

引用格式:郭雨鑫,刘升,高文欣,等. 多策略改进哈里斯鹰优化算法[J]. 微电子学与计算机,2021,38(7):18-24. [ GUO Y X, LIU S,GAO W X,et al. Improved harris hawks optimization algorithm with multiple strategies [J]. Microelectronics & Computer,2021,38(7):18-24. ]

# 多策略改进哈里斯鹰优化算法

郭雨鑫,刘 升,高文欣,张 磊  
(上海工程技术大学 管理学院,上海 201620)

**摘 要:**为解决基本哈里斯鹰算法(Harris hawks optimization,HHO)易陷入局部最优和收敛精度低的问题,提出多策略优化的哈里斯鹰优化算法(Multi-Strategy Harris hawks optimization,MHHO).在探索阶段,引入柯西分布函数变异全局位置,增加种群多样性;在过渡阶段,利用随机收缩指数函数非线性化能量方程,更好地协调全局探索和局部开采;在开采阶段,引入自适应权重因子更新局部位置,提高局部开采能力.通过求解多个单峰、多峰和高维度测试函数,结果表明融合三种策略的 MHHO 算法具有更好的寻优精度和稳定性.

**关键词:**哈里斯鹰优化算法;柯西变异;随机收缩指数函数;自适应权重;高维优化

**中图分类号:** TP18                      **文献标识码:** A                      **文章编号:** 1000-7180(2021)07-0018-07

## Improved Harris Hawks optimization algorithm with multiple strategies

GUO Yuxin,LIU Sheng,GAO Wenxin,ZHANG Lei

(School of Management,Shanghai University of Engineering and Technology,Shanghai 201620,China)

**Abstract:** In order to solve the problems of the basic Harris hawks optimization (HHO) that are easy to fall into local optimization and low convergence accuracy,a multi-strategy optimization Harris hawks optimization (MHHO) is proposed. In the exploration stage, the Cauchy distribution function is introduced to mutate the global position which increases population diversity. In the transition stage, the random contraction exponential function is used to nonlinear the energy equation in order to coordinate global exploration and local mining effectively. In the mining stage, the adaptive weight factor updates the local position and elevates the local mining capacity. By solving multiple single peak, multi peak and high-dimensional test functions,the results show that the MHHO algorithm combining the three strategies has better optimization accuracy and stability.

**Key words:** Harris Hawks optimization algorithm; Cauchy mutation; random contraction index function; adaptive weight; high-dimensional optimization

### 1 引言

自然启发算法或元启发式方法是对自然界简单行为的模拟,成功弥补传统优化方法的不足<sup>[1]</sup>,可以有效处理复杂的现实问题.哈里斯鹰算法(Harris Hawks Optimization,HHO)是由 Heidari 等人<sup>[2]</sup>在

2019 年提出的一种自然启发算法,源于观察哈里斯鹰和它们的猎物(兔子)之间的追逐逃跑行为.该算法自提出以来一直受到广泛关注.陈慧玲等<sup>[3]</sup>使用哈里斯鹰算法进行光伏电池和模块的参数识别,研究表明该新型算法具有良好的性能.

然而,与其他群智能算法相似,传统的哈里斯鹰

算法结构本身存在一定缺陷,搜索过程容易出现陷入局部最优、收敛精度低的问题.为了改进哈里斯鹰算法的不足,赵世杰等<sup>[4]</sup>提出能量周期性递减机制和牛顿局部增强策略的改进方法,增强了HHO算法的开采能力.Gupta等<sup>[5]</sup>使用新的过渡规则和对立学习的方法改进HHO算法,该改进算法可以有效进行全局优化.Kamboj等<sup>[6]</sup>提出了混合哈里斯鹰—正弦余弦算法,加速算法的全局探索阶段.

现存的HHO改进策略,虽然在一定程度上改进了算法的探索寻优性能,但是一般只是针对哈里斯鹰算法中某一个更新策略进行改进或者改变能量降低方式,并没有有效地避免全局和局部寻优的盲目性.针对以上问题,提出了融合柯西变异、随机收缩指数函数以及自适应权重的多策略改进哈里斯鹰优化算法(Multi-Strategy Harris Hawks Optimization, MHHO).

## 2 哈里斯鹰优化算法(HHO)

HHO算法用数学公式来模拟现实中哈里斯鹰在不同机制下捕捉猎物的策略.在HHO中,哈里斯鹰是候选解,猎物随迭代逼近最优解.HHO算法包括两个阶段:全局探索阶段、局部开采阶段.

### 2.1 全局探索阶段

在这一阶段中,哈里斯鹰处于等待状态,仔细检查和监控搜索空间 $[lb, ub]$ 以发现猎物.它根据两种策略在随机的地方寻找猎物,迭代时以概率 $q$ 进行位置更新,数学表达式为:

$$X_{t+1} = \begin{cases} X_{\text{rand}} - r_1 |X_{\text{rand},t} - 2r_2 X_t|, & q \geq 0.5 \\ (X_{\text{rabb},t} - X_{m,t}) - r_3(lb + r_4(ub - lb)), & q < 0.5 \end{cases} \quad (1)$$

式中, $X_{t+1}$ 和 $X_t$ 分别为哈里斯鹰第 $t+1$ 次和第 $t$ 次迭代时的位置, $X_{\text{rabb},t}$ 表示猎物第 $t$ 次迭代时的位置, $q$ 和 $r_1, r_2, r_3, r_4$ 是区间 $(0,1)$ 内的随机数字, $lb$ 是搜索空间的下界, $ub$ 是搜索空间的上界, $X_{\text{rand},t}$ 表示第 $t$ 次迭代时哈里斯鹰的随机位置, $X_{m,t}$ 表示第 $t$ 次迭代时哈里斯鹰的平均位置,公式如下:

$$X_{m,t} = \frac{1}{N} \sum_{i=1}^N X_{i,t} \quad (2)$$

### 2.2 过渡阶段

任何群智能算法的精确运行,需要保持探索和开采之间适当的平衡.HHO通过猎物的能量方程实现从探索到开采的过渡,其模型如下:

$$E = 2E_0(1 - \frac{t}{T}) \quad (3)$$

式中, $E$ 表示猎物逃跑的能量, $E_0$ 是猎物能量的初始状态,公式为 $E_0 = 2 * \text{rand} - 1$ , $\text{rand}$ 是 $(0,1)$ 之间的随机数字, $T$ 为最大迭代次数, $t$ 为当前迭代次数.当 $|E| \geq 1$ 时,哈里斯鹰算法将执行全局探索;否则,HHO算法进入局部开采.

### 2.3 局部开采阶段

在这一阶段,哈里斯鹰根据前一阶段的检测执行突袭攻击预期猎物,而猎物试图逃离危险.根据猎物的逃跑行为和哈里斯鹰的追逐策略,HHO算法提出了四种可能的策略来模拟攻击行为.用 $\lambda$ 表示猎物成功逃脱的概率,当 $\lambda < 0.5$ 时,猎物逃脱成功;当 $\lambda \geq 0.5$ 时,猎物逃脱失败.用参数 $E$ 模拟哈里斯鹰软或硬的围攻策略.当 $|E| \geq 0.5$ 时,执行软围攻;否则,执行硬围攻.

Case1:软围攻.当 $|E| \geq 0.5, \lambda \geq 0.5$ 时,猎物有足够的能量且以跳跃的方式逃脱围捕,而哈里斯鹰会逐渐消耗猎物的能量,然后选择最佳的位置突袭俯冲逮捕猎物.更新位置的方程如下:

$$X_{t+1} = \Delta X_t - E |J X_{\text{rabb},t} - X_t| \quad (4)$$

$$\Delta X_t = X_{\text{rabb},t} - X_t \quad (5)$$

式中, $X_t$ 是迭代时猎物与哈里斯鹰的位置之差, $J = 2(1 - r_5)$ 表示猎物逃跑过程中的随机跳跃, $r_5$ 是介于0到1之间的随机数字.

Case2:硬围攻.当 $|E| < 0.5, \lambda \geq 0.5$ 时,猎物筋疲力尽,哈里斯鹰选择迅速突袭.位置更新如下:

$$X_{t+1} = X_{\text{rabb},t} - E |\Delta X_t| \quad (6)$$

Case3:累速俯冲式软围攻.当 $|E| \geq 0.5, \lambda < 0.5$ 时,猎物有足够的能量 $E$ 逃跑,哈里斯鹰在突袭之前会建立一个软围攻.为了模拟猎物的逃跑模式和跳跃动作,将levy函数LF集成在HHO算法中.更新位置的策略为:

$$X_{t+1} = \begin{cases} Y: X_{\text{rabb},t} - E |J X_{\text{rabb},t} - X_t|, & \text{if } F(Y) < F(X_t) \\ Z: Y + S \times \text{LF}(D), & \text{if } F(Z) < F(X_t) \end{cases} \quad (7)$$

式中, $D$ 为问题维度, $S$ 为 $D$ 维随机行向量.

Case4:累速俯冲式硬围攻.当 $|E| < 0.5, \lambda < 0.5$ 时,猎物能量 $E$ 低,哈里斯鹰在突袭前构建硬围攻捕捉猎物,位置更新如下:

$$X_{t+1} = \begin{cases} Y: X_{\text{rabb},t} - E |J X_{\text{rabb},t} - X_{m,t}|, & \text{if } F(Y) < F(X_t) \\ Z: Y + S \times \text{LF}(D), & \text{if } F(Z) < F(X_t) \end{cases} \quad (8)$$

HHO 算法用猎物能量  $E$  和因子  $\lambda$  调节哈里斯鹰和猎物(兔子)之间的四种围捕机制,来实现优化求解问题。

### 3 多决策改进哈里斯鹰优化算法

为了改进 HHO 算法现存的问题,论文从三个方面对 HHO 算法进行改进。首先,在全局探索中,通过引入柯西分布函数的方法变异哈里斯鹰位置信息,提高 HHO 算法的全局搜索能力;其次,通过随机收缩指数函数修正能量递减机制,有效调节过渡;最后,通过引入自适应权重因子更新局部位置,提高 HHO 算法的局部搜索能力。因此论文提出一种多策略改进的哈里斯鹰优化算法(MHHO)。

#### 3.1 柯西变异

针对 HHO 优化算法易陷入局部最优值的问题,利用柯西(Cauchy)分布函数增加哈里斯鹰种群多样性,增加搜索空间,提高 HHO 算法的全局搜索能力。论文受文献[7]的启发,结合了柯西算子,充分利用柯西分布函数两端的变异效应优化全局最优对象,标准柯西分布函数公式如下:

$$f(x) = \frac{1}{\pi} \left( \frac{1}{x^2 + 1} \right) \quad (9)$$

Cauchy 函数的峰值相对较小,哈里斯鹰在柯西变异后会更多地搜索全局最优值,使用相对较少的时间开采局部区间;另外,Cauchy 函数从峰值向两侧下降相对平缓,在通过柯西变异更新位置后,哈里斯鹰受局部极值点的约束力下降,可以跳出局部极值。柯西变异的数学模型为获得当前全局最优解  $X_{\text{best}}$  后以下列公式更新最优解:

$$X'_{\text{best}} = X_{\text{best}} + X_{\text{best}} \times \text{Cauchy}(0, 1) \quad (10)$$

#### 3.2 随机收缩指数函数

在 HHO 算法中,猎物能量  $E$  的大小起着调节和过渡全局搜索和局部开采的重要作用, $E$  越小,HHO 算法越倾向于执行局部开采; $E$  越大,算法越倾向于进行全局探索。但在传统 HHO 算法的能量方程中, $E$  由最大值到最小值呈线性递减,并不能有效描述自然界中哈里斯鹰群体围捕猎物的真实过程。数学家模拟捕食者—猎物的相互作用<sup>[8]</sup>,得出这样的结论:随机收缩指数函数更适合于表达猎物逃跑时的能量变化。因此,论文提出修正能量线性递减调控机制,将随机收缩指数函数融入猎物能量  $E$  的递减过程,其对应的能量方程为:

$$E = 2E_0(2\text{rand} \times \exp(-(\frac{\pi}{2} \times \frac{t}{T})) \quad (11)$$

#### 3.3 自适应权重

惯性加权因子是非常重要的参数。当惯性加权因子较大时,该算法使用相对较多的时间进行的全局搜索;当惯性加权因子较小时,该算法使用相对较多的时间进行局部搜索,可以精细地寻找最佳解决方案。猎物位置代表 HHO 算法当前最优解,同时反向引导哈里斯鹰栖息地的位置,它的迭代更新直接决定了算法的优化性能。因此,为了提高算法的局部开采能力,有必要对猎物位置的邻域进行再更新,以找出更优解。

考虑到上述情况,受文献[9]的启发,引入了一种自适应权重方法。在四种围捕机制中,哈里斯鹰搜寻猎物时,猎物会使用较小的自适应权重更新位置,提高哈里斯鹰算法局部寻优能力。自适应权重公式和猎物位置更新如式(12)、(13)所示:

$$\omega = \sin\left(\frac{\pi \times t}{2T} + \pi\right) + 1 \quad (12)$$

$$X'_{\text{rabbit}} = \omega \times X_{\text{rabbit}} \quad (13)$$

式中, $T$  为最大迭代次数, $t$  为当前迭代次数。通过融合自适应权重因子  $\omega$ ,使 MHHO 算法具有更好的局部开采能力。

#### 3.4 改进算法的执行流程

多策略改进的哈里斯鹰算法(MHHO),从多方面增强哈里斯鹰算法的寻优性能,算法流程如图 1 所示。

### 4 实验仿真与结果分析

#### 4.1 仿真实验环境

本仿真测试环境为:操作系统 Win7、64 位操作系统, CPU 为 AMD A8-7410 APU with AMD Radeon R5 Graphics,主频 2.2 GHz,内存 4 GB,仿真软件为 Matlab2018b。

#### 4.2 初始参数设置

论文选取了两种新颖且寻优效果较好的自然启发算法——蝴蝶算法(Butterfly Optimization Algorithm, BOA)<sup>[10]</sup>、鲸鱼优化算法(Whale Optimization Algorithm, WOA)<sup>[11]</sup>,以及基本哈里斯鹰优化算法(HHO)与多策略改进的哈里斯鹰算法(MHHO)进行对比。为了体现实验的公平性和客观性,所有算法种群规模  $N$  设为 30,迭代次数  $T$  设为 500,四种算法共有参数保持一致。

#### 4.3 测试函数

为了进一步验证 MHHO 的优化效果,论文选取 8 个不同特点的测试函数进行函数优化的对比测试,包括四个( $f_1 \sim f_4$ )单峰函数、四个( $f_5 \sim f_8$ )多峰函数,其具体信息如表 1 所示。

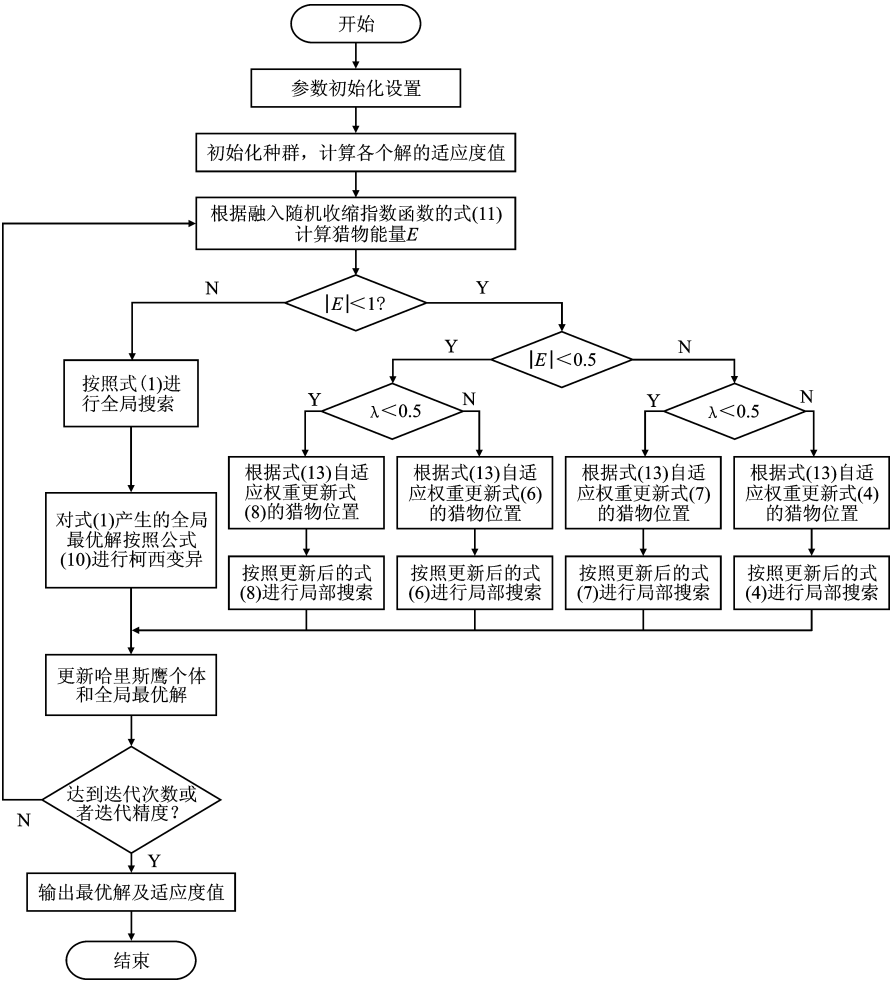


图 1 MHHO 算法的流程图

表 1 测试函数

| 函数                                                                                                  | 函数名称          | 维度 | 范围            | 最优值 |
|-----------------------------------------------------------------------------------------------------|---------------|----|---------------|-----|
| $f_1 = \sum_{i=1}^n x_i^2$                                                                          | Sphere        | 30 | $[-100, 100]$ | 0   |
| $f_2 = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $                                                    | Schwefel 2.22 | 30 | $[-10, 10]$   | 0   |
| $f_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$                                                           | Schwefel 1.2  | 30 | $[-100, 100]$ | 0   |
| $f_4 = \max_i \{  x_i , 1 \leq i \leq n \}$                                                         | Schwefel 2.21 | 30 | $[-100, 100]$ | 0   |
| $f_5 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | Griewank      | 30 | $[-600, 600]$ | 0   |
| $f_6 = \sum_{i=1}^d  x_i \sin(x_i)  + 0.1x_i$                                                       | Alpine        | 30 | $[-10, 10]$   | 0   |
| $f_7 = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^d x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^d  x_i^2 }$       | Salomon       | 30 | $[-100, 100]$ | 0   |
| $f_8 = 0.5 + \frac{\sin^2(\sqrt{x_i^2 + x_j^2} - 0.5)}{(1.0 + 0.001(x_i^2 + x_j^2))^2}$             | Schaffter     | 2  | $[-100, 100]$ | 0   |

4.4 实验结果与分析

为了验证论文 MHHO 算法改进的有效性,论文对算法性能在 8 个测试函数上进行了试验,蝴蝶

算法(BOA)<sup>[10]</sup>、鲸鱼优化算法(WOA)<sup>[11]</sup>、基本哈里斯鹰算法(HHO)、多策略改进的哈里斯鹰算法(MHHO)在各个标准测试函数上独立运行 30 次的

结果如表 2 所示。

表 2 测试函数实验结果

| 函数    | 算法   | 平均值       | 标准差       | 最差值       | 最优值       |
|-------|------|-----------|-----------|-----------|-----------|
| $f_1$ | BOA  | 1.28E-11  | 7.91E-13  | 1.41E-11  | 1.14E-11  |
|       | WOA  | 2.01E-74  | 1.07E-73  | 5.88E-73  | 8.11E-87  |
|       | HHO  | 9.61E-95  | 4.87E-94  | 2.67E-93  | 7.97E-117 |
|       | MHHO | 6.29E-254 | 0         | 1.88E-252 | 0         |
| $f_2$ | BOA  | 4.20E-09  | 1.30E-09  | 5.68E-09  | 1.84E-09  |
|       | WOA  | 2.33E-52  | 6.61E-52  | 3.24E-51  | 4.83E-61  |
|       | HHO  | 7.02E-51  | 3.17E-50  | 1.71E-49  | 5.95E-58  |
|       | MHHO | 2.63E-110 | 1.44E-109 | 7.89E-109 | 1.95E-175 |
| $f_3$ | BOA  | 1.22E-11  | 1.12E-12  | 1.43E-11  | 9.73E-12  |
|       | WOA  | 4.68E+04  | 1.29E+04  | 8.31E+04  | 2.68E+04  |
|       | HHO  | 1.82E-77  | 9.07E-77  | 4.97E-76  | 6.61E-96  |
|       | MHHO | 1.27E-224 | 0         | 3.81E-223 | 0         |
| $f_4$ | BOA  | 6.13E-09  | 3.45E-10  | 6.84E-09  | 5.18E-09  |
|       | WOA  | 5.35E+01  | 2.96E+01  | 9.01E+01  | 1.42E-01  |
|       | HHO  | 5.73E-49  | 2.11E-48  | 8.75E-48  | 1.02E-55  |
|       | MHHO | 1.72E-112 | 9.41E-112 | 5.15E-111 | 5.80E-168 |
| $f_5$ | BOA  | 4.47E-12  | 2.61E-12  | 1.13E-11  | 9.63E-13  |
|       | WOA  | 4.01E-3   | 2.19E-02  | 1.20E-01  | 0         |
|       | HHO  | 0         | 0         | 0         | 0         |
|       | MHHO | 0         | 0         | 0         | 0         |
| $f_6$ | BOA  | 1.35E-09  | 3.35E-09  | 1.83E-08  | 1.85E-10  |
|       | WOA  | 8.46E-01  | 4.64      | 2.54E+01  | 2.02E-55  |
|       | HHO  | 1.34E-51  | 3.28E-51  | 1.20E-50  | 2.30E-57  |
|       | MHHO | 4.35E-114 | 2.38E-113 | 1.31E-112 | 8.74E-176 |
| $f_7$ | BOA  | 9.31E-01  | 5.67E-02  | 9.41E-01  | 6.33E-01  |
|       | WOA  | 3.86E-01  | 2.13E-01  | 9.42E-01  | 2.85E-40  |
|       | HHO  | 1.71E-48  | 7.04E-48  | 3.71E-47  | 2.24E-57  |
|       | MHHO | 5.12E-121 | 2.79E-120 | 1.53E-119 | 0         |
| $f_8$ | BOA  | 3.75E-02  | 5.58E-03  | 3.75E-02  | 1.47E-08  |
|       | WOA  | 5.51E-03  | 4.89E-03  | 9.71E-03  | 0         |
|       | HHO  | 0         | 0         | 0         | 0         |
|       | MHHO | 0         | 0         | 0         | 0         |

从表 2 的测试结果中,很容易看出对于所选测试函数,MHHO 算法的寻优性能最强,明显优于 BOA<sup>[10]</sup>、WOA<sup>[11]</sup> 和 HHO 算法. 函数  $f_1$ 、 $f_3$ 、 $f_5$ 、 $f_7$ 、 $f_8$  寻优效果达到 100%,可以直接搜索到最优值 0;对于函数  $f_1$ 、 $f_3$ 、 $f_5$ 、 $f_8$ ,改进算法的标准差为 0,说明其具有较高稳定性;对于函数  $f_2$ ,基础 HHO 算法寻优能力高于 BOA 算法<sup>[10]</sup>,但是略低于 WOA 算法<sup>[11]</sup>,改进后的 MHHO 寻优效果明显高于其他三种算法,其最优值达到 1.95E-175 的水平,比基础的 HHO 算法提高了上百个数量级的寻优精度;对于函数  $f_4$ 、 $f_6$ ,MHHO 算法的最优值比 HHO 算法同样提高了超过 100 个精度,说明改进的 MHHO 算法克服了基本 HHO 算法寻优精度不高的问题,极大地提高了基本算法的寻优性能,与其他算法对比具有明显的竞争优势.

为了直观展示 MHHO 算法的寻优性能,论文

给出部分基准测试函数的收敛曲线图,如图 2 所示.

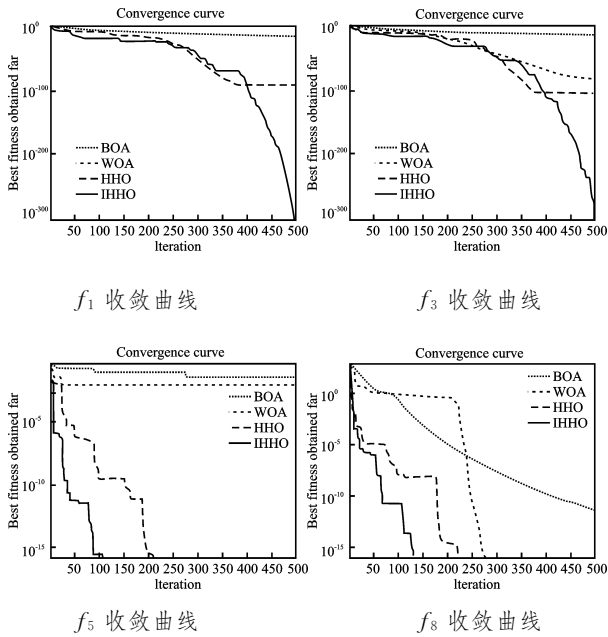


图 2 收敛曲线图

由函数的收敛曲线可知,改进的 MHHO 在收敛精度上远远高于 BOA、WOA、HHO 算法,并且在部分测试函数的收敛速度上也快于其他三种算法.  $f_1 \sim f_4$  是单峰函数常常用来测试算法的收敛能力,由函数  $f_1$  和  $f_3$  收敛曲线图可以直观看出改进后 MHHO 的收敛精度大幅提高; $f_5 \sim f_8$  是多峰函数,WOA、HHO、MHHO 对于函数  $f_5$  和  $f_8$  的最优值均为 0,但由收敛曲线图可知 MHHO 的收敛速度最快,其在 100 到 150 代之间达到最优值,说明改进后的算法在某些测试函数上具有更快的收敛速度; $f_8$  是低维测试函数,由收敛曲线图可知在迭代过程中出现了多个拐点,证明改进后的算法容易跳出局部最优值,具有更好的全局寻优效果.

4.5 求解高维函数的实验分析

根据上述实验结果可知,对于低维测试函数,论文改进的哈里斯鹰算法在最优值和标准差上都取得了较好的收敛效果.但是一般的改进策略在较为复杂的工程问题上容易陷入“维数灾难”,为了验证 MHHO 算法对高维函数求解的适应性,更好地测试改进算法能够有效地应用于求解大规模复杂问题,论文进行 MHHO 算法的高维函数测试实验,在 100 维、500 维、1 000 维和 10 000 维的测试函数上独立运行 30 次,对比 HHO 与 MHHO,种群规模 N 设置为 30,迭代次数 T 为 500,由于函数  $f_3$  耗时过长,函数  $f_8$  为低维函数,所以选取其它六个函数作为高维测试函数,所得实验结果见表 3.

表 3 MHHO 高维函数测试结果

| 函数    | 维度     | HHO      |          |          |           | MHHO             |                  |                  |                  |
|-------|--------|----------|----------|----------|-----------|------------------|------------------|------------------|------------------|
|       |        | 平均值      | 标准差      | 最差值      | 最优值       | 平均值              | 标准差              | 最差值              | 最优值              |
| $f_1$ | 100    | 1.99E-96 | 6.81E-96 | 3.46E-95 | 8.77E-117 | <b>1.19E-228</b> | <b>0</b>         | <b>2.39E-227</b> | <b>0</b>         |
|       | 500    | 2.51E-94 | 9.29E-94 | 3.91E-93 | 1.62E-110 | <b>1.15E-234</b> | <b>0</b>         | <b>3.46E-233</b> | <b>0</b>         |
|       | 1 000  | 1.20E-92 | 4.57E-92 | 1.91E-91 | 3.39E-110 | <b>1.15E-224</b> | <b>0</b>         | <b>3.46E-223</b> | <b>0</b>         |
|       | 10 000 | 1.11E-90 | 5.86E-90 | 3.21E-89 | 5.42E-110 | <b>4.25E-228</b> | <b>0</b>         | <b>1.27E-226</b> | <b>0</b>         |
| $f_2$ | 100    | 9.77E-49 | 5.33E-48 | 2.92E-47 | 5.17E-57  | <b>3.36E-107</b> | <b>1.84E-106</b> | <b>1.01E-105</b> | <b>1.90E-167</b> |
|       | 500    | 6.64E-50 | 3.18E-49 | 1.74E-48 | 2.73E-58  | <b>4.61E-117</b> | <b>2.52E-116</b> | <b>1.38E-115</b> | <b>1.77E-168</b> |
|       | 1 000  | 1.79E-51 | 4.28E-51 | 1.63E-50 | 1.05E-63  | <b>3.57E-115</b> | <b>1.95E-114</b> | <b>1.07E-113</b> | <b>1.82E-167</b> |
|       | 10 000 | 2.79E-47 | 1.23E-46 | 6.69E-46 | 1.08E-55  | <b>2.25E-113</b> | <b>1.23E-109</b> | <b>4.09E-112</b> | <b>4.09E-181</b> |
| $f_4$ | 100    | 1.74E-48 | 6.79E-48 | 3.67E-47 | 6.97E-57  | <b>2.97E-107</b> | <b>1.62E-106</b> | <b>8.91E-106</b> | <b>4.10E-160</b> |
|       | 500    | 7.77E-50 | 1.97E-49 | 8.23E-49 | 2.83E-56  | <b>1.67E-103</b> | <b>9.16E-103</b> | <b>5.02E-102</b> | <b>1.33E-166</b> |
|       | 1 000  | 7.42E-47 | 3.69E-46 | 2.02E-45 | 8.15E-58  | <b>6.58E-113</b> | <b>3.60E-112</b> | <b>1.97E-111</b> | <b>2.30E-177</b> |
|       | 10 000 | 3.49E-49 | 1.42E-48 | 7.57E-48 | 7.80E-56  | <b>5.20E-105</b> | <b>2.85E-104</b> | <b>1.56E-103</b> | <b>1.73E-167</b> |
| $f_5$ | 100    | 0        | 0        | 0        | 0         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|       | 500    | 0        | 0        | 0        | 0         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|       | 1 000  | 0        | 0        | 0        | 0         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
|       | 100 00 | 0        | 0        | 0        | 0         | <b>0</b>         | <b>0</b>         | <b>0</b>         | <b>0</b>         |
| $f_6$ | 100    | 1.85E-47 | 9.17E-47 | 5.03E-46 | 6.38E-60  | <b>5.36E-115</b> | <b>2.93E-114</b> | <b>1.61E-113</b> | <b>1.36E-161</b> |
|       | 500    | 3.92E-50 | 1.21E-49 | 5.56E-49 | 2.14E-58  | <b>3.68E-118</b> | <b>2.01E-117</b> | <b>1.11E-116</b> | <b>2.25E-169</b> |
|       | 1 000  | 1.92E-04 | 1.05E-03 | 5.76E-03 | 5.63E-59  | <b>2.68E-121</b> | <b>1.47E-120</b> | <b>8.05E-120</b> | <b>6.14E-172</b> |
|       | 10 000 | 2.34E-06 | 1.28E-05 | 7.04E-05 | 1.59E-60  | <b>6.24E-122</b> | <b>3.41E-121</b> | <b>1.86E-120</b> | <b>1.50E-169</b> |
| $f_7$ | 100    | 3.44E-48 | 1.82E-47 | 9.98E-47 | 7.33E-58  | <b>3.88E-120</b> | <b>2.12E-119</b> | <b>1.16E-118</b> | <b>0</b>         |
|       | 500    | 1.81E-48 | 5.73E-48 | 2.88E-47 | 1.65E-55  | <b>1.45E-113</b> | <b>6.11E-113</b> | <b>3.16E-112</b> | <b>0</b>         |
|       | 1 000  | 4.06E-47 | 1.35E-46 | 5.68E-46 | 1.22E-58  | <b>1.15E-125</b> | <b>6.30E-125</b> | <b>3.45E-124</b> | <b>0</b>         |
|       | 10 000 | 6.92E-48 | 3.25E-47 | 1.78E-46 | 8.23E-55  | <b>5.96E-113</b> | <b>3.26E-112</b> | <b>1.78E-111</b> | <b>0</b>         |

由表 3 结果可知,改进的 MHHO 算法寻优精度的四项指标值均优于基本 HHO 算法,在求解高维函数上表现良好.对于函数  $f_2$ 、 $f_4$ 、 $f_6$ ,MHHO 算法的寻优精度接近低维函数的寻优效果,最优值比 HHO 算法提升了上百个数量级;对于函数  $f_1$ 、 $f_5$ 、 $f_7$ ,MHHO 算法在各个维度上最优值均为 0,说明改进的算法在处理复杂的大规模问题上鲁棒性较强,对高维函数仍保持较好适用性.

选取与文献[12]改进的鲸鱼优化算法(Improved Whale Optimization Algorithm,IWOA)共同使用的测试函数  $f_1$ 、 $f_2$ 、 $f_5$ 、 $f_6$ ,在 1000 维上对比高维测试结果,见表 4.对于函数  $f_1$ ,MHHO 算法的平均值和标准差均为 0,说明改进策略具有较强的稳定性;对于函数  $f_2$  和  $f_6$ ,MHHO 算法的寻优精度比 IWOA 算法高近 50 个数量级,说明改进算法具有较高的寻优能力.综上,改进的 MHHO 算法

在处理高维函数上具有较强的鲁棒性和更好的寻优性能,因此,其可以有效处理现实生活中复杂的高维问题.

表 4 IWOA 与 MHHO 高维对比结果

| 初始化参数相同: $N=30,T=500,D=1000$ |           |           |                  |                  |
|------------------------------|-----------|-----------|------------------|------------------|
| 函数名                          | IWOA      |           | MHHO             |                  |
|                              | 平均值       | 标准差       | 平均值              | 标准差              |
| $f_1$                        | 1.86E-111 | 9.00E-112 | <b>0</b>         | <b>0</b>         |
| $f_2$                        | 3.00E-65  | 3.37E-65  | <b>3.57E-115</b> | <b>1.95E-114</b> |
| $f_5$                        | 0         | 0         | <b>0</b>         | <b>0</b>         |
| $f_6$                        | 1.02E-66  | 1.63E-66  | <b>8.05E-120</b> | <b>6.14E-172</b> |

5 结束语

哈里斯鹰优化算法是近年来提出的一种新颖群智能算法,其寻优性能较好.为了使算法性能更加优秀,论文提出了多策略改进的哈里斯鹰优化算法.在

探索阶段,引入柯西分布函数更新全局位置,增加了哈里斯鹰种群的多样性;在过渡阶段,利用随机收缩指数函数将能量方程非线性化,更适应于表达猎物的能量变化,有效实现全局搜索与局部搜索的过渡;在开采阶段,引入自适应惯性权重因子,提高了算法局部开采能力。通过单峰、多峰多个测试函数进行验证,研究结果表明融合三种改进策略较好地提升了算法的寻优能力和鲁棒性。今后研究方向主要是进一步提高 MHHO 寻优速度和拓展 MHHO 算法的实际应用领域。

## 参考文献:

- [1] XU Y T, CHEN H L, HEIDARI A A, et al. An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks[J]. *Expert Systems with Applications*, 2019(129): 135-155. DOI: 10.1016/j.eswa.2019.03.043.
- [2] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications [J]. *Future Generation Computer Systems*, 2019(97): 849-872. DOI: 10.1016/j.future.2019.02.028.
- [3] CHEN H L, JIAO S, WANG M J, et al. Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts[J]. *Journal of Cleaner Production*, 2020(244): 118778. DOI: 10.1016/j.jclepro.2019.118778.
- [4] 赵世杰, 高雷阜, 于冬梅, 等. 融合能量周期性递减与牛顿局部增强的改进 HHO 算法[J]. *控制与决策*, 2021, 36(3): 629-636. DOI: 10.13195/j.kzyjc.2019.0810.  
ZHAO S J, GAO L F, YU D M, et al. Improved Harris hawks optimization coupling energy cycle decline mechanism and Newton local enhancement strategy[J]. *Control and Decision*, 2021, 36(3): 629-636. DOI: 10.13195/j.kzyjc.2019.0810.
- [5] GUPTA S, DEEP K, HEIDARI A A, et al. Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis[J]. *Expert Systems with Applications*, 2020, 158: 113510. DOI: 10.1016/j.eswa.2020.113510.
- [6] KAMBOJ V K, NANDI A, BHADORIA A, et al. An intensify Harris Hawks optimizer for numerical and engineering optimization problems [J]. *Applied Soft Computing*, 2020(89): 106018. DOI: 10.1016/j.asoc.2019.106018.
- [7] 高文欣, 刘升, 肖子雅, 等. 柯西变异和自适应权重优化的蝴蝶算法[J]. *计算机工程与应用*, 2020, 56(15): 43-50. DOI: 10.3778/j.issn.1002-8331.1907-0048.
- GAO W X, LIU S, XIAO Z Y, et al. Butterfly optimization algorithm based on Cauchy variation and adaptive weight[J]. *Computer Engineering and Applications*, 2020, 56(15): 43-50. DOI: 10.3778/j.issn.1002-8331.1907-0048.
- [8] MOUSTAFA M, MOHD M H, ISMAIL A I, et al. Dynamical analysis of a fractional-order Rosenzweig-MacArthur model incorporating a prey refuge [J]. *Chaos, Solitons & Fractals*, 2018(109): 1-13. DOI: 10.1016/j.chaos.2018.02.008.
- [9] 郭振洲, 王平, 马云峰, 等. 基于自适应权重和柯西变异的鲸鱼优化算法[J]. *微电子学与计算机*, 2017, 34(9): 20-25. DOI: 10.19304/j.cnki.issn1000-7180.2017.09.005.  
GUO Z Z, WANG P, MA Y F, et al. Whale optimization algorithm based on adaptive weight and Cauchy mutation[J]. *Microelectronics & Computer*, 2017, 34(9): 20-25. DOI: 10.19304/j.cnki.issn1000-7180.2017.09.005.
- [10] ARORA S, SINGH S. Butterfly optimization algorithm: a novel approach for global optimization[J]. *Soft Computing*, 2019, 23(3): 715-734. DOI: 10.1007/s00500-018-3102-4.
- [11] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. *Advances in Engineering Software*, 2016, 95: 51-67. DOI: 10.1016/j.advengsoft.2016.01.008.
- [12] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法[J]. *系统工程理论与实践*, 2017, 37(11): 2983-2994.  
LONG W, CAI S H, JIAO J J, et al. Improved whale optimization algorithm for solving large-scale optimization problems [J]. *Theory and practice of systems engineering*, 2017, 37(11): 2983-2994.

## 作者简介:

郭雨鑫 女(1996-), 硕士研究生. 研究方向为智能优化、技术经济管理。

刘升(通讯作者) 男,(1966-), 博士, 教授, 研究生导师. 研究方向为智能算法、机器学习。

E-mail: ls6601@sina.com.

高文欣 女,(1995-), 硕士研究生. 研究方向为群智能计算、大数据处理与分析。

张磊 男,(1996-), 硕士研究生. 研究方向为智能计算、大数据处理与分析。