})$$

优化$\boldsymbol{X}_a$、$\boldsymbol{X}_b$、$\boldsymbol{X}_c$、$\boldsymbol{X}_d$过程中仅对搜索机制进行改进,并未引入其余算子,故在种内互助及优化阶段位置更新及适应度函数计算的时间复杂度为

$$\begin{aligned}f(T_2)&=O(N\times\text{Max\_Iter}\times\text{Dim})+O(N\times\text{Max\_Iter})\\&=O(N\times\text{Max\_Iter}\times\text{Dim})\end{aligned}$$

在$f(T_2)$基础上,假设生成Bernoulli混沌权重所需的时间为t_3,则引入Bernoulli混沌权重后位置更新公式及适应度函数计算的时间复杂度为

$$\begin{aligned}f(T_3)&=O(N\times\text{Max\_Iter}\times\text{Dim}+t_3)+O(N\times\text{Max\_Iter})\\&=O(N\times\text{Max\_Iter}\times\text{Dim})\end{aligned}$$

假设生成柯西-高斯变异算子所需时间为t_4,则结合此变异的最优个体位置更新和适应度函数计算的时间杂度为

$$f(T_4)=O(\text{Max\_Iter}\times\text{Dim}+t_4)=O(\text{Max\_Iter}\times\text{Dim})$$

则MSIAO算法的时间复杂度为

$$f(T')=f(T_1)+f(T_3)+f(T_4)=O(N\times\text{Max\_Iter}\times\text{Dim})$$

综上所述,MSIAO与标准AO算法的时间复杂度一致.

4 仿真实验与结果分析

4.1 基准测试函数

为验证MSIAO算法性能,选取10个基准测试函数[21]进行寻优,如表1所示. f_1~f_5为单峰函数(30维),f_6~f_7为多峰函数(30维),f_8~f_{10}为固定维多模态函数. 测试环境:Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz (4 CPUs), ~2.7 GHz,8 GB内存,64 bit操作系统,Matlab2017(b).

4.2 改进策略有效性分析

为分析不同改进策略对AO性能的影响,将标准AO、改进种群策略的AO(Aquila Optimization of Improved Population, IPAO)、基于种内互助及优化策略的AO(Aquila Optimization based on Intraspecific Mutual Assistance, IMAAO)、自适应权重策略的AO(Adaptive

表1　测试函数

序号	基准测试函数	误差精度μ
f_1	Sphere	1.00E-3
f_2	Schwefel′problem 2.22	1.00E-3
f_3	Schwefel′problem 1.2	1.00E-3
f_4	Schwefel′problem 2.21	1.00E-3
f_5	Generalized Rosenbrock′s Function	1.00E-2
f_6	Generalized Schwefel′s problem 2.26	1.00E+2
f_7	Ackley′s Function	1.00E-2
f_8	Shekell′s Foxholes Function	1.00E-2
f_9	Shekel′s Family 3	1.00E-2
f_{10}	Hatman′s Function 2	1.00E-2

Weight Aquila Optimization,AWAO)、结合柯西-高斯变异策略的AO(Cauchy-Gauss Aquila Optimization,CGAO)和MSIAO对表1中的测试函数进行寻优,均独立运行30次,实验结果如表2所示. 各算法参数与标准AO保持一致,迭代次数T=500,种群数目N=30.

由表2可知:IPAO在对大部分函数寻优时,通过其最优值与最差值可看出,IPAO具备更加广泛的寻优结果,表明改进初始种后,算法在有限寻优条件下能够更充分挖掘搜索空间. IMAAO在对函数进行寻优时,其收敛质量相比于AO有着不同程度的提升. 对f_1寻优时,IMAAO能够搜索到最优解,对f_2、f_3、f_4的最优值和均值均有显著提升,对于f_5的指标略有提高. IMAAO对多峰函数求解的最优值和平均值都有着更高的精度. 由此表明,基于种内互助及优化策略改进的算法能够有效提高局部开发能力,改善全局寻优性能. AWAO在对单峰及多峰函数寻优过程中,通过其最优值与均值可知,自适应权重策略能够有效平衡算法的全局和局部搜索性能. 通过CGAO的寻优最差值可看出其相比于AO的

表2　不同改进策略实验结果

函数	算法	最优值	最差值	平均值	标准差	函数	算法	最优值	最差值	平均值	标准差
f_1	AO	1.88E-142	2.16E-132	7.83E-134	3.95E-133	f_6	AO	-1.25E+04	-3.51E+03	-6.45E+03	2.09E+03
	IPAO	2.53E-156	1.18E-130	5.40E-132	2.28E-131		IPAO	-1.26E+04	-3.68E+03	-9.32E+03	2.38E+03
	IMAAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		IMAAO	-1.26E+04	-1.13E+03	-1.08E+04	1.59E+03
	AWAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		AWAO	-1.26E+04	-5.36E+03	-9.70E+03	1.80E+03
	CGAO	2.83E-211	5.49E-166	1.92E-167	0.00E+00		CGAO	-1.18E+04	-3.73E+03	-8.12E+03	2.69E+03
	MSIAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		MSIAO	-1.26E+04	-8.09E+03	-1.24E+04	9.27E+02
f_2	AO	4.30E-72	6.23E-67	4.10E-68	1.19E-67	f_7	AO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	IPAO	5.62E-81	2.99E-58	9.96E-60	5.46E-59		IPAO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	IMAAO	2.17E-196	6.09E-190	1.24E-170	0.00E+00		IMAAO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	AWAO	2.76E-273	3.81E-253	1.29E-254	0.00E+00		AWAO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	CGAO	7.10E-107	5.09E-84	3.54E-85	1.16E-84		CGAO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	MSIAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		MSIAO	8.88E-16	8.88E-16	8.88E-16	0.00E+00
f_3	AO	2.83E-139	1.08E-127	3.94E-129	1.97E-128	f_8	AO	9.98E-01	1.27E+01	3.87E+00	4.24E+00
	IPAO	3.59E-155	1.00E-97	3.34E-99	1.83E-98		IPAO	9.98E-01	1.27E+01	3.19E+00	3.61E+00
	IMAAO	6.84.E-308	8.26E-301	9.90E-290	0.00E+00		IMAAO	9.98E-01	6.84E+00	2.89E+00	2.81E+00
	AWAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		AWAO	9.98E-01	5.93E+00	2.15E+00	1.19E+00
	CGAO	5.77E-206	2.69E-156	8.97E-158	4.91E-157		CGAO	9.98E-01	1.27E+01	3.55E+00	3.62E+00
	MSIAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		MSIAO	9.98E-01	9.98E-01	9.98E-01	9.59E-13
f_4	AO	1.11E-71	1.58E-66	1.19E-67	3.03E-67	f_9	AO	-1.05E+01	-5.10E+00	-8.92E+00	2.34E+00
	IPAO	2.54E-87	7.28E-56	2.43E-57	1.33E-56		IPAO	-1.05E+01	-1.04E+01	-1.05E+01	4.43E-02
	IMAAO	3.98.E-298	3.64E-292	7.85E-280	4.30E-138		IMAAO	-1.05E+01	-1.05E+01	-1.05E+01	2.92E-02
	AWAO	2.65E-276	1.94E-249	6.48E-251	0.00E+00		AWAO	-1.05E+01	-5.07E+00	-8.24E+00	2.60E+00
	CGAO	4.32E-100	2.96E-83	1.45E-84	5.50E-84		CGAO	-1.05E+01	-9.72E+00	-1.03E+01	1.87E-01
	MSIAO	0.00E+00	0.00E+00	0.00E+00	0.00E+00		MSIAO	-1.05E+01	-1.03E+01	-1.05E+01	2.16E-02
f_5	AO	4.69E-05	1.44E-01	1.02E-02	2.61E-02	f_{10}	AO	-3.28E+00	-2.84E+00	-3.12E+00	9.03E-02
	IPAO	8.97E-06	6.45E-02	5.25E-03	1.23E-02		IPAO	-3.31E+00	-2.98E+00	-3.14E+00	9.18E-02
	IMAAO	3.34E-06	5.26E-04	6.55E-03	1.14E-02		IMAAO	-3.08E+00	-3.01E+00	-2.92E+00	1.17E-01
	AWAO	1.03E-05	4.04E-02	8.56E-03	1.15E-02		AWAO	-3.31E+00	-2.95E+00	-3.16E+00	8.65E-02
	CGAO	1.82E-04	1.38E-01	1.32E-02	2.78E-02		CGAO	-3.32E+00	-3.01E+00	-3.15E+00	6.99E-02
	MSIAO	3.04E-07	4.60E-03	6.53E-05	1.17E-03		MSIAO	-3.32E+00	-2.85E+00	-3.20E+00	7.24E-02

寻优精度有着不同数量级的提升，说明引入柯西-高斯变异策略能够有效逃逸局部极值. MSIAO的最优值、平均值以及标准差都优于单一改进策略的AO，寻优性能显著，表明MSIAO能够有效嵌合各改进策略的优点.

4.3 算法性能对比分析

设k为实验运行次数，$F(k)$为测试函数实际寻优结果，F^*为理论最优值，则适应度误差[21]如式(28)所示：

$$E(k)=F(k)-F^*,k=1,2,\cdots,30 \tag{28}$$

且定义成功标记变量为

$$C(k)=\begin{cases}1, & |E(k)|<\mu\\ 0, & |E(k)|\geq\mu\end{cases} \tag{29}$$

式(29)中，误差精度μ值见表1. 则算法寻优成功率P_s为

$$P_s=\frac{1}{30}\sum_{k=1}^{30}C(k) \tag{30}$$

将GWO[6]、SSA[7]、WOA[8]、PSO[9]、标准AO[1]及MSIAO等算法做测试函数寻优对比. 通过30次独立实验后，各算法的寻优结果如表3所示.

由表3可知：MSIAO在10组测试函数中的寻优能力和收敛精度明显优于其他5种对比算法. 对于高维单峰函数，MSIAO在求解f_1~f_4时，都能搜索到最优值，相对于AO寻优精度显著提高；在f_5上，MSIAO的求解精度、平均值和标准差相比其余算法表现更为优异. 对于高维多峰函数，在f_6中，MSIAO与WOA都能寻到最优值，但MSIAO标准差更小，其稳定性更优，相比于AO，MSIAO寻优性能明显提升；对于f_7，AO与MSIAO寻优结果相当，其余算法稍逊. 对于低维多模态函数，各算法搜索结果相差不多，但从标准

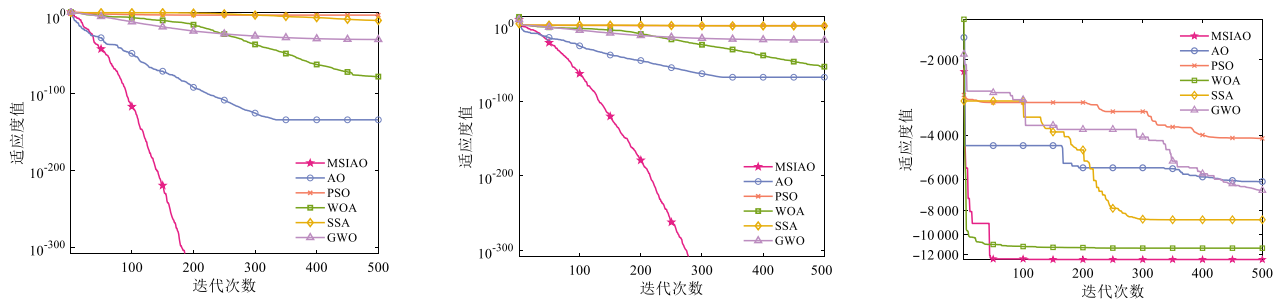
表3 各算法优化基准函数结果对比

函数	算法	最优值	平均值	标准差	时间	成功率	函数	算法	最优值	平均值	标准差	时间	成功率
f_1	GWO	7.46E−33	9.96E−31	1.38E−30	0.14	100%	f_6	GWO	−7.77E+03	−5.98E+03	1.53E+03	0.17	0
	SSA	2.99E−08	2.06E−07	4.37E−07	0.10	100%		SSA	−9.17E+03	−7.56E+03	7.40E+02	0.12	0
	WOA	1.53E−88	4.62E−67	2.53E−66	0.09	100%		WOA	−1.26E+04	−1.22E+04	7.02E+02	0.10	50%
	PSO	1.86E+00	7.62E+00	2.90E+00	0.11	0		PSO	−5.67E+03	−3.37E+03	6.15E+02	0.14	0
	AO	4.07E−142	2.35E−133	9.09E−133	0.16	100%		AO	−1.25E+04	−5.84E+03	1.57E+03	0.20	26.67%
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.16	100%		MSIAO	**−1.26E+04**	**−1.25E+04**	**1.68E+02**	0.22	100%
f_2	GWO	1.60E−19	2.17E−18	3.42E−18	0.14	100%	f_7	GWO	1.51E−14	2.14E−14	3.81E−15	0.15	100%
	SSA	2.50E−01	1.90E+00	1.22E+00	0.11	0		SSA	1.78E+00	2.60E+00	6.19E−01	0.12	0
	WOA	2.59E−59	7.28E−51	3.90E−50	0.09	100%		WOA	8.88E−16	4.09E−15	2.16E−15	0.12	100%
	PSO	6.25E+00	1.44E+01	6.01E+00	0.11	0		PSO	2.87E+00	4.58E+00	9.56E−01	0.12	0
	AO	1.87E−71	5.11E−68	1.25E−67	0.16	100%		AO	8.88E−16	8.88E−16	0.00E+00	0.18	100%
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.15	100%		MSIAO	**8.88E−16**	**8.88E−16**	**0.00E+00**	0.20	100%
f_3	GWO	6.78E−06	4.25E−03	1.04E−02	0.32	43.33%	f_8	GWO	9.98E−01	2.14E+00	2.60E+00	0.67	73.33%
	SSA	3.71E+02	1.41E+03	9.57E+02	0.30	0		SSA	9.98E−01	1.23E+00	5.64E−01	0.61	70%
	WOA	2.53E+04	4.98E+04	1.14E+04	0.23	0		WOA	9.98E−01	2.73E+00	3.03E+00	0.60	63.33%
	PSO	1.95E+02	5.26E+02	2.59E+02	0.29	0		PSO	9.98E−01	1.68E+00	1.35E+00	0.67	80%
	AO	3.93E−141	3.17E−124	1.73E−123	0.34	100%		AO	9.98E−01	2.47E+00	2.67E+00	0.82	70%
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.36	100%		MSIAO	**9.98E−01**	**9.98E−01**	**5.36E−16**	0.84	100%
f_4	GWO	3.14E−06	4.26E−05	7.57E−05	0.14	100%	f_9	GWO	−1.05E+01	−1.05E+01	**1.84E−04**	0.11	90%
	SSA	5.53E+00	1.24E+01	4.06E+00	0.10	0		SSA	−1.05E+01	−8.75E+00	2.83E+00	0.72	76.67%
	WOA	1.61E+01	5.65E+01	2.55E+01	0.08	0		WOA	−1.05E+01	−8.22E+00	3.60E+00	0.13	76.67%
	PSO	3.77E+00	6.50E+00	1.44E+00	0.11	0		PSO	−1.05E+01	−5.24E+00	3.47E+00	0.14	73.33%
	AO	5.82E−71	9.07E−68	2.50E−67	0.16	100%		AO	−1.05E+01	−8.61E+00	2.51E+00	0.19	76.67%
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.17	100%		MSIAO	**−1.05E+01**	**−1.05E+01**	8.69E−02	0.21	100%
f_5	GWO	2.59E+01	2.66E+01	7.91E−01	0.17	0	f_{10}	GWO	−3.32E+00	**−3.25E+00**	7.24E−02	0.09	90%
	SSA	1.68E+01	2.38E+02	4.90E+02	0.13	0		SSA	−3.32E+00	−3.23E+00	**5.96E−02**	0.09	86.67%
	WOA	6.57E+00	2.72E+01	3.91E+00	0.12	0		WOA	−3.32E+00	−3.24E+00	1.65E−01	0.08	86.67%
	PSO	3.60E+02	1.12E+03	1.03E+03	0.13	0		PSO	−3.32E+00	−3.21E+00	1.25E−01	0.10	86.67%
	AO	3.28E−05	4.52E−03	5.88E−03	0.22	80%		AO	−3.30E+00	−3.13E+00	9.04E−02	0.14	83.33%
	MSIAO	**1.83E−07**	**8.32E−05**	**1.89E−04**	0.24	93.33%		MSIAO	**−3.32E+00**	−3.19E+00	7.66E−02	0.15	86.67%

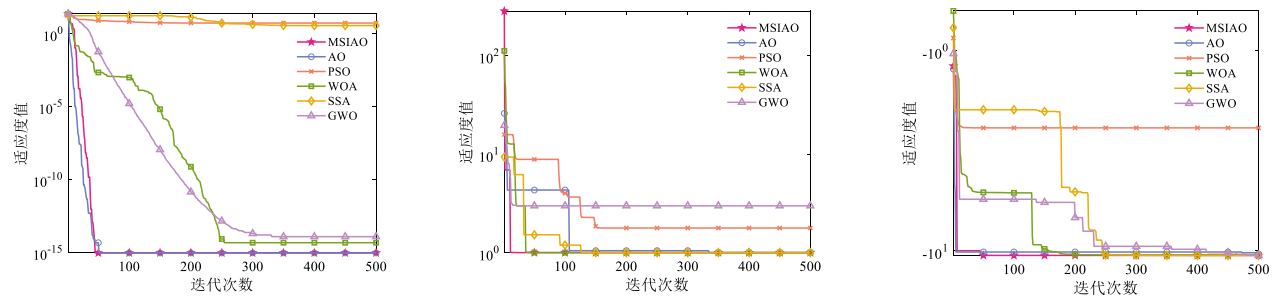
差可看出,MSIAO 寻优稳定性更高. 由上述可知,MSIAO 寻优精度与稳定性相较对比算法均有显著优势.

MSIAO 寻优时间与 AO 基本一致,但 MSIAO 在 8 个测试函数上的寻优成功率达到 100%. 其余算法虽然在寻优时间上具有一定优势,但寻优成功率参差不齐. 综上所述,MSIAO 稳定性较高,寻优性能显著,表明 MSIAO 相较对比算法具有较强竞争力.

为反映 MSIAO 与其余 5 种算法的动态收敛特性,图 2 给出了 6 个基准函数的收敛曲线. 对于单峰函数,MSIAO 相比其他算法优势显著,其收敛曲线在迭代过程中并未出现停滞情况,且均搜索到了最优值,而其余算法均出现寻优停滞情况,且收敛精度不高. 对于多峰高维函数,MSIAO 在 f_6 上的寻优表现最为突出,相较于其余算法具有更高的收敛精度;在 f_7 上,MSIAO 与 AO 基本保持一样的寻优效率,但是与对比算法相比仍然表现出明显的优势. 对于低维多模态函数 f_8、f_9,MSIAO 也具有较高收敛精度与较快收敛速度.



(a) f_1 函数收敛曲线　(b) f_2 函数收敛曲线　(c) f_6 函数收敛曲线



(d) f_7 函数收敛曲线　(e) f_8 函数收敛曲线　(f) f_9 函数收敛曲线

图 2　6 种算法收敛曲线对比

4.4　MSIAO 与较新改进算法性能对比

为更进一步验证 MSIAO 算法的优越性,选取部分较新改进算法:变异驱动改进的灰狼优化算法(Mutation-Driven Grey Wolf Optimization, MDGWO)[6]、三学习策略粒子群算法(Three-Learning Strategy Particle Swarm Algorithm, TLSPSO)[9]、自适应混沌天鹰优化算法(Adaptive Chaos Aquila Optimization, ACAO)[10]、混合天鹰优化和哈里斯鹰优化的改进算法(Improved Hybrid Aquila Optimization and Harris Hawks Optimization, IHAOHHO)[13]、改进的黏菌优化算法(Improved Slime Mold Algorithm, ISMA)[22]与之进行对比,实验结果如表 4 所示.

由表 4 可知,MSIAO 寻优性能表现最为出色,ISMA 次之,TLSPSO 稍逊. 对于单峰函数 f_1~f_4,MSIAO 均能寻到最优值,且较为稳定;对于 f_5,MSIAO 寻优结果依然能保持较高收敛精度,表明其具有较强的局部开发能力. 对于多峰函数,MSIAO 也表现出较为优异的性能,并且通过标准差可看出,MSIAO 具有更强的稳定性,表明 MSIAO 在全局搜索方面也具备一定优势. 由上述可知,MSIAO 算法与较新改进优化算法相比也表现出较强竞争力.

4.5　CEC2014 测试函数实验分析

为了更加全面地评估 MSIAO 的稳定性与有效性,在 CEC2014 中选取部分单峰函数(Unimodal Functions, UN)、多峰函数(Multimodal Functions, MF)、混合函数(Hybrid Functions, HF)和复合函数(Composition Functions, CF)进行优化求解,各函数信息如表 5 所示. 设种群规模为 50,最大迭代次数为 2 000,维度为 30. 各算法

表4 MSIAO与较新改进算法寻优结果对比

函数	算法	最优值	平均值	标准差	函数	算法	最优值	平均值	标准差
f_1	TLSPSO	9.86E-24	5.27E-19	8.64E-17	f_6	TLSPSO	-4.98E+03	-4.12E+03	4.35E+02
	MDGWO	6.37E-280	2.91E-272	0.00E+00		MDGWO	-7.59E+03	-6.53E+03	5.10E+02
	ISMA	0.00E+00	7.42E-117	4.06E-116		ISMA	-1.26E+04	-1.24E+04	3.52E+02
	ACAO	1.53E-270	1.65E-246	0.00E+00		ACAO	-1.26E+04	-1.25E+04	8.80E+02
	IHAOHHO	0.00E+00	0.00E+00	0.00E+00		IHAOHHO	-5.66E+03	-4.64E+03	3.43E+02
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**		MSIAO	**-1.26E+04**	**-1.25E+04**	**1.68E+02**
f_2	TLSPSO	8.61E-12	5.82E-10	6.49E-08	f_7	TLSPSO	6.86E-03	3.95E-02	8.53E-02
	MDGWO	8.62E-153	3.68E-151	3.65E-156		MDGWO	4.44E-15	7.88E-15	1.14E-15
	ISMA	1.62E-203	1.38E-54	7.51E-54		ISMA	8.88E-16	8.88E-16	0.00E+00
	ACAO	1.34E-135	5.89E-121	3.23E-120		ACAO	8.88E-16	8.88E-16	0.00E+00
	IHAOHHO	1.72E-173	5.09E-168	0.00E+00		IHAOHHO	8.88E-16	8.88E-16	0.00E+00
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**		MSIAO	**8.88E-16**	**8.88E-16**	**0.00E+00**
f_3	TLSPSO	3.59E-29	9.16E-27	7.65E-25	f_8	TLSPSO	9.98E-01	9.98E-01	5.69E-02
	MDGWO	2.06E-276	4.01E-270	0.00E+00		MDGWO	9.98E-01	9.98E-01	**1.13E-16**
	ISMA	0.00E+00	1.96E-90	1.08E-89		ISMA	9.98E-01	9.98E-01	5.11E-16
	ACAO	1.09E-247	5.56E-219	0.00E+00		ACAO	9.98E-01	2.18E+00	2.19E+00
	IHAOHHO	7.76E-322	4.98E-310	0.00E+00		IHAOHHO	9.98E-01	9.98E-01	1.46E-03
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**		MSIAO	**9.98E-01**	**9.98E-01**	5.36E-16
f_4	TLSPSO	5.29E-16	4.89E-13	5.79E-11	f_9	TLSPSO	-1.05E+01	-6.01E+00	5.49E-01
	MDGWO	5.44E-148	1.98E-146	3.18E-141		MDGWO	-1.05E+01	-1.05E+01	1.95E-01
	ISMA	2.48E-209	5.84E-53	3.09E-52		ISMA	-1.05E+01	-1.05E+01	1.82E-01
	ACAO	6.05E-129	3.50E-118	1.92E-117		ACAO	-1.04E+01	-6.66E+00	2.23E+00
	IHAOHHO	1.87E-164	2.14E-159	4.47E-159		IHAOHHO	-9.66E+00	-5.84E+00	1.44E+00
	MSIAO	**0.00E+00**	**0.00E+00**	**0.00E+00**		MSIAO	**-1.05E+01**	**-1.05E+01**	**8.69E-02**
f_5	TLSPSO	9.69E-01	1.08E+00	1.01E+00	f_{10}	TLSPSO	-3.32E+00	-3.21E+00	8.55E-02
	MDGWO	2.56E+00	2.59E+00	1.50E+00		MDGWO	-3.32E+00	**-3.29E+00**	**5.19E-02**
	ISMA	6.55E-03	4.38E+00	8.96E+00		ISMA	-3.32E+00	-3.26E+00	5.35E-02
	ACAO	1.94E-06	3.93E-03	9.12E-03		ACAO	-3.29E+00	-3.08E+00	1.09E-01
	IHAOHHO	2.74E+01	2.80E+01	4.11E-01		IHAOHHO	-3.19E+00	-3.07E+00	8.04E-02
	MSIAO	**1.83E-07**	**8.32E-05**	**1.89E-04**		MSIAO	**-3.32E+00**	-3.19E+00	7.66E-02

均独立运行30次,结果如表6所示.

表5 CEC2014函数

函数	维度	特征	定义域	最佳值
CEC03	30	UN	[-100,100]	300
CEC05	30	MF	[-100,100]	500
CEC10	30	MF	[-100,100]	1 000
CEC15	30	MF	[-100,100]	1 500
CEC19	30	HF	[-100,100]	1 900
CEC22	30	HF	[-100,100]	2 200
CEC25	30	CF	[-100,100]	2 500
CEC28	30	CF	[-100,100]	2 800

由表6可知,在多峰、混合、复合函数上MSIAO优势明显,对于多峰函数:在CEC05函数和CEC15函数上,MSIAO与其他算法都收敛到理论值附近,但是MSIAO标准差最小,表明MSIAO具有更好的稳定性;在CEC10函数寻优中,MSIAO收敛值最接近理论值,其他算法性能表现相对较差. 对于混合函数:在CEC19函数上,各算法都能收敛到最优值附近,但是MSIAO具有更优的稳定性;在CEC22函数中,MSIAO寻优结果更接近理论值. 对于复合函数:在CEC25、CEC28函数寻优中,MSIAO均能稳定地收敛到最优值附近. 综上分析,MSIAO具有更强的鲁棒性.

5 MSIAO算法机械优化设计应用

为进一步验证MSIAO的实用性与可行性,本文利用AO[1]、GWO[6]、SSA[7]、WOA[8]、PSO[9]、黑猩猩优化算法(Chimp Optimization Algorithm, ChOA)[21]、黏菌优化算法(Slime Mold Algorithm, SMA)[22]、正弦余弦算法(Sine Cosine Algorithm, SCA)[23]与MSIAO对压力容器

表6　CEC2014函数优化结果对比

函数	指标	CEC03	CEC05	CEC10	CEC15	CEC19	CEC22	CEC25	CEC28
GWO	平均值	9.98E+03	5.21E+02	3.50E+03	1.52E+03	1.92E+03	2.52E+03	2.71E+03	3.78E+03
	标准差	4.39E+03	4.07E−02	1.55E+03	2.57E+01	1.69E+01	1.86E+02	2.51E+00	1.32E+02
SSA	平均值	2.41E+04	5.20E+02	4.90E+03	1.51E+03	1.92E+03	2.73E+03	2.71E+03	3.93E+03
	标准差	6.46E+03	8.30E−02	6.40E+02	2.95E+00	1.16E+01	2.18E+02	3.75E+00	2.29E+02
WOA	平均值	6.57E+04	5.20E+02	5.42E+03	1.59E+03	1.96E+03	3.06E+03	2.72E+03	5.07E+03
	标准差	3.74E+04	1.64E−01	7.36E+02	2.45E+01	4.44E+01	2.16E+02	1.59E+01	5.83E+02
PSO	平均值	3.55E+03	5.21E+02	1.73E+03	1.51E+03	1.92E+03	2.49E+03	2.71E+03	4.27E+03
	标准差	3.00E+03	6.49E−02	2.87E+02	2.81E+00	2.09E+01	1.67E+02	2.13E+00	4.99E+02
AO	平均值	3.90E+04	5.21E+02	3.81E+03	1.53E+03	1.92E+03	2.90E+03	2.70E+03	3.00E+03
	标准差	8.17E+03	1.30E−01	7.34E+02	6.65E+00	2.25E+01	1.74E+02	0.00E+00	0.00E+00
MSIAO	平均值	3.69E+03	5.06E+02	2.83E+03	1.51E+03	1.91E+03	2.65E+03	2.66E+03	2.91E+03
	标准差	8.56E+02	3.59E−02	2.75E+02	2.63E+00	1.03E+01	1.96E+02	0.00E+00	0.00E+00

和焊接梁进行机械优化设计实验对比.

5.1　压力容器优化设计问题

压力容器优化设计问题[22]的目的是降低建造成本,难点在于确定壳体厚度x_1、封头厚度x_2、内半径x_3和圆柱截面长度x_4的尺寸. 问题约束条件如式(31)所示,其目标函数如式(32)所示.

各算法分别独立求解30次,寻优结果如表7所示. MSIAO对于压力容器设计问题的优化效果明显优于其他算法,其总体造价相比于排名第二的AO,开销平均值节约了4.62%.

表7　压力容器设计问题结果对比

算法	变量最优值				开销平均值	时间/s
	x_1	x_2	x_3	x_4		
PSO	0.883 3	0.456 2	45.962 3	170.632 1	7 763.352 3	0.17
WOA	0.833 0	0.561 3	45.617 5	178.659 4	7 860.278 8	0.14
SSA	0.879 9	0.434 9	45.591 7	140.196 4	7 588.192 4	0.14
GWO	0.839 2	0.405 3	45.637 1	179.635 4	7 296.534 1	0.15
SCA	1.156 3	0.685 9	45.621 7	82.617 2	7 315.392 6	0.13
SMA	0.921 6	0.763 5	49.621 7	98.524 3	7 529.687 4	0.15
ChOA	1.216 2	0.632 9	62.259 3	25.447 5	8 582.465 5	0.20
AO	1.153 4	0.369 2	48.687 1	96.869 9	7 140.947 3	0.14
MSIAO	0.862 9	0.509 1	42.642 7	162.9673	6 825.356 4	0.16

$$\text{s.t. } g_1(\boldsymbol{X})=-x_1+0.0193x_3\le 0;$$
$$g_2(\boldsymbol{X})=-x_2+0.00954x_3\le 0;$$
$$g_3(\boldsymbol{X})=-\pi x_3^2x_4-\frac{4}{3}\pi x_3^2+1296000\le 0;$$
$$g_4(\boldsymbol{X})=-x_4-240\le 0;$$
$$0\le x_1,x_2\le 99,10\le x_3,x_4\le 200 \tag{31}$$
$$\min f(\boldsymbol{X})=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1661x_4x_1^2+19.84x_3x_1^2 \tag{32}$$

5.2　焊接梁优化设计问题

焊接梁优化设计问题[23]目的是节约焊接材料,减少开销. 焊接梁设计问题数学模型描述如式(33)~(39)所示,其中,y_1、y_2、y_3和y_4分别表示焊接梁的焊缝宽度、横梁的长度、高度和厚度,受剪切应力τ、横梁弯曲应力η、屈曲载荷P_c、横梁挠度ψ以及其他内部参数约束.

$$\tau(\boldsymbol{Y})=\sqrt{(\tau')^2+2\tau'\tau''\frac{y_2}{2R}+(\tau'')^2},\quad \tau'=\frac{p}{\sqrt{2}\,y_1y_2},\tau''=\frac{MR}{J} \tag{33}$$

$$P_c(\boldsymbol{Y})=\frac{4.013E\sqrt{\frac{y_3^2y_4^6}{36}}}{L^2}\left(1-\frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{34}$$

$$\eta(\boldsymbol{Y})=\frac{6PL}{y_4y_3^2},\psi(\boldsymbol{Y})=\frac{4PL^3}{Ey_3^2y_4} \tag{35}$$

$$M=P(L+\frac{y_2}{2}),R=\sqrt{\frac{y_2^2}{4}+(\frac{y_1+y_3}{2})^2} \tag{36}$$

$$J=2\left\{\sqrt{2}\,y_1y_2\left[\frac{y_2^2}{4}+(\frac{y_1+y_3}{2})^2\right]\right\} \tag{37}$$

$$L=14\text{ in},\psi_{\max}=0.25\text{ in},E=30\times10^6\text{ psi},G=12\times10^6\text{ psi},\tau_{\max}=1.3\times10^4\text{ psi},\eta_{\max}=3\times10^4\text{ psi},P=6\times10^3\text{1 b} \tag{38}$$

$$\text{s.t. } g_1'(\boldsymbol{Y})=\tau(\boldsymbol{Y})-\tau_{\max}\le 0;$$
$$g_2'(\boldsymbol{Y})=\psi(\boldsymbol{Y})-\psi_{\max}\le 0,g_3'(\boldsymbol{Y})=\eta(\boldsymbol{Y})-\eta_{\max}\le 0;$$
$$g_4'(\boldsymbol{Y})=y_1-y_4\le 0,g_5'(\boldsymbol{Y})=P-P_c(\boldsymbol{Y})\le 0;$$
$$g_6'(\boldsymbol{Y})=0.125-y_1\le 0;$$
$$g_7'(\boldsymbol{Y})=0.10471y_1^2+0.04811y_3y_4(14+y_2)-5\le 0;$$
$$0.1\le y_1,y_4\le 2,0.1\le y_2,y_3\le 10.$$
$$\min f(\boldsymbol{Y})=0.10471y_1^2y_2+0.04811y_3y_4(14+y_2) \tag{39}$$

各算法分别独立求解30次,寻优结果如表8所示. 从其开销最小值和平均值可看出,MSIAO对于焊接梁设计问题优化效果明显优于其他算法,与排名第二的AO对比,开销平均值节约了0.77%.

表8 焊接梁设计问题结果对比

算法	变量最优值				开销平均值	时间/s
	y_1	y_2	y_3	y_4		
PSO	0.205 9	2.419 3	9.336 2	0.229 7	1.816 9	0.21
WOA	0.265 3	1.792 6	7.957 9	0.265 3	1.748 9	0.18
SSA	0.173 5	3.533 4	9.209 5	0.210 5	1.766 3	0.20
GWO	0.125 0	4.536 8	9.062 4	0.205 6	1.745 6	0.20
SCA	0.157 0	4.085 0	9.099 3	0.210 8	1.796 8	0.27
SMA	0.125 0	4.516 6	9.036 6	0.205 7	1.749 2	0.25
ChOA	0.157 7	4.105 5	8.879 1	0.215 3	1.782 6	0.62
AO	0.155 9	3.532 8	8.764 6	0.218 7	1.721 6	0.22
MSIAO	0.150 7	3.256 1	8.631 1	0.226 2	1.708 5	0.22

综上对两个实际工程优化设计案例分析可知，MSIAO在处理工程优化问题时也具备一定优势，进一步体现了MSIAO的适用性.

6 结论

本文通过有机融合Tent映射与反向学习、柯西-高斯变异、Bernoulli混沌权重及种内互助机制策略对AO进行改进，实验结果表明，改进后的AO减小了算法陷入局部极值的概率，具备更高的寻优精度和更快的收敛速率，对于压力容器与焊接梁的优化设计问题具有实际工程应用价值. 在后续研究中，可进一步优化寻优机制，在多目标、离散多约束等工程优化问题方面拓展其实际应用领域.

参考文献

[1] ABUALIGAH L, YOUSRI D, ELAZIZ M ABD, et al. Aquila optimizer: A novel meta-heuristic optimization algorithm[J]. Computers & Industrial Engineering, 2021, 157: 107250.

[2] ALRASSAS A M, AL-QANESS M A A, EWEES A A, et al. Optimized ANFIS model using aquila optimizer for oil production forecasting[J]. Processes, 2021, 9(7): 1194.

[3] KANDAN M, KRISHNAMURTHY A, SELVI S, et al. Quasi oppositional aquila optimizer-based task scheduling approach in an IoT enabled cloud environment[J]. The Journal of Supercomputing, 2022, 78(7): 10176-10190.

[4] WANG S, JIA H, ABUALIGAH L, et al. An improved hybrid aquila optimizer and Harris hawks algorithm for solving industrial engineering optimization problems[J]. Processes, 2021, 9(9): 1551.

[5] AL-QANESS M A A, EWEES A A, FAN H, et al. Modified aquila optimizer for forecasting oil production[J/OL]. Geo-Spatial Information Science. DOI: 10.1080/10095020.2022. 2068385.

[6] SINGH S, BANSAL J C. Mutation-driven grey wolf optimizer with modified search mechanism[J]. Expert Systems with Applications, 2022, 194: 116450.

[7] BAIRATHI D, GOPALANI D. An improved salp swarm algorithm for complex multi-modal problems[J]. Soft Computing, 2021, 25(15): 10441-10465.

[8] CHAKRABORTY S, SAHA A K, CHAKRABORTY R, et al. An enhanced whale optimization algorithm for large scale optimization problems[J]. Knowledge-Based Systems, 2021, 233: 107543.

[9] ZHANG X, LIN Q. Three-learning strategy particle swarm algorithm for global optimization problems[J]. Information Sciences, 2022, 593: 289-313.

[10] LI X, MOBAYEN S. Optimal design of a PEMFC-based combined cooling, heating and power system based on an improved version of aquila optimizer[J]. Concurrency and Computation-Practice & Experience, 2022, 34(15): e6976.

[11] WANG S, MA J, LI W, et al. An optimal configuration for hybrid SOFC, gas turbine, and proton exchange membrane electrolyzer using a developed aquila optimizer[J]. International Journal of Hydrogen Energy, 2022, 47(14): 8943-8955.

[12] MA L, LI J, ZHAO Y. Ma L, Li J, Zhao Y. Population forecast of China's rural community based on CFANGBM and improved aquila optimizer algorithm[J]. Fractal and Fractional, 2021, 5(4): 190.

[13] WANG S, JIA H, LIU Q, et al. An improved hybrid aquila optimizer and Harris Hawks optimization for global optimization[J]. Mathematical Biosciences and Engineering, 2021, 18(6): 7076-7109.

[14] EWEES A A, ALGAMAL Z Y, ABUALIGAH L, et al. A cox proportional-hazards model based on an improved aquila optimizer with whale optimization algorithm operators[J]. Mathematics, 2022, 10(8): 1273.

[15] 杜彦斌, 周志杰, 许磊, 等. 基于灰色关联分析与自适应混沌差分进化算法的激光熔覆工艺参数优化方法[J]. 计算机集成制造系统, 2022, 28(01): 149-160.
DU Y B, ZHOU Z J, XU L, et al. Laser cladding process parameter optimization method based on grey relational analysis and ACDE algorithm[J]. Computer Integrated Manufacturing Systems, 2022, 28(01): 149-160. (in Chinese)

[16] 周鹏, 董朝轶, 陈晓艳, 等. 基于阶梯式Tent混沌和模拟退火的樽海鞘群算法[J]. 电子学报, 2021, 49(9): 1724-1735.

ZHOU P, DONG C Y, CHEN X Y, et al. A salp swarm algorithm based on stepped Tent chaos and simulated annealing[J]. Acta Electronica Sinica, 2021, 49(9): 1724-1735. (in Chinese)

[17] 龙文, 伍铁斌, 唐明珠, 等. 基于透镜成像学习策略的灰狼优化算法[J]. 自动化学报, 2020, 46(10): 2148-2164.
LONG W, WU T B, TANG M Z, et al. Grey wolf optimizer algorithm based on lens imaging learning strategy [J]. Acta Automatica Sinica, 2020, 46(10): 2148-2164. (in Chinese)

[18] TSUNEDA A. Orthogonal chaotic binary sequences based on Bernoulli map and Walsh functions[J]. Entropy, 2019, 21(10): 930.

[19] 褚鼎立, 陈红, 王旭光. 基于自适应权重和模拟退火的鲸鱼优化算法[J]. 电子学报, 2019, 47(5): 992-999.
CHU D L, CHEN H, WANG X G. Whale optimization algorithm based on adaptive weight and simulated annealing[J]. Acta Electronica Sinica, 2019, 47(5): 992-999. (in Chinese)

[20] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用[J]. 控制与决策, 2021, 37(1): 87-96.
FU H, LIU H. Improved sparrow search algorithm with multi-strategy integration and its application[J]. Control and Decision, 2021, 37(1): 87-96. (in Chinese)

[21] 刘成汉, 何庆. 融合多策略的黄金正弦黑猩猩优化算法[J/OL]. 自动化学报. DOI:10.16383/j.aas.c210313.
LIU C H, HE Q. Golden sine chimp optimization algorithm integrating multiple strategies[J/OL]. Acta Automatica Sinica, 2022. DOI: 10.16383/j. aas. c210313. (in Chinese)

[22] DHAWALE D, KAMBOJ V K, ANAND P. An effective solution to numerical and multi-disciplinary design optimization problems using chaotic slime mold algorithm[J/OL]. Engineering with Computers. DOI: 10.1007/s00366-021-01409-4.

[23] LONG W, WU T, LIANG X, et al. Solving high-dimensional global optimization problems using an improved sine cosine algorithm[J]. Expert Systems with Applications, 2019, 123: 108-126.

作者简介

张长胜　男, 1970 年 6 月生于陕西省平利县. 现为昆明理工大学副教授、硕士生导师, 从事复杂工业过程建模、智能优化算法等研究.
E-mail: 122832170@qq.com

张健忠(通讯作者)　男, 1997 年 10 月出生于云南省宣威市. 硕士研究生. 主要研究方向为智能优化算法与机械优化设计.
E-mail: 2980824588@qq.com