

基于多策略改进人工兔优化算法的三维无人机 路径规划方法

王文涛¹, 叶 晨², 田 军^{1*}

(1. 南开大学软件学院, 天津 300350; 2. 江西农业大学计算机与信息工程学院, 江西南昌 330045)

摘 要: 三维无人机路径规划问题旨在满足安全性条件的前提下为无人机规划出一条最佳的飞行路径. 本文通过数学建模的方式构建出无人机路径规划的成本函数, 从而将无人机路径规划问题转化为多约束的优化问题, 并使用元启发式算法来求解该问题. 针对人工兔优化算法收敛慢以及易陷入局部最优的缺陷, 本文开发了一种基于 Levy 飞行、自适应柯西变异以及精英群遗传策略改进的人工兔优化算法 (Artificial Rabbit Optimization algorithm based on Levy flight, adaptive Cauchy mutation, and elite population Genetic strategy, LCGARO). 将 LCGARO 与 6 个经典和先进的元启发式算法在 29 个 CEC2017 测试函数和 6 个复杂度不同的三维无人机路径规划地形场景中进行多方面对比实验. 对比实验结果证明, 在 CEC2017 测试函数的对比实验中, 本文提出的 LCGARO 算法在 22 个测试函数中具有更优的寻优精度. 在无人机路径规划实验中, LCGARO 算法在 5 个地形场景中能够规划出总成本函数值最小的飞行路径.

关键词: 三维无人机路径规划; 人工兔优化算法; Levy 飞行; 自适应柯西变异; 精英群遗传策略; 元启发式算法

基金项目: 国家重点研发计划 (No.2021YFB0300104)

中图分类号: TP301; TP391.9 **文献标识码:** A

文章编号: 0372-2112(2024)11-3780-18

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20230382

A 3D UAV Path Planning Method Based on Multi-Strategy Improved Artificial Rabbit Optimization Algorithm

WANG Wen-tao¹, YE Chen², TIAN Jun^{1*}

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

2. School of Computer and Information Engineering, Jiangxi Agriculture University, Nanchang, Jiangxi 330045, China)

Abstract: The 3D UAV (Unmanned Aerial Vehicle) path planning problem aims to plan an optimal flight path for the UAV while satisfying safety conditions. In this paper, a cost function for UAV path planning is constructed by means of mathematical modeling, so that the UAV path planning problem is transformed into a multi-constrained optimization problem, and metaheuristic algorithms are applied to solve this problem. Aiming at the shortcomings of artificial rabbit optimization algorithm which is slow to converge and easy to fall into local optimum, this paper develops an improved Artificial Rabbit Optimization algorithm based on Levy flight, adaptive Cauchy mutation, and elite population Genetic strategy (LCGARRO). Multifaceted comparison experiments are conducted between LCGARRO and six classical and advanced heuristic algorithms in 29 CEC2017 test functions and six 3D UAV path-planning terrain scenarios of varying complexity. The results of the comparison experiments prove that the LCGARRO algorithm proposed in this paper has better optimization accuracy among 22 test functions in the comparison experiments of CEC2017 test functions. In the UAV path planning experiments, the LCGARRO algorithm is able to plan a flight path with the smallest total cost function value in five terrain scenarios.

Key words: 3D UAV (Unmanned Aerial Vehicle) path planning; artificial rabbit optimization algorithm; Levy flight; adaptive Cauchy mutation; elite group genetic strategy; metaheuristic algorithm

Foundation Item(s): National Key Research and Development Program of China (No.2021YFB0300104)

1 引言

无人驾驶飞行器通常称为无人机. 与有人驾驶的飞机相比, 无人机有更灵活、环境适应能力强、可拓展性更高的特点. 由于科学技术的进步和无人机生产成本的降低, 人们对无人机的各种应用产生了愈发浓厚的兴趣. 无人机的多功能性和可拓展性的特点使其广泛应用在目标追踪^[1]、智慧农业^[2]、救援任务^[3]以及物流应用^[4]等多方面领域. 在无人机独立执行任务时, 其所面临的实时环境很难完美反馈给控制器或者操作人员, 因此无人机要完成上述任务, 最基本的要求就是无人机需要拥有完备的自主规划路径的能力.

无人机路径规划问题是在确定飞行条件、始发点和目的地的前提下, 为无人机规划一条无碰撞的飞行路径. 随着无人机研究的深入, 各种复杂任务对无人机的飞行路径的要求也逐渐变得严格. 目前来看, 无人机的飞行路径规划问题是复杂且多约束的问题^[5]. 这要求无人机所规划的路径不仅要满足无磕碰的前提, 并且要同时考虑路径成本、燃料成本、安全成本和时间成本等多个因素. 截至目前, 已有很多优质的方法用于辅助无人机进行路径规划, 如快速随机树算法^[6]、Dijkstra算法^[7]、元启发式算法^[8]、人工势场法^[9]、D* Lite算法^[10]等. Wang等人^[11]基于改进蜉蝣算法提出了一种多约束的无人机路径规划方法, 这种方法能够保证在规划出高质量的路径的同时保证较快的计算速度. Souza等人^[12]提出了一种改进的人工势场算法(Modified Artificial Potential Field, MAPF)用于解决无人机群在三维环境下的路径规划问题, MAPF相对于传统人工势场算法而言又具有更好的可拓展性, 能够有效地规划多架无人机的路径. Zhang等人^[13]采用了改进的A*算法、稀疏A*搜索算法(SAS)和动态A*算法(D*)算法来解决三种不同威胁情景中隐身无人机的渗透路线计划的问题.

无人机三维路径规划问题需要综合考虑所规划路径的飞行距离、飞行高度、无人机爬升成本、安全成本等诸多影响因素, 因此无人机三维路径规划问题是一个NP-Hard问题, 问题的复杂性随着问题的大小而增加^[14]. 在NP-Hard问题的求解及优化上, 元启发式算法是一个不错的选择. 随着研究者们对元启发式算法研究的不断深入与创新, 元启发式算法目前主要分为基于进化的算法、基于群体智能的算法、基于人类的算法和基于物理和化学的算法. 基于进化的算法主要是模拟达尔文法则即自然界中优胜劣汰的进化规律来对算法中的候选解进行更新, 基于进化的算法主要有遗传算法(Genetic Algorithm, GA)^[15]和差分进化算法(Differential Evolution, DE)^[16]. 一些经典的和新颖的基于群体智能的算法有粒子群算法(Particle Swarm Optimization, PSO)^[17]、侏儒猫优化算法(Dwarf Mongoose

Optimization Algorithm, DMOA)^[18]和沙猫群优化算法(Sand Cat Swarm Optimization, SCSO)^[19]等. 社会进化与学习优化算法(Socio Evolution & Learning Optimization algorithm, SELO)^[20]和多元宇宙优化算法(Multi-Verse Optimizer, MVO)^[21]分别是基于人类的算法和基于物理和化学的算法中极具代表性的算法.

人工兔优化算法(Artificial Rabbits Optimization, ARO)^[22]是受到兔子在自然界中迂回觅食和随机躲藏的生存策略启发从而提出的一种新型的元启发式优化算法. 人工兔优化算法一经提出便被应用在各个领域, 且均取得了不错的成效. Elshahed等人^[23]将ARO算法应用在电网运行问题上, 在考虑24小时不同负荷情况下最小化电网每日能量损失和每日电压曲线. Riad等人^[24]使用ARO算法来为质子交换膜燃料电池模型寻找不确定参数的最优值. Mazloumi等人^[25]基于ARO算法设计了一种带储水箱(Water Storage Tank, WST)的光伏(PV)抽水系统最优尺寸. 虽然ARO具有较强的局部开发潜力和良好的搜索效率, 但ARO仍然存在收敛速度慢, 在处理一些复杂或高维问题时容易陷入局部最优的缺点. Wang等人^[26]结合Levy飞行策略和选择性反向学习策略开发了ARO的增强变体LARO. Wang等人^[27]将天鹰优化(Aquila Optimizer, AO)^[28]算法和人工兔优化算法进行融合, 改善了两种单一算法的局限性, 与此同时引入混沌反向学习策略驱动AO与ARO进行整合, 从而提出CHAOARO算法.

本文主要针对ARO收敛速度慢以及在复杂问题上容易被局部极值所吸引的缺陷, 开发了一种基于Levy飞行、自适应柯西变异以及精英群遗传策略的人工兔优化算法(Artificial Rabbit Optimization algorithm based on Levy flight, adaptive Cauchy mutation, and elite population Genetic strategy, LCGARO). Levy飞行算子显著改良了LCGARO算法处理复杂问题的能力, 自适应柯西变异算子增加了算法的种群多样性, 精英群遗传策略利用种群中优质的候选解去引导其他候选解进行位置更新, 从而极大地加快了算法的收敛速度. 将LCGARO在CEC2017基准函数上与其他一些优质算法进行对比分析, 实验结果显示LCGARO在CEC2017测试函数上展现出更优质的综合优化性能. 将本文提出的LCGARO算法应用在6种复杂度不同的三维地形场景下来辅助无人机进行路径规划, 并将规划所得的路径结果与其他一些经典的和新颖的算法所规划的结果进行对比, 对比实验结果证明LCGARO在三维无人机路径规划问题上的性能显著优于其他算法.

2 三维无人机路径规划数学模型

三维无人机路径规划问题是指在满足一定的约束条件下, 辅助无人机规划得到优质的飞行路径. 本文所

研究的路径规划问题涉及约束条件和成本函数等^[29].

2.1 距离成本

无人机飞行路径的距离成本主要与无人机的始发点和终点间的飞行距离相关. 在本研究中, 将无人机的飞行路径划分为多个路径节点. 每个路径节点可以用坐标 $P_{ij}=(x_{ij}, y_{ij}, z_{ij})$ 来表示. 飞行线路中相邻两个节点的欧式距离用 $\|P_{ij}P_{i,j+1}\|$ 表示. 因此, 在本研究中的无人机距离成本函数的数学计算式如下:

$$F_1(X_i) = \sum_{j=1}^{n-1} \|P_{ij}P_{i,j+1}\| \quad (1)$$

其中, n 代表将飞行路径划分成的节点个数, 第一个节点代表飞行路径的始发点, 第 n 个节点代表飞行路径的目的地. X_i 表示无人机的飞行路径即需要飞过的 n 个节点的集合.

2.2 安全成本和可行性约束

在执行任务时, 无人机的飞行安全是重中之重. 在飞行过程中, 无人机需要跨越诸多障碍物不发生碰撞, 在本研究中假设每一个障碍物都用一个圆柱体来表示, 令 K 为所有障碍物的集合, 障碍物投影的中心坐标为 C_k , 半径为 R_k . 图 1 显示了从节点 P_{ij} 到节点 $P_{i,j+1}$ 途经的障碍物区域图.

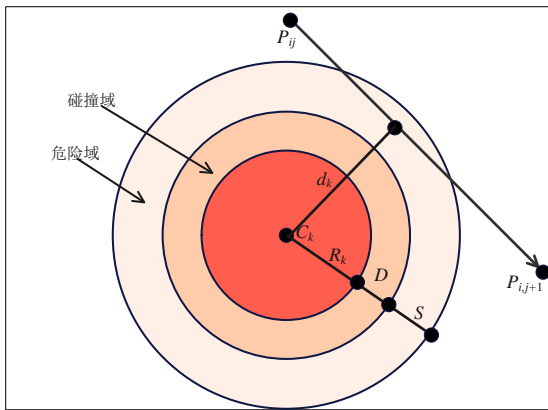


图1 无人机途经障碍物投影图

D 表示无人机的直径, S 表示无人机与碰撞域之间的安全距离, d_k 表示无人机飞行路径到障碍物中心的直线距离. 对于障碍物集合 K 而言, 无人机从节点 P_{ij} 出发到节点 $P_{i,j+1}$ 的安全成本函数的数学模型如下:

$$F_2(X_i) = \sum_{j=1}^{n-1} \sum_{k=1}^K T_k(P_{ij}P_{i,j+1}) \quad (2)$$

$$T_k(P_{ij}P_{i,j+1}) = \begin{cases} 0, & \text{if } d_k > S + D + R_k \\ (S + D + R_k) - d_k, & \text{if } D + R_k < d_k \leq S + D + R_k \\ \infty, & \text{if } d_k \leq D + R_k \end{cases} \quad (3)$$

2.3 飞行高度成本函数

当无人机在执行任务时, 选择合适的飞行高度是至关重要的. 例如, 使用无人机进行搜索任务时, 无人机所拍摄的照片清晰度与飞行高度有着很大的关系. 因此, 在本文研究中, 将无人机的飞行高度限制在一定的范围内. 本文所研究的无人机路径规划问题是最小化问题, 若无人机的飞行高度超出预设的界限, 则通过增大飞行高度成本函数的值从而对其进行修正. 飞行高度成本函数的数学表达式如下:

$$F_3(X_i) = \sum_{j=1}^n H_{ij} \quad (4)$$

$$H_{ij} = \begin{cases} \left| h_{ij} - \frac{(h_{\max} + h_{\min})}{2} \right|, & \text{if } h_{\min} \leq h_{ij} \leq h_{\max} \\ \infty, & \text{otherwise} \end{cases} \quad (5)$$

其中, h_{ij} 是无人机在飞行过程中距离地面的距离, h_{\max} 和 h_{\min} 表示距离地面飞行高度范围的最大和最小值.

2.4 平滑度成本函数

平滑成本用于评估所生成可行飞行路径的转弯率和爬升率. 图 2 展示了无人机从节点 P_{ij} 到节点 $P_{i,j+2}$ 飞行路径在三维坐标系下的示意图. P'_{ij} 、 $P'_{i,j+1}$ 和 $P'_{i,j+2}$ 分别是路径点 P_{ij} 、 $P_{i,j+1}$ 和 $P_{i,j+2}$ 在 XOY 平面上的投影点. 将线段 $P_{ij}P_{i,j+2}$ 投影到 XOY 平面上, 线段 $P'_{ij}P'_{i,j+2}$ 的延长线与线段 $P'_{i,j+1}P'_{i,j+2}$ 所形成的夹角记作 ϕ_{ij} . i 、 j 和 k 分别是 X 、 Y 和 Z 轴上的单位向量. ϕ_{ij} 的计算式如下:

$$\phi_{ij} = \arctan \left(\frac{\|P'_{ij}P'_{i,j+1} \times P'_{i,j+1}P'_{i,j+2}\|}{P'_{ij}P'_{i,j+1} \cdot P'_{i,j+1}P'_{i,j+2}} \right) \quad (6)$$

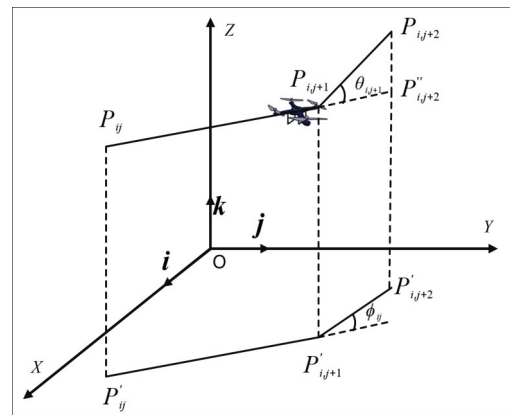


图2 三维坐标系下无人机飞行路径图

$P''_{i,j+2}$ 是路径点 $P_{i,j+2}$ 在线段 $P_{ij}P_{i,j+1}$ 所在平面上的投影点. 线段 $P_{i,j+1}P_{i,j+2}$ 与线段 $P_{i,j+1}P''_{i,j+2}$ 所形成的夹角记作 $\theta_{i,j+1}$, 其计算式如下:

$$\theta_{i,j+1} = \arctan \left(\frac{z_{i,j+1} - z_{ij}}{\|P'_{ij}P'_{i,j+1}\|} \right) \quad (7)$$

无人机飞行路线中形成的角 ϕ 和角 θ 反映了飞行路径的转弯率和爬升率,因此平滑度成本函数的数学建模可以表示如下:

$$F_4 = a_1 \sum_{j=1}^{n-2} \phi_{ij} + a_2 \sum_{j=1}^{n-1} |\theta_{ij} - \theta_{i,j-1}| \quad (8)$$

2.5 三维无人机路径规划问题的总成本函数

综合考虑一条可行路径 X_i 的距离成本、安全约束、飞行高度和路径平滑度约束,因此可以将总成本函数建模如下:

$$F(X_i) = \sum_{k=1}^4 b_k F_k(X_i) \quad (9)$$

其中, b_k 表示第 k 个成本函数的权重系数. $F(X_i)$ 较为贴切地建立了三维无人机路径规划问题的优化目标和约束,这为本文后续使用元启发式算法来优化三维无人机路径规划问题提供了充分的条件.

2.6 基于球面向量的三维无人机路径规划方法

基于球面矢量的三维无人机路径规划方法将每条可行路径 X_i 从始发点到终点编码成一组矢量. 在该方法中,每个向量具体描述了无人机如何从一个路径节点到飞行到下一个路径节点的过程. 此方法设计矢量在球面坐标系中可以具体由以下三个向量组成: $\rho \in (0, \text{path_length})$, $\theta \in (-\pi/2, \pi/2)$ 和 $\phi \in (-\pi, \pi)$. ρ 表示飞行幅度, θ 表示无人机的飞行仰角, ϕ 表示无人机方位角. 一条具有 N 个节点的可行飞行路径 Ω_i 可以由以下 $3N$ 维的球面向量表示:

$$\Omega_i = (\rho_{i1}, \theta_{i1}, \phi_{i1}, \rho_{i2}, \theta_{i2}, \phi_{i2}, \dots, \rho_{in}, \theta_{in}, \phi_{in}) \quad (10)$$

基于球面矢量的路径规划方法的基本原理是利用飞行过程中的幅度、仰角和方位角与无人机的速度、转角和爬升角之间的关系来提高规划路径的安全性和有效性. 例如,无人机的速度矢量在水平面上的投影反映了其飞行幅度,速度矢量与水平面间的夹角就是仰角,通过改变速度矢量在垂直方向上的分量就可以改变无人机的仰角. 可以通过调整无人机的仰角和爬升角来躲避障碍物,通过调整幅度和转角来实现无人机的转向等等. 这种方法可以帮助元启发式算法在配置空间而不是笛卡尔空间进行搜索,因此它可以有效地减少算法的搜索空间,从而提高算法发现更好的飞行路线的能力.

3 人工兔优化算法

人工兔优化算法起源于兔子在自然界中的生存策略. ARO 算法中主要有迂回觅食和随机躲藏两种候选解更新策略. 为了防止洞穴被天敌发现,兔子会迂回到远离洞穴的地方进行觅食. 狡兔三窟,兔子会在洞穴周围挖掘许多洞穴,当天敌来袭时兔子会随机躲藏进其中一个洞穴中.

3.1 迂回觅食

假设 ARO 中兔子群落中的每只兔子都有自己的区域,区域内有 d 个洞穴,不同的兔子将随机访问彼此的位置进行觅食. 在迂回觅食的游荡过程中兔子进行进食,所以该策略使得每个兔子对群体中随机选择的另一个个体的位置进行扰动,迂回觅食的数学模型如下:

$$v_i(t+1) = x_j(t) + R \cdot (x_i(t) - x_j(t)) + \text{round}(0.5 \cdot (0.05 + r_1)) \cdot n_1, \quad (11)$$

$$i, j = 1, 2, \dots, N \text{ and } i \neq j$$

$$R = L \cdot c \quad (12)$$

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2} \right) \cdot \sin(2\pi r_2) \quad (13)$$

$$c(k) = \begin{cases} 1, & \text{if } k = g(l) \\ 0, & \text{else} \end{cases}, \quad (14)$$

$$k = 1, 2, \dots, d; l = 1, 2, \dots, \lceil r_3 \cdot d \rceil$$

$$g = \text{randperm}(d) \quad (15)$$

$$n_1 \sim N(0, 1) \quad (16)$$

其中, $v_i(t+1)$ 是种群中第 i 只兔子在第 $t+1$ 次迭代中的候选位置, $x_i(t)$ 是第 i 只兔子在第 t 次迭代中的位置, N 代表 ARO 算法的种群大小, d 表示所优化问题的维度, T 是最大的迭代次数, round 代表四舍五入函数, randperm 将会返回从 $1 \sim d$ 的随机排列. r_1, r_2 和 r_3 是 3 个 $(0, 1)$ 范围内的随机数, L 表示进行迂回觅食过程中的跑动长度. $N(0, 1)$ 是标准正态分布函数, n_1 是服从标准正态分布的随机数.

3.2 随机躲藏

在 ARO 的每次迭代过程中,种群中的兔子将会沿着搜索空间的每个维度在周围挖掘 d 个洞穴,随后兔子将会随机选择一个洞穴进行隐藏来躲避天敌. 第 i 个兔子挖掘的第 j 个洞穴的数学建模公式如下:

$$b_{i,j}(t) = x_i(t) + H \cdot g \cdot x_i(t), \quad i = 1, 2, \dots, N; j = 1, 2, \dots, d \quad (17)$$

$$H = \frac{T-t+1}{T} \cdot r_4 \quad (18)$$

$$n_2 \sim N(0, 1) \quad (19)$$

$$g(k) = \begin{cases} 1, & \text{if } k = j \\ 0, & \text{else} \end{cases}, k = 1, 2, \dots, d \quad (20)$$

其中, H 是隐藏系数,在随机扰动下中从 1 线性地减少到 $1/T$. 兔子在受到天敌的攻击时,兔子需要找到安全的地方进行躲藏,这种随机躲藏的数学建模如下:

$$v_i(t+1) = x_j(t) + R \cdot (r_4 \cdot b_{i,r}(t) - x_i(t)), \quad i = 1, 2, \dots, N \quad (21)$$

$$g_r(k) = \begin{cases} 1, & \text{if } k = \lceil r_5 \cdot d \rceil \\ 0, & \text{else} \end{cases}, k = 1, 2, \dots, d \quad (22)$$

其中, $b_{i,r}$ 表示兔子从它的 d 个洞穴中随机选择的一个用于躲藏的洞穴, r_4 和 r_5 是两个 $(0, 1)$ 范围内的随机数.

每一次迭代过程中,ARO 在执行完迂回觅食和随机躲藏其中之一之后,第 i 只兔子的位置使用如下计算

式进行更新:

$$x_i(t+1) = \begin{cases} x_i(t) f(x_i(t)) < f(v_i(t+1)) \\ v_i(t+1) f(v_i(t+1)) < f(x_i(t)) \end{cases} \quad (23)$$

该方程表示将第*i*只兔子的候选位置和当前位置的适应度值进行比较,并保留适应度值更好的位置.

3.3 能量收缩

在 ARO 算法中,种群中的兔子倾向于在迭代初期进行迂回觅食在迭代后期进行随机躲藏.这种切换机制与兔子本身的能量相关.因此,ARO 设计了能量收缩机制来控制算法在迂回觅食和随机躲藏两种位置更新策略之间切换,能量因子的定义如下:

$$A(t) = 4 \left(1 - \frac{t}{T} \right) \ln \frac{1}{r_6} \quad (24)$$

其中, r_6 是(0,1)范围内的随机数.由式(24)可知,能量因子 $A(t)$ 在迭代过程中是振荡并且逐渐减小的.当能量因子 $A(t) > 1$ 时,兔子将会选择迂回觅食策略,反之则倾向于选择随机躲藏策略.

4 LCGARO 改进算法

4.1 Levy 飞行

Levy 分布是 20 世纪 30 年代由法国数学家莱维提出的一种概率分布. Levy 飞行是一种服从 Levy 分布的随机搜索策略,其步长具有随机性的特点,在搜索过程中,Levy 飞行频繁使用短步长偶尔使用长距离步长的跳变方式,使得 Levy 飞行更适合在大规模且未知的范围内进行搜索. Levy 分布以及 Levy 飞行步长的计算如式(25)、式(26)所示:

$$\text{Levy}(\lambda) \sim |s|^{-\lambda}, \quad 1 < \lambda < 3 \quad (25)$$

$$s = \mu / |v|^{1/\lambda}, \quad 0 < \lambda < 2 \quad (26)$$

其中, μ 和 v 代表符合正态分布的随机值, $\mu \sim N(0, \delta_\mu^2)$ 、 $v \sim N(0, \delta_v^2)$. $\delta_v = 1$, δ_μ 的计算式如下:

$$\delta_\mu = \left\{ \frac{\Gamma(1+\lambda) \sin(\pi\lambda/2)}{2^{(\lambda-1)/2} \Gamma[(1+\lambda)/2]} \right\}^{1/\lambda} \quad (27)$$

其中, Γ 代表 Gamma 函数,在本文研究中取 $\lambda = 1.5$.

在基本 ARO 算法迂回觅食阶段,每只兔子随机访问其它某一只兔子的位置,兔子的位置更新时受到随机选择的另一只兔子的位置扰动.这个随机选择的过程具有一定的盲目性和随机性,若随机选择的位置是一个局部最优的候选解,则在扰动的过程中有一定几率使得算法被局部最优的情况所吸引.针对迂回觅食策略中的不足,结合 Levy 飞行高频短步长和低频长步长跳变的特点来对兔子随机选择的方式进行改良,从而减小 ARO 在执行迂回觅食策略时被局部最优值所吸引的概率,以及增加算法的全局开发能力.在 LCGARO 算法执行完迂回觅食策略后,再使用式(28)和式(29)

对所产生的候选位置进一步扰动更新:

$$\text{Lv}_i(t+1) = v_i(t+1) + \text{Levy}(\lambda) \cdot v_i(t+1) \quad (28)$$

$$v_i(t+1) = \begin{cases} v_i(t+1) f(v_i(t+1)) < f(\text{Lv}_i(t+1)) \\ \text{Lv}_i(t+1) f(\text{Lv}_i(t+1)) < f(v_i(t+1)) \end{cases} \quad (29)$$

借鉴原始 ARO 算法中方程式(23)中提到的位置选择方式,比较 $\text{Lv}_i(t+1)$ 和 $v_i(t+1)$ 的适应度值,保留适应度值更优的候选解,从而使得改进后的迂回觅食策略能够在更新过程中偏向更优的方向.

4.2 自适应柯西变异

在 ARO 中,算法的种群多样性代表了候选解分布的广泛程度.丰富的种群多样性有助于算法在搜索空间内进行更广泛的搜索从而达到提高找到全局最优解的概率.同时,丰富的种群多样性能够防止算法早熟收敛,候选解过度聚合容易增加算法被局部极值吸引的风险.本研究将自适应柯西变异策略引入到随机躲藏过程中以增加算法种群多样性.

标准柯西分布的概率密度可以用以下函数表示:

$$f(x) = \frac{1}{\pi} \times \frac{1}{1+x^2}, \quad -\infty < x < +\infty \quad (30)$$

将柯西变异策略与自适应算子相结合,使得柯西变异在策略在早期迭代中产生较大的步长扰动,在后期逐步使用小步长对种群中的兔子位置扰动,促使算法进行局部搜索.使用自适应柯西变异策略对 ARO 算法中随机隐藏所产生的候选解进一步变异扰动,再结合式(32)引导变异结果往更优质的方向发展,自适应柯西变异策略的计算式如下:

$$\text{Cv}_i(t+1) = v_j(t+1) + v_j(t+1) \times \text{Cauchy}(0, 1) \times e^{(t-1) \times \alpha} \quad (31)$$

$$v_i(t+1) = \begin{cases} \text{Cv}_i(t+1) f(\text{Cv}_i(t+1)) < f(v_i(t+1)) \\ v_i(t+1) f(v_i(t+1)) < f(\text{Cv}_i(t+1)) \end{cases} \quad (32)$$

其中, $v_j(t+1)$ 是式(21)产生的候选位置, $\text{Cauchy}(0, 1)$ 是标准柯西分布产生的随机值. α 是收缩系数,在本文研究中取值为 0.15.使用自适应柯西变异策略对随机隐藏产生的候选位置进行扰动以增加算法的种群多样性.在寻优初期,较大的扰动步长有助于算法获得更广泛的开发能力从而对解空间进行更大范围的搜索,在寻优后期,较小的扰动促进算法在上一代候选解的周围精细搜索从而加速算法的收敛.

4.3 精英群遗传策略

遗传算法是一种经典的元启发式算法,它模拟了自然界生物的进化过程中染色体的交叉和遗传过程.目前,已有许多学者将遗传算法的思想与新颖的元启发式算法相融合以达到提高性能的目的.如 Zervoudakis 等人^[30]将遗传算法与萤火虫算法融合提出的 FAGA 算法,该算法能够有效解决产品线设计问题.通过对先前学者的研究进行学习思考,本文提出了一种精英群遗传策略对 ARO 算法进行改良.将 ARO 算法与遗传

算法中的交叉和变异算子进行融合,使得本文提出的 LCGARO 算法能够同时具有 ARO 算法和 GA 算法的优点. 在原始 ARO 算法的迂回觅食和随机躲藏的位置更新策略中,没有充分利用最优个体的位置信息来引导种群中其他兔子的位置信息进行更新迭代,这不利于算法的快速收敛. 其他个体在向最优个体靠拢的过程中,也可以同步探索周围的解空间区域,从而保持一定的种群多样性. 在本文提出的精英群遗传策略中,将兔子种群中所有个体按照适应度值从小到大排列,兔子种群中适应度值最优的 3 个个体组成精英群,3 个精英个体分别记作 $e_1 \sim e_3$. 将 3 个精英个体分别与种群中的所有个体 $r_1 \sim r_N$ 进行比较,若兔子 e_i 的适应度值优于兔子 r_j ,则对两只兔子进行基因杂交操作,反之则对两只兔子 e_i 和 r_j 进行基因变异操作.

在 ARO 算法中,一个基因的具体意义等同于兔子自己区域内的一个洞穴,属于一只兔子区域内 d 个洞穴的意义等同于一个个体有 d 个基因. 两个个体进行基因交叉操作将会产生两个新的个体 R^a 和 R^b . 基因交叉的数学建模如下:

$$R_k^a = (r_7 + 1) \cdot X_k^{e_i} + (1 - r_7) \cdot X_k^{r_j}, i \in [1, 3], j \in [1, N], k \in [1, d] \quad (33)$$

$$R_k^b = (r_7 + 1) \cdot X_k^{r_j} + (1 - r_7) \cdot X_k^{e_i}, i \in [1, 3], j \in [1, N], k \in [1, d] \quad (34)$$

其中, R_k^a 、 R_k^b 、 $X_k^{e_i}$ 和 $X_k^{r_j}$ 表示个体 R^a 、 R^b 、 X^{e_i} 和 X^{r_j} 的第 k 号基因, r_7 表示 $[0, 2]$ 范围内的随机数.

同样的,在执行基因变异过程中也会产生 R^a 和 R^b 两个新个体. 在基因变异过程中,对两个个体的一部分基因进行变异操作,这有助于提升算法种群的多样性. 在这个过程中,基因变异的概率设为 γ ,因此执行变异操作的两个个体中,每个个体将会有 $\gamma \times d$ 个基因发生变异. 基因变异操作的数学建模如下:

$$R_k^a = X_k^{e_i} + \delta \cdot \kappa, i \in [1, 3], k \in [1, \gamma \times d] \quad (35)$$

$$R_k^b = X_k^{r_j} + \delta \cdot \kappa, j \in [1, N], k \in [1, \gamma \times d] \quad (36)$$

$$\delta = 0.1 \cdot (\text{ub} - \text{lb}) \quad (37)$$

其中, ub 和 lb 表示算法搜索空间的上界和下界, κ 是一个服从正态分布的随机数.

在对所有个体执行完交叉或者变异操作之后,会产生一个新的种群记作 R ,将 R 与原始兔子种群 X 进行融合,并在保持种群规模 N 不变的情况下,保留适应度值更优的个体作为合成种群中的新个体. 在本文提出的 LCGARO 算法的每次迭代过程中,50% 的概率选择执行精英群遗传策略,50% 概率执行迂回觅食和随机躲藏策略. 精英群遗传策略充分利用种群中的精英个体对其他个体的更新进行引导,从而加快算法的收敛速度,同时变异操作也进一步丰富了算法的种群多样

性,有助于防止算法过早被局部极值所吸引.

4.4 LCGARO 算法改进思想及其伪代码

虽然 ARO 算法具有操作简单,易于理解并且超参数较少的优点,但是仍然存在收敛速度慢,易陷入局部最优的缺陷. 在原始 ARO 算法中,能量因子 A 控制着算法由迂回觅食阶段切换到随机躲藏阶段的转换过程,当能量因子 A 大于 1 时,ARO 算法进入迂回觅食阶段,反之进入随机躲藏阶段. 为了更直观的观察能量因子 A 在算法迭代过程中的变化,将其在最大迭代次数为 1 000 时的图像展示在图 3 中.

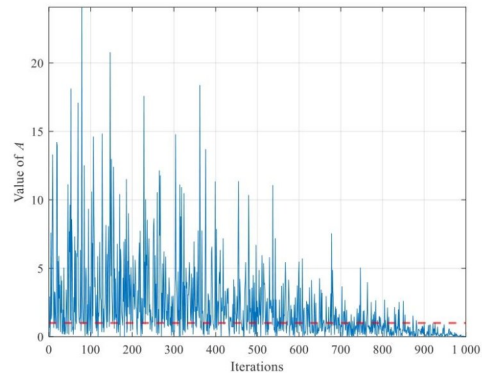


图3 能量因子 A 变化图

图 3 中红色虚线表示直线 $y=1$,从图 3 可以清楚,在算法迭代前期和中期, A 取值普遍大于 1,在迭代后期 A 取值更倾向于小于 1 的方向发展. 在 ARO 算法的迭代初期,由于种群中候选解位置的生成方式是随机且无序的,算法种群中的候选解距离理论最优解所在位置较远,特别是在复杂优化问题中,候选解与理论最优解之间的偏差则更加明显. 因此,在这个阶段中,增强算法的全局开发能力以及使用长步长来更新候选解能够显著提高其找到理论最优解的概率. 在 ARO 算法迭代前期和中期,算法更倾向于选择迂回觅食策略来更新种群中的候选解,本文使用 Levy 飞行策略对迂回觅食阶段进行改进的思想良好契合了增强其全局开发能力的目的,在改进的算法中利用 Levy 飞行策略中长短步长相结合的跳变特点来进一步更新候选解,长步长有助于算法对解空间进行更广泛的搜索以避免陷入局部最优的困境,短步长促使算法对候选解周围区域展开精细搜索以提高发现高质量解的概率. 在迭代后期,ARO 算法更倾向于使用随机躲藏策略来更新种群中的候选解,相比于初期,此时算法种群中候选解和理论最优解之间的偏差已经显著减小,因此增加算法种群多样性以及用小步长对种群中候选解位置更新过程进行扰动更能够促进算法寻找到高质量解的概率. 使用自适应柯西变异策略对随机躲藏阶段进行更新的思想很好的契合了增加算法种群多样性的目的. 将柯西变异与自适应算子相结合,使其在迭代初期对算法种群中候

求解更新产生较大的扰动,在后期自适应算子产生的随机因子较小以丰富算法的种群多样性,进一步分散算法种群中聚集在一起的候选解,从而促使算法对搜索空间进行更加精细的搜索以提高获取到更加优质解的概率.

从收敛速度角度来看,收敛速度较慢也是 ARO 算法需要进一步改善的缺陷.在 ARO 算法中,候选解的更新方式缺乏优质解的引导,质量较差的候选解不能够及时优质解靠拢,在更新过程中劣质候选解的信息容易导致新产生的候选解质量较差,从而导致算法陷入慢收敛的困境.精英群遗传算法的引入高效提升了 ARO 算法对种群中优质个体的利用率,其中,基因交叉算子利用种群中排名前三的优质候选解引导其他候选解进行更新,优质候选解的位置信息帮助劣质候选解快速向其靠拢,从而达到加速算法种群的整合以提高算法收敛速度的目的.与此同时,基因变异算子能够进一步丰富算法种群多样性以提高算法的全局开发能力.综上,将前文所介绍的三种改进策略引入 ARO 算法的不同阶段以提出 LCGARO 算法,如算法 1 所示.

算法 1 LCGARO

```

输入: 种群规模  $N$ , 维度  $d$ , 最大迭代次数  $T$ 
初始化兔子群集合  $X$ , 计算每个兔子位置的适应度值  $Fit_i$ 
WHILE
IF 产生(0,1)范围内的随机数 < 0.5
FOR 每一只兔子  $X_i$  DO
    使用式(24)计算能量因子  $A$ 
    IF  $A > 1$ 
        随机选择一只不同于  $i$  的兔子
        使用式(12)~(16)计算  $R$ 
        使用式(1)执行迂回觅食并计算候选位置的适应度值  $Fit_i$ 
        使用式(28)、式(29)执行 Levy 飞行策略
    ELSE
        使用式(21)执行随机躲藏并计算候选位置的适应度值  $Fit_i$ 
        使用式(31)、式(32)执行自适应柯西变异策略
    END IF
    使用式(23)更新当前个体
    更新目前找到的最优兔子位置  $X_{best}$ 
END FOR
ELSE
    使用式(33)~(37)执行精英群遗传策略
END IF
END WHILE
输出: 最优兔子的位置  $X_{best}$ 

```

4.5 ARO 与 LCGARO 的时间复杂度分析

时间复杂度是评价算法效率的重要衡量指标. ARO 算法与 LCGARO 算法的时间复杂度与迭代次数 T 、种群规模 N 、优化问题维度 d 及位置更新机制密切相关. 原始 ARO 算法的时间复杂度主要由初始化 $O(N)$ 、

计算个体适应度值 $O(T \cdot N)$ 、迂回觅食更新策略 $O(1/2T \cdot N \cdot d)$ 和随机躲藏更新策略 $O(1/2T \cdot N \cdot d)$ 的时间复杂度构成, 因此 ARO 算法的时间复杂度计算如下:

$$O(\text{ARO}) = O(N + T \cdot N + \frac{1}{2}T \cdot N \cdot d + \frac{1}{2}T \cdot N \cdot d) \quad (38)$$

$$= O(N + T \cdot N + T \cdot N \cdot d)$$

相同的, LCGARO 时间复杂度主要由初始化 $O(N)$ 、计算个体适应度值 $O(T \cdot N)$ 、Levy 算子改进迂回觅食更新策略 $O(1/2T(N \cdot d + N \cdot d + 1))$ 、自适应柯西变异算子改进随机躲藏更新策略 $O(1/2T(N \cdot d + N \cdot d + 1))$ 以及 50% 概率选择精英群遗传策略 $O(3T \cdot N \cdot d + T \cdot N + 2T)$ 的时间复杂度构成, LCGARO 时间复杂度计算如式(39)所示:

$$O(\text{LCGAR0}) = O(N + \frac{1}{2}(T \cdot N + \frac{1}{2}T(2N \cdot d + 1))$$

$$+ \frac{1}{2}T(2N \cdot d + 1)) + \frac{1}{2}(3T \cdot N \cdot d + T \cdot N + 2T))$$

$$= O(N + T \cdot N + \frac{5}{2}T \cdot N \cdot d + \frac{3}{2}T)$$

$$\approx O(N + T \cdot N + T \cdot N \cdot d + T) \quad (39)$$

由此可见, LCGARO 的时间复杂度相比于 ARO 而言有所增加, 当迭代次数 T 较小时, 二者时间复杂度近似相等.

5 CEC2017 测试函数对比实验

5.1 CEC2017 测试函数介绍

为了全方位评估 LCGARO 的性能, 本文基于 CEC2017 测试函数测试集设计了一系列对比实验, CEC2017 函数集的详细信息如表 1 所示. $F_1 \sim F_2$ 为单峰函数, $F_3 \sim F_9$ 为简单多峰函数, $F_{10} \sim F_{19}$ 为混合函数, $F_{20} \sim F_{29}$ 为组合函数.

5.2 实验参数设置

为了评估本文所提出的算法是否具有高效性, 本文选取了一些经典的和新型的元启发式算法设计了对比实验, 这些算法包括: 遗传算法(GA)、鲸鱼优化算法(Whale Optimization Algorithm, WOA)^[31]、白鲸优化算法(Beluga Whale Optimization, BWO)^[32]、基于莱维飞行和选择性反向学习的人工兔优化算法(LARO)、向量加权平均算法(weIghtedmeaNoFvectOrs, INFO)^[33] 和人工兔优化算法(ARO). 各个算法的参数设置如表 2 所示. 其中 GA 和 WOA 是元启发式算法中两个具有代表性的算法、BWO 和 INFO 是近期新提出的元启发式算法, LARO 是 ARO 算法的改进版本. 在本文实验中, 算法种群大小设为 30, 问题维度设为 100, 迭代次数设为 1 000, 为了避免随机性与偶然性, 每个算法将重复运行 15 次后统计数值结果. 本文实验在 Windows 11 系统下进行使用 Matlab 2020a 进行, 设备处理器为 12th Gen Intel(R) Core(TM) i5-12600KF.

表 1 CEC2017测试函数详细信息

函数编号	维度	定义域	最优值	函数编号	维度	定义域	最优值
F_1	100	[-100, 100]	100	F_{16}	100	[-100, 100]	1 600
F_2	100	[-100, 100]	200	F_{17}	100	[-100, 100]	1 700
F_3	100	[-100, 100]	300	F_{18}	100	[-100, 100]	1 800
F_4	100	[-100, 100]	400	F_{19}	100	[-100, 100]	1 900
F_5	100	[-100, 100]	500	F_{20}	100	[-100, 100]	2 000
F_6	100	[-100, 100]	600	F_{21}	100	[-100, 100]	2 100
F_7	100	[-100, 100]	700	F_{22}	100	[-100, 100]	2 200
F_8	100	[-100, 100]	800	F_{23}	100	[-100, 100]	2 300
F_9	100	[-100, 100]	900	F_{24}	100	[-100, 100]	2 400
F_{10}	100	[-100, 100]	1 000	F_{25}	100	[-100, 100]	2 500
F_{11}	100	[-100, 100]	1 100	F_{26}	100	[-100, 100]	2 600
F_{12}	100	[-100, 100]	1 200	F_{27}	100	[-100, 100]	2 700
F_{13}	100	[-100, 100]	1 300	F_{28}	100	[-100, 100]	2 800
F_{14}	100	[-100, 100]	1 400	F_{29}	100	[-100, 100]	2 900
F_{15}	100	[-100, 100]	1 500	—	—	—	—

表 2 各算法参数设置

算法	GA	WOA	BWO	LARO	ARO	INFO	LCGAR0
参数设置	$P_c=0.85, P_m=0.01, E_r=0.05$	$b=1$	$W_j \in [0.05, 0.1]$	$\beta=1.5$	—	$c=2, d=4$	$\gamma=0.05, \beta=1.5$

5.3 对比实验结果分析

表 3 统计了 7 种算法在 CEC2017 测试函数集上的平均值, 标准差和 Friedman 统计结果等数据. 从表 3 中的 CEC2017 结果来看, LCGAR0 的综合优化表现最好. 从平均值角度对各个算法开展具体分析, 在 F_{10} 和 F_{22} 函数中 LCGAR0 的寻优精度平均值略微差于 ARO 算法, 在 $F_2, F_{13}, F_{15}, F_{17}$ 和 F_{26} 函数中 INFO 算法取得了最优平均值, 在剩下的 22 个测试函数中, LCGAR0 均能够计算得到最小的收敛精度平均值. 从 29 个函数测试结果的均值角度来展开分析, LCGAR0 在优化不同种类的问题时展现出了较强的普适性. 在本次实验中 LCGAR0 表现出了比其他 6 种竞争算法更强的全局开发能力, 从而能够搜索得到更接近理论最优的解集. 值得一提的是, 在 F_1 函数的测试结果中 LCGAR0 算法的收敛精度的优势最明显, LCGAR0 的平均值比表现次优的 INFO 小了 4 个数量级, 比原始 ARO 算法得到的平均值小了 4 个数量级. 从标准差方面来看, LCGAR0 在 13 个函数的测试结果中标准差最小, 说明 LCGAR0 在一些类型的优化问题上的稳定性并不是最优, 但是在整个 CEC2017 函数集的测试结果中, LCGAR0 表现出了不错的算法鲁棒性. 表 3 的实验结果充分论证了 Levy 飞行算子长短步长相结合的跳变能够帮助 ARO 算法克服局部最优值的吸引, 从而能够

有效提高算法的寻优精度. 自适应柯西变异帮助 ARO 算法在早期迭代中产生较大的步长扰动, 在后期逐步使用小步长对 ARO 算法种群中的兔子位置扰动, 促使算法进行局部搜索, 这有助于丰富 ARO 的种群多样性, 进一步使得种群中的候选解分布更加广泛, 从而提高算法发现更优质解的能力. 在 Levy 飞行策略和自适应柯西变异策略的基础上, 结合精英群遗传策略能够更加充分地利用种群中得到的优质候选解, 前两种策略有效地提高算法找到更优质候选解的能力, 精英群遗传策略促进优质候选解与其他候选解之间的信息交流, 促使其他候选解向精英解靠拢来达到快速整合和收敛的目的, 从而加快算法的收敛速度. 实验数据表明基于以上三种策略改进得到的 LCGAR0 算法相比于 ARO 算法能够从收敛精度、收敛速度和全局开发能力等诸多方面进行改进和提高. Friedman 检验是通过秩均值来评估多个样本间是否存在显著性差异. 由于本文的问题是通过元启发式算法来求解最小化问题, 因此在 Friedman 检验中秩均值越小, 证明算法的性能越好. 从表 3 检验结果来看, LCGAR0 算法以 1.24 的秩均值稳居榜首, 其次是 INFO 算法, 其秩均值为 2.36, 排在第二位, 排在第三位的是 ARO 算法, 其秩均值为 2.53. 统计结果有效证明了 LCGAR0 寻优性能显著优于其他对比算法.

表3 CEC2017 测试结果

函数编号	统计量	GA	WOA	BWO	LARO	ARO	INFO	LCCARO
F_1	平均值	1.95E+11	6.47E+10	2.58E+11	2.41E+10	9.21E+09	2.72E+09	7.55E+05
	标准差	1.50E+10	1.10E+10	3.80E+09	4.87E+09	3.79E+09	2.40E+09	6.39E+05
F_2	平均值	2.44E+11	8.89E+05	3.61E+05	3.29E+05	3.42E+05	2.37E+05	2.53E+05
	标准差	3.86E+11	1.87E+05	2.32E+04	2.00E+04	2.56E+04	3.60E+04	1.91E+04
F_3	平均值	9.94E+04	1.15E+04	9.71E+04	3.39E+03	2.00E+03	1.14E+03	8.62E+02
	标准差	1.66E+04	2.25E+03	6.44E+03	8.03E+02	3.32E+02	1.64E+02	7.25E+01
F_4	平均值	1.75E+03	1.87E+03	2.11E+03	1.41E+03	1.25E+03	1.33E+03	1.21E+03
	标准差	6.67E+01	1.32E+02	2.61E+01	8.11E+01	8.54E+01	7.25E+01	8.41E+01
F_5	平均值	6.80E+02	7.04E+02	7.12E+02	6.60E+02	6.51E+02	6.55E+02	6.40E+02
	标准差	4.79E+00	8.13E+00	1.46E+00	4.89E+00	7.11E+00	7.07E+00	6.07E+00
F_6	平均值	3.45E+03	3.71E+03	3.89E+03	3.12E+03	2.71E+03	2.82E+03	2.32E+03
	标准差	1.21E+02	1.86E+02	4.74E+01	2.06E+02	2.71E+02	2.00E+02	1.33E+02
F_7	平均值	2.12E+03	2.34E+03	2.60E+03	1.83E+03	1.62E+03	1.69E+03	1.58E+03
	标准差	8.33E+01	1.32E+02	3.62E+01	6.03E+01	6.62E+01	1.14E+02	1.23E+02
F_8	平均值	6.60E+04	7.07E+04	8.07E+04	3.24E+04	2.78E+04	2.49E+04	2.28E+04
	标准差	6.37E+03	1.21E+04	2.30E+03	3.22E+03	2.52E+03	2.33E+03	3.18E+03
F_9	平均值	2.70E+04	2.81E+04	3.24E+04	1.82E+04	1.65E+04	1.78E+04	1.63E+04
	标准差	1.47E+03	1.80E+03	7.24E+02	1.57E+03	1.51E+03	2.20E+03	1.58E+03
F_{10}	平均值	1.18E+09	1.71E+05	3.42E+05	5.91E+04	4.34E+04	9.37E+03	6.78E+03
	标准差	2.08E+09	3.76E+04	6.16E+04	2.21E+04	8.93E+03	3.03E+03	2.61E+03
F_{11}	平均值	1.26E+11	1.20E+10	1.90E+11	1.27E+09	7.34E+08	6.21E+08	3.84E+07
	标准差	1.94E+10	2.82E+09	1.03E+10	4.55E+08	2.04E+08	1.77E+09	1.18E+07
F_{12}	平均值	3.78E+10	8.34E+08	4.36E+10	8.23E+05	1.08E+05	2.23E+04	5.86E+03
	标准差	6.77E+09	6.12E+08	2.63E+09	4.68E+05	4.83E+04	1.03E+04	3.12E+03
F_{13}	平均值	4.69E+08	1.23E+07	6.55E+07	5.76E+06	3.83E+06	8.66E+05	1.89E+06
	标准差	1.11E+08	5.27E+06	2.57E+07	2.33E+06	1.79E+06	5.03E+05	6.41E+05
F_{14}	平均值	2.05E+10	1.35E+08	2.19E+10	3.25E+04	8.45E+03	8.20E+03	2.86E+03
	标准差	3.51E+09	5.56E+07	2.33E+09	1.24E+04	3.20E+03	4.03E+03	1.23E+03
F_{15}	平均值	1.90E+04	1.54E+04	2.20E+04	6.86E+03	6.20E+03	5.91E+03	6.10E+03
	标准差	2.35E+03	1.46E+03	1.94E+03	6.47E+02	6.51E+02	5.35E+02	9.68E+02
F_{16}	平均值	1.02E+08	1.17E+04	2.79E+06	5.70E+03	5.08E+03	5.81E+03	5.04E+03
	标准差	7.57E+07	3.52E+03	1.82E+06	5.34E+02	5.82E+02	6.93E+02	4.74E+02
F_{17}	平均值	7.17E+08	1.56E+07	1.95E+08	4.55E+06	3.66E+06	1.10E+06	1.38E+06
	标准差	3.14E+08	1.15E+07	2.82E+07	2.05E+06	1.68E+06	5.19E+05	8.00E+05
F_{18}	平均值	1.82E+10	1.41E+08	2.21E+10	4.37E+04	9.28E+03	8.37E+03	4.89E+03
	标准差	4.12E+09	8.17E+07	1.90E+09	1.40E+04	4.86E+03	6.61E+03	3.44E+03
F_{19}	平均值	7.57E+03	6.68E+03	7.70E+03	5.41E+03	5.10E+03	5.64E+03	5.15E+03
	标准差	3.00E+02	4.31E+02	3.66E+02	3.82E+02	5.89E+02	4.73E+02	5.25E+02
F_{20}	平均值	4.27E+03	4.24E+03	4.68E+03	3.39E+03	3.10E+03	3.30E+03	3.02E+03
	标准差	1.26E+02	2.00E+02	9.74E+01	1.48E+02	8.85E+01	1.46E+02	9.83E+01
F_{21}	平均值	3.03E+04	3.13E+04	3.44E+04	2.14E+04	2.02E+04	2.02E+04	1.99E+04
	标准差	1.27E+03	1.22E+03	5.36E+02	1.55E+03	1.56E+03	1.93E+03	1.15E+03
F_{22}	平均值	5.77E+03	5.17E+03	6.06E+03	3.96E+03	3.60E+03	3.98E+03	3.63E+03
	标准差	2.50E+02	3.86E+02	8.52E+01	1.92E+02	1.06E+02	1.69E+02	1.58E+02
F_{23}	平均值	8.47E+03	6.51E+03	9.06E+03	5.20E+03	4.63E+03	4.80E+03	4.52E+03
	标准差	4.83E+02	4.78E+02	4.18E+02	2.89E+02	2.64E+02	2.85E+02	1.35E+02
F_{24}	平均值	2.84E+04	8.11E+03	2.66E+04	5.14E+03	4.55E+03	3.79E+03	3.53E+03
	标准差	3.93E+03	7.67E+02	1.43E+03	3.53E+02	3.22E+02	1.80E+02	6.61E+01
F_{25}	平均值	4.55E+04	3.62E+04	5.11E+04	2.66E+04	2.33E+04	2.48E+04	1.83E+04
	标准差	3.72E+03	3.71E+03	1.28E+03	2.51E+03	1.91E+03	3.56E+03	6.39E+03

续表

函数编号	统计量	GA	WOA	BWO	LARO	ARO	INFO	LCGAR0
F_{26}	平均值	9.36E+03	5.60E+03	1.21E+04	4.62E+03	4.23E+03	4.02E+03	4.18E+03
	标准差	1.05E+03	5.43E+02	7.52E+02	3.34E+02	1.59E+02	1.98E+02	2.30E+02
F_{27}	平均值	2.91E+04	1.11E+04	2.73E+04	6.51E+03	5.89E+03	3.95E+03	3.65E+03
	标准差	3.32E+03	1.14E+03	8.15E+02	6.97E+02	6.04E+02	2.88E+02	4.61E+01
F_{28}	平均值	1.87E+07	1.86E+04	3.84E+05	8.79E+03	7.86E+03	8.20E+03	7.34E+03
	标准差	9.51E+06	3.24E+03	1.22E+05	6.28E+02	4.93E+02	7.70E+02	5.17E+02
F_{29}	平均值	3.32E+10	1.43E+09	3.80E+10	1.03E+07	4.79E+06	4.05E+05	9.88E+04
	标准差	5.58E+09	6.99E+08	5.29E+09	9.89E+06	3.51E+06	2.28E+05	6.87E+04
Friedman	秩均值	6.07	5.25	6.66	3.86	2.53	2.36	1.24
	排名	6	5	7	4	3	2	1

基于各算法在 CEC2017 测试集上的收敛曲线展开分析能够进一步探讨 LCGAR0 的优化性能. 为了避免冗长, 本文在 CEC2017 测试集的不同类型的测试函数中选取了部分函数的收敛曲线展示在图 4 中. 从各算法在不同类型的测试函数上的收敛曲线可以明显看出 LCGAR0 具有更出色的收敛性能. 在 F_1 函数中, GA、BWO 和 WOA 算法均陷入了不同程度的局部最优情况中, LCGAR0 展现出了更强的全局开发能力从而能够找到精度更高的解. 在 F_3 、 F_{12} 、 F_{18} 、 F_{25} 和 F_{27} 函数中, LCGAR0 不仅能够快速收敛而且还能够寻找到适应度值更小的解, 在其余函数中 LCGAR0 算法的收敛速度略有下降, 但是在收敛精度和收敛速度方面相比于 INFO 等竞争算法仍然具有一定优势. 将 LCGAR0 和 ARO 进行对比可以清楚发现, 在所列出的 9 个测试函数收敛曲线中, LCGAR0 的能够有效的避免被局部极值所吸引, 并且收敛速度有显著的提升. 具体来看, 在 F_3 和 F_{12} 中, LCGAR0 算法能够在 400 次迭代次数左右收敛到最优候选解附近, ARO 算法在 800 次迭代次数才完成收敛, 在 F_{18} 中 LCGAR0 算法能够在 200 次左右的迭代次数之内完成收敛, 而 ARO 算法需要在 800 次迭代时才能够完成收敛, 在 F_{25} 和 F_{27} 函数中 LCGAR0 算法能够在 300 次和 600 次的迭代次数完成收敛, 而 ARO 算法在 F_{25} 和 F_{27} 中需要 600 次和 800 次迭代才能收敛到算法中最优候选解附近. 收敛曲线图更加清晰明了地说明本文提出的 Levy 算子和自适应柯西变异算子能够改善 ARO 算法的全局开发能力并且丰富算法种群多样性, 从而增加了算法发现更优质解的概率. 此外, 精英群遗传策略能够帮助算法使用种群中的优质候选解来带动和引导其他候选解的更新, 从而显著提高了本文算法的收敛速度.

5.4 统计检验分析

为了更加全面地对实验数据展开分析和对比, 本文使用 Wilcoxon 秩和检验和 Friedman 检验从统计学角度来分析 LCGAR0 算法和对比算法的性能差异. Wilcoxon 秩和检验中显著性水平 α_1 为 0.05, 在 Wilcoxon 方

法的检验结果中, 若 p 值小于 0.05 则说明两个算法的实验结果之间存在显著性差异, 若 p 值大于 0.05 则说明两个算法之间的差异在统计上不显著. Wilcoxon 方法的检验结果如表 4 所示.

从表 4 可以清晰看出, 在 LCGAR0 与 GA、WOA 和 BWO 的 Wilcoxon 统计结果中, p 值均小于 0.05, 在 LCGAR0 与 LARO 中的 Wilcoxon 统计结果中, 仅在 F_{19} 函数中 p 值大于 0.05. 在 LCGAR0 与 ARO 中的 Wilcoxon 统计结果中在 20 个函数中 p 值小于 0.05, 在 LCGAR0 算法与 INFO 算法的统计结果中有 22 个函数的统计结果 p 值小于 0.05. 因此, 从统计学角度来分析 LCGAR0 与其他算法相比, 优势较为显著.

6 LCGAR0 三维无人机路径规划问题

元启发式算法是无人机路径规划问题的一类优质的解决方案. 本文使用 LCGAR0、GA、BWO、WOA、LARO、ARO 和 INFO 等元启发式算法为无人机在三维地形环境规划出可行路径. 为了验证 LCGAR0 在三维无人机路径规划问题比其他算法有着更强的搜索和优化性能, 本文设计了 6 种复杂度不同的地形场景, 并设计了一系列对比实验用于比较 LCGAR0 与其他算法之间的差异性. LCGAR0 等算法将被用于优化 2.5 节中设计的总成本函数方程式 (9). 各种算法在优化过程中均基于球面矢量的规划方法进行搜索与优化, 将每条搜索的路径编码成一组由幅度、仰角和方位角构成的向量组并在配置空间内进行搜索, 并规划出一条总成本函数最低的飞行路径.

6.1 实验参数设置

在本次三维无人机路径规划实验中, 各个算法元启发式算法的种群大小设置为 100, 最大迭代次数设置为 200, 为了避免偶然性影响实验结果, 每个地形场景的对比实验均重复进行 15 次. 三维无人机路径规划问题中涉及的模型参数以及实验参数展示在表 5 中. 6 个不同地形场景中的障碍物信息数据列在表 6 中.

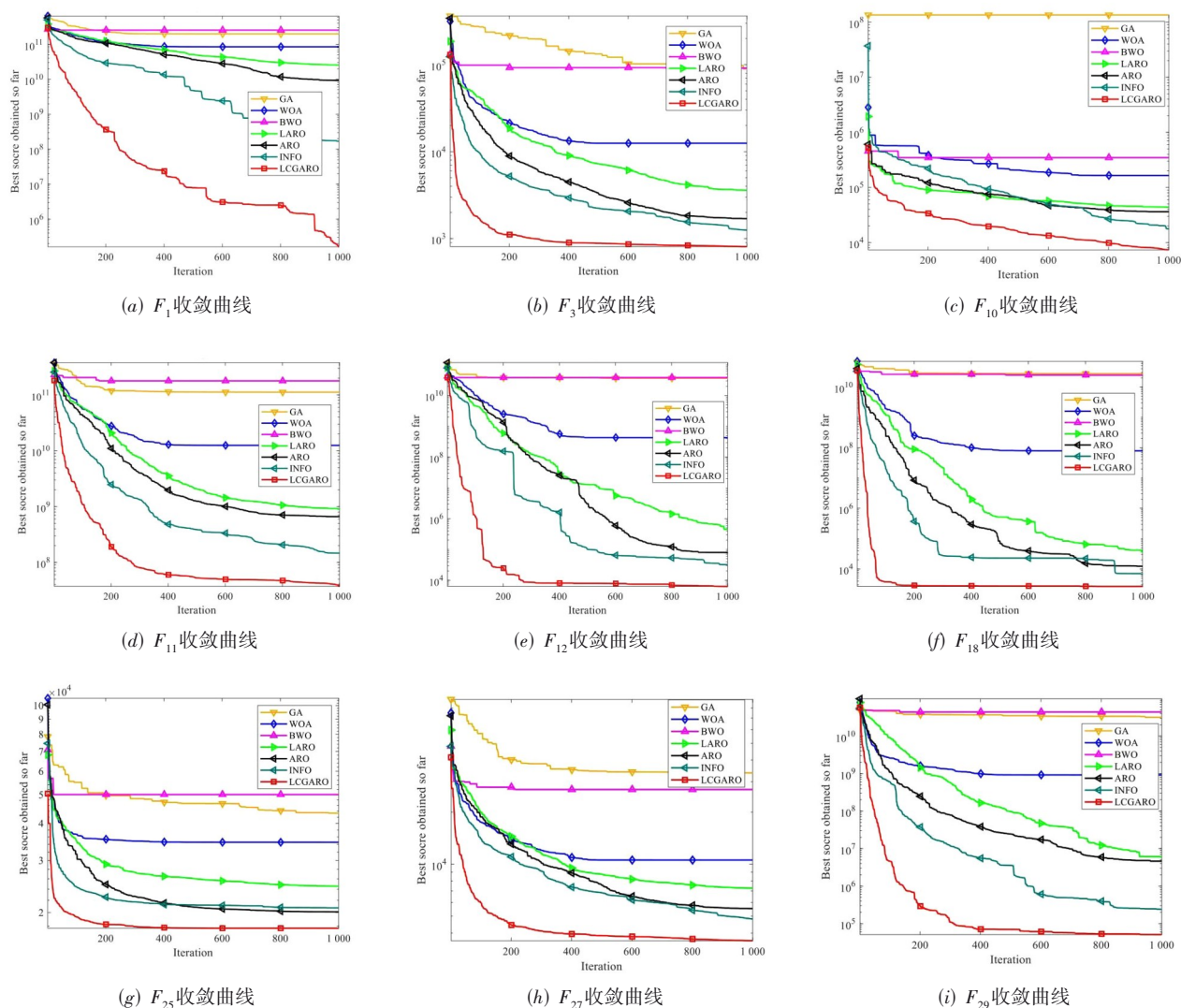


图4 部分CEC2017测试函数收敛图

6.2 三维无人机路径规划比较实验

在6种不同地形场景下重复15次实验, LCGARO等算法规划所得路径的总成本函数平均值, 标准差以及Friedman统计检验结果等数据展示在表7中. Wilcoxon秩和检验能够验证多个样本之间是否有显著性差异, 在无人机路径规划实验中各算法的Wilcoxon秩和检验结果如表8所示.

从表7数据来看, 在6个场景里, LCGARO算法能够在其中5个地形场景规划得到具有最小适应度函数值的飞行路径, INFO算法在场景3中表现最优. 这说明本文提出的LCGARO算法在大多数场景下能够为无人机找到一条更加优质的路径, 该路径从多方面节约了无人机的飞行成本, 以及能更好地帮助无人机满足飞

行时安全限制等约束条件. 在简单场景1~3中, 除了GA算法外, 其他算法均能够规划得到比较优质的飞行路径, LCGARO算法与其他算法的差异并不大. 在复杂场景4~6中, LCGARO算法的优势逐渐显示出来. 在场景5和6中, LCGARO算法与排名第二的INFO算法之间的差值分别为474.562和852.956. 这说明LCGARO算法能够在更加复杂的地形环境中规划得到更接近全局最优的飞行路径. 从Friedman分析结果来看, LCGARO算法秩均值为1.17排名第一, 排名第二和第三的分别是INFO算法和LARO算法. 从统计学角度看, LCGARO算法在6种地形场景的路径规划性能具有一定的优势.

从表8可以看出, 除了在场景3和4中, LCGARO算

表 4 Wilcoxon 秩和检验结果

函数 编号	LCGAR0					
	vs GA	vs WOA	vs BWO	vs LARO	vs ARO	vs INFO
F_1	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06
F_2	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	2.13E-01
F_3	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	6.84E-05
F_4	3.39E-06	3.39E-06	3.39E-06	1.61E-05	2.45E-01	1.05E-03
F_5	3.39E-06	3.39E-06	3.39E-06	3.39E-06	2.23E-04	5.74E-05
F_6	3.39E-06	3.39E-06	3.39E-06	3.39E-06	2.62E-04	1.94E-05
F_7	3.39E-06	3.39E-06	3.39E-06	1.10E-05	3.40E-01	2.79E-02
F_8	3.39E-06	3.39E-06	3.39E-06	3.39E-06	1.60E-04	1.25E-01
F_9	3.39E-06	3.39E-06	3.39E-06	1.62E-03	7.72E-01	8.15E-02
F_{10}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	2.02E-02
F_{11}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	4.02E-05
F_{12}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	9.07E-06
F_{13}	3.39E-06	3.39E-06	3.39E-06	1.60E-04	5.76E-04	1.15E-04
F_{14}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	9.07E-06	3.36E-05
F_{15}	3.39E-06	3.39E-06	3.39E-06	2.25E-02	6.19E-01	5.61E-01
F_{16}	3.39E-06	3.39E-06	3.39E-06	4.21E-03	8.03E-01	2.15E-03
F_{17}	3.39E-06	3.39E-06	3.39E-06	1.94E-05	2.62E-04	5.34E-01
F_{18}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	1.87E-03	8.97E-03
F_{19}	3.39E-06	7.48E-06	3.39E-06	7.45E-02	8.03E-01	1.14E-02
F_{20}	3.39E-06	3.39E-06	3.39E-06	9.07E-06	7.02E-03	4.02E-05
F_{21}	3.39E-06	3.39E-06	3.39E-06	1.28E-02	6.48E-01	9.34E-01
F_{22}	3.39E-06	3.39E-06	3.39E-06	9.66E-05	6.19E-01	4.02E-05
F_{23}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	6.80E-02	7.02E-03
F_{24}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	1.33E-05
F_{25}	3.39E-06	3.39E-06	3.39E-06	7.48E-06	6.19E-03	5.45E-03
F_{26}	3.39E-06	3.39E-06	3.39E-06	1.05E-03	3.10E-02	8.90E-02
F_{27}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	4.14E-06
F_{28}	3.39E-06	3.39E-06	3.39E-06	1.61E-05	7.94E-03	1.62E-03
F_{29}	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	6.84E-05

表 5 无人机路径规划实验参数

参数意义	参数值
路径节点数 n	12
平滑度成本函数的权重参数	$a_1 = 1, a_2 = 1$
式(9)中各成本函数的权重参数	$b_1 = 5, b_2 = 1, b_3 = 10, b_4 = 1$
无人机飞行高度限制	100~200
无人机直径	1
无人机与障碍物间安全距离	1

法和 INFO 算法的检验结果大于 0.05 外,其他所有检验结果的 p 值均是小于 0.05 的. 这说明在大多数情况下, LCGAR0 算法所规划的路径质量显著优于其他对比算法, 在场景 3 和场景 4 中, LCGAR0 算法与 INFO 算法性能的差距不明显.

各算法在 6 个不同场景中规划路径的俯视图如图 5 所示. 由图 5 可以更加清晰地看出各算法规划的路径之间的差异. 在简单场景 1~3 中, 除了 GA 算法外, 各算法所规划路径的差异是微小的. 在复杂地形 4~6 中, LCGAR0 算法表现出了更强的全局探索能力, 在场景 4 中, 原始 ARO 算法陷入了局部最优的情况, 一直在对局部最优解的周围区域探索. 在场景 5 中 LCGAR0 算法和 BWO 算法能够规划得出更加接近全局最优的飞行路径, 同时 LCGAR0 算法能对周围局部空间精细搜索, 从而找到比 BWO 算法更优质的路径. 在场景 6 中, BWO、LARO、INFO 和 LCGAR0 算法能够规划得出较为优质的路径.

各算法在 6 个地形场景中所规划路径的 3D 视图如图 6 所示. 从飞行路径的 3D 视图来看, 各个算法在 6 个不同地形场景中规划的飞行路径没有与障碍物发生碰撞, 飞行高度符合预设的高度范围, 总体来看, 各个算法规划的飞行路径是平滑且有效的.

表 6 6 个地形场景中的障碍物信息数据

场景编号	1		2		3			
障碍物坐标	(400 300 100)	(700 300 100)	(500 500 100)	(700 400 150)	(400 200 100)	(800 500 100)	(600 350 150)	(400 500 100)
障碍物半径	70	70	80	100	70	80	80	70
场景编号	4				5			
障碍物坐标	(400 350 150)	(440 530 150)	(600 350 150)	(690 600 150)	(400 500 100)	(520 350 150)	(700 250 100)	(670 650 100)
障碍物半径	80	80	80	80	70	70	70	60
场景编号	6							
障碍物坐标	(390 500 100)	(540 350 150)	(710 250 100)	(450 200 100)	(550 710 100)	(690 520 100)	(700 686 100)	—
障碍物半径	70	70	70	60	60	60	50	—

各算法在无人机路径规划实验寻优的收敛曲线图如图 7 所示. 从各算法在无人机路径规划实验中的收敛曲线分析得出, 除了在场景 3 以外的其他场景中, LCGAR0 算法的寻优精度优于其他对比算法, 这说明

本文提出的 Levy 飞行策略和自适应柯西变异策略能够有效增强 LCGAR0 算法的全局开发能力. LCGAR0 算法的收敛速度相比于 ARO 而言有明显提高, 证明本文提出的精英群遗传策略能够使用最优候选解去引导其

表 7 各算法规划所得路径的成本函数值数据

场景编号	性能指标	GA	WOA	BWO	LARO	ARO	INFO	LCGAR0
1	平均值	7 420.621	5 148.741	5 385.945	5 135.972	5 156.556	4 683.161	4 508.831
	标准差	462.499	291.643	207.218	195.368	127.489	209.086	125.087
2	平均值	7 638.118	5 408.351	5 606.403	5 266.324	5 296.398	4 755.531	4 628.602
	标准差	647.281	357.547	246.026	241.813	156.437	104.782	175.821
3	平均值	8 333.279	6 345.887	6 782.851	6 372.757	6 214.227	5 469.449	5 647.861
	标准差	669.148	334.206	302.618	149.194	211.924	300.881	238.027
4	平均值	7 715.730	7 024.048	7 069.219	6 593.658	6 596.685	5 723.662	5 647.438
	标准差	309.527	334.493	388.476	350.544	217.309	199.275	678.572
5	平均值	7 709.358	6 539.175	6 107.241	6 043.247	6 017.429	5 328.289	4 853.727
	标准差	488.617	355.979	248.992	270.653	249.303	143.549	182.606
6	平均值	8 456.153	6 912.165	6 404.007	6 119.125	6 967.006	6 116.355	5 263.399
	标准差	571.195	694.062	306.661	353.229	503.096	479.427	735.424
Friedman	秩均值	7.00	4.83	5.50	3.50	4.17	1.83	1.17
	排名	7	5	6	3	4	2	1

表 8 无人机路径规划实验 Wilcoxon 检验结果

场景编号	LCGAR0 vs GA	LCGAR0 vs WOA	LCGAR0 vs BWO	LCGAR0 vs LARO	LCGAR0 vs ARO	LCGAR0 vs INFO
1	3.39E-06	1.94E-05	3.39E-06	3.39E-06	3.39E-06	1.61E-02
2	3.39E-06	7.48E-06	3.39E-06	9.07E-06	5.05E-06	5.45E-03
3	1.83E-04	5.83E-04	1.83E-04	1.83E-04	4.40E-04	1.86E-01
4	3.39E-06	1.10E-05	1.10E-05	4.22E-04	1.89E-04	9.67E-01
5	3.39E-06	3.39E-06	3.39E-06	3.39E-06	3.39E-06	1.10E-05
6	4.11E-05	7.82E-04	7.77E-03	2.44E-02	1.65E-04	1.88E-02

他候选解进行更新,从而加快算法的收敛.通过以上一系列实验可以证明,本文提出的 LCGAR0 算法在三维无人机路径规划问题是有效的.

6.3 改进策略的有效性分析

为了评估本文提出的三种改进算子在三维无人机路径规划问题中的有效性以及三种算子对无人机路径规划的影响,本文将仅使用 Levy 飞行策略改进的人工兔算法 LeARO、仅使用自适应柯西变异策略改进的人工兔算法 CARO 和仅使用精英群遗传策略改进的人工

兔算法 GARO 应用在无人机路径规划问题中.由于以上三种算法为改进算法以及为了避免文章篇幅过长,本文将这三种单策略改进算法应用于复杂地形场景 4~6 中.各算法在每种地形场景下重复实验 15 次,平均值和标准差数据记录在表 9 中.

结合表 7 中实验数据对表 9 中数据展开分析,可以清楚了解到,在复杂地形场景 4~6 中,LeARO 算法、CARO 算法和 GARO 算法所规划路径的成本函数平均值皆是小于 ARO 算法且大于 LCGAR0 算法所计算得到

表 9 三种单策略改进算法规划所得路径的成本函数值数据

场景编号	性能指标	LeARO	CARO	GARO
4	平均值	6 528.279	6 346.845	6 126.180
	标准差	172.455	342.389	674.326
5	平均值	5 505.236	5 975.133	5 366.018
	标准差	240.108	251.794	514.838
6	平均值	6 061.400	6 478.879	6 045.807
	标准差	388.150	491.740	1 293.940

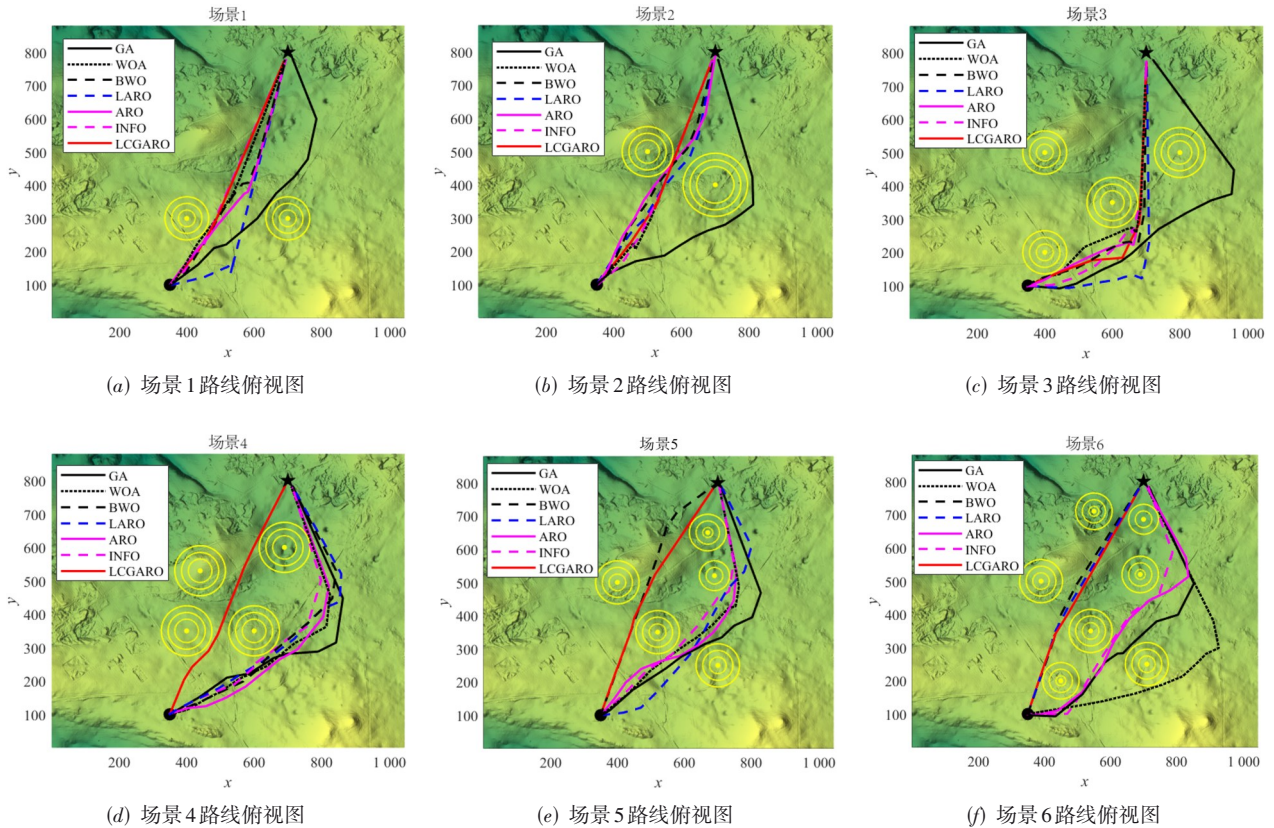


图5 各算法规划路径的俯视图

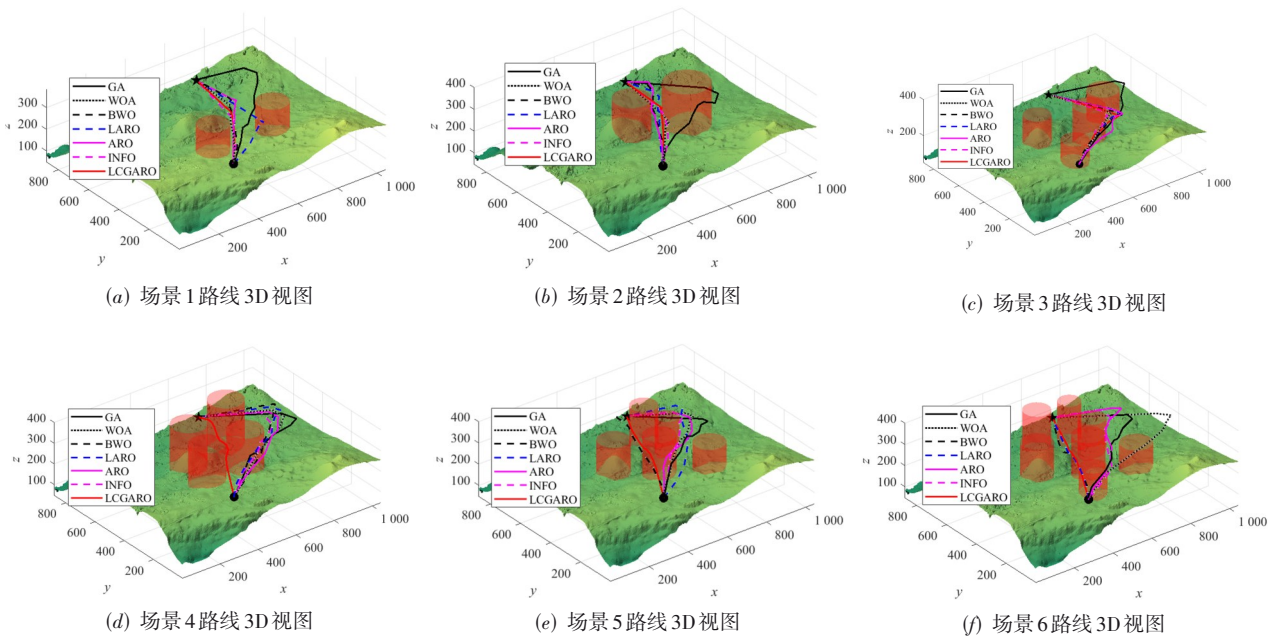


图6 各个算法规划路径的3D视图

的平均值的,这充分说明本文所提出的三种单一策略在无人机路径规划问题上对 ARO 算法的改进是有效的,与此同时,将三种策略相结合能够帮助 ARO 算法获

得更加优质的寻优性能. 针对表 9 中数据展开分析, GARO 算法在场景 4~6 中的计算得到的平均值均小于 LeARO 算法和 CARO 算法,但是 GARO 算法在场景 4~6

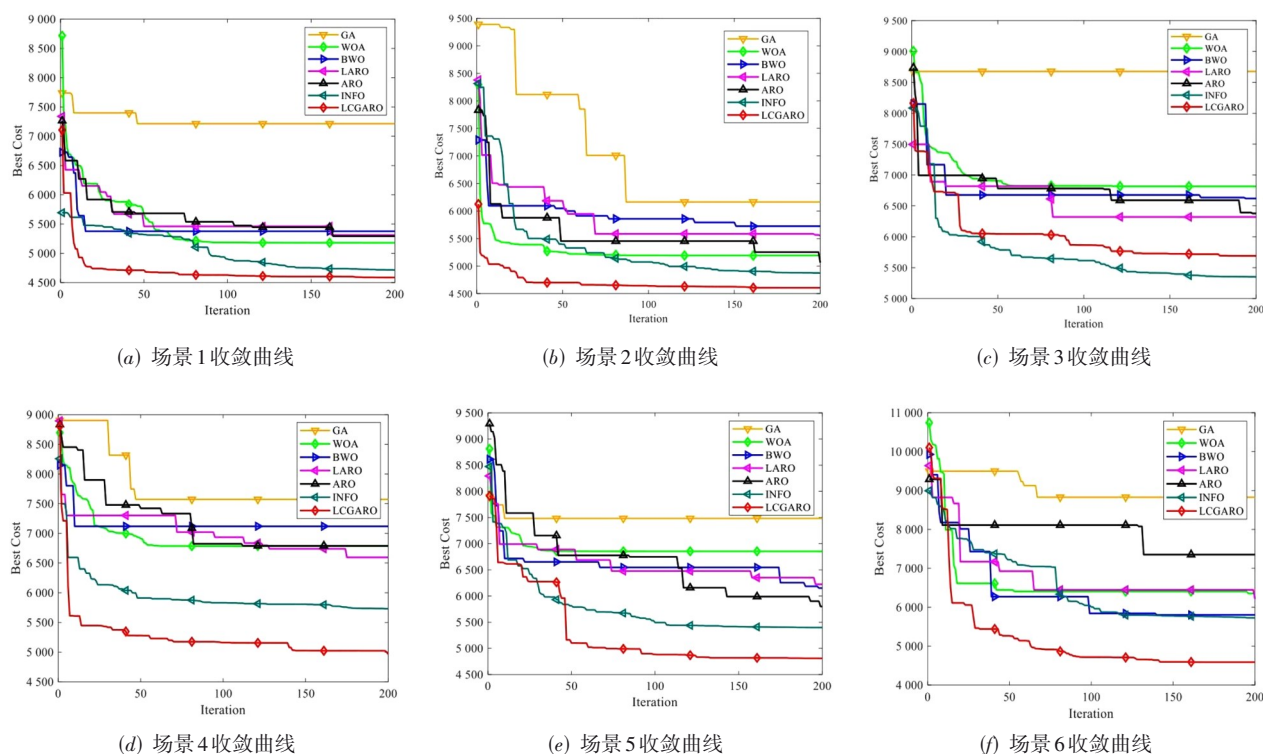


图7 各算法无人机路径规划收敛曲线

中计算得到的标准差却显著大于 LeARO 算法和 CARO 算法,这说明 GARO 算法的鲁棒性不强,这是因为精英群遗传策略对算法种群中的优质候选解有所依赖,一旦算法中出现高质量的候选解便可以快速整合并引导其他候选解进行更新,从而寻找到更接近全局最优解的解决方案. LeARO 算法在场景 4~6 中计算得到的标准差是三种算法里最小的,这说明 Levy 飞行高频短步长和低频长步长的跳变更新方式能够帮助 ARO 算法更加稳定地提高全局开发能力. 总体来看, CARO 算法相比于 ARO 算法而言寻优性能提升较小,这是因为在算法的迭代初期自适应柯西变异策略会产生较大的扰动步长,对算法的全局开发能力提升较大,在迭代后期,扰动步长会逐渐减小,这能够促进算法对局部空间展开更充分的探索,因此如果 CARO 算法在迭代后期没有找到全局优质解时,算法难以跳出局部最优的情况,规划得到的无人机飞行路径的成本函数值也会偏高,但是自适应柯西变异策略能够更好地帮助算法平衡全局开发与局部探索之间的关系. 综合来看,本文将三种策略相融合来改进 ARO 算法是有效的,Levy 飞行策略能够稳定提高算法的全局开发性能,自适应柯西变异策略能够有效平衡算法全局开发与局部探索的能力,在迭代后期促使算法对优质解的周围空间进行探索,精英群遗传策略高效地利用算法种群中的优质解来对其

他候选解进行引导更新,从而使得算法种群快速整合并加快算法收敛.

三种算法所规划路径的 3D 视图和收敛曲线如图 8 所示.

从三种算法规划路径的 3D 视图可以清晰看出,在场景 4~6 中,三种算法所规划的飞行路径均未与障碍物发生碰撞,飞行路径是平滑且有效的. 结合图 6 对图 8 展开分析,从飞行线路上来看,相对于 LCGARO 算法而言 ARO 算法所规划的路径是一条局部最优的线路,三种基于单策略改进的算法均能够规划得到更接近全局最优的飞行线路. 从三种算法的收敛曲线图上看,在场景 4 中 GARO 算法能够在 50 次迭代次数内收敛到最优候选解附近, LeARO 算法和 CARO 算法需要 100 左右和 130 次左右的迭代次数完成收敛,在场景 5 和 6 中, GARO 算法在 50 次左右和 60 次左右的迭代次数内能够收敛到最优候选解附近,而 LeARO 算法和 CARO 算法相对收敛较慢. 由此可以看出,精英群遗传策略中的交叉算子能够利用算法种群内的优质候选解引导其他候选解进行更新,从而促使其他候选解向优质解靠拢进而加快算法收敛,精英群遗传策略中的变异算子能够在一定程度上增加算法的种群多样性,从而提高算法发掘优质解的能力. 综合来看,本文提出的三种改进策略对提高 ARO 算法在无人机路径规划问题上的优化性能是有效的.

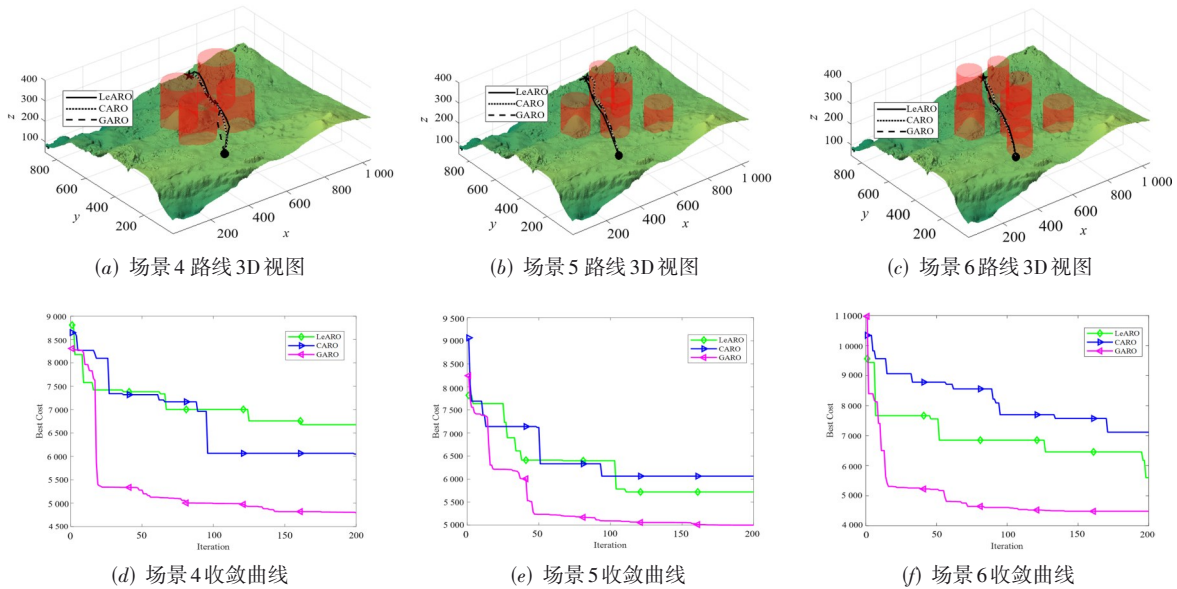


图8 三种单策略改进算法规划路径的3D视图和收敛曲线

7 结论

本文通过对三维无人机路径规划问题进行建模,构建出由飞行距离、安全因素、飞行高度和路径平滑度组合而成的成本函数进而将无人机路径规划问题转化为多约束的优化问题,并使用多种元启发式算法来优化该问题.为了改善人工兔优化算法收敛速度慢以及容易被局部极值吸引的缺陷,本文提出了一种基于多策略改进的LCGARO算法.利用Levy飞行算子长短步长相结合的跳变特点对ARO算法的迂回觅食阶段进行改进增强了算法的全局开发能力.在随机躲藏阶段引入自适应变异策略增强了算法的种群多样性从而防止算法早熟收敛.在迭代过程中按照50%概率选择精英群遗传策略进行更新,通过使用精英个体去引导算法中其他个体进行位置更新,从而达到加快算法收敛速度的目的.通过CEC2017测试函数集对LCGARO算法进行测验,测验结果证明LCGARO算法在22个CEC2017测试函数中优于其他算法.将LCGARO等算法应用在6种不同地形的无人机路径规划问题中,结果证明LCGARO算法在5种场景中能够规划出更接近全局最优的飞行路径.使用Wilcoxon秩和检验和Friedman检验从统计学角度对CEC2017和无人机路径规划的结果进行分析,结果表明LCGARO算法的性能显著优于其他算法.研究过程中发现LCGARO算法的时间复杂度相比ARO算法而言有所提高,在未来的工作中将进一步研究如何在保证LCGARO优化性能不变的前提下降低其时间复杂度.

参考文献

- [1] LIU Z, SHANG Y, LI T, et al. Robust multi-drone multi-target tracking to resolve target occlusion: A benchmark[J]. *IEEE Transactions on Multimedia*, 2023, 25: 1462-1476.
- [2] GO S H, LEE D H, NA S I, et al. Analysis of growth characteristics of kimchi cabbage using drone-based cabbage surface model image[J]. *Agriculture*, 2022, 12(2): 216.
- [3] CHENG M L, MATSUOKA M, LIU W, et al. Near-real-time gradually expanding 3D land surface reconstruction in disaster areas by sequential drone imagery[J]. *Automation in Construction*, 2022, 135: 104105.
- [4] HUANG Y, HAN H, ZHANG B, et al. Supply distribution center planning in UAV-based logistics networks for post-disaster supply delivery[C]//2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM). Piscataway: IEEE, 2021: 1-6.
- [5] HUO L, ZHU J, WU G, et al. A novel simulated annealing based strategy for balanced UAV task assignment and path planning[J]. *Sensors (Basel, Switzerland)*, 2020, 20(17): 4769.
- [6] LI B H, CHEN B D. An adaptive rapidly-exploring random tree[J]. *CAA Journal of Automatica Sinica*, 2022, 9(2): 283-294.
- [7] BANERJEE A, SUFIAN A, PAUL K K, et al. EDTP: Energy and delay optimized trajectory planning for UAV-IoT environment[J]. *Computer Networks*, 2022, 202: 108623.
- [8] LI Z J, XIA X W, YAN Y H. A novel semidefinite pro-

- gramming-based UAV 3D localization algorithm with gray wolf optimization[J]. *Drones*, 2023, 7(2): 113.
- [9] SUN Y H, CHEN W Q, LV J Y. UAV path planning based on improved artificial potential field method[C]//2022 International Conference on Computer Network, Electronic and Automation (ICCNEA). Piscataway: IEEE, 2022: 95-100.
- [10] YU J B, YANG M, ZHAO Z Y, et al. Path planning of unmanned surface vessel in an unknown environment based on improved D*Lite algorithm[J]. *Ocean Engineering*, 2022, 266: 112873.
- [11] WANG X, PAN J S, YANG Q Y, et al. Modified mayfly algorithm for UAV path planning[J]. *Drones*, 2022, 6(5): 134.
- [12] SOUZA R M J A, LIMA G V, MORAIS A S, et al. Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments[J]. *Sensors (Basel, Switzerland)*, 2022, 22(4): 1558.
- [13] ZHANG Z, WU J, DAI J Y, et al. A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment[J]. *IEEE Access*, 2020, 8: 122757-122771.
- [14] ZHANG X Y, DUAN H B. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning[J]. *Applied Soft Computing*, 2015, 26: 270-284.
- [15] MIRJALILI S. Genetic algorithm[M]//*Evolutionary Algorithms and Neural Networks*. Cham: Springer, 2019: 43-55.
- [16] PRICE K V. Differential evolution[M]//ZELINKAI, SNÁŠELV, ABRAHAMA. *Handbook of Optimization*. Berlin: Springer, 2013: 187-214.
- [17] DU K L, SWAMY M N S. Particle swarm optimization[M]//*Search and Optimization by Metaheuristics*. Cham: Birkhäuser, 2016: 153-173.
- [18] AGUSHAKA J O, EZUGWU A E, ABUALIGAH L. Dwarf mongoose optimization algorithm[J]. *Computer Methods in Applied Mechanics and Engineering*, 2022, 391: 114570.
- [19] SEYYEDABBASI A, KIANI F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems[J]. *Engineering with Computers*, 2023, 39(4): 2627-2651.
- [20] KUMAR M, KULKARNI A J, SATAPATHY S C. Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology[J]. *Future Generation Computer Systems*, 2018, 81: 252-272.
- [21] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-verse optimizer: A nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [22] WANG L Y, CAO Q J, ZHANG Z X, et al. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems[J]. *Engineering Applications of Artificial Intelligence*, 2022, 114: 105082.
- [23] ELSHAHED M, TOLBA M A, EL-RIFAIE A M, et al. An artificial rabbits' optimization to allocate PVSTATCOM for ancillary service provision in distribution systems[J]. *Mathematics*, 2023, 11(2): 339.
- [24] RIAD A J, HASANIEN H M, TURKY R A, et al. Identifying the PEM fuel cell parameters using artificial rabbits optimization algorithm[J]. *Sustainability*, 2023, 15(5): 4625.
- [25] MAZLOUMI A, POOLAD A, MOKHTARI M S, et al. Optimal sizing of a photovoltaic pumping system integrated with water storage tank considering cost/reliability assessment using enhanced artificial rabbits optimization: A case study[J]. *Mathematics*, 2023, 11(2): 463.
- [26] WANG Y Y, HUANG L Q, ZHONG J Y, et al. LARO: Opposition-based learning boosted artificial rabbits-inspired optimization algorithm with Levy flight[J]. *Symmetry*, 2022, 14(11): 2282.
- [27] WANG Y W, XIAO Y N, GUO Y L, et al. Dynamic chaotic opposition-based learning-driven hybrid Aquila optimizer and artificial rabbits optimization algorithm: Framework and applications[J]. *Processes*, 2022, 10(12): 2703.
- [28] ABUALIGAH L, YOUSRI D, ELAZIZ M ABD, et al. Aquila optimizer: A novel meta-heuristic optimization algorithm[J]. *Computers & Industrial Engineering*, 2021, 157: 107250.
- [29] PHUNG M D, HA Q P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization[J]. *Applied Soft Computing*, 2021, 107: 107376.
- [30] ZERVOUDAKIS K, TSAFARAKIS S, PARASKEVI-PANAGIOTA S. A new hybrid firefly - genetic algorithm for the optimal product line design problem[C]//International Conference on Learning and Intelligent Optimization. Cham: Springer, 2020: 284-297.
- [31] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.

- [32] ZHONG C T, LI G, MENG Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm[J]. Knowledge-Based Systems, 2022, 251: 109215.
- [33] AHMADIANFAR I, HEIDARI A A, NOSHADIAN S, et al. INFO: An efficient optimization algorithm based on weighted mean of vectors[J]. Expert Systems with Applications, 2022, 195: 116516.

作者简介



王文涛 男, 2000 年出生, 江西赣州人。南开大学软件学院软件工程专业硕士研究生。主要研究方向为智能优化算法和人工智能。
E-mail: wangwt@mail.nankai.edu.cn



叶晨 女, 2000 年出生, 江西赣州人。江西农业大学计算机与信息工程学院计算机科学与技术专业硕士研究生。主要研究方向为元启发式算法和人工智能。
E-mail: chen_ye@stu.jxau.edu.cn



田军 男, 1970 年出生, 山东淄博人。南开大学软件学院教授、博士生导师。主要研究方向为人工智能和物联网。
E-mail: jtian@nankai.edu.cn