DOI:10.19652/j.cnki.femt.2305410

基于双重动态调整的改进非洲秃鹫优化算法

陈麒羽[1] 邵 洁[1] 王超群[2] 陈 乐[2] 邰兴雨[3]
(1.上海电力大学电子与信息工程学院 上海 200090;2.航天智慧能源研究院 上海 201201;
3.上海航天能源股份有限公司 上海 201201)

摘 要:针对非洲秃鹫优化算法(African vulture optimization algorithm,AVOA)多样性低、探索开发能力不平衡、易发生早熟的现象,提出了一种基于双重动态调整的改进非洲秃鹫优化算法(improvement African vulture optimization algorithm,IAVOA)。改进后的算法分为3个部分,通过引入混沌映射初始化种群,以确保种群在前期寻优中具有较高的多样性;加入动态调整因子来确定当前最优个体,用来平衡前期探索与后期开发的能力;针对AVOA中饥饿率的变化情况加入动态调整的高斯扰动,用于防止早熟问题的发生,提高最终解的质量。改进后的算法在9个标准测试函数上进行测试。结果表明,该算法表现出更佳的求解性能。

关键词:非洲秃鹫优化算法;高斯干扰;动态调整因子;混沌映射
中图分类号:TP18;TP301 **文献标识码**:A **国家标准学科分类代码**:520.10

Improved African vulture optimization algorithm based on dual dynamic adjustment

Chen Qiyu[1] Shao Jie[1] Wang Chaoqun[2] Chen Le[2] Tai Xingyu[3]
(1. School of Electronic and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China; 2. Aerospace Smart Energy Research Institute, Shanghai 201201, China; 3. Shanghai Aerospace Energy Corporation, Shanghai 201201, China)

Abstract: Abstract: Aiming at the phenomena of low diversity, unbalanced exploration and exploitation ability, and susceptibility to premature maturity in African vulture optimization algorithm (AVOA), an improved African vulture optimization algorithm (IAVOA) based on double dynamic adjustment is proposed. The improved algorithm is divided into three parts, initializing the population by introducing chaotic mapping to ensure that the population has high diversity in the pre-optimization; adding dynamic adjustment factors to determine the current optimal individual, which is used to balance the ability of pre-exploration and post-exploitation; and adding dynamically adjusted Gaussian interference for the change of the starvation rate in the AVOA, which is used to prevent premature maturation and improve the quality of the final solution. In order to verify the effectiveness of the algorithm, the algorithm is subjected to simulation experiments on nine standard test functions. The results show that the algorithm exhibits better solution accuracy as well as convergence speed.

Keywords: African vulture optimization algorithm; Gaussian interference; dynamic adjustment factor; chaos mapping

0 引 言

近些年,元启发式优化算法在许多研究领域中广泛应用。元启发式优化算法通过算法寻优来求解非线性以及多约束问题[1]。例如鲸鱼优化算法(whale optimization algorithm,WOA)[2]、灰狼优化算法(gray wolf optimization algorithm,GWO)[3]、粒子群优化算法(particle swarm optimization algorithm,PSO)[4]等。由于元启发式优化算法不依赖梯度信息,目前已经成功应用在工程设计[5]、非线性预测[6]、故障诊断[7]等领域。针对传统算法容易陷入局部最优解的缺点,研究人员逐渐将研究的重心放在算法优化当中。

收稿日期:2023-08-08

非洲秃鹫优化算法(African vulture optimization algorithm,AVOA)是 Benyamin 等[8]在 2021 年提出的一种新型元启发式优化算法,AVOA 通过选取最优秃鹫、饥饿率、搜索、开发 4 个阶段来解决求解问题。AVOA 通过觅食行为来模拟群体的多样性,这使得算法能够在解的空间进行全局搜索,以寻找更优的解。其次,AVOA 具有并行处理的能力,可以同时搜索多个候选解,这也意味着 AVOA 能够在处理单元或者多线程环境中高效执行,加快求解速度,以上性能使 AVOA 在大多数工程案例中提供卓越的性能[9-10]。

作为一种新提出的元启发式优化算法,AVOA 仍处于起步阶段。考虑到算法的随机性,AVOA 很容易陷入种群多样性低与局部最优解的困境。为了应对这些问题,研究人员对 AVOA 存在的问题进行改进,Liu 等[11]通过在探索阶段加入准对立学习机制与差分因子,平衡了 AVOA 探索与开发能力,但如果前期进入开发阶段很容易陷入局部最优的问题。Chen 等[12]利用反对学习机制防止 AVOA 陷入局部最优,并用它来确定质子交换膜燃料电池反应器的最佳参数,改进后的 AVOA 对比其他方案有着最小的误差值,但其仅考虑寻找最优解能力,在收敛速度上并没有明显提升。Zheng 等[13]通过在 AVOA 开发阶段加入选择堆积机制提升全局优化能力,提高了癌症数据集的多层感知分类精度,但其计算复杂度更高,收敛速度有所下降。

以上研究虽然对 AVOA 寻优能力有所提升,但在收敛速度与优化精度上仍有改进空间。例如在搜索阶段,种群最优位置的生成仅参考初始值,不能够随阶段变化而动态更新,这会导致算法前期收敛速度慢,后期种群多样性低致使开发不彻底。除此之外,AVOA 在前期就使用莱维飞行很容易造成算法跳过了潜在的更优解,从而陷入局部最优解的早熟的情况。针对以上问题,提出了一种基于双重动态调整的改进 AVOA(IAVOA)算法,IAVOA 算法在前期通过动态调整因子来确定当前最优种群,加快收敛速度;引入动态调整的高斯干扰,用来防止早熟现象的发生;后期通过减小调整因子来聚焦于局部开发中,提高解的收敛精度;最后通过 9 个基准测试函数的实验来验证算法的收敛速度与优化性能。

1 AVOA

AVOA 从随机个体开始,在第 1 个阶段开始前设置 N 只随机生成的秃鹫作为初始种群。

1)最优秃鹫选取阶段。种群中最优秃鹫与次优秃鹫被设置并保存在 AVOA 中,剩下的秃鹫通过跟随最优或次优秃鹫来寻找食物。设置方程来确定当前应跟随哪只秃鹫,如式(1)所示。

$$R(i)=\begin{cases}Best\_vulture1\_X, & p_i=L_1\\ Best\_vulture2\_X, & p_i=L_2\end{cases} \tag{1}$$

$$L_1+L_2=1 \tag{2}$$

式中:$R(i)$ 为选中的秃鹫;$Best\_vulture1\_X(i)$、$Best\_vulture2\_X(i)$ 表示种群中最优秃鹫与次优秃鹫;p_i 由式(3)得到;L_1 和 L_2 为区间[0,1]的参数。轮盘赌算法用于选择最优或次优秃鹫,如式(3)所示,$F(x_i)$ 为最优秃鹫或次优秃鹫的选择概率。

$$P(X_i)=\frac{F(x_i)}{\sum_{j=1}^{N}F(x_i)} \tag{3}$$

2)设置饥饿率阶段。受到秃鹫吃饱或饥饿行为启发,研究人员对于这种行为创建数学方程式:

$$t=h\times\left(\sin^{w}\left(\frac{\pi}{2}\times\frac{iteration}{\max iterations}\right)+\cos\left(\frac{\pi}{2}\times\frac{iteration}{\max iterations}\right)-1\right) \tag{4}$$

$$F=(2\times rand_1+1)\times z\times\left(1-\frac{iteration}{\max iterations}\right)+t \tag{5}$$

式中:F 表示秃鹫饥饿率;$iteration$ 表示当前迭代次数;$\max iterations$ 表示迭代总数;z 是介于−1~1 的随机数,每次迭代都会改变;h 是−2~2 的随机数;$rand_1$ 是 0~1 的随机值。秃鹫饥饿率变化情况如图 1 所示。

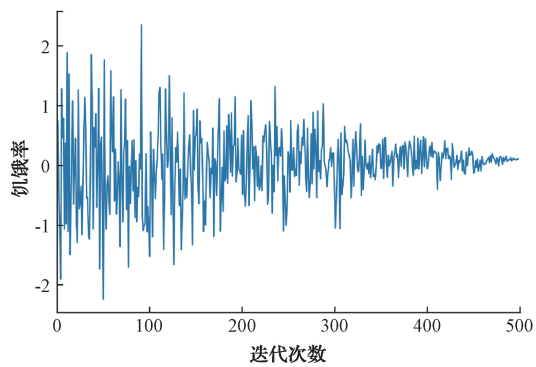


图 1 迭代过程中秃鹫饥饿率的变化

3)探索阶段。如果|F|>1,进入探索阶段,个体在空间中寻找最优解的过程由式(6)定义。在这种情况下,每只秃鹫都会在环境中随机搜索使它感到饱腹感的区域。

$$P(i+1)=\begin{cases}R(i)-D(i)\times F, & P_1\geqslant r_{p_1}\\ R(i)-F+r_1\times((ub-lb)\times r_2+lb), & P_1<r_{p_1}\end{cases} \tag{6}$$

$$D(i)=|X\times R(i)-P(i)| \tag{7}$$

式中:X 为区间(0,1)的随机数,用来模拟保护食物不被其他秃鹫发现;r_{p_1}、r_1、r_2 均为随机生成介于 0~1 的数;lb 与 ub 分别表示上下两个边界。

4)开发阶段。当值|F|介于 0.5~1 时,AVOA 进入开发阶段的第 1 阶段。执行两种不同策略,分别是旋转飞行和围攻策略。

在围攻策略中,此时秃鹫相对能量充足,较弱的秃鹰

试图通过聚集在健康秃鹰周围并引起小冲突来从健康秃鹰那里获取食物式(8)与(9)来模拟这个阶段。

$$P(i+1)=D(i)\times(F+r_3)-d(t),P_2\geqslant r_{P_2} \quad (8)$$

$$d(t)=R(i)-P(i) \quad (9)$$

其中，P_2 用于确定每个策略的选择，该值介于 0～1。在该阶段开始时，生成 r_{P_2}，它是一个介于 0～1 的随机数。如果该数字$\geqslant P_2$，则实施围攻策略。如果该随机数$<P_2$，则执行旋转飞行策略。

秃鹰经常进行旋转飞行，通过螺旋模型对旋转飞行进行数学建模。旋转飞行如下：

$$S_1=R(i)\times\left(\frac{r_3\times P(i)}{2\pi}\right)\times\cos(P(i)) \quad (10)$$

$$S_2=R(i)\times\left(\frac{r_4\times P(i)}{2\pi}\right)\times\sin(P(i)) \quad (11)$$

$$P(i+1)=R(i)-(S_1+S_2),\quad P_2<r_{P_2} \quad (12)$$

如果$|F|<0.5$，进入开发阶段的第 2 阶段，此时大多数秃鹫已经吃饱了，但选中的最佳秃鹫在长时间飞行后变得虚弱与饥饿，则执行算法的此阶段。

有时，秃鹫会挨饿，这将会伴随着大量的食物竞争，可能会在一个食物来源上积累几种秃鹫。此时进入食物竞争策略，如式(13)、(14)所示。

$$A_1=Best\_vulture1\_X(i)-\frac{Best\_vulture1\_X(i)\times P(i)}{Best\_vulture1\_X(i)-P(i)^2}\times F \quad (13)$$

$$A_2=Best\_vulture2\_X(i)-\frac{Best\_vulture2\_X(i)\times P(i)}{Best\_vulture2\_X(i)-P(i)^2}\times F \quad (14)$$

式中：F 为秃鹫饥饿率；$P(i)$ 为当前秃鹫的位置向量。

$$P(i+1)=\frac{A_1+A_2}{2},\quad P_3\geqslant r_{P_3} \quad (15)$$

其中，P_3 用于确定开发的第 2 阶段策略的选择，该值介于 0～1。r_{P_3} 是一个介于 0～1 的随机数。如果该数字$\geqslant P_3$，则实施食物竞争策略，如果该随机数$<P_3$，则执行莱维飞行策略。最后，使用式(15)对所有秃鹫进行融合，$P(i+1)$ 是下一次迭代的秃鹫位置。

2 IAVOA

传统 AVOA 在最优秃鹫选取阶段仅参考初始值，不能够随迭代变化动态更新，这会导致 AVOA 前期收敛速度慢，后期局部开发不彻底。除此之外，AVOA 在前期就使用莱维飞行很容易造成算法跳过潜在的更优解，从而陷入局部最优解的早熟的情况。为了针对以上问题，提出了一种基于双重动态调整的改进 AVOA 以改善前期探索与后期开发性能。

2.1 基于混沌映射优化初始种群

在 AVOA 进行最优秃鹫选取阶段之前要首先准备初始化种群，初始种群的设定将直接影响最终解。原 AVOA 种群初始化采用 Mersenne Twister 算法，但这种方法的状态空间很大，通常为$(2^{19\,937}-1)$，这意味着该算法可以生成极长周期的随机数序列，然而对于小概率事件会以很低的概率生成随机数，当进行随机抽样时，很可能遇到截尾现象，这会对种群多样性产生影响。为了确保最终解的准确性，通常需要增加抽样次数。然而，在实际应用中，增加抽样次数并不是一个可行的方式。

为应对以上情况，引入了混沌映射的思想，利用混沌性质的非线性特征生成高度复杂、随机且非重复的序列。混沌模型具有的非线性特征能够更好地捕捉到异常和稀有事件，增加 AVOA 种群多样性。Tent 混沌映射已被证实在多个方面比 Logistic 混沌映射更具优势，包括遍历性、均匀性和迭代速度[14]。因此采用 Tent 混沌系统对种群初始化，用来提高初始解的覆盖范围，从而提高种群多样性。考虑到 Tent 映射集中在 0.2～0.8，在其他区间分布较差，这导致最优解为边缘值时才容易求得。采用改进的 Tent 混沌映射初始化种群，通过添加扰动，使得映射结果更加均匀和随机，如式(16)所示。

$$x_{n+1}=\begin{cases}2\times x_n+rand(0,1)\times\mu, & x_n<0.5\\ 2-2\times(x_n+rand(0,1)\times\mu), & x_n\geqslant 0.5\end{cases} \quad (16)$$

即：

$$x_{n+1}=[2\times x_n+rand(0,1)\times\mu]\mathrm{mod}1 \quad (17)$$

式中：x_n 表示第 n 次映射函数值，n 表示映射次数，μ 为扰动参数，实验取 $\mu=0.1$，$x_n\in[0,1]$，rand(0,1)能够均匀生成 0～1 的数，mod1 为取余函数，mod1 函数可以确保 x_n 值一直在(0,1)内。

将混沌变量逆映射在种群解空间，即可获得种群初始化的位置，如式(18)所示。

$$x'_n=l_n+(u_n-l_n)\times x_n \quad (18)$$

式中：l_n、u_n 为优化变量区间的最小值与最大值；x'_n 为初始秃鹫的位置。

2.2 基于动态调整的秃鹫选择

1)最优秃鹫选择

在 AVOA 的最优秃鹫选择阶段，下一个最佳位置的选择对算法的效率与结果至关重要。原 AVOA 用 α、β 来代表选择全局最优秃鹫与次优秃鹫的概率，当 α 值接近 1，β 参数值接近 0 时，其全局寻优能力较好，当 α 值接近 0，β 参数值接近 1 时，则会导致 AVOA 种群多样性增加，有助于探索更多可能的解空间。由于传统 AVOA 的 α 和 β 均为固定不变的，因此限制了 AVOA 的局部与全局寻优能力。

为了加快 AVOA 收敛速度的同时在探索阶段挖掘局部最优解，本文对 α 和 β 取值进行了改进。针对 AVOA 寻找最优解过程非常复杂这一特点，参数选择线性变换难以体现 AVOA 实际优化过程，因此选择使其由固定值转为非线性变化，这种方法能够有效避免某些情况线性变化导致算法过早的陷入局部最优解。

$$\alpha = \alpha_{\max} \times \left(\frac{\alpha_{\min}}{\alpha_{\max}}\right)^{\left(\frac{t}{T}\right)^{\partial}} \tag{19}$$

$$\beta = 1 - \alpha \tag{20}$$

式中：t 为当前迭代次数；T 为迭代最大次数；∂为调节参数。通过调整∂可以设置 α 下降速度的快慢，$\partial \in [0,1]$，最终 α 会处于 $\alpha_{\min}$ 的值。α 值随迭代变化如图 2 所示。

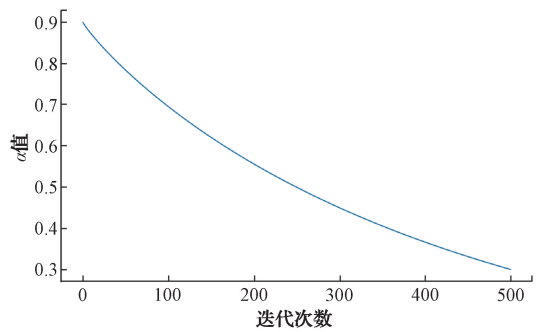


图 2 α 值随迭代变化过程

由图 2 可知，选择这样的动态调整参数能够加快 AVOA 收敛速度的同时提升算法的准确性。基于动态调整的最优秃鹫选择可以在算法迭代前期注重全局最优解寻求，并在迭代后期注重局部最优解的挖掘。其中 α 应在迭代前期设置较大的数，在迭代的前期，α 值接近 1，其全局寻优能力较好，以增强算法的全局搜索能力，并加快收敛速度，在迭代后期，β 参数值接近 1，增加 AVOA 的多样性，精细地寻找最优解。

2）食物竞争策略

在开发阶段的食物竞争策略，如式(15)所示，AVOA 选取当前位置为最优和次优秃鹫的中间位置，但在 AVOA 后期，如果两者对当前位置选取的影响相同，不能够充分发挥后期开发阶段的性能。因此选用式(19)、(20)对当前位置进行更新。改进后的位置更新，式(15)改写为：

$$P(i+1) = A_1 \times \alpha + A_2 \times \beta, \quad P_3 \geqslant r_{p_3} \tag{21}$$

式中：A_1 和 A_2 分别由式(13)、(14)得到。考虑到 AVOA 后期最优和次优秃鹫趋于相近的值，为防止秃鹫聚集而造成局部最优解的情况，添加了服从贝塔分布的随机数，用于对当前位置进行扰动变异，以增强 AVOA 后期开发能力，使食物竞争策略不容易陷入局部最优。最终位置更新如下：

$$P'(i+1) = A_1 \times \alpha + A_2 \times \beta + \delta \times betarnd(0,1), \quad P_3 \geqslant r_{p_3} \tag{22}$$

式中：$betarnd(0,1)$ 为贝塔分布生成的随机数；δ 为贝塔分布的调整因子，实验取 $\delta = 0.01$。

2.3 基于动态高斯扰动的自适应莱维飞行

在 AVOA 的开发阶段，此时种群利用探索阶段找到的解进行局部搜索。AVOA 使用莱维飞行策略来模拟秃鹫行为，通过全局搜索来找到最优解。莱维飞行是一种源于混沌理论的非高斯步行模型，其主要原理是模拟自然界中昆虫的飞行，莱维飞行的核心是短距离游走和长距离跳跃交替变化，当处于短距离游走时种群的多样性将会提高，处于长距离跳跃时种群搜索具有方向的多样性，搜索更为详细[15]。

$$P(i+1) = R(i) - |R(i) - P(i)| \times F \times levy(\lambda) \tag{23}$$

$$\sigma = \left(\frac{\Gamma(\beta) + 1 \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{24}$$

$$levy(\lambda) = \frac{\mu \times \sigma}{|V|^{\frac{1}{\beta}}} \tag{25}$$

式中：β 在标准莱维飞行中取值为固定默认值 1.5；$P(i+1)$ 为秃鹫下一个要去的位置；$levy(\lambda)$ 为莱维随机搜索路径，λ 是一个正实数，表示莱维分布的尺度参数；$\Gamma(\beta)$ 为标准的伽玛函数；$d(i)$ 为最优秃鹫与当前位置的距离；μ 与 V 为服从正态分布的随机数。莱维飞行轨迹如图 3 所示。

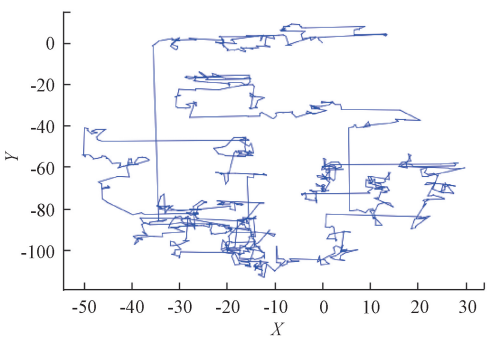


图 3 莱维飞行轨迹

虽然莱维飞行强化了算法局部开发，但其应用在 AVOA 中仍存在着如下两个问题。

1）莱维飞行在空间遍历中存在边缘个体位置信息利用不充分的问题，无法利用自身信息进行候选解的更新。

2）若 AVOA 前期就使用莱维飞行策略很容易导致算法早熟。

针对第 1 个问题，提出了莱维飞行动态调整因子，使得步长因子随迭代次数进行动态调整，动态调整因子 γ 公式为：

$$\gamma = e^{\frac{1-T}{T+1-t}} \tag{26}$$

$$levy'(\lambda) = levy(\lambda) \times \gamma \tag{27}$$

式中：t 为当前所处的迭代次数；T 为最大迭代次数；γ 可随着迭代次数的增加而自动调整自身值的大小，这样可以在早期迭代阶段更多的进行搜索，而在后期迭代阶段更加侧重利用已经发现的好的解，有利于充分利用自身位置信息平衡全局探索与局部利用的能力。

针对第 2 个问题，提出了动态高斯扰动，在不影响莱

维飞行随机性的前提，使得莱维飞行前期较小的影响AVOA优化过程。

$$G = Gaussian(0,\varepsilon) \tag{28}$$

$$\varepsilon = \frac{(e^{\frac{t}{T}}-1)}{(e^{\frac{t}{T}}+1)} \tag{29}$$

式中：高斯扰动因子G是均值为0；方差为ε得到的值，$\varepsilon \in [-1,1]$。由图1可知，AVOA饥饿率随迭代变化呈现上下波动且趋于0的形状，为了应对这种变化趋势，引入动态高斯扰动因子。在迭代前期，饥饿率处于较大值时，此时AVOA算法注重在第3阶段全局探索中，加入高斯扰动因子能够使莱维飞行较小的影响全局搜索，避免了算法早熟现象。在迭代后期，此时饥饿率处于趋于0的情况，AVOA注重第4阶段局部开发阶段，为应对开发阶段容易陷入局部最优解的问题，较大的高斯扰动能够帮助秃鹫跳出局部最优解，高斯扰动因子随迭代变化如图4所示。

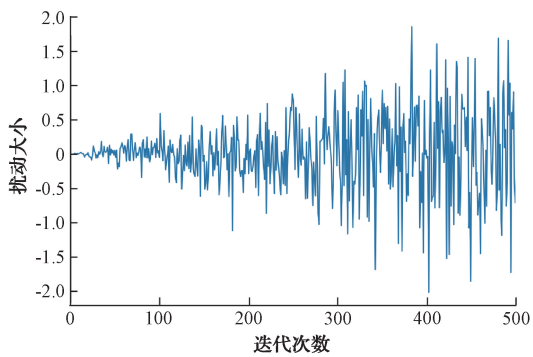


图4 高斯扰动因子随迭代变化

最终的AVOA位置更新通过式(31)实现。

$$P'(i) = G(|R(i) - P'(i)| \times F \times levy'(i)) \tag{30}$$

$$P'(i+1) = R(i) - P'(i) \tag{31}$$

式中：$R(i)$为当前选取的最佳位置；$P'(i)$为当前所处的位置；$|F|$为当前种群饥饿率；$P'(i+1)$为更新后的下一个位置。相比较原AVOA算法，基于高斯扰动的动态的莱维飞行能够自动适应不同问题的特征与搜索空间的变化，有着更高效的搜索效率，能够有效的避免陷入局部最优解。

算法1 IAVOA

1: Inputs: The population size N and maximum number of iterations T
2: Outputs: The location of Vulture and its fitness value
3: Initialize the random population Pi(i =1,2,…,N) using Eq. (16)
4: while (stopping condition is not met) do
5:Calculate the fitness values of Vulture
6: Set BestVulture1 and BestVulture2 as the location of Vulture
8: for (each Vulture (Pi)) do
9: Select R(i) using Eq. (19)
10: Update the F using Eq. (5)
11: if (|F|≥1) then
13: Update the location Vulture using Eq. (6)
16: if (|F|<1) then
17: if (|F|≥0.5) then
18: if (P2≥randP2) then
19: Update the location Vulture using Eq. (8)
20: else
21: Update the location Vulture using Eq. (12)
22: else
23: if (P3≥randP3) then
24: Update the location Vulture using Eq. (21)
25: else
26: Update the location Vulture using Eq. (31)
27: Return BestVulture1

3 仿真实验对比

3.1 参数设置

为探究和验证IAVOA的寻优性能，共进行两组实验：第1组实验利用基准测试函数测试IAVOA、基本AVOA、AVOA变体的性能对比；第2组实验通过与其他元启发式优化算法的对比实验验证IAVOA的优化性能。实验硬件使用Intel(R) Core(TM)i5-8250UCPU@1.80 GHz，实验软件为JetBrains PyCharm Community Edition 2018，实验环境为Python 3.7。

本文选取单峰、多峰合计9个经典测试基准函数进行算法寻优测试时，根据测试函数不同的性能来验证算法寻优能力。单峰函数(F_1～F_6)主要用来检验算法的收敛速度，多峰函数(F_7～F_9)主要用来检验算法的全局探索能力和跳出局部最优解的能力，单峰、多峰基准函数如表1所示，基准测试函数的形状如图5、6所示。

3.2 与基本AVOA和AVOA变体的比较

将IAVOA、AVOA和基于准对立学习的改进非洲秃鹫算法(quasi-oppositional African vulture optimization algorithm，QOAVOA)[11]进行比较，验证所提算法性能。实验设置公共参数种群规模$N=30$，最大迭代次数为500，各个算法重复运算30次记录其平均值、标准差与最优值。经过30次独立运行之后，实验结果如表2所示。

对于单峰基准函数，优化函数最终仅能得到一个全局最优解，并且没有其他局部最优解，因此可以通过单峰基准函数来测试算法的开发能力。由表2可知，IAVOA与QOAVOA两种改进方式性能均优于传统AVOA，并且在

表 1 基准函数

函数	维度	区间	最优值
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[−100,100]	0
$F_2=\sum_{i=1}^{n}\lvert x_i^2\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[−100,100]	0
$F_3=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)$	30	[−100,100]	0
$F_4=\max\{\lvert x_i\rvert,1\leqslant i\leqslant n\}$	30	[−100,100]	0
$F_5=\sum_{i=1}^{n}100(x_{i+1}-x_i^2)+(x_i-1)^2$	30	[−100,100]	0
$F_6=\sum_{i=1}^{n}ix_i^4+random[0,1)$	30	[−100,100]	0
$F_7=\sum_{i=1}^{n}[x_i-10\cos(2\pi x_i)+10]$	30	[−100,100]	0
$F_8=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}10\cos(2\pi x_i)\right)+20+e$	30	[−100,100]	0
$F_9=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1]$	30	[−100,100]	0

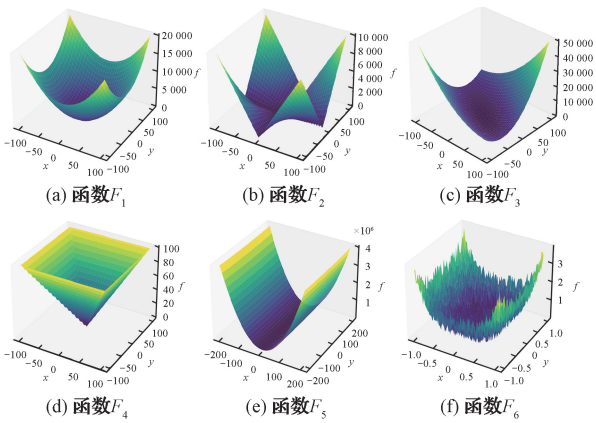


图 5 单峰基准函数

大多数单峰基准测试函数中，IAVOA 均展示出了更好的性能。对于基准函数 F_1 与 F_3，3 种优化算法均能达到理论最优值。对于基准函数 F_2 与 F_4，IAVOA 仍然能够找出最优值，对比 AVOA 与 QOAVOA 展示出了优势。对于基准函数 F_5 与 F_6，3 种寻优算法都不能得到理论最优值，而 IAVOA 的平均值与标准差均高于其他两个算法。

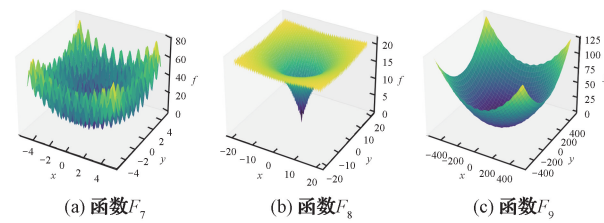


图 6 多峰基准函数

对于多峰基准函数，优化函数有多个局部最优解，仅有一个为全局最优，因此可以通过多峰基准函数来测试算法局部开发能力与收敛速度。由表 2 可知，对于基准函数 F_7 与 F_9，3 种优化算法均能达到理论最优值，但 IAVOA 在收敛性上展现了更加高效的收敛速度。对于多峰基准

表 2 IAVOA、QOAVOA、AVOA 算法的寻优结果

函数	算法	最优值	平均值	标准差
F_1	IAVOA	**0**	**0**	**0**
	AVOA	0	0	0
	QOAVOA	0	0	0
F_2	IAVOA	**0**	**0**	**0**
	AVOA	6.21×10^{-155}	7.26×10^{-152}	0
	QOAVOA	4.23×10^{-202}	8.54×10^{-195}	0

续表

函数	算法	最优值	平均值	标准差
F_3	IAVOA	**0**	**0**	**0**
	AVOA	0	0	0
	QOAVOA	0	0	0
F_4	IAVOA	**0**	**0**	**0**
	AVOA	4.95×10^{-145}	9.98×10^{-144}	0
	QOAVOA	5.77×10^{-247}	5.41×10^{-247}	0
F_5	IAVOA	8.85×10^{-1}	**1.36**	**8.75**
	AVOA	1.37×10^{-2}	5.62×10^{2}	1.89×10^{1}
	QOAVOA	$\mathbf{2.26\times10^{-1}}$	5.41×10^{3}	4.43×10^{3}
F_6	IAVOA	7.57×10^{-5}	$\mathbf{2.51\times10^{-5}}$	6.56×10^{-5}
	AVOA	1.90×10^{-3}	8.73×10^{-3}	4.34×10^{-3}
	QOAVOA	$\mathbf{5.42\times10^{-5}}$	7.15×10^{-5}	$\mathbf{3.22\times10^{-5}}$
F_7	IAVOA	**0**	$\mathbf{6.47\times10^{-16}}$	$\mathbf{3.83\times10^{-16}}$
	AVOA	0	4.41×10^{-15}	3.10×10^{-15}
	QOAVOA	0	1.77×10^{-15}	8.47×10^{-14}
F_8	IAVOA	$\mathbf{4.14\times10^{-17}}$	$\mathbf{1.28\times10^{-16}}$	$\mathbf{9.69\times10^{-17}}$
	AVOA	9.84×10^{-16}	6.47×10^{-15}	2.24×10^{-15}
	QOAVOA	8.15×10^{-16}	7.32×10^{-15}	3.98×10^{-15}
F_9	IAVOA	**0**	1.31×10^{-13}	$\mathbf{1.22\times10^{-13}}$
	AVOA	0	5.46×10^{-14}	9.69×10^{-13}
	QOAVOA	0	$\mathbf{2.44\times10^{-14}}$	7.17×10^{-13}

函数 F_8，IAVOA 也展现出了更好的性能。经分析单峰基准函数与多峰基准函数，可以得出 IAVOA 对比 QOAVOA 与 AVOA 性能得到了一定的提升。

如图 7 所示，经过测试函数的 30 次平均结果可知，IAVOA 相比较 QOAVOA 与 AVOA 展现出了更加高效的收敛性能，其中在 F_1、F_3 测试函数在迭代 200 次内能够收敛到最优值，并且在其他测试函数中都有着明显收敛速度上的提升，因此可以说明 IAVOA 对比 AVOA 收敛性能得到显著提升。

在算法前期，IAVOA 使用了改进 Tent 混沌映射优化初始种群，这使得 IAVOA 在前期就有着较好的种群多样性，初始值对比其他算法更佳。同时，在最优解的选择中选取了动态调整因子，加快了迭代前期全局搜索效率，弥补了算法前后期搜索开发能力不平衡的问题。由图 7 可知，对比 AVOA 与 QAVOA，IAVOA 均能够达到三者中的最快收敛速度。

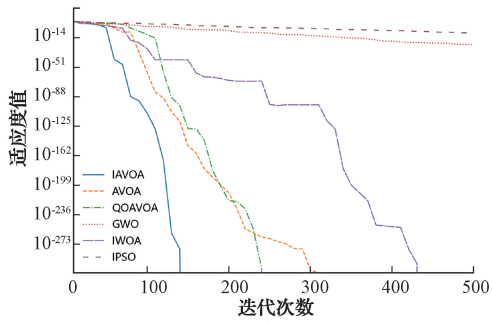
在算法中期，种群饥饿率逐渐趋于 0，此时莱维飞行的加入容易使得算法早期陷入局部最优解。针对种群饥饿率变化趋势，加入了动态调整的高斯扰动因子来提高算法全局探索性能。由图 7(b)、(d)可知，算法在基准函数 F_2 与 F_4 中不容易像其他算法陷入局部最优解，从而进一步找到全局最优解。

在算法后期，种群饥饿率已经逼近 0，局部搜索作为阶段重点，此时通过加入动态莱维飞行来增加开发阶段的探索范围，增强局部搜索能力。

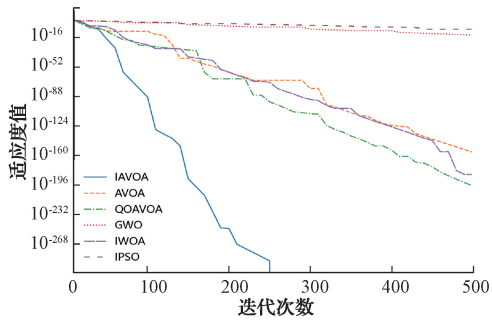
3.3 与其他元启发式优化算法对比实验

为验证 IAVOA 算法在函数寻优应用中的能力，将 IAVOA 与其他研究较多的改进元启发式优化算法进行横向对比。如 GWO[3]、改进粒子群优化算法(IPSO)[5]、改进鲸鱼优化算法(IWOA)[7]进行比较，并按照 3.2 节参数的环境进行实验，记录其平均值、最优值以及标准差，将最终实验结果如表 3 所示。

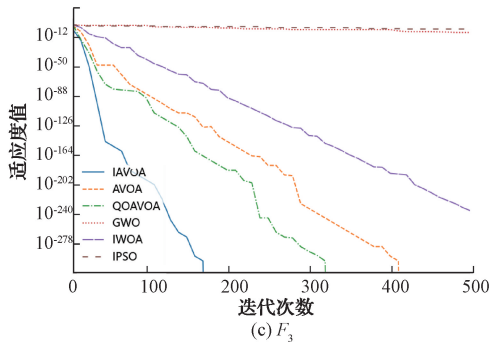
如表 3 所示，将 IAVOA 与其他 3 种优化算法进行了比较，可以看到对于单峰测试函数，IAVOA 均能找到理论的最优解，且均值与标准差在 $F_1\sim F_4$ 均处于最小的值，这说明 IAVOA 在单峰测试函数中比较其他元启发式优化算法有着更好的性能。由图 7 可以看到，随着迭代次数的增加，IAVOA 在大多数测试函数中能够快速寻找到最优解，并且其迭代次数比较其他优化算法更少，能够在应对实际问题中减少实验时长，而其他智能算法在寻优过程中收敛速度慢，难以搜索到最优解。由图 7(g)可知，对于多峰优化函数 F_7，IAVOA 的平均值与标准差并不是



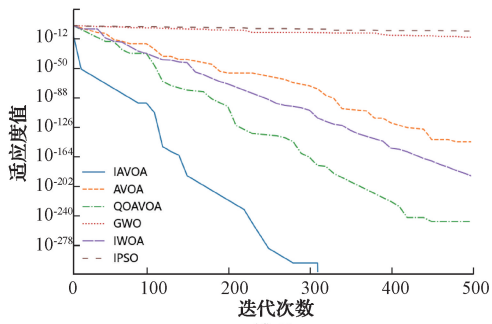
(a) F_1



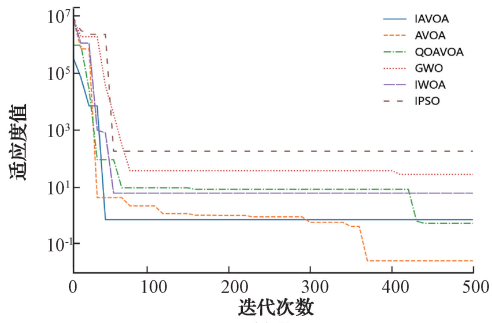
(b) F_2



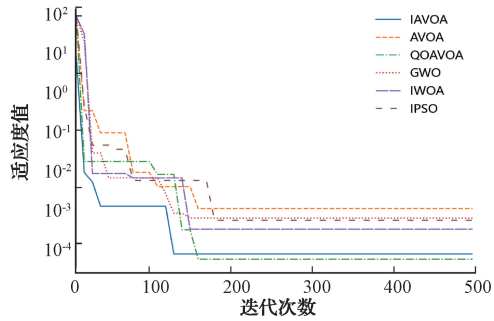
(c) F_3



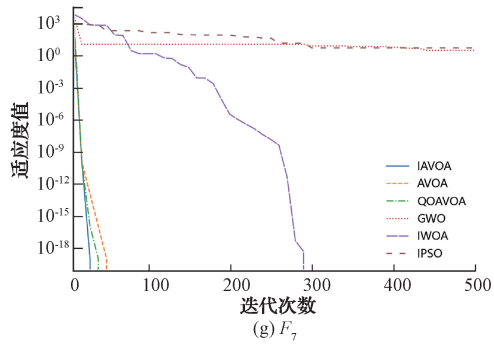
(d) F_4



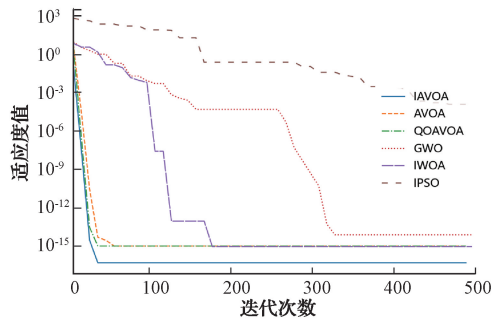
(e) F_5



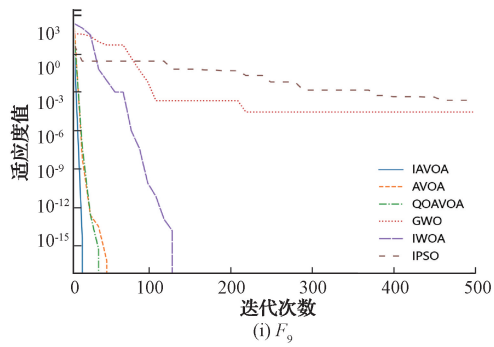
(f) F_6



(g) F_7



(h) F_8



(i) F_9

图 7　智能算法收敛曲线对比

最佳，但 IAVOA 有着更优的收敛速度，对比 IWOA、IPSO 能够更快寻找到理论最优解，能够减少寻找最优解所需要的时间。对于多峰函数 F_8、F_9，IAVOA 搜索到的最优解更为理想，不容易陷入局部最优，有着更快的收敛速度。以上实验结果可以表明，IAVOA 对比其他 3 个元启发式优化算法有着更好的优化性能。

表 3 IAVOA、GWO、IWOA、IPSO 算法的寻优结果

函数	算法	最优值	平均值	标准差
F_1	IAVOA	**0**	**0**	**0**
	GWO	6.85×10^{-26}	7.10×10^{-23}	3.52×10^{-25}
	IWOA	0	0	0
	IPSO	2.72×10^{-10}	4.57×10^{-9}	7.61×10^{-9}
F_2	IAVOA	**0**	**0**	**0**
	GWO	6.07×10^{-14}	1.06×10^{-14}	8.14×10^{-15}
	IWOA	2.63×10^{-180}	8.88×10^{-133}	5.77×10^{-146}
	IPSO	3.57×10^{-6}	4.78×10^{-4}	5.18×10^{-4}
F_3	IAVOA	**0**	**0**	**0**
	GWO	6.42×10^{-6}	3.72×10^{-5}	4.54×10^{-6}
	IWOA	8.97×10^{-236}	3.26×10^{-211}	3.25×10^{-224}
	IPSO	1.19	5.92×10^{1}	6.97×10^{1}
F_4	IAVOA	**0**	**0**	**0**
	GWO	7.96×10^{-10}	8.21×10^{-8}	2.56×10^{-9}
	IWOA	7.48×10^{-189}	3.49×10^{-156}	2.75×10^{-167}
	IPSO	8.42×10^{-3}	2.51	6.75×10^{-1}
F_5	IAVOA	$\mathbf{8.85\times10^{-1}}$	**1.36**	8.75
	GWO	1.48×10^{2}	7.95×10^{2}	3.41×10^{2}
	IWOA	1.42	4.23×10^{1}	**2.90**
	IPSO	2.18×10^{2}	6.48×10^{2}	7.26×10^{1}
F_6	IAVOA	$\mathbf{7.57\times10^{-5}}$	$\mathbf{2.51\times10^{-5}}$	6.56×10^{-5}
	GWO	6.21×10^{-4}	9.42×10^{-3}	4.31×10^{-4}
	IWOA	3.72×10^{-4}	8.62×10^{-4}	$\mathbf{5.85\times10^{-5}}$
	IPSO	5.10×10^{-4}	4.52×10^{-3}	1.27×10^{-4}
F_7	IAVOA	**0**	6.47×10^{-16}	3.83×10^{-16}
	GWO	3.27	6.14	4.18×10^{-1}
	IWOA	0	$\mathbf{3.42\times10^{-16}}$	$\mathbf{8.62\times10^{-17}}$
	IPSO	5.66	7.82×10^{1}	4.32×10^{1}
F_8	IAVOA	$\mathbf{4.14\times10^{-17}}$	$\mathbf{1.28\times10^{-16}}$	$\mathbf{9.69\times10^{-17}}$
	GWO	9.74×10^{-15}	6.26×10^{-14}	1.44×10^{-15}
	IWOA	8.66×10^{-16}	1.28×10^{-14}	9.67×10^{-15}
	IPSO	4.71×10^{-4}	6.12×10^{-2}	2.92×10^{-3}
F_9	IAVOA	**0**	$\mathbf{1.31\times10^{-13}}$	$\mathbf{1.22\times10^{-13}}$
	GWO	6.77×10^{-4}	6.17×10^{-3}	4.69×10^{-4}
	IWOA	0	3.81×10^{-13}	2.56×10^{-13}
	IPSO	8.05×10^{-3}	7.44×10^{-2}	9.47×10^{-2}

4 结 论

针对标准AVOA算法存在的缺点，提出了一种基于双重动态调整的改进AVOA算法。通过引入混沌映射来让种群获得较高的多样性；加入动态调整因子、变异因子、动态调整的高斯扰动，用来分阶段动态调整AVOA的优化性能。为了验证算法有效性，对9个标准测试函数进行仿真实验。实验结果表明IAVOA优于标准AVOA、QOAVOA和3个其他元启发式优化算法，验证了IAVOA更高效的局部搜索和全局搜索能力，收敛速度与鲁棒性得到了有效提升。在后续研究中考虑将改进后的非洲秃鹫优化算法应用在不同领域的实际工程问题当中，进一步验证算法应对现实问题的性能。

参 考 文 献

[1] 高卫峰，罗宇婷，原杨飞. 求解非线性方程组的智能优化算法综述[J]. 控制与决策，2021，36(4)：769-77.

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

[3] MIRJALILI S，MEIDANI K，HEMMASIAN A P，et al. Adaptive grey wolf optimizer[J]. Neural Computing and Applications，2022，34(10)：7711-7731.

[4] 刘秀丽，王鸽，吴国新，等. VMD及PSO优化SVM的行星齿轮箱故障诊断[J]. 电子测量与仪器学报，2022，36(1)：54-61.

[5] 覃玉红，唐求，邱伟，等. 多应力下电能计量设备基本误差预估[J]. 仪器仪表学报，2022，43(4)：18-25.

[6] 王智勇，李丽敏，温宗周，等. 基于GWO-XGBoost泥石流灾害预测[J]. 电子测量技术，2023，46(3)：92-99.

[7] 姜媛媛，牛牧原，陈万利. 基于IWOA-SVM的电路软故障诊断[J]. 电子测量技术，2022，45(2)：159-165.

[8] BENYAMIN A，SOLEIMANIAN F G，SEYEDALI M. African vultures optimization algorithm：A new nature-inspired metaheuristic algorithm for global optimization problems[J]. Computers & Industrial Engineering，2021，DOI：10.1016/j.cie.2021.107408.

[9] YAKOUT A H，KOTB H，ABORAS K，et al. Comparison among different recent metaheuristic algorithms for parameters estimation of solid oxide fuel cell：Steady-state and dynamic models[J]. Alexandria Engineering Journal，2022，61(11)：8507-8523.

[10] BUGRA A，MALARVIZHI C K. Performance comparison of recent population-based metaheuristic optimisation algorithms in mechanical design problems of machinery components[J]. Machines，2021，9(12)：341-341.

[11] LIU R，WANG T，ZHOU J，et al. Improved African vulture optimization algorithm based on quasi-oppositional differential evolution operator[J]. IEEE Access，2022，10：95197-95218.

[12] CHEN Y G，ZHANG G L. New parameters identification of Proton exchange membrane fuel cell stacks based on an improved version of African vulture optimization algorithm[J]. Energy Reports，2022，8(1)：3030-3040.

[13] ZHENG R，HUSSIEN A G，QADDOURA R，et al. A multi-strategy enhanced African vultures optimization algorithm for global optimization problems[J]. Journal of Computational Design and Engineering，2023，10(1)：329-356.

[14] 张娜，赵泽丹，包晓安，等. 基于改进的Tent混沌万有引力搜索算法[J]. 控制与决策，2020，35(4)：893-900.

[15] 徐大也，胡立坤，王小勇，等. 基于概率路线图法的窄道采样与轨迹优化[J]. 国外电子测量技术，2023，42(2)：1-8.

作 者 简 介

陈麒羽，硕士研究生，主要研究方向为元启发式优化算法、图像处理等。

E-mail：592320978@qq.com

邵洁，博士，副教授，主要研究方向为计算机视觉、图像处理等。

E-mail：shaojie@shiep.edu.cn

王超群，本科，研究员，主要研究方向为大型仿真模拟系统。

E-mail：wangchaoqun@htny3.wecom.work

陈乐，硕士研究生，工程师，主要研究方向为数值预测与文本挖掘。

E-mail：ts_37yebook@163.com

邰兴雨，本科，主要研究方向为数值预测。

E-mail：taixy@sae.sh.cn