

邻域搜索和改进莱维因子的人工蜂鸟优化算法

何永康, 李旭芳

上海工程技术大学管理学院, 上海

收稿日期: 2023年10月4日; 录用日期: 2023年10月28日; 发布日期: 2024年3月4日

摘要

针对人工蜂鸟算法(AHA)寻优精度较低、易陷入局部最优等问题, 提出新的基于自适应距离围猎和改进莱维因子的人工蜂鸟优化算法(ALAHA)。首先, 在AHA引导觅食、领地觅食阶段引入改进的莱维飞行作为自适应权重因子调节搜索步长, 提高种群全局搜索能力; 然后, 根据种群收敛情况, 在蜂鸟个体周围进行自适应距离围猎搜索, 提高算法搜索精度。本文选取了23个基准测试函数对算法进行了实验, 并与其他算法进行了比较, 以不同角度对于算法的性能进行测试, 并使用Wilcoxon秩和检验来证明算法的性能, 结果表明了ALAHA算法改进在寻优能力、稳定性和鲁棒性等方面有所提升。

关键词

邻域搜索, 莱维飞行, 人工蜂鸟算法

Neighborhood Search and Artificial Hummingbird Optimization Algorithms with Improved Lévy Factors

Yongkang He, Xufang Li

School of Management, Shanghai University of Engineering Science, Shanghai

Received: Oct. 4th, 2023; accepted: Oct. 28th, 2023; published: Mar. 4th, 2024

Abstract

Aiming at the problems of low search accuracy and easy to fall into local optimization of artificial hummingbird algorithm (AHA), a new artificial hummingbird optimization algorithm (ALAHA) based on adaptive distance rounding and improved Lévy factor is proposed. First, the improved Lévy flight is introduced as an adaptive weighting factor to regulate the search step in the AHA guided foraging and territorial foraging phases, which improves the global search ability of the

population; then, according to the convergence of the population, adaptive distance hunting search is performed around the individual hummingbird to improve the algorithm's search accuracy. In this paper, 23 benchmark test functions were selected to experiment the algorithm and compared with other algorithms to test the performance of the algorithm from different perspectives, and the Wilcoxon rank-sum test was used to prove the performance of the algorithm, and the results showed that the ALAHA algorithm improved in terms of optimization ability, stability and robustness.

Keywords

Neighborhood Search, Lévy Flight, Artificial Hummingbird Algorithm

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

面对过往几十年不断复杂的优化问题, 学者们逐渐发展出了多样化的优化方法。当前优化算法主要是基于自然界存在的自然现象, 通过观察从中获取不同的灵感, 主要包括生物进化、生物的群居行为、物理或化学现象, 以及与人类相关的行为和意识形态。其中具有代表性的主要有遗传算法(GA) [1]、粒子群算法(PSO) [2]、蚁群算法(ACO) [3]、莱维飞行算法(LFD) [4]。随着学者对优化算法的不断研究, 一些新的备受关注的算法不断被开发, 例如鲸鱼优化算法(WOA) [5]、灰狼优化算法(GWO) [6]、樽海鞘群算法(SSA) [7]、哈里斯鹰优化算法(HHO) [8]、基于教学的优化算法(TLBO) [9]、黏菌算法(SMA) [10]、蝠鲼觅食优化算法(MRFO) [11]、平衡优化器(EO) [12]等。在各种智能计算问题上, 群智能算法具有算法结构简单、收敛速度快、全局收敛性好等优点, 这些新的智能算法的改进版本被开发用于解决各种工程问题。但是, 在求解单目标模型上, 智能算法改进仍然面临保持多样性和提高收敛精度的矛盾。一方面, 智能算法需要尽可能遍历整个搜索空间来寻找全局最优解; 另一方面, 优化问题存在一定的搜索精度需求, 需要智能算法在计算时提高最优解的收敛性能。

赵卫国等人在 2021 年提出了一种新型智能优化算法人工蜂鸟算法[13] (AHA), 较好的平衡算法全局搜索能力和搜索精度, 该算法通过引导觅食、领地觅食和迁徙觅食三种觅食策略来引导算法迭代, 相较于其他算法和人工蜂群算法, 其不同在于提出了访问表策略, 该策略较好得平衡了全局搜索能力和搜索精度。AHA 与其他群智能优化算法相比具有参数少, 寻优能力强等优点, 因此各种人工蜂鸟改进算法也已被广泛应用于各类工程应用领域, 例如汽车电池[14]、无人机定位[15]、光伏领域[16]和医学图像分类[17]等。然而, AHA 同样存在着收敛速度较慢, 易陷入局部极小值的问题, Yildiz, BS [18]通过将人工蜂鸟算法和模拟退火问题相结合, 提出了混合元启发算法 AHA; 赵卫国[19]采用外部存档来保存帕累托最优解, 并开发了一种基于动态消除的拥挤距离(DECDD)方法, 开发了一种多目标 AHA, 提高了算法的收敛性; Ali, MAS [20]使用随机基于对立的学习(ROBL)和基于对立的学习(OBL)来增强利用阶段, 防止局部最优并加速收敛, 并运用到基于特征选择的废物分类问题。

全局搜索能力和搜索精度是相互制约的, 提高搜索精度的同时很可能会降低全局搜索能力, 反之亦然, 所以, 如果过于重视算法的搜索精度, 这会使算法充分达到当前区域最优值, 但是容易陷入局部最优解。反之算法如果过于重视算法的全局搜索能力, 这能让算法找到最优解所在的区域, 但是不能保证

算法有足够的搜索精度以找到全局最优解。为了平衡算法全局搜索能力和搜索精度, 针对人工蜂鸟算法中存在的寻优精度不高、收敛速度慢的缺陷, 本文采取两大策略改进, 具体工作如下:

(1) 提出自适应距离围猎策略, 来提高算法局部搜索能力。提高搜索精度的一种常用方法是局部搜索策略。其主要思想是在当前个体周围基于概率模型生成新个体。本文受鲸鱼优化算法的围捕猎物策略启发, 在蜂鸟个体觅食的食物源周围生成虚拟个体, 并且沿着螺旋形的路径觅食, 寻找更优的食物源。但是, 如果在每个个体周围都进行搜索的话, 会大大提高算法计算量降低搜索时间。所以概率模型的参数如何设置是一个难题。文献[21]提出的算法使用了固定的标准差, 但是一个固定的取值鲁棒性较差。本文借助一种根据每次迭代的种群适应度的方差来判断是否在目标周围进行围猎的策略。较好的降低了算法的计算数量并充分提高了收敛效率和搜索精度。

(2) 引入使用非线性收敛控制的莱维飞行因子, 用于替代原算法步长因子。由于在加入自适应距离围猎策略后, 算法收敛速度过快, 容易陷入局部最优解, 而原算法步长因子取值服从标准正态分布, 取值范围小, 不能满足改进算法全局搜索需求。所以本文引进莱维飞行因子来替代原算法步长因子, 但是莱维飞行因子相比于原算法步长因子前期搜索能力提高较小, 后期精确搜索能力不足, 本文因此引入非线性收敛因子来控制莱维飞行因子取值大小, 在搜索前期加大莱维飞行因子取值振幅, 从而进一步提高算法全局搜索能力, 在迭代后期缩小莱维飞行因子取值, 提高算法收敛精度, 并加快收敛速度。

2. 人工蜂鸟算法

人工蜂鸟算法(Theartificial hummingbird optimization algorithm, AHA)是赵卫国等人在2022年受蜂鸟觅食行为启发而提出的一种新型基于群的元启发式算法, 该算法通过引导觅食、领地觅食和迁徙觅食三种觅食策略来引导算法迭代, 相较于其他算法和人工蜂群算法, 其不同在于提出了访问表策略, 该策略较好得平衡了全局搜索能力和搜索精度。AHA 与其他群智能优化算法相比具有参数少, 寻优能力强等优点, 因此各种人工蜂鸟改进算法也已被广泛应用于各类工程应用领域。其仿生学原理如下。

蜂鸟通过三种觅食行为: 引导觅食、领地觅食和迁徙觅食, 对每个食物源进行访问, 食物源的花蜜填充率用函数的适应度值表示。同时, 在觅食行为中模拟了轴向飞行、对角飞行和全向飞行三种飞行技能。

访问表: 访问表是存储食物源访问信息的重要组成部分, 它跟踪每只蜂鸟对每个食物源的访问级别。访问表中的值越大, 表明蜂鸟不访问该食物源的时间越长, 这意味着该食物源为该蜂鸟积累了更多的花蜜量, 将促进访问该食物源的优先级。种群中的所有蜂鸟都可以在引导觅食阶段找到自己觅食的目标食物源。同时, 在每个不同的觅食阶段, 访问表根据候选解决方案的质量进行更新。

在觅食过程中, 通过引入方向切换向量, 充分利用了全向飞行、对角飞行和轴向飞行三种飞行技能, 使得蜂鸟就可以飞向目标食物来源觅食。

其中轴向飞行的定义如下:

$$D^{(i)} = \begin{cases} 1 & \text{if } i = \text{randi}([1, d]) \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, d \quad (1)$$

对角飞行的定义如下:

$$D^{(i)} = \begin{cases} 1 & \text{if } i = P(j), j \in [1, k], \\ & P = \text{randp}(k), \\ & k \in [2, [r_1 \cdot (d-2)] + 1] \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, d \quad (2)$$

全向飞行定义如下:

$$D^{(i)} = 1 \quad i=1, \dots, d \quad (3)$$

其中, $\text{randi}([1, d])$ 生成一个从 1 到 d 的随机整数, $\text{randp}(k)$ 创建一个从 1 到 k 的随机整数排列, r_1 是 $(0, 1)$ 中的一个随机数。

模拟引导觅食行为和候选食物源的数学方程如下:

$$v_i(t+1) = x_{i, \text{tar}}(t) + a \cdot D \cdot (x_i(t) - x_{i, \text{tar}}(t)) \quad (4)$$

其中 $v_i(t+1)$ 表示第 $t+1$ 只蜂鸟的位置, $x_i(t)$ 是第 i 个食物源在时间 t 的位置, $x_{i, \text{tar}}(t)$ 是第 i 只蜂鸟打算访问的目标食物源的位置, a 是一个引导因子, 它受制于正态分布 $N(0, 1)$, 平均值 = 0, 标准偏差 = 1。第 1 个食物源的位置更新如下:

$$x_i(t+1) = \begin{cases} x_i(t) & f(x_i(t)) \leq f(v_i(t+1)) \\ v_i(t+1) & f(x_i(t)) > f(v_i(t+1)) \end{cases} \quad (5)$$

其中 $f(\cdot)$ 表示函数的适应度值。

模拟领地觅食行为和候选食物源的数学方程如下:

$$v_i(t+1) = x_i(t) + b \cdot D \cdot x_i(t) \quad (6)$$

其中 b 为区域因子, 服从正态分布 $N(0, 1)$ 。

模拟迁移觅食行为和候选食物源的数学方程如下:

$$x_{\text{wor}}(t+1) = \text{Low} + r \cdot (\text{Up} - \text{Low}) \quad (7)$$

其中 x_{wor} 是种群中花蜜填充率最差的食物来源, M 是迁移系数, 蜂鸟可能在 $2n$ 次迭代后访问与目标源相同的食物源。

3. 优化人工蜂鸟优化算法(ALAHA)

3.1. 领域搜索策略

邻域搜索策略的表达式如下:

$$\begin{cases} X_i(t) = D \cdot e^{bl} \cos(2\pi l) + X_i^*(t) \\ D = |C \cdot X_i^*(t) - X_i(t-1)| \end{cases} \quad (8)$$

其中 M 表示第 i 只蜂鸟到花蜜(当前最优解)的距离, l 为是 $[-1, 1]$ 中的随机数。

该策略通过蜂鸟在猎物周围游动, 形成一个缩小的圆圈来搜索猎物。这种螺旋环缩模型可以有效搜索猎物周边情况, 提高猎物周边搜索效率。但是, 该策略的环缩速度会影响算法搜索精度, 螺旋模型大小会影响算法全局搜索能力, 为了较好解决这一不足, 本文提出一种自适应距离围猎策略。围猎策略借助每次迭代的种群适应度的方差来判断是否在目标周围进行围猎搜索。该策略通过摒弃环缩机制来减少对算法全局搜索能力的影响, 直接用较小的螺旋模型在目标周围精确搜索, 提高算法收敛精度, 并且用目标适应度的方差来减少算法计算量, 以较少计算增长量大幅提高算法搜索效率。蜂鸟个体位置更新如图 1 所示。

标准差设置是局部搜索策略的常用方式, 如果对每个迭代个体都使用近距离围猎策略, 一方面会大幅提高算法搜索时间, 另一方面过高的收敛速度会降低算法全局搜索能力, 取值鲁棒性较差。所以本文借助一种根据每次迭代的种群适应度的方差来判断是否在目标周围进行围猎觅食。其数学公式如下:

$$\theta = \sum_i^n \left(\frac{X_i(t) - X_{ave}}{X_{up} - X_{low}} \right)^2 \quad (9)$$

其中 θ 为方差, n 为蜂鸟的种群数量, $X_i(i)$ 为第 i 只蜂鸟种群的适应度, X_{ave} 为蜂鸟种群的平均适应度, X_{up} 为种群中的最优适应度, X_{low} 为种群中的最差适应度。

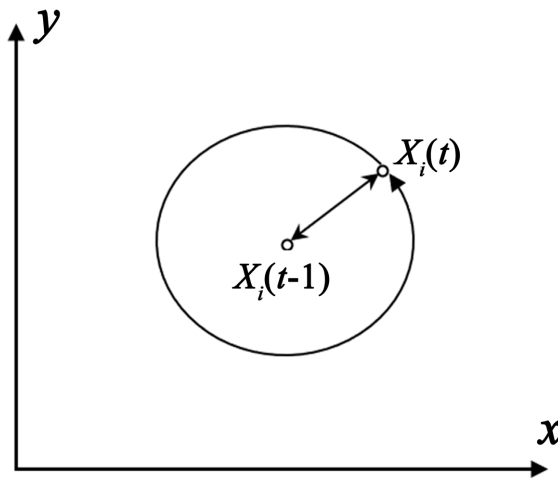


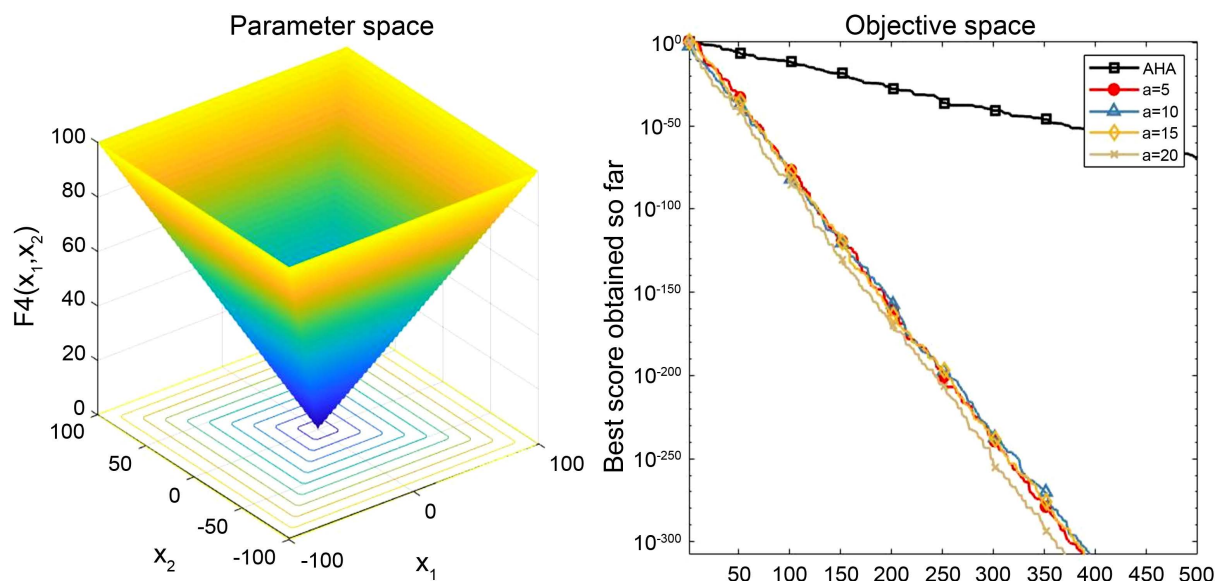
Figure 1. Target individual location update
图 1. 目标个体位置更新

当 $\theta < a$ 时, 为了提高搜索精度, 对蜂鸟种群位置更新如下:

$$X_i(t) = X_i(t-1) \cdot \cos(2\pi\mu) + X_i(t-1) \quad (10)$$

其中 $X_i(t-1)$ 表示第 i 只蜂鸟在 $t-1$ 时刻的种群适应度, μ 为在 $(-1, 1)$ 区间内服从正态分布的随机数。

为了测试邻域搜索策略的有效性和方差取值大小, 以基本 AHA 算法为基础, 分别在经典测试函数单峰函数 Schwefel's 和多峰函数 Quartic 上测试, 适应值如图 2 所示。



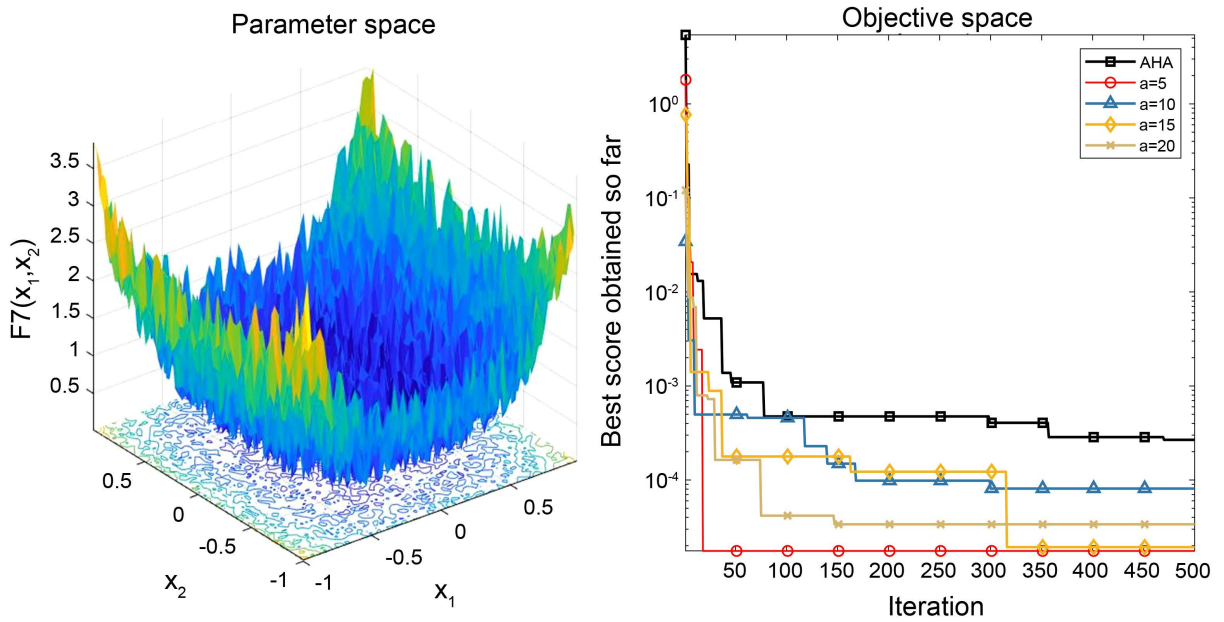


Figure 2. Comparison of policy adaptation values
图 2. 策略适应度对比

从图 2 中可以看出, 在测试函数中, 采取了邻域搜索策略的 AHA 算法比基本 AHA 算法有更好的收敛速度和收敛精度。对于方差 a 的选取, 可以从图 2 中看出, a 取值越大, 需要计算的蜂鸟个体越多, 计算量越大, 计算时间越长, 收敛速度越快。在单峰函数上, a 取值对算法的影响有其上限, 过多提高 a 的取值并不会明显提高算法收敛性, a 取值敏感性不高, 但是太低的取值又会使策略发挥不了作用; 在多峰函数上, a 取值越大全局搜索能力不断减弱, 更加容易陷入局部最优, 可见 a 取值不应该太高。综合考虑本文认为 a 取值 5 比较合适。

3.2. 改进莱维飞行因子

由于加入了邻域搜索策略后原算法收敛性大幅加强, 导致算法前期容易快速收敛到局部最优点, 不容易充分探索搜索空间, 在某些复杂问题上降低了算法全局搜索能力。由式(4)和式(6)可知, 标准 AHA 算法种群位置更新步长由因子 b 控制, 在搜索过程中, 由于步长限制, 在加入围猎策略后全局搜索能力不足以适应新算法的需求, 导致算法鲁棒性下降, 搜索后期容易陷入局部最优, 种群搜索陷入停滞。

文献[22]证明了莱维飞行策略其搜索范围远近交替的随机飞行的特征, 在与麻雀搜索算法相结合后, 可以提高算法全局搜索能力。本文通过提取莱维飞行因子, 结合算法实际状况提出针对性改进方法, 将其与引导觅食和领地觅食策略相结合, 更新种群个体位置公式如式(11)、式(12)所示。

$$v_i(t+1) = x_{i,tar}(t) + Lv \cdot a \cdot D \cdot (x_i(t) - x_{i,tar}(t)) \tag{11}$$

$$v_i(t+1) = x_i(t) + Lv \cdot b \cdot D \cdot x_i(t) \tag{12}$$

其中 Lv 为飞行步长, $v_i(t+1)$ 表示第 $t+1$ 个蜂鸟的位置, $x_i(t)$ 是第 i 个食物源在时间 t 的位置, $x_{i,tar}(t)$ 是第 i 只蜂鸟打算访问的目标食物源的位置。

标准莱维飞行因子 Lv 计算公式如下:

$$Lv = \frac{u}{|v|^{1/\beta}} \tag{13}$$

u 、 v 服从正态分布, $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \sigma_v^2)$ 。

σ_u 、 σ_v 定义如下:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\beta\pi/2)}{\Gamma((1+\beta)/2)2^{(\beta-1)/2}\beta} \right\}^{1/\beta}, \quad \sigma_v = 1 \quad (14)$$

其中 β 通常取值于 $[0, 2]$ 之间, 这里取值 1.5。

将莱维飞行因子引入代替原步长因子 b 可以增强算法的随机性, 相比于原步长因子 b , 莱维飞行因子取值震荡性更大, 更有利于算法随机搜索, 可以提高算法前中期的全局搜索能力。但是从图中也可以看出, 在搜索前期莱维飞行因子相对于原步长因子 b 取值相差不大, 全局能力有继续加强的空间, 而在迭代后期, 过高的全局能力不利于算法收敛, 需要降低莱维飞行因子取值震幅。可见, 经典莱维飞行因子不符合当前算法的寻优过程。

为此, 本文提出一种新的非线性收敛控制的改进莱维飞行因子。本文通过在公式(14)中加入非线性收敛因子来控制 u 的大小, 从而控制莱维飞行因子的振幅大小。非线性收敛因子公式如下:

$$\omega = \cos\left(\frac{t\pi}{t_{\max}}\right) + 1.5 \quad (15)$$

然后将非线性收敛因子加入公式(17), 组成的新公式如下:

$$\sigma_u = \omega \left\{ \frac{\Gamma(1+\beta)\sin(\beta\pi/2)}{\Gamma((1+\beta)/2)2^{(\beta-1)/2}\beta} \right\}^{1/\beta} \quad (16)$$

改进前后的非线性收敛因子 ω 变化图如图 3 所示。 ω 将从初期的 2.5 变化到后期的 0.5, 可见在迭代初期改进莱维飞行因子具有更大的振幅空间, 有利于算法更充分地探索搜索空间, 并且非线性的收敛因子使改进莱维飞行因子在迭代初期变化相对平缓, 后期较小的 ω 也使得改进莱维飞行因子振幅更小, 局部开采能力逐渐增强。

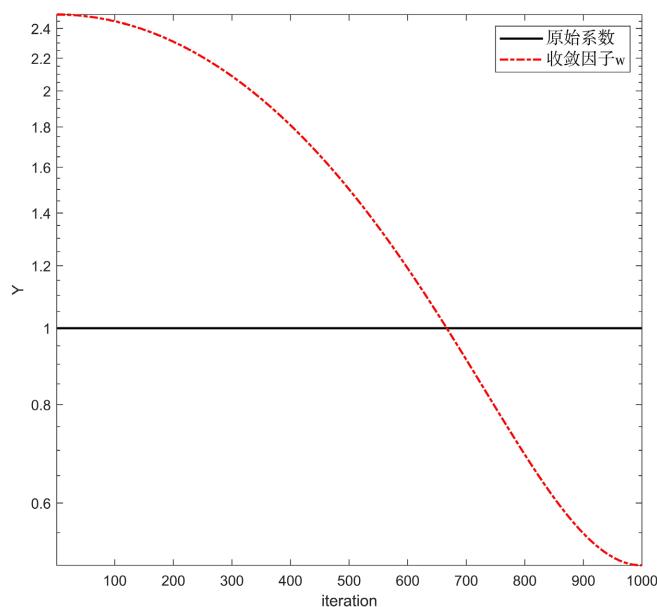


Figure 3. Improvement of the value comparison between Lévy factor and cause factor

图 3. 改进莱维因子与原因子取值对比

为了进一步验证策略的全局能力, 把改进莱维因子的 AHA 和标准 AHA 算法进行比较, 测试函数为 Schwefel's, 种群数量为 100, 维数为 30, 分别选取算法迭代第 100 代和第 300 代散点分布图进行对比, 对比图如图 4 所示。

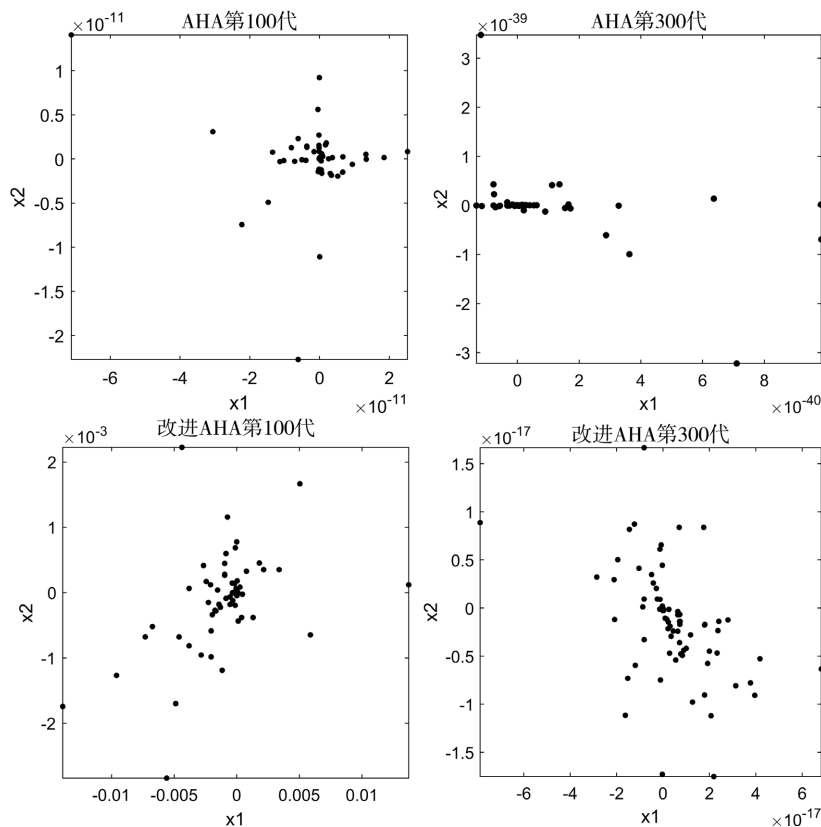


Figure 4. Strategy scatter distribution comparison
图 4. 策略散点分布对比

从图中可以看出, 在单纯使用改进莱维因子策略的情况下, 算法迭代 100 次和 300 次改进 AHA 算法精度分别为 10^{-2} 和 10^{-17} , 而标准 AHA 算法精度在 10^{-11} 和 10^{-40} , 远低于改进 AHA 算法, 可见改进 AHA 算法在迭代过程可以更好地遍历搜索空间, 拥有更好全局优化能力。

3.3. ALAHA 算法的步骤

ALAHA 算法流程图如图 5 所示, ALAHA 算法伪代码和如下表 1:

Table 1. ALAHA algorithm pseudocode
表 1. ALAHA 算法伪代码

Begin
1: 初始化种群, 初始化访问表, 计算蜂鸟个体适应度, 设置初始参数: 种群规模 n , 最大迭代数 $MaxIter$,
2: For $It = 1:MaxIter$ do
3: For $i = 1:n$
4: 根据式(1) (2) (3)定义飞行矢量
5: 根据式(13) (15) (16)计算改进莱维飞行因子

续表

- 6: If $\text{rand} < 0.5$ do
 7: 根据式(11)更新蜂鸟觅食位置
 8: 更新访问表
 9: Else
 10: 根据式(12)更新蜂鸟觅食位置
 11: 更新访问表
 12: End
 13: If $\theta < 5$
 14: 根据式(10)更新蜂鸟位置
 15: End
 16: End for
 17: End while
 18: 对比种群个体, $x_{i,\text{star}}(t)$ 为当前全局最优解
 19: End

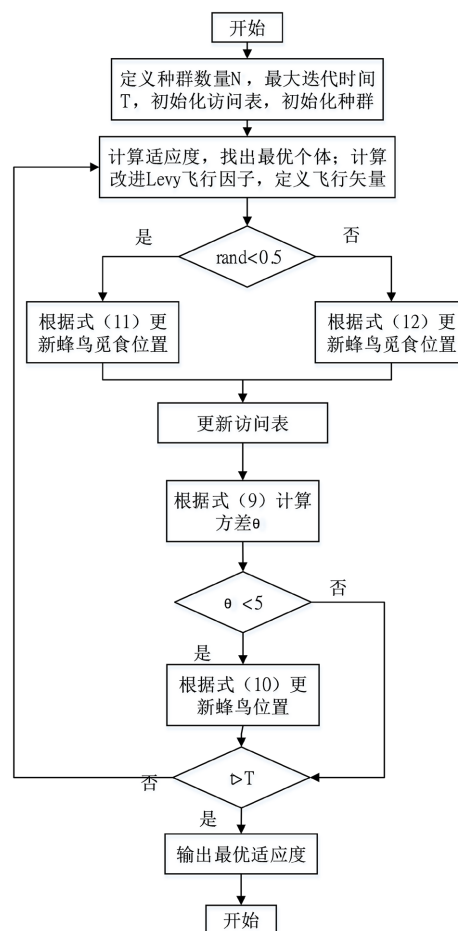


Figure 5. ALAHA algorithm flowchart
 图 5. ALAHA 算法流程图

3.4. 时间复杂度计算

AHA 首先初始化一组随机解决方案和一个访问表。在每次迭代中, 有 50% 的概率进行引导觅食或领地觅食。引导觅食使蜂鸟能够朝着由访问表和花蜜再填充率确定的各自目标食物来源移动。领地觅食迫使蜂鸟扰乱自己的当地社区。每 $2n$ 次迭代, 进行一次迁移觅食。这三种觅食行为都使用了三种飞行技能, 包括全向、对角和轴向飞行。所有操作和计算都是交互执行的, 直到达到停止标准。最终, 具有最佳花蜜再填充率的食物源作为全局最优值的近似值返回。

AHA 算法的计算复杂度与初始化、适应度评估(c)、蜂鸟位置更新、蜂鸟种群大小(N)、最大迭代次数(T)、变量维数(d)。AHA 的整体计算复杂度可以表示为:

$$\begin{aligned} O(\text{AHA}) &= O\left(1 + nd + Tcn + \frac{1}{2}Tnd + \frac{1}{2}Tnd + \frac{T}{2n}nd\right) \\ &\cong O\left(Tcn + Tnd + \frac{Td}{2}\right) \end{aligned} \quad (17)$$

改进算法 ALAHA 相比原算法 AHA 在每次迭代过程中都需要计算一次莱维飞行因子, 在引导觅食和领地觅食过程中需要根据收敛情况进行一次近距离围猎, 并且蜂鸟个体进行位置更新时判断是否接受差解。所以改进后算法所需的时间复杂度为:

$$\begin{aligned} O(\text{ALAHA}) &= O\left(1 + nd + Tcn + \frac{1}{2}Tnd + \frac{1}{2}Tnd + \frac{1}{2}Tnd + \frac{1}{2}Tnd + \frac{1}{2}Tnd + \frac{1}{2}Tnd\right) \\ &\cong O(Tcn + 3Tnd) \end{aligned} \quad (18)$$

4. 实验仿真和分析

本文实验的运行环境为 64 位 Windows 10 操作系统, 处理器是 Intel(R) Core(TM)i7-8550U, 使用的软件是 MATLAB R2020b。本文使用 23 个标准测试函数集对 ALAHA 进行试验测试。

4.1. 对比算法与参数设置

为了研究改进算法的有效性, 文章选取改进的人工蜂鸟算法 ALAHA 与人工蜂鸟算法(AHA)、莱维飞行算法(LFD)、鲸鱼优化算法(WOA)、灰狼优化算法(GWO)、樽海鞘群算法(SSA)和哈里斯鹰优化算法(HHO)进行对比。这些算法已被证实拥有良好的搜索能力。为了使算法验证过程公平公正, 文章将算法的种群规模全部设置成 50, 空间维度设置成 30, 最大迭代次数为 500 次, 每个算法独立运行 30 次, 算法相关参数设置如表 2 所示。将平均值、标准差与 Wilcoxon 秩和检验作为算法评价标准。

Table 2. Parameter settings of each algorithm

表 2. 各算法参数设置

算法	参数设置
MAHA	$\theta = 4, M = 2n$
AHA	$M = 2n$
PSO	$c_1 = 2, c_2 = 2, v_{max} = 6, w_{max} = 9, w_{min} = 0.2$
HHO	$s = 0.01, \beta = 1.5$

续表

WOA	$a_1 = [2, 0], a_2 = [-2, 1], b = 1$
GWO	$a = [2, 0]$
SSA	$c_1 \in [0, 1], c_2 \in [0, 1]$

为了进一步验证 MAHA 的搜索性能和鲁棒性优于标准 AHA 和其他优化算法, 文章对多个不同特点的测试函数进行测试对比, 测试函数的函数编号、函数名称、函数维度、搜索范围和理论最优值如表 3 所示。

Table 3. Test function
表 3. 测试函数

编号	函数名称	维数	范围	最优值
F1	Sphere Function	30	[-100, 100]	0
F2	Schwefel's problem 2.22	30	[-10, 10]	0
F3	Schwefel's problem 1.2	30	[-100, 100]	0
F4	Schwefel's problem 2.21	30	[-100, 100]	0
F5	Generalized Rosen-Brock's Function	30	[-30, 30]	0
F6	Step function	30	[-100, 100]	0
F7	Quartic Function	30	[-1.28, 1.28]	0
F8	Generalized Schwefel's problem 2.26	30	[-500, 500]	-12569
F9	Generalized Rastrigin's Function	30	[-5.12, 5.12]	0
F10	Ackley's Function	30	[-32, 32]	0
F11	Generalized Griewank Function	30	[-600, 600]	0
F12	Generalized Penalized Function	30	[-50, 50]	0
F13	Generalized Penalized Function	30	[-65, 65]	0
F14	Shekel's Foxholes Function	2	[-5, 5]	0.998
F15	Kowalik's Function	4	[-5, 5]	0.0003
F16	Six-Hump Camel-Back Function	2	[-5, 5]	-1.0316
F17	Branin Function	2	[0, 1]	0.398
F18	Goldstein-Price Function	2	[-2, 2]	3
F19	Hartmann 3-D Function	3	[1, 3]	-3.86
F20	Hartmann 4-D Function	6	[0, 1]	-3.322
F21	Shekel's Family	4	[0, 10]	-10.15
F22	Shekel's Family	4	[0, 10]	-1040
F23	Shekel's Family	4	[0, 10]	10.54

4.2. 算法测试分析

ALAHA 与其对比算法的平均适应度和标准差见表 4, 对于单模态函数 F1~F7 来说, ALAHA 与自身两个原始算法 AHA 和 LFD 相比较而言, 平均寻优能力获得极大的提高, 在 F1~F4 函数上均达到函数理论最优值, 根据标准差表现, ALAHA 搜索稳定性最好。对于 F5, ALAHA 虽然没有收敛到理论最优值, 但是搜索精度仅次于 HHO, 超过原始算法和其他算法。对于 F6, ALAHA 搜索精度仅次于 SSA 和 HHO, 相比原始算法精度有所提高。在 F7 中, ALAHA 搜索能力和稳定性都是最好的。由此可知, ALAHA 在局部开发阶段引入自适应距离搜索策略与莱维飞行因子能够带来优秀的搜索能力, 提高算法的搜索精度, 并且具有良好的稳定性。

F8~F13 为多模态测试函数, 函数具有多个局部最优解和一个全局最优解, 可以用来验证算法全局搜索能力和跳出局部最优的能力。对于 F8、F13, ALAHA 搜索效果仅次于 HHO, 排名第二, 且相比原始算法 AHA 和 LFD 开发能力更强。对于 F9、F10、F11, ALAHA 达到理论最优值。对于 F12, ALAHA 搜索效果最好, 跳出局部最优能力最强。由此可知, ALAHA 的全局搜索能力和跳出局部最优的能力都高于原始算法 AHA 和 LFD, 只有两个函数略逊于 HHO, 但是总体来看算法效果还是最好的。说明本次改进确实提高了了算法的全局搜索能力以及给予了算法跳出局部最优解的能力。

F14~F23 是固定维度多模态测试函数, 用来实验算法在平衡全局开发和局部探索能力的性能高低。对于 F8、F13, ALAHA 搜索效果仅次于 HHO, 排名第二, 且相比原始算法 AHA 和 LFD 开发能力更强。对于 F9、F10、F11, ALAHA 达到理论最优值。对于 F12, ALAHA 搜索效果最好, 跳出局部最优能力最强。由此可知, ALAHA 的全局搜索能力和跳出局部最优的能力都高于原始算法 AHA 和 LFD, 只有两个函数略逊于 HHO, 但是总体来看算法效果还是最好的。说明本次改进确实提高了了算法的全局搜索能力以及给予了算法跳低。对于 F14~F23, ALAHA 和原算法 AHA 的平均适应度全部能够达到理论最优值或者及其逼近理论最优值, 但是相对于原算法 AHA, ALAHA 的标准差更好, 而相比原始算法 LFD 和其他优化算法, ALAHA 在最优值和标准差上都要更优秀。在固定维度多模态测试函数上, ALAHA 的实验结果都是最优, 根据测试函数本身特性, 证实了 ALAHA 平衡算法收敛和全局搜索能力上具有优秀的性能, 在算法的稳定性上相比原始算法也更强。

Table 4. Test results of benchmark functions of each algorithm

表 4. 算法标准函数测试结果

函数	统计值	ALAHA	AHA	LFD	HHO	WOA	GWO	SSA
F1	平均值	0.000E+00	1.676E-144	1.055E-08	4.727E-101	1.547E-85	2.881E-33	2.134E-08
	标准差	0.000E+00	5.071E-144	6.175E-09	2.313E-100	6.132E-85	5.435E-33	6.787E-09
F2	平均值	0.000E+00	1.116E-76	9.949E-03	1.273E-51	7.080E-54	6.213E-20	1.078E+00
	标准差	0.000E+00	2.640E-76	7.647E-03	6.333E-51	2.775E-53	5.630E-20	1.042E+00
F3	平均值	0.000E+00	4.691E-132	9.642E-05	2.658E-76	3.462E+04	6.380E-08	6.308E+02
	标准差	0.000E+00	1.508E-131	6.503E-05	1.431E-75	1.092E+04	1.297E-07	4.931E+02
F4	平均值	0.000E+00	2.724E-66	3.349E-03	2.679E-52	4.318E+01	2.139E-08	7.435E+00
	标准差	0.000E+00	1.301E-65	2.177E-03	1.056E-51	2.912E+01	1.902E-08	3.058E+00
F5	平均值	2.586E+01	2.631E+01	2.789E+01	5.684E-03	2.743E+01	2.677E+01	1.292E+02
	标准差	2.222E-01	3.804E-01	1.455E-01	8.455E-03	4.267E-01	5.406E-01	2.367E+02

续表

F6	平均值	2.071E-05	1.842E-02	3.643E-03	4.266E-05	5.759E-02	4.373E-01	1.994E-08
	标准差	8.507E-05	6.180E-02	5.515E-03	5.439E-05	5.223E-02	2.944E-01	5.778E-09
F7	平均值	2.243E-05	2.017E-04	9.945E-03	7.056E-05	2.616E-03	1.231E-03	1.024E-01
	标准差	2.339E-05	1.574E-04	5.725E-03	5.203E-05	2.967E-03	6.930E-04	4.000E-02
F8	平均值	-1.231E+04	-1.205E+04	-1.045E+04	-1.257E+04	-1.082E+04	-6.096E+03	-7.563E+03
	标准差	2.540E+02	2.948E+02	9.772E+03	3.953E-01	1.856E+03	7.115E+02	7.364E+02
F9	平均值	0.000E+00	0.000E+00	1.029E+01	0.000E+00	0.000E+00	2.983E+00	4.033E+01
	标准差	0.000E+00	0.000E+00	2.862E+01	0.000E+00	0.000E+00	3.305E+00	1.432E+01
F10	平均值	8.882E-16	8.882E-16	1.391E-04	8.882E-16	4.086E-15	4.293E-14	2.009E+00
	标准差	0.000E+00	0.000E+00	2.359E-05	0.000E+00	2.487E-15	6.090E-15	6.367E-01
F11	平均值	0.000E+00	0.000E+00	7.691E-09	0.000E+00	5.497E-03	3.941E-03	9.844E-03
	标准差	0.000E+00	0.000E+00	2.914E-08	0.000E+00	2.960E-02	9.510E-03	1.380E-02
F12	平均值	1.346E-06	3.074E-05	1.745E-03	2.807E-06	8.842E-03	2.851E-02	4.994E+00
	标准差	2.037E-06	1.756E-05	8.779E-04	3.983E-06	1.305E-02	1.535E-02	2.351E+00
F13	平均值	5.547E-04	1.486E+00	8.747E-01	6.172E-05	2.085E-01	4.158E-01	2.621E+00
	标准差	2.477E-03	6.960E-01	1.234E+00	8.866E-05	1.858E-01	1.942E-01	6.512E+00
F14	平均值	9.980E-01	9.980E-01	1.097E+00	1.230E+00	2.857E+00	3.521E+00	1.097E+00
	标准差	0.000E+00	0.000E+00	3.928E-01	5.546E-01	3.587E+00	3.158E+00	2.982E-01
F15	平均值	3.075E-04	3.075E-04	9.483E-04	3.376E-04	7.158E-04	3.721E-03	1.520E-03
	标准差	2.302E-12	1.735E-11	3.810E-04	2.277E-05	4.483E-04	7.446E-03	3.511E-03
F16	平均值	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00
	标准差	5.889E-16	6.148E-16	3.283E-10	6.069E-11	3.411E-10	1.157E-08	1.377E-14
F17	平均值	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	标准差	0.000E+00	0.000E+00	4.206E-09	3.935E-06	1.300E-06	1.221E-05	1.163E-14
F18	平均值	3.000E+00	3.000E+00	3.026E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00
	标准差	1.112E-15	1.253E-15	2.206E-02	2.007E-08	1.120E-05	1.137E-05	1.889E-13
F19	平均值	-3.863E+00	-3.863E+00	-3.861E+00	-3.861E+00	-3.860E+00	-3.861E+00	-3.863E+00
	标准差	1.665E-15	2.651E-15	1.337E-03	2.247E-03	4.456E-03	2.650E-03	1.160E-13
F20	平均值	-3.318E+00	-3.314E+00	-3.208E+00	-3.169E+00	-3.255E+00	-3.272E+00	-3.231E+00
	标准差	2.134E-02	2.966E-02	9.047E-02	8.207E-02	1.061E-01	6.876E-02	5.519E-02
F21	平均值	-1.015E+01	-1.015E+01	-8.634E+00	-5.559E+00	-8.448E+00	-8.970E+00	-7.063E+00
	标准差	6.339E-15	6.188E-15	2.578E+00	1.515E+00	2.405E+00	2.142E+00	3.396E+00
F22	平均值	-1.040E+01	-1.040E+01	-7.543E+00	-5.435E+00	-8.788E+00	-1.022E+01	-8.856E+00
	标准差	7.252E-16	4.558E-07	3.116E+00	1.304E+00	2.592E+00	9.539E-01	2.860E+00
F23	平均值	-1.054E+01	-1.054E+01	-8.035E+00	-5.306E+00	-8.634E+00	-1.026E+01	-9.766E+00
	标准差	1.123E-15	1.319E-14	3.122E+00	9.657E-01	2.969E+00	1.456E+00	2.313E+00

4.3. 收敛曲线分析

为了更加清晰直观地展现各个算法的全局搜索能力和收敛情况, 本文从 23 个测试函数中挑选了 9 个函数, F1、F3、F7 为单模态函数, 用来检验算法收敛速度和精度; F9、F12、F12 为多模态函数, 可以观察算法的全局搜索能力; F15、F20、F22 为固定维多模态函数, 可以清晰的观察到算法在平衡局部开发能力和全局探索能力上的优劣, 收敛曲线如图 6 所示。在 F1 和 F3 上, ALAHA 收敛性远高于其他算法, 收敛精度最高, AHA 算法在迭代 500 次后还未达到最优值, 而 ALAHA 只需迭代不到 200 次即可达到理论最优值, 收敛速度快。在 F7 中, ALAHA 虽然未达到理论最优值, 但是其全局搜力更强, 跳出局部最优的速度更快, 使得收敛精度高于其他算法。对于函数 F9, 虽然 ALAHA、AHA、HHO 和 WOA 算法都收敛到全局最优值, 但是相比其他算法, ALAHA 只需要 10 次迭代就能找到最优值, 搜索效率高于其他算法。对于 F12 和 F13, ALAHA 收敛速度和收敛精度都更好, 并且相比其他算法具有优秀的跳出局部最优的能力。在固定维多模态函数 F15、F20、F22 中, ALAHA 收度快, 并且在极短的时间内就逼近全局最优解, 证明了算法优秀的全局探索和局部开发能力。

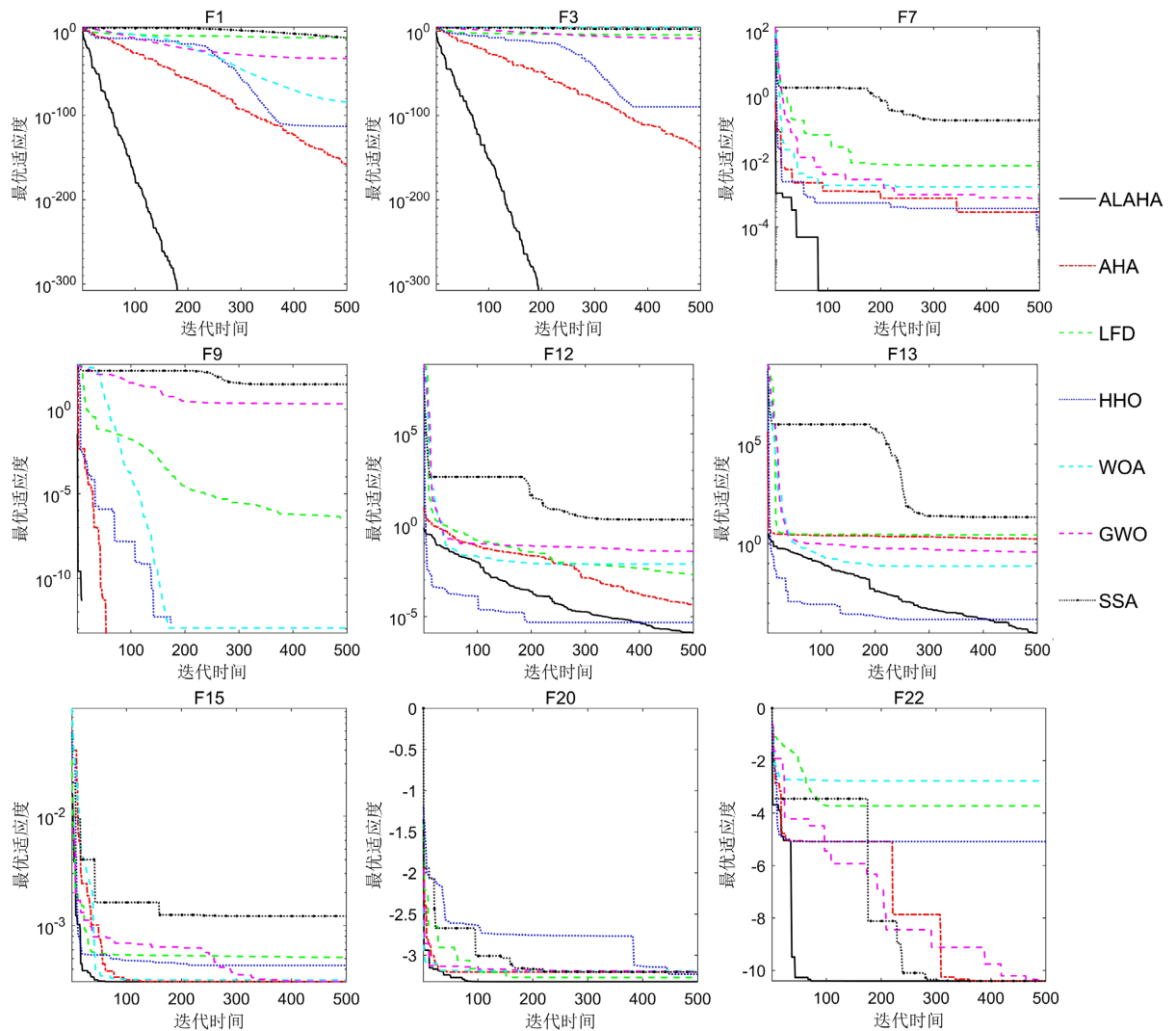


Figure 6. ALAHA convergence curves
图 6. ALAHA 收敛曲线

4.4. Wilcoxon 秩和检验

为了更真实地评估 ALAHA 算法的性能, 本文采用 Wilcoxon 秩和检验来验证实验结果的显著性差异。秩和检验在显著性等于 5% 的水平下进行验证, 当 $p < 0.05$ 时, 可以认为两种算法存在显著差异, 否则认为两种算法寻优能力没有显著差异。本文将 7 种算法作为样本, 在以种群规模全部设置成 50, 空间维度设置成 30, 最大迭代次数为 500 次, 每个算法独立运行 30 次的条件下, 对 23 个基准测试函数求解, 并且将 ALAHA 与其他算法所得结果进行对比, 检验结果是否存在显著差异。Wilcoxon 检验 p 值如表 5 所示, 其中 N/A 表示实验样本数据相同, 数据无效。从表中可以看出, ALAHA 的 p 值除了在 F17~F23 上与原始算法 AHA 对比大于 0.05, 这也是因为原算法 AHA 在这些函数上都逼近最优值所导致的, 这也证明了优化算法没有降低原算法的良好性能。除此之外 ALAHA 与其他算法的 p 值基本小于 0.05, 说明 ALAHA 算法对比其他算法差异明显, 性能更好。

Table 5. Wilcoxon rank sum test results of each algorithm

表 5. 各算法 Wilcoxon 秩和检验结果

函数	AHA	LFD	HHO	WOA	GWO	SSA
F1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F2	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F3	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F4	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F5	8.31E-04	1.73E-06	1.73E-06	1.73E-06	6.89E-05	2.88E-06
F6	1.73E-06	2.37E-05	3.52E-06	1.73E-06	7.69E-06	1.73E-06
F7	6.98E-06	1.73E-06	2.60E-05	1.73E-06	1.73E-06	1.73E-06
F8	3.16E-02	3.59E-04	1.73E-06	2.11E-03	1.73E-06	1.73E-06
F9	N/A	1.73E-06	N/A	N/A	7.60E-06	1.73E-06
F10	N/A	1.73E-06	N/A	1.83E-05	1.30E-06	1.73E-06
F11	N/A	1.73E-06	N/A	1.56E-02	2.44E-04	1.73E-06
F12	2.35E-06	1.73E-06	8.22E-03	1.73E-06	1.92E-06	1.73E-06
F13	5.22E-06	3.41E-05	1.64E-05	3.72E-05	3.11E-05	1.25E-04
F14	1.56E-02	2.26E-02	7.27E-03	9.63E-04	1.13E-05	4.67E-02
F15	3.16E-03	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F16	N/A	2.60E-06	1.31E-04	1.73E-06	1.73E-06	4.48E-05
F17	5.00E-01	4.73E-06	1.82E-05	1.73E-06	1.73E-06	1.73E-06
F18	2.19E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.71E-06
F19	N/A	1.73E-06	1.73E-06	1.73E-06	1.73E-06	2.20E-06
F20	2.50E-01	2.37E-05	3.18E-06	8.19E-05	1.24E-05	3.52E-06
F21	1.25E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F22	1.25E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
F23	5.00E-01	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06

5. 结束语

对于标准 AHA 算法在全局搜索能力和搜索精度不足等问题, 本文通过引入莱维飞行因子, 结合领地觅食和引导觅食策略, 提高全局搜索程度; 采用邻域搜索策略提高算法寻优精度。仿真实验证明 ALAHA 算法相比原算法, 在继承原算法较好的性能基础上, 较大的提高了寻优能力和搜索精度, 但实验也看出了 ALAHA 算法在跳出局部最优方面还有进一步提高的空间。

基金项目

上海社会化养老模式创新研究(2019BGL020); 数字化驱动上海制造业绿色创新的机制及路径研究(22692105100)。

参考文献

- [1] Shefaei, A., Vahid-Pakdel, M.J. and Mohammadi-Ivatloo, B. (2018) Application of a Hybrid Evolutionary Algorithm on Reactive Power Compensation Problem of Distribution Network. *Computers & Electrical Engineering*, **72**, 125-136. <https://doi.org/10.1016/j.compeleceng.2018.09.012>
- [2] Kennedy, J. and Eberhart, R. (1995) Particle Swarms Optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, 27 November -1 December 1995, 1942-1948.
- [3] Dorigo, M. and Di Caro, G. (1999) Ant Colony Optimization: A New Meta-Heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Washington DC, 6-9 July 1999, 1470-1477.
- [4] Weiguo, Z., Liying, W. and Seyedali, M. (2022) Artificial Hummingbird Algorithm: A New Bio-Inspired Optimizer with Its Engineering Applications. *Computer Methods in Applied Mechanics and Engineering*, **388**, Article ID: 114194.
- [5] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [6] Assien, E., Masadeh, R. and Alzaqebah, A. (2017) Grey Wolf Optimization Applied to the 0/1 Knapsack Problem. *International Journal of Computer Applications*, **169**, 11-15. <https://doi.org/10.5120/ijca2017914734>
- [7] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., et al. (2017) Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Advances in Engineering Software*, **114**, 163-191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- [8] Eidari, A.A., Mirjalili, S., Faris, H., et al. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [9] Rao, R.V., Savsani, V.J. and Vakharia, D.P. (2012) Teaching—Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems. *Information Sciences*, **183**, 1-15. <https://doi.org/10.1016/j.ins.2011.08.006>
- [10] Li, S., Chen, H., Wang, M., Heidari, A.A. and Mirjalili, S. (2020) Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Generation Computer Systems*, **111**, 300-323. <https://doi.org/10.1016/j.future.2020.03.055>
- [11] Zhao, W., Zhang, Z. and Wang, L. (2020) Manta Ray Foraging Optimization: An Effective Bio-Inspired Optimizer for Engineering Applications. *Engineering Applications of Artificial Intelligence*, **87**, Article ID: 103300. <https://doi.org/10.1016/j.engappai.2019.103300>
- [12] Faramarzi, A., Heidarinejad, M., Stephens, B. and Mirjalili, S. (2020) Equilibrium Optimizer: A Novel Optimization Algorithm. *Knowledge-Based Systems*, **191**, Article ID: 105190. <https://doi.org/10.1016/j.knsys.2019.105190>
- [13] Zhao, W.G., Wang, L.Y. and Mirjalili, S. (2022) Artificial Hummingbird Algorithm: A New Bio-Inspired Optimizer with Its Engineering Applications. *Computer Methods in Applied Mechanics and Engineering*, **388**, Article ID: 114194. <https://doi.org/10.1016/j.cma.2021.114194>
- [14] Mohamed, A.H., Ragab, A.E., Ahmed, R.G., Ehab, E., Abdullah, M.S., et al. (2022) Parameter Identification and State of Charge Estimation of Li-Ion Batteries Used in Electric Vehicles Using Artificial Hummingbird Optimizer. *Journal of Energy Storage*, **51**, Article ID: 104535. <https://doi.org/10.1016/j.est.2022.104535>
- [15] Wang, M.X., Jiang, W., Ye, X.L., Wang, Z.W., Li, J. and Liu, Z.G. (2022) Optimal UAVs Placement for Localization Based on Artificial Hummingbird Algorithm. *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Xi'an, 25-27 October 2022, 1-6.
- [16] Abdelhady, R., Salah, K., Mohamed, H.H., Emad, M.A. and Hany, M.H. (2022) Accurate Photovoltaic Models Based

-
- on an Adaptive Opposition Artificial Hummingbird Algorithm. *Electronics*, **11**, Article 318. <https://doi.org/10.3390/electronics11030318>
- [17] Mohamed, A.E., Abdelghani, D., Shaker, E., Alhassan, M. and Mohamed, M.G. (2022) AHA-AO: Artificial Hummingbird Algorithm with Aquila Optimization for Efficient Feature Selection in Medical Image Classification. *Applied Sciences*, **12**, Article 9710. <https://doi.org/10.3390/app12199710>
- [18] Yildiz, B.S., Mehta, P., Sait, S.M., *et al.* (2022) A New Hybrid Artificial Hummingbird-Simulated Annealing Algorithm to Solve Constrained Mechanical Engineering Problems. *Materials Testing*, **64**, 1043-1050. <https://doi.org/10.1515/mt-2022-0123>
- [19] Zhao, W., Zhang, Z., Mirjalili, S., *et al.* (2022) An Effective Multi-Objective Artificial Hummingbird Algorithm with Dynamic Elimination-Based Crowding Distance for Solving Engineering Design Problems. *Computer Methods in Applied Mechanics and Engineering*, **398**, Article ID: 115223. <https://doi.org/10.1016/j.cma.2022.115223>
- [20] Ali, M.A.S., Fathimathul Rajeena, P.P. and Salama Abd Elminaam, D. (2022) A Feature Selection Based on Improved Artificial Hummingbird Algorithm Using Random Opposition-Based Learning for Solving Waste Classification Problem. *Mathematics*, **10**, Article 2675. <https://doi.org/10.3390/math10152675>
- [21] Yang, Q., Chen, W.N., Li, Y., *et al.* (2017) Multimodal Estimation of Distribution Algorithms. *IEEE Transactions on Cybernetics*, **47**, 636-650. <https://doi.org/10.1109/TCYB.2016.2523000>
- [22] 马卫, 朱娴. 基于莱维飞行扰动策略的麻雀搜索算法[J]. 应用科学学报, 2022, 40(1): 116-130.