

增强型白鲸蝠鲞融合算法机械臂插值轨迹优化方法

高永新, 刘俊毅*, 汪 洋, 贾 东

(辽宁工程技术大学机械工程学院, 辽宁阜新 123000)

摘要: 为克服传统白鲸优化算法(Beluga Whale Optimization, BWO)在3-5-3多项式插值机械臂轨迹优化中存在的路径长、时间耗费高及易陷入局部最优的问题, 本文提出了一种增强型白鲸-蝠鲞融合优化算法(Enhanced Beluga Whale and manta ray fusion Optimization algorithm, EBWO). 该算法以机械臂最优运动时间为目标, 构建约束优化模型, 并通过增广拉格朗日乘子法转化为无约束形式. 首先, 利用改进的对数非线性Halton混沌序列优化种群初始化, 提高搜索多样性与质量; 其次, 设计多方向正余弦白鲸位置更新机制, 增强开发阶段搜索能力; 再次, 在中期迭代阶段引入改进的蝠鲞旋风链式觅食策略, 并结合Levy飞行机制构建新觅食因子, 以强化局部开发与全局跳跃能力; 最后, 提出基于资源竞争耦合机制的自适应鲸落策略, 并引入量子隧穿效应, 以提升算法跳出局部最优的能力与收敛速度. 实验结果表明: 在3-5-3轨迹优化中, EBWO较于传统BWO将时间优化效果提升了8.69%, 并且与未优化的轨迹相比, 优化后的时间缩短了42.13%. 这一结果验证了其在复杂优化任务时的有效性与实用性.

关键词: 白鲸优化算法(BWO); Halton混沌序列; 多方向正余弦抢食白鲸; 旋风链式觅食; 量子隧穿; 机械臂轨迹

基金项目: 国家科学信息技术部研究中心“十四五”全国科学技术发展研究规划重点课题(No.KXJS71057); 农业部“十四五”国家科技支撑计划重点课题(No.NYFZ51050); 教育部“十四五”教育科研规划重点课题(No.JXKY24391)

中图分类号: TP242; TP18

文献标识码: A

文章编号: 0372-2112(2025)08-2915-21

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

DOI: 10.12263/DZXB.20250355

The Enhanced Whale Manta Ray Fusion Algorithm for Robotic Arm Trajectory Optimization Method

GAO Yong-xin, LIU Jun-yi*, WANG Yang, JIA Dong

(School of Mechanical Engineering, Liaoning Technical University, Fuxin, Liaoning 123000, China)

Abstract: To overcome the issues of path length, high time consumption, and the tendency to get trapped in local optima in the traditional beluga whale optimization algorithm (BWO) for 3-5-3 polynomial interpolation robotic arm trajectory optimization, this paper proposes an enhanced whale-manta ray fusion optimization algorithm (EBWO). The algorithm aims to optimize the robotic arm's motion time, constructing a constrained optimization model, which is then converted into an unconstrained form using the augmented Lagrangian multiplier method. Firstly, an improved logarithmic nonlinear Halton chaotic sequence is used to optimize population initialization, enhancing search diversity and quality. Secondly, a multi-directional cosine whale position update mechanism is designed to strengthen the search ability in the exploitation phase. In the mid-iteration stage, an improved manta ray whirlwind chain hunting strategy is introduced, combined with a Levy flight mechanism to build a new hunting factor, enhancing both local exploitation and global jumping abilities. Lastly, an adaptive whale fall strategy based on a resource-competition coupling mechanism is proposed, incorporating quantum tunneling effects to improve the algorithm's ability to escape local optima and convergence speed. Experimental results show that EBWO improves time optimization by 8.69% over traditional BWO in 3-5-3 trajectory optimization and reduces time by 42.13% compared to the non-optimized trajectory, demonstrating its effectiveness and practicality in complex optimization tasks.

Key words: beluga whale optimization algorithm (BWO); Halton chaotic sequence; multi-directional sine-cosine competitive beluga; cyclone chain foraging; quantum tunneling; robotic arm trajectory

Foundation Item(s): National Key Research and Development Program of the 14th Five-Year Plan (2021—2025)

under the Ministry of Science and Technology (No.KXJS71057); National Key Research and Development Program of the 14th Five-Year Plan (2021—2025) under the Ministry of Agriculture and Rural Affairs (No.NYFZ51050); Key Research and Development Project of the 14th Five-Year Plan (2021—2025) under the Ministry of Education (No.JXKY24391)

1 引言

随着智能制造与自主移动机器人技术的发展,机械臂轨迹优化成为机器人领域的核心问题^[1,2].在许多应用中,机械臂需沿预定轨迹执行任务,缩短运动时间对提升生产效率至关重要.因此,如何高效快速地到达目标点成为研究热点^[3,4].时间最优轨迹规划作为核心方法之一,旨在满足运动规律与约束条件的前提下,最大限度地缩短运行时间^[5,6].

近年来,多策略融合在提升群体智能算法性能方面取得了显著进展.将群体智能算法应用于机械臂最优轨迹求解已成为有效手段,通过优化机械臂轨迹显著缩短运动时间,提高作业效率^[7].文献[8]提出了一种引入混沌映射、动态非线性收敛因子、贪婪选择等策略的改进蜣螂算法,在3-5-3机械臂轨迹上减少30%运行时间.文献[9]提出了一种结合莱维飞行和自适应阈值策略的金枪鱼群算法,优化了6自由度机械臂轨迹,提升了寻优精度和跳出局部最优解能力.文献[10]提出了一种混合蜜獾算法优化3-5-3轨迹,验证路径长度和关节时间减少.文献[11]提出了一种自适应混沌麻雀算法优化4-5-4机械臂轨迹,提高了收敛精度和速度.文献[12]设计了一种基于终端交叉和导向的扰动粒子群算法来优化5-7-5轨迹,降低了运动时间.文献[13]通过简化粒子群算法优化机械臂轨迹,将时间缩短了19%.文献[14]提出了一种多策略改进沙猫群算法优化5-7-5轨迹,将时间缩短了42.7%.文献[15]提出了一种改进粒子群算法来优化3-5-3轨迹,实现关节运动平滑.文献[16]提出了一种增强型白鲸优化算法(Beluga Whale Optimization, BWO),通过自适应高斯和反向学习策略提高了跳出局部最优解的能力.文献[17]提出了一种改进鲸鱼优化算法,利用惯性权重因子和强化学习生成平滑轨迹.以上文献利用不同的仿生智能算法对多项式插值函数进行机械臂轨迹优化.

BWO凭借鲸落机制和平衡的探索-开发能力^[18],在机器人领域展现应用潜力.文献[19]将改进的白鲸算法应用于锂电池SOC估计,文献[20]将其与时间弹性带结合应用于机器人路径规划,文献[21]应用于分布式发电多目标优化.然而,经典BWO在高维非凸优化中存在种群多样性不足、局部搜索精度有限和易陷入早熟收敛等缺陷,限制了其工程实践中的适用性.

为克服传统BWO在机械臂轨迹优化中的不足,本文提出了一种增强型白鲸-蝠鲼融合优化算法(Enhanced Beluga Whale and manta ray fusion Optimization

algorithm, EBWO).首先,通过改进的对数非线性Halton混沌序列优化初始化,提升种群质量与遍历性;其次,设计多方向正弦弦抢食型白鲸位置更新,丰富开发阶段更新机制;再次,在迭代中期引入改进型蝠鲼旋风链式觅食策略,并结合Levy飞行机制增强局部开发与全局跳跃性;最后,改进鲸落概率为耦合资源与竞争的形式,构建自适应尺度因子的隧穿效应,提升跳出局部最优解能力并加速收敛.仿真与实物平台验证了该算法在机械臂轨迹优化中的优越性与实用性.

2 机械臂轨迹数学模型

2.1 机械臂运动学参数

串联机械臂由多个刚性连杆和旋转关节组成^[22,23],建立其运动学模型是精确控制运动轨迹的基础.

本文采用实验室开源六自由度机械臂系统,具备末端位姿精确控制能力,如图1所示.



图1 机械臂实物模型

D-H (Denavit-Hartenberg)法通过齐次变换矩阵描述关节位姿,将机械臂几何转化为坐标变换以求末端位姿,连杆D-H坐标系见图2.

本文选用改进D-H方法参数建立坐标系,六轴机械臂D-H参数如表1所示.

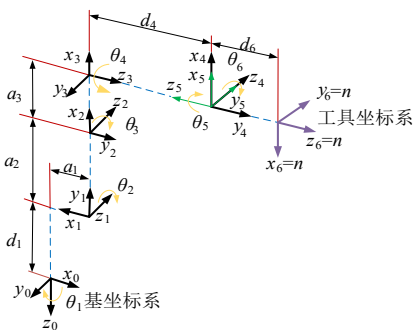


图2 连杆D-H参数坐标系

表1 改进型D-H参数表

关节编号	d/mm	$\alpha/(\circ)$	a/mm	$\theta/(\circ)$
1	169.77	-90	64.2	θ_1
2	0	0	305.0	θ_2
3	0	90	0	θ_3
4	-222.63	-90	0	θ_4
5	0	90	0	θ_5
6	-36.25	0	0	θ_6

通过查阅相关资料与以上的D-H参数表得到各关节所有运动约束,如表2所示.

表2 D-H参数关约束表

关节序号	$\theta_{max}/(\circ)$	最大角速度/ ($\text{rad}\cdot\text{s}^{-1}$)	最大角加速度/ ($\text{rad}\cdot\text{s}^{-2}$)
1	-170~170	6.0	8.0
2	-42~90	5.5	7.5
3	-90~52	5.5	7.5
4	-165~165	8.0	10.0
5	-105~105	8.0	10.0
6	-155~155	10.0	12.0

在机械臂工作空间内选取4个路径点;起点、路径点1、路径点2、终点,笛卡儿坐标如表3所示.

表3 轨迹1笛卡儿坐标

路径点	位置
起点	(364.13,0,-165.32)
路径点1	(243.37,139.10,-297.70)
路径点2	(160.16,186.02,-330.00)
终点	(0,65.44,93.22)

通过逆运动学求解起点、路径点1、路径点2、终点的关节角度,如表4所示.

表4 对应关节角度信息

关节序号	起始点/rad	路径点1/ rad	路径点2/ rad	终止点/ rad
1	0	0.523 6	0.785 4	1.570 8
2	0.253 1	0.785 4	1.047 2	-0.523 6
3	-0.275 8	-0.349 1	-0.523 6	0.785 4
4	-1.570 8	1.047 2	1.570 8	2.879 8
5	1.832 6	1.832 6	1.832 6	1.570 8
6	0	-0.523 6	-1.047 2	1.308 9

2.2 机械臂运动学模型

机械臂运动学模型将关节角度映射为末端执行器的位置和姿态,连杆*i*与*i-1*的齐次变换矩阵表示为

$$T_i^{i-1} = \text{Rot}(Z, \theta_i) \cdot \text{Trans}(Z, d_i) \cdot \text{Trans}(X, a_i) \cdot \text{Rot}(X, \alpha_i)$$

$$= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & \cos(\alpha_i) & \sin(\theta_i) & \sin(\alpha_i) & a_i \sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & \cos(\alpha_i) & -\cos(\theta_i) & \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

其中, T_i^{i-1} 为后一个关节连杆相对于前一个的位姿;Rot为旋转变换函数;Trans为平移变换函数; θ_i 为关节角度.

按顺序相乘变换矩阵,得到基座到末端执行器的整体变换矩阵*T*,即末端相对于基座的变换矩阵:

$$T_6^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

其中, $n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, p_x, p_y, p_z$ 均为矩阵中的元素,其具体式如下:

$$\begin{cases} n_x = -c_6 [s_5 c_1 s_{23} + c_5 (s_1 s_4 - c_4 c_1 c_{23})] \\ \quad - s_6 (c_4 s_1 + s_4 c_1 c_{23}) \\ n_y = s_6 (c_1 c_4 - s_4 s_1 c_{23}) \\ \quad - c_6 [s_5 s_1 s_{23} - c_5 (c_1 s_4 + c_4 s_1 c_{23})] \\ n_z = s_4 s_6 s_{23} - c_6 (s_5 c_{23} + c_4 c_5 c_{23}) \\ o_x = s_6 [s_5 c_1 s_{23} + c_5 (s_1 s_4 - c_4 c_1 c_{23})] \\ \quad - c_6 (c_4 s_1 + s_4 c_1 c_{23}) \\ o_y = s_6 [s_5 s_1 s_{23} - c_5 (s_1 s_4 + c_4 s_1 c_{23})] \\ \quad - c_6 (c_1 c_4 - s_4 s_1 c_{23}) \\ o_z = s_6 (s_5 c_{23} + c_4 c_5 s_{23}) + c_6 s_4 s_{23} \\ a_x = c_5 c_1 s_{23} - s_5 (s_1 s_4 - c_4 c_1 c_{23}) \\ a_y = c_5 s_1 s_{23} + s_5 (c_1 s_4 + c_4 s_1 c_{23}) \\ a_z = c_5 c_{23} - c_4 s_5 s_{23} \\ p_x = a_1 c_1 + d_6 [c_5 c_1 s_{23} - s_5 (s_1 s_4 - c_4 c_1 c_{23})] \\ \quad + a_3 c_1 c_{23} + d_4 c_1 s_{23} + a_2 c_1 c_2 \\ p_y = a_1 s_1 + d_6 [c_5 s_1 s_{23} + s_5 (c_1 s_4 + c_4 s_1 c_{23})] \\ \quad + a_3 s_1 c_{23} + d_4 s_1 s_{23} + a_2 c_2 s_1 \\ p_z = d_1 - a_2 s_2 - a_3 s_{23} + d_4 c_{23} + d_6 (c_5 c_{23} - c_4 s_5 s_{23}) \end{cases} \quad (3)$$

其中, $s_i = \sin(\theta_i)$; $c_i = \cos(\theta_i)$; a_i 和 d_i 为D-H参数的数据末端执行器的目标位置和姿态. 通常用齐次变换矩阵*T_d*表示:

$$T_d = \begin{bmatrix} R_d & P_d \\ 0 & 1 \end{bmatrix} \quad (4)$$

其中, \mathbf{R}_d 为 3×3 的旋转矩阵; \mathbf{P}_d 为 3×1 的目标位置向量.

通过目标姿态矩阵递推求解关节角度, 剔除不符合角度限制和奇异解的解^[24,25]. 当求解逆运动学时, 定义角度限制 $\min \Delta \theta_i, \theta_i \in [\theta_{i,\min}, \theta_{i,\max}]$, 验证并剔除不可行解, 选择最优可行解.

2.3 3-5-3 多项式插值函数轨迹模型

2.3.1 多项式插值运动学方程

在轨迹优化中, 低阶插值导致加速度不连续, 高阶插值增加计算量并引起振动^[26]. 为确保位移、速度、加速度的连续性并降低复杂度, 本文采用 3-5-3 多项式插值, 平滑连接路径点, 优化轨迹规划.

设置 4 个插值点对六自由度机械臂插值, 将路径点分为 3 个阶段, 原理图如图 3 所示.

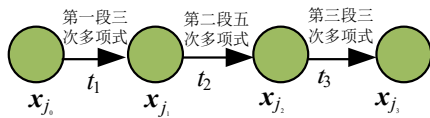


图3 3-5-3插值简易原理

$$\mathbf{A} = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\mathbf{b} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \mathbf{X}_{j_3} \ 0 \ 0 \ \mathbf{X}_{j_0} \ 0 \ 0 \ \mathbf{X}_{j_2} \ \mathbf{X}_{j_1} \ 0]^T \quad (8)$$

$$\mathbf{a} = \mathbf{A}^{-1} \cdot \mathbf{b} = [a_{j_{13}} \ a_{j_{12}} \ a_{j_{11}} \ a_{j_{10}} \ a_{j_{25}} \ a_{j_{24}} \ a_{j_{23}} \ a_{j_{22}} \ a_{j_{21}} \ a_{j_{20}} \ a_{j_{31}} \ a_{j_{32}} \ a_{j_{31}} \ a_{j_{30}}]^T$$

其中, t_1, t_2, t_3 为第 j 关节的三段多项式插值时间; \mathbf{X}_{j_i} 为第 j 个关节第 i 段插值的位移.

当四个路径点对应的关节角和每段运动时间已知时, 可以求得所有系数 \mathbf{a} .

2.3.2 时间最优模型

时间最优函数评估机械臂轨迹性能, 目标是最小化各关节运动时间, 确保机械臂在最短时间内完成轨迹^[27,28], 具体式为

$$f(t) = \min(t_1 + t_2 + t_3) \quad (9)$$

将关节角之间的变化视为关于时间的函数, 3-5-3 多项式插值函数的表达式为

$$\begin{cases} \theta_{j_1}(t) = a_{j_{13}} t^3 + a_{j_{12}} t^2 + a_{j_{11}} t + a_{j_{10}} \\ \theta_{j_2}(t) = a_{j_{25}} t^5 + a_{j_{24}} t^4 + a_{j_{23}} t^3 + a_{j_{22}} t^2 + a_{j_{21}} t + a_{j_{20}} \\ \theta_{j_3}(t) = a_{j_{33}} t^3 + a_{j_{32}} t^2 + a_{j_{31}} t + a_{j_{30}} \end{cases} \quad (5)$$

其中, $\theta_{j_1}(t), \theta_{j_2}(t), \theta_{j_3}(t)$ 为每个时间段内的插值函数; $a_{j_{1i}}, a_{j_{2i}}, a_{j_{3i}}$ 分别为第 j 关节轨迹第 1 段、第 2 段、第 3 段插值函数的第 i 个未知系数.

将轨迹添加两个路径点, 分为三段, 确保平滑性和连续性, 起终点速度和加速度均为 0, 各路径点处速度与加速度连续.

已知第 j 个关节的 4 个插值点 $\mathbf{X}_{j_0}, \mathbf{X}_{j_1}, \mathbf{X}_{j_2}, \mathbf{X}_{j_3}$, 且在 这些点处速度和加速度连续, 得到矩阵方程为

$$\begin{cases} \mathbf{A} \cdot \mathbf{a} = \mathbf{b} \\ \mathbf{a} = \mathbf{A}^{-1} \cdot \mathbf{b} \end{cases} \quad (6)$$

其中, \mathbf{A} 为约束矩阵; \mathbf{a} 为系数矩阵; \mathbf{b} 为插值点矩阵. 确定 3 段运动时间 t_1, t_2, t_3 后, 解得系数矩阵 \mathbf{a} , 并回代插值多项式, 得到关节 j 的 3 段运动学方程为

最大角加速度.

2.4 增广拉格朗日乘法优化目标函数

为解决惩罚因子导致的收敛速度问题,本文采用增广拉格朗日乘法优化,数学模型为

$$\begin{cases} \min f(x) \\ \text{s.t. } g_i(x) \leq 0 \quad i=1,2,\dots,n \end{cases} \quad (11)$$

其中, x 为自变量; $f(x)$ 为目标函数; $g(x)$ 为约束条件.

使用增广拉格朗日乘法可以转化为

$$\begin{cases} L(x, \lambda_i, r_i) = f(x) + \sum_{i=1}^n (\lambda_i \varphi_i + r_i \varphi_i^2) \\ \varphi_i = \max \left[g_i(x), -\frac{\lambda_i}{2r_i} \right] \end{cases} \quad (12)$$

其中, λ_i 为拉格朗日乘子; r_i 为惩罚因子; $L()$ 为拉格朗日函数; φ_i 为修正之后的约束函数.

通过对式(12)的求解得到原约束问题,其 λ_i 和 r_i 的更新公式为

$$\begin{cases} \lambda_i^{k+1} = \lambda_i^k + 2r_i \varphi_i \\ r_i^{k+1} = \begin{cases} 2r_i^k, & |g_i(x^k)| > |g_i(x^{k-1})| \cap |g_i(x^k)| > \varepsilon_g \\ r_i^k/2, & |g_i(x^k)| \leq \varepsilon_i \\ r_i^k, & \text{else} \end{cases} \end{cases} \quad (13)$$

其中, k 为拉格朗日乘子的迭代次数; ε_g 为约束误差精度.

当 $k=1$ 时,迭代终止条件 c 为

$$c = \max \{ \max [0, g_i(x^k)] \leq \varepsilon \} \quad (14)$$

当 $k=[2, k_{\max}]$ 时,终止条件为

$$|f(x^k) - f(x^{k-1})| \cap c \leq \varepsilon \quad (15)$$

其中, ε 为收敛精度; k_{\max} 为最大迭代次数.

该方法避免了传统惩罚函数在达到无穷大或无穷小时才能找到最优解的问题.

解算机械臂最优运动时间过程如下:

步骤 1: 给定初始值,将约束优化问题式(11)转化为无约束优化问题式(12).

步骤 2: 初始化种群,随机初始化白鲸个体的位置,且位置位于指定边界内.

步骤 3: 验证终止条件式(14). 若满足条件,则输出当前解为最优解;若不满足条件,则保留当前最优解,重新初始化个体位置,并设置 $n=1$.

步骤 4: 基于 EBWO 算法更新白鲸位置,评估每个位置的适应度值.

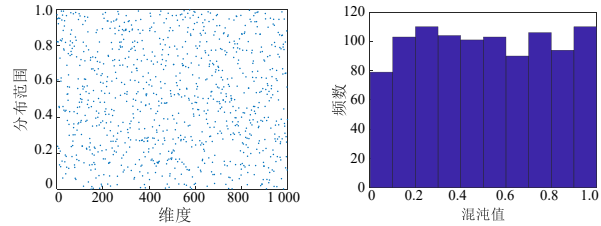
步骤 5: 验证终止条件式(15). 若满足条件,则得到最优解;若不满足条件,则根据式(13)更新拉格朗日乘子和惩罚因子,设置 $k=k+1$,返回步骤 4.

3 EBWO 算法

3.1 改进 Halton 混沌序列兼多方向正余弦抢食型白鲸位置更新

3.1.1 对数非线性 Halton 混沌序列

原始白鲸算法初期搜索空间大,但易陷入局部最优. 传统 BWO 算法使用伪随机数初始化种群,影响全局搜索能力,种群分布见图 4.



(a) 维度方向的种群分布范围 (b) 混沌值对应的频数

图 4 伪随机数生成的种群分布图

传统伪随机数生成方法在种群分布上存在聚集和中间密度偏高的问题,这会影晌全局搜索能力. 为增加多样性,文献[22,23]采用 Fuch 和 Tent 混沌映射. Fuch 映射初值不敏感、均衡遍历、快速收敛,但边界密度高; Tent 映射易陷入局部最优. 为解决此问题,本文引入改进的 Halton 混沌序列,增强初期随机性.

传统 Halton 序列通过选取两个质数作为基础量,切分生成均匀分布且不重复的点,数学模型为

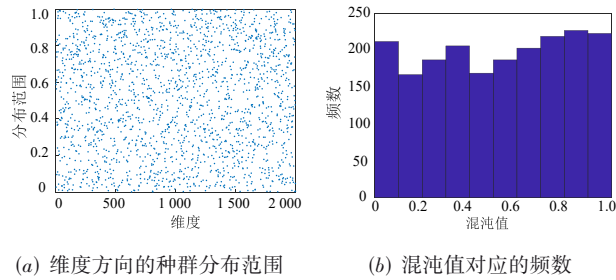
$$\begin{cases} n = \sum_{i=0}^m b_i \cdot p^i = b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p^1 + b_0 \\ \theta(n) = b_0 p^{-1} + b_1 p^{-2} + \dots + b_m p^{-m-1} \\ H(n) = [\theta_1(n), \theta_2(n)] \end{cases} \quad (16)$$

其中, n 为 $[1, N]$ 的任意整数; p 为大于等于 2 的质数,为 Halton 序列的基础量; b_i 为 $[0, p-1]$ 的常数; $\theta(n)$ 为定义的序列函数; $H(n)$ 为最后得到的二维均匀 Halton 序列.

如图 5~图 7 所示,尽管传统 Halton 序列解决了 Tent 和 Fuch 映射问题,但仍有边界密度高的问题. 为避免局部聚集,本文在 Halton 序列基础上引入对数非线性变换,增强全局覆盖能力,数学模型为

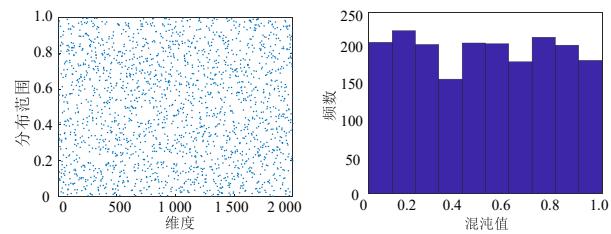
$$\begin{cases} \theta_1(n) = \sum_{i=0}^m b_i \cdot p^i \\ \theta_2(n) = \sum_{i=0}^m b_i \cdot p^{i-1} \\ H_{\text{nonlinear}}(n) = [f(\theta_1(n)), f(\theta_2(n))] \end{cases} \quad (17)$$

其中, $f(\theta)$ 为对数函数,其底数与序列的基础量设置为一致.



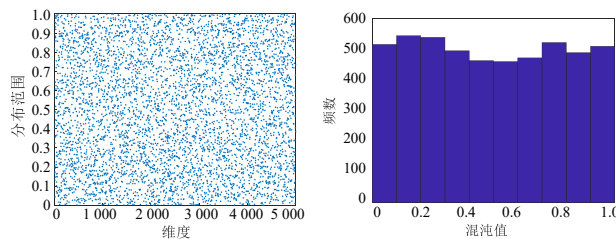
(a) 维度方向的种群分布范围 (b) 混沌值对应的频数

图5 Fuch混沌映射种群分布图



(a) 维度方向的种群分布范围 (b) 混沌值对应的频数

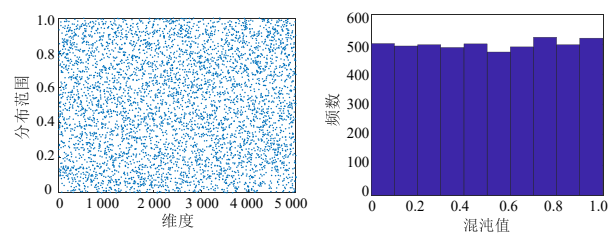
图6 Tent混沌映射种群分布图



(a) 维度方向的种群分布范围 (b) 混沌值对应的频数

图7 传统Halton混沌序列种群分布图

改进后的Halton种群分布如图8所示,解决了局部密集现象,使种群分布更加均匀且随机,其中基础量为base1=2,base2=3.



(a) 维度方向的种群分布范围 (b) 混沌值对应的频数

图8 对数非线性Halton混沌序列种群分布图

3.1.2 多方向正余弦抢食型白鲸位置更新

在白鲸算法中,勘探阶段转移到开发阶段由平衡因子决定,具体为

$$B_f = B_0 \cdot (1 - T / (2 \cdot T_{\max})) \quad (18)$$

其中, B_0 为初始平衡系数; T 为当前迭代次数; T_{\max} 为最大迭代次数.

白鲸算法探索阶段的位置更新由镜像距离和成对

白鲸共同决定,其数学模型为

$$\mathbf{X}_{i,j}^{T+1} = \begin{cases} \mathbf{X}_{i,P_i}^T + (\mathbf{X}_{r,P_i}^T - \mathbf{X}_{i,P_i}^T)(1 + r_1) \sin(2\pi r_2), & j = \text{even} \\ \mathbf{X}_{i,P_i}^T + (\mathbf{X}_{r,P_i}^T - \mathbf{X}_{i,P_i}^T)(1 + r_1) \cos(2\pi r_2), & j = \text{odd} \end{cases} \quad (19)$$

其中, $\mathbf{X}_{i,j}^{T+1}$ 为第 i 条白鲸在第 j 维更新后的位置; \mathbf{X}_{i,P_i}^T 为该维度下第 i 条白鲸的位置; P_i 为种群中随机选取的第 i 条白鲸; r_1 和 r_2 为 $(0, 1)$ 的随机数.

传统白鲸算法开发阶段的位置更新模拟白鲸的觅食行为,其数学模型为

$$\mathbf{X}_i^{T+1} = r_3 \mathbf{X}_{\text{best}}^T - r_4 \mathbf{X}_i^T + C_1 \cdot L_F (\mathbf{X}_i^T - \mathbf{X}_i^T) \quad (20)$$

其中, r_3, r_4 均为 $(0, 1)$ 的随机数; C_1 为用于衡量Levy飞行的随机跳跃程度的因子; L_F 为服从Levy飞行分布的随机数.

传统白鲸算法开发阶段忽略最差位置信息,为提高更新多样性,考虑多种信息类型可增强位置丰富性,提升全局搜索能力.

借鉴蜣螂算法偷盗行为,设计一种考虑最差位置的更新方式,数学模型为

$$\mathbf{X}_i^{T+1} = \mathbf{X}_{\text{best}}^T + S \cdot \mathbf{g} \cdot (|\mathbf{X}_i^T - \mathbf{X}_{\text{worst}}^T| + |\mathbf{X}_i^T - \mathbf{X}_{\text{best}}^T|) \quad (21)$$

其中, S 为常量; \mathbf{g} 为服从正态分布的 D 维随机向量; $\mathbf{X}_{\text{worst}}^T$ 为当前的最差个体; $\mathbf{X}_{\text{best}}^T$ 为当前的最优个体.

在上述公式中,当前位置与最差位置权重相同.为加速收敛,需为两者设置不同权重,并进行参数敏感性测试,如图9所示.

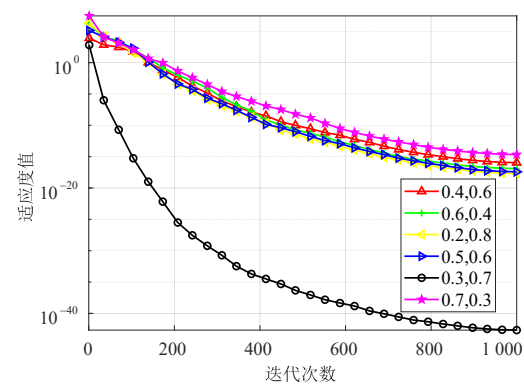


图9 不同权重系数参数敏感测试

经过参数权重实验可知,在当前位置与最差位置为0.3,最优位置为0.7时,BWO表现最佳,因此改进式变为

$$\mathbf{X}_i^{T+1} = \mathbf{X}_{\text{best}}^T + S \cdot \mathbf{g} \cdot (0.3 |\mathbf{X}_i^T - \mathbf{X}_{\text{worst}}^T| + 0.7 |\mathbf{X}_i^T - \mathbf{X}_{\text{best}}^T|) \quad (22)$$

常量 S 影响算法性能, Sphere 函数基准测试结果如图10所示,当 $S=3$ 时,算法性能最优,因此选用 $S=3$.

当前白鲸在固定方向上更新位置进行觅食,容易错过全局最优解.为解决此问题,本文在式(22)基础上加入正弦和余弦抢食行为,以提升觅食效率和全局最优解成功率.改进式为式(23),觅食示意图如图11所示.

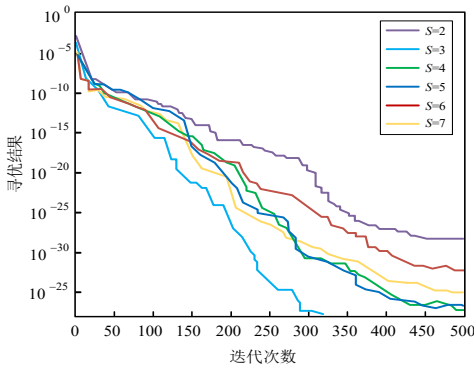


图 10 不同S常量下的性能收敛曲线

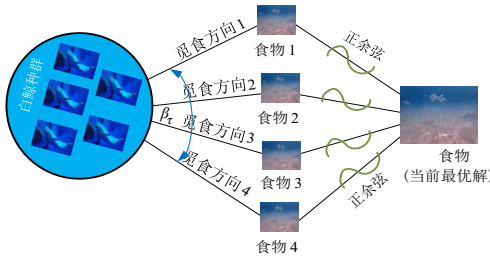


图 11 多方向正余弦抢食白鲸觅食示意图

$$X_i^{T+1} = \begin{cases} X_{best}^T + S \times g \times \gamma \times \sin(\beta_r) \\ \times (0.3 |X_i^T - X_{worst}^T| + 0.7 |X_i^T - X_{best}^T|), \text{rand} > 0.5 \\ X_{best}^T + S \times g \times \gamma \times \cos(\beta_r) \\ \times (0.3 |X_i^T - X_{worst}^T| + 0.7 |X_i^T - X_{best}^T|), \text{rand} \leq 0.5 \end{cases} \quad (23)$$

$$\gamma = 2 - \frac{2t^2}{T^2}$$

其中, γ 为非线性衰减自适应因子; β_r 为 $[0, 2\pi]$ 的感知角度; g 为服从正态分布的 D 维随机向量; rand 为 $[0, 1]$ 的任意随机数。

比较式 (23) 和式 (20) 的位置更新输出, 选择适应度最优解集作为当前个体进行下一步操作。

3.2 融合蝠鲸算法改进型旋风链式觅食机制

多方向正余弦抢食型白鲸优化算法容易陷入局部最优。为此, 本文融合蝠鲸算法的旋风链式觅食机制, 以增强局部开发能力和多样性。

蝠鲸算法的链式觅食策略数学模型为

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + r \times [X_{best,j}(t) - X_{i,j}(t)] \\ + \alpha \times [X_{best,j}(t) - X_{i,j}(t)], i = 1 \\ X_{i,j}(t) + r \times [X_{i-1,j}(t) - X_{i,j}(t)] \\ + \alpha \times [X_{best,j}(t) - X_{i,j}(t)], i = 2, 3, \dots, n \end{cases} \quad (24)$$

$$\alpha = 2r \sqrt{|\log(r)|}$$

其中, α 为链式觅食的权重因子; r 为 $(0, 1]$ 的随机量, \log 为以数学常数 e 为底的对数, 下同。

蝠鲸算法的链式觅食示意图如图 12 所示。

蝠鲸算法的旋风觅食策略数学模型为

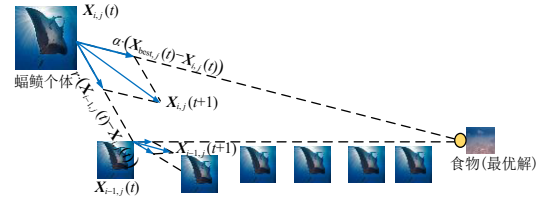


图 12 蝠鲸算法的链式觅食示意图

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + r \times [X_{best,j}(t) - X_{i,j}(t)] \\ + \alpha \times [X_{best,j}(t) - X_{i,j}(t)], i = 1 \\ X_{i,j}(t) + r \times [X_{i-1,j}(t) - X_{i,j}(t)] \\ + \alpha \times [X_{best,j}(t) - X_{i,j}(t)], i = 2, 3, \dots, n \end{cases} \quad (25)$$

$$\beta = 2 \times e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1)$$

其中, β 为旋风觅食的权重因子; T_{max} 为算法的最大迭代; t 为当前迭代次数; r 和 r_1 均为 $(0, 1]$ 随机量。

蝠鲸算法的旋风觅食示意图如图 13 所示。

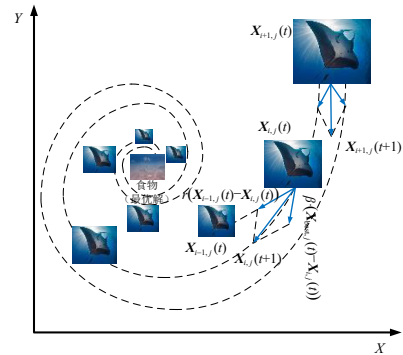


图 13 蝠鲸算法的旋风觅食示意图

链式觅食与旋风觅食更新相似, 但权重因子不同。为平衡全局与局部搜索, 本文结合两者因子, 形成新觅食因子, 并将链式因子作为旋风因子的幅值控制系数。改进模型为

$$\gamma = \alpha \times \beta = 2r \sqrt{|\log(r)|} \times 2e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1) \quad (26)$$

为增强信息交换, 本文在权重因子中引入 Levy 飞行机制, 增强重尾稳定分布。引入的权重因子为

$$\beta_2 = 2\text{Levy}(\lambda) e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1) \quad (27)$$

其中, $\text{Levy}(\lambda)$ 的数学模型为

$$L_F = \frac{\mu \cdot \sigma}{|v|^{1/\lambda}} \quad (28)$$

$$\sigma = \begin{cases} \left(\frac{\Gamma(1+\lambda) \times \sin\left(\pi \cdot \frac{\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\frac{\lambda-1}{2}}} \right)^{\frac{1}{\lambda}}, \Gamma(x) = (x-1)! \end{cases}$$

其中, $u, v \sim N(0, 1)$, 其均值为 0, 方差为 1; λ 为一个设定

为 1.5 的常数. 其最终的改进型觅食因子为

$$g = Levy(\lambda) \times \alpha \times \beta_2$$

$$= 2rLevy(\lambda) \sqrt{|\log(r)|} \times 2e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1) \quad (29)$$

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + r \times [X_{beat,j}(t) - X_{i,j}(t)] + \delta \times [X_{beat,j}(t) - X_{i,j}(t)], & i = 1 \\ X_{i,j}(t) + r \times [X_{i-1,j}(t) - X_{i,j}(t)] + \delta \times [X_{beat,j}(t) - X_{i,j}(t)], & i = 2, 3, \dots, n \end{cases} \quad \text{if } 0.3T_{max} < T_{max} < 0.7T_{max}$$

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + r \times [X_{best,j}(t) - X_{i,j}(t)] + \beta \times [X_{best,j}(t) - X_{i,j}(t)], & i = 1 \\ X_{i,j}(t) + r \times [X_{i-1,j}(t) - X_{i,j}(t)] + \beta \times [X_{best,j}(t) - X_{i,j}(t)], & i = 2, 3, \dots, n \end{cases} \quad \text{else} \quad (30)$$

$$\delta = g \times \alpha \times \gamma \times \beta_\tau$$

$$= 2rLevy(\lambda) \sqrt{|\log(r)|} \times 2e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1) \times g \times (2 - \frac{2t^2}{T^2}) \times \beta_\tau$$

$$\beta = 2 \times e^{r_1 \frac{T_{max}-t+1}{T_{max}}} \times \sin(2\pi r_1)$$

其中,参数量同上.

新型多方向旋风链式觅食示意图见图 14,应用开发阶段解集进行迭代,有效增强局部开发能力,避免局部最优.

该机制总的流程示意图如图 15 所示.

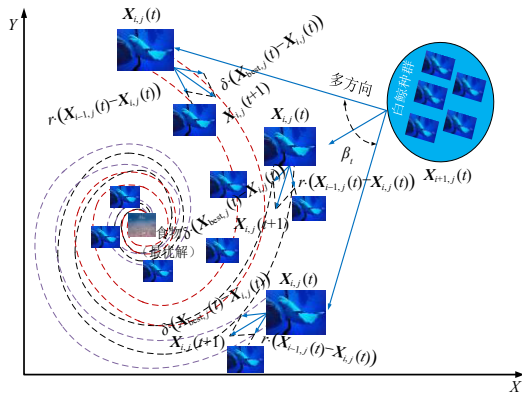


图 14 多方向旋风链式觅食示意图

将改进型旋风觅食策略与多方向因子结合,应用于前 30%~70% 迭代阶段,其余阶段采用传统旋风觅食策略. 位置更新模型为

3.3 非线性耦合概率鲸落量子隧穿

白鲸因外界威胁死亡,通过位置和下落步长更新个体,形成鲸落模型.

$$\begin{cases} X_i^{T+1} = r_5 X_i^T - r_6 X_r^T + r_7 X_{step} \\ X_{step} = (u_b - l_b) \exp\left(-\frac{C_2 \cdot T}{T_{max}}\right) \end{cases} \quad (31)$$

其中, r_5, r_6, r_7 均为 $(0, 1)$ 内的随机数; X_{step} 为白鲸下落的步长; l_b 和 u_b 为搜索空间的下界和上界; C_2 为阶跃因子,其数值由白鲸下落的概率 W_f 和种群规模 n 所决定,其具体计算式为

$$C_2 = 2 \cdot W_f \cdot n \quad (32)$$

白鲸下落的概率 W_f 数学模型为

$$W_f = 0.1 - 0.05T/T_{max} \quad (33)$$

其中, W_f 由迭代初始的 0.1 线性变化到最终的 0.05,表明了白鲸越接近食物,死亡概率越小.

传统线性鲸落概率随迭代增加减少,易导致过早收敛. 为此,本文设计耦合资源与竞争的鲸落概率.

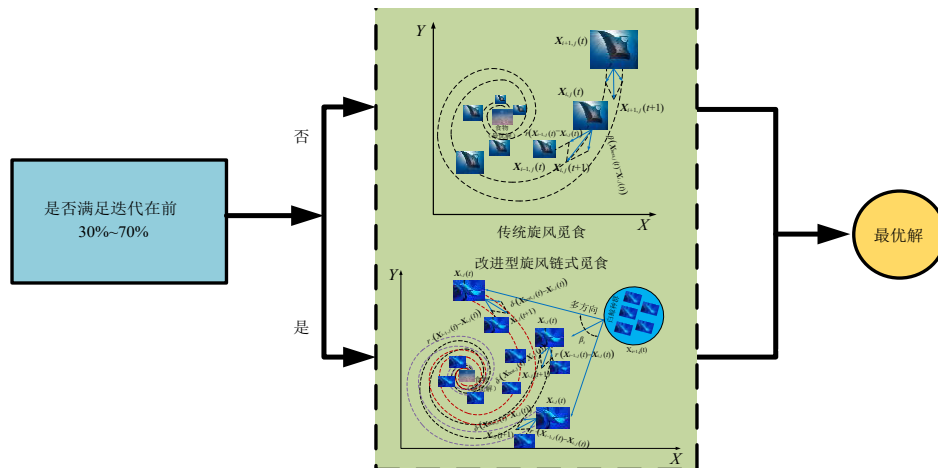


图 15 改进型旋风链式觅食机制总示意图

首先,根据当前种群平均适应度反比设计资源丰幅度 R ,具体为

$$R = \frac{1}{(\text{mean}(\text{fit}) + \text{eps})} \quad (34)$$

其中,mean 为求解平均值;eps 为极小常数,以避免分母为0.

采用种群的适应度方差表示计算竞争强度 C ,具体为

$$C = \text{var}(\text{fit}) \quad (35)$$

其中,var()为计算种群适应度方差.

为扩大后期搜索范围,将传统线性鲸落概率改为非线性正弦鲸落概率,具体为

$$W_{\rho} = 0.1 - 0.05 \times \sin\left(\frac{T}{T_{\max}} \times \pi\right) \quad (36)$$

将资源丰富度 R 和竞争强度 C 与非线性正弦鲸落耦合,形成多因素耦合鲸落概率,具体为

$$W_f = W_{\rho} \times \frac{1}{1 + e^{-k(R - R_0)}} \times \frac{1}{1 + \eta C} \quad (37)$$

其中, R_0 为资源阈值; k 为控制资源系数; η 为竞争抑制系数,参数选取需通过参数敏感实验,使用 Sphere 函数进行对比.

保持资源阈值0.5不变,测试竞争抑制系数1~5,寻优迭代如图16所示.

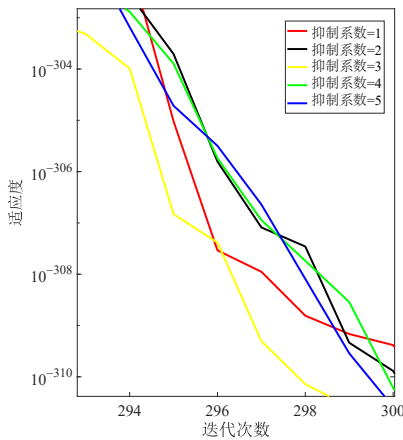


图16 竞争抑制系数参数实验

由图16可知,竞争系数为2时算法性能最佳.保持竞争系数2不变,改变资源阈值进行实验,测试值为0.3、0.4、0.5、0.7、0.9,如图17所示.

由图17可知,阈值参数为0.4时算法性能最佳.保持竞争系数2和阈值0.4不变,改变资源系数 k ,测试值为2、5、10、15、20,见图18.

通过以上的参数敏感性实验最终选取阈值参数选为0.4,竞争系数选为2,控制系数选为15.

为缓解局部最优问题,本文引入量子隧穿效应的

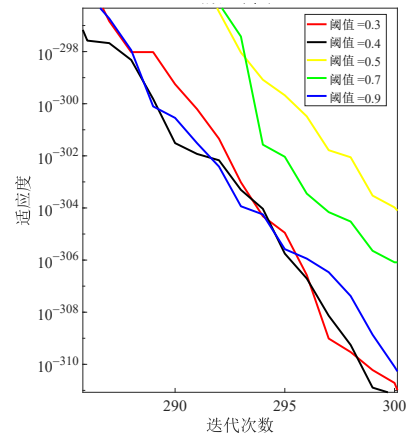


图17 阈值参数实验

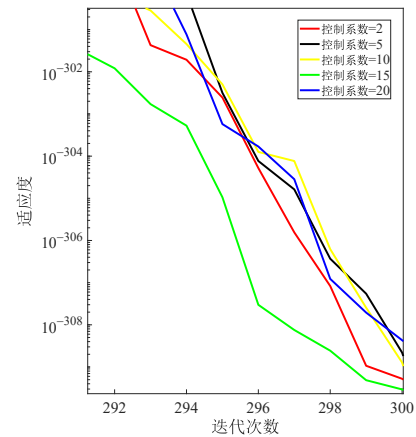
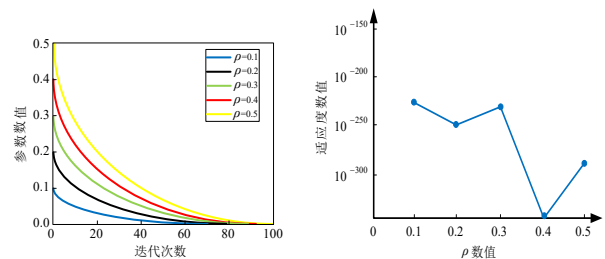


图18 资源系数参数实验

鲸落机制,增强局部搜索.隧穿概率与适应度差异和迭代进程相关,适应度差的个体隧穿概率更高.模型为

$$P_{\text{tunnel}}(t) = \frac{\rho \cdot (f(i) - f_{\text{best}})}{(f_{\text{max}} - f_{\text{best}} + \epsilon) \cdot \left(1 + \frac{t}{T_{\text{max}}}\right)} \quad (38)$$

其中, ρ 为量子隧穿概率的系数; ϵ 为一个非常小的数,避免出现错误; t 为当前迭代次数; f_{best} 为最优解适应度,根据图19的参数敏感性测试选出系数为0.4; $f(i)$ 为个体 i 的适应度值; f_{best} 为当前最优解的适应度; f_{max} 为当前



(a) 参数迭代图像 (b) 参数函数测试

图19 ρ 参数敏感测试

种群中所有个体的最大适应度; t 为当前的迭代次数; T_{\max} 为当前最大迭代次数,是一个非常小的数,防止零错误.

当隧穿概率 P_{tunnel} 小于随机数时,执行常规鲸落更新,否则进行量子隧穿.为增强探索性并避免局部最优,本文设计了自适应尺度因子的隧穿机制

$$\mathbf{X}_{\text{new}} = \mathbf{X}_{\text{best}} + \sigma_t \cdot \mathbf{N}(0, 1) \cdot (\mathbf{u}_b - \mathbf{l}_b) \quad (39)$$

其中, \mathbf{X}_{new} 为个体新位置; \mathbf{X}_{best} 为当前最优解的位置; σ_t 为自适应尺度因子; $\mathbf{N}(0, 1)$ 为标准正态分布随机数.

其自适应尺度因子 σ_t 数学模型为

$$\sigma_t = \omega \cdot \left(1 - \frac{t}{T_{\max}}\right) \quad (40)$$

其中, ω 为一个控制隧穿幅度的常数,根据图20的参数敏感性测试选出常数为0.3,随着迭代次数的增加而逐渐减小.

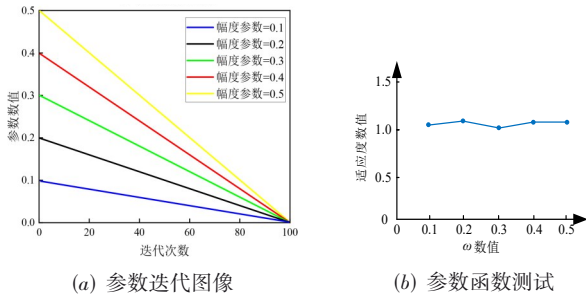


图20 ω 参数敏感测试

其最终的鲸落阶段的总位置更新数学模型为

$$\mathbf{X}_{\text{new}} = \begin{cases} (\mathbf{u}_b - \mathbf{l}_b) \exp\left(-\frac{C_2 \cdot T}{T_{\max}}\right), & \text{if } P_{\text{tunnel}} < \text{rand} \\ \mathbf{X}_{\text{best}} + \sigma_t \cdot \mathbf{N}(0, 1) \cdot (\mathbf{u}_b - \mathbf{l}_b), & \text{if } P_{\text{tunnel}} > \text{rand} \end{cases} \quad (41)$$

通过式(41)局部调整后的白鲸位置如图21所示,相比原始鲸落更新式(31),该方法使适应度较差的个体跃迁至最优解邻域,从而提高算法精度.

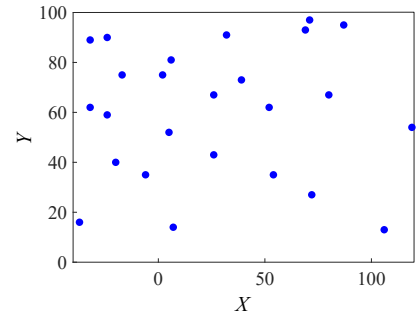
其量子隧穿鲸落位置更新示意图如图22所示.

3.4 算法流程及伪代码

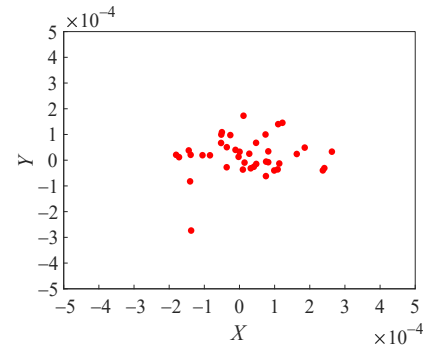
增强型白鲸蝠鲞优化算法流程图如图23和算法1所示.

3.5 时间复杂度分析

设白鲸种群数量为 N ,搜索空间复杂度为 d ,最大迭代次数为 T_{\max} ,传统BWO算法时间复杂度为 $O(N_d T_{\max})$.EBWO算法时间复杂度由三部分组成:Halton混沌序列和多方向正余弦更新 $O_4=O(N_d T_{\max})$ 、改进旋风链式觅食机制 $O_5=O(N_d T_{\max})$ 、非线性耦合鲸落量子隧穿机制 $O_6=O(0.1N_d T_{\max})$.因此,EBWO总时间复杂度为 $O=O_4+O_5+O_6=O(2.1N_d T_{\max})$ 近似等于 $O(N_d T_{\max})$,改进机制未增加复杂度,符合要求.



(a) 原始鲸落位置更新



(b) 改进后的鲸落位置更新

图21 原始鲸落位置更新与改进鲸落位置更新种群分布图

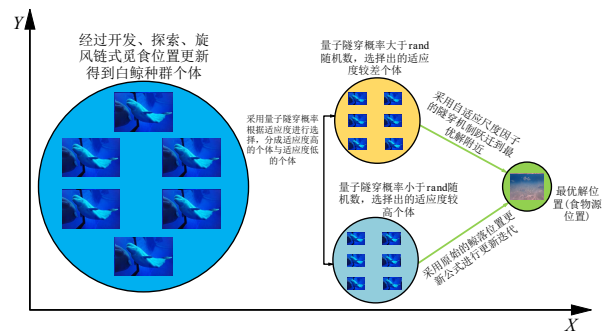


图22 非线性概率鲸落量子隧穿示意图

4 算法仿真

仿真实验在Windows11 64 bit操作系统上进行,配备128 GB内存,13th Gen Intel i9-13900HX CPU (2.20 GHz),使用Matlab R2023b进行仿真.

根据以上改进算法的参数敏感性以及参数性能对比等实验得出最终的参数数据如表5所示.

4.1 改进策略有效性分析

为验证三项改进策略对BWO性能的影响,本文将三种改进算法与BWO进行比较:基于Halton序列的白鲸优化算法(Halton sequence-based Beluga Whale Optimization algorithm, HBWO)(采用Halton混沌序列和多方向正余弦更新)、混沌白鲸优化算法(Chaotic Beluga

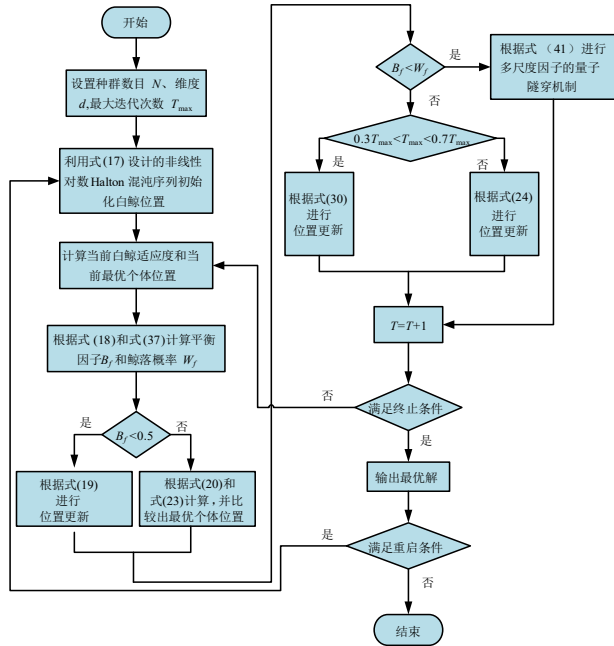


图 23 增强型白鲸蝠鲸优化算法流程图

Whale Optimization, CBWO)(采用改进旋风链式觅食机制)、自适应双行为白鲸优化算法(Adaptive Dual-behavior Beluga Whale Optimization, ADBWO)(采用非线性耦合鲸落量子隧穿机制)。

I_{MAE} 反映算法最优值与理论最优值的差距,计算式为

$$I_{MAE} = \frac{1}{n} \sum_{i=1}^n |m_i - o_i| \quad (42)$$

其中, n 为基准测试函数数量; m_i 为算法求解的平均最优值; o_i 为对应函数的理论最优值。

本文的 EBWO 算法与其他三种改进策略单独运行 30 次取平均值的 I_{MAE} , 具体如表 6 所示。

去除改进策略后, HBWO、CBWO、ADBWO 的平均绝对误差 (Mean Absolute Error, MAE) 较 BWO 增加, 表明改进策略有效。改进型旋风链式觅食位置更新对算法影响最大, 其次为 Halton 混沌序列和多方向正余弦更新、非线性耦合鲸落量子隧穿。

4.2 主流算法测试对比

为验证算法的可行性与优越性, 选取 CEC2005 基准函数进行测试, 包括 $F_1 \sim F_5$ 、 F_7 、 F_8 、 F_{14} 、 F_{15} 、 F_{17} 、 F_{20} 、 F_{23} 等 12 个函数, 并与哈里斯鹰 (Harris hawks Optimization Algorithm, HOA)、蝴蝶 (Butterfly Optimization Algorithm, BOA)、改进白鲸 (Improved Beluga Whale Optimization algorithm, IBWO)、改进灰狼 (Perturbed Grey Wolf Optimizer, PGWO)、白鲸 (BWO)、常春藤 (IVY optimization algorithm, IVY) 对比。所有算法最大迭代次数为 500, 种

算法 1 增强型白鲸蝠鲸优化算法伪代码

输入: 种群规模 N 、维度 d 、总迭代次数 T_{max}

输出: 最优位置 X_{best} 、最优适应度 F_{best}

1. BEGIN
2. 采用非线性对数 Halton 混沌序列初始化种群[式(17)], 计算白鲸适应度值并记录最优个体
3. While ($T < T_{max}$) do
4. 计算平衡因子 B_j [式(18)]和非线性耦合鲸落概率 W_j [式(38)]
5. for $i = 1$ to N do
6. if $B_j < 0.5$ then
7. 根据式(19)更新白鲸个体的探索位置
8. else
9. 根据式(20)和式(23)计算白鲸和多方向正余弦抢食白鲸的位置, 并进行比较选出最佳白鲸
10. end if
11. if $B_j < W_j$ then
12. 根据式(41)进行量子隧穿的鲸落阶段
13. end if
14. end for
15. for $i = 1$ to N do
16. if $0.3T_{max} < T < 0.7T_{max}$ then
17. 根据式(30)进行改进型旋风链式觅食更新位置
18. else
19. 根据式(24)传统旋风觅食机制进行更新位置
20. end for
21. 更新当前最优解 X_{best} 及适应度 F_{best}
22. $T = T + 1$
23. if $T > T_{max}$ 则迭代结束
24. end while
25. 输出最优解 X_{best} 和问题最优适应度 F_{best}

表 5 EBWO 算法参数设置

参数	数值
base1	2
base2	3
S	3
λ	1.5
ω	0.3
ρ	0.4
k	15
η	2
R_0	0.4

表 6 I_{MEA} 单一策略实验数据

算法	I_{MAE}	相比较 EBWO 的变化
EBWO	2.74×10^{-39}	0
HBWO	6.77×10^{-20}	6.76×10^{-20}
CBWO	1.23×10^{-29}	1.22×10^{-29}
ADBWO	1.33×10^{-17}	1.32×10^{-17}

群数为 100,测试维度为 30,函数信息见表 6. 算法寻优 迭代图如图 24 所示. 数据如表 7 所示.

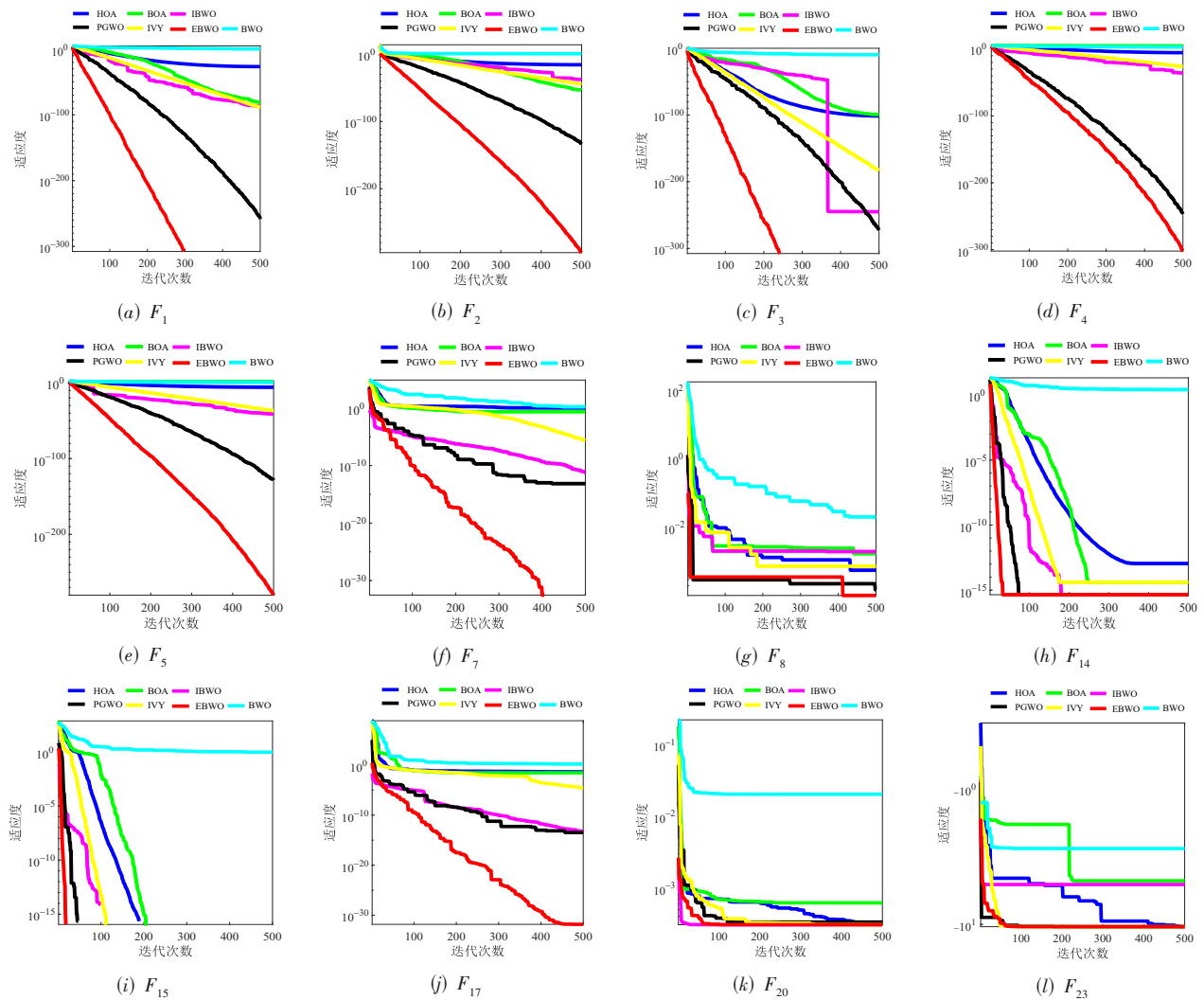


图 24 基准测试函数迭代收敛曲线

表 7 基准测试函数迭代数据

函数	算法	最优解	最差解	平均值	运行时间/s
F_1	HOA	2.65×10^{-78}	4.51×10^{-23}	8.75×10^{-37}	1.954
	BOA	3.6×10^{-156}	3.02×10^{-98}	6.5×10^{-107}	1.923
	IBWO	6.5×10^{-173}	2.45×10^{-95}	4.3×10^{-121}	1.877
	PGWO	0	3.3×10^{-145}	3.3×10^{-270}	1.902
	IVY	3.4×10^{-132}	9.74×10^{-87}	2.55×10^{-95}	1.895
	EBWO	0	0	0	1.806
	BWO	3.32×10^{-67}	3.85×10^{-23}	1.78×10^{-46}	1.873
F_2	HOA	8.65×10^{-60}	3.75×10^{-33}	4.77×10^{-45}	1.955
	BOA	7.85×10^{-77}	2.65×10^{-65}	3.75×10^{-64}	1.935
	IBWO	7.1×10^{-236}	3.3×10^{-43}	2.9×10^{-166}	1.892
	PGWO	0	3.6×10^{-270}	1.7×10^{-286}	1.903
	IVY	3.8×10^{-140}	7.71×10^{-85}	2.2×10^{-104}	1.922
	EBWO	0	0	0	1.890

续表

函数	算法	最优解	最差解	平均值	运行时间/s
F_3	BWO	7.87×10^{-70}	3.75×10^{-25}	1.18×10^{-44}	1.906
	HOA	4.4×10^{-104}	3.57×10^{-94}	3.5×10^{-103}	1.780
	BOA	3.8×10^{-113}	2.75×10^{-93}	7.86×10^{-84}	1.703
	IBWO	0	3.3×10^{-276}	4.8×10^{-288}	1.664
	PGWO	0	3.4×10^{-301}	2.9×10^{-331}	1.707
	IVY	3.8×10^{-179}	2.6×10^{-156}	3.8×10^{-166}	1.774
	EBWO	0	0	0	1.656
	BWO	3.41×10^{-94}	2.64×10^{-23}	4.55×10^{-65}	1.729
F_4	HOA	6.6×10^{-104}	4.8×10^{-34}	2.4×10^{-103}	1.224
	BOA	7.7×10^{-144}	2.70×10^{-47}	3.66×10^{-95}	1.653
	IBWO	0	5.6×10^{-94}	3.5×10^{-167}	1.156
	PGWO	0	0	0	1.199
	IVY	2.66×10^{-99}	3.74×10^{-29}	5.89×10^{-64}	1.512
	EBWO	0	0	0	1.048
	BWO	1.75×10^{-93}	4.77×10^{-11}	5.02×10^{-68}	1.243
F_5	HOA	5.5×10^{-122}	4.7×10^{-67}	4.8×10^{-103}	1.954
	BOA	9.7×10^{-140}	4.7×10^{-60}	6.8×10^{-124}	2.048
	IBWO	0	5.64×10^{-95}	3.75×10^{-277}	1.904
	PGWO	0	0	0	1.922
	IVY	3.3×10^{-105}	4.86×10^{-66}	1.8×10^{-107}	1.947
	EBWO	0	0	0	1.857
	BWO	5.55×10^{-97}	2.23×10^{-44}	3.94×10^{-95}	1.936
F_7	HOA	5.55×10^{-16}	3.44×10^{-7}	8.64×10^{-12}	1.945
	BOA	6.75×10^{-15}	4.85×10^{-7}	3.22×10^{-13}	1.864
	IBWO	3.35×10^{-25}	2.23×10^{-16}	8.87×10^{-23}	1.864
	PGWO	7.85×10^{-29}	1.66×10^{-15}	4.73×10^{-27}	1.904
	IVY	8.75×10^{-16}	3.65×10^{-9}	2.33×10^{-14}	1.843
	EBWO	2.23×10^{-32}	3.34×10^{-30}	1.88×10^{-30}	1.847
	BWO	6.87×10^{-15}	3.55×10^{-10}	2.65×10^{-12}	1.848
F_8	HOA	2.33×10^{-3}	1.66×10^{-1}	3.65×10^{-2}	1.847
	BOA	3.54×10^{-3}	4.51×10^{-2}	7.78×10^{-3}	1.759
	IBWO	1.55×10^{-5}	1.66×10^{-3}	2.22×10^{-4}	1.875
	PGWO	1.44×10^{-5}	2.34×10^{-2}	7.64×10^{-4}	1.643
	IVY	3.32×10^{-3}	2.56×10^{-1}	2.36×10^{-2}	1.859
	EBWO	1.13×10^{-6}	2.74×10^{-4}	3.32×10^{-0}	1.746
	BWO	4.45×10^{-2}	2.77	4.77×10^{-2}	1.748
F_{14}	HOA	8.77×10^{-14}	3.23×10^{-8}	4.55×10^{-13}	1.077
	BOA	2.87×10^{-13}	7.66×10^{-9}	3.86×10^{-10}	1.064
	IBWO	1.18×10^{-15}	3.32×10^{-12}	5.44×10^{-14}	1.068
	PGWO	5.44×10^{-15}	5.99×10^{-13}	2.74×10^{-15}	1.094
	IVY	4.66×10^{-13}	2.77×10^{-10}	8.87×10^{-12}	1.131
	EBWO	4.45×10^{-16}	2.33×10^{-15}	4.02×10^{-16}	1.093
	BWO	4.45×10^{-13}	7.65×10^{-11}	2.23×10^{-11}	1.088
F_{15}	HOA	4.43×10^{-11}	3.45×10^{-9}	1.13×10^{-10}	0.745
	BOA	5.43×10^{-12}	7.65×10^{-8}	6.65×10^{-11}	0.794
	IBWO	1.88×10^{-13}	2.54×10^{-12}	1.89×10^{-13}	0.648

续表

函数	算法	最优解	最差解	平均值	运行时间/s
	PGWO	3.38×10^{-15}	1.33×10^{-16}	3.34×10^{-13}	0.684
	IVY	2.44×10^{-10}	7.76×10^{-9}	7.76×10^{-10}	0.993
	EBWO	0	2.33×10^{-15}	3.33×10^{-14}	0.646
	BWO	9.98×10^{-5}	2.44×10^{-3}	1.18×10^{-4}	0.756
F_{17}	HOA	4.34×10^{-16}	1.45×10^{-4}	8.77×10^{-13}	0.884
	BOA	1.24×10^{-11}	2.22×10^{-5}	2.23×10^{-10}	0.657
	IBWO	5.44×10^{-24}	5.88×10^{-13}	4.32×10^{-20}	0.757
	PGWO	5.77×10^{-28}	4.43×10^{-13}	8.75×10^{-25}	0.975
	IVY	8.65×10^{-14}	1.87×10^{-3}	6.54×10^{-12}	0.945
	EBWO	1.57×10^{-32}	2.34×10^{-28}	1.55×10^{-30}	0.776
	BWO	2.33×10^{-15}	7.76×10^{-4}	3.44×10^{-9}	0.756
F_{20}	HOA	1.23×10^{-3}	1.54×10^{-2}	3.54×10^{-2}	0.772
	BOA	8.87×10^{-2}	8.77×10^{-1}	6.55×10^{-2}	0.947
	IBWO	7.55×10^{-4}	5.43×10^{-3}	2.88×10^{-3}	0.652
	PGWO	8.77×10^{-4}	2.23×10^{-2}	1.54×10^{-3}	0.765
	IVY	6.55×10^{-2}	4.45×10^{-1}	7.65×10^{-1}	0.964
	EBWO	3.08×10^{-4}	1.74×10^{-3}	3.02×10^{-4}	0.645
	BWO	5.44×10^{-2}	3.64×10^{-1}	7.23×10^{-2}	0.761
F_{23}	HOA	-9.741	-8.751	-9.451	1.087
	BOA	-9.365	-9.075	-9.225	1.091
	IBWO	10.055	-9.643	-9.884	0.987
	PGWO	-10.224	-9.764	-10.053	1.321
	IVY	-9.651	-9.351	-9.773	0.974
	EBWO	-10.403	-10.039	-10.255	0.938
	BWO	-9.652	-9.286	-9.531	1.431

实验结果表明:EBWO在多个标准测试函数上表现优越,特别是在精度和收敛速度上. EBWO在大多数函数中实现了最优解或接近最优解,且明显优于HOA、BOA、IBWO等算法. 与BWO相比,EBWO具有更低的误差和更快的收敛速度,展示了其高效的全局搜索和局部开发能力. EBWO在 F_1 、 F_2 、 F_3 、 F_4 、 F_5 等函数中达到了理论最优解,并在复杂问题中显示出良好的稳定性和鲁棒性.

4.3 机械臂轨迹优化

4.3.1 机械臂时间最优轨迹优化

本文以实验室开源机械臂为仿真对象,运行环境与算法测试一致,通过Matlab分析正逆运动学,D-H参数、关约束、路径点坐标及关节角度见表1~表4.

为验证算法在机械臂时间最优轨迹方面的优越性,将本文算法与传统BWO算法及文献[15~17]的改进算法进行对比. 文献[15]采用动态变化因子、非线性递减惯性权重进行改进粒子群算法,文献[16]采用自适应高斯策略和凸透镜成像学习来改进白鲸算法,文献[17]采用惯性权重和强化学习进行改进鲸鱼算法. 所有算法种群数设为100,最大迭代次数为100,变量维度为3,最大速度为2 rad/s.

为验证3-5-3多项式,设定各区间关节运行时间 t_1 、 t_2 、 t_3 为4 s,总时间为12 s. 角位移、角速度、角加速度如图25所示.

4.3.2 时间最优轨迹单一策略消融实验

本文对改进算法进行单一策略消融实验,验证每个策略在机械臂轨迹中的可行性. 实验设置一致,种群数为100,变量维度为3,最大速度为2 rad/s,最大迭代次数为100,独立运行6次. 收敛迭代见图26,时间最优数据见表8.

由图26和表8可知,CBWO策略的最优时间接近EBWO,HBWO与EBWO相差较大. 各策略能有效跳出局部最优,相较传统BWO算法,性能有所提升,验证了改进策略在机械臂应用中的有效性.

4.3.3 时间最优轨迹主流算法对比

将本文的EBWO算法与主流文献算法应用于机械臂六个关节,运行30次,得到各段关节的最优适应度收敛迭代对比图,如图27所示.

EBWO算法在6个关节上表现最佳,时间最小,避免了BWO陷入局部最优. BWO迭代过程为横线,表明早期陷入局部最优,而EBWO多次跳跃转折,提升收敛

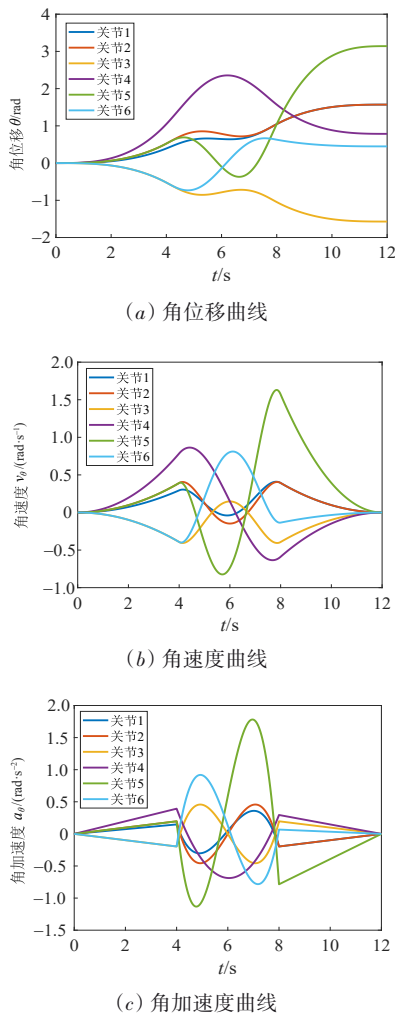


图 25 3-5-3 传统插值运动曲线

精度与速度,总误差最小,验证了其优越性.

为确保机械臂稳定运行,各关节轨迹时间一致,取最大时间作为总运行时间,总时间对比见表 9.

表 8 单一策略运行时间数据

时间	t_1/s	t_2/s	t_3/s	总时间/s	最优时间/s	平均时间/s	标准差	运行成功率/%
HBWO	1.897 2	2.478 4	3.206 8	7.582 4	7.443 2	7.598 8	1.847×10^{-2}	90.2
EBWO	1.841 1	1.992 8	3.144 7	6.978 6	6.947 6	7.002 1	2.984×10^{-3}	96.1
CBWO	1.845 3	1.992 9	3.151 7	6.989 9	6.972 2	7.045 5	1.895×10^{-2}	93.3
ADBWO	1.841 7	1.994 4	3.146 7	6.982 8	6.974 6	7.023 1	3.331×10^{-2}	93.2

表 9 EBWO 与文献算法最优时间数据

时间	t_1/s	t_2/s	t_3/s	总时间/s	最优时间/s	平均时间/s	标准差	运行成功率/%
BWO	1.962 3	1.677 0	4.000 0	7.639 3	7.605 5	7.734 4	2.33×10^{-1}	83.5
EBWO	1.812 6	1.991 0	3.141 7	6.945 3	6.945 0	6.998 7	2.99×10^{-3}	97.6
文献[15]	1.864 6	1.979 8	3.523 1	7.367 5	7.332 1	7.388 7	2.43×10^{-2}	92.0
文献[16]	1.883 4	2.115 0	3.148 5	7.146 9	7.107 8	7.234 4	2.65×10^{-2}	94.3
文献[17]	1.874 1	1.916 3	3.271 2	7.061 6	7.047 8	7.071 6	2.77×10^{-3}	95.5

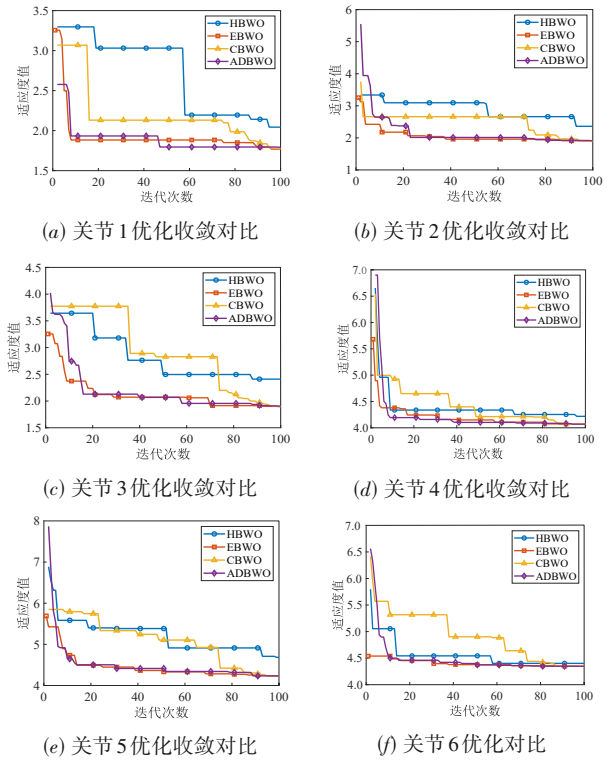


图 26 单一策略各关节最优收敛曲线

EBWO 算法最优时间为 6.945 0 s,比文献[15~17]分别缩短了 5.28%、2.29%、1.46%,比传统 BWO 缩短 8.69%.在 3-5-3 多项式插值轨迹中,EBWO 将总时间(12 s)缩短了 42.13%,证明了其在精度和稳定性上的提升.

由表格数据可知,文献[17]算法与本文 EBWO 算法差值最小.因此,对角位移、角速度、角加速度三种参数变量进行对比,如图 28 所示.

由图 28 可知,在关节 5 和 6 的角速度及角加速度上,EBWO 算法优于文献[17],且满足物理约束,确保系统稳定.同时,EBWO 实现了更短的轨迹执行时间,

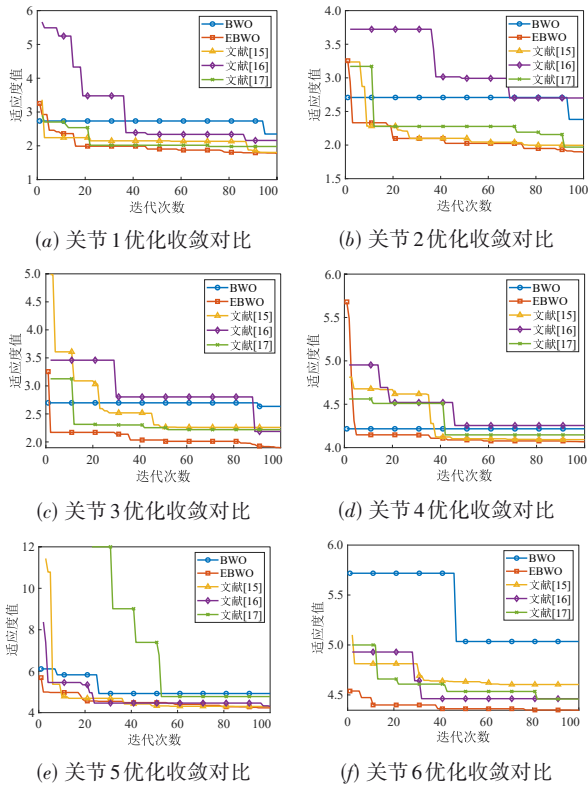


图27 主流算法关节运行迭代收敛曲线对比

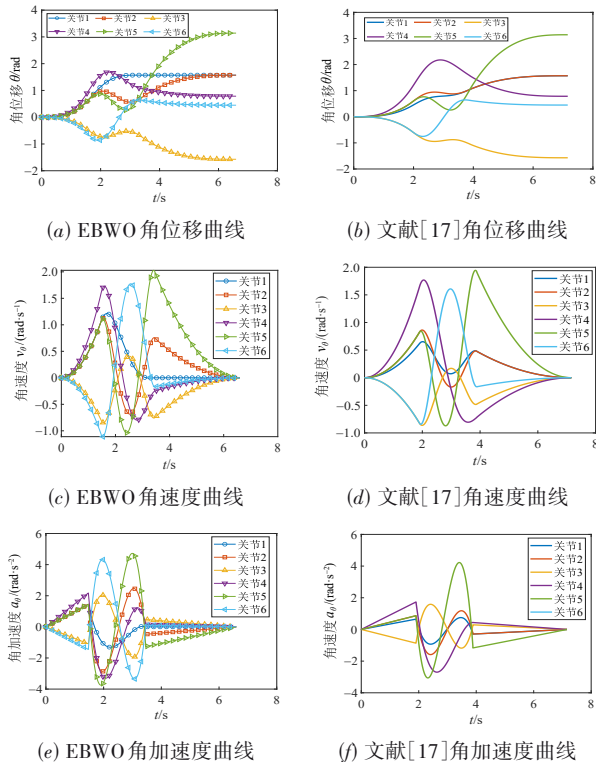


图28 EBWO算法与文献[17]关节曲线对比

验证了其在时间最优轨迹规划中的有效性.

为直观观察机械臂末端轨迹,将EBWO算法、文献[17]

算法和3-5-3传统插值方法进行比较,输出模型轨迹.二维轨迹见图29,三维机械臂模型轨迹见图30.

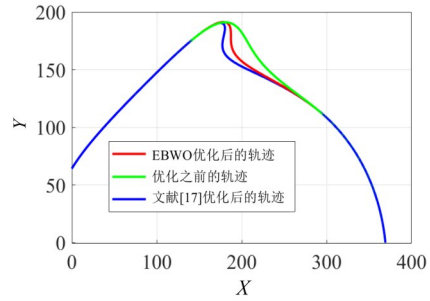


图29 末端轨迹位置二维曲线

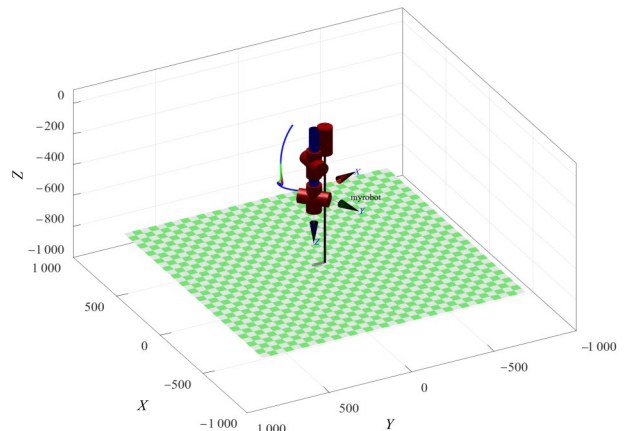


图30 末端轨迹空间曲线

为验证本文优化算法在不同速度约束下的表现,将速度约束从2 rad/s 改为3 rad/s 和5 rad/s,得到不同约束下的关节三段运动时间(见表10),角位移、角速度、角加速度曲线如图31所示.

表10 不同约束下运行最优时间数据

速度约束/ (rad · s ⁻¹)	t ₁ /s	t ₂ /s	T/s	总时间 t/s	平均时间 t/s
2	1.812 6	1.991 0	3.141 7	6.945 3	6.998 1
3	1.224 8	1.324 4	2.099 7	4.648 1	4.775 6
5	0.738 1	0.789 1	1.257 4	2.784 6	2.997 6

随着最大速度约束的放宽,机械臂轨迹执行时间缩短.在2 rad/s 条件下,最优时间为6.998 1 s,3 rad/s 和5 rad/s 条件下分别缩短2.225 s 和4.000 5 s,验证了本文算法在不同速度限制下的适应性和优化性能.

4.4 长轨迹验证对比

为验证算法在不同轨迹下的实用性,本文选择另一组关节角度作为初始配置,通过模拟不同关节角度组合验证其可行性与实用性,具体关节角位置见表11.

为验证算法优越性,本文将EBWO 与其他算法在轨迹2 中进行对比,最优粒子迭代收敛如图32 所示,数

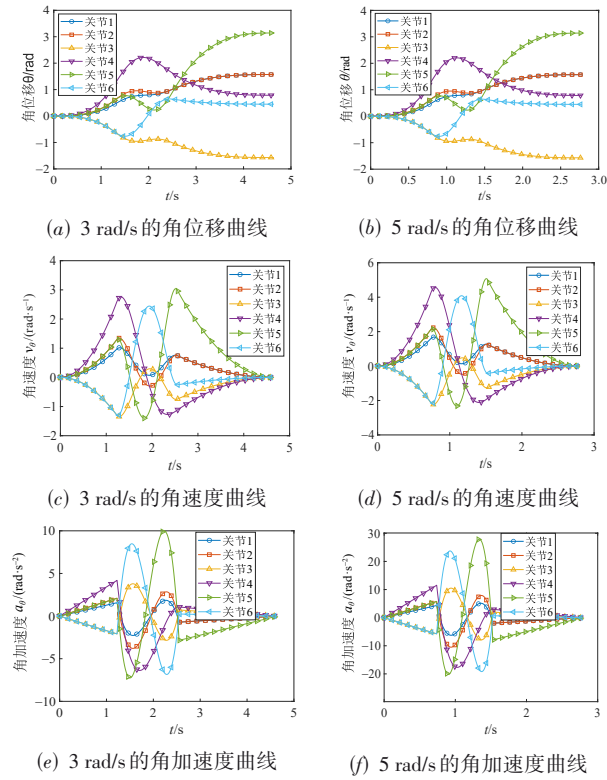


图 31 3 rad/s 和 5 rad/s 约束下的关节曲线

表 11 轨迹 2 关节角位置 单位:rad

关节序号	起始点	路径点 1	路径点 2	终止点
1	0	0.523 6	1.570 8	1.570 8
2	0	0.523 6	0.785 4	1.570 8
3	0	-0.392 7	-0.785 4	-1.570 8
4	0	0.785 4	1.047 2	0.785 4
5	0	0.523 6	1.047 2	1.047 2
6	0	-0.523 6	0.628 3	0.448 8

据见表 12.

由图 32 和表 12 数据可知,在复杂轨迹 2 中,EBWO 比文献 [15~17] 最优时间分别缩短了 3.42%、5.87%、0.58%,比 BWO 缩短了 19.41%,验证了其在复杂轨迹规划中的优越收敛性能和跳出局部最优能力.

图 33 显示了机械臂在轨迹 2 中的空间移动示意图,绿色为优化前的 3-5-3 插值轨迹,红色为 EBWO 优化后的轨迹,验证了算法在不同应用场景下的有效性.

5 实物实验验证

5.1 实务平台搭建

为验证时间最优轨迹规划,本文将实验数据应用于实物平台.具体如图 34 所示.平台由主控制器、示教器、伺服系统、编码器、伺服电机、上位机、机械臂和软

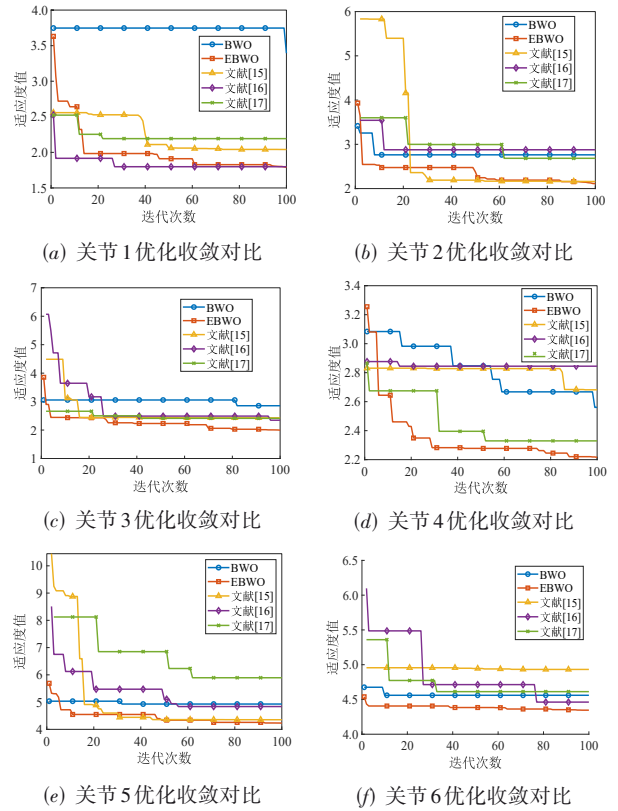


图 32 轨迹 2 不同算法关节收敛曲线

表 12 不同算法最优时间数据 单位:s

算法	t_1	t_2	t_3	总时间	平均时间
BWO	2.371 5	2.512 9	3.388 1	8.272 5	8.292 0
EBWO	1.461 1	1.977 8	3.141 8	6.580 7	6.667 1
文献[15]	2.161 9	1.527 2	3.141 6	6.830 7	6.903 3
文献[16]	2.083 2	1.712 3	3.205 9	7.001 4	7.083 2
文献[17]	1.565 1	1.809 5	3.207 6	6.582 2	6.705 7

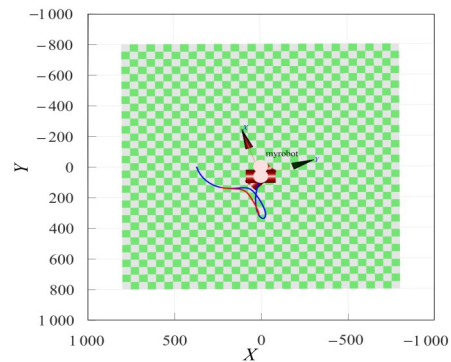


图 33 轨迹 2 末端空间曲线

件组成,示教器输入参数,主控制器计算并发送指令,通过 EtherCAT 传输至伺服控制柜.编码器解析报文驱动电机,Arduino 实时优化调整指令,确保机械臂精确运动.

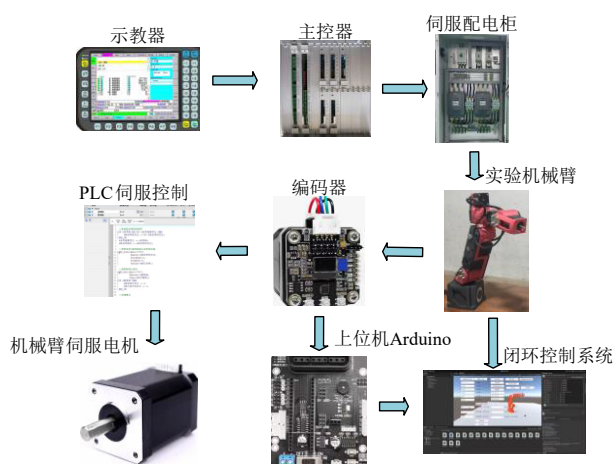


图34 机械臂实验平台

5.2 实物验证对比

路径点和最优时间输入主控制器,主控制器生成运动指令并发送至伺服系统,实现最优时间控制. 机械臂经过中间路径点,路径如图35所示.



图35 机械臂轨迹1关节运行图

如图36所示,机械臂实际轨迹平滑,关节1、5、6的加速度虽增加,但仍在约束范围内. 与Matlab仿真数据对比,实际最优时间6.977 3 s,误差与6.945 0 s在可接受范围内,验证了算法的实用性与优越性.

本文采用轨迹2进行仿真验证,并在实物平台上进行验证. 起点、路径点1、路径点2、终点的机械臂运行图如图37所示.

机械臂轨迹2的实际角位移曲线、角速度曲线、角加速度曲线与仿真对比如图38所示.

由图38的角位移、角速度、角加速度曲线显示,实

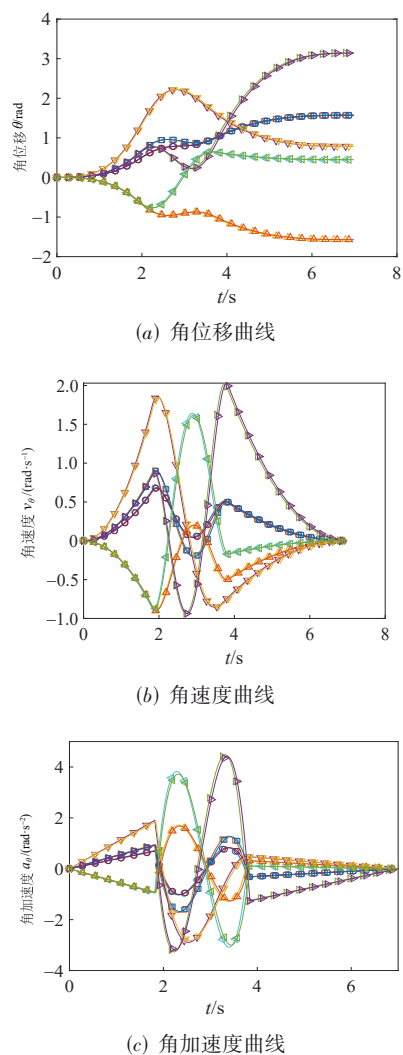


图36 轨迹1实际实验数据与仿真实验数据汇总

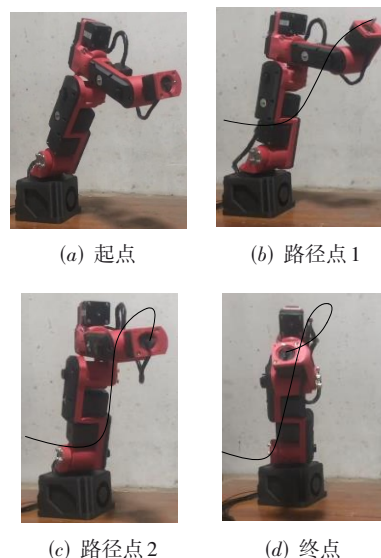


图37 机械臂轨迹2关节运行图

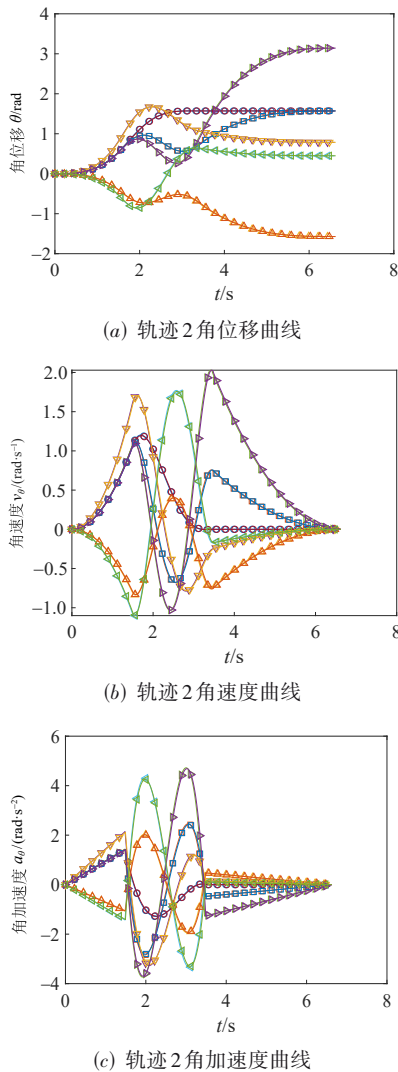


图 38 轨迹 2 实际实验数据与仿真实验数据汇总

际输出与仿真曲线基本重合,误差在允许范围内,验证了本文算法在不同轨迹中的实用性.

为避免实验的偶然性,同时验证了算法的稳定性,将实验运行 10 次,并绘制数据图像进行对比,分析其具体数据结果,如图 39 和表 13 所示.

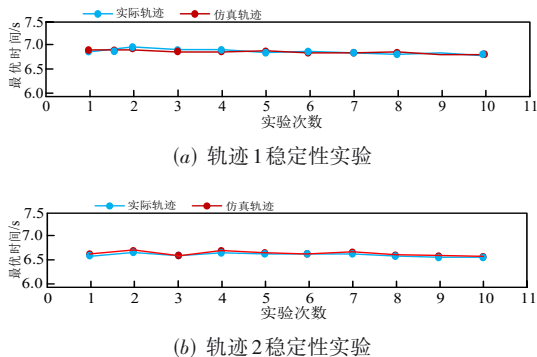


图 39 10 次实验的实际和仿真数据对比

表 13 10 次实验数据

单位:s

实验类别	轨迹 1 最优时间	轨迹 2 最优时间	轨迹 1 平均时间	轨迹 2 平均时间
实际实验	6.977 3	6.636 3	6.989 4	6.637 4
仿真实验	6.945 0	6.607 5	6.967 0	6.618 8

图 39 和表 13 数据显示,实验数据与仿真结果高度一致,验证了仿真模型的有效性和可靠性.实际与仿真实验在轨迹 1 和轨迹 2 的最优时间及平均时间差异均小于 0.5% (轨迹 1 误差 0.46%, 轨迹 2 误差 0.13%).

6 结论

本文提出的 EBWO 通过改进初始化、多阶段协同搜索和鲸落阶段,解决了传统 BWO 在机械臂轨迹优化中的路径冗长、收敛缓慢和易陷入局部最优的问题.通过引入对数非线性 Halton 混沌序列、设计多方向正余弦位置更新、融合改进型旋风链式觅食策略以及结合非线性正弦鲸落概率与量子隧穿机制,构建了多阶段动态协同优化框架.

实验结果表明:EBWO 在标准测试函数和轨迹任务中表现优异,能够有效提高机械臂轨迹的平滑度、动态可控性,并显著缩短运行时间.实物平台验证了算法的稳定性和有效性.

EBWO 算法兼顾全局搜索和局部精度,具备较强的适应性和工程应用价值,未来可广泛应用于多自由度机械臂控制、无人驾驶路径规划和智能制造等领域.

参考文献

[1] 李玲,侯玉龙,李瑶,等.基于改进麻雀算法的机械臂多目标轨迹优化方法[J/OL]. 工程设计学报, 1-9. [2025-05-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?filename=GCSJ20250417005&dbname=CJFD&dbcode=CJFQ>.
 LI L, HOU Y L, LI Y, et al. Multi-objective trajectory optimization method for manipulator based on the newly improved sparrow search algorithm[J/OL]. Chinese Journal of Engineering Design, 1-9. [2025-05-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?filename=GCSJ20250417005&dbname=CJFD&dbcode=CJFQ>. (in Chinese)
 [2] 刘小松,康磊,单泽彪,等.基于 RTSR-RRT^{*}算法的机械臂路径规划[J]. 仪器仪表学报, 2025, 46(3): 65-73.
 LIU X S, KANG L, SHAN Z B, et al. Path planning of robotic arm based on RTSR-RRT^{*} algorithm[J]. Chinese Journal of Scientific Instrument, 2025, 46(3): 65-73. (in Chinese)
 [3] 张盛,沈捷,曹恺,等.基于改进 DDPG 的机械臂 6D 抓取方法研究[J/OL]. 计算机工程与应用, 1-9. [2025-05-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?filename=>

- JSGG20250402002&dbname=CJFD&dbcode=CJFQ.
ZHANG S, SHEN J, CAO K, et al. Research on 6D robotic arm grasping method based on improved DDPG[J/OL]. Computer Engineering and Applications, 1-9. [2025-05-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?filename=JSGG20250402002&dbname=CJFD&dbcode=CJFQ>. (in Chinese)
- [4] 孟凡希, 张乃文, 李博元, 等. 改进狼群与蝠鲼混合算法的机械臂轨迹规划[J]. 机械设计与研究, 2025, 41(1): 337-344.
MENG F X, ZHANG N W, LI B Y, et al. Improving the hybrid algorithm of wolf pack and manta ray for robotic arm trajectory planning[J]. Machine Design & Research, 2025, 41(1): 337-344. (in Chinese)
- [5] LIU J Q, LIU S H, SONG M, et al. Time-optimal robotic arm trajectory planning for coating machinery based on a dynamic adaptive PSO algorithm[J]. Coatings, 2025, 15(1): 2.
- [6] MOUSA M A A, ELGOHR A T, KHATER H A. Trajectory optimization for a 6 DOF robotic arm based on reachability time[J]. Annals of Emerging Technologies in Computing, 2024, 8(1): 22-35.
- [7] LIU J J, WANG H, LI X, et al. Robotic arm trajectory optimization based on multiverse algorithm[J]. Mathematical Biosciences and Engineering, 2023, 20(2): 2776-2792.
- [8] 裴志恒, 孙军锋, 徐贞, 等. 多策略改进蜣螂优化算法的机械臂最优时间规划研究[J/OL]. 机械科学与技术, 1-12. [2025-05-03]. <https://link.cnki.net/doi/10.13433/j.cnki.1003-8728.20250040>.
PEI Z H, SUN J F, XU Z, et al. Research on optimal time planning of manipulator with multi-strategy improved dung beetle optimization algorithm[J/OL]. Mechanical Science and Technology for Aerospace Engineering, 1-12. [2025-05-03]. <https://link.cnki.net/doi/10.13433/j.cnki.1003-8728.20250040>. (in Chinese)
- [9] 吴继春, 张斋武, 杨永达, 等. 基于改进金枪鱼群算法的机械臂时间最优轨迹规划[J]. 计算机集成制造系统, 2024, 30(12): 4292-4301.
WU J C, ZHANG Z W, YANG Y D, et al. Time optimal trajectory planning of manipulator based on improved tuna swarm algorithm[J]. Computer Integrated Manufacturing Systems, 2024, 30(12): 4292-4301. (in Chinese)
- [10] 黄成, 王涛, 许家忠. 基于混合蜜獾算法的机械臂最优运动规划方法[J]. 仪器仪表学报, 2024, 45(4): 234-247.
HUANG C, WANG T, XU J Z. Optimal motion planning method of manipulator based on hybrid honey badger algorithm[J]. Chinese Journal of Scientific Instrument, 2024, 45(4): 234-247. (in Chinese)
- [11] 赵亮, 刘瑞雪, 张玮奇, 等. 基于自适应混沌麻雀搜索算法的机械臂最优时间轨迹规划[J]. 信息与控制, 2024, 53(6): 739-749.
ZHAO L, LIU R X, ZHANG W Q, et al. Optimal time trajectory planning of robotic arm based on adaptive chaotic sparrow search algorithm[J]. Information and Control, 2024, 53(6): 739-749. (in Chinese)
- [12] 许家伟, 李磊, 汪建华, 等. 基于 TCSPSO 算法的机械臂运动时间最优轨迹规划[J]. 现代制造工程, 2025(3): 69-76, 83.
XU J W, LI L, WANG J H, et al. Time-optimal trajectory planning of manipulator based on TCSPSO algorithm[J]. Modern Manufacturing Engineering, 2025(3): 69-76, 83. (in Chinese)
- [13] HU X, WU H, SUN Q L, et al. Robot time optimal trajectory planning based on improved simplified particle swarm optimization algorithm[J]. IEEE Access, 2023, 11: 44496-44508.
- [14] LU Z K, YOU Z C, XIA B H. Time optimal trajectory planning of robotic arm based on improved sand cat swarm optimization algorithm[J]. Applied Intelligence, 2025, 55(5): 323.
- [15] 王桂荣, 倪志强, 周坤, 等. 多策略改进粒子群算法的机械臂时间最优轨迹规划[J]. 中国机械工程, 2025, 36(5): 1044-1053.
WANG G R, NI Z Q, ZHOU K, et al. Time-optimal trajectory planning of robotic arms based on MIPS algorithm[J]. China Mechanical Engineering, 2025, 36(5): 1044-1053. (in Chinese)
- [16] 文裕杰, 张达敏. 增强型白鲸优化算法及其应用[J]. 山东大学学报(工学版), 2025, 55(3): 88-99.
WEN Y J, ZHANG D M. Enhanced beluga whale optimization algorithm and its application[J]. Journal of Shandong University (Engineering Science), 2025, 55(3): 88-99. (in Chinese)
- [17] DU J, HOU J, WANG H Y, et al. Application of an improved whale optimization algorithm in time-optimal trajectory planning for manipulators[J]. Mathematical Biosciences and Engineering, 2023, 20(9): 16304-16329.
- [18] ZHONG C T, LI G, MENG Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm[J]. Knowledge-Based Systems, 2022, 251: 109215.
- [19] 柳博, 吴松荣, 付聪, 等. 基于改进白鲸算法优化 BiTCN-BiGRU 的锂电池 SOC 估计[J]. 电子测量技术, 2025, 48(9): 75-83.

- LIU B, WU S R, FU C, et al. SOC estimation of lithium batteries based on BiTCN-BiGRU optimized by improved beluga whale algorithm[J]. *Electronic Measurement Technology*, 2025, 48(9): 75-83. (in Chinese)
- [20] 梁凯, 韩西. 基于IBWO-TEB算法的机器人自主路径规划[J]. *制造技术与机床*, 2025(3): 49-55.
- LIANG K, HAN X. Mobile robot autonomous path planning based on IBWO-TEB algorithm[J]. *Manufacturing Technology & Machine Tool*, 2025(3): 49-55. (in Chinese)
- [21] LI L L, FAN X D, WU K J, et al. Multi-objective distributed generation hierarchical optimal planning in distribution network: Improved beluga whale optimization algorithm[J]. *Expert Systems with Applications*, 2024, 237: 121406.
- [22] 陈心怡, 张孟健, 王德光. 基于Fuch映射的改进白鲸优化算法及应用[J]. *计算机工程与科学*, 2024, 46(8): 1482-1492.
- CHEN X Y, ZHANG M J, WANG D G. Improved beluga whale optimization algorithms based on Fuch mapping and applications[J]. *Computer Engineering & Science*, 2024, 46(8): 1482-1492. (in Chinese)
- [23] 王亚辉, 张虎晨, 王学兵, 等. 基于混沌反向学习和水波算法改进的白鲸优化算法[J]. *计算机应用研究*, 2024, 41(3): 729-735.
- WANG Y H, ZHANG H C, WANG X B, et al. Improved beluga whale optimization algorithm based on chaotic inverse learning and water wave algorithm[J]. *Application Research of Computers*, 2024, 41(3): 729-735. (in Chinese)
- [24] OLADEJO S O, EKWE S O, MIRJALILI S. The hiking optimization algorithm: A novel human-based metaheuristic approach[J]. *Knowledge-Based Systems*, 2024, 296: 111880.
- [25] ARORA S, SINGH S. Butterfly optimization algorithm: A novel approach for global optimization[J]. *Soft Computing*, 2019, 23(3): 715-734.
- [26] FENG Z X, AN J H, HAN M Y, et al. Office building energy consumption forecast: Adaptive long short term memory networks driven by improved beluga whale optimization algorithm[J]. *Journal of Building Engineering*, 2024, 91: 109612.
- [27] 薛阳, 倪大斌, 卢秋红, 等. 基于PGWO算法的机器人路径规划[J]. *控制与决策*, 2025, 40(4): 1395-1401.
- XUE Y, NI D B, LU Q H, et al. Path planning of mobile robot based on PGWO algorithm[J]. *Control and Decision*, 2025, 40(4): 1395-1401. (in Chinese)
- [28] GHASEMI M, ZARE M, TROJOVSKÝ P, et al. Optimization based on the smart behavior of plants with its engineering applications: Ivy algorithm[J]. *Knowledge-Based Systems*, 2024, 295: 111850.

作者简介



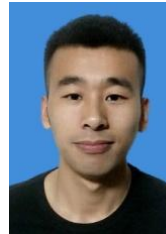
高永新 男, 1973年11月出生于辽宁省阜新市. 现为辽宁工程技术大学机械工程学院副教授. 主要研究方向为机电一体化.
E-mail: gaoyx1125@126.com



刘俊毅 男, 2001年6月出生于山东省潍坊市. 现为辽宁工程技术大学机械工程学院硕士研究生. 主要研究方向为机器人设计与轨迹优化.
E-mail: 1312529551@qq.com



汪洋 男, 2000年11月出生于江苏省无锡市. 现为辽宁工程技术大学机械工程学院硕士研究生. 主要研究方向为新能源汽车驱动系统.
Email: 2356384131@qq.com



贾东 男, 1995年5月出生于黑龙江省齐齐哈尔市. 现为辽宁工程技术大学机械工程学院硕士研究生. 主要研究方向为故障诊断.
Email: 1332755792@qq.com