

文章编号: 1007-7294(2025)12-1885-10

北方苍鹰算法优化快速扩展随机树的 无人艇路径规划研究

辜慧岚, 马国军, 张 龙, 程理泽, 王亚军

(江苏科技大学海洋学院, 江苏 镇江 212003)

摘要: 针对无人艇路径规划中的路径冗余及算法执行时间长的问题, 本文提出了一种融合北方苍鹰优化算法 (NGO) 与改进快速扩展随机树 (RRT*) 的路径规划方法。首先, 设计具有避障及目标导向的适应度函数, 并用该函数优化 NGO 初始种群, 根据该适应度函数设计 RRT* 算法的自适应采样步长, 提高在地图中的搜索效率; 然后, 设计最佳邻节点采样机制, 该机制模拟北方苍鹰向最近同伴传递信息行为, 结合无人艇航向角约束进行 RRT* 节点采样; 最后, 为提高路径平滑度, 引入模拟退火算法中的 Metropolis 准则, 并结合具有平滑度和最小路径曲率信息的适应度函数用于判断父节点优劣, 设计动态重布线策略, 该策略能够重新选择更合适的父节点进行动态重布线。实验结果表明, 与 RRT*、Informed-RRT* 和 RRT*-smart 算法相比, 本文改进算法在路径长度上分别缩短 19.36%、3.36% 和 5.98%, 搜索时间上分别减少 49.33%、57.01% 和 59.16%, 同时路径曲率也明显减小。

关键词: 水面无人艇; 路径规划; RRT* 算法; 北方苍鹰优化算法; 动态重布线

中图分类号: U664.82 **文献标识码:** A **doi:** 10.3969/j.issn.1007-7294.2025.12.006

Path planning for USVs based on improved NGO-RRT*

GU Hui-lan, MA Guo-jun, ZHANG Long, CHENG Li-ze, WANG Ya-jun

(Ocean College, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: In order to solve the problem of path redundancy and long algorithm execution time, this paper proposed a path planning method that combines the Northern Goshawk Optimization (NGO) algorithm with the improved rapidly-exploring random tree (RRT*). First, a fitness function with obstacle avoidance and goal orientation was designed to optimize the initial NGO population. Additionally, the adaptive sampling step size of the RRT* algorithm was designed according to the fitness function to improve the search efficiency in a large-scale map. Then, the optimal neighbor node sampling mechanism was designed to simulate behavior of the northern goshawk transmitting information to its nearest companions, while also the RRT* node sampling was constrained by considering the USV's (unmanned surface vehicle) heading angle. Finally, in order to improve path smoothness, the Metropolis criterion was introduced and the smoothness and minimum rudder angle design fitness function were combined to select a more suitable parent node for dynamic rerouting. The experimental results show that compared with RRT*, Informed-RRT* and RRT*-smart algorithms, the improved algorithm reduces the path length by 19.36%, 3.36% and 5.98%, and decreases the search time by 49.33%, 57.01% and 59.16%, respectively. At the same time, the curvature of the path also decreases significantly.

收稿日期: 2025-05-28

基金项目: 国家自然科学基金资助项目 (61371114)

作者简介: 辜慧岚 (2000-), 女, 硕士研究生;

马国军 (1976-), 男, 博士, 副教授, 通讯作者, E-mail: 2013mgj@sina.com。

Key words: unmanned surface vehicle; path planning; RRT* algorithm; NGO algorithm; dynamic rewiring

0 引言

近年来,海洋资源的进一步开发和海上作战的需求推动了水面无人艇(Unmanned Surface Vehicle, USV)的研究^[1]。无人艇因其体积小、灵活且执行力强等优点,被广泛应用于日常及军事活动中^[2]。路径规划是保障无人艇自主、安全航行的一项关键技术。由于无人艇有其自身的运动特性和航行特点,且容易受到风浪、水流等环境因素的影响,水面和水下的障碍物都可能危及到无人艇的航行安全。因此,在无人艇路径规划中必须充分考虑其操纵特性以及环境因素对航行带来的挑战。文献[3]考虑了无人艇航行特点和动态操纵特性,设计了与障碍物相对的方位角信息和动态转艏限制角,进行路径规划。文献[4]利用无人艇运动学特性,使无人艇远离障碍物较多的区域,同时引入偏流角减少水流环境对路径规划的影响。

改进的快速扩展随机树算法(Improved Rapidly-exploring Random Tree, RRT*)^[5-6]在全局路径规划中随机生成节点形成路径,通过多次迭代得到一条渐近最优的路径,但该算法在无人艇路径规划中存在对密集障碍物的处理复杂、运行时间较长以及平滑连接难度大等问题。文献[7-8]针对随机树收敛速度慢的问题,提出了改进的双向RRT的无人艇路径规划算法,对延伸节点增加转向角约束,使两棵随机树平滑连接。姜兆祯等^[9]提出了改进的人工势场法结合RRT*方法,考虑无人艇四个方向所受合力避免陷入局部最优以此获得初始路径,根据初始路径限制RRT*的采样区域,提高算法鲁棒性。Lin等^[10]针对电子海图环境下无人艇采用RRT*算法采样效率低等问题,提出了Informed-RRT*算法,设置了目标采样区域,增强了目标区域内随机树的探索能力,提高了整体性能。Wen等^[11]针对RRT*算法在路径规划时产生高碰撞率、易受水流干扰等问题,提出了基于双采样域的启发式RRT*算法,设置了椭圆区域采样空间,并通过基于安全距离的碰撞检测,使无人艇可以更加安全地避开障碍物区域。

虽然以上改进方法在缩短路径长度方面取得了显著进展,但采样效率无明显提高。近年来,群智能算法已经广泛应用于工程优化问题中。文献[12-13]提出了蚁群算法结合RRT*的路径规划方法,通过蚁群算法进行分区采样,限制RRT*采样区域。Huynh等^[14]提出了粒子群算法结合RRT*的路径规划算法,利用粒子群将随机树扩展以调整和增强算法性能。然而以上算法并未充分考虑无人艇在路径规划中的运动和操纵特性,本文利用2021年首次提出的北方苍鹰优化算法(Northern Goshawk Optimization, NGO)^[15],设计了基于改进北方苍鹰优化的RRT*算法。首先,针对RRT*算法采样范围过大的问题,提出了两个适应度函数来约束NGO初始种群,限制RRT*的采样范围。对于避障能力不足的问题,模拟了北方苍鹰的捕猎行为,并提出了结合无人艇动力学约束的适应度函数计算方法,以此调整RRT*算法自适应步长;其次,设计了动态重布线机制,并重新选择了更优的RRT*父节点,以此得到较优初始路径;最后引入Metropolis准则并结合约束条件,提高路径平滑度。仿真结果表明,采用改进算法规划的路径,其运行时间明显缩短,路径更加平滑,更有利于无人艇的航行。

1 传统RRT*算法

RRT*算法旨在生成一条从初始节点 Q_{init} 到目标节点 Q_{goal} 的无碰撞路径。图1为随机树生成新节点 Q_{new} 示意图,以 Q_{init} 为根节点,每次采样中随机选择 Q_{rand} ,并在整个树中找到距离最近的节点 $Q_{nearest}$ 。然后,根据预定义的步长,在 $Q_{nearest}$ 到 Q_{rand} 方向根据障碍物信息确定新节点 Q_{new} 。

RRT*算法通过重新选择父节点和重布线两个关键步骤优化随机树,使其趋向于生成最优路径。如图2(a)所示, Q_{new} 初始父节点为 $Q_{nearest}$,在以 Q_{new} 为圆心,半径为 r 的圆域内,所有节点都被视为候选父节点,选择路径代价最小的节点作为 Q_{new} 的最新父节点。如图2(b)所示,重新选择 Q_1 作为 Q_{new} 的父节点。

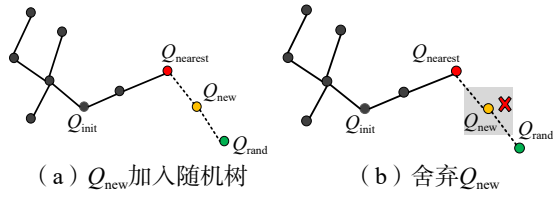
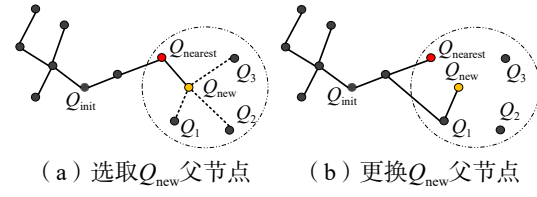
图 1 新节点 Q_{new} 选取Fig.1 Selection of new node Q_{new} 

图 2 父节点重新选取

Fig.2 Re-selection of the parent node

图 3 为 RRT* 算法重布线示意图, 在 Q_{new} 找到其最新父节点后进行重布线, 在圆域范围内计算节点之间路径代价, 去除高代价路径。如图 3(b) 中去除 Q_3 和 $Q_{nearest}$ 之间的连线, 连接 Q_3 与 Q_{new} , 更新随机树的结构, 减少冗余路径。

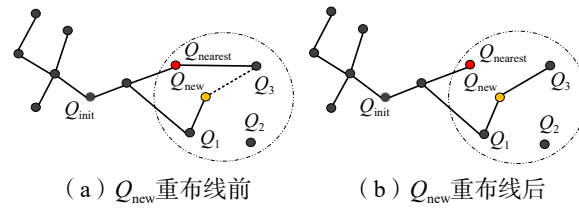


图 3 重布线

Fig.3 Rewiring

2 北方苍鹰优化算法

北方苍鹰优化算法是一种受北方苍鹰的自然行为启发的基于群的优化算法^[15], 它模拟鹰群捕获猎物的行为, 将捕猎策略分为识别猎物和逃生追逐两个阶段。

在识别猎物的第一阶段, 北方苍鹰随机选择一只猎物并且对其发起攻击, 通过全局搜索来识别最优区域。第 i 只北方苍鹰所要捕获的猎物位置为

$$P_i = X_k \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, i-1, i+1, \dots, N \quad (1)$$

第一阶段后第 i 只北方苍鹰第 j 维的新位置为

$$x_{i,j}^{new,P1} = \begin{cases} x_{i,j} + \delta(p_{i,j} - I * x_{i,j}), & F_{p_i} \leq F_i \\ x_{i,j} + \delta(x_{i,j} - p_{i,j}), & F_{p_i} \geq F \end{cases} \quad (2)$$

式中: F_{p_i} 为目标函数值, X_k 表示第 k 只北方苍鹰的位置, 参数 δ 和 I 是用于在搜索和更新中生成 NGO 行为的随机数, 其中 $\delta \in [0, 1]$, I 为 1 或 2。北方苍鹰种群的新位置为

$$X_i = \begin{cases} X_i^{new,P1}, & F_i^{new,P1} < F_i \\ X_i, & F_i^{new,P1} \geq F_i \end{cases} \quad (3)$$

式中: $F_i^{new,P1}$ 代表第一阶段更新后第 i 只北方苍鹰的目标函数值, $X_i^{new,P1}$ 代表第一阶段后第 i 只北方苍鹰的新位置。

在逃生追逐的第二阶段, 猎物试图逃跑, 因此北方苍鹰会继续追逐猎物并完成猎杀, 从而提高 NGO 算法局部探索能力。此时, 北方苍鹰位置为

$$x_{i,j}^{new,P2} = x_{i,j} + Z \cdot (2\delta - 1) \cdot x_{i,j} \quad (4)$$

$$Z = 0.02 \left(1 - \frac{t}{T} \right) \quad (5)$$

式中: Z 表示北方苍鹰狩猎范围的半径, $x_{i,j}^{new,P2}$ 表示第二阶段后第 i 只北方苍鹰第 j 维的新位置, X_k 表示第二阶段中第 k 只北方苍鹰的位置, t 表示当前迭代的次数, T 表示最大迭代次数。第二阶段北方苍鹰

种群新位置为

$$X_i = \begin{cases} X_i^{\text{new},P2}, & F_i^{\text{new},P2} < F_i \\ X_i, & F_i^{\text{new},P2} \geq F_i \end{cases} \quad (6)$$

式中: $F_i^{\text{new},P2}$ 表示第二阶段更新后第 i 只北方苍鹰的目标函数值。 $X_i^{\text{new},P2}$ 表示第二阶段中第 i 只北方苍鹰的新位置。

3 NGO-RRT*算法

3.1 改进的 NGO 种群初始化

传统 NGO 算法随机生成初始种群, 容易出现种群分布不均匀的问题, 导致算法性能较低且收敛速度慢, 如式(7)所示的 Tent 混沌映射常作为种群初始化的一种方法。

$$x_{i+1} = \begin{cases} \frac{x_i}{u}, & 0 \leq x_i < u \\ \frac{1-x_i}{u}, & u \leq x_i < 1 \end{cases} \quad (7)$$

在无人艇路径规划中, 为使初始种群能朝着靠近目标点位置延伸, 从而减少对无效空间的搜索, 本文设计包含障碍物与目标点信息的第一类适应度函数 f_1 , 结合 Tent 混沌映射优化北方苍鹰初始种群, 初始化北方苍鹰个体位置为

$$x_{i+1} = \begin{cases} \frac{x_i}{f_1}, & 0 \leq x_i < f_1 \\ \frac{1-x_i}{1-f_1}, & f_1 \leq x_i < 1 \end{cases} \quad (8)$$

式中: f_1 为包含障碍物与终点信息的第一类适应度函数, 且 $f_1 \in (0,1)$, 其定义为

$$f_1 = \alpha \left(1 - \frac{d_{\text{obs}}}{d_{\text{obs}} + d_{\text{goal}}} \right) \quad (9)$$

式中: α 表示扰动因子, d_{goal} 和 d_{obs} 分别表示当前点到目标点和最近障碍物的欧氏距离。

3.2 RRT*自适应步长

在传统 RRT*算法中, 采样步长为固定值, 难以适应不同复杂度的地图。当采样步长过大时, 难以精细探索空间, 易错过最佳路径甚至碰撞障碍物; 若步长过小, 则需要更多迭代次数, 从而导致路径冗余, 增加计算成本^[16]。为避免固定步长产生的问题, 本文在随机树采样时设计动态自适应步长, 使无人艇根据环境信息动态改变采样步长。

如图4所示, 以无人艇中心点为圆心, T 为半径, 设置无人艇安全阈值, 若当前节点 Q_i 与障碍物之间距离大于 T , 则将当前所处环境定义为安全区域, 此时与障碍物碰撞概率较小; 反之, 则定义为风险区域。在规划无人艇路径时, 在未达到目标采样区域且距离障碍物较远时, ρ 取较小值, 无人艇采用小步长; 当障碍物相对较少时, ρ 取值较大, 无人艇采用大步长以减少搜索时间。

自适应采样步长为

$$\rho = \begin{cases} \rho_{\max}, & d(Q_i, Q_{\text{obs}}) \geq T \\ \mu[(f_1 \cdot d_{\text{goal}}) + (1-f_1)d_{\text{obs}}] \rho_{\max}, & d(Q_i, Q_{\text{obs}}) < T \end{cases} \quad (10)$$

式中: μ 为比例权重, $\rho_{\max}$ 为最大步长, ρ 为采样步长, 根据所处环境适应度函数 f_1 动态调整。

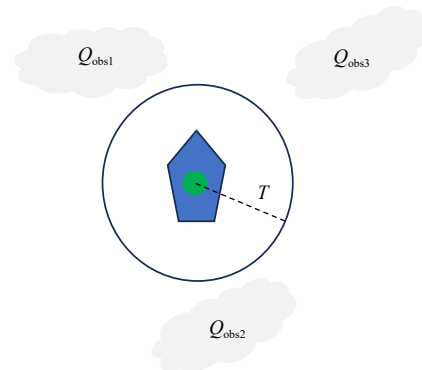


图4 无人艇安全区域图

Fig.4 Safety area map of USV

3.3 采样机制改进策略

3.3.1 最佳邻节点采样机制

无人艇在行驶过程中易受到风与海流等外界因素的干扰,增加能量消耗,导致偏离航向,增加了无人艇与障碍物碰撞概率^[16]。为减少外界因素对无人艇的影响,本文在邻节点选取过程中添加航向角偏差。图 5 为无人艇运动模型示意图,图 5(a)为静水环境中无人艇航向图, $\vec{OA}$ 为当前无人艇前进方向;图 5(b)为外界因素影响下无人艇航向图, $\vec{OA}$ 为无人艇初始航向, $\vec{OB}$ 为实际航向, $\vec{OC}$ 为外界因素对无人艇阻力方向, θ_{act} 为航向角度,且 $\theta_{act} \in (0, \pi/4)$,定义为

$$\theta_{act} = \arctan(\sin(\theta_{goal} - \theta_{current}), \cos(\theta_{goal} - \theta_{current}) + F_{res}) \quad (11)$$

其中, $\theta_{current}$ 表示当前艇体朝向角度, F_{res} 表示阻力影响。

$$F_{res} = -D \cdot V_{act} \cdot \cos(\theta_{res} - \theta_{act}) \quad (12)$$

其中, D 为阻尼系数,表示外界因素对无人艇航行速度的衰减程度, V_{act} 为无人艇实际航速。

RRT*采样过程中随机选取采样节点,导致算法鲁棒性差,采样效率低。NGO 算法中北方苍鹰探索空间时,选择其附近最合适的对象传递猎物信息。本文模拟其行为,并结合无人艇航向角偏差,设计了最佳邻节点采样机制,如图 6 所示。首先,定义无人艇初始位置 Q_{init} 为当前北方苍鹰所处位置,将随机树采样节点视为潜在的猎物。其次,在无人艇初始位置附近随机生成一个节点 Q_{rand} 。接着,以无人艇所处位置为圆心、安全阈值 T 为半径的圆域边缘设置采样节点,作为候选邻节点。然后根据自适应采样步长范围结合适应度函数值评估所有候选邻节点,选出相对较优的节点作为最佳邻节点 Q_{nei} 并加入随机树中。最佳邻节点位置为

$$Q_{nei} = \left(|Q_{rand} - Q_{init}| + |Q_{rand} - Q_{goal}| + |Q_{rand} - Q_{obs}| \right)_{\min} \quad (13)$$

其中, Q_{rand} 表示随机节点位置, Q_{init} 表示初始点位置。在最佳邻节点 Q_{nei} 与随机点 Q_{rand} 连线方向上,以 Q_{nei} 为起点,确定新节点 Q_{new} 。此时,采样的新节点坐标为

$$\begin{cases} x_{new} = x_{nei} + \rho \cos \theta_{act} \\ y_{new} = y_{nei} + \rho \sin \theta_{act} \end{cases} \quad (14)$$

式中: θ_{act} 表示航向角度, (x_{nei}, y_{nei}) 表示最佳邻节点坐标。

3.3.2 动态重布线

RRT*随机树在选择 Q_{new} 父节点时,其选择范围受限于固定圆域,如图 7(a)所示。为了适应不同复杂度的地图,优化路径的同时降低时间成本,本文提出模拟北方苍鹰动态追击过程的动态重布线策略,设计满足无人艇最小曲率约束和路径平滑度最大化的适应度函数 f_2 ,并结合模拟退火算法 Metropolis 准则,改进重选 Q_{new} 父节点机制并进行动态重布线减少路径冗余。

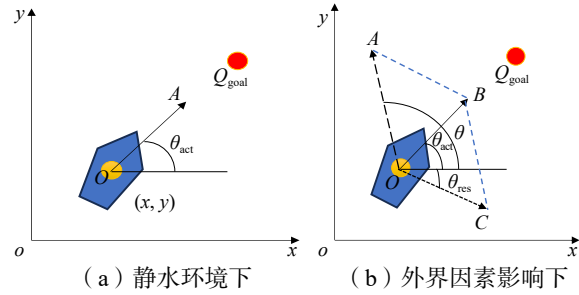


图 5 无人艇运动模型示意图

Fig.5 Model diagram of USV motion

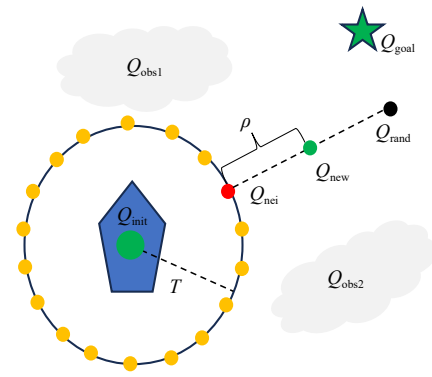


图 6 Q_{nei} 采样图

Fig.6 Sampling pattern of Q_{nei}

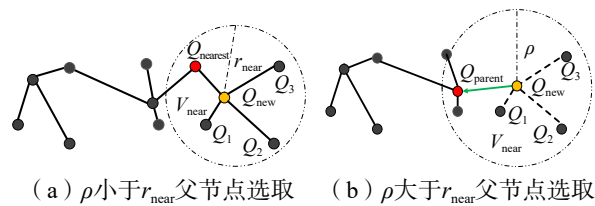


图 7 改进父节点选取

Fig.7 Selection of improved parent node

定义 Q_{new} 的候选父节点集合 V_{near} 为一半径为 r_{near} 的圆域。动态选择父节点的具体步骤如下

- (1) 以 Q_{new} 为圆心、 r_{near} 为半径计算出 V_{near} 集合范围;
- (2) 引入 Metropolis 准则判断是否需要更换父节点, 概率计算公式为

$$p = \begin{cases} 1, & f_{\text{new}} > f_{\text{current}} \\ e^{\frac{-(f_{\text{current}} - f_{\text{new}})}{T}}, & f_{\text{new}} \leq f_{\text{current}} \end{cases} \quad (15)$$

式中: f_{new} 和 f_{current} 分别代表新节点和当前节点代入适应度函数 f_2 中所得的适应度值, f_2 函数定义为

$$f_2 = \omega_1 C_{\min} + \omega_2 S \quad (16)$$

式中: $C_{\min}$ 为无人艇最小路径曲率, S 为平滑度值, ω_1 和 ω_2 为权重系数。

- (3) 根据式(10)计算当前 Q_{new} 的步长 ρ 。

(4) 确定父节点更换区域, 比较 ρ 和半径 r_{near} 的大小。如果 $\rho \leq r_{\text{near}}$, 如图 7(a) 所示, 则以 r_{near} 为半径, Q_{new} 为圆心, RRT* 算法在此范围内寻找最合适的父节点。遍历集合 V_{near} 中所有节点, 根据式(15)判断是否替换 Q_{new} 父节点。

(5) 若 $\rho > r_{\text{near}}$, 如图 7(b), 则以 ρ 为半径重新作圆, 重新划分父节点集合范围, 遍历新区域内所有节点, 根据路径代价选出 Q_{new} 新的父节点。

(6) 若父节点产生变化, 则重新计算范围内节点代价, 如图 7(b), 去除点 Q_{nearest} 及其与 Q_{new} 的连线。

综上所述, NGO-RRT* 算法具体流程如图 8 所示。首先, 设置无人艇最大步长 $\rho_{\max}$, 最大速度限制 $V_{\max}$, 北方苍鹰种群数量 N 和最大迭代次数 M 。接着, 利用 Tent 混沌映射结合第一类适应度函数 f_1 进行北方苍鹰种群初始化。然后, 根据适应度函数 f_1 公式计算当前节点采样步长。其次, 模拟 NGO 捕食策略设计最佳邻节点机制, 结合无人艇动力学约束, 确定相对较优的随机树新节点位置; 接着, 进行碰撞检测, 若新节点落在安全可行域则将新节点加入随机树, 继续执行节点采样, 否则去除该节点重新采样; 采样成功后回溯路径得到基本路径; 最后, 通过引入 Metropolis 准则和适应度函数 f_2 , 设计动态重布线策略, 以确定 Q_{new} 的相对较优父节点 Q_{parent} , 从而得到平滑的路径。

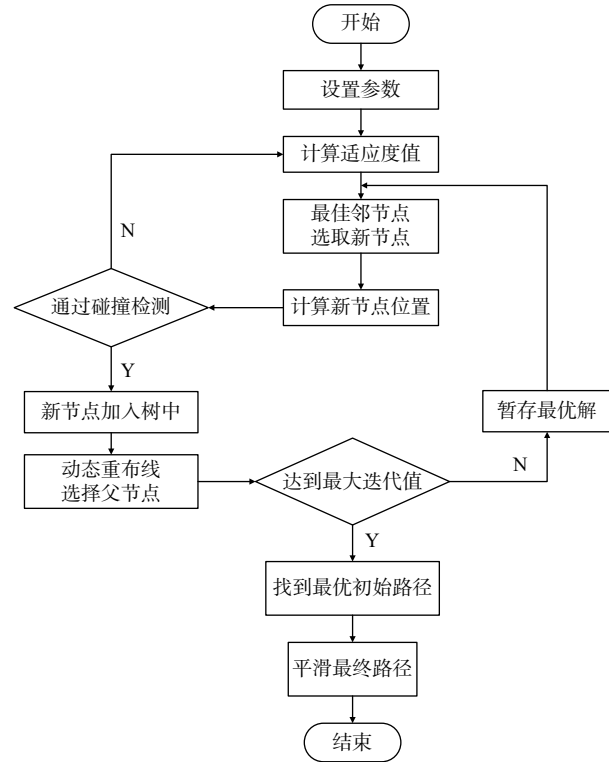
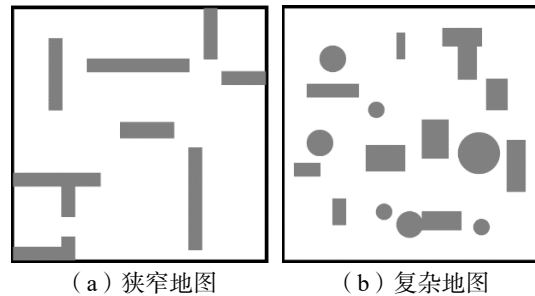


图 8 NGO-RRT* 算法流程图

Fig.8 Flow chart of NGO-RRT* algorithm

4 仿真实验

本文实验计算机为 Intel(R) Core i5-12500H CPU@2.5 GHz, 16 GB 内存; 操作系统为 Windows 11, 编程语言为 Python 3.9, 编程环境为 Visual Studio 2021。为了验证本文所提出的无人艇路径规划算法的性能, 我们设计了两种地图: 一种是狭窄障碍物环境, 另一种是障碍物较多的复杂环境, 如图 9 所示。地图大小均为



(a) 狭窄地图

(b) 复杂地图

图 9 地图空间

Fig.9 Planning area

100 m×100 m, 灰色区域为障碍物, 白色区域为无人艇可行域, 无人艇初始位置设置为[6 m, 10 m], 目标点位置为[90 m, 90 m]。

将 NGO-RRT*算法与 RRT*、Informed-RRT*和 RRT*-smart^[17] 分别进行 50 次仿真实验, 通过路径长度、算法搜索时间和路径平滑度三个指标评价算法性能。其中, 路径平滑度利用路径曲率判断, 每隔十个采样节点计算一次路径曲率, 曲率越小, 路径越平滑。本文实验中设计最大步长为 3 m, 无人艇长度为 2 m, 最大转向角为 $\pi/4$, 安全距离为 1 m, NGO 种群大小为 20, 最大迭代次数为 1500。

四种算法规划路径的实验结果如图 10 所示。在狭窄地图中, RRT*算法具有随机性, 随机树生长路线不具备倾向性, 只能遍历全区域才能找到路径, 路径曲折且较长, 路径质量不高。Informed-RRT*算法采用目标偏置采样策略, 使其采样范围具有目标引导性, 能够减少采样范围, 但仍存在大量重复的无效采样且路径曲折; RRT*-smart 算法优化了路径曲折度, 但在狭窄地图中采样范围没有得到限制, 增加了无效采样。本文算法利用障碍物信息, 结合 Tent 混沌映射和适应度函数, 生成偏离障碍物并偏向目标点的初始种群, 减少无效采样区域。设计自适应步长, 根据无人艇与障碍物和目标点的距离动态调整采样步长, 路径曲折度和路径采样效率均得到明显改善, 且整体路径不存在过大转向角。

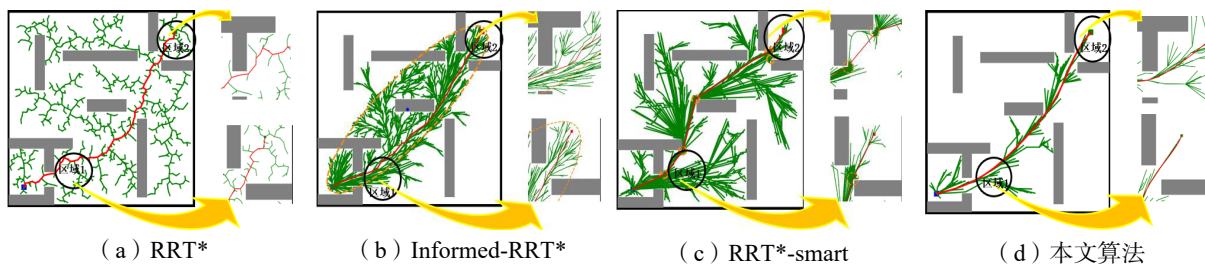


图 10 100 m×100 m 狭窄地图

Fig.10 Narrow map of 100 m×100 m

为了观察局部路径质量, 我们分别取图 10 中区域 1 和区域 2 两处位置的局部放大图进行对比。结果显示, RRT*算法路径具有较强的随机性, 在区域 1 和区域 2 的路径均出现较多不必要的拐点; Informed-RRT*路径质量提高, 但仍存在明显拐点; 而 RRT*-smart 转向角数量相对减少, 但弯曲处转向角过于尖锐。相比之下, 本文算法模拟北方苍鹰择优选择消息传递对象行为, 由无人艇航向角约束的最佳邻节点进行节点采样, 确定新节点位置。在动态重布线中, 我们根据路径最小曲率和平滑度最大化的适应度函数, 通过动态重布线改进了 RRT*随机树的父节点选取, 限制并及时调整路径转向角度, 因此几乎不存在较大转向角。这可以看出, 在狭窄地图中, 本文算法更有利于无人艇航行。

在障碍物较多的复杂地图中, 如图 11 所示, RRT*算法形成的最终路径是由多条短路径组成的。Informed-RRT*算法在复杂地图中搜索时, 目标导向性能降低, 采样范围增大至几乎全地图, 目标采样率降低且路径长度无明显改善; RRT*-smart 算法路径曲折度和采样效率都得到明显提高; 在路径质量方面, 与 RRT*算法和 Informed-RRT*算法相比, 本文算法的适应度函数包含了障碍物和目标点信息,

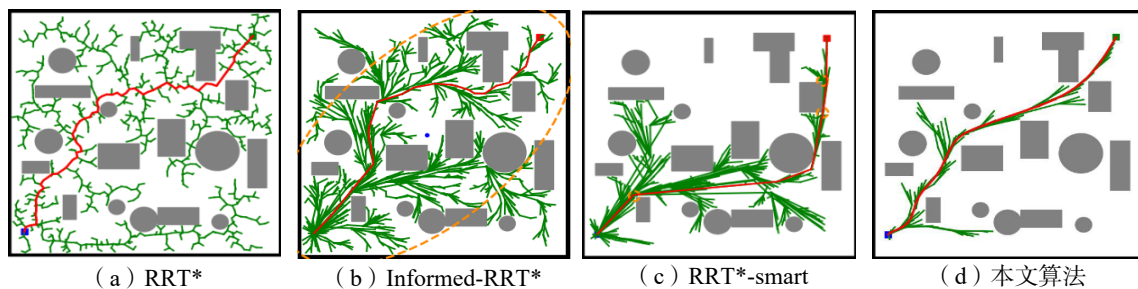


图 11 100 m×100 m 复杂地图

Fig.11 Complex map of 100 m×100 m

确保采样过程中能在远离障碍物的前提下最大程度接近目标点,通过最佳邻节点机制提高随机树采样效率,减少无效采样次数。本文算法利用模拟退火算法的 Metropolis 准则判断是否需要更换采样节点的父节点,实现动态重布线,使路径质量明显提高,采样范围更小。

本文算法与其他三种算法在两种地图中寻找最优路径的平均搜索时间和路径长度如表 1 所示。RRT*算法中路径长度最长且算法搜索时间相对较长; Informed-RRT*算法和 RRT*-smart 算法中路径长度明显改善,但是搜索时间也明显增加;本文算法通过结合目标导向适应度函数和 Tent 混沌映射优化北方苍鹰初始种群,使随机树朝向目标点生长,从而减少随机性并缩短探索时间。同时,为了判断新节点位置是否需要更换,设计动态重布线缩短路径长度。本文算法生成的路径在保障无人艇与障碍物无碰撞行驶的前提下,路径长度最短且搜索速度最快,相比于 RRT*算法、RRT*-smart 算法和 Informed-RRT*算法,在狭窄障碍物地图中,本文算法平均路径长度分别缩短 15.20%、2.35% 和 3.34%,搜索时间分别减少 40.56%、49.53% 和 48.80%。在复杂地图中,平均路径长度分别缩短 23.52%、4.37% 和 8.62%,搜索速度分别提升 58.09%、64.49% 和 69.52%。这表明本文算法在两种地图中均具有较高的路径质量和搜索效率。

表 1 算法性能对比
Tab.1 Comparison of four algorithms

		RRT*	Informed-RRT*	RRT*-smart	本文算法
地图 (a)	长度/m	140.98	122.43	123.68	119.55
	时间/s	1.80	2.12	2.09	1.07
地图 (b)	长度/m	154.25	148.96	134.96	123.32
	时间/s	2.72	3.21	3.74	1.14

在两种地图中,对四种算法在路径质量方面进行实验,图 12 为算法在狭窄地图中的对比图。图 12 (a)为路径曲率盒线图,图 12(b)为路径长度收敛时间折线图。由图 12(a)可知,在无其他平滑策略前提下, RRT*算法路径曲率的最大值、最小值以及平均值都明显大于其他算法,说明 RRT*算法路径最曲折; Informed-RRT*和 RRT*-smart 曲率数值接近,整体曲率略大于本文算法。本文算法路径曲率值较为集中,大部分曲率小于 0.1,说明本文算法稳定且路径平滑。由图 12(b)可知,所有算法均具有最优收敛的性质,其中本文算法收敛到最优路径时长为 1.25 s,仅为 RRT*算法(2.60 s)的 48.08%, Informed-RRT*算法(2.99 s)的 41.67% 和 RRT*-smart 算法(3.40 s)的 36.76%。

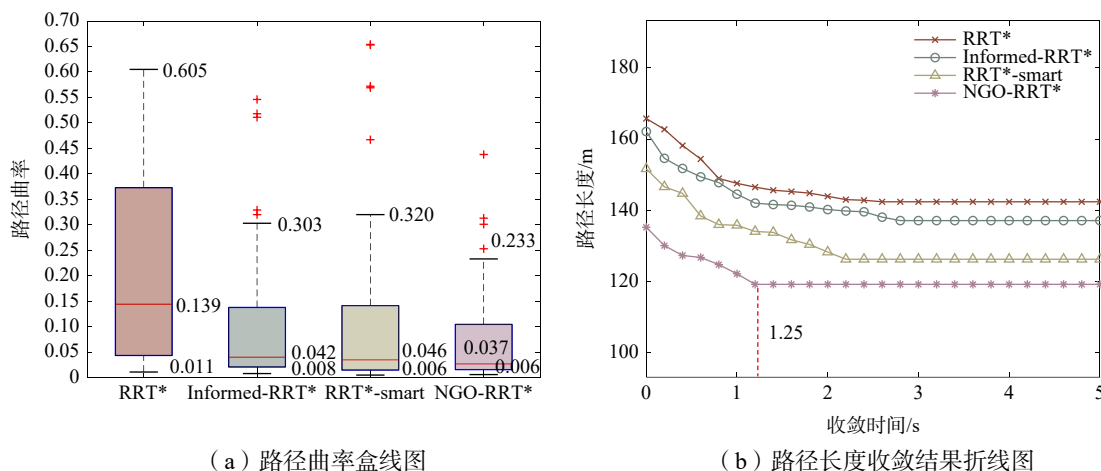


图 12 算法在狭窄地图中对比图

Fig.12 Comparison of algorithms in narrow map

图 13 为算法在复杂地图中的对比图,图 13(a)为路径曲率盒线图,图 13(b)为路径长度收敛时间折线图。RRT*算法和 Informed-RRT*路径曲率明显大于其他算法。本文算法相对 RRT*-smart,在纵轴上跨度和曲率均较小。与 RRT*、Informed-RRT*和 RRT*-smart 相比,本文算法路径曲率无论在平均值还是离散度上均明显较优,表明本文算法路径更加平滑。由图 13(b)可知,本文算法收敛到最优路径时长为 1.40 s,仅为 RRT*算法(2.59 s)的 53.85%,Informed-RRT*算法(3.01 s)的 46.47% 和 RRT*-smart 算法(3.40 s)的 41.18%。综上所述,本文算法在收敛时间、路径长度和路径平滑度三个方面均具有明显优势。

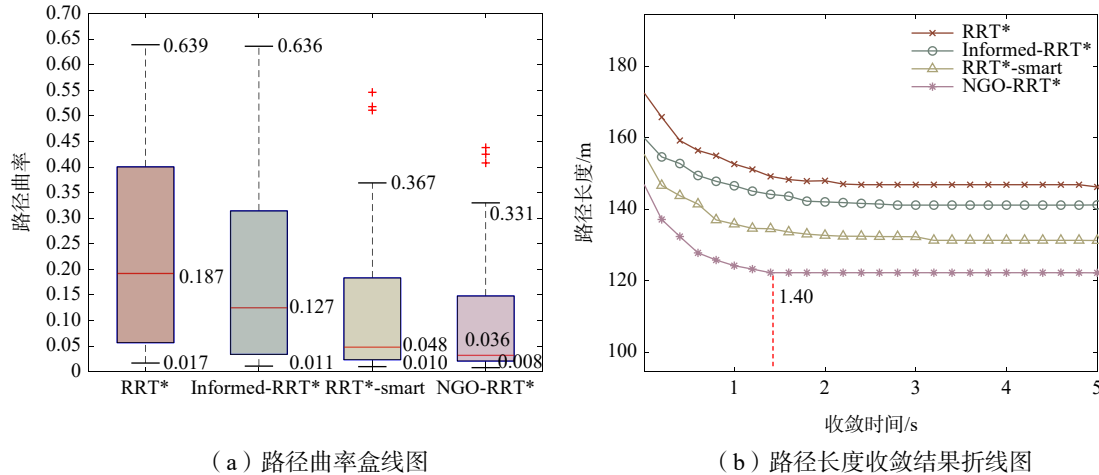


图 13 算法在复杂地图中对比图

Fig.13 Comparison of algorithms in complex map

5 结 论

本文针对无人艇采用 RRT*算法在狭窄地图和复杂地图中进行路径规划时产生过多冗余节点、搜索时间过长的的问题,提出了一种基于 NGO 优化的 RRT*算法。仿真实验结果表明:

(1)根据无人艇航行环境障碍物信息,本文设计避障与目标导向结合的适应度函数用以优化北方苍鹰初始种群,快速收敛到较优解,这不仅减少了算法的搜索空间和收敛时间,还使得算法更具鲁棒性,提高了算法的效率。

(2)基于北方苍鹰捕食猎物行为设计的自适应步长,可以根据环境的复杂程度动态调整。本文针对复杂地图和狭窄障碍物地图,通过评估当前解的适应度,无人艇可以智能地调整步长,以适应不同类型地图的需求。

(3)根据北方苍鹰全局搜索时的行为和无人艇操纵特性,本文提出了结合无人艇航向角约束的最佳邻节点机制,通过衡量最近点的优劣,选择相对较优的采样节点,提高了算法的运行效率,并能够灵活地调整采样方向。

(4)针对路径冗长及不平滑问题,本文提出了满足第二类适应度函数约束的动态重布线机制。该机制通过重新选择父节点使路径更平滑,同时算法能快速收敛到相对较优路径,进一步提高路径规划的质量和效率。

在仿真验证中,改进的 NGO-RRT*算法运行时间更短,路径更加平滑,更利于无人艇的航行。与其他三种算法相比,本文算法在路径长度和搜索时间等方面均取得了明显改善。

参 考 文 献:

- [1] 高霄鹏, 刘冬雨, 霍 聪. 水面无人艇运动规划研究综述[J]. 舰船科学技术, 2023, 45(16): 1–6.
Gao Xiaopeng, Liu Dongyu, Huo Cong. Research review on motion planning of surface unmanned boat[J]. Ship Science and Technology, 2023, 45(16): 1–6. (in Chinese)
- [2] 郑华荣, 魏 艳, 瞿逢重. 水面无人艇研究现状[J]. 中国造船, 2020, 61(S1): 228–240.
Zheng Huarong, Wei Yan, Qu Fengzhong. Research status of surface unmanned boat[J]. Shipbuilding of China, 2020, 61(S1): 228–240. (in Chinese)
- [3] 汪 咏, 徐海祥, 冯 辉, 等. 考虑船舶操纵性的无人艇路径规划方法[J]. 华中科技大学学报(自然科学版), 2024, 52(3): 91–96.
Wang Yong, Xu Haiyang, Feng Hui, et al. Path planning method of unmanned surface vehicle considering ship maneuverability[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2024, 52(3): 91–96. (in Chinese)
- [4] 徐鹏飞, 丁延旭, 曹清波. 基于环境优化的无人艇全局路径规划研究[J]. 中国造船, 2022, 63(5): 206–220.
Xu Pengfei, Ding Yanxu, Cao Qingbo. Research on global path planning of unmanned surface vehicle based on environmental optimization[J]. Shipbuilding of China, 2022, 63(5): 206–220. (in Chinese)
- [5] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846–894.
- [6] Qi J, Yang H. MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment[J]. IEEE Transactions on Industrial Electronics, 2021, 68(8): 7244–7251.
- [7] 向金林, 王鸿东, 欧阳子路, 等. 基于改进双向 RRT 的无人艇局部路径规划算法研究[J]. 中国造船, 2020, 61(1): 157–166.
Xiang Jinlin, Wang Hongdong, Ouyang Zilu, et al. Research on local path planning algorithm for unmanned craft based on improved bidirectional RRT[J]. Shipbuilding of China, 2020, 61(1): 157–166. (in Chinese)
- [8] Wang Jiankun, Li Baopu, Meng Max Q H. Kinematic constrained bi-directional RRT with efficient branch pruning for robot path planning[J]. Expert Systems with Applications, 2021, 170: 114541.
- [9] 姜兆祯, 王文龙, 孙文祺. 基于改进 RRT* 算法的无人艇路径规划快速求解算法[J]. 系统仿真学报, 2024, 36(4): 888.
Jiang Zhaozhen, Wang Wenlong, Sun Wenqi. Path planning rapid algorithm based on modified RRT* for unmanned surface vessel[J]. Journal of System Simulation, 2024, 36(4): 888. (in Chinese)
- [10] Lin X, Zhao E, Wang X. Unmanned surface vehicle path planning based on improved informed-RRT* algorithm[C]// 2023 IEEE International Conference on Mechatronics and Automation (ICMA). Harbin, China, 2023.
- [11] Wen N, Zhang R, Wu J, et al. Online planning for relative optimal and safe paths for USVs using a dual sampling domain reduction-based RRT* method[J]. International Journal of Machine Learning and Cybernetics, 2020, 11: 2665–2687.
- [12] Viseras A, Losada R O, Merino L. Planning with ants: Efficient path planning with rapidly exploring random trees and ant colony optimization[J]. International Journal of Advanced Robotic Systems, 2016, 13(5): 1729881416664078.
- [13] 刘 挺, 王晓燕, 康智强. 基于蚁群的环境分区目标偏置 RRT 算法路径规划[J]. Journal of Measurement Science and Instrumentation, 2023, 14(1): 55–65.
Liu Ting, Wang Xiaoyan, Kang Zhiqiang. Environmental zoning target bias RRT algorithm path planning based on ant colony[J]. Journal of Measurement Science and Instrumentation, 2023, 14(1): 55–65. (in Chinese)
- [14] Huynh L Q, Tran L V, Phan P N K, et al. Intermediary RRT*-PSO: A multi-directional hybrid fast convergence sampling-based path planning algorithm[J]. Computers, Materials & Continua, 2023, 76(2): 2282–2300.
- [15] Dehghani M, Hubálovský Š, Trojovský P. Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems[J]. IEEE Access, 2021, 9: 162059–162080.
- [16] 刘正锋, 张隆辉, 魏纳新, 等. 限制区域水面无人艇路径规划与跟踪控制研究[J]. 船舶力学, 2021, 25(9): 1127–1136.
Liu Zhengfeng, Zhang Longhui, Wei Naxin, et al. Research on path planning and tracking control of surface unmanned craft in restricted area[J]. Journal of Ship Mechanics, 2021, 25(9): 1127–1136. (in Chinese)
- [17] Nasir J, Islam F, Malik U, et al. RRT*-SMART: A rapid convergence implementation of RRT*[J]. International Journal of Advanced Robotic Systems, 2013, 10(7): 1–12.