

基于非线性混合天鹰 - 哈里斯优化算法的 仓储机器人路径规划研究

武紫豪¹ 杨紫伊¹
WU Zihao YANG Ziyi

摘要

针对哈里斯鹰优化算法在仓储机器人路径规划中收敛速度慢且路径过长的问题,提出了一种基于非线性混合天鹰 - 哈里斯鹰优化 (nonlinear hybrid Aquila and Harris hawks optimization, NHAHHO) 算法。首先,通过引入 Circle 混沌映射对种群进行初始化,增加种群多样性,扩大搜索范围,提高算法的收敛速度;其次,针对哈里斯鹰优化算法全局搜索阶段性能不足的情况,用天鹰优化算法中的扩展探索和缩小探索策略替换哈里斯鹰优化算法的全局搜索,提高算法的搜索性能;最后,在算法的局部开发阶段设计了非线性控制参数来平衡算法的探索和开发时间。搭建二维栅格场景模拟真实仓储进行路径规划实验,结果表明 NHAHHO 算法性能有明显提升, NHAHHO 算法在两种场景中所寻找到的最短路径和平均路径长度更短:场景一,平均路径长度为 28.158 9 m,比其他三种算法短 4.5% 以上;场景二,平均路径长度为 49.638 8 m,比其他三种算法至少短 2.9%。验证了所提算法的可行性和优越性,可以更有效解决仓储机器人路径规划问题。

关键词

哈里斯鹰优化算法;天鹰优化算法;仓储机器人;路径规划

doi: 10.3969/j.issn.1672-9528.2025.05.010

0 引言

仓储机器人作为一种更具智能化和自动化的工具,被许多大型仓储系统引入替代人工。使用仓储机器人,可以减少仓储系统劳动力数量,优化资源配置,同时也能提升系统的工作效率。仓储机器人提升仓储系统的工作效率主要取决于其路径规划:规划的路径长度越短,仓储机器人的工作效率越高,系统的工作效率也会随之提高^[1]。

目前,在仓储机器人路径规划方面主流的算法有粒子群优化 (particle swarm optimization, PSO)^[2] 算法、遗传算法 (genetic algorithm, GA)^[3] 等较为传统的启发式优化算法以及一些近年来新提出的群智能优化算法,包括天鹰优化 (aquila optimization, AO)^[4] 算法、哈里斯鹰优化 (harris hawks optimization, HHO)^[5] 算法等。其中哈里斯鹰优化算法具有参数少、原理简单和寻优速度快等优点,被广泛应用于路径规划方面^[6-8]。其中,文献[6]提出了一种基于改进的哈里斯鹰优化的无人机路径规划算法,通过引入 Cauchy 变异策略和自适应权重,增加种群多样性、扩展搜索空间并提高搜索能力。此外,为减少陷入局部极值的可能性,使用正弦余弦算法,并考虑其振荡特性,逐渐收敛到最优解。模拟结果表明,所提出的算法具有较高的优化精度、收敛速度和

鲁棒性,可以为无人机生成更优化的路径规划结果。文献[7]提出了一种多策略改进的哈里斯鹰优化算法,通过融合多种策略,提高了机器人室内路径规划的搜索性能、规划效率和路径质量。文献[8]提出了一种基于改进的哈里斯鹰优化算法的全局机器人路径规划方法,该方法通过粗粒度和细粒度规划,以及引入基于排名的动态更新策略和非线性扰动控制因子,有效地解决了路径规划问题。虽然哈里斯鹰优化算法在路径规划中的应用效果不错,但是哈里斯鹰优化算法仍有初始种群多样性不足、收敛速度慢、易陷入局部最优和寻优效果不佳等不足之处。

因此,国内外学者对 HHO 算法做出了不同的改进以优化算法的性能^[9-11]。其中,文献[9]在 HHO 算法探索阶段引入混沌映射对哈里斯鹰种群进行初始化以增加种群的多样性,扩大搜索范围,但未考虑算法探索阶段性能不足的问题。文献[10]详细介绍了天鹰算法,天鹰算法的前期全局探索能力较强,但其后期局部开发能力不足。文献[11]制定了非线性探索因子替代原有方案,提高算法的探索能力,但缺少对算法全局探索阶段的优化。上述文献对哈里斯鹰优化算法做出了各种改进,但是这类改进并没有对哈里斯鹰优化算法的优缺点进行综合考虑,仍存在收敛速度慢,寻优效果不佳等问题。

为此,本文提出了一种非线性混合天鹰 - 哈里斯鹰优

1. 湖北大学人工智能学院 湖北武汉 430062

化 (nonlinear hybrid Aquila and Harris hawks optimization, NHAHHO) 算法。引入 Circle 混沌映射对哈里斯鹰种群进行初始化, 然后将算法的全局探索阶段替换为天鹰优化算法中的扩展搜索和缩小搜索策略, 最后在局部搜索阶段利用非线性控制参数平衡算法的开发和探索时间。同时将 NHAHHO 算法应用在仓储机器人路径规划中, 解决其收敛速度慢且路径过长的问题。

1 HHO 算法介绍

HHO 算法分为全局探索和局部开发两个阶段, 由猎物的逃逸能量决定算法的阶段选取^[12]。

1.1 全局探索阶段

HHO 探索阶段的模型为:

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - c|X_{\text{rand}}(t) - 2\delta X(t)| & q \geq 0.5 \\ (X_{\text{prey}}(t) - X_M(t)) - \varepsilon(LB + \phi(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

式中: t 为当前迭代次数; $X(t)$ 为当前种群中第 t 次迭代时哈里斯鹰的位置; $X_{\text{rand}}(t)$ 为第 t 次迭代时, 种群中随机一只哈里斯鹰的位置; $X_{\text{prey}}(t)$ 为当前猎物的位置, 同时也是当前得到的最佳位置; UB 和 LB 分别为搜索空间的上界和下界; q 、 χ 、 δ 、 ε 、 ϕ 均为 $[0,1]$ 之间的随机数; $X_M(t)$ 为当前种群的平均位置, 其计算公式为:

$$X_M(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2)$$

1.2 全局探索到局部开发的过渡

猎物的逃逸能量定义为 E , 其计算公式为:

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (3)$$

$$E_0 = 2\rho - 1$$

式中: E 为逃逸能量; T 为最大迭代次数; E_0 为猎物逃逸能量的初始状态, 其大小为 $[-1,1]$ 的随机数, 每次迭代自动更新; r_s 为 $(0,1)$ 的随机值。当 $|E| \geq 1$ 时执行全局探索策略, 当 $|E| < 1$ 时执行局部开发策略。

1.3 局部开发阶段

局部开发阶段总共有 4 种策略, 用 $(0,1)$ 内的随机数 r 表示猎物逃脱状态, 当 $r < 0.5$ 时, 猎物成功逃脱概率较高, 当 $r \geq 0.5$ 时, 表示猎物逃脱失败。

1.3.1 软围攻

当 $0.5 \leq |E| < 1$ 且 $r \geq 0.5$ 时, 采取软围攻策略, 具体模型为:

$$X(t+1) = \Delta X(t) - E|J X_{\text{prey}}(t) - X(t)|$$

$$\Delta X(t) = X_{\text{prey}}(t) - X(t) \quad (4)$$

$$J = 2(1 - \varphi)$$

式中: $\Delta X(t)$ 为猎物在迭代 t 中的位置向量与当前位置之间的差值; J 为跳跃强度; φ 为区间 $(0,1)$ 的随机数。

1.3.2 硬围攻

当 $|E| < 0.5$ 且 $r \geq 0.5$ 时, 采取硬围攻策略, 公式为:

$$X(t+1) = X_{\text{prey}}(t) - E|\Delta X(t)| \quad (5)$$

1.3.3 渐进式快速俯冲的软包围

当 $0.5 \leq |E| < 1$ 且 $r < 0.5$ 时, 采用渐进式快速俯冲的软包围策略, 具体模型公式为:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (6)$$

$$Y = X_{\text{prey}}(t) - E|J X_{\text{prey}}(t) - X(t)| \quad (7)$$

$$Z = Y + S \times LF(D)$$

式中: D 是问题维度; S 是 $1 \times D$ 的随机向量; LF 是莱维飞行函数:

$$LF(x) = 0.01 \times \frac{\mu \times \sigma}{|\nu|^{\frac{1}{\beta}}} \quad (8)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (9)$$

式中: μ 、 ν 都是 $(0,1)$ 的随机数; β 是大小为 1.5 的常数。

1.3.4 渐进式快速俯冲的硬包围

当 $|E| < 0.5$ 且 $r < 0.5$ 时, 采取渐进式俯冲的硬包围更新位置, 其数学模型为:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (10)$$

$$Y = X_{\text{prey}}(t) - E|J X_{\text{prey}}(t) - X(t)|$$

$$Z = Y + S \times LF(D) \quad (11)$$

2 NHAHHO 算法

为弥补 HHO 算法存在的不足, 并提高算法的收敛速度以及寻优能力, 本小节将从三个方面对 HHO 算法进行改进。

2.1 Circle 混沌映射

HHO 算法中哈里斯鹰种群的初始位置是随机的, 分布可能不均, 会导致算法的种群多样性降低, 寻优速度受到影响, 而采用混沌映射对种群进行初始化可以有效该问题。常用的混沌映射有 Logistic 映射、Tent 映射和 Circle 映射等。其中, Logistic 映射的中间取值概率分布均匀, 而两端分布较为密集, 整体取值分布不够均匀; Tent 映射虽然具有相对均匀的取值概率分布, 但存在小周期点和不稳定周期点; Circle 映射不仅能避免 Logistic 映射和 Tent 映射出现的问题, 而且可以产生更为均匀的概率分布, 因此引进

Circle 混沌映射改进哈里斯鹰种群的初始化过程, Circle 混沌映射的表达式为:

$$X(t+1) = \text{mod} \left(X(t) + 0.2 - \left(\frac{0.5}{2\pi} \right) \sin(2\pi X(t)), 1 \right) \quad (12)$$

式中: mod 为取余函数。

相比于随机初始化的种群位置, 利用 Circle 混沌映射进行初始化的种群位置分布更加均匀, 搜索范围更大, 提高种群位置的多样性, 减小了算法陷入局部最优的可能, 提高了算法的收敛速度。

2.2 混合天鹰优化算法优化全局探索

针对 HHO 算法前期全局搜索能力差的问题, 将哈里斯鹰前期搜寻猎物时所选择的两种栖息方案更改为天鹰优化算法的扩展搜索策略和缩小搜索策略:

$$X(t+1) = \begin{cases} X_{\text{best}}(t) \times \left(1 - \frac{t}{T} \right) + (X_M(t) - X_{\text{best}}(t) \times \alpha) & q < 0.5 \\ X_{\text{best}}(t) \times \text{LF}(D) + X_R(t) + (y - x) \times \gamma & q \geq 0.5 \end{cases} \quad (13)$$

式中: $X_{\text{best}}(t)$ 表示到目前为止得到的最佳位置; $X_M(t)$ 表示当前迭代中所有哈里斯鹰位置的平均值; $X_R(t)$ 表示哈里斯鹰的随机位置; q 、 α 和 γ 表示 (0,1) 范围内的随机数; y 和 x 表示搜索中的螺旋形状; $X_M(t)$ 以及 y 和 x 的计算公式为:

$$\begin{cases} x = \lambda \times \sin(\theta) \\ y = \lambda \times \cos(\theta) \end{cases} \quad (14)$$

$$\begin{cases} \lambda = \sigma + 0.005 \cdot 65 \times D_1 \\ \theta = -\omega \times D_1 + \frac{3\pi}{2} \end{cases} \quad (15)$$

式中: σ 表示 1 到 20 之间的搜索周期数; D_1 由从 1 到问题维度大小 D 的整数组成; ω 等于 0.005。

2.3 非线性控制参数

HHO 算法作为一种元启发式算法, 不可避免地会存在无法在全局探索和局部开发两个阶段中找到最佳平衡的缺陷。因此, 在局部开发阶段设计一个非线性控制参数 b 平衡算法的探索和开发行为, b 的表达式为:

$$b = 1.6 - \arctan \left(\frac{5t}{T} \right)^5 \quad (16)$$

将 b 引入渐进式快速俯冲的软包围, 公式为:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \quad (17)$$

$$Y = bX_{\text{prey}}(t) - E \left| JX_{\text{prey}}(t) - X(t) \right|$$

$$Z = bY + S \times \text{LF}(D)$$

将 b 引入渐进式快速俯冲的硬包围, 公式变为:

$$\begin{aligned} X(t+1) &= \begin{cases} Y & \text{if } F(Y) < F(X(t)) \\ Z & \text{if } F(Z) < F(X(t)) \end{cases} \\ Y &= bX_{\text{prey}}(t) - E \left| JX_{\text{prey}}(t) - X_M(t) \right| \\ Z &= bY + S \times \text{LF}(D) \end{aligned} \quad (18)$$

2.4 时间复杂度分析

时间复杂度是衡量算法运行效率的关键指标, 当种群规模为 N , 问题维度为 D , 算法最大迭代次数为 T 时, HHO 算法的时间复杂度如文献 [5] 所示 $O(N \times (T + TD + 1))$ 。而 NHAHHO 算法引入 Circle 混沌映射和非线性控制参数, 并在算法全局探索阶段引入天鹰算法的扩展搜索策略和缩小搜索策略替换原有的位置更新公式, 但这些方案只是替换了算法的一些计算方式并没有额外增加算法的复杂程度, 所以 NHAHHO 算法的时间复杂度和 HHO 算法的一样, 仍为 $O(N \times (T + TD + 1))$ 。

NHAHHO 算法的流程图如图 1 所示。

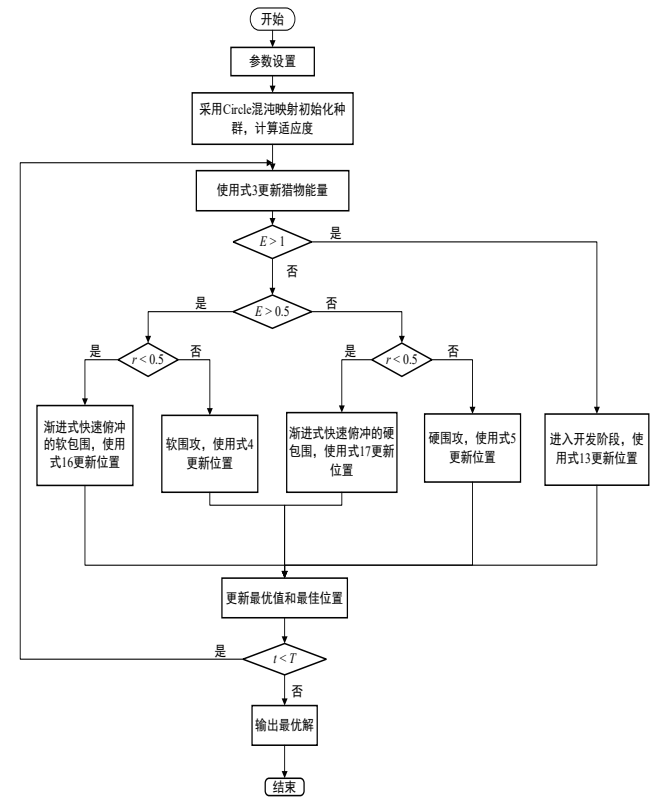


图 1 NHAHHO 算法流程图

3 实验与结果分析

3.1 仓储机器人路径规划实验

使用 NHAHHO 算法对仓储机器人进行路径规划, 通过碰撞检测消除冗余点对路径进行优化。使用栅格法进行场景模拟实验, 验证算法的可行性。

为充分发挥算法的作用, 做出 3 个假设:

(1) 把机器人视为一个没有质量和大小的质点, 不考

虑其他环境影响因素。

(2) 仓储环境看作二维环境。

(3) 仓储环境内的障碍物已知, 不用考虑高度信息。

3.1.1 环境建模

假设场景中的障碍物是静态的且数量大小已知。在静态路径规划中, 地图的构建尤为重要, 常见的构建方法有栅格法、可视图法和拓扑法等。其中栅格法因其具有原理简单, 易于路径规划算法实现等优点被广泛应用于仓储机器人路径规划实验中, 故本文采用栅格法构建地图。

栅格法把真实仓储场景还原到二维平面, 并将场景划分为一个个长宽皆为 1 m 的正方形小网格。使用二值化的方法把场景中的所有网格进行标记, 有障碍物的网格标记为“1”, 可以自由通行的网格标记为“0”, 同时将有障碍物的网格进行加黑处理, 方便识别障碍物信息。考虑到仓储机器人的大小以及机器人与移动过程中所遇障碍物之间需要保持一定的安全距离避免发生碰撞的情况, 对仓储环境中的障碍物进行膨胀处理。当障碍物的面积小于一个网格的大小时, 将障碍物所在网格整体当作障碍物, 当障碍物面积大于一个网格或几个障碍物临近时, 需要把这几个网格整体作为一个障碍物进行处理。

选取两种不同栅格模型进行实验, 场景一模拟简单环境, 场景二模拟复杂环境。两种环境栅格模型如图 2 所示。

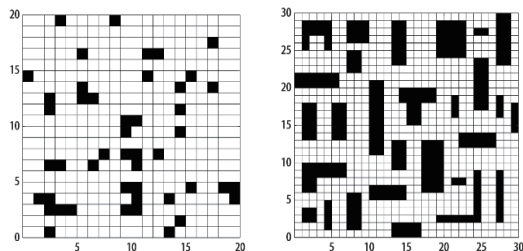


图 2 场景一和场景二环境模型

3.1.2 实验分析

选择两种不同大小的场景, 通过对比几种算法的仿真结果, 验证 NHAHHO 算法在机器人路径规划方面的合理性和有效性。将每个算法单独实验 30 次, 最大迭代次数设置为 200。

场景一的起点坐标为 (0, 0), 终点坐标为 (20, 20), 4 种算法在场景一下的路径结果如表 1 所示。4 种算法在场景一下的路径规划结果和收敛曲线如图 3、4 所示。

表 1 4 种算法在场景一下的路径比较

算法	路径长度 /m		算法收敛迭代次数	
	最优值	均值	最小值	均值
AO	29.264 6	30.848 7	149	175.300 0
HHO	28.226 4	29.489 2	8	37.133 3
NCHHO	28.226 4	29.528 3	8	36.666 7
NHAHHO	26.959 0	28.158 9	6	29.433 3

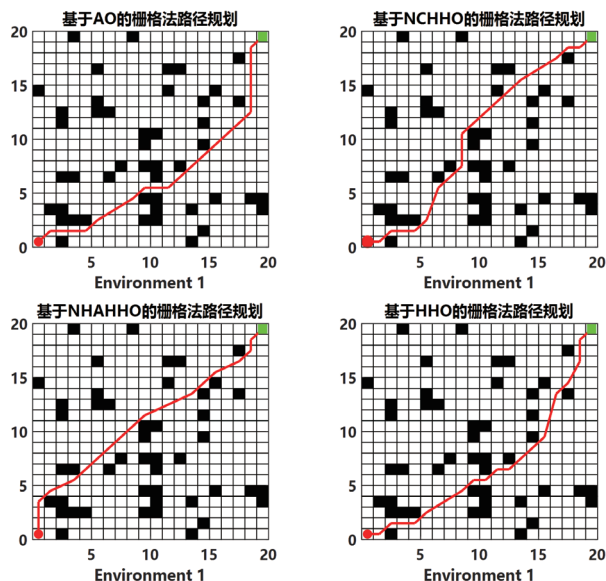


图 3 4 种算法在场景一的路径规划

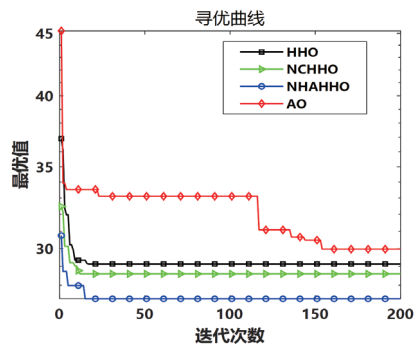


图 4 4 种算法在场景一下的收敛曲线

场景二的起点坐标为 (0, 0), 终点坐标为 (30, 30), 4 种算法在场景二的路径规划如图 5 所示, 4 种算法在场景二下的收敛曲线如图 6 所示。

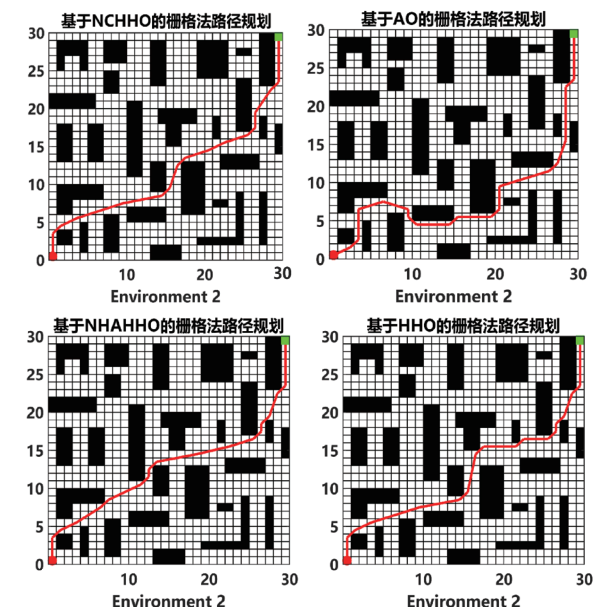


图 5 4 种算法在场景二的路径规划

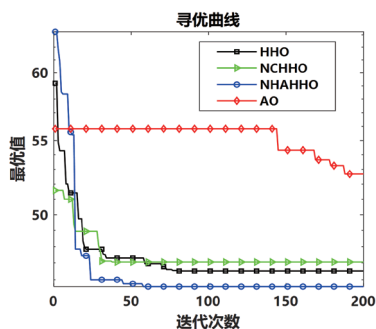


图 6 4 种算法在场景二下的收敛曲线

表 2 统计了 4 种算法在场景二的仿真结果。通过分析表 1 和表 2 中数据可知，相比于其他 3 种算法，NHAHHO 算法经过多次迭代后所搜寻到的最优路径更短，迭代次数也相对较少。在场景一中，NHAHHO 算法的平均路径长度比 NCHHO 算法短 4%，比 HHO、AO 算法至少短 4.6%，更接近最短路径长度，算法的收敛也要快很多。在场景二中，相比其他 3 种算法，NHAHHO 算法的平均路径长度要短 2.9%，算法收敛的迭代次数也相对较少。总的来说，相对于其他 3 种算法，NHAHHO 算法的寻优能力更佳，收敛速度也更快，表明该算法在机器人路径规划方面的可行性和优越性。

表 2 4 种算法在场景二下的路径比较

算法	路径长度 /m		算法收敛迭代次数	
	最优值	均值	最小值	均值
AO	50.899 0	57.415 6	162	183.700 0
HHO	46.397 0	51.864 9	13	69.300 0
NCHHO	46.759 9	51.163 7	22	59.833 3
NHAHHO	45.172 8	49.638 8	17	58.366 7

4 结论

为了更好地解决仓储机器人路径规划问题，本文提出了一种混合天鹰 - 哈里斯鹰优化（NHAHHO）算法。首先，引入 Circle 混沌映射策略，加快了算法的收敛。其次，将哈里斯鹰优化算法的全局探索阶段替换为天鹰优化算法的探索策略，提高了算法的寻优能力。然后，设计了一种非线性控制参数，平衡了算法的探索和开发行为。最后，将该算法应用到仓储机器人的路径规划上，通过搭建二维栅格场景进行模拟实验，实验结果表明，使用 NHAHHO 算法规划出的仓储机器人路径更短，且收敛速度更快，验证了该算法在仓储机器人路径规划方面的有效性和优越性。

参考文献：

[1] 黄志锋, 刘媛华, 任志豪, 等. 融合改进哈里斯鹰和改进动态窗口的机器人动态路径规划 [J]. 计算机应用研究, 2024, 41(2): 450-458.

[2] WANG F, WANG X J, SUN S L. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization [J]. Information sciences, 2022, 602: 298-312.

[3] ZHANG Z J, LU R, ZHAO M L, et al. Robot path planning based on genetic algorithm with hybrid initialization method [J]. Journal of intelligent & fuzzy systems, 2022, 42(3): 2041-2056.

[4] ABUALIGAH L, YOUSRI D, ELAZIZ M A, et al. Aquila optimizer: a novel meta-heuristic optimization algorithm [J]. Computers & industrial engineering, 2021, 157: 107250.

[5] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications [J]. Future generation computer systems, 2019, 97: 849-872.

[6] ZHANG R, LI S, DING Y M, et al. UAV path planning algorithm based on improved harris hawks optimization [J]. Sensors, 2022, 22(14): 5232.

[7] LI C Y, SI Q, ZHAO J N, et al. A robot path planning method using improved harris hawks optimization algorithm [J]. Measurement and control, 2024, 57(4): 469-482.

[8] LI G Q, ZHU D Q, YU Y, et al. Robot path planning based on improved harris hawk optimization algorithm [C]//2022 5th International Conference on Intelligent Autonomous Systems. Piscataway: IEEE, 2022: 209-214.

[9] CHEN H, HEIDARI A A, CHEN H L, et al. Multi-population differential evolution-assisted harris hawks optimization: framework and case studies [J]. Future generation computer systems, 2020, 111: 175-198.

[10] YU H J, JIA H M, ZHOU J P, et al. Enhanced aquila optimizer algorithm for global optimization and constrained engineering problems [J]. Mathematical biosciences and engineering, 2022, 19(12):14173-14211.

[11] DEHKORDI A A, SADIQ A S, MIRJALILI S, et al. Nonlinear-based chaotic harris hawks optimizer: algorithm and internet of vehicles application [J]. Applied soft computing, 2021, 109: 107574.

[12] 汤安迪, 韩统, 徐登武, 等. 混沌精英哈里斯鹰优化算法 [J]. 计算机应用, 2021, 41(8): 2265-2272.

【作者简介】

武紫豪（2001—），男，河南商丘人，硕士研究生，研究方向：路径规划、群智能优化算法。

杨紫伊（1999—），女，四川成都人，硕士研究生，研究方向：车载通信、智能反射面。

（收稿日期：2024-12-13 修回日期：2025-05-06）