DOI:10.19651/j.cnki.emt.2518563

多策略改进的人工旅鼠算法及工程应用*

杨 原 陈明霞 陆俊良 严一踔
(桂林理工大学广西高校先进制造与自动化技术重点实验室 桂林 541006)

摘 要: 人工旅鼠算法是一种新提出的元启发式算法,能够通过模拟旅鼠的 4 种不同行为,有效探索复杂的搜索空间,但该算法仍存在过早收敛、探索不足、缺乏鲁棒性以及易陷入局部最优。针对上述问题,本文提出一种多策略改进的人工旅鼠算法,首先,引入 Halton 序列实现初始种群均匀分布,以提升全局搜索能力;其次,结合精英池策略与惯性权重,避免搜索过度依赖最优个体,增强种群的跳跃搜索能力,从而抑制早熟收敛;最后,在算法迭代后期引入非线性权重的黄金正弦策略,与觅食行为相结合,以提高局部搜索的精度与稳定性。为验证改进算法的性能,本文选取 CEC2017 测试函数集进行实验,并采用 Wilcoxon 秩和检验进行统计分析。实验结果表明,改进后的算法在收敛速度、寻优精度以及稳定性方面均优于 5 种对比算法,相较于原算法平均值误差降低了 27.36%,标准差平均降低了 36.99%,在 3 个工程优化问题中,改进后的算法均取得了最小目标函数值,优于对比算法,表现出较好的适用性和优越性。

关键词: 人工旅鼠算法;Halton 序列;精英池;黄金正弦;行星轮系设计;多盘离合器制动器;工业制冷系统
中图分类号: TP301.6;TN7 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Multi-strategy improved artificial lemming algorithm and engineering applications

Yang Yuan Chen Mingxia Lu Junliang Yan Yichuo
(Key Laboratory of Advanced Manufacturing and Automation Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin University of Technology, Guilin 541006, China)

Abstract: The artificial lemming algorithm is a newly proposed metaheuristic method that simulates four distinct behaviors of lemmings to effectively explore complex search spaces. However, it still suffers from premature convergence, limited exploration, lack of robustness, and susceptibility to local optima. To address these limitations, a multi-strategy improved artificial lemming algorithm is proposed. First, the Halton sequence is employed to generate a uniformly distributed initial population, enhancing global search capability. Second, an elite pool strategy combined with inertia weights is introduced to reduce excessive reliance on the best individuals and to improve the population's ability to jump across the search space, thereby suppressing premature convergence. Finally, a nonlinear weighted golden sine strategy, combined with foraging behavior, is incorporated in the later stages of iteration to enhance the precision and stability of local search. To verify the performance of the improved algorithm, experiments are conducted on the CEC2017 benchmark function set, and statistical analysis is performed using the Wilcoxon rank-sum test. Experimental results show that the improved algorithm outperforms five comparative algorithms in terms of convergence speed, optimization accuracy, and stability. Compared to the original algorithm, the improved version achieves an average error reduction of 27.36% and a reduction of 36.99% in the average standard deviation. In three engineering optimization problems, the improved algorithm obtains the minimum objective function values in all cases, demonstrating better applicability and superiority over the comparative methods.

Keywords: artificial lemming algorithm; Halton sequence; elite pool; gold sine; planetary gear train design; multiple disk clutch brake; industrial refrigeration system

0 引 言

伴随科学技术的持续迭代与工程系统复杂性的显著提升,优化问题日益呈现出高维度、非线性、多峰性及约束条件耦合等特征,对求解方法的鲁棒性与全局搜索能力提出了更高的要求。元启发式算法因其简单性、灵活性以及较

收稿日期:2025-04-10
* 基金项目:广西科技计划项目-广西重点研发计划(桂科 AB22080093)、梧州市中央引导地方科技发展资金项目(202201001)、国家自然科学基金(61863009)项目资助

强的鲁棒性,广泛应用于复杂的优化问题中。近年来,提出的元启发式算法主要有蜣螂优化算法(dung beetle optimizer,DBO)[1]、蛇优化算法(snake optimizer,SO)[2]、遗传算法(genetic algorithm, GA)[3]、粒子群优化算法(particle swarm optimization, PSO)[4]、哈里斯鹰算法(harris hawks optimization,HHO)[5]和灰狼优化算法(grey wolf optimizer, GWO)[6]等具代表性的元启发式优化算法,凭借其良好的适应性和较低的计算复杂度,被广泛应用于图像分割[7]、电力系统控制[8]和电力负荷预测[9]等领域。

人工旅鼠算法(artificial lemming algorithm,ALA)是由 Xiao 等[10]于 2025 年新提出的一种受生物启发的元启发式算法。该算法通过对旅鼠长距离迁徙、挖洞、觅食和躲避捕食者 4 种不同的行为进行数学建模,从而实现对复杂搜索空间的有效探索,并采用了一种能量降低机制,能够动态平衡勘探与开发过程,增强了算法跳出局部最优解的能力。但在求解实际工程设计优化问题时,ALA 算法仍存在迭代初期收敛速度慢和迭代后期易陷入局部最优解的问题,并且算法对参数设置敏感,过度依赖单一最优个体作为搜索引导,导致种群过早聚集于局部最优解,限制了全局搜索能力,且降低了找到全局最优解的概率。为解决上述问题,并进一步提高优化算法的性能,本文提出了一种结合精英池和黄金正弦多策略改进的人工旅鼠算法(elite pool and golden sine-improved artificial lemming algorithm,EGALA)。首先,通过引入 Halton 序列初始化方法来改善初始种群的分布,使得种群在解空间中的分布更加均匀;其次,采用精英池策略与惯性权重结合的方法,增强算法在局部区域的跳跃性搜索能力,避免了算法迭代过程中因过早收敛而陷入局部最优。为了进一步提升搜索精度和全局探索能力,本文还采用非线性权重的黄金正弦策略与觅食行为相结合,确保在迭代的后期阶段能够有效控制算法的收敛过程,提高局部搜索的精度与稳定性。为验证所提改进算法的有效性,本文使用 CEC2017 测试函数中的 29 个标准测试函数进行实验,并通过 Wilcoxon 秩和检验进行了统计分析。实验结果表明,改进后的 EGALA 算法相较于 ALA 算法在测试函数上表现出更高的寻优精度和收敛速度,并在多个工程设计优化问题中也体现了其优越的性能。

1　人工旅鼠算法

在自然界中,旅鼠表现出独特的群体协作与自适应机制,涵盖了长距离迁移、挖洞、觅食以及逃避捕食者等不同行为。

1.1　初始化种群

人工旅鼠算法是一种基于种群的算法,要求在进入迭代过程之前初始化所有旅鼠的位置。设旅鼠种群候选解集是一个由 N 行和 Dim 列组成的矩阵,所有候选解位于上下界之间,如式(1)所示。

$$\vec{\boldsymbol{Z}}=\begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,Dim-1} & z_{1,Dim} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,Dim-1} & z_{2,Dim} \\ \cdots & \cdots & z_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{N-1,1} & z_{N-1,1} & \cdots & z_{N-1,Dim-1} & z_{N-1,Dim} \\ z_{N,1} & z_{N,2} & \cdots & z_{N,Dim-1} & z_{N,Dim} \end{bmatrix} \tag{1}$$

式中:i 为旅鼠种群中的第 i 个体,j 是问题变量的维数。通过式(2)计算每个维度的决策变量:

$$\vec{\boldsymbol{Z}}=LB_j+rand\times(UB_j-LB_j) \tag{2}$$

式中:$rand$ 是范围在[0,1]之间的随机值,LB_j 表示求解问题第 j 维的下限,UB_j 是求解问题第 j 维的上限。

1.2　长距离迁移行为

当食物资源稀缺时,旅鼠会进行随机的长距离迁移,旅鼠通过探索当前位置与种群中随机个体的位置来搜索潜在资源丰富的区域,从而寻求更有利的生存条件和资源。并且,旅鼠迁移的方向和距离受多种因素影响,表现出不确定性和动态变化,对旅鼠长距离迁移搜索食物资源的行为进行数学建模,公式如式(3)所示。

$$\vec{\boldsymbol{Z}}_i(t+1)=\vec{\boldsymbol{Z}}_{best}(t)+F\times\overrightarrow{\boldsymbol{BM}}\times(\vec{\boldsymbol{R}}\times(\vec{\boldsymbol{Z}}_{best}(t)-\vec{\boldsymbol{Z}}_i(t))+(\boldsymbol{1}-\vec{\boldsymbol{R}})\times(\vec{\boldsymbol{Z}}_i(t)-\vec{\boldsymbol{Z}}_a(t))) \tag{3}$$

式中:$\vec{\boldsymbol{Z}}_i(t+1)$ 表示第 i 个旅鼠在 $t+1$ 次迭代中的位置,$\vec{\boldsymbol{Z}}_{best}(t)$ 表示当前搜索到的旅鼠位置最优解,$\vec{\boldsymbol{Z}}_i(t)$ 表示第 i 个搜索旅鼠的当前位置,$\vec{\boldsymbol{Z}}_a(t)$ 表示从旅鼠种群中随机选择的搜索个体的位置,a 是介于 1 和 N 之间的整数索引。F 用于改变搜索方向,$\overrightarrow{\boldsymbol{BM}}$ 表示随机数向量,布朗运动具有动态和均匀的步长,可以增强旅鼠全局搜索能力,$\vec{\boldsymbol{R}}$ 是一个 $1\times Dim$ 的向量,其元素是均匀分布在区间[−1,1]中的随机数,参数 F、$\overrightarrow{\boldsymbol{BM}}$ 和 $\vec{\boldsymbol{R}}$ 具体公式如式(4)所示。

$$\begin{cases} F=\begin{cases}1, & \lfloor 2\times rand+1\rfloor=1\\ -1, & \lfloor 2\times rand+1\rfloor=2\end{cases} \\ f_{BM}(x;0,1)=\dfrac{1}{\sqrt{2\pi}}\times\exp\left(-\dfrac{x^2}{2}\right) \\ \vec{\boldsymbol{R}}=2\times rand(1,Dim)-1 \end{cases} \tag{4}$$

1.3　挖洞行为

旅鼠会基于当前最优解位置和旅鼠种群中随机个体的位置进行扰动,在栖息地随机挖掘新的洞穴,新隧道能够帮助旅鼠迅速逃避捕食者的威胁,还能提高其觅食效率。位置更新公式如式(5)所示。

$$\vec{\boldsymbol{Z}}_i(t+1)=\vec{\boldsymbol{Z}}_i(t)+F\times L\times(\vec{\boldsymbol{Z}}_{best}(t)-\vec{\boldsymbol{Z}}_b(t)) \tag{5}$$

式中:$\vec{\boldsymbol{Z}}_b$ 表示从旅鼠种群中随机选择的搜索个体,b 是介于 1 和 N 之间的随机整数索引。L 的计算公式如式(6)所示。

$$L=rand\times\left(1+\sin\left(\frac{t}{2}\right)\right) \tag{6}$$

1.4　觅食行为

旅鼠通常会在栖息地内设定一个相对较小的觅食区域，该区域的大小取决于食物的丰富度和可获取性，为了最大化食物的摄取，旅鼠会在该觅食区域内随机游动，这一行为提高了算法在后期找到局部最优解的概率，采用螺旋缠绕机制模拟旅鼠在栖息地洞穴内觅食的行为，位置更新公式如式(7)和(8)所示。

$$\vec{\boldsymbol{Z}}_i(t+1)=\vec{\boldsymbol{Z}}_{best}(t)+F\times spiral\times rand\times\vec{\boldsymbol{Z}}_i(t) \tag{7}$$

$$spiral=radius\times(\sin(2\times\pi\times rand)+\cos(2\times\pi\times rand)) \tag{8}$$

式中：$spiral$ 表示觅食行为过程中随机搜索的螺旋形状，$radius$ 表示觅食行为范围的半径，即当前旅鼠位置与最优解之间的欧几里得距离，计算公式如式(9)所示。

$$radius=\sqrt{\sum_{j=1}^{Dim}(z_{best,j}(t)-z_{i,j}(t))^2} \tag{9}$$

1.5　躲避天敌行为

洞穴作为旅鼠的避难所，在发现敌人时，旅鼠会利用其卓越的奔跑能力迅速返回洞穴。同时，旅鼠还会通过实施欺骗性动作来躲避捕食者的追击。相应的数学模型如式(10)所示。

$$\vec{\boldsymbol{Z}}_i(t+1)=\vec{\boldsymbol{Z}}_{best}(t)+F\times G\times Levy(Dim)\times(\vec{\boldsymbol{Z}}_{best}(t)-\vec{\boldsymbol{Z}}_i(t)) \tag{10}$$

$$G=2\times\left(1-\frac{t}{T_{\max}}\right) \tag{11}$$

式中：G 是旅鼠的逃避系数，$Levy(\cdot)$ 是莱维飞行函数，逃避能力会随着迭代次数的增加而降低，$T_{\max}$ 表示最大迭代次数。莱维飞行的步长公式如式(12)所示。

$$\begin{cases}Levy(x)=0.01\times\dfrac{u\times\sigma}{|v|^{\frac{1}{\beta}}}\\ \sigma=\left(\dfrac{\Gamma(1+\beta)\times\sin\left(\dfrac{\pi\beta}{2}\right)}{\Gamma\left(\dfrac{1+\beta}{2}\right)\times\beta\times2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}}\end{cases} \tag{12}$$

式中：u 和 v 是区间[0,1]中的随机值，β 是常数，取值为 1.5。

1.6　从勘探到开发的过渡

在人工旅鼠算法中，4 种搜索策略与旅鼠的能量水平相关。初期，旅鼠进行勘探行为，寻找有利区域；后期，旅鼠则专注于局部开发，以优化全局解。为平衡勘探与开发，设计了一个逐渐减少的能量因子。能量因子的计算公式如式(13)所示。

$$E(t)=4\times\arctan\left[1-\frac{t}{T_{\max}}\right]\times\ln\left(\frac{1}{rand}\right) \tag{13}$$

当 $E(t)>1$ 能量充足时，旅鼠执行勘探迁移行为或挖洞行为；否则，能量不足时，执行觅食行为或逃避捕食者行为。

2　改进人工旅鼠算法

2.1　Halton 序列初始化种群

原始人工旅鼠算法采用随机初始化种群方法，在高维复杂解空间中易导致个体分布不均，降低了搜索速度和全局寻优能力。常用的混沌初始化方法有 Logistic、Tent、Cubic、Chebyshev 及 Circle 映射，不同的混沌映射虽具有一定遍历性，但存在边界聚集、点间相关性强、易陷入局部区域等问题，难以保证全局覆盖性。针对上述问题，本文引入 Halton 低差异序列替代随机初始化，以增强种群在解空间中的均匀分布性，并提高人工旅鼠算法的收敛速度和精度。二维 Halton 序列的生成过程通过选取两个不同的质数 p 作为基础量，利用这些基础量对每个正整数 n 进行分解和组合，得到一组均匀且不重复的点[11]。具体数学模型如式(14)、(15)和(16)所示。

$$n=\sum_{i=0}^{m}b_i\cdot p^i=b_m\cdot p^m+\cdots+b_1\cdot p^1+b_0 \tag{14}$$

$$\theta(n)=b_0p^{-1}+b_1p^{-2}+\cdots+b_mp^{-m-1} \tag{15}$$

$$H(n)=[\theta_1(n),\theta_2(n)] \tag{16}$$

式中：n 为[1,N]区间内任意整数，b_i 为常数，$\theta(n)$ 是定义的序列函数，$H(n)$ 为二维 Halton 序列。为了验证 Halton 序列初始化的优越性，本文将其与传统的随机初始化方法进行对比，种群规模设为 100，取值范围为[0,1]。图 1 和图 2 展示了随机初始化与 Halton 序列初始化的种群分布。

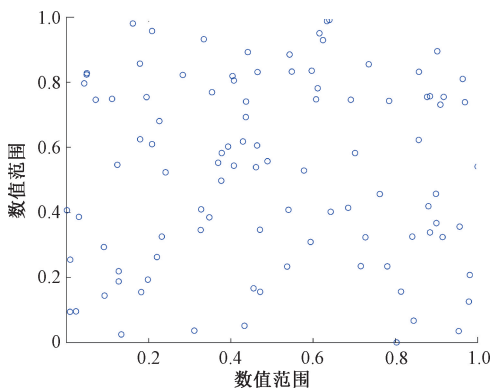


图 1　种群随机初始化个体分布图

Fig. 1　Randomly initialized population distribution map

对比图 1 和 2 并进行分析，结果表明，Halton 序列生成的初始种群在解空间中的分布更加均匀，有助于提升种群多样性与早期搜索效率，提高了种群初始解方案的质量。

2.2　精英池策略

原始的 ALA 算法依赖单一最优个体作为搜索引导，这种方式容易导致种群在局部最优解附近聚集，限制了搜索的广度，并降低了全局最优解的发现概率。为进一步提升人工旅鼠算法在搜索过程中的效率与种群多样性，本文提出了一种动态精英池和惯性权重结合的引导策略，通过

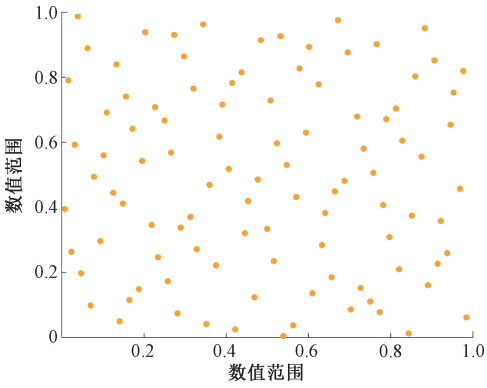


图 2　使用 Halton 序列产生的初始种群分布图

Fig. 2　Halton sequence initialized populationdistribution map

综合多个优质个体的信息,增强个体在优质解空间附近的跳跃性搜索能力,从而避免过早收敛陷入局部最优的问题,精英池策略如图 3 所示。

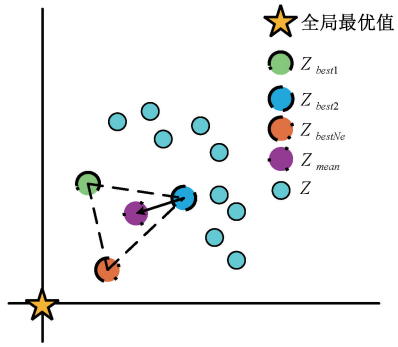


图 3　精英池策略示意图

Fig. 3　Schematic diagram of elite pool strategy

精英池由当前精英种群大小 N_e 和 Z^t_{mean} 动态确定,N_e 为当前种群中适应度最优的个体个数,Z^t_{mean} 为最优个体的加权平均位置,并且 Z^t_{mean} 代表了当前优势子群体的演化趋势,有助于引导整个群体进行有效的局部开发,为增强种群搜索的多样性,并有效利用精英池中的多个优质解,本文在算法迭代过程中设定种群规模为 $N=30$,并使用动态调整因子 α 控制精英池大小,其中每 α 个个体将从精英池中随机选择一个最优解,以确定该子群体个体的搜索方向,引导其搜索方向,经过多次实验,当 $\alpha=10$ 时,寻优结果最好。精英池策略公式如式(17)所示。

$$\begin{cases} Elitet=[Z^t_{best1},Z^t_{best2},\cdots,Z^t_{bestN_e},Z^t_{mean}] \\ N_e=\left(\dfrac{N}{\alpha}\right) \\ Z^t_{mean}=\dfrac{1}{N_e}\sum\limits_{i=1}^{N_e}Z^t_{besti} \end{cases} \tag{17}$$

式中:N 为种群大小,α 为动态调整因子,$Z^t_{bestN_e}$ 为当前种群适应度中第 N_e 个最优个体。将精英池策略与惯性权重相结合,以优化算法全局搜索与局部搜索之间的平衡,当惯性权重较大时,旅鼠倾向于进行更广泛的全局搜索,而较小时则偏向于局部搜索。改进后的旅鼠挖洞行为位置更新策略如式(18)所示。

$$\vec{\boldsymbol{Z}}_i(t+1)=\begin{cases} \vec{\boldsymbol{Z}}_i(t)+w\times(\vec{\boldsymbol{Z}}_{elite}(t)-\vec{\boldsymbol{Z}}_i(t))+c\times \\ (\vec{\boldsymbol{Z}}_b-\vec{\boldsymbol{Z}}_i(t)), \quad c\geqslant 0.5 \\ \vec{\boldsymbol{Z}}_i(t)+F\times L\times(\vec{\boldsymbol{Z}}_{elite}(t)- \\ \vec{\boldsymbol{Z}}_b(t)), \quad c<0.5 \end{cases}$$

$$\vec{\boldsymbol{Z}}_{elite}\in\{Z^t_{best1},Z^t_{best2},\cdots,Z^t_{bestN_e},Z^t_{mean}\} \tag{18}$$

式中:$\vec{\boldsymbol{Z}}_{elite}(t)$ 为精英池中随机选择的个体,c 为范围在[0,1]之间的随机数,w 为惯性权重,w 计算公式如式(19)所示。

$$w=w_{\max}-\frac{t}{T_{\max}}(w_{\max}-w_{\min}) \tag{19}$$

式中:$w_{\max}$ 和 $w_{\min}$ 分别是最大和最小惯性权重,经过多次实验,当 $w_{\max}=0.9$ 和 $w_{\min}=0.2$ 时,该算法的寻优性能最好。通过采用精英池策略和惯性权重的方法,有效避免了算法过度依赖单一最优个体,并且无法跳出局部最优值的问题,提升了种群在搜索空间中的多样性以及算法的收敛速度与寻优精度。

2.3　非线性权重的黄金正弦策略

人工旅鼠算法在迭代后期会出现解的空间分布趋于集中,使得种群多样性下降和种群收敛过快,导致错失寻得最优解的机会和陷入局部最优等问题。为解决上述问题并提升局部搜索精度,本文在人工旅鼠算法的后期阶段采用非线性权重的黄金正弦搜索策略[12],黄金正弦算法具有优秀的寻优能力和鲁棒性,通过引入黄金分割系数对搜索过程进行动态调节,确保在潜在最优解区域内进行探索,提升了算法搜索效率,其数学模型如式(20)所示。

$$X_i^{t+1}=X_i^t\times|\sin(R_1)|+R_2\times\sin(R_1)\,|\,c_1\times P_i^t-c_2\times X_i^t\,| \tag{20}$$

式中:X_i^t 表示第 i 个体在第 t 次迭代的位置,P_i^t 为当前全局最优解;R_1 和 R_2 分别为取值属于[0,2π]和[0,π]的随机数,分别为控制移动幅度和方向的随机数;c_1 和 c_2 为黄金分割系数,用来缩小搜索空间,引导个体向最优值收敛;c_1、c_2 和 τ 的数学公式如式(21)所示。

$$\begin{cases} c_1=-\pi+(1-\tau)\times 2\pi \\ c_2=-\pi+2\pi\times\tau \\ \tau=\dfrac{\sqrt{5}-1}{2} \end{cases} \tag{21}$$

采用基于非线性下降的自适应权重项 w_2,该权重控制个体对当前位置信息与全局最优解的响应程度,为防止算法在迭代中后期快速收敛陷入局部最优,在黄金正弦公式中引入非线性权重,w_2 计算公式如式(22)所示。

$$w_2=(w_{\max 2}-w_{\min 2})\times\left[1-\sin\left(\frac{\pi}{2}\times\frac{t}{T_{\max}}\right)\right]+w_{\min 2} \tag{22}$$

式中:$w_{\max 2}$ 和 $w_{\min 2}$ 分别为最大与最小权重,t 为当前迭代

次数，T_{max} 为最大迭代次数。将非线性权重的黄金正弦策略与旅鼠觅食行为结合，改进后的觅食行为位置更新公式如式(23)所示。

$$\vec{\boldsymbol{Z}}_i^{t+1} = (1-w_2)\vec{\boldsymbol{Z}}_i^t \times |\sin(R_1)| + w_2 \times R_2 \times \sin(R_1) \times |c_1 \times P_i^t - c_2 \times \vec{\boldsymbol{Z}}_i^t| \tag{23}$$

式中：$\vec{\boldsymbol{Z}}_i^{t+1}$ 为更新后的旅鼠位置，当权重参数取 $w_{max2}=1$ 和 $w_{min2}=0.1$ 时，算法寻优结果表现出更高的收敛速度与解的稳定性。权重 w_2 使得算法在迭代初期能够摆脱当前局部最优，从而进行全局搜索，在后期能够有效地引导搜索过程，平衡了全局探索和局部开发的能力，提升了算法的寻优精度与鲁棒性。

2.4　EGALA 算法流程

多策略改进后的人工旅鼠算法 EGALA 流程如下所示：

步骤 1)设置种群大小 N，最大迭代次数 T_{max} 等；

步骤 2)使用 Halton 序列初始化旅鼠种群；

步骤 3)计算适应度值，确定最优个体，更新精英池；

步骤 4)根据旅鼠能量水平，执行相应的行为，切换概率根据不同策略更新个体位置；

步骤 5)进行边界约束，更新旅鼠种群适应度；

步骤 6)判断新旧个体适应度值，保留最优解；

步骤 7)使用 ALA 算法自带的莱维飞行策略，强化搜索，更新个体位置；

步骤 8)利用贪婪策略选择最优个体，并更新精英池；

步骤 9)检查是否满足终止条件。如果满足，则跳转至下一步骤，否则，返回步骤 4)继续迭代；

步骤 10)结束算法迭代过程，输出最终最优解。

2.5　改进算法的复杂度分析

假设旅鼠种群规模为 N，搜索空间的维度为 D，最大迭代次数为 T，则 ALA 算法的时间复杂度为 $O(T\times N\times D)$。在 EGALA 算法中，采用了 Halton 序列初始化种群，其时间复杂度为 $O(N\times D)$。精英池策略涉及对当前种群中最优个体的选择和加权平均计算，在 T 轮迭代中，EGALA 算法虽改进了挖洞行为与结合了黄金正弦策略的觅食行为，但 N 个旅鼠在每次迭代中只进行一次位置更新，并未增加新的循环体，所以时间复杂度与 ALA 算法一致，为 $O(T\times N\times D)$。适应度计算阶段与 ALA 算法一致，时间复杂度为 $O(T\times N)$，莱维飞行变异仅对最优个体进行动态调整，时间复杂度为 $O(T)$。因此，EGALA 算法的总体时间复杂度为 $O(T\times N\times D)$，与原始的 ALA 算法保持一致，说明改进的 EGALA 算法计算负担并未增加，且执行效率保持不变。

3　仿真实验结果与分析

3.1　测试函数

为了验证 EGALA 算法的性能，本文采用 IEEE CEC2017 标准测试集进行实验测试，测试函数如表 1 所示。CEC2017 测试集包括 29 个测试函数，其中 F1、F3 属于单峰测试函数，用于检验算法的收敛精度；F4～F10 属于多峰测试函数，主要用于测试算法跳出局部最优的能力；F11～F20 和 F21～F30 分别为混合函数和复合函数，这两类函数组成复杂，并包含额外的偏置和权重，进一步增加了算法优化的难度。将 EGALA 算法与其他优化算法进行对比测试，对比算法包括：DBO[13]、HHO[5]、GWO[6]、改进正弦算法引导的蜣螂优化算法(multi-strategy adaptive dragonfly optimization，MSADBO)[14] 和 ALA。为了确保实验结果的可靠性，每个算法的测试均独立运行 30 次，统一设定种群数量为 $N=30$，最大迭代次数为 $T_{max}=500$，记录每次实验的平均值和标准差，最后使用 Friedman 检验分析，进一步验证本文提出的 EGALA 算法有效性及其性能优越性。实验结果如表 2 所示。

3.2　实验环境及结果分析

所有算法在相同的实验环境下运行，实验所使用的硬件配置为：NVIDIA GeForce MX350 GPU，内存 16 GB，操作系统为 Windows 10，算法编程基于 MATLAB R2014。

由表 2 可以看出，在单峰函数 F1 和 F3 中，EGALA 算法相较于其他对比算法展现了出色的寻优精度。对于多峰函数 F4～F10，EGALA 算法依旧具有较强的稳定性和全局搜索能力，在 F6 测试函数中已十分接近理论最优值表现优异，尽管在 F5、F9 和 F10 上稍微逊色于 ALA 算法与 GWO 算法，但整体寻优精度仍优于其他对比算法。在混合函数 F11～F20 和复合函数 F21～F30 的测试中，EGALA 算法同样表现出色，尤其是 F12、F15、F18、F19 和 F21 中，EGALA 算法在精度和稳定性方面均优于其他对比算法，展现了其在应对复杂优化问题时的优越性。尽管 EGALA 算法在某些函数，如 F22 和 F26 上的表现有所波动，但根据 Friedman 检验结果，EGALA 算法排在第 1 位，ALA 算法排在第 2 位，MSADBO 算法排在第 3 位，GWO 算法排在第 4 位，DBO 算法排在第 5 位，HHO 算法排在第 6 位，可知 EGALA 算法整体表现最优。

综上所述，EGALA 算法在所有测试函数中的综合性能最为优秀，EGALA 算法不仅在单峰函数中具有优异的收敛性能，而且在多峰函数的求解中能够有效避免局部最优，保持较高的全局搜索能力，对于混合函数和复合函数，EGALA 算法在应对复杂优化问题时也展现了出色的全局搜索能力和良好的稳定性。

为了更直观地对比各算法的收敛过程，本文绘制了收敛曲线对比图，展示了 EGALA 算法与其他优化算法在不同测试函数上的表现，实验参数设置与前述一致。对比算法包括：DBO 算法、HHO 算法、GWO 算法、MSADBO 算法和 ALA 算法。图 4 清晰地展示了各算法在 $Dim=30$ 的维度为下，分别独立运行 30 次后的平均适应度收敛曲线图。

表 1　CEC2017 函数集

Table 1　CEC2017 function set

函数	名称	维度	范围	最优值
F1	Shifted and Rotated Bent Cigar Function	30	[−100,100]	100
F3	Shift and Rotated Zakharov Function	30	[−100,100]	300
F4	Shift and Rotated Rosenbrock′s Function	30	[−100,100]	400
F5	Shift and Rotated Rastrigin′s Function	30	[−100,100]	500
F6	Shift and Rotated Expanded Scaffer′s F6 Function	30	[−100,100]	600
F7	Shift and Rotated Lunacek Bi-Rastrigin′s Function	30	[−100,100]	700
F8	Shift and Rotated Non-Continuous Rastrigin′s Function	30	[−100,100]	800
F9	Shift and Rotated Levy Function	30	[−100,100]	900
F10	Shift and Rotated Schwefel′s Function	30	[−100,100]	1 000
F11	Hybrid function 1 (M=3)	30	[−100,100]	1 100
F12	Hybrid function 2 (M=3)	30	[−100,100]	1 200
F13	Hybrid function 3 (M=3)	30	[−100,100]	1 300
F14	Hybrid function 4 (M=4)	30	[−100,100]	1 400
F15	Hybrid function 5 (M=4)	30	[−100,100]	1 500
F16	Hybrid function 6 (M=4)	30	[−100,100]	1 600
F17	Hybrid function 6 (M=5)	30	[−100,100]	1 700
F18	Hybrid function 6 (M=5)	30	[−100,100]	1 800
F19	Hybrid function 6 (M=5)	30	[−100,100]	1 900
F20	Hybrid function 6 (M=6)	30	[−100,100]	2 000
F21	Composition function 1 (M=3)	30	[−100,100]	2 100
F22	Composition function 2 (M=3)	30	[−100,100]	2 200
F23	Composition function 3 (M=4)	30	[−100,100]	2 300
F24	Composition function 4 (M=4)	30	[−100,100]	2 400
F25	Composition function 5 (M=5)	30	[−100,100]	2 500
F26	Composition function 6 (M=5)	30	[−100,100]	2 600
F27	Composition function 7 (M=6)	30	[−100,100]	2 700
F28	Composition function 8 (M=6)	30	[−100,100]	2 800
F29	Composition function 9 (M=3)	30	[−100,100]	2 900
F30	Composition function 10 (M=3)	30	[−100,100]	3 000

表 2　EGALA 算法与对比算法在 CEC2017 测试函数上的结果对比

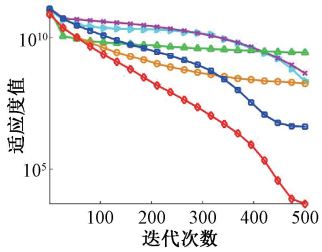
Table 2　Comparison of EGALA and benchmark algorithms on CEC2017 test functions

函数	指标	DBO	HHO	GWO	MSADBO	ALA	EGALA
F1	平均值	2.72×10^{8}	5.05×10^{8}	2.77×10^{9}	1.26×10^{8}	2.61×10^{6}	$\mathbf{5.28\times10^{3}}$
	标准差	2.11×10^{8}	3.25×10^{8}	1.59×10^{9}	9.63×10^{7}	2.37×10^{6}	$\mathbf{5.47\times10^{3}}$
F3	平均值	9.47×10^{4}	5.58×10^{4}	6.15×10^{4}	6.90×10^{4}	2.99×10^{4}	$\mathbf{1.05\times10^{4}}$
	标准差	3.45×10^{4}	7.49×10^{3}	1.26×10^{4}	9.65×10^{3}	1.04×10^{4}	$\mathbf{3.99\times10^{3}}$
F4	平均值	6.54×10^{2}	7.10×10^{2}	6.61×10^{2}	5.41×10^{2}	5.23×10^{2}	$\mathbf{5.05\times10^{2}}$
	标准差	1.13×10^{2}	8.97×10^{1}	1.05×10^{2}	3.34×10^{1}	2.70×10^{1}	$\mathbf{1.56\times10^{1}}$
F5	平均值	7.57×10^{2}	7.66×10^{2}	6.39×10^{2}	6.98×10^{2}	$\mathbf{6.12\times10^{2}}$	6.19×10^{2}
	标准差	7.05×10^{1}	3.51×10^{1}	4.78×10^{1}	3.82×10^{1}	$\mathbf{3.35\times10^{1}}$	3.42×10^{1}
F6	平均值	1.04×10^{5}	6.68×10^{2}	6.15×10^{2}	6.24×10^{2}	6.15×10^{2}	$\mathbf{6.03\times10^{2}}$
	标准差	3.79×10^{4}	7.16×10^{0}	5.84×10^{0}	1.04×10^{1}	8.46×10^{0}	$\mathbf{3.66\times10^{0}}$
F7	平均值	1.03×10^{3}	1.30×10^{3}	9.22×10^{2}	1.05×10^{3}	9.09×10^{2}	$\mathbf{8.59\times10^{2}}$
	标准差	9.77×10^{1}	5.89×10^{1}	5.16×10^{1}	5.86×10^{1}	4.93×10^{1}	$\mathbf{3.96\times10^{1}}$
F8	平均值	1.03×10^{3}	9.81×10^{2}	9.20×10^{2}	9.82×10^{2}	9.14×10^{2}	$\mathbf{9.11\times10^{2}}$
	标准差	4.61×10^{1}	2.43×10^{1}	4.09×10^{1}	4.24×10^{1}	3.62×10^{1}	$\mathbf{2.59\times10^{1}}$
F9	平均值	6.62×10^{3}	8.39×10^{3}	2.76×10^{3}	4.93×10^{3}	$\mathbf{2.33\times10^{3}}$	2.40×10^{3}
	标准差	2.15×10^{3}	1.03×10^{3}	1.36×10^{3}	2.06×10^{3}	$\mathbf{1.01\times10^{3}}$	1.36×10^{3}

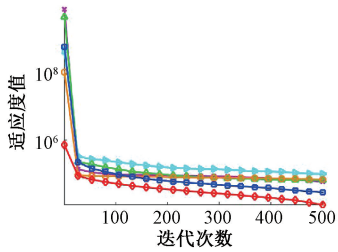
表 2(续)

Table 2 (continued)

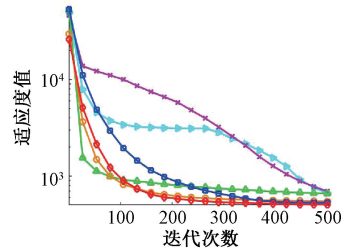
函数	指标	DBO	HHO	GWO	MSADBO	ALA	EGALA
F10	平均值	6.54×10^{3}	6.23×10^{3}	$\mathbf{4.82\times10^{3}}$	6.97×10^{3}	5.99×10^{3}	5.83×10^{3}
	标准差	1.22×10^{3}	9.39×10^{2}	1.03×10^{3}	1.32×10^{3}	$\mathbf{7.35\times10^{2}}$	8.86×10^{2}
F11	平均值	2.21×10^{3}	1.63×10^{3}	2.26×10^{3}	1.50×10^{3}	1.26×10^{3}	$\mathbf{1.21\times10^{3}}$
	标准差	1.29×10^{3}	1.96×10^{2}	1.01×10^{3}	1.12×10^{2}	5.29×10^{1}	$\mathbf{3.33\times10^{1}}$
F12	平均值	7.73×10^{7}	6.90×10^{7}	6.12×10^{7}	1.21×10^{7}	2.69×10^{6}	$\mathbf{9.42\times10^{5}}$
	标准差	1.25×10^{8}	5.78×10^{7}	4.05×10^{7}	7.10×10^{6}	2.02×10^{6}	$\mathbf{1.09\times10^{6}}$
F13	平均值	1.86×10^{7}	1.18×10^{6}	2.36×10^{7}	7.91×10^{5}	3.42×10^{4}	$\mathbf{1.66\times10^{4}}$
	标准差	3.95×10^{7}	7.39×10^{5}	6.74×10^{7}	1.38×10^{6}	2.66×10^{4}	$\mathbf{1.70\times10^{4}}$
F14	平均值	4.77×10^{5}	2.01×10^{6}	5.34×10^{5}	2.11×10^{5}	1.96×10^{3}	$\mathbf{1.61\times10^{3}}$
	标准差	7.89×10^{5}	1.93×10^{6}	8.74×10^{5}	1.53×10^{5}	4.50×10^{2}	$\mathbf{5.85\times10^{1}}$
F15	平均值	8.26×10^{5}	1.03×10^{5}	8.21×10^{5}	4.42×10^{4}	2.13×10^{4}	$\mathbf{3.57\times10^{3}}$
	标准差	3.01×10^{6}	4.70×10^{4}	1.46×10^{6}	2.15×10^{4}	1.35×10^{4}	$\mathbf{1.51\times10^{3}}$
F16	平均值	3.50×10^{3}	3.72×10^{3}	2.69×10^{3}	3.07×10^{3}	2.74×10^{3}	$\mathbf{2.42\times10^{3}}$
	标准差	5.43×10^{2}	4.29×10^{2}	3.47×10^{2}	3.83×10^{2}	3.00×10^{2}	$\mathbf{1.94\times10^{2}}$
F17	平均值	2.59×10^{3}	2.77×10^{3}	2.20×10^{3}	2.37×10^{3}	2.17×10^{3}	$\mathbf{2.02\times10^{3}}$
	标准差	2.64×10^{2}	3.30×10^{2}	2.13×10^{2}	2.07×10^{2}	1.78×10^{2}	$\mathbf{1.71\times10^{2}}$
F18	平均值	5.01×10^{6}	4.61×10^{6}	3.22×10^{6}	2.14×10^{6}	8.05×10^{4}	$\mathbf{2.61\times10^{4}}$
	标准差	1.35×10^{7}	6.13×10^{6}	6.76×10^{6}	2.24×10^{6}	5.29×10^{4}	$\mathbf{1.53\times10^{4}}$
F19	平均值	3.98×10^{6}	1.64×10^{6}	1.48×10^{6}	4.31×10^{4}	2.69×10^{4}	$\mathbf{3.31\times10^{3}}$
	标准差	8.84×10^{6}	1.00×10^{6}	2.85×10^{6}	4.71×10^{4}	2.79×10^{4}	$\mathbf{1.10\times10^{3}}$
F20	平均值	2.83×10^{3}	2.84×10^{3}	2.57×10^{3}	2.63×10^{3}	2.54×10^{3}	$\mathbf{2.44\times10^{3}}$
	标准差	2.12×10^{2}	2.44×10^{2}	2.18×10^{2}	2.13×10^{2}	2.26×10^{2}	$\mathbf{1.80\times10^{2}}$
F21	平均值	2.55×10^{3}	2.59×10^{3}	2.40×10^{3}	2.50×10^{3}	2.42×10^{3}	$\mathbf{2.38\times10^{3}}$
	标准差	4.30×10^{1}	4.50×10^{1}	3.39×10^{1}	5.44×10^{1}	3.05×10^{1}	$\mathbf{2.10\times10^{1}}$
F22	平均值	5.33×10^{3}	6.61×10^{3}	4.36×10^{3}	$\mathbf{2.37\times10^{3}}$	6.55×10^{3}	2.59×10^{3}
	标准差	2.55×10^{3}	1.79×10^{3}	1.98×10^{3}	4.47×10^{1}	2.26×10^{3}	$\mathbf{1.11\times10^{3}}$
F23	平均值	2.98×10^{3}	3.30×10^{3}	2.80×10^{3}	2.85×10^{3}	2.79×10^{3}	$\mathbf{2.74\times10^{3}}$
	标准差	7.68×10^{1}	1.37×10^{2}	5.41×10^{1}	3.74×10^{1}	3.72×10^{1}	$\mathbf{2.83\times10^{1}}$
F24	平均值	3.15×10^{3}	3.51×10^{3}	2.96×10^{3}	3.03×10^{3}	2.95×10^{3}	$\mathbf{2.92\times10^{3}}$
	标准差	7.77×10^{1}	1.68×10^{2}	6.74×10^{1}	3.55×10^{1}	3.73×10^{1}	$\mathbf{2.64\times10^{1}}$
F25	平均值	2.98×10^{3}	3.00×10^{3}	3.00×10^{3}	2.94×10^{3}	2.91×10^{3}	$\mathbf{2.89\times10^{3}}$
	标准差	5.47×10^{1}	3.03×10^{1}	5.21×10^{1}	2.38×10^{1}	1.49×10^{1}	$\mathbf{1.40\times10^{1}}$
F26	平均值	6.69×10^{3}	7.97×10^{3}	5.04×10^{3}	5.40×10^{3}	4.98×10^{3}	$\mathbf{3.96\times10^{3}}$
	标准差	1.26×10^{3}	1.04×10^{3}	5.12×10^{2}	1.03×10^{3}	$\mathbf{4.51\times10^{2}}$	8.95×10^{2}
F27	平均值	3.34×10^{3}	3.64×10^{3}	3.27×10^{3}	3.26×10^{3}	3.24×10^{3}	$\mathbf{3.23\times10^{3}}$
	标准差	7.19×10^{1}	2.90×10^{2}	3.54×10^{1}	2.54×10^{1}	2.97×10^{1}	$\mathbf{2.12\times10^{1}}$
F28	平均值	3.48×10^{3}	3.51×10^{3}	3.49×10^{3}	3.30×10^{3}	3.50×10^{3}	$\mathbf{3.24\times10^{3}}$
	标准差	2.16×10^{2}	9.86×10^{1}	1.33×10^{2}	2.56×10^{1}	7.60×10^{2}	$\mathbf{2.15\times10^{1}}$
F29	平均值	4.49×10^{3}	5.19×10^{3}	4.03×10^{3}	4.24×10^{3}	4.04×10^{3}	$\mathbf{3.69\times10^{3}}$
	标准差	3.61×10^{2}	5.09×10^{2}	2.34×10^{2}	2.95×10^{2}	2.17×10^{2}	$\mathbf{2.12\times10^{2}}$
F30	平均值	4.52×10^{6}	1.31×10^{7}	1.03×10^{7}	8.84×10^{5}	7.55×10^{4}	$\mathbf{2.06\times10^{4}}$
	标准差	9.92×10^{6}	1.92×10^{7}	8.45×10^{6}	5.95×10^{5}	1.07×10^{5}	$\mathbf{1.39\times10^{4}}$
Friedman 平均排名		4.900 0	5.383 3	3.666 7	3.533 3	2.316 7	**1.200 0**
最终排名		5	6	4	3	2	**1**



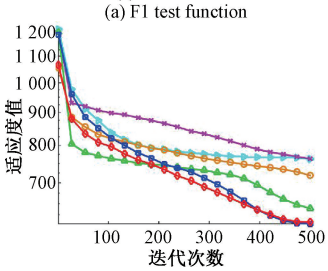
(a) F1测试函数
(a) F1 test function



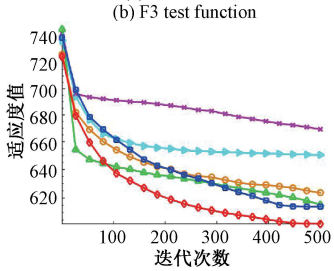
(b) F3测试函数
(b) F3 test function



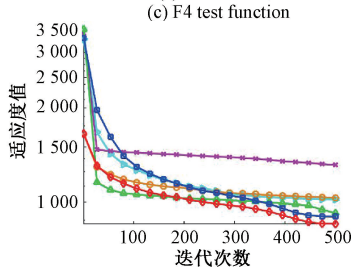
(c) F4测试函数
(c) F4 test function



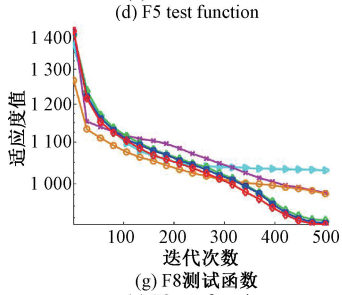
(d) F5测试函数
(d) F5 test function



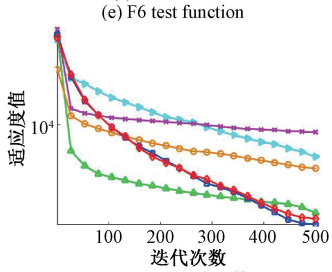
(e) F6测试函数
(e) F6 test function



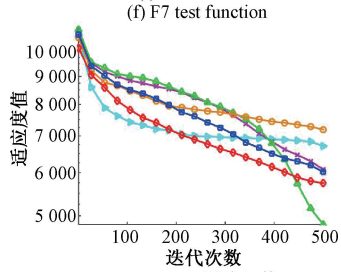
(f) F7测试函数
(f) F7 test function



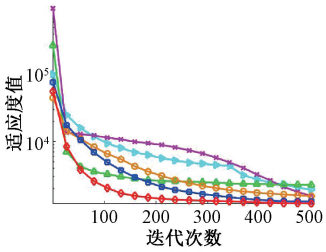
(g) F8测试函数
(g) F8 test function



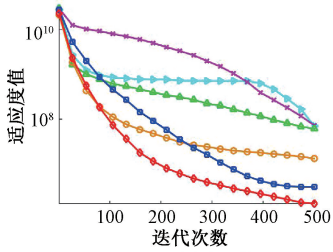
(h) F9测试函数
(h) F9 test function



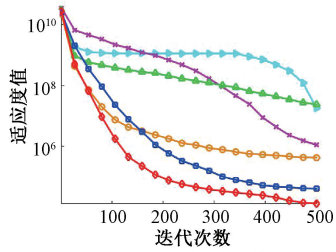
(i) F10测试函数
(i) F10 test function



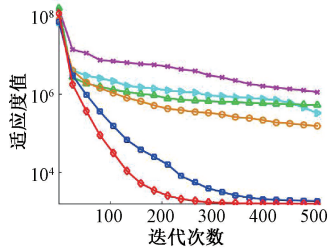
(j) F11测试函数
(j) F11 test function



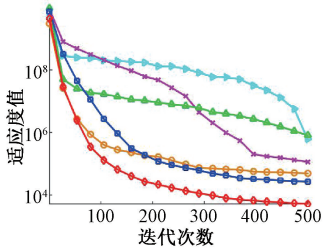
(k) F12测试函数
(k) F12 test function



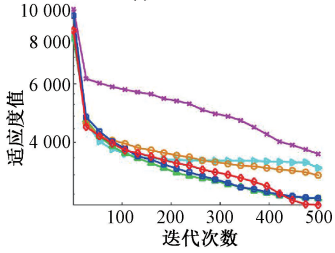
(l) F13测试函数
(l) F13 test function



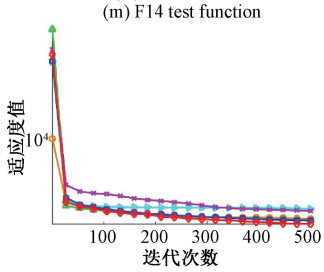
(m) F14测试函数
(m) F14 test function



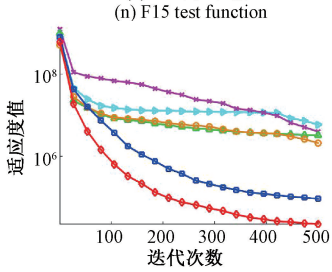
(n) F15测试函数
(n) F15 test function



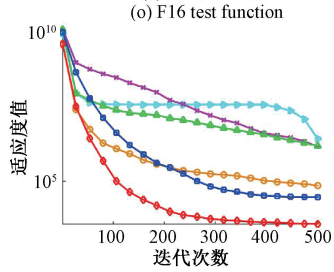
(o) F16测试函数
(o) F16 test function



(p) F17测试函数
(p) F17 test function



(q) F18测试函数
(q) F18 test function



(r) F19测试函数
(r) F19 test function

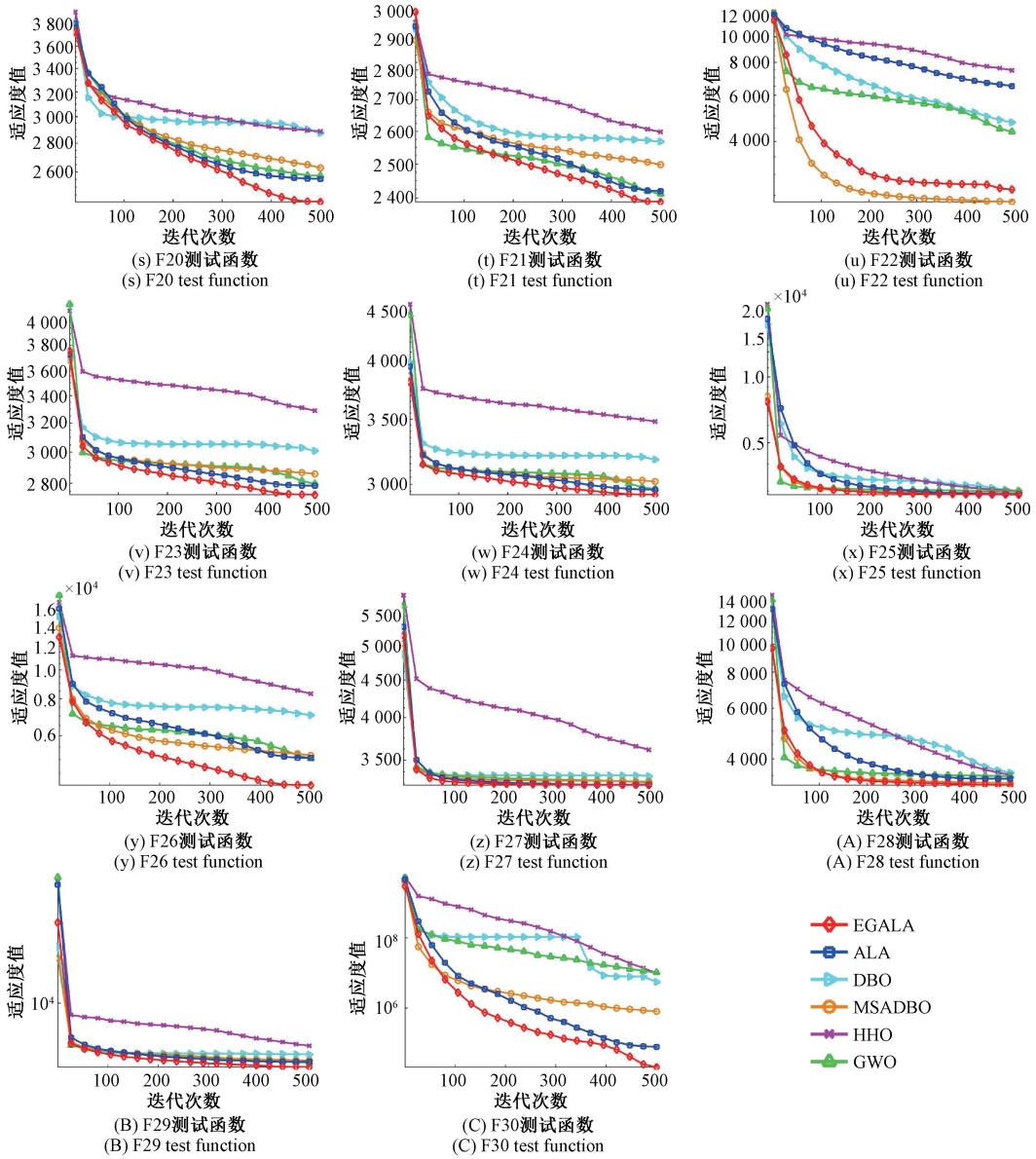


图 4　各个算法在 CEC2017 函数上的收敛曲线

Fig. 4　The convergence curves of each algorithm on the CEC2017 function

从图 4 可以看出，EGALA 算法在收敛速度和精度方面相较于其他对比算法具有明显优势，特别是在混合函数和复杂函数优化任务中，表现出卓越的性能。在单峰函数 F1 和 F3 中，EGALA 算法表现出快速的收敛趋势，并且最优值精度更高。在处理多峰函数 F6 和 F7 时，EGALA 算法相较于 DBO 等对比算法表现出明显优势。通过采用 Halton 序列初始化方法，EGALA 算法确保了种群在解空间中的均匀分布，所以在迭代初期能够快速识别最优解的可能区域，从而提高了初期解的多样性和收敛速度。在混合函数 F11～F20 和复合函数 F21～F30 的测试中，EGALA 算法同样展现了优异的表现，在 F12、F13、F14、F15、F18、F19 和 F26 中，EGALA 算法通过精英池策略动态引导多个精英个体的信息，增强了局部搜索的精度和全局搜索的能力，并且相比其他 5 种对比算法收敛速度更快。在函数 F11、F20、F24 和 F30 测试中，EGALA 算法相比 ALA 算法在迭代后期表现出更强的跳出局部最优的能力，通过引入非线性权重调节的黄金正弦策略，EGALA 算法在迭代后期能更快地向最优解收敛，并优化了全局和局部搜索的平衡，保持了较高的收敛精度。

为全面评估 EGALA 算法的优越性，本文采用 Wilcoxon 秩和检验对 EGALA 算法与其他对比算法在 F1 至 F30 测试函数上的测试结果进行统计学比较，并验证各算法整体结果的显著性差别，Wilcoxon 秩和检验结果如表 3 所示。假设 EGALA 算法与其他算法之间不存在显

著差异,若 p 值大于 0.05,则接受原假设;若 p 值小于 0.05,则拒绝原假设,并表明两种算法之间寻优性能差异较大。根据检验结果,EGALA 算法与其他对比算法在所有测试函数上的绝大部分 p 值均小于 5%,表明 EGALA 算法与对比算法之间存在显著的优化性能差异。综上所述,EGALA 算法在求解测试函数问题时,相较于其他算法表现出更优的收敛速度和寻优精度,且这些改进差异具有统计学上的意义,证明了改进的有效性和优越性。

表 3　各个算法在 CEC2017 函数上的 p 值

Table 3　The p-values of each algorithm on the CEC2017 function

函数	维度	DBO	HHO	GWO	MSADBO	ALA
F1	30	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F3	30	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	2.92×10^{-9}
F4	30	3.34×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	4.11×10^{-7}	3.67×10^{-3}
F5	30	1.33×10^{-10}	3.34×10^{-11}	4.86×10^{-3}	5.57×10^{-10}	$\mathbf{9.71\times10^{-1}}$
F6	30	3.02×10^{-11}	3.02×10^{-11}	6.07×10^{-11}	3.34×10^{-11}	1.46×10^{-10}
F7	30	4.50×10^{-11}	3.02×10^{-11}	5.19×10^{-7}	4.08×10^{-11}	3.16×10^{-5}
F8	30	7.38×10^{-10}	1.10×10^{-8}	3.27×10^{-2}	1.19×10^{-6}	$\mathbf{1.81\times10^{-1}}$
F9	30	1.73×10^{-7}	9.92×10^{-11}	2.12×10^{-1}	3.37×10^{-5}	$\mathbf{6.63\times10^{-1}}$
F10	30	7.60×10^{-7}	$\mathbf{3.71\times10^{-1}}$	2.77×10^{-5}	4.44×10^{-7}	$\mathbf{3.48\times10^{-1}}$
F11	30	4.08×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.34×10^{-11}	2.39×10^{-8}
F12	30	5.49×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.34×10^{-11}	4.86×10^{-3}
F13	30	6.70×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.75×10^{-5}
F14	30	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	2.61×10^{-10}
F15	30	1.31×10^{-8}	3.02×10^{-11}	3.02×10^{-11}	6.07×10^{-11}	7.77×10^{-9}
F16	30	5.97×10^{-9}	3.69×10^{-11}	$\mathbf{7.73\times10^{-2}}$	9.26×10^{-9}	$\mathbf{1.45\times10^{-1}}$
F17	30	3.82×10^{-10}	2.92×10^{-9}	1.03×10^{-2}	1.33×10^{-4}	1.60×10^{-3}
F18	30	3.02×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	6.72×10^{-10}
F19	30	5.07×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	6.12×10^{-10}
F20	30	3.52×10^{-7}	2.15×10^{-10}	2.71×10^{-2}	1.60×10^{-3}	1.52×10^{-3}
F21	30	3.69×10^{-11}	3.02×10^{-11}	1.38×10^{-2}	6.70×10^{-11}	3.56×10^{-4}
F22	30	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F23	30	3.02×10^{-11}	3.02×10^{-11}	5.53×10^{-8}	3.02×10^{-11}	8.20×10^{-7}
F24	30	3.02×10^{-11}	3.02×10^{-11}	4.84×10^{-2}	3.82×10^{-10}	1.52×10^{-3}
F25	30	8.99×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	9.76×10^{-10}	2.03×10^{-7}
F26	30	1.78×10^{-10}	3.34×10^{-11}	1.09×10^{-5}	7.20×10^{-5}	3.84×10^{-6}
F27	30	2.44×10^{-9}	4.08×10^{-11}	1.89×10^{-4}	6.36×10^{-5}	$\mathbf{4.55\times10^{-1}}$
F28	30	1.61×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.35×10^{-8}	2.20×10^{-8}
F29	30	1.33×10^{-10}	3.69×10^{-11}	8.29×10^{-6}	1.36×10^{-7}	1.43×10^{-5}
F30	30	1.09×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	5.61×10^{-5}

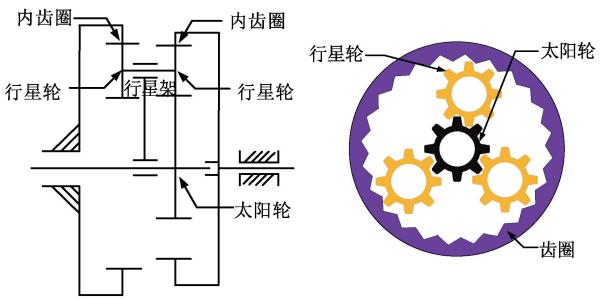
4　工程设计优化问题

为了验证本文提出的 EGALA 算法在工程实践中的优越性,本文针对 3 个复杂的工程优化问题进行了求解,包括行星轮系设计优化问题、多盘离合器制动器设计问题和工业制冷系统优化设计问题。在求解过程中,本文将 EGALA 算法的表现与多种对比算法进行了比较,包括 DBO 算法、HHO 算法、GWO 算法、MSADBO 算法和 ALA 算法。通过这些对比,可以有效评估 EGALA 算法在不同工程问题中的优化性能和优势。

4.1　行星轮系设计问题

行星轮系设计优化是一个多约束、多目标、多变量的复杂问题,需综合考虑多个因素以实现整体性能的最优化。设计目标包括提高传动效率、增强耐用性和可靠性、

减小体积和重量、以及降低噪音和振动等[15-16]。行星轮系设计的主要目标是最小化汽车传动系统中使用的传动比的最大误差，行星齿轮运动简图如图 5(a)所示，行星齿轮传动结构如图 5(b)所示。



(a) 运动简图
(a) Motion schematic diagram

(b) 齿轮传动结构
(b) Gear transmission structure

图 5　行星轮系设计问题示意图

Fig. 5　Schematic view of planetary gear train design problem

行星轮系设计问题是动力机械系统中的一个约束优化问题，该问题包含 3 个优化变量：轮齿数（$x_1 \sim x_6 = N_1 \sim N_6$）、齿轮模数（$x_7, x_8 = m_1, m_2$）和行星齿轮齿数（$x_9 = p$）以及 11 个不同几何约束和装配约束的约束[17]。其数学模型描述如下：

1)设计变量为：

$$x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = (N_1, N_2, N_3, N_4, N_5, N_6, m_1, m_2, p) \tag{24}$$

2)目标函数为：

$$\begin{cases} f(x) = \max \mid i_k - i_{0k} \mid, k = \{1, 2, \cdots, R\} \\ i_1 = \dfrac{N_6}{N_4}, i_{01} = 3.11, i_2 = \dfrac{N_6(N_1N_3 + N_3N_2)}{N_1N_3(N_6 + N_4)}, \\ i_{0R} = -3.11, I_R = -\dfrac{N_2N_6}{N_1N_3}, i_{02} = 1.84 \end{cases} \tag{25}$$

3)约束条件为：

$$\begin{cases} g_1(x) = m_2(N_6 + 2.5) - D_{\max} <= 0 \\ g_2(x) = m_1(N_1 + N_2) + m_1(N_2 + 2) - D_{\max} <= 0 \\ g_3(x) = m_2(N_4 + N_5) + m_2(N_5 + {}_2) - D_{\max} <= 0 \\ g_4(x) = \mid m_1(N_1 + N_2) - m_2(N_6 + N_3) \mid - m_1 - m_2 <= 0 \\ g_5(x) = -(N_1 + N_2)\sin(\pi/p) + N_2 + 2 + \delta_{22} <= 0 \\ g_6(x) = -(N_6 - N_3)\sin(\pi/p) + N_3 + 2 + \delta_{33} <= 0 \\ g_7(x) = -(N_4 + N_5)\sin(\pi/p) + N_5 + 2 + \delta_{55} <= 0 \\ g_8(x) = (N_3 + N_5 + 2 + \delta_{35})^2 - (N_6 - N_3)^2 - \\ (N_4 + N_5)^2 + 2(N_6 - N_3)(N_4 + N_5)\cos(\dfrac{2\pi}{p} - \beta) <= 0 \\ g_9(x) = N_4 - N_6 + 2N_5 + 2\delta_{56} + 4 <= 0 \\ g_{10}(x) = 2N_3 - N_6 + N_4 + 2\delta_{34} + 4 <= 0 \end{cases} \tag{26}$$

$$h_1(x) = \frac{N_6 - N_4}{p} = integer \tag{27}$$

其中，$\delta_{22} = \delta_{33} = \delta_{55} = \delta_{35} = \delta_{56} = 0.5$，$D_{\max} = 220$，

$$\beta = \frac{\cos^{-1}((N_4 + N_5)^2 + (N_6 - N_3)^2 - (N_3 + N_5)^2)}{2(N_6 - N_3)(N_4 + N_5)}$$

4)取值范围为：

$$\begin{cases} p = (3, 4, 5) \\ m_1,\ m_2 = (1.75, 2.0, 2.25, 2.5, 2.75, 3.0) \\ 17 \leqslant N_1 \leqslant 96 \\ 14 \leqslant N_2 \leqslant 54 \\ 14 \leqslant N_3 \leqslant 51 \\ 17 \leqslant N_4 \leqslant 46 \\ 14 \leqslant N_5 \leqslant 51 \\ 48 \leqslant N_6 \leqslant 124 \end{cases} \tag{28}$$

将 EGALA 算法应用于行星轮系设计问题，并与 DBO 算法、HHO 算法、GWO 算法、MSADBO 算法和 ALA 算法进行比较，实验参数设置与前述一致，均为独立运行 30 次迭代，测试结果如表 4 所示。分析测试结果可知，EGALA 算法在解决行星轮系设计问题时也具有优越的性能，并且在多个评估指标上均优于其他算法，表现出较好的寻优精度和稳定性。

表 4　行星轮系设计问题测试结果对比

Table 4　Comparison of test results for planetary gear train design problem

指标	DBO	HHO	GWO	MSADBO	ALA	EGALA
平均值	60 000.708 38	0.534 119 541	0.538 449 962	0.533 864 899	0.533 078 076	**0.528 594 435**
标准差	228 337.422 9	0.004 800 181	0.006 424 957	0.007 028 776	0.006 545 669	**0.002 939 874**
最优值	0.528 815 592	0.526 363 636	0.528 186 424	0.526 280 566	0.525 967 366	**0.525 768 707**

4.2　多盘离合器制动器设计问题

多盘离合器制动器设计问题是动力机械工程中的一个约束优化问题[17-18]，涉及 5 个优化变量及多个约束条件。其主要目标是确定 5 个决策变量的最优值，以实现质量最轻的离合器制动器，同时确保其性能和可靠性。优化变量包括内半径 r_i、外半径 r_o、盘厚度 t、动作力 F 和摩擦表面数 Z。多盘离合器制动器的结构如图 6 所示。

多盘离合器制动器设计数学模型如下：

1)设计变量为：

$$\vec{y} = [y_1, y_2, y_3, y_4, y_5] = [r_i, r_o, t, F, Z] \tag{29}$$

2)目标函数为：

$$f(\vec{y}) = \pi(y_2^2 - y_1^2)y_3(y_5 + 1)p \tag{30}$$

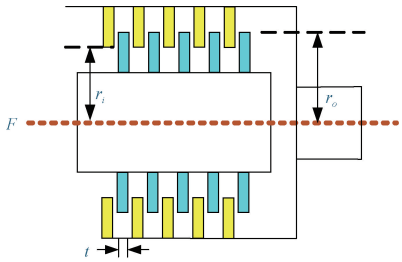


图 6　多盘离合器制动器设计示意图

Fig. 6　Schematic view of multi-disk clutch brake design problem

3)约束条件为:

$$\begin{cases} g_1(\vec{\boldsymbol{y}}) = p_{rz} - p_{\max} \leqslant 0 \\ g_2(\vec{\boldsymbol{y}}) = p_{rz}B_{sr} - B_{sr,\max}p_{\max} \leqslant 0 \\ g_3(\vec{\boldsymbol{y}}) = \Delta R + y_1 - y_2 \leqslant 0 \\ g_4(\vec{\boldsymbol{y}}) = -L_{\max} + (y_5+1)(y_3+\delta) \leqslant 0 \\ g_5(\vec{\boldsymbol{y}}) = sM_s - M_h \leqslant 0 \\ g_6(\vec{\boldsymbol{y}}) = T \geqslant 0 \\ g_7(\vec{\boldsymbol{y}}) = -V_{sr,\max} + V_{sr} \leqslant 0 \\ g_8(\vec{\boldsymbol{y}}) = T - T_{\max} \leqslant 0 \end{cases} \tag{31}$$

$$\begin{cases} M_h = \dfrac{2}{3}\mu y_4 y_5 \dfrac{y_2^3 - y_1^3}{y_2^2 - y_1^2}\ \text{N.mm} \\ \omega = \dfrac{\pi n}{30}\ \text{rad/s}, \\ A = \pi(y_2^2 - y_1^2)\text{mm}^2, \\ p_{rz} = \dfrac{y_4}{A}\ \text{N/mm}^2, B_{sr} = \dfrac{\pi R_{sr} \cdot n}{30}\ \text{mm/s}, \\ R_{sr} = \dfrac{2}{3}\dfrac{y_2^3 - y_1^3}{y_2^2 y_1^2}\ \text{mm}, T = \dfrac{L_z\omega}{M_h + M_f}, \\ \Delta R = 20\ \text{mm}, L_{\max} = 30\ \text{mm}, \\ \mu = 0.6, B_{sr,\max} = 10\ \text{m/s}, \\ \delta = 0.5\ \text{mm}, s = 1.5, \\ T_{\max} = 15\ \text{s}, n = 250\ \text{rpm}, \\ I_z = 55\ \text{kg} \cdot \text{m}^2, M_s = 40\ \text{Nm}, \\ M_f = 3\ \text{Nm}, p_{\max} = 1 \end{cases} \tag{32}$$

4)取值范围为:

$$\begin{cases} 60 \leqslant y_1 \leqslant 80 \\ 90 \leqslant y_2 \leqslant 110 \\ 1 \leqslant y_3 \leqslant 3 \\ 0 \leqslant y_4 \leqslant 1\ 000 \\ 2 \leqslant y_5 \leqslant 9 \end{cases} \tag{33}$$

将 EGALA 算法与 5 种对比算法的测试结果进行比较,测试结果如表 5 所示。从表中的数据可以明显看出,EGALA 算法和 ALA 算法在优化中均达到了最小质量数 0.235 242 458,表现优异,HHO 算法排名最后,最小质量数为 0.235 254 896。相比之下,EGALA 算法在整体性能和稳定性方面均优于 ALA 算法以及其他对比算法。

4.3　工业制冷系统的优化设计问题

随着常规能源的日益枯竭,能源高效利用与减排控制已成为当前各行业关注的重点问题。其中,工业制冷系统因其在整体能源消耗中占据较大比重,成为优化与调控的重点对象。该系统的优化问题涉及 14 个连续设计变量与 15 个非线性约束条件,属于高度复杂的多约束高维工程优化问题。这个问题可以表述为非线性不等式 COP[17],其数学模型如下:

1)设计变量为:

$$\vec{\boldsymbol{x}} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}] \tag{34}$$

2)目标函数为:

$$\begin{aligned} f(\vec{\boldsymbol{x}}) = {} & 63\ 098.88x_2x_4x_{12} + 5\ 441.5x_2^2x_{12} + \\ & 115\ 055.5x_2^{1.664}x_6 + 6\ 172.27x_2^2x_6 + 63\ 098.88x_3x_{11} + \\ & 5\ 441.5x_2^2x_{11} + 115\ 055.5x_2^{1.664}x_5 + 6\ 172.27x_2^2x_5 + \\ & 140.53x_1x_{11} + 281.29x_3x_{11} + 70.26x_2x_3 + 281.29x_3^2 + \\ & 14\ 437x_8x_{12}^{0.381\ 2}x_{14}^{0.342\ 4}x_{10}^{-1}x_7x_9^{-1} + 20\ 470x_7^{893}x_{11}^{0.316}x_1^2 \end{aligned} \tag{35}$$

3)约束条件为:

$$\begin{cases} g_1(\vec{\boldsymbol{x}}) = 1.524x_7^{-1} \leqslant 1 \\ g_2(\vec{\boldsymbol{x}}) = 1.524x_8^{-1} \leqslant 1 \\ g_3(\vec{\boldsymbol{x}}) = 0.077\ 89x_1 - 2x_7^{-3}x_9 - 1 \leqslant 0 \\ g_4(\vec{\boldsymbol{x}}) = 7.053\ 05x_9^{-1}x_7^{-1}x_8^{-1}x_{10}^{-1}x_8^{-1}x_2^{-1}x_{14}^{-1} \leqslant 0 \\ g_5(\vec{\boldsymbol{x}}) = 0.083\ 3x_1x_{14}^{-1} - 1 \leqslant 0 \\ g_6(\vec{\boldsymbol{x}}) = 47.136x_2^{333}x_1^{-1}x_{12} - 1.333x_8^{1.195}x_{13} + \\ \quad 62.08x_{12}^{2.119\ 5}x_{12}^{0.2}x_{10}^{-1}x_1^{-1} - 1 \leqslant 0 \\ g_7(\vec{\boldsymbol{x}}) = 0.047\ 71x_{10}^{18.812}x_{12}^{0.342\ 4} - 1 \leqslant 0 \\ g_8(\vec{\boldsymbol{x}}) = 0.048\ 8x_9^{893}x_{11}^{0.316} - 1 \leqslant 0 \\ g_9(\vec{\boldsymbol{x}}) = 0.009\ 9x_1x_3^{-1} - 1 \leqslant 0 \\ g_{10}(\vec{\boldsymbol{x}}) = 0.019\ 3x_2x_4^{-1} - 1 \leqslant 0 \\ g_{11}(\vec{\boldsymbol{x}}) = 0.029\ 8x_5^{-1} - 1 \leqslant 0 \\ g_{12}(\vec{\boldsymbol{x}}) = 0.056x_2^{-1} - 1 \leqslant 0 \\ g_{13}(\vec{\boldsymbol{x}}) = 2x_5^{-1} - 1 \leqslant 0 \\ g_{14}(\vec{\boldsymbol{x}}) = 2x_{10}^{-1} - 1 \leqslant 0 \\ g_{15}(\vec{\boldsymbol{x}}) = x_{12}x_{11}^{-1} - 1 \leqslant 0 \end{cases} \tag{36}$$

表 5　多盘离合器制动器设计问题测试结果对比

Table 5　Comparison of test results for multi-disk clutch brake design problem

指标	DBO	HHO	GWO	MSADBO	ALA	EGALA
平均值	0.238 428 033	0.267 098 717	0.235 305 451	0.235 247 276	0.244 799 183	**0.235 242 458**
标准差	0.017 448 113	0.045 820 622	$6.635\ 91\times10^{-5}$	$4.977\ 02\times10^{-6}$	0.029 160 299	**$8.469\ 02\times10^{-17}$**
最优值	0.235 242 722	0.235 254 896	0.235 244 936	0.235 242 861	**0.235 242 458**	**0.235 242 458**

4)取值范围为：

$$0.001 \leqslant x_i \leqslant 5, i = 1, \cdots, 14 \tag{37}$$

本文利用 EGALA 算法对工业制冷系统的 14 个关键设计变量进行优化，并求得最优解。为全面评估 EGALA 算法的性能，将其与 5 种优化算法进行对比并分析。表 6 列出了各算法得到的最优解及相关变量的具体数值。

表 6　各算法求解工业制冷系统问题的测试结果比较

Table 6　Comparison of test results for industrial refrigeration system problem solved by various algorithms

设计变量	DBO	HHO	GWO	MSADBO	ALA	EGALA
x_1	0.001 0	0.001 0	0.001 0	0.001 0	0.001 0	**0.001 0**
x_2	0.001 0	0.002 5	0.001 0	0.001 0	0.001 0	**0.001 0**
x_3	0.001 0	0.002 6	0.001 0	0.001 0	0.001 0	**0.001 0**
x_4	0.001 0	0.008 8	0.001 9	0.001 0	0.001 0	**0.001 0**
x_5	0.001 0	0.001 0	0.001 0	0.002 6	0.001 2	**0.001 0**
x_6	0.001 0	0.085 2	0.001 4	0.001 1	0.001 0	**0.001 0**
x_7	1.542 8	2.081 2	1.524 3	1.543 3	1.524 0	**1.524 0**
x_8	1.579 3	1.524 0	1.535 2	1.560 0	1.524 0	**1.524 0**
x_9	5.000 0	4.217 0	4.977 2	4.824 7	4.999 3	**5.000 0**
x_{10}	3.934 8	4.217 0	2.000 9	3.708 1	3.130 6	**2.000 0**
x_{11}	0.001 0	0.323 8	0.010 3	0.026 7	0.001 0	**0.001 0**
x_{12}	0.001 0	0.243 6	0.009 9	0.006 2	0.001 0	**0.001 0**
x_{13}	0.001 0	0.106 9	0.021 0	0.022 0	0.009 0	**0.007 3**
x_{14}	0.088 1	1.283 7	0.249 1	0.262 3	0.108 6	**0.087 6**
最优值	0.055 768 151 58	1.323 193 661 43	0.042 743 962 28	0.067 968 195 21	0.038 015 099 20	**0.032 212 995 73**

从表 6 的数据可以看出，针对具有 14 个设计变量的工程问题，EGALA 算法表现出良好的优化性能。各参数值均得到了不同程度的改善，最终得到的最优解为 0.032 212 995 73，求解结果优于其他 5 种对比算法。这表明，EGALA 算法在求解包含多个约束条件的工程优化问题时，不仅具有更高的求解精度和稳定性，而且在整体性能上得到了较好提升。

5　结　　论

本文提出的多策略改进的人工旅鼠优化算法，通过引入 Halton 序列实现初始种群均匀分布，结合精英池策略与惯性权重提升跳跃搜索能力，并在迭代后期采用非线性权重的黄金正弦策略与觅食行为相结合，增强了局部搜索的精度与稳定性，在全局探索与局部开发之间取得了更优的平衡。本文使用 CEC2017 测试函数集中的 29 个基准函数上进行了实验，并采用 Wilcoxon 秩和检验与 Friedman 检验对结果进行统计分析，实验结果表明，改进算法表现出更快的收敛速度和更高的寻优精度，整体性能优于其他 5 种对比算法。此外，本文还将改进算法成功应用于行星轮系设计、多盘离合器制动器设计及工业制冷系统优化 3 个不同的工程设计优化问题，实验结果显示，改进算法在解决复杂工程问题时表现出良好的稳定性和适应性，能够有效提高优化效率和解的质量，进一步验证了其工程应用潜力和实用价值。未来工作将聚焦于进一步提升算法在高维、多约束问题中的求解能力，并探索其在故障诊断、生产调度、图像处理及智能控制等实际工程场景中的应用，以增强算法的通用性与跨领域适配能力。

参考文献

[1] 朱国庆，韩东颖，米振涛，等. 多策略改进的蜣螂优化算法及其工程实例应用[J]. 电子测量技术，2024，47(21)：82-99.
ZHU G Q，HAN D Y，MI ZH T，et al. Multi-strategy improved dung beetle optimization algorithm and its application in engineering examples [J]. Electronic Measurement Technology，2024，47(21)：82-99.

[2] HASHIM F A，HUSSIEN A G. Snake optimizer：A novel meta-heuristic optimization algorithm [J]. Knowledge-Based Systems，2022，242：108320.

[3] ALHIJAWI B，AWAJAN A. Genetic algorithms：Theory，genetic operators，solutions，and applications [J]. Evolutionary Intelligence，2024，17(3)：1245-1256.

[4] TIJJANI S，AB WAHAB M N，NOOR M H M. An enhanced particle swarm optimization with position update for optimal feature selection [J]. Expert Systems with Applications，2024，247：123337.

[5] HEIDARI A A，MIRJALILI S，FARIS H，et al. Harris hawks optimization：Algorithm and

applications [J]. Future Generation Computer Systems, 2019, 97: 849-872.

[6] MAKHADMEH S N, AL-BETAR M A, DOUSH I A, et al. Recent advances in grey wolf optimizer, its versions and applications[J]. IEEE Access, 2023, 12: 22991-3028.

[7] 李鹏, 丁倩雯. 基于麻雀算法优化的 OSTU 分割算法[J]. 电子测量技术, 2021, 44(19): 148-154.
LI P, DING Q W. OSTU segmentation algorithm based on sparrow algorithm optimization [J]. Electronic Measurement Technology, 2021, 44(19): 148-154.

[8] 鲍新宇, 高红亮, 周瑞,等. 基于灰狼优化算法的电力系统低频振荡抑制策略设计[J]. 电器工业, 2025(4): 35-39.
BAO X Y, GAO H L, ZHOU R, et al. Design of power system low-frequency oscillation damping strategy based on grey wolf optimization algorithm[J]. China Electrical Equipment Industry, 2025(4): 35-39.

[9] 程换新, 黄震. 基于改进 PSO 优化 RNN 的短期电力负荷预测模型[J]. 电子测量技术, 2019, 42(20): 94-98.
CHENG H X, HUANG ZH. Short-term electric load forecasting model based on improved PSO optimized RNN[J]. Electronic Measurement Technology, 2019, 42(20): 94-98.

[10] XIAO Y, CUI H, KHURMA R A, et al. Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems [J]. Artificial Intelligence Review, 2025, 58(3): 84.

[11] ABDEL-SALAM M, ABUALIGAH L, ALZAHRANI A I, et al. Boosting crayfish algorithm based on halton adaptive quadratic interpolation and piecewise neighborhood for complex optimization problems [J]. Computer Methods in Applied Mechanics and Engineering, 2024, 432: 117429.

[12] 匡鑫, 阳波, 马华, 等. 多策略改进的蜣螂优化算法[J]. 计算机工程, 2024, 50(10): 119-136.
KUANG X, YANG B, MA H, et al. Multi-strategy improved dung beetle optimization algorithm [J]. Computer Engineering, 2024, 50(10): 119-136.

[13] XUE J, SHEN B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, 2023, 79(7): 7305-7336.

[14] 潘劲成, 李少波, 周鹏, 等. 改进正弦算法引导的蜣螂优化算法[J]. 计算机工程与应用, 2023, 59(22): 92-110.
PAN J CH, LI SH B, ZHOU P, et al. Dung beetle optimization algorithm guided by improved sine algorithm[J]. Computer Engineering and Applications, 2023, 59(22): 92-110.

[15] ZHANG J, XIAO M, GAO L, et al. Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems [J]. Applied Mathematical Modelling, 2018, 63: 464-490.

[16] WANG K G, GUO M, DAI C, et al. Information-decision searching algorithm: Theory and applications for solving engineering optimization problems [J]. Information Sciences, 2022, 607: 1465-1531.

[17] KUMAR A, WU G, ALI M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results[J]. Swarm and Evolutionary Computation, 2020, 56: 100693.

[18] ZHAO W G, ZHANG ZH X, WANG L Y. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications[J]. Engineering Applications of Artificial Intelligence, 2020, 87: 103300.

作者简介

杨原,硕士研究生,主要研究方向为智能优化算法。
E-mail:3467043945@qq.com

陈明霞(通信作者),本科,教授级高工,主要研究方向为装备自动化技术与过程控制系统。
E-mail:chenmxfly@qq.com