

精英反向学习及柯西扰动引导的瞪羚优化算法*

班云飞 张达敏 左锋琴 沈倩雯
(贵州大学大数据与信息工程学院 贵阳 550025)

摘要:针对瞪羚优化算法收敛精度低和易陷入局部最优的问题,提出一种精英反向学习及柯西扰动引导的瞪羚优化算法(improved gazelle optimization algorithm, IGOA)。首先,对瞪羚个体利用精英反向学习策略进行初始化,提升初始解的质量并增加种群多样性;其次,在算法迭代初期,利用二阶段非线性惯性权重引导种群的位置更新方式,提高算法的精度并均衡算法的全局搜索和局部搜索;最后,将存活率引导的柯西扰动策略引入勘探阶段种群的位置更新公式中,提升算法跳出局部最优的能力。利用12个基准测试函数和Wilcoxon秩和检验在8个对比算法上进行实验检测,结果表明改进算法寻优精度更高、收敛速度更快且具有跳出局部最优的能力。在齿轮系和三杆桁架设计两个实际工程问题上验证了IGOA的实用性和有效性。

关键词:瞪羚优化算法;精英反向学习;二阶段非线性惯性权重;柯西扰动;工程问题

中图分类号: TP301.6 **文献标识码:** A **国家标准学科分类代码:** 520.10

Elite reverse learning and Cauchy perturbation-guided gazelle optimization algorithm

Ban Yunfei Zhang Damin Zuo Fengqin Shen Qianwen
(School of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

Abstract: Aiming at the problems of low convergence accuracy of gazelle optimization algorithm and easy to fall into local optimum, an elite inverse learning and Cauchy perturbation-guided improved gazelle optimization algorithm (IGOA) is proposed. Firstly, the gazelle individuals are initialized using an elite reverse learning strategy to improve the quality of the initial solution and increase the population diversity. Secondly, at the beginning of the algorithm iteration, the two-stage nonlinear inertia weights are used to guide the position updating method of the population, which improves the accuracy of the algorithm and balances the global and local searches of the algorithm. Finally, the survival rate-guided Cauchy perturbation strategy is introduced into the position updating formula of the population in the exploration stage to improve the ability of the algorithm to jump out of the local optimum. Experimental tests are carried out on 8 comparison algorithms using 12 benchmark test functions and Wilcoxon rank sum test, and the results show that the improved algorithm has higher optimization accuracy, faster convergence speed and ability to jump out of local optimum. The practicality and effectiveness of IGOA are verified on two real engineering problems, namely, gear train and three-bar truss design.

Keywords: gazelle optimization algorithm; elite reverse learning; two-stage nonlinear inertia weights; Cauchy perturbation; engineering problems

0 引言

群智能算法的挖掘与创新为现代社会的运转带来了良好的经济效益,其鲁棒性和稳健性等优势也在各行各业中充分发挥着重要作用。近几年,越来越多的研究人员受

各类动植物种群内或种群间复杂的生物关系的启发,开创了金豺优化算法(golden jackal optimization, GJO)^[1]、正弦余弦优化算法(sine-cosine algorithm, SCA)^[2]、多元宇宙优化算法(multi-verse optimizer, MVO)^[3]等一系列性能完善的智能算法,并将该类算法应用于实际生活的多元领

收稿日期:2024-03-08

* 基金项目:国家自然科学基金(62166006)项目资助

域中,例如工程优化^[4]、神经网络的超参数优化^[5-6]、路径规划^[7-8]以及资源调度^[9]等。

瞪羚优化算法(gazelle optimization algorithm,GOA)于2023年由Agushaka等^[10]提出的,模拟了瞪羚群体逃离捕食者的运动行为。瞪羚作为一种食草性动物,主要在旱地和沙漠地区活动。出于安全性和社会行为原因,瞪羚适应于群居生活,群体的社会结构以性别为基础,雌性群体及其幼崽组成一个较小的群体,雄性群体帮助供养和保护该群体。群体中的社交活动有助于提高整个群体的安全性。瞪羚个体自身有强烈的听觉、视觉和嗅觉,身体轻盈,奔跑速度快,面对危险会通过拍打尾巴、跺脚或在空中跳跃等行为相互警告,其身体优势及习性为瞪羚的生存提供了有力保障。为了在包含天敌的自然环境下生存和繁衍,瞪羚必须以更快速度找到安全区域,超越和战胜捕食者。瞪羚优化算法将瞪羚逃离捕食者的过程划分为开发阶段和勘探阶段,不断向全局最优个体、当前最优个体和随机个体学习以及积累经验,进而更新迭代自身位置到安全区域,逃离捕食者的追逐。与其他群智能算法相比,GOA属于比较新的群体智能算法,具有参数少、简单易行和寻优能力强等性能优势。

然而,GOA也存在与其他算法类似的收敛精度低、易陷入局部最优等内部缺点。文献[11]提出了基于罗森布罗克直接旋转策略的正交学习方法来改进GOA,在数据聚类问题和基准测试函数上进行了测试,表明了改进算法的可行性和有效性。文献[12]沿用了文献[11]的改进策略,避免算法陷入局部最优的同时增强了算法的多样性,并将其应用在网络入侵检测中,结果显示改进的深度残差收缩网络能够获得良好的分类检测性能。文献[13]将GOA应用在智能电网的资源调度中,考虑多个约束条件并构造出相应的目标函数,实验表明GOA优化的电池模型优于其他对比算法。文献[14]将正交学习机制和Rosenbrock策略用来改进GOA,寻优效率得到了提升,在机器人路径规划任务上表现出更好的适用性和有效性。文献[15]提出一种基于瞪羚优化和模拟退火算法的协同作用方法以增强IIR系统识别,实验验证了混合的群智能算法在确定未知系统具有良好效果。上述文献的优化方案虽然在测试函数和应用上能取得良好性能,但还存在不足。1)随机初始化方式会导致种群分布不均匀,且缺乏多样性;2)瞪羚群体的寻优是依靠位置的随机迭代更新,引导性较弱,不能更好的均衡全局和局部搜索;3)在迭代后期,种群不断朝着最优个体聚拢,算法极易陷入局部最优。

综上,针对GOA存在的缺点,本文提出一种精英反向学习及柯西扰动引导的瞪羚优化算法(IGOA),首先利用精英反向学习策略替换原来的随机初始化方式,对种群进行初始化以此扩大候选解的搜索范围,提高原始解的质量的同时增加种群的多样性。其次提出一种二阶段非线性惯性权重,对开发阶段瞪羚种群的位置更新进行引导,

进而提高算法的寻优速度和精度。最后设计一种存活率引导的柯西扰动因子,能够在迭代后期扰动瞪羚的位置更新过程,避免算法陷入局部最优。基于12个基准测试函数对8个对比算法进行仿真对比,并利用Wilcoxon秩和检验避免实验出现的偶然性,实验结果表明,改进的瞪羚优化算法具有更优的寻优能力和稳定性。再将IGOA应用到齿轮系设计和三杆桁架设计两个工程问题中,结果表明,改进算法能够获得更优解,进一步证明了算法的有效性和可行性。

1 经典瞪羚优化算法

1.1 初始化

GOA是一种以种群为参考对象的智能算法,将随机初始化的瞪羚个体作为候选解,生成大小为 $n \times d$ 的随机矩阵,每行代表一个解,每个维度都由上下界约束。

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & x_{i,j} & \vdots & \vdots \\ x_{n,1} & x_{n,1} & \cdots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (1)$$

式中: \mathbf{X} 候选解构成的位置向量矩阵; $x_{i,j}$ 为第 i 个瞪羚个体在第 j 维度的向量位置; n 表示瞪羚的总数量; d 表示优化问题定义的维数。 $x_{i,j}$ 的初始化方式如下:

$$x_{i,j} = rand \times (UB_j - LB_j) + LB_j \quad (2)$$

式中: $rand$ 表示(0,1)的随机数; UB_j 和 LB_j 分别表示第 j 维的上界和下界。每次迭代时,每个瞪羚个体均会产生位置更新,为当前种群获得的最优解构造一个精英矩阵Elite,以便引导瞪羚下一次迭代的位置更新过程。由于捕食者捕猎时与猎物的运动方向相同,故GOA将瞪羚和捕食者均视为搜索个体。

$$\mathbf{Elite} = \begin{bmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,d-1} & x'_{1,d} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,d-1} & x'_{2,d} \\ \vdots & \vdots & x'_{i,j} & \vdots & \vdots \\ x'_{n,1} & x'_{n,2} & \cdots & x'_{n,d-1} & x'_{n,d} \end{bmatrix} \quad (3)$$

1.2 开发阶段

该阶段考虑无捕食者或有捕食者跟踪瞪羚的情况下,瞪羚在平静的吃草。利用均匀控制步长的布朗运动模拟瞪羚运动状态。

$$f_B(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x)^2}{2}\right) \quad (4)$$

式(4)将布朗运动定义为一种服从标准正态分布 $N(0,1)$ 的函数,均值 $\mu=0$,方差 $\sigma^2=1$ 。开发阶段的具体模型如式(5)所示。

$$gazelle_{i+1} = gazelle_i + s \cdot \mathbf{R} \cdot \mathbf{R}_B \cdot (\mathbf{Elite}_i - \mathbf{R}_B \cdot gazelle_i) \quad (5)$$

式中: $gazelle_{i+1}$ 为瞪羚下一次迭代的位置矢量; $gazelle_i$

当前位置矢量; s 为瞪羚的吃草速度; \mathbf{R}_B 为布朗运动对应的随机向量; \mathbf{R} 为在 $[0,1]$ 的随机向量。

1.3 勘探阶段

勘探阶段描述瞪羚从发现猎物做出反应(拍打尾巴、跺脚或跳跃),再到逃跑整个过程的运动状况。利用 Levy 飞行模拟瞪羚小步运动和偶尔跳跃行为,并假定每次迭代均会发生方向变化,迭代次数为奇数和偶数时的运动方向不同。

$$Levy(\alpha) = 0.05 \times \frac{x}{|y|^{1/\alpha}} \quad (6)$$

$$x \sim N(0, \sigma_x), \quad y \sim N(0, \sigma_y)$$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha) \sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{1+\alpha}{2}\right) \times \alpha \times 2^{\frac{\alpha-1}{2}}}\right]^{1/\alpha}$$

$$\sigma_y = 1, \quad \alpha = 1.5 \quad (7)$$

Levy 飞行的小步跟踪和长距离跳跃有助于算法实现局部搜索和对种群进行扰动,式(6)表示飞行步长, x 和 y 均服从正态分布。当瞪羚遇到捕食者时,先做出反应,以 Levy 飞行方式实现位置更新,模型如下:

$$gazelle_{i+1} = gazelle_i + S \cdot \mu \cdot \mathbf{R} \cdot \mathbf{R}_L \cdot (Elite_i - \mathbf{R}_L \cdot gazelle_i) \quad (8)$$

式中: S 为瞪羚能达到的最大速度; μ 为随迭代次数改变的方向常数,取值为 1 或 -1; \mathbf{R}_L 为 Levy 飞行对应随机向量。捕食者追逐瞪羚的位置更新方式如下:

$$gazelle_{i+1} = gazelle_i + S \cdot \mu \cdot CF \cdot \mathbf{R}_B \cdot (Elite_i - \mathbf{R}_L \cdot gazelle_i) \quad (9)$$

$$CF = \left(1 - \frac{iter}{Max_iter}\right)^{2 \frac{iter}{Max_iter}} \quad (10)$$

式中: CF 表示捕食者的累积效应; $iter$ 表示当前迭代次数; Max_iter 表示最大迭代次数。

瞪羚的存活率为 0.66,意味着捕食者的成功率 $PSRs$ 为 0.34, $PSRs$ 影响瞪羚的逃跑能力,可避免算法陷入局部最优。 $PSRs$ 的影响效应建模如下:

$$gazelle_{i+1} = \begin{cases} gazelle_i + CF[\mathbf{LB} + \mathbf{R} \cdot (\mathbf{UB} - \mathbf{LB})] \cdot \mathbf{U}, & r \leq PSRs \\ gazelle_i + [PSRs(1-r) + r] \cdot (gazelle_{r1} - gazelle_{r2}), & \text{其他} \end{cases} \quad (11)$$

$$\mathbf{U} = \begin{cases} 0, & r < 0.34 \\ 1, & \text{其他} \end{cases} \quad (12)$$

式中: r 表示在 $[0,1]$ 的随机数; $gazelle_{r1}$ 和 $gazelle_{r2}$ 表示种群内两个随机个体的位置矢量; \mathbf{U} 表示由成功率约束的二进制向量。

2 改进的瞪羚优化算法

2.1 精英反向学习策略的种群初始化

瞪羚种群初始化获得的候选解在一定程度上会影响 GOA 的收敛能力和寻优性能^[16]。考虑到随机初始产

生的位置矢量分布具有随机性和盲目性,进而增加了寻优所需的时间损耗,即降低了寻优效率。为此,本文生成反向解以扩大搜索空间,增加种群多样性并提高算法的全局搜索能力。精英反向学习策略的主要思想是根据随机初始种群原始解按照反学习方式生成种群的反向解,再将每个瞪羚个体的原始解和反向解的适应度值作比较,若反向解的适应度值更优,则将反向解保留。采取该策略能够保留更优的精英个体,规避了初始化的随机性,加快了算法的收敛速度。

$$\mathbf{x}_i = \mathbf{LB} + rand \times (\mathbf{UB} - \mathbf{LB}) \quad (13)$$

$$\mathbf{x}'_i = rand \times (\mathbf{LB} + \mathbf{UB}) + \mathbf{x}_i \quad (14)$$

$$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i, & fit(\mathbf{x}'_i) < fit(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{其他} \end{cases} \quad (15)$$

式中: \mathbf{x}_i 表示第 i 个瞪羚个体的位置初始化位置矢量; \mathbf{x}'_i 表示第 i 个个体对应的反向解; $fit(\cdot)$ 表示对个体进行适应度求解的函数。

2.2 二阶段非线性惯性权重

作为猎物,瞪羚在开发阶段主要是做以均匀步长更新的布朗运动,运动过程未受到外界干扰。但是,结合瞪羚强烈的听觉、视觉和嗅觉等特征,考虑它会朝着一个远离捕食者的安全的区域进行吃草,其运动状况需要获得一种引导。因此,从文献[17]的可调节非线性收敛因子对应的构造方式获得启发,本文在开发阶段提出了一种二阶段非线性惯性权重,进一步均衡算法全局搜索和局部搜索,提升其收敛速度和寻优能力。其主要原理是在迭代前期,瞪羚的运动状态趋于平稳,构造权值变化随着迭代次数逐渐缓慢增加,有助于瞪羚运动变化缓慢加快。在迭代后期瞪羚通过自身感官发现猎物跟踪,会以一个较快的运动状态实现位置矢量的更新,因此构造的权值变化率增大。

$$w = \begin{cases} 0.1 \cdot \sin\left(\frac{iter}{Max_iter} \cdot \frac{\pi}{4}\right) \cdot \left(\frac{iter}{k}\right)^2, & iter < \frac{2k^2}{Max_iter} \\ 0.2 - \frac{(iter - Max_iter)^2}{Max_iter^2 - k^2}, & \text{其他} \end{cases} \quad (16)$$

式中: k 表示一个可调节因子。二阶段权重变化情况如图 1 所示,可以看到 k 的取值不同,所获得的自适应曲线是不同的。本文经过多次实验和对比测试,将权重的最大值设定为 0.2,对于可调节因子 k ,取值为 400 时,迭代前期权重的变化是平缓上升的,迭代后期的权重最终仅达到 0.1 左右,未能达到设定的权重最大值,对于瞪羚种群来说,后期接近捕食者,但是较低的权重变化使瞪羚个体无法以更大的步径移动,处于一个较小的运动范围内,故算法后期难以跳出局部最优。当 k 继续增大到 500 时,可以看到迭代前期权值的变化率较小,种群相邻迭代次数间移动量变少,算法收敛速度将会变慢。因此,本文构造的权值变化中的调节因子 $k=300$,满足迭代前期有

效引导瞪羚个体的运动状态,同时符合迭代后期瞪羚遇到捕食者的快速逃离反应状态。

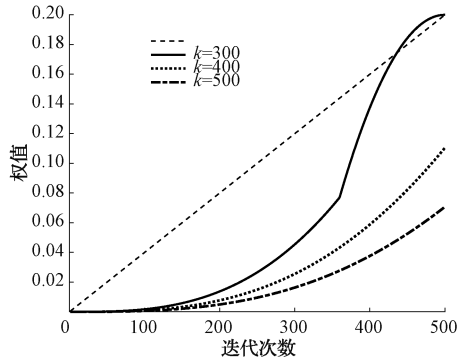


图1 不同 k 值对应的非线性惯性权重曲线
Fig.1 Nonlinear inertia weight curves corresponding to different values of k

在开发阶段,将动态变化的惯性权重放置在位置更新公式中,如式(17)所示,该部分调节了运动步长,使其更符合瞪羚运动行为的变化情况。

$$\begin{aligned} \text{stepsize} &= \mathbf{R}_n \cdot (\text{Elite}_i - \text{gazelle}_i) \\ \text{gazelle}_{i+1} &= \omega \cdot \text{gazelle}_i + s \cdot \mathbf{R} \cdot \text{stepsize} \end{aligned} \quad (17)$$

2.3 存活率引导的柯西扰动策略

在勘探阶段,瞪羚发现捕食者并做出一系列反应,从开始的 Levy 飞行小步前进和大步跳跃运动,再到后来在累积效应因子影响下做布朗运动。在此过程中,为了躲避捕食者的攻击和追逐,瞪羚群体会朝着远离捕食者的安全区域逃窜,随着迭代次数的增加,种群个体的更新位置不断趋近于群体中最优个体位置,这会使得算法陷入局部最优。GOA 跳出局部最优的方案是利用瞪羚在自然界中存活率来改变其运动状态,但是获得的效果还具有一定的提升空间。在了解柯西扰动^[18]的实际效用后,提出了一种存活率引导的柯西扰动策略,也可以称为一种变体的柯西扰动。当产生的随机数大于捕猎成功率 $PSRs$ 时,构造存活率、最优适应度值和柯西扰动乘法算式,更新瞪羚个体的位置矢量。其中,柯西扰动对应的一维标准柯西分布概率密度函数如式(18)所示,对应的曲线如图2所示。

$$f(x) = \frac{1}{\pi} \times \frac{1}{1+x^2}, \quad -\infty < x < \infty \quad (18)$$

$$\text{stepsize} = [\text{PSRs}(1-r) + r](\text{gazelle}_{r_1} - \text{gazelle}_{r_2}) \quad (19)$$

$$\text{gazelle}_{i+1} = \text{gazelle}_i + \text{stepsize} - (1 - \epsilon) \cdot \text{Top_gazelle_fit} \cdot \text{Cauchy}(0,1) \quad (20)$$

式中: ϵ 表示瞪羚的存活率,取值为 0.66; Top_gazelle_fit 表示当前最优适应度值; $\text{Cauchy}(0,1)$ 表示标准柯西分布函数。

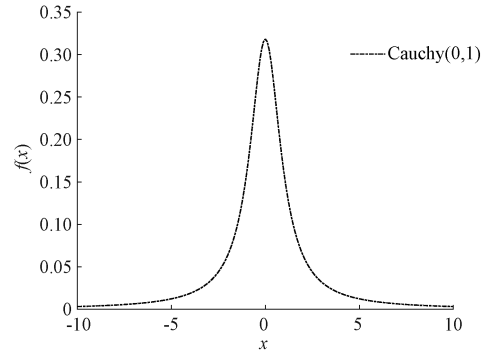


图2 柯西分布概率密度曲线
Fig.2 Cauchy distribution probability density curve

2.4 IGOA 的算法步骤及流程

改进算法的流程如图3所示。

步骤1)初始化瞪羚种群数量 N 、维度 D 、最大迭代次数 Max_iter 等参数。

步骤2)利用精英反向学习初始化策略实现种群的初始化。

步骤3)计算种群中每个瞪羚个体的适应度值,记录种群中最优个体并构造精英矩阵。

步骤4)根据式(16)计算非线性惯性权重 ω 。

步骤5)在开发阶段利用改进的位置更新公式(式(17))更新个体位置。

步骤6)执行勘探阶段的位置更新过程,并利用改进的位置更新公式(式(20))更新个体位置。

步骤7)判断是否达到最大迭代次数,若未达到,则跳转至步骤3),否则结束算法,输出全局最优瞪羚个体的位置和适应度值。

2.5 时间复杂度分析

时间复杂度分析有利于评估算法的执行效率和寻优能力。在GOA中,种群规模为 N ,最大迭代次数为 T ,经典GOA的时间复杂度分析如下:随机初始化种群的位置矢量为 $O(N)$,参数的初始化为 $O(1)$,计算种群适应度值 $O(NT)$,开发勘探位置更新过程 $O(NDT)$,GOA的时间复杂度为:

$$O(1) + O(N) + O(NT) + O(NDT) = O(NDT) \quad (21)$$

在IGOA中,基于精英反向学习策略的种群初始化替代了原来的随机初始化方法,复杂度还是 $O(N)$,提出的二阶段非线性惯性权重 $O(T)$,存活率引导的柯西扰动策略加载在勘探阶段后期的位置更新公式上,时间复杂度不变。IGOA的时间复杂度为:

$$O(1) + O(N) + O(T) + O(NT) + O(NDT) = O(NDT) \quad (22)$$

从式(21)和(22)可见,改进算法的时间复杂度与原算法在数量级上处于同一级别,故IGOA未增加算法时间复杂度。

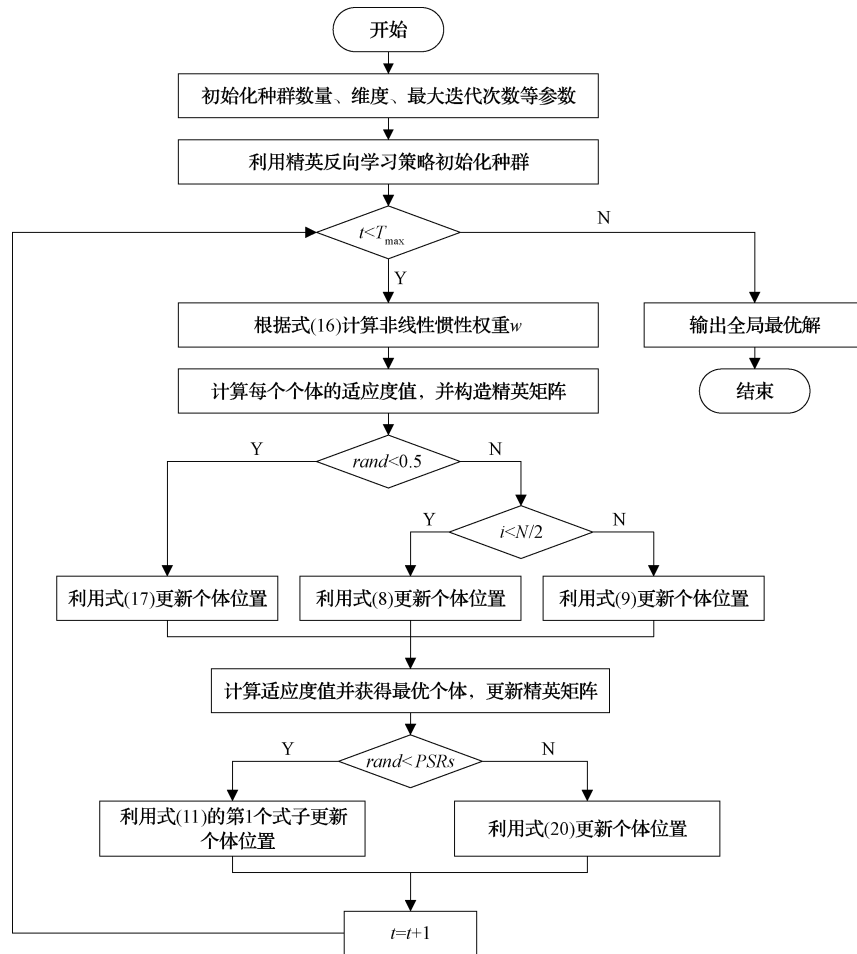


图3 IGOA 的流程

Fig. 3 Flowchart of IGOA

3 实验仿真及性能评估

3.1 实验设置

为了分析和测试 IGOA 的性能和实际效果,利用 12 个基准测试函数进行仿真实验,如表 1 所示,F1~F7 为单峰函数,能够验证算法的收敛精度和寻优速度;F8~F12 为复杂多峰函数,可用于验证算法跳出局部最优的能力。分别将 IGOA 与 4 个经典算法 GOA^[10]、GJO^[11]、SCA^[2]、MVO^[3],以及 3 个改进算法 SCChOA^[19]、ISOA^[20]、IHGWO^[21]进行实验对比。对比算法对应的参数设置如表 2 所示。

为了使各个算法处于公平的对比状态,每个算法的种群数量均设置为 30,最大迭代次数为 500。为了避免偶然性出现算法效果时好时坏的情况,选取独立运行次数为 30,统计参量为算法 30 次独立运行获得的适应度值对应的最优值、平均值和标准差,以检测算法的收敛能力和鲁棒性。

3.2 与其他群智能算法对比

采用表 1 中 12 个基准测试函数进行测试。如表 3 所示,对于单峰函数 F1~F7,IGOA 在函数 F1~F4 上均能获得理论最优值,虽然在 F4 上平均值未达到几个算法中的最优,但获得的适应度寻优精度量级均远高于其他对比算法。在 F5 上,IGOA 寻找到的最优值和均值均优于其他算法,但稳定性略低于原始的 GOA。IGOA 在 F6 和 F7 上表现优良,获得的平均值和标准差均低于其他算法,表明 IGOA 的稳定性和鲁棒性比起其他算法更为突出。

从多峰函数 F8~F12 上看,IGOA 在 F8 上取得的最优值与理论值最接近,但其平稳性相较于 SCChOA 不具有竞争力。从 F9~F11 的仿真数据上看,IGOA 在 F9 和 F11 上均能收敛到理论最优值,在 F10 上的除了与 SCChOA 的收敛性能相当之外,收敛精度和稳定性均可胜于其他算法。IGOA 在 F12 上虽然未达到理论最优,但在最优值上的精度高于他算法将近两个数量级,表征了其跳出局部最优能力相较于其他算法更优,并且表现的稳健性更强。

表 1 12 个基准测试函数
Table 1 12 benchmarking functions

序号	函数名称	最优值	区间	维度	特征
F1	Sphere	0	[-100,100]	30	Unimodal
F2	Schwefel 2.22	0	[-10,10]	30	Unimodal
F3	Schwefel 1.2	0	[-100,100]	30	Unimodal
F4	Schwefel 2.21	0	[-100,100]	30	Unimodal
F5	Rosebrock	0	[-30,30]	30	Unimodal
F6	Step	0	[-100,100]	30	Unimodal
F7	Quartic	0	[-1.28,1.28]	30	Unimodal
F8	Schwefel 2.26	-12 569.5	[-500,500]	30	Multimodal
F9	Rastrigin	0	[-5.12,5.12]	30	Multimodal
F10	Ackley	0	[-32,32]	30	Multimodal
F11	Griewank	0	[-600,600]	30	Multimodal
F12	Penalized	0	[-50,50]	30	Multimodal

表 2 算法参数设置
Table 2 Algorithm parameter setting

算法	参数设置
IGOA	$PSR_s = 0.34, S = 0.88, s = rand(0,1),$ $k = 300, \epsilon = 0.66$
GOA	$PSR_s = 0.34, S = 0.88, s = rand(0,1)$
GJO	$\beta = 1.5$
SCA	$a = 2$
MVO	$w_{max} = 1, w_{min} = 0.2$
SCChOA	$\lambda = 2$
ISOA	$\beta = 1.5, f_c = 2$
IHGWO	$a_{max} = 2, a_{min} = 0$

综上,从单峰函数和多峰函数上的测试情况来看,IGOA 比起 4 个经典算法和 SCChOA、ISOA 以及 IHGWO 3 个具有代表性的先进算法,搜索到的最优值更加接近理论最优值,总体寻优性能更好,在收敛精度和鲁棒性更好,且能在一定程度上跳出局部最优。

如图 4 所示,对不同算法在基准测试函数上进行数据的可视化,从收敛曲线上可以更直观地看到算法的收敛速度和寻优精度。从 F1 的收敛曲线可以看出,IGOA 在前 200 次迭代就可以收敛到理论最优值,经典算法与 SCChOA、ISOA、IHGWO 3 个改进算法在 300 次迭代后才能搜索到最优值,相较而言收敛速度较慢。从 F6 和 F12 曲线看出,其他算法在迭代后期均平稳的收敛于某一个小区间,表现的波动性不强,而 IGOA 可以在 100 次迭

表 3 IGOA 与其他算法对比数据

Table 3 Comparative data between IGOA and other algorithms

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F1	Best	0	1.02×10^{-59}	7.76×10^{-58}	4.53×10^{-3}	7.28×10^{-1}	2.87×10^{-268}	2.21×10^{-26}	0
	Mean	0	5.24×10^{-19}	1.67×10^{-54}	1.48×10	1.28	2.03×10^{-212}	3.62×10^{-24}	0
	Std	0	2.86×10^{-18}	2.98×10^{-54}	2.64×10	2.95×10^{-1}	0	6.66×10^{-24}	0
F2	Best	0	5.83×10^{-37}	5.65×10^{-34}	1.66×10^{-5}	5.07×10^{-1}	5.16×10^{-137}	1.68×10^{-17}	4.01×10^{-230}
	Mean	0	3.19×10^{-13}	2.99×10^{-32}	1.82×10^{-2}	2.84	4.78×10^{-108}	3.88×10^{-16}	1.77×10^{-227}
	Std	0	1.75×10^{-12}	5.47×10^{-32}	2.25×10^{-2}	6.29	2.62×10^{-107}	3.94×10^{-16}	0
F3	Best	0	7.90×10^{-11}	1.42×10^{-23}	1.56×10^3	7.98×10	8.52×10^{-291}	1.16×10^{-2}	0
	Mean	0	5.23×10^{-2}	1.06×10^{-16}	8.76×10^3	2.27×10^2	1.51×10^{-192}	1.63	0
	Std	0	2.53×10^{-1}	3.38×10^{-16}	5.72×10^3	1.05×10^2	0	1.89	0
F4	Best	0	6.48×10^{-21}	7.97×10^{-19}	1.81×10	6.38×10^{-1}	2.25×10^{-134}	2.17×10^{-4}	1.27×10^{-211}
	Mean	1.19×10^{-280}	9.43×10^{-5}	4.40×10^{-16}	3.72×10	1.98	4.77×10^{-106}	4.67×10^{-3}	3.91×10^{-209}
	Std	0	5.11×10^{-4}	8.78×10^{-16}	1.26×10	7.05×10^{-1}	2.61×10^{-105}	6.34×10^{-3}	0
F5	Best	3.62×10^{-11}	2.44×10	2.62×10	3.01×10	3.85×10	2.88×10	2.33×10	2.54×10
	Mean	9.37	2.52×10	2.78×10	2.86×10^4	5.11×10^2	2.90×10	2.92×10	2.69×10
	Std	1.26×10	4.03×10^{-1}	6.91×10^{-1}	6.51×10^4	6.98×10^2	4.52×10^{-2}	1.87×10	9.01×10^{-1}

续表

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F6	Best	4.39×10^{-10}	3.64×10^{-4}	1.98	4.58	6.12×10^{-1}	1.05	4.44×10^{-3}	5.55×10^{-5}
	Mean	7.22×10^{-3}	1.28×10^{-1}	2.70	2.56×10	1.27	1.65	5.41×10^{-1}	5.06×10^{-1}
	Std	1.34×10^{-2}	8.61×10^{-2}	3.93×10^{-1}	4.27×10	3.91×10^{-1}	2.49×10^{-1}	2.83×10^{-1}	3.59×10^{-1}
F7	Best	1.62×10^{-5}	7.97×10^{-4}	1.28×10^{-5}	5.63×10^{-3}	1.19×10^{-2}	1.00×10^{-6}	9.54×10^{-4}	8.27×10^{-6}
	Mean	8.55×10^{-5}	2.94×10^{-3}	5.29×10^{-4}	8.07×10^{-2}	2.98×10^{-2}	1.02×10^{-4}	5.03×10^{-3}	1.12×10^{-4}
	Std	7.00×10^{-5}	1.98×10^{-3}	3.46×10^{-4}	6.80×10^{-2}	1.01×10^{-2}	1.25×10^{-4}	2.94×10^{-3}	9.01×10^{-5}
F8	Best	-1.26×10^4	-1.01×10^4	-6.40×10^3	-4.58×10^3	-9.46×10^3	-5.75×10^3	-7.84×10^3	-9.39×10^3
	Mean	-1.19×10^4	-7.96×10^3	-4.34×10^3	-3.81×10^3	-7.87×10^3	-5.50×10^3	-6.74×10^3	-6.73×10^3
	Std	8.06×10^2	9.65×10^2	1.13×10^3	3.22×10^2	7.67×10^2	1.46×10^2	4.56×10^2	1.41×10^3
F9	Best	0	0	0	8.50×10^{-3}	6.72×10	0	1.05×10^{-4}	0
	Mean	0	2.32	0	3.69×10	1.13×10^2	0	1.82×10	0
	Std	0	6.27	0	2.72×10	2.70×10	0	7.72	0
F10	Best	8.88×10^{-16}	4.44×10^{-15}	4.44×10^{-15}	1.00×10^{-1}	6.42×10^{-1}	8.88×10^{-16}	8.62×10^{-14}	8.88×10^{-16}
	Mean	8.88×10^{-16}	3.01×10^{-14}	7.05×10^{-15}	1.56×10	2.40	8.88×10^{-16}	1.25×10^{-12}	1.13×10^{-15}
	Std	0	1.29×10^{-13}	1.60×10^{-15}	8.14	3.32	0	1.61×10^{-12}	9.01×10^{-16}
F11	Best	0	0	0	3.98×10^{-1}	6.67×10^{-1}	0	0	0
	Mean	0	0	0	8.95×10^{-1}	8.55×10^{-1}	0	8.11×10^{-3}	0
	Std	0	0	0	2.75×10^{-1}	7.47×10^{-2}	0	7.67×10^{-3}	0
F12	Best	4.43×10^{-6}	2.80×10^{-4}	6.99×10^{-2}	7.22×10^{-1}	1.19×10^{-2}	6.56×10^{-2}	7.57×10^{-4}	6.64×10^{-3}
	Mean	6.37×10^{-4}	5.99×10^{-3}	2.12×10^{-1}	3.82×10^4	1.81	1.28×10^{-1}	2.08×10^{-2}	3.18×10^{-2}
	Std	7.28×10^{-4}	3.43×10^{-3}	8.36×10^{-2}	1.64×10^5	1.23	5.91×10^{-2}	1.61×10^{-2}	1.45×10^{-2}

代之前开始逐渐呈现下降趋势,跳出局部最优,并在500次迭代时获得一个精度更高的适应度值。因此,从收敛曲线看出,可以更进一步说明IGOA的优良性以及改进策略的可行性和有效性。

3.3 高维情况下与其他算法对比

为了检验IGOA在高维问题上是否还能具有良好的寻优性能,本文在12个100维和200维基准测试函数上分别对算法进行实验,结果如表4、5所示。从表4可知,IGOA在100维的测试函数与30维测试函数获得的实验结果相比,两者在F1、F2、F3、F9、F10和F11获得的最优值、平均值和标准差均相等。虽然在F4、F6和F12上收敛精度在一定程度上降低了数量级,但各方面性能均能优于其他对比算法。IGOA在F5上的收敛精度更高,但是其标准差低于SCChOA,说明其稳定性还有待提高,在F7上的最优值比IHGWO和SCChOA高一个数量级,但稳定性相对较好。

由表5可知,在200维的测试函数结果中,IGOA在F1、F3、F9和F11取得理论最优值。比起其他算法,IGOA在F2、F4、F6、F7、F10和F12上能获得更高的求解精度,且标准差也均为最小,说明该算法的稳定性更强。在F5上收敛的最优值和平均值相对较好,同样平稳性不具有竞争力。

综上,在100和200维的高维情况下,IGOA除了在

F5和F8上表现的稳定性略低于SCChOA和IHGWO之外,总体上仍然能够在寻优性能上优于4个经典算法和3个改进算法,并保持自身的稳定性和鲁棒性。也进一步说明了改进算法具备更强的解决高维问题的能力。

3.4 Wilcoxon 秩和检验

衡量算法的优良性还需要从统计学的角度进一步分析,在此将IGOA与其他对比算法的寻优结果进行Wilcoxon秩和检验。该秩和检验在显著水平位为5%的条件下进行实验检测,当 $p < 0.05$ 时,表示两种算法具有显著性差异,N/A表示二者性能相当。实验设置7个对比算法,12个基准测试函数,种群规模为30,最大迭代次数为500,维度为30,独立运行次数为30。利用‘+’、‘-’、‘=’分别表示ICOA优于、劣于和等于对比算法。

如表6所示,从单峰函数F1~F7可见,IGOA仅在F7上劣于SCChOA和IHGWO,其余情况下,IGOA均能与其他对比算法产生显著性差异。从多峰函数F8~F12可见,IGOA在F9函数上劣于原始算法,以及在F10上劣于IHGWO,但在其他情况下,IGOA均能处于优于或相等的状态。因此,从统计的‘+’、‘-’、‘=’三种符号数量来看,IGOA优于或相当于其他算法的占比达到10次以上,表明IGOA相较于其他7个对比算法存在显著性差异,且在统计上的优越性能也是显著的,即认为IGOA具有更高的收敛精度。

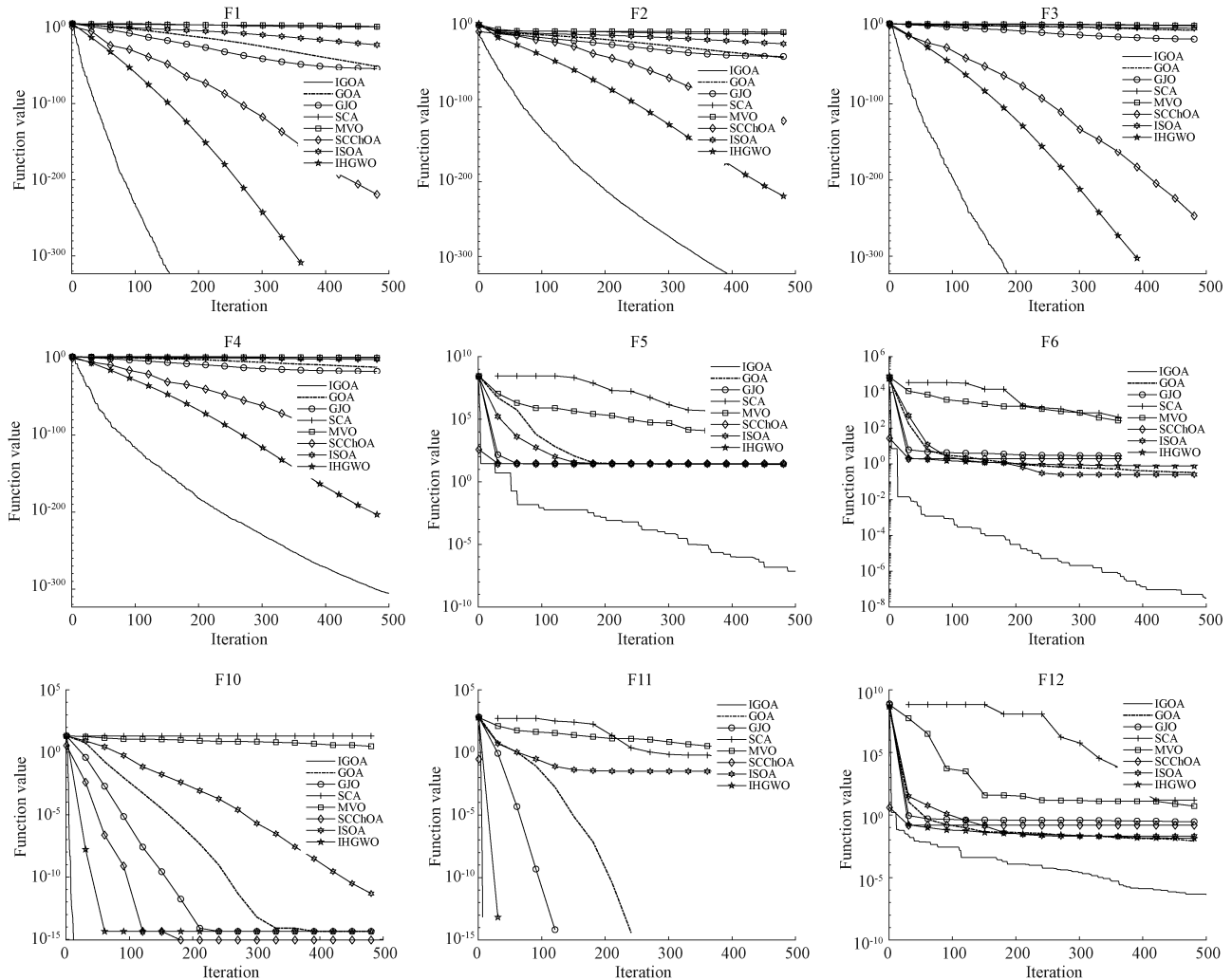


图4 不同算法在基准测试函数上的收敛曲线

Fig. 4 Convergence curves of different algorithms on benchmark test functions

表4 IGOA 和对比较算法在100维基准测试上的实验结果

Table 4 Experimental results of IGOA and comparison algorithms on a 100-dimensional benchmark test

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F1	Best	0	1.43×10^{-41}	9.64×10^{-30}	2.86×10^3	1.02×10^2	2.71×10^{-276}	5.44×10^{-9}	0
	Mean	0	6.37×10^{-8}	3.82×10^{-28}	1.10×10^4	1.56×10^2	9.80×10^{-204}	2.56×10^{-8}	0
	Std	0	3.49×10^{-7}	4.04×10^{-28}	6.94×10^3	2.61×10	0	2.37×10^{-8}	0
F2	Best	0	1.13×10^{-27}	1.23×10^{-18}	8.28×10^{-1}	3.46×10^2	1.67×10^{-132}	5.38×10^{-7}	1.95×10^{-215}
	Mean	0	2.30×10^{-6}	7.74×10^{-18}	7.47	1.15×10^{23}	5.59×10^{-95}	1.38×10^{-6}	1.32×10^{-213}
	Std	0	1.24×10^{-5}	6.41×10^{-18}	6.70	4.71×10^{23}	3.06×10^{-94}	7.77×10^{-7}	0
F3	Best	0	3.88	8.19×10^{-6}	1.20×10^5	4.93×10^4	5.93×10^{-261}	7.92×10^3	0
	Mean	0	1.47×10^3	3.11×10^{-1}	2.58×10^5	6.54×10^4	3.94×10^{-193}	2.08×10^4	0
	Std	0	2.22×10^3	9.74×10^{-1}	6.22×10^4	8.04×10^3	0	7.82×10^3	0
F4	Best	2.44×10^{-270}	5.41×10^{-13}	3.69×10^{-3}	7.77×10	4.17×10	8.12×10^{-146}	3.21	3.07×10^{-200}
	Mean	3.29×10^{-226}	7.96×10^{-3}	9.82	8.91×10	5.90×10	5.48×10^{-99}	7.94	7.90×10^{-198}
	Std	0	3.55×10^{-2}	1.25×10	2.99	8.93	2.98×10^{-98}	3.02	0
F10	Best	2.76×10^{-4}	9.60×10	9.67×10	2.06×10^7	3.26×10^3	9.90×10	9.54×10	9.52×10
	Mean								
	Std								

续表

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F5	Mean	6.40×10	9.76×10	9.82×10	1.02×10^8	7.99×10^3	9.90×10	1.22×10^2	9.78×10
	Std	4.35×10	8.54×10^{-1}	5.33×10^{-1}	4.96×10^7	5.03×10^3	4.10×10^{-3}	7.67×10	7.77×10^{-1}
	Best	5.79×10^{-8}	5.33	1.47×10	1.22×10^3	1.08×10^2	6.65	4.28	7.39
F6	Mean	1.08×10^{-1}	9.33	1.65×10	1.32×10^4	1.70×10^2	8.63	6.37	9.42
	Std	4.16×10^{-1}	1.64	9.39×10^{-1}	9.84×10^3	3.44×10	1.47	8.69×10^{-1}	1.01
	Best	1.09×10^{-5}	1.77×10^{-3}	2.43×10^{-4}	4.56×10	3.82×10^{-1}	4.63×10^{-6}	1.80×10^{-2}	1.40×10^{-6}
F7	Mean	7.38×10^{-5}	7.89×10^{-3}	1.53×10^{-3}	1.47×10^2	6.56×10^{-1}	7.74×10^{-5}	5.59×10^{-2}	9.88×10^{-5}
	Std	6.42×10^{-5}	1.32×10^{-2}	9.36×10^{-4}	7.61×10	1.70×10^{-1}	6.89×10^{-5}	2.69×10^{-2}	6.51×10^{-5}
	Best	-4.19×10^4	-1.63×10^4	-1.83×10^4	-8.47×10^3	-2.52×10^4	-1.76×10^4	-1.66×10^4	-2.40×10^4
F8	Mean	-3.33×10^4	-1.36×10^4	-9.93×10^3	-7.00×10^3	-2.30×10^4	-1.30×10^4	-1.42×10^4	-1.68×10^4
	Std	6.18×10^3	1.19×10^3	4.25×10^3	6.15×10^2	1.29×10^3	5.73×10^3	1.04×10^3	5.49×10^3
	Best	0	0	0	8.13×10	5.98×10^2	0	9.41×10	0
F9	Mean	0	8.80	7.58×10^{-15}	2.55×10^2	7.25×10^2	0	1.76×10^2	0
	Std	0	4.52×10	4.15×10^{-14}	9.62×10	5.98×10	0	4.43×10	0
	Best	8.88×10^{-16}	4.44×10^{-15}	3.64×10^{-14}	2.03	4.03	8.88×10^{-16}	1.00×10^{-5}	8.88×10^{-16}
F10	Mean	8.88×10^{-16}	1.07×10^{-13}	4.94×10^{-14}	1.82×10	8.83	8.88×10^{-16}	5.40×10^{-5}	2.31×10^{-15}
	Std	0	3.53×10^{-13}	9.67×10^{-15}	5.21	6.35	0	7.49×10^{-5}	1.77×10^{-15}
	Best	0	0	0	4.19	2.10	0	2.73×10^{-9}	0
F11	Mean	0	2.91×10^{-4}	0	9.60×10	2.47	0	3.98×10^{-3}	0
	Std	0	1.59×10^{-3}	0	9.15×10	2.17×10^{-1}	0	6.07×10^{-3}	0
	Best	2.65×10^{-5}	5.66×10^{-2}	4.73×10^{-1}	1.18×10^7	1.17×10	1.27×10^{-1}	7.76×10^{-2}	1.10×10^{-1}
F12	Mean	1.87×10^{-3}	1.47×10^{-1}	6.21×10^{-1}	2.68×10^8	2.13×10	2.06×10^{-1}	5.47×10^{-1}	1.91×10^{-1}
	Std	2.00×10^{-3}	4.60×10^{-2}	8.59×10^{-2}	1.17×10^8	6.77	7.72×10^{-2}	8.14×10^{-1}	3.27×10^{-2}

表5 IGOA 和对比较算法在 200 维基准测试上的实验结果

Table 5 Experimental results of IGOA and comparison algorithms on a 200-dimensional benchmark test

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F1	Best	0	2.66×10^{-36}	2.51×10^{-21}	1.00×10^4	1.92×10^3	4.89×10^{-245}	6.52×10^{-4}	0
	Mean	0	2.28×10^{-17}	4.69×10^{-20}	4.93×10^4	2.82×10^3	1.33×10^{-199}	5.62×10^{-3}	0
	Std	0	1.22×10^{-16}	4.29×10^{-20}	2.83×10^4	3.33×10^2	0	3.44×10^{-3}	0
F2	Best	0	2.56×10^{-24}	1.60×10^{-13}	5.04	3.42×10^{26}	3.83×10^{-127}	1.00×10^{-3}	3.51×10^{-211}
	Mean	3.40×10^{-308}	5.82×10^{-5}	4.95×10^{-13}	2.86×10	8.61×10^{72}	6.80×10^{-102}	2.20×10^{-3}	1.52×10^{-209}
	Std	0	3.12×10^{-4}	2.26×10^{-13}	1.86×10	3.02×10^{73}	3.72×10^{-101}	7.51×10^{-4}	0
F3	Best	0	1.58×10^2	6.23×10^{-1}	6.82×10^5	2.35×10^5	4.12×10^{-242}	1.13×10^5	0
	Mean	0	1.52×10^4	6.91×10^2	1.09×10^6	3.27×10^5	7.12×10^{-173}	1.64×10^5	0
	Std	0	1.46×10^4	1.93×10^3	2.35×10^5	3.17×10^4	0	2.52×10^4	0
F4	Best	5.58×10^{-241}	1.17×10^{-10}	2.61×10	9.16×10	7.68×10	1.12×10^{-127}	2.15×10	7.79×10^{-196}
	Mean	8.15×10^{-208}	4.08×10^{-1}	5.59×10	9.63×10	8.39×10	1.50×10^{-93}	3.53×10	7.56×10^{-194}
	Std	0	1.45	1.19×10	1.53	3.78	8.23×10^{-93}	6.81	0
F5	Best	2.31×10^{-5}	1.97×10^2	1.97×10^2	2.17×10^8	2.28×10^5	1.99×10^2	1.98×10^2	1.96×10^2
	Mean	1.57×10^2	1.98×10^2	1.98×10^2	5.27×10^8	3.88×10^5	1.99×10^2	3.68×10^2	1.98×10^2
	Std	6.96×10	4.01×10^{-1}	3.09×10^{-1}	1.46×10^8	1.28×10^5	4.64×10^{-3}	3.74×10^2	7.16×10^{-1}
F6	Best	7.54×10^{-6}	2.31×10	3.63×10	1.18×10^4	2.13×10^3	1.44×10	2.19×10	2.38×10
	Mean	2.50×10^{-1}	3.04×10	3.88×10	5.41×10^4	2.78×10^3	1.91×10	2.43×10	2.76×10
	Std	1.02	3.06	1.12	2.65×10^4	3.79×10^2	2.90	1.24	1.64
F7	Best	1.22×10^{-6}	2.12×10^{-3}	3.26×10^{-4}	6.32×10^2	4.15	2.93×10^{-6}	1.19×10^{-1}	7.92×10^{-6}
	Mean	7.65×10^{-5}	4.67×10^{-2}	2.55×10^{-3}	1.50×10^3	5.72	1.48×10^{-4}	2.80×10^{-1}	1.35×10^{-4}

续表

函数	指标	IGOA	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F8	Std	6.04×10^{-5}	1.45×10^{-1}	2.16×10^{-3}	4.86×10^2	9.55×10^{-1}	1.28×10^{-4}	8.75×10^{-2}	1.07×10^{-4}
	Best	-8.38×10^4	-2.33×10^4	-2.44×10^4	-1.16×10^4	-4.46×10^4	-3.29×10^4	-2.54×10^4	-4.04×10^4
	Mean	-5.80×10^4	-1.91×10^4	-1.54×10^4	-9.83×10^3	-4.06×10^4	-1.84×10^4	-2.21×10^4	-2.82×10^4
F9	Std	1.87×10^4	1.39×10^3	7.32×10^3	8.07×10^2	2.26×10^3	1.40×10^4	1.75×10^3	1.12×10^4
	Best	0	0	0	1.12×10^2	1.74×10^3	0	5.28×10^2	0
	Mean	0	1.54×10^{-1}	3.03×10^{-14}	5.31×10^2	1.93×10^3	0	7.36×10^2	0
F10	Std	0	8.41×10^{-1}	7.86×10^{-14}	2.32×10^2	1.15×10^2	0	1.08×10^2	0
	Best	8.88×10^{-16}	4.44×10^{-15}	2.70×10^{-12}	9.85	1.98×10	8.88×10^{-16}	5.03×10^{-3}	8.88×10^{-16}
	Mean	8.88×10^{-16}	4.14×10^{-7}	1.13×10^{-11}	1.93×10	2.05×10	8.88×10^{-16}	1.46×10^{-2}	3.14×10^{-15}
F11	Std	0	2.26×10^{-6}	7.21×10^{-12}	3.17	1.74×10^{-1}	0	1.27×10^{-2}	1.74×10^{-15}
	Best	0	0	0	1.09×10^2	2.05×10	0	5.46×10^{-4}	0
	Mean	0	2.53×10^{-4}	3.70×10^{-18}	3.39×10^2	2.70×10	0	9.39×10^{-3}	0
F12	Std	0	1.21×10^{-3}	2.03×10^{-17}	1.33×10^2	3.30	0	1.66×10^{-2}	0
	Best	1.48×10^{-5}	2.27×10^{-1}	6.61×10^{-1}	7.21×10^8	6.95×10	1.39×10^{-1}	7.20×10^{-1}	3.07×10^{-1}
	Mean	1.39×10^{-3}	1.31	7.75×10^{-1}	1.38×10^9	3.47×10^3	2.24×10^{-1}	3.67	3.75×10^{-1}
	Std	2.20×10^{-3}	3.70	4.75×10^{-2}	3.69×10^8	5.71×10^3	8.68×10^{-2}	1.86	3.49×10^{-2}

表 6 IGOA 与其他算法的秩和检验对比结果

Table 6 Results of the rank sum test comparing IGOA with other algorithms

函数	GOA	GJO	SCA	MVO	SCChOA	ISOA	IHGWO
F1	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
F2	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
F3	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
F4	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F5	5.26×10^{-4}	1.89×10^{-4}	3.02×10^{-11}	3.02×10^{-11}	4.98×10^{-11}	3.37×10^{-4}	2.84×10^{-4}
F6	2.44×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.46×10^{-10}	4.69×10^{-8}
F7	3.02×10^{-11}	3.35×10^{-8}	3.02×10^{-11}	3.02×10^{-11}	8.19×10^{-1}	-3.02×10^{-11}	3.87×10^{-1}
F8	4.50×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.34×10^{-11}
F9	5.58×10^{-3}	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
F10	1.59×10^{-13}	2.43×10^{-13}	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	1.61×10^{-1}
F11	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	3.45×10^{-7}	NaN
F12	7.38×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.09×10^{-10}	3.02×10^{-11}
+/=/-	10/1/1	10/2/0	12/0/0	12/0/0	8/3/1	12/0/0	6/4/2

4 工程应用

4.1 齿轮系设计问题

齿轮系设计问题^[22]属于机械工程领域中经典问题之一,是一种在边界约束条件下,使输出轴角速度与输入轴角速度之比最小化,即传动比最小。优化变量为齿轮的齿数 $(n_A, n_B, n_C, n_D) = (x_1, x_2, x_3, x_4)$,其数学模型如下:

$$\min f(x) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2 \quad (23)$$

$$12 \leq x_i \leq 60 \quad (i = 1, 2, 3, 4)$$

实验设置种群数量为 30,最大迭代次数为 500,独立运行 30 次。选取 GOA、GJO、SCA、MVO 4 种经典算法和 SCChOA、ISOA、IHGWO 3 种改进算法与 IGOA 进行

对比。

8 个智能算法对目标函数进行寻优以实现齿轮系的设计,统计值取独立运行 30 次获得的最优适应度值。如表 7 所示,从统计结果可以看出,所提出的 IGOA 在几个对比算法中表现最优,居于首位,比原算法低一个数量级。虽然 ISOA、IHGWO 与 IGOA 处于同一数量级,但是收敛值更小,这进一步体现了改进算法寻优效果更佳。

4.2 三杆桁架设计问题

三杆桁架设计问题^[23]描述的是基于应力、挠度和屈曲以及杆的横截面积的约束的条件,使目标函数轻杆结构的质量最小化。由于三杆桁架的结构具有对称性,故其中有两个杆的横截面积变量取值相同,因此该约束问题构造为 3 个非线性约束和两个决策变量,其具体的数学模型如下:

表7 不同算法解决齿轮系设计问题的对比结果

Table 7 Comparative results of different algorithms for solving gear train design problems

算法	x_1	x_2	x_3	x_4	最优值	排名
IGOA	42.762 9	12.444 5	15.211 8	30.682 1	1.50×10^{-17}	1
GOA	59.997 0	12.480 2	37.476 8	54.032 0	1.61×10^{-16}	4
GJO	44.117 0	16.031 0	12.000 0	30.222 7	4.62×10^{-15}	6
SCA	30.431 9	14.591 4	12.000 0	39.879 8	6.68×10^{-12}	8
MVO	53.298 8	14.205 7	12.000 0	22.167 7	1.21×10^{-15}	5
SCChOA	27.423 0	12.000 0	12.000 0	36.395 6	2.79×10^{-12}	7
ISOA	56.643 3	30.920 6	13.883 6	52.528 8	3.36×10^{-17}	2
IHGWO	36.342 5	14.084 2	21.520 6	57.805 3	4.39×10^{-17}	3

$$\begin{aligned} \min f(x) &= L(x_2 + 2\sqrt{2}x_1) \\ \text{s. t. } g_1 &= \frac{x_2}{2x_1x_2 + \sqrt{2}x_1^2}P - \sigma \leq 0 \\ g_2 &= \frac{x_2 + \sqrt{2}x_1}{2x_1x_2 + \sqrt{2}x_1^2}P - \sigma \leq 0 \\ g_3 &= \frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0 \\ 0 &\leq x_i \leq 1, \quad i = 1, 2, \end{aligned} \quad (24)$$

$$L = 100 \text{ cm}, P = 2 \text{ kN/cm}^2, \sigma = 2 \text{ kN/cm}^2$$

式中： L 表示桁架的挠度； P 表示屈曲载荷； σ 表示应力。在求解约束问题时，构造惩罚函数并将其加入到目标函数中，使其变为无约束问题。本文采用外点惩罚函数对该约束问题进行转化，获得的新的目标函数如下：

$$\begin{aligned} H(i) &= \begin{cases} 0, & g(i) \leq 0 \\ 1, & \text{其他} \end{cases} \\ Z &= \sum_{i=1}^n [\lambda \cdot (g(i))^2 \cdot H(i)] \\ f(x) &= (2\sqrt{2}x_1 + x_2) + Z \end{aligned} \quad (25)$$

式中： λ 为惩罚常数； Z 为惩罚值； n 为约束条件的总个数。

同样，采用8个算法进行实验对比，设置种群数为30，最大迭代次数为500，独立运行次数为30，并统计独立运行30次获得的最优适应度值。如表8所示，IGOA虽然获得的最优适应度值竞争力微弱，但是相比于其他算法更优，进一步证明了改进算法的可行性与有效性。

表8 不同算法解决三杆桁架设计问题的对比结果

Table 8 Comparative results of different algorithms for solving the three-bar truss design problem

算法	x_1	x_2	最优值	排名
IGOA	0.788 6	0.408 4	263.895 85	1
GOA	0.788 4	0.409 1	263.896 04	2
GJO	0.790 2	0.404 0	263.897 79	6
SCA	0.783 7	0.422 6	263.939 86	8
MVO	0.789 2	0.406 7	263.896 06	3
SCChOA	0.792 0	0.399 1	263.922 73	7
ISOA	0.788 2	0.409 5	263.896 20	4
IHGWO	0.788 7	0.408 3	263.896 28	5

5 结论

本文针对瞪羚优化算法存在的缺点，提出了一种精英反向学习及柯西扰动引导的瞪羚优化算法。通过融合精英反向学习、二阶段非线性惯性权重和变体柯西扰动3种策略，实现对原始算法的改进。利用12个基准测试函数和Wilcoxon秩和检验证明了IGOA与其他算法对比收敛精度、速度更优，并且具有跳出局部最优的能力。为了探测IGOA是否在高维度情况下的寻优能力，分别在100维和200维的基准测试函数上进行实验检测，结果表明改进算法具有解决高维问题的能力。最后，利用两个工程优化问题案例验证了IGOA在解决实际问题时能获得更好的寻优效果。

IGOA总体性能优越，进一步的改进以及对其他复杂问题的求解可能还需要进行实际测试和对比验证。在未来的研究中，尝试利用更加普适性和鲁棒性的改进策略优化IGOA，使其表现出更好的寻优能力和泛化能力，并在后续阶段考虑将其应用在神经网络的超参数优化、资源调度等实际问题中。

参考文献

- [1] NITISH C, MUHAMMAD A M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications [J]. Expert Systems With Applications, 2022, 198(5):116924.
 - [2] ZUBAIR M R, ABDULLAH K, ROZAIDA G, et al. A new multi sine-cosine algorithm for unconstrained optimization problems [J]. PloS One, 2021, 16(8): e0255269-e0255269.
 - [3] ABUALIGAH L. Multi-verse optimizer algorithm: A comprehensive survey of its results, variants, and applications [J]. Neural Computing and Applications, 2020, 32(16):1-21.
 - [4] 赵建萍, 张达敏. 混合策略改进的草原土拨鼠算法及其应用 [J]. 国外电子测量技术, 2023, 42(11): 129-142.
- ZHAO J P, ZHANG D M. Hybrid strategy improved

- prairie dog algorithm and its application[J]. *Foreign Electronic Measurement Technology*, 2023, 42(11): 129-142.
- [5] 刘杰,曹静,赵昕. 基于 OOB-GWO-SVR 的风电机组齿轮箱故障预警[J]. *电子测量与仪器学报*, 2022, 36(12):97-105.
LIU J, CAO J, ZHAO X. Wind turbine gearbox fault warning based on OOB-GWO-SVR[J]. *Journal of Electronic Measurement and Instrumentation*, 2022, 36(12):97-105.
- [6] 邢燕好,于昊,张佳,等. 基于粒子群参数优化的 O-VMD 数据处理方法研究[J]. *仪器仪表学报*, 2023, 44(4):304-313.
XING Y H, YU H, ZHANG J, et al. Research on the O-VMD thickness measurement data processing method based on particle swarm optimization [J]. *Chinese Journal of Scientific Instrument*, 2023, 44(4):304-313.
- [7] 万怡华,张雪梅. 混合多策略改进蜣螂算法的避障路径规划[J]. *电子测量技术*, 2024, 47(2):69-78.
WAN Y H, ZHANG X M. Obstacle avoidance path planning of hybrid multi-strategy improved dung beetle optimizer [J]. *Electronic Measurement Technology*, 2024, 47(2):69-78.
- [8] BEISHENALIEVA A, YOO S J. Multi-objective 3-D UAV movement planning in wireless sensor networks using bioinspired swarm intelligence [J]. *IEEE Internet of Things Journal*, 2022, DOI: 10.1109/JIOT.2022.3231302.
- [9] ZHAO Y, LIU Y, WU Z, et al. Improving sparrow search algorithm for optimal operation planning of hydrogen-electric hybrid microgrids considering demand response[J]. *Symmetry*, 2023, 15(4):919.
- [10] AGUSHAKA J O, EZUGWU A E, ABUALIGAH L. Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer [J]. *Neural Computing and Applications*, 2022, 35(5): 4099-4131.
- [11] LAITH A, ALI D, ABU R Z. Orthogonal learning rosenbrock's direct rotation with the gazelle optimization algorithm for global optimization [J]. *Mathematics*, 2022, 10(23):4509-4509.
- [12] GUNUPUDI K C S, REDDI K K, KURICHETI K V P, et al. Deep residual convolutional neural network: An efficient technique for intrusion detection system[J]. *Expert Systems With Applications*, 2023, 238(4):121912.
- [13] HASANIEN H M, ALSALEH I, TOSTADO-VÉLIZ M, et al. Optimal parameters estimation of lithium-ion battery in smart grid applications based on gazelle optimization algorithm [J]. *Energy*, 2023, 285:129509.
- [14] 邓毅,廖秋丽. 基于改进瞪羚优化算法的机器人路径规划[J]. *机械设计与制造工程*, 2024, 53(1):51-54.
DENG Y, LIAO Q L. The robot path planning based on improved gazelle optimization algorithm [J]. *Machine Design and Manufacturing Engineering*, 2024, 53(1):51-54.
- [15] SERDAR E, DAVUT I. Enhancing IIR system identification: Harnessing the synergy of gazelle optimization and simulated annealing algorithms [J]. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 2023, 5:100225.
- [16] 李江华,王鹏晖,李伟. 一种混合多策略改进的麻雀搜索算法[J]. *计算机工程与科学*, 2024, 46(2):303-315.
LI J H, WANG P H, LI W. A hybrid multi-strategy improved sparrow search algorithm [J]. *Computer Engineering & Science*, 2024, 46(2):303-315.
- [17] 音凌一,向凤红,毛剑琳. 改进灰狼优化算法在特征栅格地图上的路径规划[J]. *机械科学与技术*, 2023, 42(9):1516-1526.
YIN L Y, XIANG F H, MAO J L. Improved grey wolf optimization algorithm for path planning on feature grid maps [J]. *Mechanical Science and Technology for Aerospace Engineering*, 2023, 42(9): 1516-1526.
- [18] 付接递,李振东,郭辉. 基于教与学和逐维柯西变异的鲸鱼优化算法[J]. *计算机工程与科学*, 2023, 45(5): 940-950.
FU J D, LI ZH D, GUO H. A whale optimization algorithm based on teaching and learning and dimensional Cauchy mutation [J]. *Computer Engineering & Science*, 2023, 45(5):940-950.
- [19] 高大唤,梁宏涛,杜军威,等. 改进黑猩猩优化算法的测试数据生成研究[J]. *计算机工程与应用*, 2022, 58(23):83-93.
GAO D H, LIANG H T, DU J W, et al. Research on test data generation for improved chimpanzee optimization algorithm [J]. *Computer Engineering and Applications*, 2022, 58(23):83-93.
- [20] 秦维娜,张达敏,尹德鑫,等. 一种基于非线性惯性权重的海鸥优化算法[J]. *小型微型计算机系统*, 2022, 43(1):10-14.
QIN W N, ZHANG D M, YIN D X, et al. Seagull optimization algorithm based on nonlinear inertia weight[J]. *Journal of Chinese Computer Systems*, 2022, 43(1):10-14.
- [21] ZHANG C, LIU H, ZHOU Q, et al. Improved

- hybrid grey wolf optimization algorithm based on dimension learning-based hunting search strategy [J]. IEEE Access, 2023, 11: 13738-13753.
- [22] PRAKASHBHAI P M, GHOSHAL S K, UDAI A D. A novel comprehensive learning Rao algorithm for engineering optimization problems[J]. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2022,45:47.
- [23] LI X, ZHANG S, YANG L, et al. Neighborhood-search-based enhanced multi-strategy collaborative artificial bee colony algorithm for constrained

engineering optimization[J]. Soft Computing, 2023, 27(19):13991-14017.

作者简介

班云飞, 硕士研究生, 主要研究方向为群智能算法优化、网络入侵检测等。

E-mail:3201536579@qq.com

张达敏(通信作者), 博士, 教授, 硕士生导师, 主要研究方向为智能计算、智能算法优化、认知无线电等。

E-mail:dmzhang@gzu.edu.cn