计算机科学
COMPUTER SCIENCE

基于多策略改进的电鳗觅食优化算法

王鑫玮, 冯锋

引用本文

王鑫玮, 冯锋. 基于多策略改进的电鳗觅食优化算法[J]. 计算机科学, 2025, 52(11): 245-254.

WANG Xinwei, FENG Feng. Multi-strategy Improved Electric Eel Foraging Optimization Algorithm[J]. Computer Science, 2025, 52(11): 245-254.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241100106

基于多策略改进的电鳗觅食优化算法

王鑫玮　冯　锋
宁夏大学信息工程学院　银川 750021
(15709589970@163.com)

摘　要　电鳗觅食优化算法 EEFO(Electric Eel Foraging Optimization)在迭代过程中会出现全局探索能力不足、容易陷入局部最优和收敛速度慢的问题。同时,算法的性能受到参数设置的影响,需要仔细调整和优化。对此,提出了一种多策略改进的电鳗觅食优化算法(IEEFO)。首先,调整能量因子策略,引入了双曲正切能量因子,使算法在迭代过程中提前加入开发行为,从而快速发现最优种群,加快收敛速度;之后,改进扰动因子,扩大电鳗游走的位置范围,有利于种群的全局寻优;然后,在迁徙阶段加入正弦余弦策略,促进算法的局部开发;最后,在每次迭代之后,加入透镜成像反向学习的策略来扩大搜索空间,使得算法跳出局部最优并加速收敛到全局最优解。将 IEEFO 分别与 6 种基本算法、4 种单策略改进的电鳗觅食优化算法进行对比,对 13 个基准函数进行仿真实验,对 IEEFO 算法进行性能评估。实验结果表明,IEEFO 相比于对比算法收敛速度更快,全局寻优能力更强,算法总体性能有显著提升。此外,通过一个机械优化设计实验进行测试分析,进一步验证了 IEEFO 的有效性和适用性。

关键词:电鳗觅食优化算法;透镜成像反向学习;能量因子;扰动因子;正弦余弦算法;群智能优化算法
中图分类号　TP301.6

Multi-strategy Improved Electric Eel Foraging Optimization Algorithm

WANG Xinwei and FENG Feng
College of Information Engineering,Ningxia University,Yinchuan 750021,China

Abstract　In response to the issues of EEFO algorithm,such as insufficient global exploration ability,susceptibility to local optima,slow convergence,and performance sensitivity to parameter settings that require careful adjustment and optimization,a multi-strategy improved Electric Eel Foraging Optimization algorithm(IEEFO)is proposed. Firstly,the energy factor strategy is adjusted by introducing a hyperbolic tangent energy factor,which allows the algorithm to incorporate exploratory behavior earlier in the iteration process,enabling rapid discovery of the optimal population and accelerating convergence speed. Secondly,the disturbance factor is improved to increase the range of positions where the electric eel can move,which is beneficial for global optimization of the population. Then,a sine cosine strategy is added during the migration phase,which is conducive to local exploration of the algorithm. Finally,after each iteration,a lens imaging reverse learning strategy is incorporated to expand the search space,which helps the algorithm escape from local optima and accelerate convergence to the global optimal solution. The IEEFO is compared with 6 basic algorithms and 4 single-strategy improved Electric Eel Foraging Optimization algorithms,and 13 benchmark functions are used for simulation experiments to evaluate the performance of the IEEFO algorithm. The experimental results show that the IEEFO has faster convergence speed and stronger global optimization ability compared to the aforementioned algorithms,with a significant improvement in overall algorithm performance. Additionally,a mechanical optimization design experiment is conducted to further test and analyze the effectiveness and applicability of the IEEFO.

Keywords　Electric eel optimization algorithm,Lens imaging ieverse learning,Energy factor,Perturbation factor,Sine cosine algorithm,Swarm intelligence optimization algorithm

1　引言

在现实世界中,优化算法的应用非常广泛,涉及工程设计的决策[1-3]、资源的分配、供应链的管理、交通网络的设计等。传统的优化算法在解决特定类型的优化问题时非常有效,例如线性规划、动态规划等。然而,在解决许多非线性、高维、多峰的优化问题中,传统的优化算法将不再适用,更先进的优化技术陆续被提出,如遗传算法、粒子群优化[4]、模拟退火、灰狼优化算法[5]等元启发式算法。元启发式算法是模仿自然界中的一些现象,如生物的进化、群体的行为、物理过程等。元启发式方法作为数学方法的理想替代方法,具有随机性强、易于实现、黑箱考虑等优点,这弥补了数学方法的不足。近年来,

到稿日期:2024-11-18　返修日期:2025-02-12
基金项目:宁夏自然科学基金重点项目(2024AAC02011)
This work was supported by the Key Project of Ningxia Natural Science Foundation(2024AAC02011).
通信作者:冯锋(feng_f@nxu.edu.cn)

国内外学者提出了很多元启发式算法，例如 Zheng 等[6]提出了苔藓生长优化算法(Moss Growth Optimization，MGO)、Abdel-Basset 等[7]提出了冠豪猪优化算法(Crested Porcupine Optimizer，CPO)、Abdel-Basset 等[8]提出了光谱优化算法(Light Spectrum Optimizer，LSO)、Trojovská 等[9]提出了斑马优化算法(Zebra Optimization Algorithm，ZOA)、Zolf[10]提出了淘金优化算法(Gold Rush Optimizer，GRO)，这些算法在处理优化问题中都取得了不错的效果。

电鳗觅食优化算法(Electric Eel Foraging Optimization，EEFO)是 Zhao 等[11]提出的一种元启发算法，EEFO 从自然界中电鳗表现出的智能群体觅食行为中汲取灵感，通过对电鳗 4 种关键的觅食行为(相互作用、休息、狩猎和迁徙)进行数学建模，以在优化过程中进行探索和利用。与其他元启发式算法相比，该算法具有收敛速度快、寻优能力强等特点，但是在处理复杂问题时，也存在容易陷入局部最优、全局探索能力不足、收敛速度慢等问题。对此，本文提出了一种基于多策略改进的电鳗觅食优化算法(Improved Electric Eel Foraging Optimization，IEEFO)。算法首先调整能量因子，然后改进扰动因子[12]，之后在算法迁徙阶段引入正弦余弦策略[13]，最后在每次迭代之后，加入透镜成像反向学习策略[14]。通过将 IEEFO 对 13 个基准函数进行仿真实验，并分别与 6 种基本算法、4 种单策略改进的电鳗优化算法进行对比，验证了该算法的可行性和准确性。

2 电鳗觅食优化算法

EEFO 是一种模拟电鳗在自然界中群体觅食行为的算法，包括初始化、相互行动、休息、狩猎和迁徙 5 个阶段。鳗鱼是群居动物，常采用社会捕食的方式狩猎，当鳗鱼遇到鱼群时，它们首先会相互游动和搅动，将鱼群赶到一个包围圈中，然后鳗鱼群以巨大的通电圈的方式围捕猎物。

2.1 初始化

将 n 条电鳗分别放在各个食物源的位置上，将其位置初始化如下：

$$\boldsymbol{X}_i=r\times(\boldsymbol{ub}_i-\boldsymbol{lb}_i)+\boldsymbol{lb}_i,i=1,2,\cdots,n \tag{1}$$

其中，$\boldsymbol{X}_i$ 表示第 i 条电鳗的初始位置，$\boldsymbol{ub}_i$ 和 $\boldsymbol{lb}_i$ 分别表示搜索空间的上界和下届，r 为[0,1]的随机数，n 为种群数量。

2.2 互动行为

电鳗通过种群中的位置信息进行交互，通过从种群中随机选择的电鳗与搜索空间中随机生成的电鳗之间的差异来更新位置，具体过程如下。

当 $fit(x_j(t))<fit(x_i(t))$ 时：

$$v_i(t+1)=\begin{cases}x_j(t)+C\times(\bar{x}-x_i(t)), & p_1>0.5\\ x_j(t)+C\times(x_r(t)-x_i(t)), & p_1\leqslant 0.5\end{cases} \tag{2}$$

当 $fit(x_j(t))\geqslant fit(x_i(t))$ 时：

$$v_i(t+1)=\begin{cases}x_i(t)+C\times(\bar{x}(t)-x_j(t)), & p_2>0.5\\ x_i(t)+C\times(x_r(t)-x_j(t)), & p_2\leqslant 0.5\end{cases} \tag{3}$$

$$\bar{x}(t)=\frac{1}{n}\sum_{i=1}^{n}x_i(t) \tag{4}$$

$$x_r=Low+\boldsymbol{r}\times(Up-Low) \tag{5}$$

其中，$fit(x_i(t))$ 是第 i 条电鳗侯选位置的第 t 代适应度值，$\boldsymbol{x}_j$ 是种群中随机选择的一条电鳗的位置，n 是种群的数量，t 是当前种群代数，p_1 和 p_2 是(0,1)的随机数，$\boldsymbol{r}$ 是(0,1)的随机向量。

2.3 休息行为

将电鳗位置的矢量任意一维投影到搜索空间的主对角线区域建立休息区，将搜索空间和电鳗的位置归一化，过程如下：

$$z(t)=\frac{x_{rand(n_t)}^{rand(d_t)}-Low^{rand(d)}}{Up^{rand(d)}-Low^{rand(d)}} \tag{6}$$

$$\boldsymbol{Z}(t)=Low+z(t)\times(Up-Low) \tag{7}$$

其中，$x_{rand(n)}^{rand(d)}$ 是随机选择的电鳗的随机维度的值，$z(t)$ 表示将电鳗位置归一化，$\boldsymbol{Z}(t)$ 表示电鳗在休息区的初始位置。

休息区建立后，电鳗在休息区休息的行为过程表示如下：

$$v_i(t+1)=R_i(t+1)+n\times(R_i(t+1)-round(r_1)\times x_i(t)) \tag{8}$$

$$R_i(t+1)=\boldsymbol{Z}(t)+\alpha\times|\boldsymbol{Z}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{9}$$

$$\alpha=2\cdot(\mathrm{e}-\mathrm{e}^{\frac{t}{T}})\times\sin(2\pi r_2) \tag{10}$$

其中，n，r_1 和 r_2 是(0,1)的随机数，$round(r_1)$ 表示四舍五入 r_1，α 是静止区域的比例，$\boldsymbol{x}_{\text{prey}}$ 是当前最优解的位置向量，$\boldsymbol{R}_i$ 是电鳗的休息位置。

2.4 迁徙行为

当电鳗发现猎物时，电鳗会从休息区迁徙到狩猎区，其过程如下：

$$v_i(t+1)=-r\times R_i(t+1)+r\times H_r(t+1)-L\times(H_r(t+1)-x_i(t)) \tag{11}$$

$$H_r(t+1)=\boldsymbol{x}_{\text{prey}}(t)+\beta\times|\bar{x}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{12}$$

$$\beta=2\cdot(\mathrm{e}-\mathrm{e}^{\frac{t}{T}})\times\sin(2\pi r) \tag{13}$$

$$L=0.01\times\left|\frac{u\cdot\sigma}{|v|^{\frac{1}{b}}}\right| \tag{14}$$

$$u,v\sim\mathrm{N}(0,1) \tag{15}$$

$$\sigma=\left(\frac{\Gamma(1+b)\times\sin\left(\frac{\pi b}{2}\right)}{\Gamma\left(\frac{1+b}{2}\right)\times b\times 2^{\frac{b-1}{2}}}\right)^{\frac{1}{b}} \tag{16}$$

其中，$\boldsymbol{H}_r$ 为狩猎区的任何位置，β 是狩猎区域的比例，$H_r(t+1)-x_i(t)$ 表示电鳗向狩猎区移动，L 是莱维飞行函数，Γ 是标准伽马函数，$b=1.5$。

2.5 狩猎行为

当电鳗发现猎物时，它们会围成一个大圈将猎物包围起来并不断缩小电圈范围，此时猎物会在狩猎区四处逃窜，电鳗的位置根据猎物的位置变化而更新。电鳗的狩猎行为如下：

$$H_{\text{prey}}(t+1)=\boldsymbol{x}_{\text{prey}}(t)+\beta\times|\bar{x}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{17}$$

$$v_i(t+1)=H_{\text{prey}}(t+1)+\eta\times(H_{\text{prey}}(t+1)-round(rand)\times x_i(t)) \tag{18}$$

$$\eta=\mathrm{e}^{\frac{r\times(1-t)}{T}}\times\cos(2\pi r) \tag{19}$$

其中，η 为卷曲因子，r 是(0,1)的随机数。

2.6 EEFO 实现流程

1)设置种群规模、迭代次数、搜索空间上下限、维度等参数。

2)通过初始化策略初始化种群位置，同时计算相应的

适应度值。

3）能量因子 $E>1$，电鳗进行交互行为，更新电鳗位置。

4）能量因子 $E<\frac{1}{3}$，电鳗进行休息行为，更新电鳗位置，更新最优解。

5）能量因子 $\frac{2}{3}<E\leqslant 1$，电鳗进行迁徙行为，更新电鳗位置，更新最优解。

6）能量因子 $\frac{1}{3}\leqslant E\leqslant\frac{2}{3}$，电鳗进行狩猎行为，跟新电鳗位置，更新最优解。

7）当算法符合终止条件时，结束算法。

3 改进的电鳗觅食优化算法

3.1 双曲正切能量因子

标准 EEFO 算法中的能量因子 E 用于决定电鳗做出探索还是开发行为，如图 1 能量因子曲线图所示。该因子前期和中期的值多数大于 1，表示电鳗多以互动行为为主；后期的值多数小于 1，表示电鳗以休息、迁徙、狩猎行为为主。这种方式缺乏种群多样性，算法容易陷入局部最优，其定义为：

$$E=4\times\sin\left(1-\frac{t}{T}\right)\times\ln\frac{1}{r} \tag{20}$$

其中，r 为(0，1)的随机数。

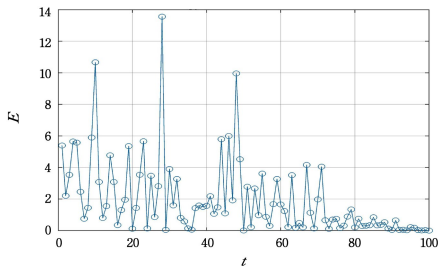


图 1 能量因子曲线

Fig. 1 Curve of energy factor

改进后的能量因子 E' 加入了双曲正切函数，如图 2 所示。双曲正切能量因子在保证算法原有结构的情况下，在算法前中期加入休息、迁徙、狩猎行为。通过引入开发行为，算法能够在前期探索更广泛的解空间，增加种群的多样性，同时有利于算法在前中期找到最优解，加快收敛速度。改进后的能量因子 E' 在算法中后期加入探索行为，有助于算法跳出局部最优，可以增强算法对全局最优的搜索能力。其定义为：

$$E'=\left|\ln\left(10\times\tanh\left(1-\frac{t}{T}\right)\right)^{r}\right| \tag{21}$$

其中，r 为(0，1)的随机数。

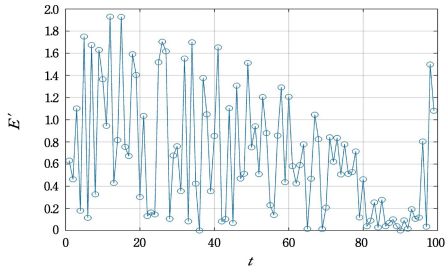


图 2 双曲正切能量因子曲线

Fig. 2 Curve of hyperbolic tangent energy factor

3.2 扰动因子

标准的 EEFO 算法中扰动因子为式(10)中的 α，用于算法的开发部分。其通过扰动因子来改变电鳗休息区、狩猎区和迁徙的范围大小，如图 3 所示。

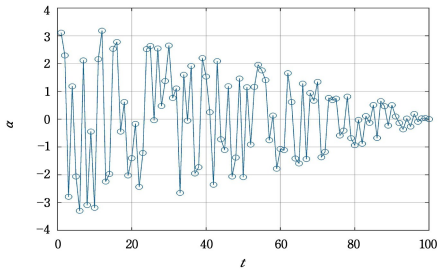


图 3 扰动因子 α 曲线

Fig. 3 Curve of perturbation factor α

从图 3 中可以看到，扰动因子 α 随着迭代次数增加，扰动能力下降，容易导致算法陷入局部最优。算法在前期加入少许开发阶段后，扰动因子 q 将在前中期减少电鳗开发行为的区域比例，以此平衡探索和开发行为；进入中后期，电鳗主要以休息、迁徙、狩猎行为为主，此时引入扰动因子 q，增加扰动开发行为的区域比例，防止算法陷入局部最优，有利于找到最优解。其计算式如下：

$$q=u\times v \tag{22}$$

$$u=a\times\left(\sin(r)\times\left(1+\frac{t}{T}\right)\right)^{3}+1 \tag{23}$$

$$v=b\times\sin(2\pi\times r) \tag{24}$$

经过大量实验得出，当 $a=20$，$b=0.01$ 时，扰动因子 q 效果最好，r 为(0，1)的随机数。

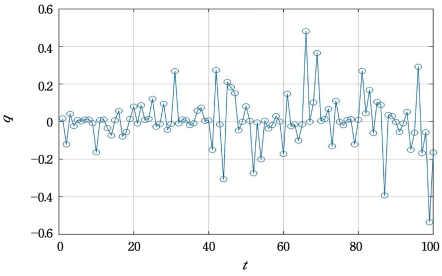


图 4 扰动因子 q 曲线

Fig. 4 Curve of disturbance factor q

改进后的电鳗休息区公式如下：

$$R_i(t+1)=\boldsymbol{Z}(t)+q\times|\boldsymbol{Z}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{25}$$

改进后的电鳗迁徙区公式如下：

$$H_r(t+1)=\boldsymbol{x}_{\text{prey}}(t)+q\times|\bar{x}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{26}$$

改进后的电鳗狩猎区公式如下：

$$H_{\text{prey}}(t+1)=\boldsymbol{x}_{\text{prey}}(t)+q\times|\bar{x}(t)-\boldsymbol{x}_{\text{prey}}(t)| \tag{27}$$

3.3 正弦余弦策略

正弦余弦策略[15]是一种基于正弦和余弦函数的优化算法的机制，其通过模拟正弦和余弦函数的波动特性来更新解的位置。标准 EEFO 算法在迁徙阶段采用莱维飞行更新种群的位置，莱维飞行具有长尾的特性，这种特性可能导致算法过早地集中在某些区域，而忽视其他可能的解区域。本文方法将正弦余弦策略引入 EEFO 算法的迁徙阶段来代替莱维

飞行，其利用正弦和余弦函数的波动特性来更新种群位置，可以避免算法过早收敛，在全局范围内搜索最优解，同时能在局部范围进行细致的搜索，在保持全局搜索能力的同时，增加了局部开发能力，可以帮助算法跳出局部最优，提高搜索效率，且在处理复杂问题时有更稳定的性能。将正弦余弦策略引入迁徙阶段的公式如下：

$$v_i'(t+1)=w_i(t+1)+y_i(t+1) \tag{28}$$

$$w_i(t+1)=r_0\times R_i(t+1)+r\times H_r(t+1) \tag{29}$$

$$y_i(t+1)=\begin{cases} r_1\times\sin r_2\times|r_3\times H_r(t+1)-x_i(t)|, & r_4<0.5 \\ r_1\times\cos r_2\times|r_3\times H_r(t+1)-x_i(t)|, & r_4\geqslant 0.5 \end{cases} \tag{30}$$

$$r_1=2-\frac{2t}{T} \tag{31}$$

其中，r，r_0，r_4 为(0,1)的随机数，r_2 为(0,2π)的随机数，r_3 为(0,2)的随机数。

3.4 透镜成像反向学习

标准 EEFO 算法的电鳗种群变异方式单一，导致种群多样性不足，进而导致算法全局搜索能力受限，在处理复杂的、多峰的或高维的优化问题时，容易陷入局部最优解。透镜成像反向学习[16]是一种扩展空间的技术，其通过模拟透镜成像原理生成动态变化的反向解来增加种群的多样性，优化较差的电鳗位置，这有助于增强算法的全局搜索能力，提高收敛精度。改进后的算法在每次迭代过程结束后都进行一次透镜成像反向学习，其公式如下：

$$x_i^*=\frac{ub_i+lb_i}{2}+\frac{ub_i+lb}{2n}-\frac{x_i}{n} \tag{32}$$

其中，x_i^* 为 x_i 的反向解空间，ub 和 lb 为上界和下界，n 为调节因子。

$$x_i=\begin{cases} x_i, & f_i<f_i^* \\ x_i^*, & f_i\geqslant f_i^* \end{cases} \tag{33}$$

其中，f_i 和 f_i^* 分别表示第 i 条电鳗在正向和反向解空间的适应度值。比较两个适应度值，选取较优的个体作为下一次迭代的个体初始位置，有利于算法更快地收敛到最优解。

$$n=\left(1+\sqrt{\left(\frac{t}{T}\right)}\right)^{10} \tag{34}$$

自适应调节因子 n 随着迭代次数的增加而变大，使得反向空间的种群变化越来越小，这有助于算法在前期进行大范围的搜索，更快找到最优解的大致位置。随着调节因子 n 的增加，算法的搜索范围缩小，加强了局部搜索能力，能够更精确地搜寻到最优解。图 5 为自适应调节因子 n 随迭代次数变化的图像。

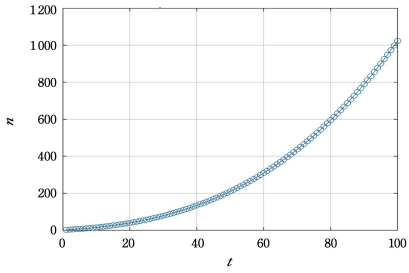


图 5 自适应调节因子曲线

Fig. 5 Curve of adaptive adjustment factor

3.5 IEEFO 的实现流程

基于多策略改进的电鳗觅食优化算法的流程图如图 6 所示，具体流程如下。

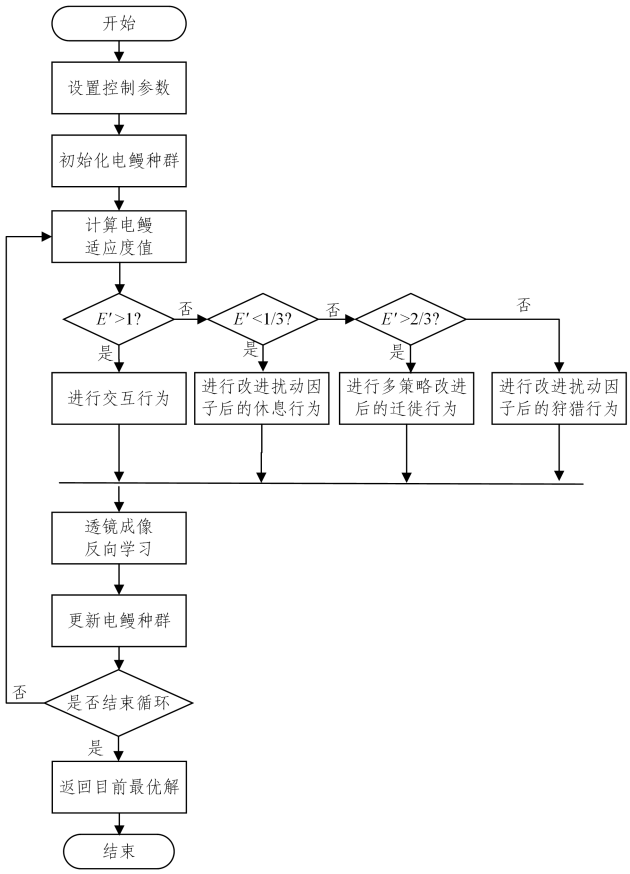


图 6 改进算法的流程图

Fig. 6 Flowchart of the improved algorithm

1)初始化算法控制参数，包括电鳗种群数量、最大迭代次数、维度、搜索空间上下限等参数。

2)随机产生均匀分布的电鳗种群。

3)每次迭代过程中，先判断双曲正切能量因子 E' 大小。当 $E'>1$ 时，电鳗执行交互行为；当 $E'<\frac{1}{3}$ 时，电鳗执行经过扰动因子 q 改进后的休息行为；当 $\frac{2}{3}<E'\leqslant 1$ 时，电鳗执行经过扰动因子 q 和正弦余弦策略改进后的迁徙行为；当 $\frac{1}{3}\leqslant E'\leqslant\frac{2}{3}$ 时，电鳗执行经过扰动因子 q 改进后的狩猎行为。

4)每次迭代后，电鳗经过透镜成像反向学习生成反向解，选取自身和反向解中的最优解作为下一代的初始值。

5)满足终止条件，则结束；不满足，则跳转至步骤 3)。

IEEFO 算法的伪代码如算法 1 所示。

算法 1 IEEFO 算法

输入：电鳗的种群大小 i，最大迭代次数 T

输出：最优电鳗位置 x_{prey} 及其适应度值 fit

1. 随机初始化种群 x_i

2. 计算电鳗适应度值

3. While t<T do

4. 通过式(21)计算能量因子 E′

5. For each x_i do
6. If $E'>1$
7. 电鳗使用式(2)、式(3)进行交互行为
8. Else If $E'<\frac{1}{3}$
9. 电鳗使用式(8)、式(25)进行改进后的休息行为
10. Else If $E'>\frac{2}{3}$
11. 电鳗使用式(25)、式(26)、式(28)进行改进后的迁徙行为
12. Else
13. 电鳗使用式(18)、式(27)进行改进后的狩猎行为
14. End if
15. 更新电鳗位置，并计算其适应度值
16. 通过透镜反向学习(见式(32))，筛选出优质的电鳗(见式(33))
17. 更新电鳗位置 x_i，并返回相应的适应度值 fit
18. End for
19. End while
20. Return x_{prey}

3.6 IEEFO 算法时间复杂度分析

算法的时间复杂度是衡量算法性能的重要因素之一，它描述了算法执行时间随输入规模增长的变化趋势。在 IEEFO 算法中，影响算法时间复杂度的参数主要包括电鳗的数量 n、所考虑问题的维度 d 和最大迭代次数 T。IEEFO 算法的时间复杂度由问题定义、初始化、函数求值、电鳗交互位置更新、电鳗休息位置更新、电鳗迁徙位置更新、电鳗狩猎位置更新、透镜反向学习位置更新。其时间复杂度为：

$$\begin{aligned}O(IEEFO)&=O(1)+O(n)+O(Tn)+O\left(\frac{1}{2}Tnd\right)+\\&\quad O\left(\frac{1}{2}Tnd\right)+O\left(\frac{1}{6}Tnd\right)+O\left(\frac{1}{6}Tnd\right)+\\&\quad O\left(\frac{1}{6}Tnd\right)+O(Tnd)\\&=O(2Tnd+Tn+n+1)\\&\cong O(Tnd)\end{aligned}$$

标准的 EEFO 算法的时间复杂度为：

$$\begin{aligned}O(EEFO)&=O(1)+O(n)+O(Tn)+O\left(\frac{1}{2}Tnd\right)+\\&\quad O\left(\frac{1}{6}Tnd\right)+O\left(\frac{1}{6}Tnd\right)+O\left(\frac{1}{6}Tnd\right)\\&=O(Tnd+Tn+n+1)\\&\cong O(Tnd)\end{aligned}$$

综上可知，IEEFO 算法和 EEFO 算法的时间复杂度相同，说明改进方法在提升算法性能的同时并没有增加算法的时间开销。

4 实验仿真与结果分析

4.1 测试环境

本文实验在一台 Intel Core i5-7500 CPU、3.4 GHz、8 GB 内存、Windows 10 64 位电脑上进行，采用 MATLAB R2022a 软件编写。通过 13 个经典测试函数对 IEEFO 算法进行验证，测试函数如表 1 所列。

表 1 测试函数

Table 1 Test functions

函数名	定义	范围	最优值
f_1	$f_1(x)=\sum_{i=1}^{n}(x_i^2)$	$[-100,100]$	0
f_2	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]$	0
f_3	$f_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	$[-100,100]$	0
f_4	$f_4(x)=\max_i\{\lvert x_i\rvert,1\leqslant i\leqslant n\}$	$[-100,100]$	0
f_5	$f_5(x)=\sum_{i=1}^{n-1}[100(x_i^2-x_{i+1})^2+(1-x_i)^2]$	$[-30,30]$	0
f_6	$f_6(x)=\sum_{i=1}^{n}(\lvert x_i+0.5\rvert)^2$	$[-100,100]$	0
f_7	$f_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	$[-1.28,1.28]$	0
f_8	$f_8(x)=\sum_{i=1}^{n}(-x_i\sin(\sqrt{\lvert x_i\rvert}))$	$[-500,500]$	-12569.5
f_9	$f_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]$	0
f_{10}	$f_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	$[-32,32]$	0
f_{11}	$f_{11}(x)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600,600]$	0
f_{12}	$f_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)\}+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+\sum_{i=1}^{n}u(x_i,10,100,4)]$，$y_i=1+\frac{1}{4}(x_i+1)$，$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\0, & -a\leqslant x_i\leqslant a\\k(-x_i-a)^m, & x_i<-a\end{cases}$	$[-50,50]$	0
f_{13}	$f_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+(x_n-1)[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	$[-50,50]$	0

f_1-f_4 为单峰函数，其中 f_3 作为单峰函数的典型，是一个具有局部极值点的凹函数，主要用于评估算法的收敛速度和搜索精度[17]。f_5-f_{13} 为多峰函数，其中 f_6-f_{12} 是基础函数，f_{13} 是拓展函数，而 f_5 是一个具有多个局部极值点的多模态函数，可用于测试算法的全局探索和跳出局部最优的能力。在实验过程中，每次实验都使用相同的参数来控制变量，种群规模为 100，迭代次数为 500，测试函数维度均为 30。每种测试函数有不同的上下限，范围如表 1 所列。种群规模较小可能导致算法早熟收敛，而种群规模过大则会增加计算复杂度和时间。100 作为一个中等规模的种群大小，可以在搜索精度和计算效率之间取得平衡。实验对每个算法在 13 种不同测试函数中分别进行 30 次独立实验，记录其平均值和标准差来检验算法的效果。在后续的实验中，均采用以上参数进行实验。

4.2 与单改进阶段的 EEFO 对比分析

4.2.1 仿真结果

为了验证本文所提策略的有效性，将 IEEFO 算法与 EEFO 算法和单策略改进的 EEFO 算法（只引入透镜成像反向学习的 REEFO 算法、只引入双曲正切能量因子的 HEEFO 算法、只引入扰动因子的 DEEFO 算法、只引入正弦余弦策略的 SEEFO 算法）在以上 13 个测试函数中进行对比实验，实验结果如表 2 所列。从表中可以看出，在单峰函数 f_1-f_4 中 IEEFO 和 REEFO 效果都为理论最优值，且标准差都为 0，效果最好；HEEFO 和 DEEFO 的最优值都优于 EEFO 且稳定性比 EEFO 强，在 f_5 中，IEEFO，REEFO 和 DEEFO 的最优值都为理论最优值，SEEFO 强于 REEFO 和 EEFO。在多峰函数 f_5-f_{13} 中，各算法表现都很出色，在 f_7 中相比于其他单策略改进算法，IEEFO 的最优值最小，标准差最小，性能最稳定。

表 2 IEEFO 与单改进 EEFO 算法的实验结果对比

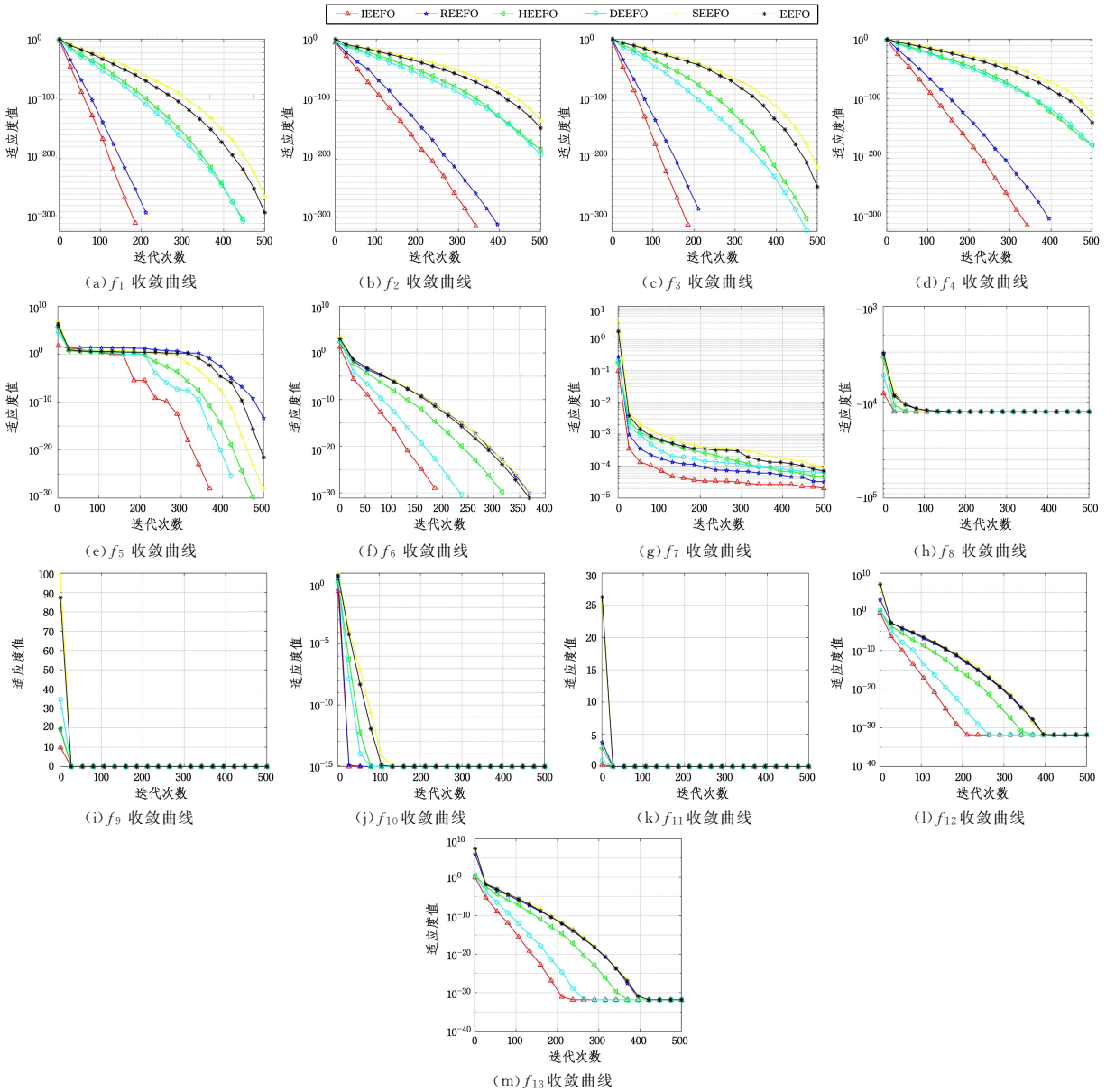
Table 2 Comparison of experimental results of IEEFO and single-strategy improved EEFO algorithm

函数名	评价	IEEFO	REEFO	HEEFO	DEEFO	SEEFO	EEFO
f_1	平均值	0	0	0	0	7.1459×10^{-265}	2.8894×10^{-292}
	标准差	0	0	0	0	0	0
f_2	平均值	0	0	2.5566×10^{-184}	6.1147×10^{-193}	4.4935×10^{-135}	1.5951×10^{-147}
	标准差	0	0	0	0	1.5723×10^{-134}	6.6309×10^{-147}
f_3	平均值	0	0	0	0	6.2242×10^{-214}	3.2364×10^{-248}
	标准差	0	0	0	0	0	0
f_4	平均值	0	0	1.4287×10^{-178}	2.6188×10^{-178}	4.5137×10^{-126}	1.5562×10^{-140}
	标准差	0	0	0	0	2.3729×10^{-125}	4.4813×10^{-140}
f_5	平均值	0	4.6828×10^{-14}	0	0	7.8901×10^{-29}	3.0870×10^{-22}
	标准差	0	2.5628×10^{-13}	0	0	4.3216×10^{-28}	1.6908×10^{-21}
f_6	平均值	0	0	0	0	0	0
	标准差	0	0	0	0	0	0
f_7	平均值	2.0330×10^{-5}	3.2154×10^{-5}	4.7268×10^{-5}	6.3692×10^{-5}	9.4062×10^{-5}	6.9841×10^{-5}
	标准差	2.3101×10^{-5}	2.5163×10^{-5}	5.0209×10^{-5}	5.8116×10^{-5}	1.1883×10^{-4}	5.1045×10^{-5}
f_8	平均值	−12569.4866	−12569.4866	−12569.4866	−12569.4866	−12569.4866	−12569.4866
	标准差	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}	1.8501×10^{-12}
f_9	平均值	0	0	0	0	0	0
	标准差	0	0	0	0	0	0
f_{10}	平均值	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}
	标准差	0	0	0	0	0	0
f_{11}	平均值	0	0	0	0	0	0
	标准差	0	0	0	0	0	0
f_{12}	平均值	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}	1.5705×10^{-32}
	标准差	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}
f_{13}	平均值	1.3498×10^{-32}	1.3498×10^{-32}	1.3498×10^{-32}	1.3498×10^{-32}	1.3498×10^{-32}	1.3498×10^{-32}
	标准差	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}	5.5674×10^{-48}

综上所述，IEEFO 算法相比单策略改进算法在寻优精度和稳定性上，在单峰函数 f_1-f_4 中表现出色，在多峰函数 f_5 和 f_7 中表现出色。

4.2.2 收敛曲线对比分析

IEEFO 算法和单策略改进阶段的 EEFO 算法的收敛图如图 7 所示。从图 7(a)－图 7(d)中可以看出：REEFO 算法相比其他单策略改进的 EEFO 算法在单峰测试函数 f_1-f_4 中表现更为出色，有更快的收敛速度；在相同的迭代次数下，REEFO 算法的精度也比其他单策略改进的 EEFO 算法高。从图 7(e)、图 7(f)、图 7(l)和图 7(m)中可以明显看出，HEEFO 算法和 DEEFO 算法相比 REEFO 算法在多峰函数 f_5，f_6，f_{12}，f_{13} 中有更快的收敛速度，证明 HEEFO 算法和 DEEFO 算法在处理复杂函数时优于 REEFO 算法。从图 7(a)－图 7(g)、图 7(l)和图 7(m)中可以看出，在 f_1-f_7，f_{12} 和 f_{13} 中 IEEFO 算法的收敛速度明显优于其他单改进阶段的算法。在单峰测试函数中，IEEFO 算法都能达到理论最优值；在多峰函数中，IEEFO 不仅不会陷入局部最优，且在 f_5-f_7，f_{12}，f_{13} 中都有更快的收敛速度。IEEFO 算法在增加算法全局探索精度的同时，保持了更高的稳定性，且收敛速度比单改进阶段的算法和原算法更快，这也证明了多策略改进的 EEFO 算法的有效性。



(a) f_1 收敛曲线　(b) f_2 收敛曲线　(c) f_3 收敛曲线　(d) f_4 收敛曲线

(e) f_5 收敛曲线　(f) f_6 收敛曲线　(g) f_7 收敛曲线　(h) f_8 收敛曲线

(i) f_9 收敛曲线　(j) f_{10} 收敛曲线　(k) f_{11} 收敛曲线　(l) f_{12} 收敛曲线

(m) f_{13} 收敛曲线

图 7　IEEFO 与单策略改进 EEFO 的收敛曲线对比

Fig. 7　Convergence curves comparison of IEEFO and single-strategy improved EEFO

4.3　与其他优化算法对比分析

4.3.1　仿真结果

为了验证 IEEFO 算法在性能上的提升，将 IEEFO 与 AHA[18]，DBO[19]，SSA[20]，PSO，WOA，EEFO 算法进行分析对比，通过以上 13 种测试函数进行 30 次测试验证，计算出每种算法最优解的平均值和标准差，结果如表 3 所列。

表 3　IEEFO 与基本算法的实验结果对比

Table 3　Experimental results comparison of IEEFO with basic algorithms

函数名	评价	IEEFO	AHA	DBO	SSA	PSO	WOA	EEFO
f_1	平均值	0	5.227×10^{-150}	1.597×10^{-125}	0	0.01202	1.7899×10^{-95}	3.323×10^{-289}
	标准差	0	2.645×10^{-149}	8.748×10^{-125}	0	0.013608	6.6523×10^{-95}	0
f_2	平均值	0	7.6152×10^{-77}	2.4335×10^{-69}	1.446×10^{-187}	0.012826	1.4658×10^{-56}	4.322×10^{-148}
	标准差	0	4.0931×10^{-76}	1.2806×10^{-68}	0	0.012481	5.2967×10^{-56}	1.925×10^{-147}
f_3	平均值	0	1.165×10^{-134}	4.002×10^{-46}	1.281×10^{-302}	775.7	15537.6073	2.6142×10^{-243}
	标准差	0	5.948×10^{-134}	2.192×10^{-45}	0	867.9324	8421.5249	0
f_4	平均值	0	2.4257×10^{-69}	1.944×10^{-50}	2.21×10^{-207}	3.9921	29.121	5.9793×10^{-141}
	标准差	0	7.8231×10^{-69}	7.4655×10^{-50}	0	0.87584	26.5347	2.0545×10^{-140}
f_5	平均值	0	25.7652	24.6081	9.2423×10^{-6}	93.8638	26.7923	1.9308×10^{-12}
	标准差	0	0.30366	0.32874	2.2862×10^{-5}	61.0068	0.22262	1.0575×10^{-11}
f_6	平均值	0	6.318×10^{-5}	4.8933×10^{-12}	1.1125×10^{-10}	0.013529	0.0043429	0
	标准差	0	3.6109×10^{-5}	6.1442×10^{-12}	2.5643×10^{-10}	0.012355	0.0018297	0
f_7	平均值	2.284×10^{-5}	9.2456×10^{-5}	0.0023512	0.00016429	0.022005	0.0012366	9.8102×10^{-5}
	标准差	2.257×10^{-5}	7.9152×10^{-5}	0.0022351	0.0001249	0.0052415	0.00090016	7.7758×10^{-5}

（续表）

函数名	评价	IEEFO	AHA	DBO	SSA	PSO	WOA	EEFO
f_8	平均值	−12569.49	−12371.827	−9413.7134	−8680.152	−8678.294	−11865.405	−12569.4866
	标准差	1.850×10^{-12}	191.9589	892.0628	581.8999	664.6934	1028.1527	1.8501×10^{-12}
f_9	平均值	0	0	23.1967	0	40.5784	0	0
	标准差	0	0	48.5051	0	13.6863	0	0
f_{10}	平均值	8.882×10^{-16}	8.8818×10^{-16}	1.0066×10^{-15}	8.8818×10^{-16}	0.033075	4.0856×10^{-15}	8.8818×10^{-16}
	标准差	0	0	6.4863×10^{-16}	0	0.023976	2.5294×10^{-15}	0
f_{11}	平均值	0	0	0	0	0.048504	0.0024073	0
	标准差	0	0	0	0	0.063143	0.013185	0
f_{12}	平均值	1.571×10^{-32}	1.3584×10^{-6}	1.7326×10^{-11}	5.4992×10^{-12}	0.0082816	0.0020116	1.5705×10^{-32}
	标准差	5.567×10^{-48}	8.7755×10^{-7}	1.9715×10^{-11}	9.3864×10^{-12}	0.026303	0.0071153	5.5674×10^{-48}
f_{13}	平均值	1.350×10^{-32}	1.0581	0.032332	3.6878×10^{-11}	0.010273	0.022782	1.3498×10^{-32}
	标准差	5.567×10^{-48}	0.533	0.050149	7.7539×10^{-11}	0.0098969	0.016834	5.5674×10^{-48}

从表 3 中可以看出，IEEFO 算法在单峰函数 f_1-f_4 中都展现出很好的效果，其 30 次测试的平均值均为 0，达到了理论最优值，并且对应的标准差都为 0，每次实验的寻优结果都相同，说明 IEEFO 算法有很高的精度和稳定性。在多峰函数 f_5-f_{13} 中，IEEFO 也表现出很好的效果，其中 f_5，f_6，f_9，f_{11} 的平均值及其标准差均为 0，说明 IEEFO 算法有较强的全局探索能力和跳出局部最优的能力。此外，f_7，f_8，f_{10}，f_{12} 的平均值也接近最优值，并且其对应的标准差非常小，说明在多峰函数中，IEEFO 也有较强的寻优能力。

综上所述，IEEFO 算法在 30 次测试验证下，无论是在单峰还是多峰函数中，都具有很强的寻优能力和鲁棒性。

4.3.2 收敛曲线对比分析

图 8 为 IEEFO 算法与 AHA，DBO，SSA，PSO，WOA，EEFO 算法在各个测试函数中的收敛曲线对比图。

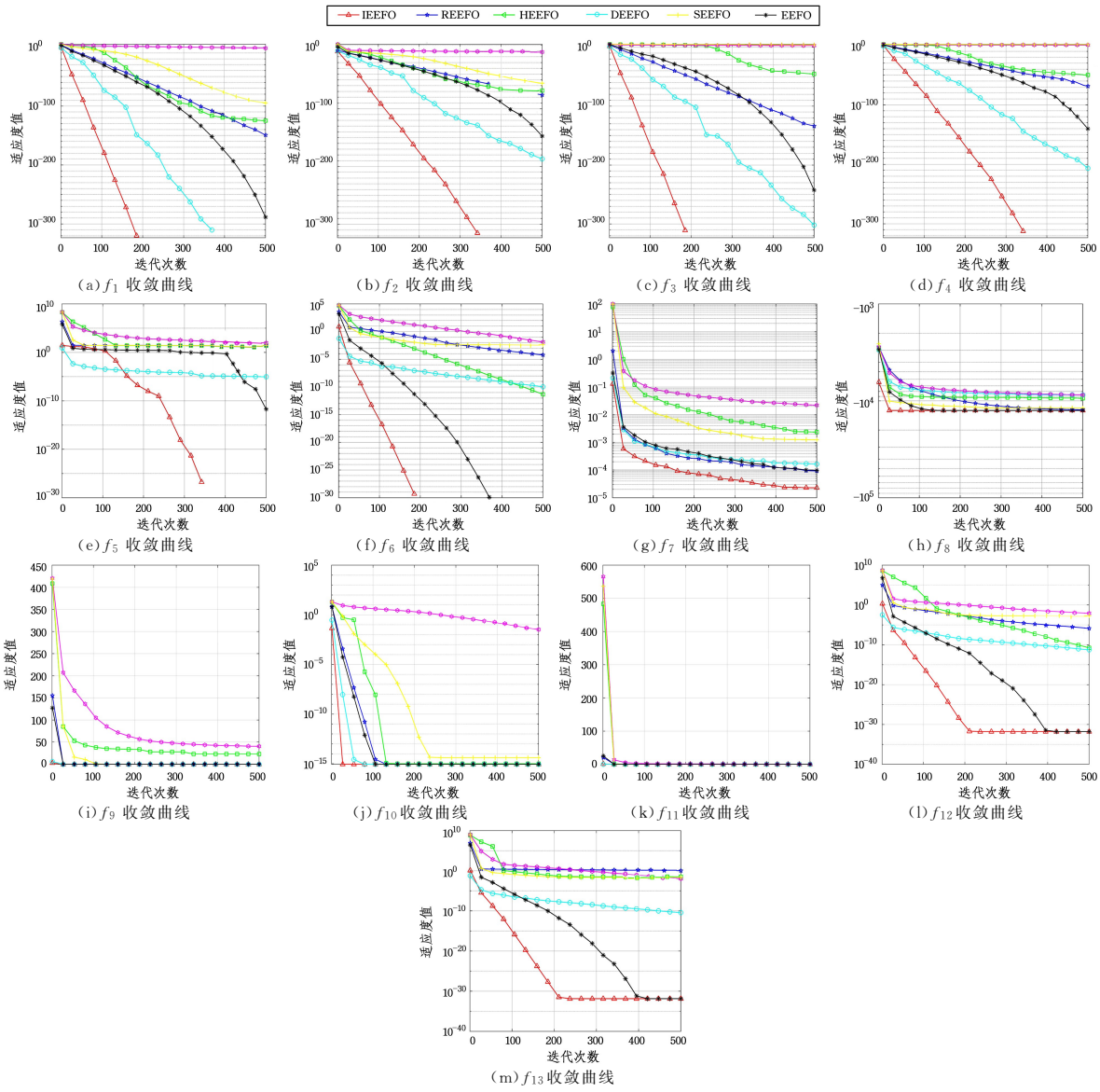


图 8 IEEFO 与基本算法的收敛曲线对比

Fig. 8 Convergence curves comparison of IEEFO and basic algorithms

从图 8(a)－图 8(d)中可以看出，相比其他算法，IEEFO 算法在 f_1-f_4 测试函数中能够快速找到最优值，且其平均值更接近最优值。从图 8(e)－图 8(m)中可以看出，IEEFO 算法在 f_5-f_{13} 的测试函数中收敛速度和搜寻精度都普遍高于其他优化算法。

综上所述，与以上 6 种优化算法相比，IEEFO 算法不论是在单峰测试函数还是在多峰测试函数中都表现良好，在收敛速度、寻优精度上都比其他算法更快、更精确。

5 机械优化设计问题

在工程设计与应用领域中，优化伸张/压缩弹簧机械设计问题是一个复杂且常见的挑战。传统的机械方法在处理这类问题时，尤其是面对非线性和高维数值优化问题时，往往难以找到有效的解决方案。区别于传统方法，将本文 IEEFO 算法用于优化伸张/压缩弹簧机械设计问题，进一步验证改进算法的可行性和适用性。

5.1 伸张/压缩弹簧优化设计案例

伸张/压缩弹簧设计问题的优化目标是减少弹簧的重量，同时满足一系列的约束条件。图 9 为伸张/压缩弹簧结构示意图。构建伸张/压缩弹簧设计问题的数学模型，约束条件包括最小偏差 g_1、剪切应力 g_2、冲击频率 g_3、外径限制 g_4，决策变量包括线径 d、平均线圈直径 D 及有效线圈数 P。设 $f(x)$ 为弹簧重量，$\min f(x)$ 为最小化弹簧重量，$x=[x_1,x_2,x_3]=[d,D,P]$。其目标函数和约束条件如式(35)和式(36)所示：

$$\min f(x)=x_1^2x_2(2+x_3) \tag{35}$$

$$\begin{cases} g_1(x)=1-\dfrac{x_2^3x_3}{71\,785x_1^4}\leqslant 0 \\ g_2(x)=\dfrac{4x_2^2-x_1x_2}{12\,566(x_2x_1^3-x_1^4)}+\dfrac{1}{5\,108x_1^2}-1\leqslant 0 \\ g_3(x)=1-\dfrac{140.45x_1}{x_2^2x_3}\leqslant 0 \\ g_4(x)=\dfrac{x_1+x_2}{1.5}-1\leqslant 0 \end{cases} \tag{36}$$

其中，$0.05\leqslant x_1\leqslant 2$，$0.25\leqslant x_2\leqslant 1.3$，$2\leqslant x_3\leqslant 15$。

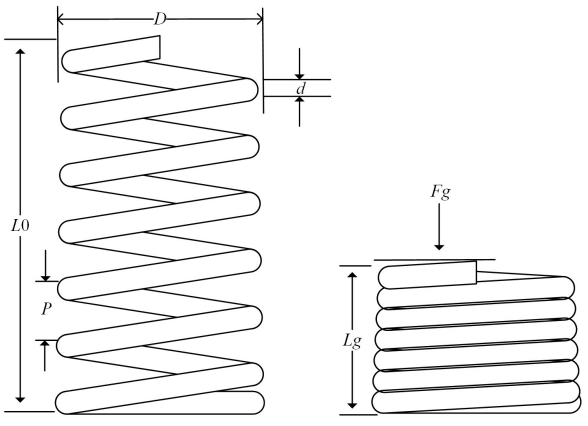


图 9 伸张/压缩弹簧结构示意图

Fig. 9 Schematic diagram of tension/compression spring structure

5.2 测试结果与分析

将本文 IEEFO 算法与 AHA 算法、PSO 算法、WOA 算法、EEFO 算法应用于优化伸张/压缩弹簧机械设计问题进行实验，实验选取种群规模为 100，最大迭代次数为 500，每种算法独立运行 30 次取平均值。图 10 为优化伸张/压缩弹簧的收敛曲线图。

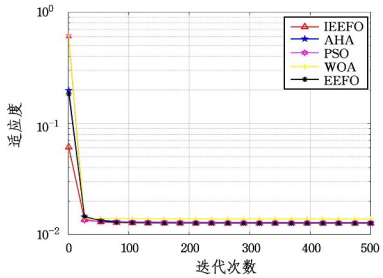


图 10 优化伸张/压缩弹簧的收敛曲线

Fig. 10 Convergence curves of optimized tension/compression spring

表 4 为各算法伸张/压缩弹簧的优化结果。从表中可以看到 IEEFO 算法与其他算法的处理约束函数值，IEEFO 算法获得的函数最优解为 $[x_1,x_2,x_3]=[0.0518,0.3598,11.1087]$，最优值 $f(x)=0.0127$，表示 IEEFO 在解决该伸张/压缩弹簧设计问题上给出了最佳解决方案，进一步证明了 IEEFO 算法在实际应用中的有效性和可行性。

表 4 伸张/压缩弹簧设计问题中各算法的最优解

Table 4 Optimal solutions of various algorithms for tension/compression spring design problems

算法	d	D	P	$f(x)$
IEEFO	0.0518	0.3598	11.1087	0.0127
AHA	0.0517	0.3580	11.2166	0.0127
PSO	0.0548	0.4360	7.8034	0.0128
WOA	0.0597	0.5822	4.6248	0.0138
EEFO	0.0517	0.3572	11.2615	0.0127

结束语 针对电鳗觅食优化算法(EEFO)在迭代过程中存在全局探索能力不足、容易陷入局部最优和收敛速度慢的问题，同时算法的性能受到参数设置的影响较大，需要进一步优化调整算法参数，本文提出了多策略改进的电鳗觅食优化算法 IEEFO。首先提出双曲正切能量因子策略改进了算法的能量因子，在算法前期加入开发行为，以快速地发现最优种群，加速算法的收敛；其次提出扰动因子 q，扩大电鳗游走位置范围，有利于种群全局寻优；之后在算法迁徙阶段加入正弦余弦策略，促进算法局部开发；最后在算法每次迭代之后加入透镜成像反向学习策略，扩大种群的搜索空间，有利于算法跳出局部最优并加速收敛，找到最优解。将改进后的算法与 6 种基本算法、4 种单策略改进的 EEFO 算法在 13 个基准函数上进行对比实验，得出结论：IEEFO 算法相比其他基本算法和单策略改进的 EEFO 算法，收敛速度更快，全局寻优能力更强，稳定性更高。将 IEEFO 算法应用在优化机械设计问题中，进一步验证了算法的有效性和适用性，为解决复杂的工程优化问题提出了一种新方法。下一步考虑将 IEEFO 算法应用在更多工程问题中，在不同的环境下进一步改进算法，以满足求解问题的实际需求。

参 考 文 献

[1] LIU J，HOU Y，LI Y，et al. Advanced strategies on update

mechanism of tree-seed algorithm for function optimization and engineering design problems[J]. Expert Systems with Applications, 2024, 236: 121312.
[2] BRAIK M S. Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems[J]. Expert Systems with Applications, 2021, 174: 114685.
[3] WEI F, ZHANG Y, LI J. Multi-strategy-based adaptive sine cosine algorithm for engineering optimization problems[J]. Expert Systems with Applications, 2024, 248: 123444.
[4] WANG D, TAN D, LIU L. Particle swarm optimization algorithm: an overview[J]. Soft Computing, 2018, 22(2): 387-408.
[5] NADIMI-SHAHRAKI M H, TAGHIAN S, MIRJALILI S. An improved grey wolf optimizer for solving engineering problems [J]. Expert Systems with Applications, 2021, 166: 113917.
[6] ZHENG B, CHEN Y, WANG C, et al. The Moss Growth Optimization(MGO): concepts andperformance[J]. Journal of Computational Design and Engineering, 2024, 11(5): 184-221.
[7] ABDEL-BASSET M, MOHAMED R, ABOUH-AWWASH M. Crested Porcupine Optimizer: A new nature-inspired metaheuristic[J]. Knowledge-Based Systems, 2024, 284: 111257.
[8] ABDEL-BASSET M, MOHAMED R, SALL-AM K M, et al. Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm[J]. Mathematics, 2022, 10(19): 3466.
[9] TROJOVSKÁ E, DEHGHANI M, TROJOVSKÝ P. Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm[J]. IEEE Access, 2022, 10: 49445-49473.
[10] ZOLFI K. Gold rush optimizer: A new population-based metaheuristic algorithm [J]. Operations Research and Decisions, 2023; 33(1): 113-150.
[11] ZHAO W G, WANG L Y, ZHANG Z X, et al. Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications[J]. Expert Systems with Applications, 2024, 238: 122200.
[12] HOU Y, GAO H, WANG Z, et al. Improved grey wolf optimization algorithm and application[J]. Sensors, 2022, 22(10): 3810.
[13] RIZK-ALLAH R M. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems[J]. Journal of Computational Design and Engineering, 2018, 5(2): 249-273.
[14] LI Z, FENG F. An Artificial Hummingbird Algorithm Based on Multi-strategy Improvement [J]. Computer Science, 2024, 51(S1): 100-108.
[15] AKAY R, YILDIRIM M Y. Multi-strategy and self-adaptive differential sine-cosine algorithm for multi-robot path planning[J]. Expert Systems with Applications, 2023, 232: 120849.
[16] HE Y, WANG M. An improved chaos sparrow search algorithm for UAV path planning[J]. Scientific Reports, 2024, 14(1): 366.
[17] YIN P, TAN G G, SONG W, et al. Comparative Study on Improved Tuna Swarm Optimization Algorithm Based on Chaotic Mapping [J]. Computer Science, 2024, 51(S1): 273-282.
[18] ZHAO W G, WANG L Y, MIRJALILI S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications[J]. Computer Methods in Applied Mechanics and Engineering, 2022, 388: 114194.
[19] XUE J K, SHEN B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. Journal of Supercomputing, 2023, 79(7): 7305-7336.
[20] JAIN M, SINGH V, RANI A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm[J]. Swarm and Evolutionary Computation, 2019, 44: 148-175.

WANG Xinwei, born in 2000, postgraduate, is a member of CCF (No. U1374G). His main research interest is intelligent optimization algorithm improvements and applications.

FENG Feng, born in 1971, professor, Ph.D supervisor. His main research interests include information system engineering and application and so on.

(责任编辑:何杨)