

刘松林,高鹰,林锐灿,等. 混合策略改进的减法平均优化算法[J]. 智能计算机与应用,2024,14(8):70-77. DOI:10.20169/j.issn.2095-2163.240812

混合策略改进的减法平均优化算法

刘松林, 高 鹰, 林锐灿, 谭伟俊

(广州大学 计算机科学与网络工程学院, 广州 510006)

摘要: 减法平均的优化算法 (Subtraction-Average-Based Optimizer, SABO) 是一种新的元启发式算法, 具有参数少、易实现、结构简单等优点, 为了进一步提高 SABO 算法性能, 提出了一种混合策略改进的减法平均的优化算法 (HSABO)。首先, 在种群初始化时, 通过引入 Tent 混沌映射, 使初始化种群成员能够均匀分布; 其次, 在计算种群新个体位置时, 加入了自适应惯性权重, 提高了算法的局部搜索能力; 最后, 引入透镜成像反向学习策略, 对新个体位置以一定的概率进行扰动变异, 避免陷入局部最优, 增强算法全局探索的能力。通过 14 个标准测试函数, 对改进算法与其他优化算法进行测试比较。仿真结果表明, HSABO 算法是可行有效的, 相比于其他基本算法, 其寻优精度和收敛速度都有了很大的提高。

关键词: 减法平均的优化算法; Tent 混沌映射; 自适应惯性权重; 透镜成像反向学习

中图分类号: TP182 **文献标志码:** A **文章编号:** 2095-2163(2024)08-0070-08

Improved Subtraction-Average-Based Optimization based on hybrid strategy

LIU Songlin, GAO Ying, LIN Ruican, TAN Weijun

(School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China)

Abstract: Subtraction-Average-Based Optimizer (SABO) a new meta-heuristic algorithm, has the advantages of few parameters, easy to implement, simple structure, etc. In order to further improve the performance of the SABO algorithm, a mixed-strategy improved Subtraction-Average-Based Optimizer (HSABO) is proposed. Firstly, in the initialisation of the population, the initialised population members can be evenly distributed by introducing the Tent chaotic mapping; secondly, in the calculation of the new individual positions of the population, adaptive inertia weights are added to improve the algorithm's local search ability; lastly, the lens imaging reverse learning strategy is introduced to perturb the mutation of the new individual positions with a certain probability, avoiding falling into the local optimum and enhancing the algorithm's global exploration ability of the algorithm. The improved algorithm is tested and compared with other optimisation algorithms through 14 standard test functions. The simulation results show that the HSABO algorithm is feasible and effective, and compared with other basic algorithms, its optimisation accuracy and convergence speed are greatly improved.

Key words: Subtraction-Average-Based Optimizer; Tent chaotic mapping; adaptive inertia weight; lens imaging reverse learning

0 引言

基于种群的随机优化算法是求解优化问题的重要方法之一,也是求解全局优化问题的有效方法,受到了国内外众多学者的关注,在图像处理、数据处理、工程应用等领域得到了广泛的应用^[1-2]。通常,对于随机生成的初始种群,使用不同的数学方法或

随机操作在搜索空间中对其演化更新,并在优化过程中实现信息互换,能方便地从不同搜索空间获取信息,兼顾了算法的全局搜索能力和局部开发能力。基于种群的优化算法,随机地寻找优化问题的最优解,并不能保证一次运行便能找到,但随着种群规模和迭代次数的增加,找到全局最优解的概率会增加。近年来提出的群智能算法主要有算术优化算法

基金项目: 北航北斗技术成果转化及产业化资金资助项目 (BARI202104)。

作者简介: 刘松林 (1998-), 男, 硕士研究生, 主要研究方向: 智能优化算法; 林锐灿 (2000-), 男, 硕士研究生, 主要研究方向: 智能优化算法; 谭伟俊 (1998-), 男, 硕士研究生, 主要研究方向: 智能优化算法, 自然语言处理。

通讯作者: 高 鹰 (1963-), 男, 博士, 教授, CCF 会员, 主要研究方向: 智能优化算法, 盲信号处理。Email: csgy@ gzhu.edu.cn

收稿日期: 2023-06-01

(Arithmetic Optimization Algorithm, AOA)^[3]、正余弦优化算法 (Sine Cosine Algorithm, SCA)^[4]、灰狼优化算法 (Grey Wolf Optimization, GWO)^[2]、樽海鞘群优化算法 (Salp Swarm Algorithm, SSA)^[5]、蛇优化算法 (Snake Optimizer, SO)^[6]、鲫鱼优化算法 (Remora Optimization Algorithm, ROA)^[7]、黏菌优化算法 (Slime Mould Algorithm, SMA)^[8] 和粒子群优化算法 (Particle Swarm Optimization, PSO) 等。群智能优化算法具有简单性、可扩展性和运行时间短等特点, 在解决各种寻优问题时, 具有良好的操作性和强大的表现力。

减法平均的优化算法 (Subtraction - Average - Based Optimizer, SABO)^[9] 是 Pavel 等在 2023 年新提出的一种元启发式算法, 提出的 SABO 的基本启示是利用搜索代理的减法平均, 来更新种群成员在搜索空间中的位置。SABO 具有参数少、易实现、结构简单等优点, 但也存在着寻优精度低和易陷入局部最优等缺点。

为了进一步提高 SABO 算法的寻优精度和收敛能力。本文提出了一种混合策略改进的 SABO 算法。HSABO 算法采用 Tent 混沌映射对种群进行初始化, 使生成的初始种群伪随机性强, 增加了种群多样性, 使初始种群分布更均匀; 通过引入线性惯性权重, 提升了 SABO 的局部搜索能力, 提升了寻优精度; 引入透镜成像反向学习, 有利于算法跳出局部最优, 增强算法全局探索的能力。

1 减法平均的优化算法

减法平均的优化算法 (SABO) 是一种基于数学概念的元启发式算法, Pavel 等首次提出了基于整个种群个体的差值计算平均值, 以进行种群个体更新的方法。这种方法可以避免对种群中特定个体的依赖, 从而更好地在开发过程与探索过程中取得平衡。SABO 算法的实现步骤如下:

步骤 1 随机初始化种群。在初始化阶段采用如下公式随机生成种群个体的位置:

$$x_{i,d} = lb_d + r_{i,d} * (ub_d - lb_d) \quad (1)$$

其中, ub_d 和 lb_d 分别是个体 $X_i (i = 1, \dots, N)$ 的第 d 维分量的上界和下界; $r_{i,d}$ 是区间 $[0, 1]$ 内的随机数; $x_{i,d}$ 代表第 i 个种群个体的第 d 维 ($d = 1, \dots, m$); 种群数量为 N ; 维度为 m 。

步骤 2 计算新个体位置。算法中采用的基于种群中所有个体的算术平均值更新个体, 而不是仅仅采用个别个体 (如: 最优个体或最差个体) 进行种

群更新, SABO 提出了一种全新的计算差值的方式 “ $-_v$ ”, $A -_v B$ 定义如下:

$$A -_v B = \text{sign}(F(A) - F(B)) (A - \vec{v} \times B) \quad (2)$$

其中, A 和 B 是维度为 m 的两个种群个体; $F(A)$ 和 $F(B)$ 是其适应度值; \vec{v} 是 m 维向量, 分量从集合 $\{1, 2\}$ 中随机生成; “ \times ” 所求为两个向量的哈达玛积; “ $\text{sign}()$ ” 是求符号的函数。

在 SABO 中, 每一轮计算新个体位置时, 种群个体 X_i 都会基于式 (2) 中的减法公式与其他所有个体做 “ $-_v$ ” 运算后求平均值, 并以此为偏移量更新位置, SABO 的种群新个体位置的计算方程如下:

$$X_i^{\text{new}} = X_i + \vec{r}_i \times \frac{1}{N} \sum_{j=1}^N (X_i -_v X_j) \quad (3)$$

其中, $X_i^{\text{new}} (i = 1, 2, \dots, N)$ 为根据总数为 N 的种群中的个体 X_i 生成的新个体位置, \vec{r}_i 是维数为 m 的向量, 其分量取自区间 $[0, 1]$ 并呈正态分布。

步骤 3 更新种群个体位置。在获取新的种群个体后, 如果新个体的适应度值优于原个体, 则保留至下一次迭代, 否则采用原个体 X_i 。更新公式如下:

$$X_i = \begin{cases} X_i^{\text{new}}, & F_i^{\text{new}} < F_i \\ X_i, & \text{else} \end{cases} \quad (4)$$

其中, F_i^{new} 为新个体 X_i^{new} 的适应度值, F_i 为原个体 X_i 的适应度值, 假设问题是目标函数 $F(X)$ 的最小化问题。

步骤 4 更新完种群中的所有个体后, 判断是否满足终止条件, 以确认进入下一次迭代或是结束迭代输出结果。

SABO 算法伪代码如下:

算法 1 SABO 算法

输入 终止条件; 种群规模 N ; 问题维度 m ; 目标函数;

输出 全局最佳位置 X_{best} 与其适应度值

1 初始化种群, 采用公式 (1) 生成 X

2 for $t = 1$ to T

3 for $i = 1$ to N

4 根据公式 (3) 得 X_i^{new}

5 if X_i^{new} 适应度值优于 X_i

6 $X_i = X_i^{\text{new}}$

7 end if

8 end for

9 保留当前最佳个体的位置

10 end for

11 输出 X_{best} 及其适应度值

2 改进减法平均的优化算法

2.1 基于 Tent 混沌映射的种群初始化

对于元启发式算法而言,种群的初始化对算法的收敛速度和收敛精度有一定的影响。为了获得更高的全局搜索能力和种群多样性,在进行迭代寻优时,要尽可能的使种群初始值在搜索空间中分布均匀。在原 SABO 算法中,种群初始化是随机的,并不能保证初始种群能够均匀分布,种群位置随机分布,使得部分个体远离最优解,从而会影响算法的收敛速度。为了解决原算法中存在的这些问题,引入了 Tent 混沌映射。

目前,混沌映射已经成为实现对优化算法改进的一种常用手段。改进方法基于混沌远离,通过混沌奇数产生随机数代替常规伪随机数生成群体的初始位置,且在大量的文献中已得到验证。如:刘志强等^[10]利用 Tent 混沌映射丰富了狼群算法的初始化,提出了 TGWO,并通过实验验证了该算法在路径寻优上的优越性。基于混沌序列的思想,本文采用了一种 Tent 混沌映射,其具有良好的分布性和随机性,并且映射呈现的结果分布密度比较均匀,其表达式如下:

$$x(t+1) = \begin{cases} x(t)/\alpha, & x(t) < \alpha \\ (1-x(t))/(1-\alpha), & x(t) \geq \alpha \end{cases} \quad (5)$$

其中, t 是迭代次数,本文对 α 取 0.5。

引入 Tent 混沌映射后,提升了种群多样性,让种群分布更加均匀,从而使算法可以快速收敛并接近最优解。

2.2 自适应惯性权重

在原算法中,前期收敛速度较快,易陷入局部最优;后期收敛速度慢,搜索范围变小。受文献[11]的启发,引入自适应惯性权重 ω ,公式如下:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times (t/T)^2 \quad (6)$$

其中, ω 是惯性权重, ω_{\max} 是惯性权重最大值, ω_{\min} 是惯性权重最小值, $\omega_{\max} = 0.9$ 、 $\omega_{\min} = 0.4$; t 是当前迭代次数; T 是最大迭代次数。

设置权重后,在迭代初期 ω 较大,有利于在全局范围内进行搜索;而在迭代后期 ω 较小,更有利于向最优解靠近。

新个体位置如下:

$$X_i^{\text{new}} = \omega \times X_i + r_i \times \frac{1}{N} \sum_{j=1}^N (X_i - v_j X_j) \quad (7)$$

2.3 透镜成像反向学习策略

反向学习由 Tizhoosh^[12]于 2005 年提出,其主要

思想是生成可行解的相反解,评价相反解并选择更好的候选解。反向学习是通过计算当前位置的反向解,从而扩大搜索范围的一种改进策略。在元启发式算法中,采用反向学习策略,可以提升算法的寻优能力,加快收敛速度。

定义 1 反向数:假设实数 $x \in [a, b]$, 则 x 的反向数 $x^* = a + b - x$ 。

定义 2 反向点:假设 $X = (x_1, x_2, \dots, D)$ 为 D 维空间中的一点,且 $x_j \in [a_j, b_j]$, 则 X 的反向点表示为 $X(x_1^*, x_2^*, \dots, x_D^*)$, 其中 $x_j^* = a_j + b_j - x_j$ 。

反向迭代策略虽然在种群初始化中起到重要作用,但是通过反向学习求得的反向解是固定的,如果个体已经陷入局部最优,且其反向解劣于当前解,则反向学习策略无法使个体跳出局部最优。受文献[13-14]的启发,利用透镜成像原理对反向学习策略进行扩展用以解决上述问题。

如图 1 所示,假设在搜索空间中寻找目标值反向点的过程为透镜成像的过程,在横坐标区间 $[a, b]$ 上的初始高度为 h 的物体,通过原点 o ($o = (a+b)/2$) 处,焦距为 r 的透镜投影到高度为 h' 处成像。此时, x_{best} 就能够以 o 为基点找到对应的反向点 x_{best}^* 。由透镜成像原理可得以下公式:

$$\frac{(a+b)/2 - x_{\text{best}}}{x_{\text{best}}^* - (a+b)/2} = k \quad (8)$$

$$\frac{r}{x_{\text{best}}^* - (a+b)/2 - r} = k \quad (9)$$

其中, $k = \frac{h}{h'}$ 为缩放因子,变换上式可以得到反

向点 x_{best}^* 的计算公式:

$$x_{\text{best}}^* = \frac{a+b}{2} - \frac{a+b}{2k} - \frac{x}{k} \quad (10)$$

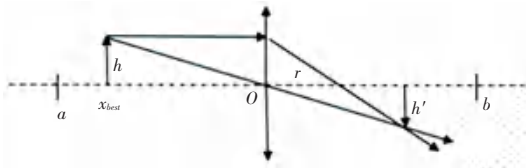


图 1 透镜成像过程图

Fig. 1 Lens imaging process diagram

由上述可知,当 $k = 1$ 时,透镜成像反向学习策略为一般的反向学习策略,采用反向学习得到的是固定反向解。而通过调整 k 的大小,可以在透镜成像反向学习中获得动态变化的反向解,提高收敛速度,使算法更有机会收敛于全局最优解。

本文引入透镜成像反向学习策略,使基本的

SABO 算法跳出局部最优。该阶段的改进公式如下:

$$X_i^{new'} = \frac{(ub_d - lb_d)}{2} + \frac{(ub_d + lb_d)}{2k} - \frac{X_i^{new}}{k} \quad (11)$$

其中, $X_i^{new'}$ 为 X_i^{new} 的透镜反向解。通过调节 k 的大小, 可以在透镜成像反向学习中获得动态变化的反向解, 进一步提升算法的寻优能力, 本文采用的 k 值计算公式如下:

$$k = (1 + (3 \times t/T)^{0.5})^8 \quad (12)$$

2.4 混合策略改进的减法平均的优化算法

基于上述改进策略, 本文提出了一种混沌策略改进的减法平均的优化算法。首先在算法初始化阶段引入 Tent 混沌映射以增强初始解分布的均匀性; 在计算新个体位置时, 采用了自适应惯性权重, 增强了算法执行前期的探索能力和后期的收敛速度; 最后, 引入了透镜成像反向学习策略, 对新个体位置进行更新, 降低算法陷入局部最优的概率。HSABO 算法流程图如图 2 所示, 分为以下 5 个步骤

- (1) 初始化参数并根据公式(5)采用 Tent 混沌映射生成初始种群;
- (2) 根据公式(6)更新自适应惯性权重;
- (3) 根据公式(7)生成新个体, 并通过公式(11)做进一步寻优;
- (4) 根据公式(4)更新种群个体位置, 并保留最优位置;
- (5) 判断是否满足终止条件已确认是否结束迭代输出结果或是回到步骤(2)。

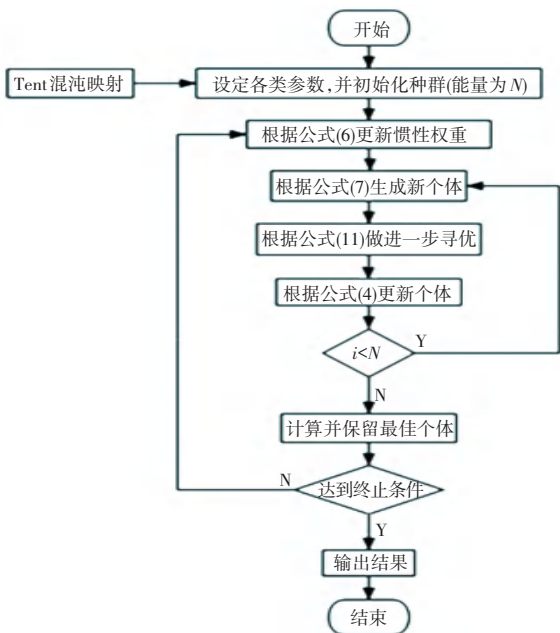


图 2 HSABO 算法流程图

Fig. 2 HSABO algorithm flowchart

3 仿真实验分析

3.1 实验环境及参数设置

为了进一步验证 HSABO 算法在收敛速度和收敛精度上的优越性, 将其与基本 SABO 算法、粒子群优化算法(PSO)、鲸鱼优化算法(Whale Optimization Algorithm, WOA)^[15]、金豺优化算法(Golden Jackal Optimization, GJO)^[16]、非洲秃鹫优化算法(African Vultures Optimization Algorithm, AVOA)^[17] 以及正余弦优化算法(SCA)进行比较。

实验环境为: Windows11 操作系统, CPU 为 Intel Core i7-12700H, 主频 2.3 GHz, 内存 16 GB, 算法用 Matlab R2022a 编写。

所要测试算法的种群规模统一设定为 30, 最大迭代次数为 500 次, 各算法其余的参数设置见表 1。

表 1 算法参数设置

Table 1 Algorithm parameter settings

算法	参数值
HSBO	$\omega_{max} = 0.9, \omega_{min} = 0.2, k = 2$
PSO	$c_1 = c_2 = 2, \omega_{max} = 0.9, \omega_{min} = 0.2$
WOA	α 由 2 线性下降到 0
GJO	$c_1 = 1.5, \beta = 1.5$
SCA	$\alpha = 2$
AVOA	$p_1 = 0.6, p_2 = 0.4, p_3 = 0.6, \omega = 0.6, \alpha = 0.8, \beta = 0.2$

3.2 测试函数

为了验证 HSABO 算法的有效性, 本文对 14 个国际基准测试函数进行仿真实验, 来检验算法的改进效果, 选取的测试函数包括单峰、多峰、固定维度多峰函数。其中, $f_1 \sim f_4$ 为单峰函数, 用于测试算法的全局探索能力; $f_5 \sim f_7$ 为多峰函数, 侧重于测试算法的全局探索能力以及跳出局部最优解的能力; $f_8 \sim f_{14}$ 是固定维度多峰函数。所用测试函数详情见表 2。

3.3 实验结果与分析

为了减小实验过程中随机性的影响, 设置最大迭代次数为 500 次, 对每个测试函数独立运行 30 次, 计算其平均值、最优值、标准差(Std), 实验结果见表 3。对于每个测试函数, 在固定的迭代次数下, 从最优值可以看出算法的寻优能力和收敛精度, 而均值和标准差反映了算法的鲁棒性和稳定性。

表2 测试函数
Table 2 Test functions

函数	维度	定义域	最优值
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	$-418.952 \ 9 \times D$
$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i + 10)]$	30	$[-5.12, 5.12]$	0
$f_7(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] \} + (x_i - 1)^2 [1 + \sin^2(2\pi x_i)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0
$f_8(x) = \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \frac{\ddot{o}^{-1}}{\varnothing}$	2	$[-65, 65]$	1
$f_9(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_1 x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.000 30
$f_{10}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1
$f_{11}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 5]$	0.3
$f_{12}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$f_{13}(x) = - \sum_{i=1}^4 c_i \exp\left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	$[1, 3]$	-3
$f_{14}(x) = \sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10

从表3中可以看出,对于14个测试函数,本文提出的HSABO算法的平均值、最优值、标准差均优于SABO、PSO、WOA、GJO、SCA以及AVOA算法,说明HSABO算法具有更高的寻优精度、稳定性和鲁棒性。其中,对于函数 f_1 、 f_2 、 f_3 、 f_4 、 f_5 、 f_6 、 f_7 、 f_8 、 f_9 、 f_{10} 、 f_{11} 、 f_{12} 和 f_{13} ,HSABO算法都能找到其理论最优值;除函数 f_{14} 外,标准方差均为0,说明经过改进后的优化算法稳定性得到大大提高。

为了能够更直观体现比较HSABO算法与其他算法的寻优性能,图3给出了7种算法在14个标准测试函数上搜索最优值过程中,随迭代次数增加的收敛曲线。从图中可以看出,不管是单峰函数还是多峰函数,HSABO的收敛速度和收敛精度都是最好的。并且,HSABO的初始最优值明显优于其他6种算法,充分证明了Tent混沌初始化能够使初始种群分布更均匀,在迭代初期就有较好的初始寻优精度。

表 3 7 种算法实验结果
Table 3 Experimental results of seven algorithms

		HSABO	SABO	PSO	WOA	GJO	SCA	AVOA
f_1	Mean	0	2.48e-113	1.53e-05	1.51e-75	5.60e-54	3.02e+01	6.29e-281
	Best	0	1.02e-113	1.38e-06	9.70e-81	8.45e-57	1.97e+00	0
	Std	0	1.04e-113	1.02e-05	2.22e-75	1.22e-53	43.937 6	0
f_2	Mean	0	7.63e-59	1.10e-03	1.08e-50	1.33e-32	3.06e-02	2.83e-147
	Best	0	4.22e-59	4.30e-05	5.22e-55	5.81e-34	4.49e-04	4.80e-171
	Std	0	2.52e-59	1.37e-03	2.17e-50	1.73e-32	6.25e-02	6.32e-147
f_3	Mean	0	6.43e-53	2.20e+03	4.68e+04	1.43e-18	1.03e+04	3.96e-220
	Best	0	4.94e-62	9.35e+02	4.19e+04	1.20e-22	4.82e+03	4.11e-266
	Std	0	1.42e-52	9.90e+02	5.81e+03	2.02e-18	4.04e+03	0
f_4	Mean	0	3.08e-37	1.68e+00	3.16e+00	3.52e-17	3.44e+00	9.65e-159
	Best	0	3.68e-37	1.21e+00	9.42e-01	7.93e-18	2.03e+00	5.99e-141
	Std	0	1.83e-37	4.23e-01	2.27e+00	5.01e-17	1.90e-01	1.34e-140
f_5	Mean	0	3.89e-146	1.51e-19	4.12e-109	3.44e-191	1.77e-05	0
	Best	0	1.97e-153	3.71e-22	1.72e-122	5.12e-203	6.31e-08	0
	Std	0	8.59e-146	1.47e-19	7.45e-109	0	2.84e-05	0
f_6	Mean	0	3.52e-114	1.36e-06	1.99e-76	3.27e-56	4.61e-01	1.04e-310
	Best	0	6.45e-115	2.65e-07	1.22e-80	3.17e-57	7.89e-02	0
	Std	0	3.63e-114	1.38e-06	4.15e-76	4.24e-56	4.55e-01	0
f_7	Mean	0	3.04e-233	1.23e-111	0	0	1.83e-99	0
	Best	0	7.90e-237	5.24e-124	0	0	2.30e-110	0
	Std	0	0	2.75e-111	0	0	4.09e-99	0
f_8	Mean	0	4.19e-123	1.81e-87	8.75e-111	0	6.11e-67	0
	Best	0	3.22e-127	9.10e-92	6.61e-126	0	7.12e-77	0
	Std	0	4.73e-123	2.57e-87	1.95e-110	0	5.77e-67	0
f_9	Mean	0	2.27e-14	5.13e+01	0	0	2.81e+01	0
	Best	0	0	2.29e+01	0	0	4.02e-01	0
	Std	0	3.11e-14	1.85e+01	0	0	2.62e+01	0
f_{10}	Mean	0	0	5.10e+01	0	0	6.02e+01	0
	Best	0	0	4.40e+01	0	0	3.06e+01	0
	Std	0	0	1.25e+01	0	0	2.93e+01	0
f_{11}	Mean	4.23e-16	1.11e-14	7.04e-01	3.29e-15	6.13e-15	2.03e+01	4.44e-16
	Best	4.21e-16	7.55e-15	6.90e-04	4.44e-16	4.00e-15	2.02e+01	4.44e-16
	Std	0	2.51e-15	1.12e+00	1.59e-15	1.95e-15	4.73e-02	0
f_{12}	Mean	0	4.08e-02	8.52e-03	0	0	1.09e+00	0
	Best	0	9.13e-02	3.22e-06	0	0	8.09e-01	0
	Std	0	0	8.50e-03	0	0	2.13e-01	0
f_{13}	Mean	0	1.64e-59	9.37e-04	2.15e-50	2.63e-32	4.42e+00	1.03e-163
	Best	0	1.06e-59	1.43e-04	4.35e-57	1.29e-33	2.59e-01	9.27e-168
	Std	0	4.83e-60	1.79e-03	4.81e-50	4.13e-32	4.63e+00	2.30e-153
f_{14}	Mean	2.53e-10	5.30e-03	8.16e-03	1.33e-05	8.13e-03	2.45e-02	4.18e-12
	Best	1.91e-14	2.47e-04	4.13e-07	1.39e-06	1.89e-03	1.82e-03	2.02e-13
	Std	4.08e-10	4.62e-03	1.07e-02	1.06e-05	6.94e-03	2.42e-02	8.14e-12

从图中可以看出,除函数 f_{14} 外,其余函数的 HSABO 算法寻优过程近似于直线,充分说明了 HSABO 算法在整个迭代过程中,寻优速度更高,有较好的全局搜索能力,避免了陷入局部最优。其中,从 14 个函数的收敛曲线可以看出,HSABO 算法在未达到最大迭代次数时,就已经找到了理论最优值。

函数 f_1 f_3 f_5 f_6 f_7 f_8 f_9 f_{10} f_{11} 和 f_{12} 函数的优化图表明,在迭代过程中,HSABO 算法可以快速接近理论最优值,这充分证明 HSABO 算法的可行性、有效性以及其寻优性能的提升。同时,通过上述可知,改进的 HSABO 算法能有效提高全局搜索能力,并在算法后期,有效提高局部开发能力。

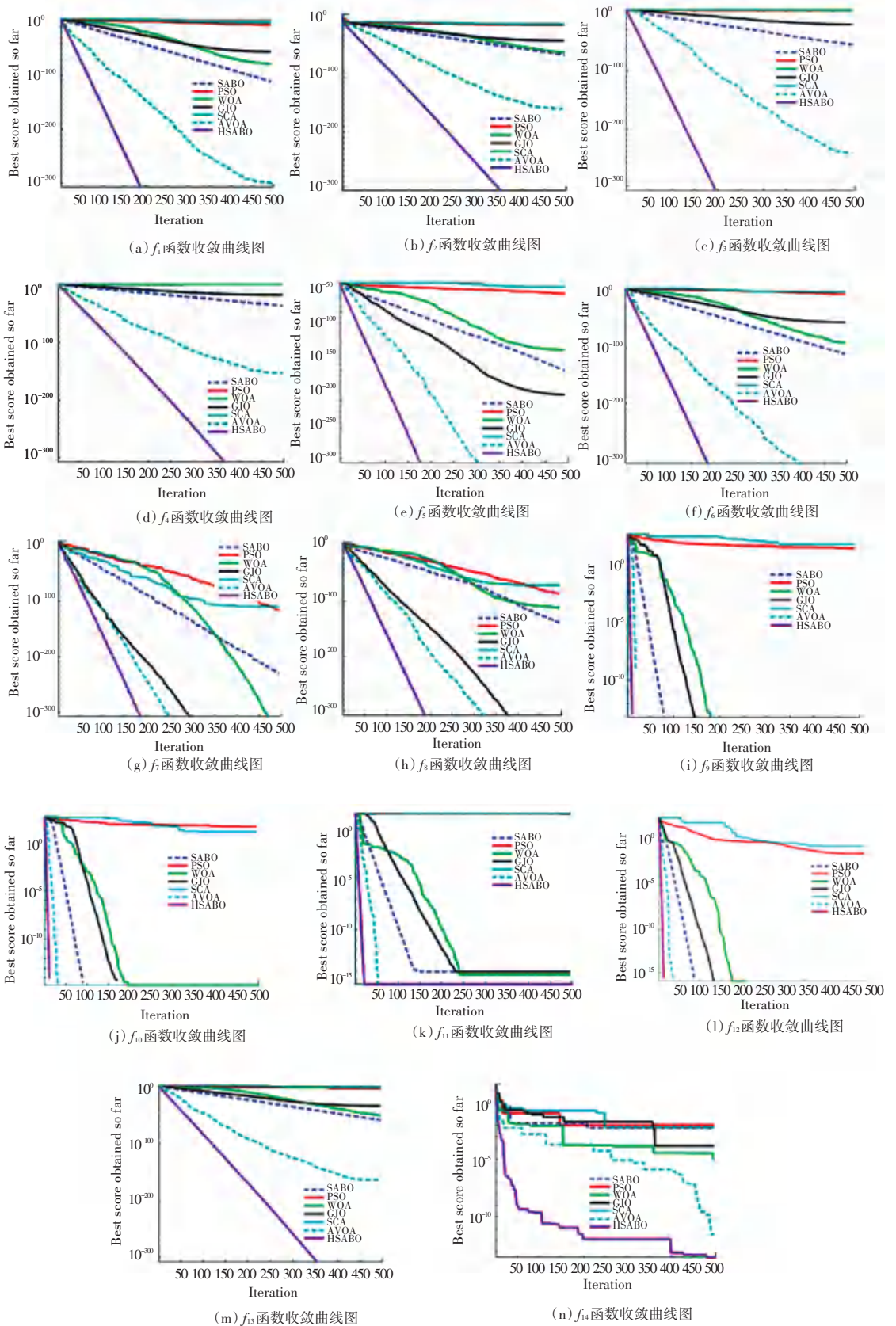


图3 测试函数收敛曲线图

Fig. 3 Test function convergence curve

4 结束语

本文在 SABO 算法的基础上提出了混合策略改进的减法平均优化算法-HSABO。通过引入 Tent 混沌映射、线性惯性权重、透镜成像反向学习策略,能够在很大程度上改善基本 SABO 算法寻优精度低和易陷入局部最优的不足。并且进行了 14 个基准函数的仿真测试。实验结果表明,改进的 SABO 算法是可行有效的,在收敛精度和收敛速度上都优于原 SABO 算法及其它比对算法,其寻优性能有了很大的提高。下一步的研究工作,重点考虑如何将改进后的算法应用到更多领域。

参考文献

- [1] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95 International Conference on Neural Networks. 1995: 1942-1948.
- [2] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey Wolf Optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.
- [3] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. Arithmetic optimization algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 113609.
- [4] MIRJALILI S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.
- [5] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [6] HASHIM F A, HUSSEIN A G. Snake optimizer: A novel meta-heuristic optimization algorithm[J]. Knowledge-Based Systems, 2022, 242: 108320.
- [7] JIA H, PENG X, LANG C. Remora optimization algorithm[J]. Expert Systems with Applications, 2021, 185: 115665.
- [8] LI S, CHEN H, WANG M, et al. Slime mould algorithm: A new method for stochastic optimization [J]. Future Generation Computer Systems, 2020, 111: 300-323.
- [9] TROJOVSKÝ P, DEGHANI M. Subtraction - average - based optimizer: A new swarm - inspired metaheuristic algorithm for solving optimization problems [J]. Biomimetics, 2023, 8(2): 149.
- [10] 刘志强, 何丽, 袁亮, 等. 采用改进灰狼算法的机器人路径规划[J]. 西安交通大学学报, 2022, 56(10): 49-60.
- [11] 鲁晓艺, 刘升, 韩斐斐, 等. 基于自适应权重的缎蓝园丁鸟优化算法[J]. 智能计算机与应用, 2018, 8(6): 94-100.
- [12] TIZHOOSH H R. Opposition-based learning: A new scheme for machine intelligence[C]//Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA - IAWTIC'06). IEEE, 2005: 695-701.
- [13] 周鹏, 董朝轶, 陈晓艳, 等. 基于 Tent 混沌和透镜成像学习策略的平衡优化器算法[J]. 控制与决策, 2023, 38(6): 1569-1576.
- [14] 龙文, 伍铁斌, 唐明珠, 等. 基于透镜成像学习策略的灰狼优化算法[J]. 自动化学报, 2020, 46(10): 2148-2164.
- [15] MIRJALILI S, LEWIS A. Whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [16] CHOPRA N, MOHSIN A M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications [J]. Expert Systems with Applications, 2022, 198: 116924.
- [17] ABDOLLAHZADEH B, GHAREHCHOPOGH F S, MIRJALILI S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems [J]. Computers & Industrial Engineering, 2021, 158: 107408.