

自适应差分变异的人工鱼群算法^①



郭长珍, 李 整

(华北电力大学(保定) 计算机系, 保定 071003)
通信作者: 郭长珍, E-mail: 22022221096@ncepu.edu.cn

摘 要: 针对人工鱼群算法存在的全局搜索能力欠缺, 鲁棒性差及易陷入局部极值等不足, 提出一种自适应差分变异的人工鱼群算法 (ADMAFSA). 首先, 该算法采用自适应视野和步长策略, 改善种群个体在较优区域的精细搜索能力, 提升算法的寻优精度. 其次, 在人工鱼群的随机行为中引入反向学习机制, 通过发掘潜在的寻优空间, 提高算法的全局搜索性能, 避免算法早熟收敛. 最后, 借鉴差分进化算法对质量较差的人工鱼进行变异操作, 从而增加鱼群的多样性, 降低算法陷入局部极值的可能性. 为验证改进算法的性能, 本文对 6 个基准测试函数和 8 个 CEC2019 函数进行仿真, 与其他 AFSA 变体、新型智能算法进行对比, 实验结果表明, ADMAFSA 在寻优精度和鲁棒性方面均有所提高. 最后, 在齿轮系设计问题上, 进一步证明了改进算法具有较好的优化效果.

关键词: 人工鱼群算法; 自适应; 差分变异; 反向学习机制; 基准测试函数

引用格式: 郭长珍, 李整. 自适应差分变异的人工鱼群算法. 计算机系统应用, 2024, 33(8): 214-221. <http://www.c-s-a.org.cn/1003-3254/9572.html>

Adaptive and Differential Mutation Artificial Fish Swarm Algorithm

GUO Chang-Zhen, LI Zheng

(Department of Computer, North China Electric Power University (Baoding), Baoding 071003, China)

Abstract: The original artificial fish swarm algorithm (AFSA) has weak global search ability and poor robustness and is easy to fall into local extremum. Given these problems, an adaptive and differential mutation artificial fish swarm algorithm (ADMAFSA) is proposed. Firstly, it utilizes an adaptive vision field and step length strategy to improve the fine search ability of individuals in better areas of the population and enhance the optimization accuracy of the algorithm. Secondly, to explore potential better areas, the opposition-based learning mechanism is introduced into the random behavior of artificial fish swarms. Thereby, the algorithm can get better global searching ability and avoid premature convergence. Finally, inspired by the differential evolution algorithm, a mutation operation is applied to poorly performing artificial fish to increase the diversity of the fish swarm and reduce the possibility of the algorithm falling into the local extremum. To validate the performance of the improved algorithm, the proposed algorithm is tested with six benchmark test functions and eight CEC2019 functions. The experimental results indicate that, compared to other AFSA variants and novel intelligent algorithms, ADMAFSA demonstrates improvements in terms of optimization accuracy and robustness. Furthermore, in designing the train of gears, the optimization effectiveness of the improved algorithm is further proved.

Key words: artificial fish swarm algorithm (AFSA); adaptive; differential mutation; opposition-based learning mechanism; benchmark test function

① 收稿时间: 2024-01-30; 修改时间: 2024-02-29; 采用时间: 2024-03-11; csa 在线出版时间: 2024-06-28
CNKI 网络首发时间: 2024-07-01

优化是指在给定的限制条件下,采用技术手段寻求某一问题的最佳解决方案^[1].现今对优化问题的求解方法主要有数学方法和元启发式算法等.传统数学方法在求解小规模优化问题时,具有简单快速,求解精度高等优点,但是对于高维复杂问题,容易出现“维度灾”现象,且计算时间大大延长.相较于传统的数学方法,元启发式算法不局限于特定的问题结构和假设^[2].这类算法通过运用启发式规则和自适应机制,能够灵活地搜索解空间,克服维度灾难,并在相对较短的时间内找到接近最优的解^[3].因此,元启发式算法在实践中被广泛应用^[4-7].群智能优化算法是一类模拟自然界中生物的生存行为而产生的元启发式算法^[8].其基本理论是利用群体之间的合作与信息交流,以及生物自身的进化等来达到寻优的目的.该类算法因其鲁棒性强,实现简单,优化精度高等优点,受到了众多学者的关注研究^[9].近年来,涌现出了许多性能卓越的新型群智能算法,比如:麻雀优化算法(sparrow search algorithm, SSA)^[10],蜣螂优化算法(dung beetle optimization, DBO)^[11],灰狼优化算法(grey wolf optimization, GWO)^[12],鲸鱼优化算法(whale optimization algorithm, WOA)^[13],哈里斯鹰优化算法(harris hawks optimization, HHO)^[14],沙丘猫群优化算法(sand cat swarm optimization, SCSO)^[15]等.

人工鱼群算法(artificial fish swarm algorithm, AFSA)是李晓磊博士提出的一种模拟鱼类基本行为的仿生算法^[16].鱼类一般通过移动聚集在营养物质最丰富的区域,根据此现象,对鱼群的觅食、聚群、追尾、随机行为进行仿真,从而达到算法全局寻优的目的.该算法实现简单,对初始值不敏感,具有较强的鲁棒性.但是,AFSA也存在后期收敛精度低、可能陷入局部极值等缺点.因此,许多文献提出了改进方法.比如,文献^[17]引入可变视野,对人工鱼移动策略做出改进,并且模仿遗传算法中的变异操作对种群进行更新,在一定程度上避免了算法陷入局部极值.文献^[18]引入反向解调整鱼群的行为,并采用自适应视野和步长策略,平衡全局搜索和局部搜索,最后运用高斯变异机制,对劣质解进行改进,使得算法的鲁棒性和寻优精度得到了一定的提升.文献^[19]结合引力搜索算法提出一种改进的人工鱼群算法,并应用于模拟机运动洗出优化,实验结果表明,改进算法具有较为优秀的求解性能,适合求解实际优化问题.

上述文献都在一定程度上改进了AFSA的寻优性

能,但是在跳出局部最优、收敛速度和精度等方面,AFSA仍有较大的提升空间.因此,本文提出一种自适应差分变异的人工鱼群算法(adaptive and differential mutation artificial fish swarm algorithm, ADMAFSA).改进算法在人工鱼的移动行为中引入反向学习机制,避免了原始随机行为的盲目性,提高了种群的多样性.为降低算法陷入局部最优的可能性,改进算法借鉴差分进化算法的变异机制,每次迭代完成后对较差人工鱼进行变异.此外,算法采用自适应视野和步长,在迭代前期保证全局搜索,在迭代后期缓慢精准地向最优区域收敛.在6个基准测试函数和8个CEC2019函数上的实验结果验证了改进算法的有效性.最后将ADMAFSA应用于求解齿轮系设计这种实际优化问题上.

1 人工鱼群算法(AFSA)

ASFA通过模拟鱼类基本行为实现群智能优化目标.鱼类一般通过移动聚集在营养物质最丰富的区域,据此,对鱼群的觅食、聚群、追尾、随机行为进行仿真,从而达到算法全局寻优的目的.设置单条人工鱼的状态为向量 $X = (x_1, x_2, \dots, x_D)$,将当前人工鱼 X 对应的食物浓度记为 $Y = f(X)$,其中 Y 为目标函数值.两条人工鱼 i 与 j 之间的距离表示为 $d_{ij} = \|X_i - X_j\|$,算法中的其他参数包括人工鱼的拥挤度因子 δ ,视野 $visual$,步长 $step$,尝试次数 try_number .其中视野 $visual$ 和步长 $step$ 分别决定人工鱼所能搜索的最大范围以及每次移动的距离.算法包括觅食、聚群、追尾、随机行为,下面对几种行为进行详细描述.

1.1 觅食行为

在求最小值问题中,当前人工鱼 X_i 在视野 $visual$ 范围内(即 $d_{ij} < visual$)进行随机搜索,找到位置 X_j ,若 $Y_i > Y_j$,说明状态 j 优于状态 i ,人工鱼按式(1)向 X_j 移动一步,否则重新随机选择 X_j .如果连续尝试 try_number 次后仍未找到满足条件的 X_j ,则执行随机行为.

$$X_{i,next} = X_i + Rand() \times step \times \frac{X_j - X_i}{\|X_j - X_i\|} \quad (1)$$

1.2 聚群行为

统计当前人工鱼 X_i 视野范围内的鱼群总数目 n_f ,如果该 n_f 条人工鱼的中心位置人工鱼 X_c 满足式(2),说明中心位置 X_c 处食物浓度较高且周围人工鱼不太拥挤,因此,按式(3)向 X_c 进行移动.否则,执行觅食

行为.

$$Y_c \cdot n_f < \delta \cdot Y_i \quad (2)$$

$$X_{i|next} = X_i + Rand() \times step \times \frac{X_c - X_i}{\|X_c - X_i\|} \quad (3)$$

1.3 追尾行为

搜索当前人工鱼 X_i 视野范围内食物浓度最高的位置 X_{max} , 如果 X_{max} 满足式 (4), 说明位置 X_{max} 处食物浓度较高且周围人工鱼不太拥挤, 因此, 按式 (5) 向 X_{max} 进行移动. 否则, 执行觅食行为.

$$Y_{max} \cdot n_f < \delta \cdot Y_i \quad (4)$$

$$X_{i|next} = X_i + Rand() \times step \times \frac{X_{max} - X_i}{\|X_{max} - X_i\|} \quad (5)$$

1.4 随机行为

随机行为是指人工鱼 X_i 在当前视野范围内 (即 $d_{ij} < visual$) 随机搜索, 选择一个位置并向其移动. 如式 (6) 所示.

$$X_{i|next} = X_i + Rand() \times visual \quad (6)$$

2 自适应差分变异的人工鱼群算法

标准 AFSA 算法具有实现相对简单, 对参数不敏感, 并行化能力强等优点, 但是相对于其他高效的优化算法, AFSA 的收敛速度和精度可能较差. 因此, 本文提出一种改进的 AFSA, 以期在合理的计算时间内找到高质量的最优解. 下面对算法的主要改进机制进行详细描述.

2.1 自适应视野和步长

AFSA 算法使用固定视野和固定步长, 这有利于算法前期迅速向较优区域收敛, 但是在迭代后期, 由于视野值和步长值过大, 鱼群执行觅食行为和随机行为频繁, 从而无法更精准地找到全局最优解. 针对以上问题, 本文引入可变视野和可变步长. 鱼群的视野 $visual$ 和步长 $step$ 按式 (7) 随着迭代次数的递增呈线性减小, 保证算法在迭代前期进行全局搜索, 在迭代后期实现在极值点周围的精细寻优.

$$\begin{cases} visual = visual_{max} - (visual_{max} - visual_{min})/d_{max} \times d \\ step = step_{max} - (step_{max} - step_{min})/d_{max} \times d \end{cases} \quad (7)$$

其中, $visual_{max}$ 、 $visual_{min}$ 分别为设置的视野最大值、最小值; d_{max} 为最大迭代次数, d 表示当前迭代次数;

$step_{max}$ 、 $step_{min}$ 分别为设置的步长最大值、最小值.

2.2 改进的随机行为

由上文可知, 在觅食行为中, 若连续尝试 try_number 次后仍未找到满足条件的 X_j 则执行随机行为. 因此, 当执行随机行为时, 算法有较大可能陷入了局部极值. 考虑到随机行为具有一定的盲目性, 算法不能保证总是向更优位置移动, 本文对随机行为引入反向学习机制. 反向学习机制的基本思路是: 当个体远离系统最优解时, 其反向个体接近系统最优解的可能性就会大大增加^[20]. 根据以上思想, 人工鱼群将按式 (8) 进行随机行为. 改进后的随机行为将反向解纳入迭代过程中, 扩大了寻优空间, 从而增加了种群的多样性, 避免了算法早熟收敛.

$$X_{i|next} = (X_j^U + X_j^L) - X_{ij} \quad (8)$$

其中, X_j^U 、 X_j^L 分别为种群搜索空间第 j 维的最大值和最小值.

2.3 差分变异操作

为了提高算法的求解效率和解的质量, 本文结合差分进化算法的变异策略对部分人工鱼进行更新. 具体操作为: 公告牌不再只存放每次迭代的最优值, 而是存放质量前 $100\%m \times N_p$ 个较优值, 其中 N_p 为种群数量. 每次迭代时更新公告牌, 如果公告牌累计 Max_time 次未更新, 则考虑算法可能陷入局部最优, 种群进行随机初始化, 否则, 按照一定比例 θ 选取本次迭代产生的最差 $N_p \times \theta$ 条人工鱼, 并对其按式 (9) 执行变异操作.

$$V_{i,g} = F_1 \cdot (X_{best,g}^p - X_{i,g}) + F_2 \cdot (X_{i,g} - X_{i-1,g}) \quad (9)$$

其中, $X_{best,g}^p$ 为随机选取的本次迭代中质量前 $100\%p \times N_p$ 的人工鱼, F_1 、 F_2 为变异算子, $X_{i-1,g}$ 为经过排序后, $X_{i,g}$ 前面的一条人工鱼.

2.4 ADMAFSA 算法的步骤及流程

ADMAFSA 算法的流程图如图 1 所示.

步骤 1. 随机初始化种群, 种群中人工鱼条数 N_p . 计算每条人工鱼的目标函数值, 将质量前 $100\%m \times N_p$ 个最优值写入公告牌. 设置算法参数初始值: 拥挤度因子 δ , 最大尝试次数 try_number , 视野范围 $visual$, 步长 $step$.

步骤 2. 执行人工鱼的追尾、聚群、觅食行为. 如果连续执行超过 try_number 次觅食行为, 则执行随机行为.

步骤 3. 计算新个体的目标函数值, 对公告牌进行更新.

步骤 4. 判断公告牌未更新的次数, 如果公告牌累计 Max_time 次未更新, 则种群进行随机初始化, 否则按照一定比例选取此次迭代生成的最差 $N_p \times \theta$ 条人工鱼进行变异.

步骤 5. 判断是否达到迭代终止条件, 如果是则算法结束, 输出最优解. 否则更新视野和步长, 返回步骤 2 继续迭代循环.

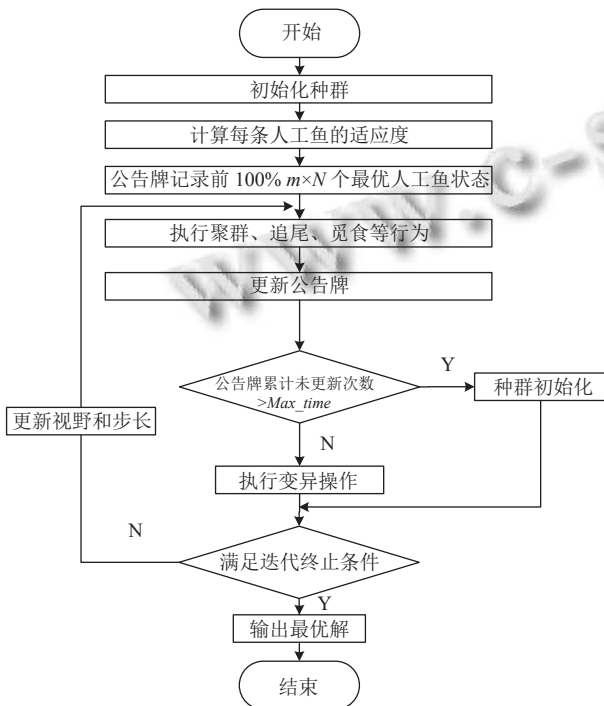


图 1 ADMAFSA 流程图

3 实验仿真与结果分析

3.1 实验环境

为保证算法对比的公平性, 本文所有实验均在 64 位 Windows 10 系统, 处理器为 Intel i5-8265U 1.60 GHz CPU 环境下, 用 Matlab R2020a 实现.

3.2 与其他 AFSA 变体对比

为验证改进算法的有效性, 本文对 ADMAFSA 和 AFSA, 改进的人工鱼群算法 (modified artificial fish swarm algorithm, MAFSA)^[17], 反向自适应高斯变异的人工鱼群算法 (opposite adaptive and gauss mutation artificial fish swarm algorithm, OAGMAFSA)^[18] 进行仿真测试, 采用 6 个经典测试函数进行比较分析. 测试函

数如下.

(1) 单峰函数 Sphere, 其全局最优值为 0, 其中 $x_i \in [-5.12, 5.12]$.

$$f(x) = \sum_{i=1}^D x_i^2$$

(2) 阶跃函数 Step, 其全局最优值为 0, 其中 $x_i \in [-100, 100]$.

$$f(x) = \sum_{i=1}^D [(|x_i + 0.5|)^2]$$

(3) 复杂多峰函数 Rastrigin, 其全局最优值为 0, 其中 $x_i \in [-5.12, 5.12]$.

$$f(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

(4) 复杂多峰函数 Griewank, 其全局最优值为 0, 其中 $x_i \in [-600, 600]$.

$$f(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

(5) 复杂多峰函数 Ackley, 其全局最优值为 0, 其中 $x_i \in [-32, 32]$.

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$$

(6) 复杂多峰函数 Schaffer, 其全局最优值为 0, 其中 $x_i \in [-100, 100]$.

$$f(x) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} \left[\sin^2\left(50(x_i^2 + x_{i+1}^2)^{0.1}\right) + 1 \right]$$

测试算法设置维度 $n=15$, 最大迭代次数 $d_{\max}=2000$, 拥挤因子 $\delta=0.618$, 种群数目 $N_p=100$. 其他参数设置为: 视野最大值 $visual_{\max} = (ub - lb)/4$, 视野最小值 $visual_{\min}=0.001$, 步长最大值 $step_{\max} = visual_{\max}/8$, 步长最小值 $step_{\min} = 0.0002$, $m=0.1$, $\theta=0.05$, $p=0.05$, $Max_time=20$. 对比算法参数设置可参考相关文献. 为了使结果更具准确性, 实验中所有算法均在每个测试函数下独立运行 30 次, 30 次运行结果的最优值、平均值和标准差如表 1 所示, 每个函数的最佳测试结果用黑体表示.

表1 ADMAFSA 与其他 AFSA 变体对比

函数名	指标	AFSA	OAGMAFSA	MAFSA	ADMAFSA
Sphere	Best	4.09E-006	3.87E-013	4.88E-017	0
	Mean	1.38E-005	8.92E-013	2.52E-012	0
	SD	1.38E-005	2.85E-013	4.29E-012	0
Step	Best	0	0	0	0
	Mean	0	0	0	0
	SD	0	0	0	0
Rastrigin	Best	9.98E-001	1.76E-010	2.84E-014	0
	Mean	4.91E+000	6.41E-010	4.04E-010	0
	SD	1.99E+000	5.47E-010	5.61E-010	0
Griewank	Best	1.96E+000	2.05E-013	1.65E-013	0
	Mean	1.16E+001	6.96E-002	3.33E-009	0
	SD	4.76E+000	5.69E-002	5.23E-009	0
Ackley	Best	3.48E-002	1.94E-006	1.39E-007	4.44E-16
	Mean	4.10E-002	2.26E-006	5.54E-006	4.44E-16
	SD	5.95E-002	2.18E-006	4.41E-006	0
Schaffer	Best	8.94E+000	7.60E-003	7.80E-003	0
	Mean	2.68E+001	1.33E-002	5.11E-002	0
	SD	1.11E+000	3.90E-003	3.55E-002	0

根据表1的实验结果可知,与其他几种算法相比,ADMAFSA的收敛精度得到明显提升.对于阶跃函数Step, AFSA、MAFSA、OAGMAFSA和ADMAFSA

均能求出其全局最优解0.在其余5个测试函数的实验中,改进算法ADMAFSA能够准确求出函数Sphere、Rastrigin、Griewank、Schaffer的全局最优值0.其中Sphere为单峰函数,Rastrigin和Griewank为多峰函数,Schaffer为有无数极值点的函数.在求解Ackley函数时,虽未求得全局最优值0,但是相较于其他3种算法,ADMAFSA的最优值、平均值和均差均最小,验证了新算法中引入的机制的有效性.在6个测试函数中,ADMAFSA的均差均为最小且为0,说明ADMAFSA具有较好的鲁棒性.综上所述,ADMAFSA与其他AFSA变体相比,算法的收敛精度与稳定性更高.

为研究ADMAFSA的收敛能力,本文将ADMAFSA与其他几种算法在测试函数迭代过程中的收敛情况进行比较.图2为AFSA、OAGMAFSA、MAFSA和ADMAFSA迭代2000次的对比曲线图.可以看出,ADMAFSA能够迅速收敛,在进化500代之内就能找到全局最优位置.由此可见,改进算法通过引入变异机制,能够不断调整和改进劣质解,从而提高了搜索效率.

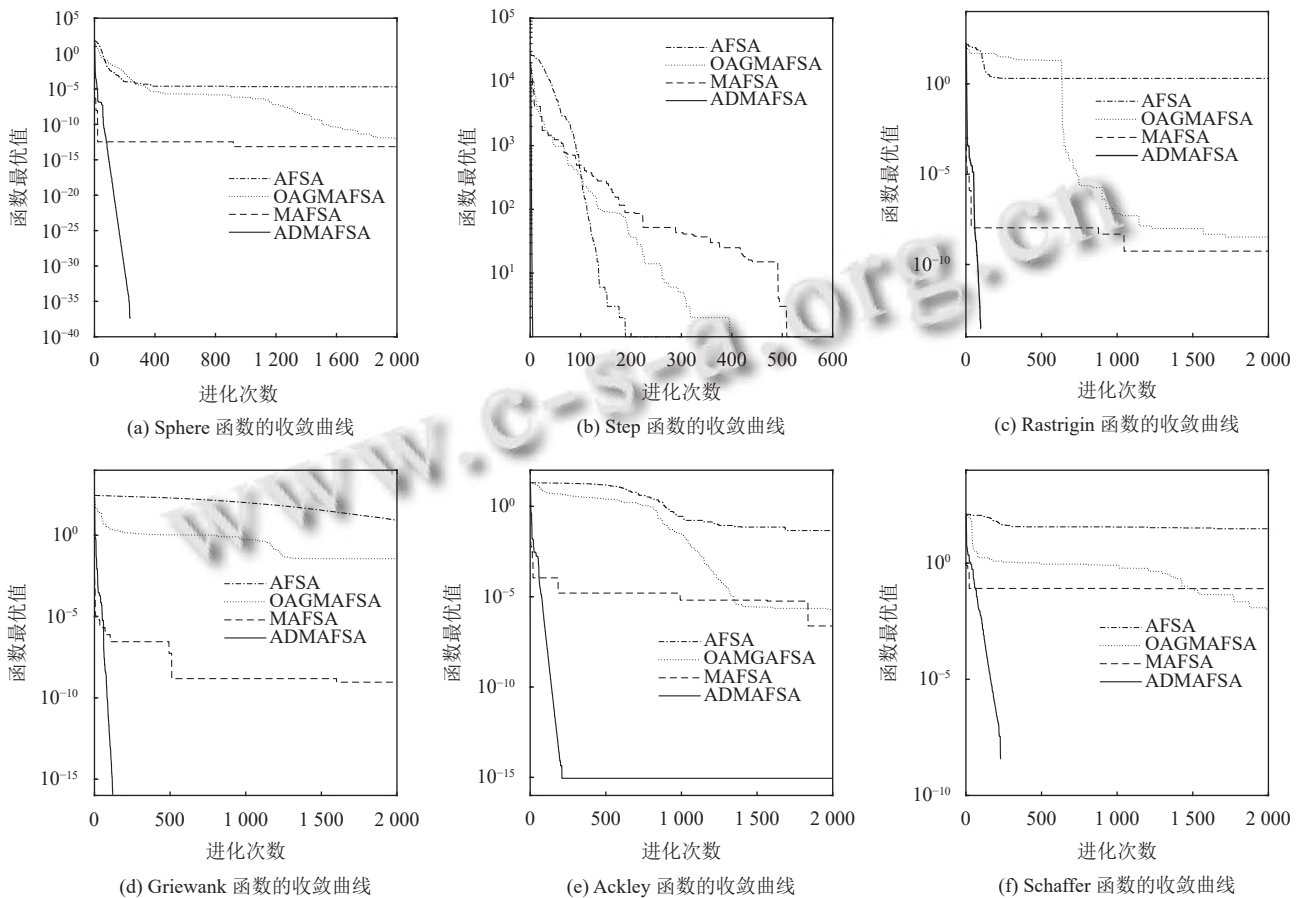


图2 收敛曲线对比结果

由于 AFSA 在进化过程中保持视野和步长不变,所以在迭代后期不能保证收敛精度.且算法没有设置陷入局部最优解的解决方法,导致优化过程中出现搜索停滞现象. OAGMAFSA 过于依赖反向解,当陷入局部极值时,如果其反向解的质量也较差,则跳出该区域具有一定的困难. MAFSA 引入可变视野,在迭代后期,算法有足够的潜力进行精细寻优.并且调整移动策略为:当发现比自身当前状态更好的解时,直接移动到该位置,不再进行随机移动,促进了鱼群向极值点位置快速收敛.另外,MAFSA 按照一定的概率进行变异,相较于 AFSA,一定程度避免了算法早熟.然而该算法设置固定的步长,不利于算法协调全局搜索和局部搜索.相比较而言,ADMAFSA 采用的变异机制、反向学习策略、自适应视野和步长方法能够使算法稳定且快速收敛,波动性较小.此外,ADMAFSA 设置算法陷入局部最优的警戒值,一旦算法达到该警戒值,便会触发种群随机初始化操作,对改善算法性能起到了良好的效果.

3.3 与其他智能算法对比

为进一步测试本文所提算法的优化效果,将 ADMAFSA 与麻雀优化算法 SSA、灰狼优化算法 GWO、蜣螂优化算法 DBO 这 3 种新型智能优化算法进行比

较.实验采用 8 个 CEC2019 测试函数进行评估,函数具体信息见表 2.所有测试算法设置种群数量 30,最大迭代次数设为 500.30 次的独立运行结果如表 3 所示,每个函数优化结果的最佳平均值用黑体表示.根据实验结果可知,ADMAFSA 在大部分基准测试函数上都优于其他 3 种算法或者基本相当,仅在函数 f_1 上,略低于 SSA 算法.此外,问题求解维度的增加对于 ADMAFSA 的优化精度和稳定性影响较小.对于单峰测试函数 f_1 - f_4 以及多峰函数 f_5 和 f_8 ,ADMAFSA 算法均能求得其全局最优值 0,并且 30 次运行结果的标准差都为 0.从而验证了改进算法不仅在求解精度方面更为突出,稳定性上也具有一定的优势.

表 2 CEC2019 测试函数

Number	CEC2019 function	Global minimum
f_1	Bent cigar function	0
f_2	Sum of different power functions	0
f_3	Zakharov function	0
f_4	Discus function	0
f_5	Rastrigin's function	0
f_6	Ackley's function	0
f_7	Levy function	0
f_8	Griewank's function	0

表 3 ADMAFSA 与其他智能算法对比

函数	算法	最优值		平均值		标准差	
		15维	50维	15维	50维	15维	50维
f_1	SSA	8.15E-312	3.45E-204	2.60E-48	2.48E-76	1.43E-47	1.36E-75
	GWO	1.29E-89	1.73E-41	5.93E-86	5.97E-40	2.69E-85	1.32E-39
	DBO	2.53E-299	1.12E-292	3.34E-218	7.17E-213	0	0
	ADMAFSA	0	0	0	0	0	0
f_2	SSA	0	4.14E-198	2.88E-81	3.03E-72	1.55E-80	1.66E-71
	GWO	5.58E-233	7.34E-205	7.99E-218	5.10E-185	0	0
	DBO	2.15E-313	2.78E-309	1.85E-224	2.29E-205	0	0
	ADMAFSA	0	0	0	0	0	0
f_3	SSA	0	0	1.22E-66	1.12E-47	6.62E-66	5.22E-47
	GWO	9.59E-49	1.67E-11	5.80E-45	7.60E-08	1.50E-44	2.62E-07
	DBO	1.06E-266	1.21E-115	1.46E-141	9.62E-32	7.99E-141	5.27E-31
	ADMAFSA	0	0	0	0	0	0
f_4	SSA	0	0	1.25E-36	2.56E-35	6.85E-36	1.40E-34
	GWO	2.68E-96	1.08E-47	5.25E-91	5.64E-46	1.76E-90	8.34E-46
	DBO	0	5.52E-312	6.83E-229	7.10E-220	0	0
	ADMAFSA	0	0	0	0	0	0
f_5	SSA	0	0	0	0	0	0
	GWO	0	0	0	3.96E-01	0	1.28E+00
	DBO	0	0	0	0	0	0
	ADMAFSA	0	0	0	0	0	0
f_6	SSA	8.88E-16	8.88E-16	8.88E-16	8.88E-16	0	0
	GWO	2.03E+01	2.09E+01	2.04E+01	2.11E+01	1.11E-01	6.61E-02
	DBO	8.88E-16	8.8818E-16	1.72E-15	1.33E+00	1.53E-15	5.07E+00
	ADMAFSA	8.88E-16	8.88E-16	8.88E-16	8.88E-16	0	0

表3 ADMAFSA 与其他智能算法对比 (续)

函数	算法	最优值		平均值		标准差	
		15维	50维	15维	50维	15维	50维
f_7	SSA	4.42E-24	3.54E-12	1.99E-19	5.33E-09	6.17E-19	1.16E-08
	GWO	1.79E-01	1.91E+00	3.19E-01	2.58E+00	1.12E-01	3.12E-01
	DBO	1.37E-28	3.59E-01	2.03E-01	1.01E+00	1.50E-01	3.47E-01
	ADMAFSA	1.97E-10	1.25E-09	3.72E-07	1.06E-06	6.45E-07	1.71E-06
f_8	SSA	0	0	0	0	0	0
	GWO	0	0	3.80E-03	2.70E-03	8.60E-03	5.80E-03
	DBO	0	0	0	0	0	0
	ADMAFSA	0	0	0	0	0	0

3.4 齿轮系设计问题

在本节中采用齿轮系设计问题,通过与 SSA、GWO、DBO、WOA、SCSO 对比最佳寻优结果,检验 ADMAFSA 算法在求解工程设计问题的优化性能。

齿轮设计问题的优化目标是通过合理设计 4 种齿轮的齿数,使得转动比成本最小化,通常被描述为带有不等式约束的优化模型^[21]。如式 (10) 所示。其中 T_a 、 T_b 、 T_d 、 T_f 分别代表 4 个不同齿轮的齿数。

$$\begin{cases} x_i = [T_a, T_b, T_d, T_f] \\ \min f(x) = \left(\frac{1}{6.931} - \frac{T_b \cdot T_d}{T_a \cdot T_f} \right)^2 \\ x_i \in N_+, \quad i = 1, 2, 3, 4 \\ 12 \leq x_i \leq 60, \quad i = 1, 2, 3, 4 \end{cases} \quad (10)$$

表 4 为 6 种算法各独立运行 30 次的最优解及对应的各参数取值,其中黑体表示本文算法的优化结果。实验设置各算法最大迭代次数 100 次,种群数量为 30。ADMAFSA 求解的最优设计方案中,目标函数最小值为 2.7009×10^{-12} ,最优设计参数为 [43, 16, 19, 49]。相比于其他 5 种算法,本文所提算法求解精度更高。因此,ADMAFSA 算法适用于求解带约束条件的工程设计问题。

表 4 齿轮系设计问题优化结果

算法	参数设计				最优值
	T_a	T_b	T_d	T_f	
SSA	43	19	16	49	9.9216×10^{-10}
GWO	53	15	26	51	2.3078×10^{-11}
DBO	60	34	14	55	1.3616×10^{-9}
WOA	51	15	26	53	2.3078×10^{-11}
SCSO	57	12	37	54	8.8876×10^{-10}
ADMAFSA	43	19	16	49	2.7009×10^{-12}

4 结论与展望

针对 AFSA 算法收敛精度低,易陷入局部最优等

问题,本文使用可变视野和步长,动态地调整寻优范围,平衡了全局搜索和精细寻优的能力;引入反向学习机制,将反向解纳入迭代过程中,扩大了寻优空间,从而避免了算法陷入局部极值;并且采用差分变异策略,使质量较差的鱼群学习历史最优解,降低了算法早熟收敛的可能性,最终提出了 ADMAFSA。为验证 ADMAFSA 的有效性和可行性,对 6 个基准测试函数和 8 个 CEC2019 函数进行仿真测试,实验结果表明,ADMAFSA 能够摆脱局部极值的限制,在兼顾算法的稳定性和准确性的前提下,提高算法的寻优精度和搜索效率。最后将改进算法应用于齿轮系设计问题,进一步验证了 ADMAFSA 算法有较强的收敛能力,在解决工程实际问题中也具有良好的潜力。未来的工作是将改进算法应用于求解多目标工程问题。

参考文献

- 张海林, 陈泯融. 基于混合策略的改进哈里斯鹰优化算法. 计算机系统应用, 2023, 32(1): 166-178. [doi: 10.15888/j.cnki.csa.008911]
- Krishna AB, Saxena S, Kamboj VK. A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired Harris hawks optimizer. Neural Computing and Applications, 2021, 33(12): 7031-7072. [doi: 10.1007/S00521-020-05475-5]
- Aydın D, Özcan Y, Sulaiman M, et al. Elitist artificial bee colony with dynamic population size for multimodal optimization problems. Swarm Intelligence, 2023, 17(4): 305-334. [doi: 10.1007/S11721-023-00228-1]
- 刘小宁, 魏霞, 谢丽蓉. 混合哈里斯鹰算法求解作业车间调度问题. 计算机应用研究, 2022, 39(6): 1673-1677. [doi: 10.19734/j.issn.1001-3695.2021.11.0640]
- 周明月, 周明伟, 刘桂岐, 等. 基于改进蝴蝶算法的机械臂时间最优轨迹规划. 计算机科学, 2023, 50(S2): 220900284. [doi: 10.11896/jsjcx.220900284]

- 6 Qu CW, Lu ZH, Lu FJ. Learning search algorithm to solve real-world optimization problems and parameter extract of photovoltaic models. *Journal of Computational Electronics*, 2023, 22(6): 1647–1688. [doi: [10.1007/S10825-023-02095-9](https://doi.org/10.1007/S10825-023-02095-9)]
- 7 Hemici M, Zouache D. A multi-population evolutionary algorithm for multi-objective constrained portfolio optimization problem. *Artificial Intelligence Review*, 2023, 56(S3): 3299–3340. [doi: [10.1007/S10462-023-10604-2](https://doi.org/10.1007/S10462-023-10604-2)]
- 8 Mahapatra AK, Panda N, Pattanayak BK. An improved pathfinder algorithm (ASDR-PFA) based on adaptation of search dimensional ratio for solving global optimization problems and optimal feature selection. *Progress in Artificial Intelligence*, 2023, 12(4): 323–348. [doi: [10.1007/S13748-023-00306-9](https://doi.org/10.1007/S13748-023-00306-9)]
- 9 Liu YT, Ding HW, Wang ZS, *et al.* A chaos-based adaptive equilibrium optimizer algorithm for solving global optimization problems. *Mathematical Biosciences and Engineering*, 2023, 20(9): 17242–17271. [doi: [10.3934/MBE.2023768](https://doi.org/10.3934/MBE.2023768)]
- 10 薛建凯. 一种新型的群智能优化技术的研究与应用——麻雀搜索算法 [硕士学位论文]. 上海: 东华大学, 2019.
- 11 Xue JK, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing*, 2023, 79(7): 7305–7336. [doi: [10.1007/s11227-022-04959-6](https://doi.org/10.1007/s11227-022-04959-6)]
- 12 Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46–61. [doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007)]
- 13 Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, 95: 51–67. [doi: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008)]
- 14 Heidari AA, Mirjalili S, Faris H, *et al.* Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 2019, 97: 849–872. [doi: [10.1016/j.future.2019.02.028](https://doi.org/10.1016/j.future.2019.02.028)]
- 15 Seyyedabbasi A, Kiani F. Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Engineering with Computers*, 2023, 39(4): 2627–2651. [doi: [10.1007/s00366-022-01604-x](https://doi.org/10.1007/s00366-022-01604-x)]
- 16 李晓磊. 一种新型的智能优化方法——人工鱼群算法 [博士学位论文]. 杭州: 浙江大学, 2003.
- 17 张朝炜, 柳云祥, 朱永利. 基于改进人工鱼群算法的大规模多目标机组组合优化. *电力系统保护与控制*, 2021, 49(8): 100–108. [doi: [10.19783/j.cnki.pspc.200811](https://doi.org/10.19783/j.cnki.pspc.200811)]
- 18 姚凌波, 戴月明, 王艳. 反向自适应高斯变异的人工鱼群算法. *计算机工程与应用*, 2018, 54(1): 179–185. [doi: [10.3778/j.issn.1002-8331.1607-0192](https://doi.org/10.3778/j.issn.1002-8331.1607-0192)]
- 19 王辉, 阿迪娜. 改进鱼群算法在模拟机运动洗出优化中的应用. *计算机系统应用*, 2022, 31(8): 265–272. [doi: [10.15888/j.cnki.csa.008633](https://doi.org/10.15888/j.cnki.csa.008633)]
- 20 Chauhan S, Vashishtha G, Abualigah L, *et al.* Boosting salp swarm algorithm by opposition-based learning concept and sine cosine algorithm for engineering design problems. *Soft Computing*, 2023, 27(24): 18775–18802. [doi: [10.1007/S00500-023-09147-Z](https://doi.org/10.1007/S00500-023-09147-Z)]
- 21 奚金明, 郑荣艳. 基于自适应权重和莱维飞行的改进海鸥优化算法. *计算机系统应用*, 2023, 32(12): 171–179. [doi: [10.15888/j.cnki.csa.009326](https://doi.org/10.15888/j.cnki.csa.009326)]

(校对责编: 张重毅)