

北京航空航天大学学报

*Journal of Beijing University of Aeronautics and Astronautics*

ISSN 1001-5965, CN 11-2625/V

## 《北京航空航天大学学报》网络首发论文

题目: 融合多策略改进的侏儒猫鼬算法  
作者: 于明洋, 李婷, 许静  
DOI: 10.13700/j.bh.1001-5965.2023-0613  
收稿日期: 2023-09-26  
网络首发日期: 2024-01-19  
引用格式: 于明洋, 李婷, 许静. 融合多策略改进的侏儒猫鼬算法[J/OL]. 北京航空航天大学学报. <https://doi.org/10.13700/j.bh.1001-5965.2023-0613>



**网络首发:** 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

**出版确认:** 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

## 融合多策略改进的侏儒猫鼬算法

于明洋<sup>1</sup>, 李婷<sup>2,3✉</sup>, 许静<sup>1</sup>

(1. 南开大学 人工智能学院, 天津 300350; 2. 南开大学 计算机学院, 天津 300350;

3. 天津津航技术物理研究所, 天津 300350)

\*通信作者 E-mail: t24725@126.com

**摘要** 针对侏儒猫鼬优化算法(Dwarf Mongoose Optimization Algorithm, DMO)易陷入局部最优和收敛效率低的问题, 提出一种多策略融合的增强型侏儒猫鼬算法(Enhanced Multi-Strategy Dwarf Mongoose Optimization Algorithm, EDMO)。该算法引入随机反向学习策略增强了猫鼬种群的多样性和质量, 以增强其全局搜索能力和提高收敛的速度。同时, 采用自适应的方式更新保姆交换系数, 以平衡全局探索与局部开发的需求。在迭代的后期, 算法利用黏菌觅食行为, 在局部与全局最优解之间进行优化。通过对 CEC2017 测试函数集的求解, 对不同的算法进行了比较。结果表明, 融合三种策略的 EDMO 在寻优精度、寻优速度和鲁棒性方面均优于对比的先进算法。通过对无人机三维路径规划实验验证, EDMO 在局部搜索方面表现优于原始 DMO 算法, 同时生成的飞行路径也更为稳定。

**关键词** 侏儒猫鼬优化算法; 多策略融合; 随机反向学习; 自适应; 黏菌觅食行为; 三维路径规划

中图分类号 V221+.3; TB553

文献标识码: A

DOI: 10.13700/j.bh.1001-5965.2023-0613

## Enhanced Dwarf Mongoose Optimization Algorithm with Multi-Strategy fusion

YU Mingyang<sup>1</sup>, LI Ting<sup>2,3✉</sup>, XU Jing<sup>1</sup>

(1. College of Artificial Intelligence, Nankai University, Tianjin, 300350, China;

2. College of Computer Science, Nankai University, Tianjin, 300350, China;

3. Tianjin Jinhang Institute of Technical Physics, Tianjin, 300350, China)

\*E-mail: t24725@126.com

**Abstract** Aiming at solving the problems in the Dwarf Mongoose Optimization Algorithm (DMO) being susceptible to local optima and its low convergence efficiency, An Enhanced Multi-Strategy Dwarf Mongoose Optimization Algorithm (EDMO) is proposed. This algorithm employs a random opposite learning strategy to amplify the diversity and quality of the mongoose population, bolstering its global search capability and enhancing convergence accuracy. Concurrently, an adaptive approach is deployed to update the babysitter exchange coefficient, striking a balance between global exploration and local exploitation. In the latter stages of iteration, the algorithm capitalizes on the foraging behavior of the slime mold, optimizing between local and global optimal solutions. By solving CEC2017 test function set, different algorithms are compared. The results show that EDMO, which fuses the three strategies, outperforms the compared advanced algorithms in terms of optimization accuracy, optimization speed and robustness. Through the experimental verification of UAV three-dimensional path planning, EDMO algorithm performs better than the original DMO algorithm in local search, and the flight path generated is more stable.

**Key words** Dwarf Mongoose Optimization Algorithm; Multi-Strategy; Random reverse learning; Adaptive; Fungal foraging behavior; Three-dimensional path planning

伴随工程和理论领域中的优化问题不断受到关注, 群体智能优化算法因其简明、高效且易于执行的特性, 已逐渐跻身研究热点<sup>[1-2]</sup>。近期提及的群体智能优化算法如: 樽海鞘群优化算法(SSA)<sup>[3]</sup>、蛇优化算法(SO)<sup>[4]</sup>、鲫鱼优化算法(ROA)<sup>[5]</sup>, 以及黏菌优化算法(SMA)<sup>[6]</sup>等, 在图像分割<sup>[7]</sup>及路径规划<sup>[8]</sup>

收稿日期: 2023-09-26

基金项目: 天津市自然科学基金(项目号 21JCYBJC00110)

Fund: Natural Science Foundation of Tianjin Municipality (21JCYBJC00110).

网络首发时间: 2024-01-19 08:48:30 网络首发地址: <https://link.cnki.net/urlid/11.2625.v.20240118.1037.003>

等众多领域中已展现出其良好应用,并获得了令人瞩目的效果<sup>[9]</sup>。

侏儒猫鼬优化算法(Dwarf Mongoose Optimization Algorithm, DMO)是基于模拟猫鼬寻找食物及探索睡眠丘休息行为而设计的一种群体智能优化方法,由 Jeffrey O. Agushaka 等于 2022 年首次提出<sup>[10]</sup>。DMO 以其简单的结构、较少的参数和出色的勘探与探索能力而著称,广泛应用于高维特征选择、数据聚类等领域的优化问题求解<sup>[11]</sup>。然而,尽管如此,DMO 算法仍然存在一些缺点,包括收敛速度较慢和容易陷入局部最优解等问题。Agushaka, J.O.等针对 DMO 算法的缓慢收敛问题提出了 ADMO 算法,通过引入捕食、土丘保护、繁殖和群体分裂行为,增强了 DMO 算法的优化性能<sup>[12]</sup>。随后,Agushaka, J.O.等进一步改进了 DMO 算法,特别是在解决约束工程设计问题方面,显著提高了 DMO 算法的勘探和开发能力<sup>[13]</sup>。此外,Akinola, OA 等在将 DMO 与模拟退火(SA)算法相结合,形成了一种二进制变体,增强了 DMO 算法的有限利用机制<sup>[14]</sup>。为了提高 DMO 算法的性能,贾鹤鸣等提出了精英池侏儒猫鼬优化算法(IDMO),利用透镜成像反向学习,加速了 DMO 算法的收敛速度,增强了其探索能力<sup>[15]</sup>。而赵世杰等则于提出了一种多目标侏儒猫鼬优化算法,该算法基于领导者引导与支配解动态缩减进化,用于挖掘多目标问题的潜在前沿,并提高了算法的多样性<sup>[16]</sup>。此外,Al-Shourbaji, I 等将人工生态优化与 DMO 相结合,开发了一种高效的特征选择算法<sup>[17]</sup>。Fu, SW 等也于对 DMO 的搜索过程进行了改进,引入非线性控制和探索策略,提高了收敛速度,同时保持了高精度<sup>[18]</sup>。Rizk-Allah, RM 等则通过结合 Nelder-Mead 单纯形法和生成不同解向量的方法,有效地改进了 DMO 算法,以在迭代过程中寻找高质量的解,并成功将其应用于未知参数的识别问题<sup>[19]</sup>。罗淑媛等将 DMO 算法与学习策略相结合,显著提升了算法跳出局部最优的能力,并将改进后的算法应用于特征选择问题<sup>[20]</sup>。

为了提高 DMO 算法的收敛速度,增强其探索能力,本文提出一种融合多策略改进的 DMO 算法,即 EDMO。首先,基于随机反向学习策略在每次迭代前期更新侏儒猫鼬 Alpha 组种群,提升了算法的收敛能力和鲁棒性;其次,提出自适应更新保姆交换参数来兼顾全局搜索和局部优化;最后,在迭代后期引入黏菌觅食机制,有利于算法后期跳出局部最优,同时扩大了算法的搜索范围。基于 29 个 CEC2017 基准函数的实验结果表明,EDMO 算法显著提高了全局优化水平,有效地提高了算法的收敛速度和精度。通过对无人机三维路径规划问题的求解,验证 EDMO 算法对工程问题同样具有较高的适用性。

## 1 侏儒猫鼬算法

DMO 算法是模拟侏儒猫鼬半游牧式生活的一种群体优化算法。侏儒猫鼬群体按照职能分为 Alpha 组、Scout 组、Babysitter 组。Alpha 组负责觅食和指导觅食方向。当达到保姆交换标准时,Alpha 组和 Babysitter 组的成员进行交换,并由 Scout 组开始负责寻找睡眠丘,以保持群体的生存平衡。Babysitter 组的数量会影响算法的行为,从而对算法的性能产生影响。此外,Babysitter 组还负责调整一些参数,这些参数的变化将改变算法的行为,以适应不同的问题和条件。

### 1.1 Alpha 组

每个成员通过式(1)进行初始化,初始化完成后,通过式(2)计算每个猫鼬的种群适应度概率值,以此选择 Alpha 组首领  $\alpha$ 。

$$x_{i,j} = \text{unfrnd}(lb, ub, Dim) \quad (1)$$

$$a = \frac{fit_i}{\sum_{i=1}^N fit_i} \quad (2)$$

式中:  $x_{i,j}$  为初始位置,  $\text{unfrnd}$  为一个均匀分布的随机数,  $ub$  和  $lb$  为问题的上下边界,  $Dim$  为决策变量维度,  $fit_i$  为第  $i$  个个体的适应度,  $N$  为种群规模。设置保姆的数量为  $bs$ , 则 Alpha 组个体数量  $n$  为  $N-bs$ , 觅食路径由  $\alpha$  选择并受到  $peep$  的影响, 食物源新位置建模如式(3)所示:

$$x_{i+1} = x_i + \text{phi} \times \text{peep} \times (x_i - x_{rand}) \quad (3)$$

式中： $x_{i+1}$  为找到的食物源新位置， $x_i$  为雌性首领的当前位置， $phi$  为均匀分布在 $[-1,1]$ 之间的随机数，根据文献[10]本文中  $peep$  选取为 2，初始的睡眠土堆设为 $\emptyset$ ， $x_{rand}$  为 Alpha 组中的随机个体的位置。评估  $x_{i+1}$  的阿尔法组适应度  $fit_{i+1}$ ，并根据式(4)计算睡眠丘值  $sm_i$ 。

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}} \quad (4)$$

根据式(5)(6)计算睡眠丘的平均值  $\varphi$  和决定猫鼬移动到新睡眠丘的方向向量  $\vec{M}$ 。

$$\varphi = \frac{\sum_{i=1}^n sm_i}{n} \quad (5)$$

$$\vec{M} = \sum_{i=1}^n \frac{X_i \times sm_i}{X_i} \quad (6)$$

当满足保姆交换条件  $C \geq L$  时，将交换 Alpha 组和 Babysitter 组的成员，重新初始化保姆组  $bs$  的位置并计算适应度值，并激活侦察阶段。其中  $C$  为时间计数器， $L$  为保姆交换参数。

### 1.2 Scout 组

当 Alpha 组找到充分的食物并接收到保姆的交换信号时，Scout 组启动并开始寻找新的睡眠丘。如果它们探索到了足够远的位置，那么可能会发现一个新的睡眠丘。根据式(7)、(8)模拟寻找新的睡眠丘的位置  $x_{sm}$ 。

$$CF = (1 - \frac{t}{T})^{\frac{(2^t)}{T}} \quad (7)$$

$$x_{sm} = \begin{cases} x_i - CF * phi * rand * [x_i - \vec{M}] & \text{if } \varphi_{i+1} > \varphi \\ x_i + CF * phi * rand * [x_i - \vec{M}] & \text{else} \end{cases} \quad (8)$$

式中： $rand$  为 0 到 1 之间的随机数， $CF$  为猫鼬种群移动能力的参数，它会随着迭代次数线性递减， $t$  为当前迭代次数， $T$  为最大迭代次数。

### 1.3 Babysitter 组

Babysitter 组通常由次级群体成员组成，负责照顾幼崽，并定期轮换，以便雌性首领  $\alpha$  能够领导其他群体成员进行日常觅食。Babysitter 组的成员数量取决于种群的大小，影响算法的一种方法是通过改变种群规模的比例。对 Babysitter 组交换参数  $L$  的调整允许重新整理和利用其他家族成员之前收集的侦查和食物来源数据。将 Babysitter 组的适应度值设定为零有助于降低下一轮迭代中 Alpha 组的平均权重，进而抑制整个群体的活动范围，这反映了现实中动物群体因照看幼崽而活动受限的情景。

由上述寻优过程可知：受种群个体属性及位置更新因子制约，该算法重点通过个体间的互动和跟随来刷新解决方案。但由于缺乏变异策略，当其进入局部最优状态时，跳出当前局部搜索范围变得困难。

## 2 改进的侏儒猫鼬算法

考虑到前述分析，我们对 DMO 算法从三个角度进行改进：1) 结合随机反向学习策略，拓宽算法的全局探索范围，并提升避免局部最优和收敛速率；2) 设计非线性比例因子，自适应更新保姆交换参数，达到全局探索与局部寻优的均衡；3) 加入黏菌觅食策略，增强侏儒猫鼬种群跨越局部最优的能力。

### 2.1 随机反向学习策略

现有算法普遍采用伪随机数初始化候选解，这种配置方式可最大程度上增强算法的全局性能，然而，强随机性使得算法无法保持稳定的目标优化精度；此外，依赖伪随机数的初始方式也易导致种群遍历性不足，种群多样性将下降。为了增强探索性能并提升种群的多样性水准<sup>[21]</sup>，Long 等研究者基于反向学习<sup>[22]</sup>推出了随机反向学习策略(ROL)<sup>[23]</sup>。这一方法对反向解加入了随机成分，有力地提高了



搜索空间的种群多样性, 有效地规避了局部最优解的问题。计算公式如下:

$$\tilde{X}_{rand} = lb + ub - phi * X \quad (9)$$

式中:  $lb$ 、 $ub$  分别为解空间的下界和上界,  $X$  和  $\tilde{X}_{rand}$  分别为当前解和随机反向解的位置,  $phi$  为  $[0,1]$ 之间的随机数。

在本文中, 每次迭代前期时使用 ROL 策略对每轮迭代的 Alpha 种群个体进行反向扰动, 刺激侏儒猫鼬 Alpha 组个体在更广的范围中实现全局搜索并拥有跳出局部最优的能力, 同时这种策略能为 DMO 积累更加多样化和高质量的搜索经验, 从而极大地增强算法的全局性能。

## 2.2 非线性比例因子

在 DMO 中, Alpha 组引导种群开发, 而 Babysitter 组作为 Alpha 组的储备, 与其进行个体交换。Babysitter 组的规模及交换参数直接种群间信息交流效率。当交换能力不足时, 种群开发则始终受 Alpha 组牵引, 根据式(2)描述, 若 Alpha 组个体普遍陷入局部最优, 雌性首领的引导有可能导致次级群体盲目聚集。此外, 若 Babysitter 组作为 Alpha 组交流过于频繁, 则易导致 Alpha 组的搜索无法兼顾全局性和局部性。标准的 DMO 中, Babysitter 组的规模为初始设定的常值, 这种固定赋值的设定模式显然无法赋能 DMO 的搜索平衡。对此, 本文提出一种基于自适应因子的 Alpha 组参数更新方式, 具体如下:

$$\alpha = 0.4 - 0.2 \times [(e^{t/T} - 1) / (e - 1)]^k \quad (10)$$

式中:  $T$  和  $k$  分别为 500、2。 $\alpha$  为 Babysitter 组的非线性比例因子, 该在搜索前中期取值较大, 有利于 Babysitter 组和 Alpha 组信息交流, 为后续搜索保存历史经验; 寻优后半段, 该值迅速衰减到较低水平, 能够加速算法快速收敛。非线性比例因子  $\alpha$  可以根据算法的当前状态和问题的特性来动态调整阈值  $L$ 。这意味着在优化过程中,  $L$  的值可以根据需要进行适度的增加或减少, 以更好地适应问题的性质。这有助于加速算法的收敛速度, 特别是在接近最优解时, 可以减小  $L$  以增加局部搜索的精度;  $\alpha$  根据问题的复杂性和不同阶段的搜索需求来自适应地调整  $L$ , 有助于防止算法陷入局部最优解, 即使在面对复杂的问题时也能够更好地保持全局搜索的能力, 增强算法的鲁棒性。

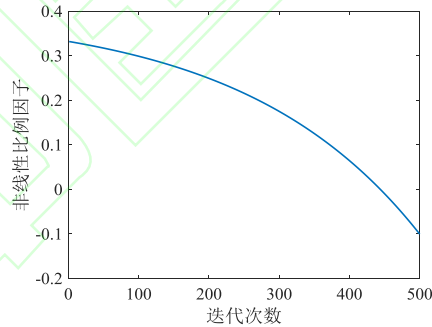


图1 非线性比例因子  $\alpha$   
Fig.1 Nonlinear scaling factor  $\alpha$

## 2.3 黏菌变异机制

对于低维测试函数, 上述策略的引入在一定程度上平衡了全局勘探和局部开发过程, 然而, 当应用于大规模复杂优化问题时, 因 DMO 搜索模式单一, 易导致侏儒猫鼬种群陷入局部极值陷阱。因此, 丰富随机个体的搜索模式是改进启发式算法全局搜索性能的另一个重要途径。黏菌算法(slime mould algorithm, SMA)是受黏菌觅食行为启发而提出的一种优化算法<sup>[24]</sup>。黏菌群体向食物源蠕动并将其覆盖的过程是黏菌算法的核心, 也是其全局搜索能力优秀的主要原因<sup>[25-26]</sup>。对此, 效仿黏菌觅食过程, 本文将其引入以补充 DMO 的搜索模式, 该阶段猫鼬-黏菌位置的数学公式为:

$$X(t+1) = \begin{cases} rand \times (ub - lb) + lb & r < z \\ X_b(t) + vb(W \times X_A(t) - X_B(t)) & r < p \\ vc \times X(t) & r \geq p \end{cases} \quad (11)$$

式中： $rand$  和  $r$  都为在 0 至 1 范围内产生的随机数； $ub$  与  $lb$  分别为搜索空间的上限和下限； $z$  为用于在搜索和开发阶段之间达到平衡的参数， $X(t+1)$  为 Alpha 组个体更新后的位置， $X_b(t)$  为历史最优解位置， $vb$  和  $vc$  是基于随机数的系数，用于控制位置更新的幅度和方向。根据文献[24]，本研究将  $z$  设定为 0.03。

#### 2.4 EDMO 算法描述

在改进的侏儒猫鼬算法 EDMO 中，个体的位置更新不仅依赖于其 Alpha 组、Scout 组或 Babysitter 组的运动方式，还受到群体中其他个体的影响和协作，以实现更好的全局搜索性能。为了提高单个体在避免陷入局部最优解时的效能，算法在位置更新之后采用了模拟黏菌觅食模式的策略。这一策略是通过应用公式(12)来实现的，目的是探索更优的位置点，从而达到更高的适应度评价。这种方法借鉴了黏菌在自然环境中寻找食物的行为，以提高算法在搜索过程中的效率和有效性。EDMO 具体流程如图 2 所示，执行步骤如下：

- a) 根据式(1)初始化侏儒猫鼬算法参数，设定种群规模  $N$ 、保姆的数量  $bs$ 、Alpha 组个体数量  $n$ 、保姆交换参数  $L$ 、最大迭代次数  $T$ 、空间搜索维度  $Dim$  等相关的开发参数；
- b) 根据式(9)使用 ROL 策略更新 Alpha 组个体，计算种群个体适应度，设置时间计数器  $C$ ，根据式(2)选择 Alpha 组首领  $\alpha$ ；
- c) 根据式(3)产生候选食物位置，根据式(4)计算当前睡眠丘值；
- d) 根据式(5)、(6)计算睡眠丘的平均值  $\varphi$ ，并确定猫鼬种群向新睡眠丘迁移的方向矢量  $\vec{M}$ ；
- e) 根据式(10)自适应更新保姆交换参数  $L$ ，当满足保姆交换条件  $C \geq L$  时，交换 Alpha 组和 Babysitter 组的成员，重新初始化 Babysitter 组  $bs$  的位置并计算适应度值，根据式(7)、(8)寻找新的睡眠丘的位置  $X_{sm}$ ；
- f) 根据式(11)进一步寻求 Alpha 组首领  $\alpha$  的最优解；
- g) 判断是否达到指定的停止条件(即是否满足  $t \geq M$ )，若没有，重复步骤 b 至 f。否则输出最优结果：最后得到的雌性首领的位置即为最优解，对应的适应度值为最优解的优劣程度；

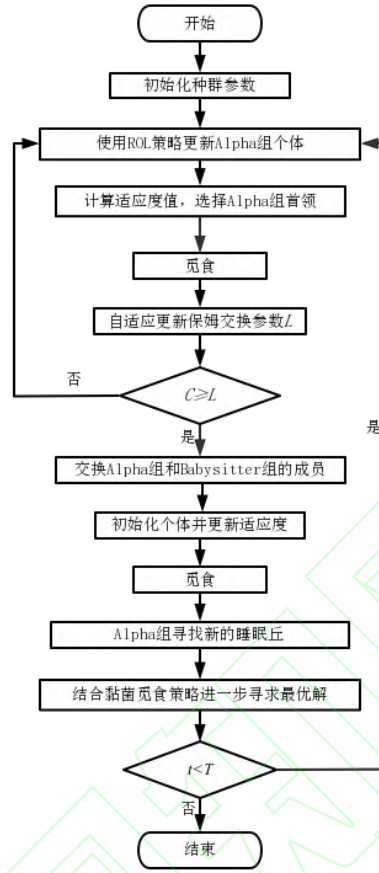


图2 EDMO 流程  
Fig.2 Flow chart of EDMO

### 3 算法性能测试与分析

本研究的仿真环境运行在 Windows10 64 位操作系统上, CPU 型号为 AMD Ryzen 74800H, 主频为 2.30GHz, 配备 16GB RAM。所用算法在 Matlab2018b 平台上进行实现。

#### 3.1 测试函数及参数设置

为验证本文所提出的 EDMO 的有效性, 采用 CEC2017 测试函数集对其进行测试。其中, CEC 系列函数是由多种基本测试函数混合构成, 不仅常被用作比较不同优化算法性能的基准, 而且广泛应用于模拟实际问题的复杂性。上述函数的基本信息如表 1 所示。其中, 29 个 CEC2017 测试函数是由多种基本测试函数混合构成, 其中  $f_1$ 、 $f_3$  为单峰函数,  $f_4 \sim f_{10}$  为多峰函数,  $f_{11} \sim f_{20}$  为混合函数,  $f_{21} \sim f_{30}$  为复合函数,  $f_2$  函数由于高维的不稳定性已被剔除。CEC2017 测试函数集的搜索范围均为  $[-100, 100]^D$ 。

实验中, 设置算法的种群数量  $N$  为 50, 最大迭代次数  $T$  为 500, 空间搜索维度  $Dim$  为 30, 分别独立运行 30 次并记录每组实验的最优适应度值。

表 1 29个CEC2017测试函数  
Table 1 29 CEC2017 test functions

函数	函数名称	最值	函数	函数名称	最值
$f_1$	Shifted and Rotated Bent Cigar Function	100	$f_{17}$	Hybrid Function 6(N=4)	1700
$f_3$	Shifted and Rotated Zakharov Function	300	$f_{18}$	Hybrid Function 6(N=5)	1800
$f_4$	Shifted and Rotated Rosenbrock's Function	400	$f_{19}$	Hybrid Function 6(N=5)	1900
$f_5$	Shifted and Rotated Rastrigin's Function	500	$f_{20}$	Hybrid Function 6(N=6)	2000
$f_6$	Shifted and Rotated Expanded Scaffer's F6 Function	600	$f_{21}$	Composition Function 1(N=3)	2100
$f_7$	Shifted and Rotated Lunacek Bi-Rastrigin Function	700	$f_{22}$	Composition Function 2(N=3)	2200
$f_8$	Shifted and Rotated Non-Continuous Rastrigin's Function	800	$f_{23}$	Composition Function 3(N=4)	2300
$f_9$	Shifted and Rotated Lévy Function	900	$f_{24}$	Composition Function 4(N=4)	2400
$f_{10}$	Shifted and Rotated Schwefel's Function	1000	$f_{25}$	Composition Function 5(N=5)	2500
$f_{11}$	Hybrid Function 1(N=3)	1100	$f_{26}$	Composition Function 6(N=5)	2600
$f_{12}$	Hybrid Function 2(N=3)	1200	$f_{27}$	Composition Function 7(N=6)	2700
$f_{13}$	Hybrid Function 3(N=3)	1300	$f_{28}$	Composition Function 8(N=6)	2800
$f_{14}$	Hybrid Function 4(N=4)	1400	$f_{29}$	Composition Function 9(N=3)	2900
$f_{15}$	Hybrid Function 5(N=4)	1500	$f_{30}$	Composition Function 10(N=3)	3000
$f_{16}$	Hybrid Function 5(N=4)	1600	—	—	—

### 3.2 消融实验

为进一步阐明 ROL 策略、自适应更新策略和黏菌觅食机制分别对 DMO 的影响, 本文对此开展了消融对比实验。将 EDMO 分别删除 ROL 改进策略、自适应更新策略、黏菌觅食机制的算法依次命名为 RDMO、ADMO、SDMO, 利用 CEC2017 函数测试集对其进行测试。运行次数、种群规模、测试维度以及迭代最大次数均与 4.1 节保持一致, 对上述算法进行重复实验。定义消融因子  $\gamma$  定量评估缺失策略的重要程度, 其计算公式为:

$$\gamma = \frac{f_{EDMO}}{f_{EDMOi}} \quad (12)$$

式中:  $f_{EDMOi}$  和  $f_{EDMO}$  分别为不完全改进 DMO 与 EDMO 的最佳适应度平均值,  $\gamma$  越小, 表明对照算法的寻优性能越差, 即所缺少策略对于改善原始算法性能的作用越大。消融实验结果如表 2 所示, RDMO、ADMO、SDMO 的  $\gamma$  值普遍大于 DMO, 表明任意两种策略的叠加均可改善 DMO 的寻优性能。

表 2 消融实验结果对比  
Table 2 Comparison of ablation results

函数	DMO	RDMO	ADMO	SDMO	函数	DMO	RDMO	ADMO	SDMO
$f_1$	0.00	0.32	0.72	0.03	$f_{17}$	0.84	0.95	0.96	0.88
$f_3$	0.51	0.52	0.92	0.31	$f_{18}$	0.02	0.78	0.77	0.32
$f_4$	0.83	0.85	0.88	0.88	$f_{19}$	0.39	0.53	0.78	0.48
$f_5$	0.88	0.92	0.94	0.88	$f_{20}$	0.88	0.95	0.94	0.91
$f_6$	0.98	0.97	0.98	1.00	$f_{21}$	0.99	0.98	0.98	0.95
$f_7$	0.84	0.93	0.88	0.87	$f_{22}$	0.57	0.68	1.04	0.77
$f_8$	0.95	0.96	0.97	0.90	$f_{23}$	0.97	0.98	0.97	0.97
$f_9$	0.33	0.51	0.49	0.67	$f_{24}$	1.00	0.96	1.01	0.98
$f_{10}$	0.54	0.61	0.52	0.55	$f_{25}$	0.98	0.98	0.99	0.99
$f_{11}$	1.05	1.12	1.11	1.12	$f_{26}$	0.79	0.81	0.77	0.72
$f_{12}$	0.39	0.65	0.46	0.43	$f_{27}$	0.93	0.99	0.99	0.98
$f_{13}$	0.31	0.41	0.45	0.55	$f_{28}$	0.92	0.96	0.97	0.99
$f_{14}$	2.14	8.33	0.13	0.04	$f_{29}$	0.83	0.91	0.89	0.91
$f_{15}$	0.12	0.21	0.23	0.24	$f_{30}$	0.06	0.43	0.21	0.43
$f_{16}$	0.83	0.88	0.88	0.79	—	—	—	—	—

本文提出的策略在单峰、多峰以及混合函数上都展现了明显的优势。相对而言, 自适应策略在  $f_1$  和  $f_3$  函数性能提升并不显著, ADMO 的  $\gamma$  值分别是 RDMO 和 SDMO 的 4.1 倍和 2.2 倍, 这主要是因为黏菌觅食策略是一种偏向全局性的改进策略<sup>[25]</sup>, 在这意味着, 当面对那些需要精细局部搜索的特定测试函数时, 这种全局搜索可能不如其他针对性更强的策略来得有效, 而  $f_1$  和  $f_3$  主要评估的是算法精



细化搜索的能力。不过这并不意味着黏菌觅食策略没有其独特的价值，在宽阔的搜索空间或那些需要广泛探索的问题上，黏菌觅食策略可能会展现出强大的搜索能力和韧性<sup>[26]</sup>。在  $f_{14}$  中，RDMO 的  $\gamma$  值为 8.33，相比之下，ADMO 仅占 RDMO 的 25.69%，SDMO 仅占 RDMO 的 1.56%，而 DMO 仅占 RDMO 的 0.48%，表明 ROL 改进策略对该函数的适应性较差。总体而言，各策略对 DMO 的提升效果没有显著差异( $\Delta\gamma < 0.3$ )，而算法整体性能的提升依赖于各策略之间的相互配合。

### 3.3 EDMO 与其他算法对比分析

CEC 系列函数可对实际问题的复杂性进行模拟，这对新算法的研究具有指导意义。为体现本文算法的优越性，将其与另外 3 种竞赛算法 CMA-ES<sup>[27]</sup>、LSHADE\_cnEpsin<sup>[28]</sup>、LSHADE<sup>[29]</sup>、原始 DMO 以及文献[15]中改进的 IDMO 进行对比，为保证内置参数的一致性，实验时运行次数、种群规模、测试维度以及迭代最大次数均与 4.1 节保持一致，独立运行 30 次并记录每组实验的最优适应度值。最优适应度值即测试函数函数值，优化算法目的是寻求的是寻求函数极小值。表 3 给出了 CMA-ES、LSHADE\_cnEpsin、LSHADE、DMO、IDMO、与 EDMO 重复 30 次实验的最优适应度均值(Mean)和标准差 (Std)，对每组测试函数中均值和标准差最优的算法加粗显示。

表 3 CEC2017测试结果  
Table 3 Test results for CEC 2017

函数	CMA-ES Mean(Std)	LSHADE_cnEpsin Mean(Std)	LSHADE Mean(Std)	DMO Mean(Std)	IDMO Mean(Std)	EDMO Mean(Std)
$f_1$	1.32E+10(4.86E+09)/+	1.73E+10(5.77E+09)/+	2.38E+07(2.23E+07)/+	5.26E+06(1.13E+07)/+	1.98E+04(3.53E+04)/+	5.71E+02(7.02E+02)
$f_3$	6.35E+04(9.89E+03)/+	4.19E+04(1.02E+04)/+	6.44E+04(1.05E+04)/+	4.21E+04(9.81E+03)/+	1.62E+04(4.67E+03)/-	2.05E+04(2.01E+03)
$f_4$	1.37E+03(7.52E+02)/+	2.30E+03(1.59E+03)/+	5.35E+02(3.14E+01)/+	5.20E+02(3.20E+01)/+	5.21E+02(3.32E+01)/+	4.39E+02(3.69E+01)
$f_5$	7.12E+02(4.82E+01)/+	7.73E+02(3.74E+01)/+	6.83E+02(3.19E+01)/+	6.31E+02(3.05E+01)/+	6.54E+02(3.39E+01)/+	5.57E+02(8.73E+00)
$f_6$	6.38E+02(6.68E+00)/+	6.62E+02(8.40E+00)/+	6.43E+02(1.04E+01)/+	6.22E+02(8.73E+00)/+	6.33E+02(1.06E+01)/+	6.02E+02(2.49E-13)
$f_7$	1.08E+03(6.14E+01)/+	1.28E+03(5.47E+01)/+	1.05E+03(5.24E+01)/+	9.40E+02(5.95E+01)/+	9.97E+02(6.77E+01)/+	7.75E+02(6.11E+00)
$f_8$	9.87E+02(3.95E+01)/+	9.96E+02(2.60E+01)/+	9.58E+02(2.57E+01)/+	9.18E+02(1.96E+01)/+	9.16E+02(2.47E+01)/+	8.65E+02(1.06E+01)
$f_9$	5.54E+03(1.40E+03)/+	5.86E+03(5.75E+02)/+	3.88E+03(4.76E+02)/+	3.45E+03(1.65E+03)/+	3.14E+03(8.49E+02)/+	1.14E+03(1.13E+02)
$f_{10}$	6.64E+03(1.43E+03)/+	5.34E+03(3.92E+02)/+	5.65E+03(2.85E+02)/+	5.73E+03(9.75E+02)/+	5.02E+03(6.75E+02)/+	3.15E+03(2.67E+02)
$f_{11}$	3.75E+03(1.57E+03)/+	2.13E+03(6.78E+02)/=	1.23E+03(4.05E+01)/=	1.32E+03(6.56E+01)/=	1.27E+03(5.44E+01)/=	1.35E+03(2.27E+01)
$f_{12}$	9.57E+08(1.13E+09)/+	1.28E+09(1.26E+09)/+	1.63E+06(1.18E+06)/+	1.28E+06(9.25E+05)/+	8.78E+05(8.12E+05)/=	4.56E+05(2.58E+05)
$f_{13}$	4.76E+08(1.33E+09)/+	1.85E+07(5.16E+07)/=	1.75E+04(1.37E+04)/=	4.38E+04(2.86E+04)/+	2.08E+04(1.97E+04)/=	1.26E+04(6.56E+03)
$f_{14}$	1.18E+06(1.19E+06)/+	2.37E+04(2.42E+04)/-	2.76E+04(3.20E+04)/-	4.97E+04(5.13E+04)/-	1.50E+04(2.18E+04)/-	1.07E+05(6.23E+04)
$f_{15}$	1.58E+07(3.31E+07)/+	5.16E+04(4.74E+04)/+	5.66E+03(3.55E+03)/+	1.98E+04(2.24E+04)/+	9.73E+03(1.08E+04)/+	2.21E+03(1.63E+03)
$f_{16}$	2.95E+03(3.88E+02)/+	3.14E+03(3.98E+02)/+	2.88E+03(1.62E+02)/+	2.74E+03(4.02E+02)/+	2.76E+03(2.83E+02)/+	2.24E+03(1.35E+02)
$f_{17}$	2.34E+03(2.12E+02)/+	2.26E+03(1.84E+02)/+	2.03E+03(1.35E+02)/+	2.27E+03(1.65E+02)/+	2.38E+03(1.98E+02)/+	1.94E+03(8.82E+01)
$f_{18}$	1.67E+06(1.94E+06)/+	4.97E+05(5.87E+05)/=	2.43E+05(1.94E+05)/=	2.07E+07(1.03E+08)/+	1.57E+05(1.55E+05)/=	1.46E+05(5.52E+04)
$f_{19}$	1.26E+07(3.22E+07)/+	1.63E+06(1.73E+06)/+	8.34E+03(7.38E+03)/+	9.64E+03(1.32E+04)/+	9.96E+03(1.13E+04)/+	3.61E+03(1.57E+03)
$f_{20}$	2.63E+03(1.89E+02)/+	2.52E+03(1.55E+02)/+	2.46E+03(1.05E+02)/+	2.60E+03(2.17E+02)/+	2.58E+03(2.28E+02)/+	2.25E+03(1.05E+02)
$f_{21}$	2.48E+03(4.12E+01)/+	2.55E+03(7.36E+01)/+	2.44E+03(2.57E+01)/+	2.41E+03(2.26E+01)/+	2.43E+03(3.02E+01)/+	2.35E+03(4.44E+01)
$f_{22}$	6.28E+03(2.67E+03)/+	5.76E+03(1.64E+03)/+	2.32E+03(3.10E+01)/-	4.24E+03(2.81E+03)/+	4.92E+03(2.38E+03)/+	2.38E+03(3.94E+01)
$f_{23}$	2.92E+03(6.34E+01)/+	3.06E+03(9.15E+01)/+	2.82E+03(3.24E+01)/+	2.82E+03(7.53E+01)/+	2.84E+03(4.55E+01)/+	2.72E+03(1.11E+01)
$f_{24}$	3.12E+03(7.15E+01)/+	3.22E+03(6.76E+01)/=	2.97E+03(3.45E+01)/=	2.97E+03(7.78E+01)/=	2.97E+03(5.75E+01)/+	2.95E+03(2.53E+01)
$f_{25}$	3.11E+03(1.25E+02)/+	3.35E+03(1.83E+02)/+	2.97E+03(2.47E+01)/+	2.95E+03(2.28E+01)/+	2.94E+03(1.84E+01)/+	2.88E+03(8.79E-01)
$f_{26}$	6.18E+03(6.04E+02)/+	7.22E+03(1.45E+03)/+	5.19E+03(1.70E+03)/+	4.72E+03(9.84E+02)/+	5.66E+03(1.47E+03)/+	3.71E+03(8.32E+02)
$f_{27}$	3.42E+03(7.92E+01)/+	3.37E+03(6.56E+01)/+	3.22E+03(1.64E+01)/+	3.46E+03(2.42E+02)/+	3.27E+03(3.88E+01)/+	3.23E+03(4.97E+00)
$f_{28}$	3.92E+03(2.99E+02)/+	4.12E+03(4.07E+02)/+	3.33E+03(3.67E+01)/+	3.56E+03(1.13E+03)/+	3.27E+03(2.56E+01)/+	3.20E+03(1.05E+01)
$f_{29}$	4.26E+03(2.46E+02)/+	4.58E+03(4.43E+02)/+	4.14E+03(2.23E+02)/+	4.22E+03(4.15E+02)/+	4.20E+03(2.81E+02)/+	3.52E+03(7.67E+01)
$f_{30}$	4.06E+07(3.28E+07)/+	1.18E+07(8.86E+06)/+	2.94E+04(3.28E+04)/+	1.37E+05(2.82E+05)/+	2.13E+04(1.12E+04)/+	7.58E+03(1.22E+03)
+/-/-	29/0/0	24/4/1	23/4/2	26/2/1	23/4/2	Wilcoxon

采用 Wilcoxon 秩和检验评估 EDMO 在显著性水平为 5%时与其他算法的显著程度, 当检验结果  $P < 5\%$  时, 表明两种算法之间差异显著, 否则差异不明显。采用秩和检验对测试结果进行评估, 其中, “+”、“-”和“=”分别表示 EDMO 寻优效果强于、弱于和接近于其他算法。统计结果表明, EDMO 在与 CMA-ES 的对比中取得绝对优势, 仅在不超两个基准函数的寻优效果稍微逊色于 LSHADE\_cnEpsin、LSHADE、DMO、IDMO。经多策略联合改进后, EDMO 的收敛精度与稳定程度均明显强于 DMO, 从而初步验证了改进策略的有效性。

EDMO 在测试单峰函数时, 表现在  $f_1$  上最取得了最优的均值和标准差,  $f_3$  的效果略逊。具体来看, EDMO 算法在  $f_3$  上的性能仅稍低于最优算法, 其均值和标准差分别比最优算法高出约 26.54%和 57.17%。这一结果表明 EDMO 算法在处理单峰函数时整体表现接近于最优, 仅在  $f_3$  函数上略有不足。在多峰函数  $f_4$  至  $f_{10}$  中, EDMO 寻优效果一直处于领先地位, 平均值和标准差都小于对比算法, 凸显其强大的避免局部最优的能力; 混合函数  $f_{11} \sim f_{20}$  与复合函数  $f_{21} \sim f_{30}$  旨在测试算法应对复杂问题时的表现, 其中除函数  $f_{11}$ 、 $f_{14}$ 、 $f_{22}$  外, EDMO 在其余函数上均展示了最优的均值和标准差。EDMO 在  $f_{11}$  和  $f_{22}$  上虽然效果不是最佳的, 但是平均收敛精度与表现最优的算法差距不大

( $\Delta\text{Mean}<130, \Delta\text{Std}<20$ ), 并且在  $f_{11}$  上与其他算法均有相似的寻优成果, 但对  $f_{14}$  的处理水平普遍逊色于其他算法, 除了 CMA-ES 算法外, 均值和标准差均高于其他算法。

考虑到  $f_{14}$  是由四个基本函数组合而成, 并且其全局最优值位于一个狭窄的抛物线底部, 这显然为算法带来了挑战。尽管黏菌觅食策略的引入增强了侏儒猫鼬的全局勘探能力, 但这也可能在某种程度上削弱了其对局部解的精确挖掘能力。特此针对  $f_{14}$  进行敏感性测试, 探究  $\alpha$  值与  $f_{14}$  的收敛性之间的关系, 并据此求得  $\alpha$  值和  $f_{14}$  的拟合曲线如图 3 所示:

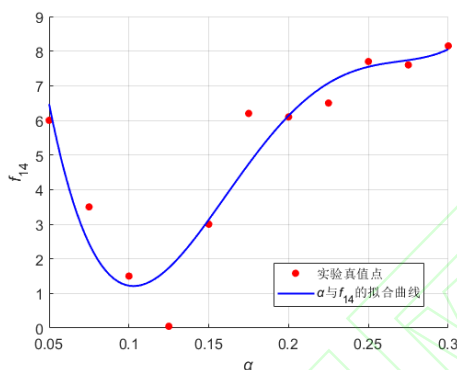


图 3  $\alpha$  和  $f_{14}$  拟合曲线  
Fig.3 Fitted curve with  $\alpha$  and  $f_{14}$

由图 3 可知, 当值从 0.3 等间隔降低至 0.125 时, 最终收敛值从 8.05 逐渐降低到 0.04。然而, 当值超过 0.125 继续降低到 0.05 时, 最终收敛值逐渐劣化到 6.09。因此, 合理的值能够显著提高优化性能, 解释了消融实验中删除自适应更新策略 ADMO 算法表现欠优的原因, 进一步验证了自适应调整机制的独特优势。为进一步分析上述算法的收敛速度及迭代进程, 选取 8 个不同类型的测试函数进行比较。由图 4 可知, EDMO 不论在收敛速度及精度上都有优于其他算法, 其收敛曲线在迭代前中期迅速下降, 在前 60% 迭代次数内实现了 95% 以上的收敛, 这得益于通过 ROL 和自适应更新策略等多策略的引导, 使得算法能够更快地探索解空间的不同区域, 找到候选解, 使得搜索过程高效进行。综上所述, EDMO 是一种一种高效的智能优化算法, 它不仅能够提供高质量的解决方案, 而且具备显著的稳定性、快速的收敛速度、高度的收敛精度以及强大的避免陷入局部最优的能力。这些特性使得 EDMO 在各种优化问题中表现出色, 特别是在那些需要快速、准确解决方案的场景中。

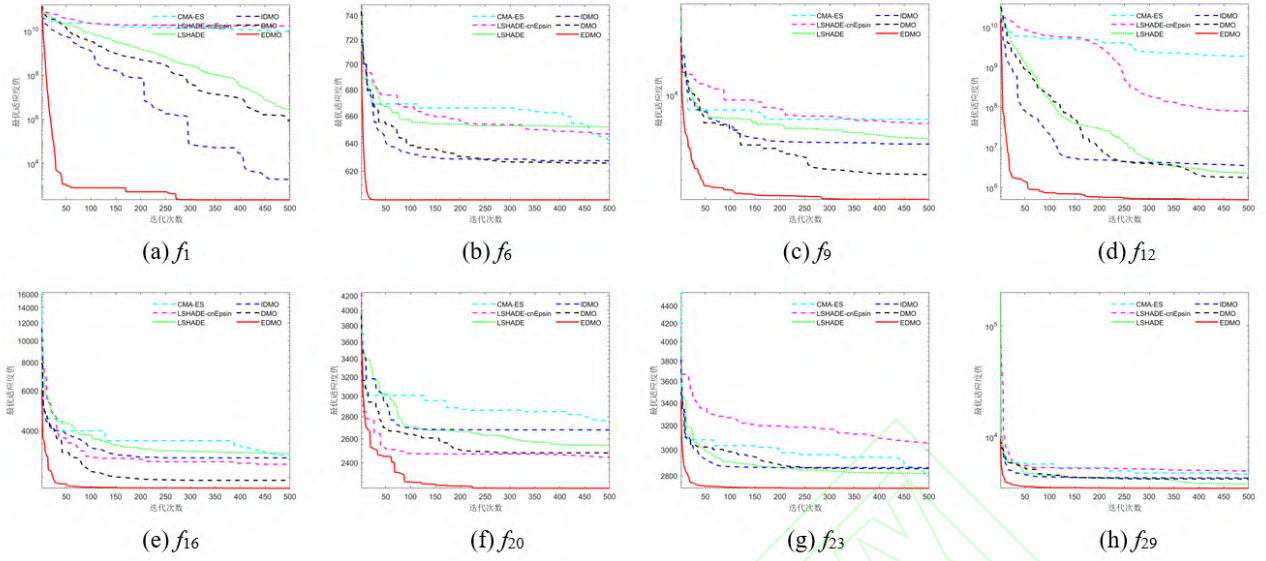


图 4 收敛曲线对比  
Fig.4 Convergence curve comparison

#### 4 EDMO 算法实际工程应用

在本文中,采用无人机三维路径规划问题探讨 EDMO 算法的实际应用效果。通过在复杂的山地模型中模拟无人机的飞行路径,展现了 EDMO 算法在应对当下热门的路径规划问题上,所具备的出色应用潜力。

##### 4.1 无人机路径规划代价函数

###### 4.1.1 飞行距离代价

飞行距离代价是评估无人机执行飞行任务时的一个重要指标,主要涉及燃油消耗。当假定无人机在执行任务时能够达到并维持一个恒定速度,其燃油消耗量与飞行的总距离呈正比关系。这意味着无人机飞行的距离越长,其燃油消耗也会相应增加,这是无人机运行成本和效率评估中的一个关键因素<sup>[30]</sup>。飞行距离代价的计算公式如下:

$$f_{range} = \frac{\varepsilon}{Q_r} \sum_{i=1}^n L_i \quad (13)$$

式中:  $\varepsilon$  为单位飞行距离的燃油消耗量,衡量无人机每飞行一定距离所需燃油量的指标。 $Q_r$  为无人机携带燃油总量,是无人机能够在不加油的情况下飞行的最大限度,  $L_i$  表示第  $i$  段航迹长度。

###### 4.1.2 飞行海拔代价

在无人机飞行中,考虑到机体受低温影响的概率与飞行高度直接相关,而地形威胁的风险则与飞行高度成反比,因此设定了最大飞行高度  $h_{max}$  和最低飞行高度  $h_{min}$  来确保无人机的安全飞行。在这种情况下,假设无人机当前的飞行高度为  $h_i$ ,那么飞行高度的代价函数可以用于量化在特定高度下飞行的整体风险。该代价函数将考虑到在  $h_{min}$  和  $h_{max}$  范围内,随着高度变化,低温影响和地形威胁如何平衡和影响无人机的安全性和效能。假设当前的飞行高度为  $h_i$ ,则飞行高度的代价函数可以表示为:

$$f_{H_i} = \begin{cases} \frac{h_i - h_{min}}{h_{max} - h_{min}}, & h_{min} < h_i < h_{max} \\ \infty, & \text{others} \end{cases} \quad (14)$$

$$f_{altitude} = \frac{1}{n} \sum_{i=1}^n f_{H_i} \quad (15)$$

#### 4.1.3 飞行风险代价

在无人机的航迹规划中,考虑到环境地形、极端天气、军事行动等飞行风险区的影响,飞行环境中的风险代价成为一个关键因素。假设在规划的第  $i$  段航迹中,无人机接近某个飞行风险点  $k$ ,如果将这段航迹  $L_i$  划分为  $m$  个小段,那么针对飞行风险点  $k$  对这段航迹  $L_i$  的飞行风险代价函数可以设定:

$$f_{n_i,k} = \frac{1}{m} (P_k(d_{k,1}) + P_k(d_{k,2}) + \cdots + P_k(d_{k,m})) \quad (16)$$

式中:  $n_i$  为航迹规划中的第  $i$  段航迹  $L_i$ ;  $P_k$  为第  $k$  个威胁点无人机的毁伤概率;  $d_{k,m}$  为第  $k$  个威胁点到航迹段  $L_i$  中第  $m$  小段的距离。无人机航迹规划受到的总威胁代价可以表示为:

$$f_{\text{risk}} = \frac{1}{n} \frac{1}{k} \sum_{i=1}^n \sum_{k=1}^k f_{n_i,k} \quad (17)$$

#### 4.2 无人机协同路径规划的性能度量函数

在无人机航迹规划中,综合考虑飞行距离代价、飞行海拔代价和飞行风险代价这三种关键因素,每种代价对航迹规划的影响程度不同。为了有效地平衡这些因素,可以为它们分别设定权重系数,分别为  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$ , 无人机协同航迹规划的目标函数可以表达为:

$$f = \omega_1 f_{\text{range}} + \omega_2 f_{\text{altitude}} + \omega_3 f_{\text{risk}} \quad (18)$$

#### 4.3 无人机三维路径规划仿真与分析

##### 4.3.1 算法应用与实验仿真

为了展现新算法的效果,我们在 Matlab 环境下对无人机的三维航迹进行了模拟。仿真环境设定在一个  $100 \text{ km} \times 150 \text{ km} \times 30 \text{ km}$  的空间里,起点设在坐标 (10, 90), 终点则在 (130, 10)。风险区域的位置信息列在表 4 中,而在仿真图像中,这些区域以粉红色显示,具体可见于图 5。

采用 DMO 算法对前述问题进行三维路径规划的求解时,作为参考的参数设置如下:算法的种群规模  $N$  设为 30, 最大迭代次数  $T$  定为 200。仿真结果如图 6~图 8 所示。

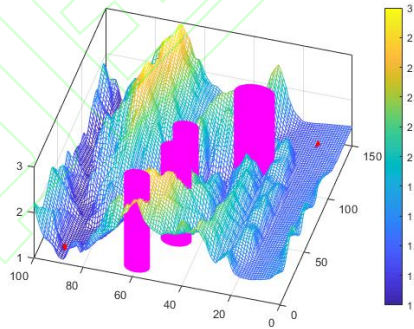


图 5 三维威胁地图

Fig.5 Three-dimensional threat topographic map

表 4 风险区域2维坐标参数

Table 4 2-D coordinate parameters of risk area

风险区域	中心点坐标	风险半径
1	(10,60)	5
2	(40,50)	6
3	(60,50)	5
4	(100,30)	8



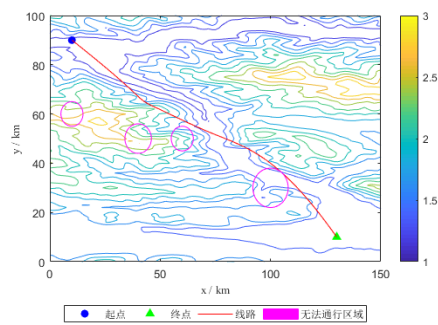


图 6 二维航迹规划图  
Fig.6 Two-dimensional track plan

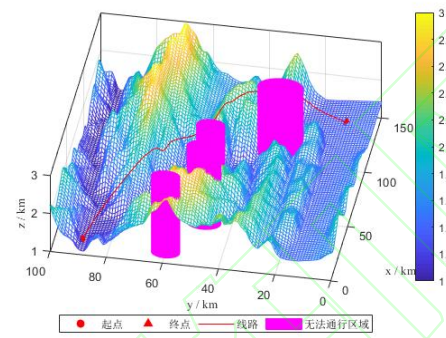


图 7 三维航迹规划图  
Fig.7 Three-dimensional track plan

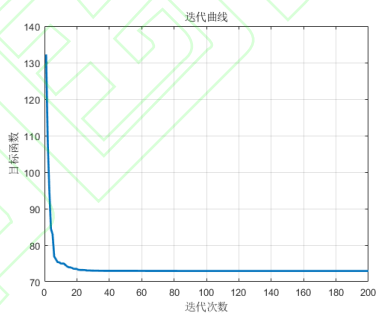


图 8 适应度变化曲线  
Fig.8 Fitness change curve

EDMO 算法参数设置与 4.1 中保持一致，EDMO 算法在无人机三维路径规划实验仿真结果如图 9~图 11 所示。

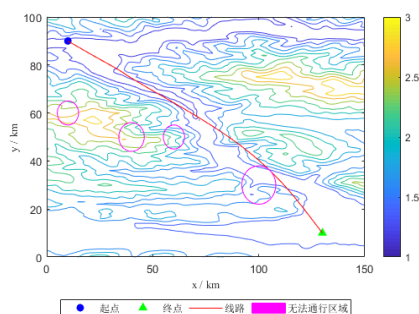


图9 二维航迹规划图

Fig.9 Two-dimensional track plan

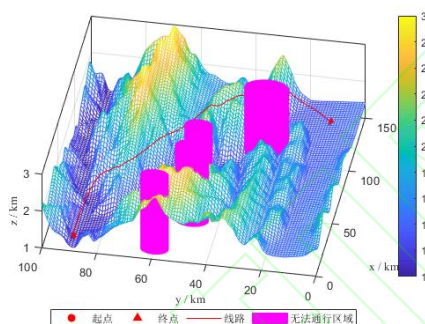


图10 三维航迹规划图

Fig.10 Three-dimensional track plan

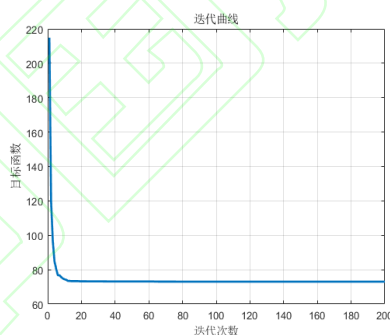


图11 适应度变化曲线

Fig.11 Fitness change curve

#### 4.3.2 仿真结果分析

通过对实验结果的分析,对比图6和图9可以明显观察到DMO算法所规划的航线飞行距离更长,燃油消耗更多。同时,通过DMO算法规划出的航迹相对靠近风险区域2和3,这种规划方式带来了较大的风险隐患。相较而言,改进后的EDMO算法成功地规避了这一问题。与DMO算法相比,EDMO算法在最大限度地避开风险区域的同时,实现了更短的飞行距离和更平滑的飞行轨迹,表明其路径规划更加合理和高效。此外,改进后的EDMO算法所规划的飞行高度更低,使无人机能够更贴近山体飞行,有效避免了由于高空低温和缺氧带来的飞行风险。

观察适应度曲线的变化,可以发现经过改进的EDMO算法执行了更多次迭代。尽管该算法在初始阶段的目标函数值较高,但它最终收敛到了与原始算法类似的结果。这种现象表明改进后的EDMO算法在搜索空间中进行了更为广泛的探索,这反映出它在全局搜索能力方面的卓越表现。综合来看,对于无人机的三维路径规划,经过多策略融合改进的EDMO算法表现得更为优越,进一步验证了本

文改进策略的成果。

在相同的测试条件下，本研究对两种算法各进行了 30 次独立的仿真实验。为了深入分析和比较这两种算法的性能，研究还包括了对它们的综合代价模型进行的统计分析。相关的统计数据被汇总在表 5 中。

表 5 无人机三维路径规划结果统计  
Table 5 Statistics of UAV three-dimensional path planning results

实验算法	最优代价	最差代价	平均代价	标准差	平均迭代次数	平均运行时间
EDMO	72.92	72.97	72.96	0.03	124.53	37.41
DMO	72.96	73.07	73.01	0.08	117.60	39.77

通过比较两组实验数据，我们发现，尽管两种算法在无人机轨迹规划上的最优代价和最差代价相近，从统计数据来看，DMO 算法在平均代价及标准差方面均明显高于改进后的 EDMO 算法。这一结果说明在进行航迹规划时，EDMO 算法在寻优的稳定性方面表现更佳。此外，EDMO 算法的平均迭代次数也明显超过 DMO 算法，这表明 EDMO 算法在搜索过程中探索的范围更广泛，具有更强的全局搜索能力。

尽管在平均运算耗时方面，改进前后的两个算法表现相似，但综合考虑其他性能指标，EDMO 算法似乎更适合用于无人机的三维路径规划。这主要归因于它在代价、稳定性和全局搜索能力方面的优势，使其成为处理这类复杂优化问题的更有效工具。

5 结 论

1) 经过对 DMO 算法的计算流程和结构限制的系统分析，本文通过集成随机反向学习策略、自适应保姆交换参数更新以及黏菌觅食机制，对该算法进行了优化。这些改进措施有效提升了算法在全局搜索能力和局部开发精度方面的性能。

2) 应用改进的 EDMO 算法对 CEC2017 函数集进行优化测试，结果显示其在精度和效果方面均优于其他比较算法，且具有更快的收敛速度和更高的稳定性与鲁棒性。

3) 将 EDMO 算法应用于无人机三维路径规划问题，结果表明其优化效果显著优于原始 DMO 算法，验证了 EDMO 算法的有效性和适用性。

参考文献(References)

[1] 宋立钦, 陈文杰, 陈伟海, 等. 基于混合策略的麻雀搜索算法改进及应用[J]. 北京航空航天大学学报, 2023, 49(8): 2187-2199.  
SONG L Q, CHEN W J, CHEN W H, et al. Improvement and application of hybrid strategy-based sparrow search algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2023, 49(8): 2187-2199 (in Chinese).

[2] 李全耀, 沈艳霞. 一种基于教与学的混合灰狼优化算法[J]. 控制与决策, 2022, 37(12): 3190-3196. .  
LI Q Y, SHEN Y X. A hybrid gray wolf optimization algorithm based on the teaching-learning optimization[J]. Control and Decision, 2022, 37(12): 3190-3196. (in Chinese).

[3] 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.

[4] HASHIM F A, HUSSIEN A G. Snake Optimizer: A novel meta-heuristic optimization algorithm[J]. Knowledge-Based Systems, 2022, 242: 108320.

[5] JIA H, PENG X, LANG C. Remora optimization algorithm[J]. Expert Systems with Applications, 2021, 185: 115665.

[6] 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.

[7] ABD ELAZIZ M, LU S, HE S. A multi-leader whale optimization algorithm for global optimization and image segmentation[J]. Expert Systems with Applications, 2021, 175: 114841.

[8] ZHANG L, ZHANG Y, LI Y. Mobile robot path planning based on improved localized particle swarm optimization[J]. IEEE Sensors Journal, 2020, 21(5): 6962-6972.

[9] 柴岩, 朱玉, 任生. 多策略协同的改进鲸鱼优化算法[J]. 计算机工程与科学, 2023, 45(7): 1308-1319.  
CHAI Y, ZHU Y, REN S. An improved whale optimization algorithm based on multi-strategy coordination[J]. Computer Engineering & Science, 2023, 45(7): 1308-1319 (in Chinese).

[10] AGUSHAKA J O, EZUGWU A E, ABUALIGAH L. Dwarf mongoose optimization algorithm[J]. Computer Methods in Applied Mechanics and Engineering, 2022, 391: 114570.

[11] AGUSHAKA J O, AKINOLA O, EZUGWU A E, et al. Advanced dwarf mongoose optimization for solving CEC 2011 and CEC 2017 benchmark problems[J]. Plos One, 2022, 17(11): e0275346.

- [12] AGUSHAKA J O, EZUGWU A E, OLAIDE O N, et al. Improved dwarf mongoose optimization for constrained engineering design problems[J]. *Journal of Bionic Engineering*, 2023, 20(3): 1263-1295.
- [13] ALDOSARI F, ABUALIGAH L, ALMOTAIRI K H. A normal distributed dwarf mongoose optimization algorithm for global optimization and data clustering applications[J]. *Symmetry*, 2022, 14(5): 1021.
- [14] AKINOLA O A, EZUGWU A E, OYELADE O N, et al. A hybrid binary dwarf mongoose optimization algorithm with simulated annealing for feature selection on high dimensional multi-class datasets[J]. *Scientific Reports*, 2022, 12(1): 14945.
- [15] 贾鹤鸣, 陈丽珍, 力尚龙, 等. 透镜成像反向学习的精英池侏儒猫鼬优化算法[J/OL]. *计算机工程与应用*: 1-12 [2023-09-18]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20221019.1554.014.html> JIA H M, CHEN L Z, LI S L, et al. Optimization algorithm of elite pool dwarf mongoose based on lens imaging reverse learning[J/OL]. *Computer Engineering and Applications*: 1-12 [2023-09-18]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20221019.1554.014.html> (in Chinese).
- [16] 赵世杰, 张红易, 马世林. 领导者引导与支配解进化的多目标矮猫鼬算法[J/OL]. *计算机科学与探索*: 1-22 [2023-09-18]. <http://kns.cnki.net/kcms/detail/11.5602.TP.20221213.1733.002.html>. ZHAO S J, ZHANG H Y, MA S L. Multi-objective Dwarf Mongoose Optimization Algorithm with Leader Guidance and Dominated Solution Evolution Mechanism[J/OL]. *Journal of Frontiers of Computer Science and Technology*: 1-22 [2023-09-18]. <http://kns.cnki.net/kcms/detail/11.5602.TP.20221213.1733.002.html> (in Chinese).
- [17] AL-SHOUBAJI I, KACHARE P, FADLELSEED S, et al. Artificial ecosystem-based optimization with dwarf mongoose optimization for feature selection and global optimization problems[J]. *International Journal of Computational Intelligence Systems*, 2023, 16(1): 1-24.
- [18] FU S, HUANG H, MA C, et al. Improved dwarf mongoose optimization algorithm using novel nonlinear control and exploration strategies[J]. *Expert Systems with Applications*, 2023, 233: 120904.
- [19] RIZK-ALLAH R M, EL-FERGANY A A, GOUDA E A, et al. Characterization of electrical 1-phase transformer parameters with guaranteed hotspot temperature and aging using an improved dwarf mongoose optimizer[J]. *Neural Computing and Applications*, 2023, 35(19): 13983-13998.
- [20] 罗淑媛, 张家豪, 宋美佳, 等. 改进矮猫鼬优化算法的特征选择[J]. *龙岩学院学报*, 2023, 41(2): 40-46.  
LUO S Y, ZHANG J H, SONG M J, et al. Improved dwarf mongoose optimization algorithm for feature selection[J]. *Journal of Longyan University*, 2023, 41(2): 40-46 (in Chinese).
- [21] 何加文, 许贤泽, 高波. 多策略融合改进的飞蛾火焰优化算法[J/OL]. *北京航空航天大学学报*: 1-12 [2023-09-18]. <https://doi.org/10.13700/j.bh.1001-5965.2022.0707>.  
HE J W, XU X Z, GAO B. Improved moth-flame optimization algorithm with multi-strategy integration [J/OL]. *Journal of Beijing University of Aeronautics and Astronautics*: 1-12 [2023-09-18]. <https://doi.org/10.13700/j.bh.1001-5965.2022.0707> (in Chinese).
- [22] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence[C]//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, 1: 695-701.
- [23] LONG W, JIAO J, LIANG X, et al. A random opposition-based learning grey wolf optimizer[J]. *IEEE Access*, 2019, 7: 113810-113825.
- [24] 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.
- [25] 郭雨鑫, 刘升, 张磊, 等. 精英反向与二次插值改进的黏菌算法[J]. *计算机应用研究*, 2021, 38(12): 3651-3656.  
GUO Y X, LIU S, ZHANG L, et al. Elite opposition-based learning quadratic interpolation slime mould algorithm[J]. *Application Research of Computers*, 2021, 38(12): 3651-3656 (in Chinese).
- [26] 任丽莉, 王志军, 闫冬梅. 结合黏菌觅食行为的改进多元宇宙算法[J]. *吉林大学学报(工学版)*, 2021, 51(6): 2190-2197.  
REN L L, WANG Z J, YAN D M. Improved multi-verse algorithm with combined slime mould foraging behavior[J]. *Journal of Jilin University(Engineering and Technology Edition)*, 2021, 51(6): 2190-2197 (in Chinese).
- [27] HANSEN N. The CMA evolution strategy: A tutorial[J]. *arXiv preprint arXiv:1604.00772*, 2016.
- [28] TANABE R, FUKUNAGA A S. Improving the search performance of SHADE using linear population size reduction[C]//2014 IEEE congress on evolutionary computation (CEC). IEEE, 2014: 1658-1665.
- [29] AWAD N H, ALI M Z, SUGANTHAN P N. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems[C]//2017 IEEE congress on evolutionary computation (CEC). IEEE, 2017: 372-379.
- [30] 蔺文轩, 谢文俊, 张鹏, 等. 基于分组优化改进粒子群算法的无人机三维路径规划[J]. *火力与指挥控制*, 2023, 48(1): 20-25.  
LIN W X, XIE W J, ZHANG P, et al. UAV 3D Path Planning based on grouping optimization improved particle swarm algorithm[J]. *Fire Control & Command Control*, 2023, 48(1): 20-25 (in Chinese).