



基于改进RRT*算法的移动机器人路径规划

郭利进, 李强

引用本文:

郭利进, 李强. 基于改进RRT*算法的移动机器人路径规划[J]. *智能系统学报*, 2024, 19(5): 1209-1217.

GUO Lijin, LI Qiang. Path planning of mobile robots based on improved RRT* algorithm[J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(5): 1209-1217.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202302010>

您可能感兴趣的其他文章

基于变步长蚁群算法的移动机器人路径规划

Mobile robot path planning based on variable-step ant colony algorithm

智能系统学报. 2021, 16(2): 330-337 <https://dx.doi.org/10.11992/tis.202004011>

融合改进A*算法和Morphin算法的移动机器人动态路径规划

Mobile-robot dynamic path planning based on improved A* and Morphin algorithms

智能系统学报. 2020, 15(3): 546-552 <https://dx.doi.org/10.11992/tis.201812023>

移动机器人全覆盖信度函数路径规划算法

Complete-coverage path planning algorithm of mobile robot based on belief function

智能系统学报. 2018, 13(2): 314-321 <https://dx.doi.org/10.11992/tis.201610006>

基于RGB-D信息的移动机器人SLAM和路径规划方法研究与实现

RGB-D-based SLAM and path planning for mobile robots

智能系统学报. 2018, 13(3): 445-451 <https://dx.doi.org/10.11992/tis.201702005>

基于改进粒子群算法的移动机器人多目标点路径规划

Mobile robot multi-goal path planning using improved particle swarm optimization

智能系统学报. 2017, 12(3): 301-309 <https://dx.doi.org/10.11992/tis.201606046>

多移动机器人的领航-跟随编队避障控制

Piloting-following formation and obstacle avoidance control of multiple mobile robots

智能系统学报. 2017, 12(2): 202-212 <https://dx.doi.org/10.11992/tis.201507029>

DOI: 10.11992/tis.202302010

网络出版地址: <https://link.cnki.net/urlid/23.1538.TP.20240322.1714.006>

基于改进 RRT* 算法的移动机器人路径规划

郭利进, 李强

(天津工业大学控制科学与工程学院, 天津 300387)

摘要: 针对传统快速随机搜索树*(rapidly-exploring random tree*, RRT*) 算法收敛速率较慢, 且不适用于动态场景等问题, 提出一种基于目标点偏置和冗余节点删除的改进 RRT* 算法, 用于解决移动机器人快速找到无碰撞最优路径的问题。此算法在 RRT* 算法基础上, 首先对采样点进行优化处理, 保证路径最优的同时减少搜寻时间; 其次引入路径节点最大值概念, 删除扩展树冗余节点以提高算法效率; 最后结合动态窗口 (dynamic window approaches, DWA) 算法提高路径的安全性和平滑性, 实现对动态障碍物的避障。通过 3 种不同地图下的仿真验证, 改进算法能有效提升路径质量, 且大幅降低运行时间。

关键词: 移动机器人; 动态环境; 最优路径; RRT* 算法; 动态窗口算法; 冗余节点; 安全性; 平滑性

中图分类号: TP242.6 **文献标志码:** A **文章编号:** 1673-4785(2024)05-1209-09

中文引用格式: 郭利进, 李强. 基于改进 RRT* 算法的移动机器人路径规划 [J]. 智能系统学报, 2024, 19(5): 1209-1217.

英文引用格式: GUO Lijin, LI Qiang. Path planning of mobile robots based on improved RRT* algorithm[J]. CAAI transactions on intelligent systems, 2024, 19(5): 1209-1217.

Path planning of mobile robots based on improved RRT* algorithm

GUO Lijin, LI Qiang

(School of Control Science and Engineering, Tiangong University, Tianjin 300387, China)

Abstract: To address the problems associated with the traditional rapidly-exploring random tree* (RRT*) algorithm, including its slow convergence rate and unsuitability for dynamic environments, we propose an improved RRT* algorithm. This new approach biases toward the target point and eliminates redundant nodes to efficiently find collision-free optimal paths for mobile robots. The improved algorithm begins by optimizing the sampling points, which ensures the discovery of the optimal path while reducing search time. Then, it introduces the concept of path maximization to identify and remove redundant nodes from the extended tree, thereby enhancing algorithm efficiency. Additionally, the improved algorithm is integrated with the dynamic window approaches algorithm, which enhances the safety and smoothness of the path and enables effective obstacle avoidance in dynamic environments. Simulations conducted on three different maps demonstrate that the improved algorithm significantly improves path quality and greatly reduces running time.

Keywords: mobile robot; dynamic environment; optimal path; RRT* algorithm; dynamic window approaches algorithm; redundant node; safety; smoothness

在机器人领域, 路径规划一直是研究的核心问题。路径规划指从起点开始到终点规划出一条安全且无碰撞的高效路径, 并且越来越多应用在复杂及人为干涉困难的环境, 这就要求规划出的路径满足路径短、用时少、有效避障等特点^[1-3]。现阶段根据路径规划算法的发展趋势可分为基于搜索的路径规划算法和基于采样的路径规划算法^[4-5]。基于搜索的路径规划算法包括 Dijkstra 算

法、A* 算法等^[6-9]。成怡等^[10]提出一种改进的 A* 算法与 Morphin 搜索树算法相结合的混合算法, 改进关键点的选取方式并删除冗余节点, 利用平均滤波的方法进行路径平滑处理, 再结合 Morphin 搜索树算法进行动态的局部优化。基于采样的路径规划算法主要有概率路线图算法 (probabilistic road maps, PRM)、快速随机搜索树算法 (rapidly-exploring random tree, RRT) 等^[11-14], 此类算法具有灵活强大的搜索能力, 且无需对空间进行建模, 能够较快地获取路径解, 但由于其采样方式随机, 得到的路径并非最优路径。为获得最优

收稿日期: 2023-02-15. 网络出版日期: 2024-03-23.

基金项目: 国家自然科学基金项目 (61973234).

通信作者: 郭利进. E-mail: Doctor_guo@tiangong.edu.cn.

路径, Karaman 等^[15] 提出引入了父节点重选和节点重连的 RRT* 算法, 得到的路径能够收敛到全局最优解; Gammell 等^[16] 提出 Informed-RRT* (下文简称为 In-RRT*) 算法, 规划出一个椭圆作为采样范围, 缩短路径长度的同时加快算法收敛速率, 但随着节点数量的增加, 计算复杂度和运行时间也随之增长。针对此问题, 李娟等^[17] 在未知的三维环境中, 建立搜索实时地图和决策函数以提高搜索效率; Spanogianopoulos 等^[18] 提出一种 RRT*FN (RRT fixed nodes) 算法可以减少路径上的多余叶子节点, 提高规划速度; 张腾龙等^[19] 提出固定节点数的动态双向渐进最优快速随机扩展树算法 (bidirectional RRT* fix-node dynamic, B-RRT* FND), 采用双向贪婪搜索方法来加快路径搜索速度, 利用固定节点算法减少计算量, 并引入动态更新以及路径修复思想, 实现了移动障碍物下的避障功能。

受 RRT* 衍生算法的启发, 本文提出一种改进的 RRT* 算法, 首先利用目标点信息对采样点进行择优选择, 加快算法收敛速度; 其次引入路径节点最大值概念对搜索树的冗余节点进行删除, 降低在后续过程中过多冗余节点造成的额外搜索成本, 拉伸路径的同时提高算法效率; 最后利用获取的全局路径结合动态窗口 (dynamic window approaches, DWA) 算法进行局部规划, 提升路径平滑度和安全性, 实现动态避障。

1 相关工作

1.1 问题描述

定义 $X = R_n$ 为路径规划问题的状态空间, 定义 $X_{obs} \subset X$ 为障碍物所在空间, 可得机器人活动的空间为 $X_{free} = C_X(X \setminus X_{obs})$, 给定机器人初始节点 X_{start} , 目标节点 X_{goal} , 有 $X_{start}, X_{goal} \in X_{free}$ 。定义 $\sigma: [0, 1] \rightarrow R_n$ 为机器人的可行路径, 且 $\sigma(0) = X_{start}, \sigma(1) = X_{goal}$ 。因此路径规划问题转化为在机器人自由活动空间 X_{free} 中求解点集 $\sigma: [0, 1]$, 定义 Σ 表示所有可行路径集合, $c: \Sigma \rightarrow R \geq 0$ 为所有路径代价集合。

对于路径最优解问题, 路径代价和路径优势成反比, 代价越小表示路径越优^[20], 则最优路径求解问题可被描述为: 在自由空间 X_{free} 中求解使路径代价最小的可行路径 σ^* :

$$\sigma^* = \operatorname{argmin} \{c(\sigma) \mid \sigma(0) = X_{start}, \sigma(1) = X_{goal}, \forall x \in [0, 1], \sigma(x) \in X_{free}\} \quad (1)$$

1.2 RRT 算法

RRT 算法是基于采样的路径规划算法中最具代表性的路径规划算法, 该算法的本质是在整个

空间中随机采样并生成路径点, 规划出一条起点到终点的无碰撞且唯一的路径^[21-22]。RRT 算法的具体实现步骤如下^[23]:

- 1) 创建地图空间 S , 起始点 X_{start} , 目标点 X_{goal} 和障碍物。
- 2) 在 X_{free} 中随机取得采样点 X_{rand} 。
- 3) 从与 X_{rand} 最近的节点 X_{near} 以步长 L 向采样点 X_{rand} 方向生长, 并得到新的路径节点 X_{new} , 若碰撞障碍物则不选用。
- 4) 不断重复步骤 2) 和 3) 扩展树枝, 当树生长到 X_{goal} 或 X_{new} 与 X_{goal} 的连线与障碍物不相交时, 找到可行路径 σ 。

1.3 RRT* 算法

RRT 算法具有概率完备性, 其规划速度相对于其他路径规划算法较快, 但由于其随机性, 因此所得路径不是最优路径^[24]。RRT* 算法在 RRT 算法的基础上引进了对路径的寻优思想, 主要有两点改进, 具体实现步骤如下^[25]:

1) 重选父节点

① 计算搜索树总节点数 $\operatorname{len}(n_node)$, 以新节点 X_{new} 为圆心计算范围半径:

$$r_{range} = \sqrt{\lg(\operatorname{len}(n_node)) / \operatorname{len}(n_node)} \quad (2)$$

② 遍历式 (2) 所求圆内节点, 分别计算与 X_{new} 的距离, 记录为新的父节点库, 再检验库中节点与 X_{new} 连线是否碰撞障碍物, 返回无碰撞且连线最短的节点作为新的父节点。

2) 节点重连

以 X_{new} 为父节点, 遍历半径为 r_{range} 的范围内节点, 分别计算与 X_{new} 间的距离并与 X_{new} 当前路径长度相加记为 s_cost , 若小于原路径长度且与障碍物无碰撞, 则此节点为 X_{new} 的子节点。

2 改进 RRT* 算法

2.1 采样点择优

由于传统 RRT* 算法的采样随机性较强, 搜索目标点的过程需花费大量时间, 因此改进算法在保留 RRT* 算法特有的随机采样特点下利用目标点信息对采样点进行择优选择以提升搜索速率^[26]。具体实现步骤如下:

1) 引入启发式思想^[27], 以一定概率将目标点 X_{goal} 选取为采样点:

$$X_{sample} = \begin{cases} X_{goal}, & p \leq a \\ X_{rand}, & p > a \end{cases} \quad (3)$$

式中: X_{sample} 为最终采样点, X_{rand} 为随机采样点, p 为 0~1 的随机数, 为防止失去 RRT 类算法特有

的随机采样特性, a 设置为 0.1。

2) 从离采样点最近的节点 X_{near} 出发, 以机器人步长 L 向采样点扩展, 生成新节点 X_{new} 。

3) 分别计算 X_{near} 和 X_{new} 到 X_{goal} 的欧氏距离, 并进行对比, 若 X_{new} 到 X_{goal} 的距离小于 X_{near} 到 X_{goal} 的距离则认为采样成功, 否则返回步骤 1) 重新采样, 如图 1 所示。

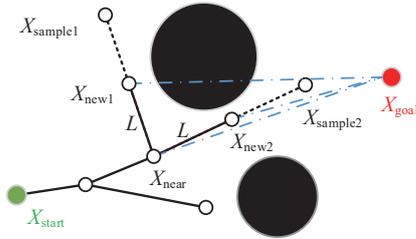


图 1 改进 RRT* 算法采样过程

Fig. 1 Improved RRT* algorithm sampling process

2.2 冗余节点删除

除此之外, RRT* 算法在寻优过程中会产生大量冗余节点, 使得算法的复杂度和运行时间呈指数级增长、寻优效率低下^[28-29]。对此, 引入路径节点最大值概念对冗余节点枝干进行修剪, 减少算法的搜索成本。具体实现步骤如下:

1) 遍历最优路径节点, 分别计算与起终点连线的距离, 取最大值并记录为节点最大值:

$$d_{MAX} = \max \{ \|X_i(X_{start} + t \cdot (X_{goal} - X_{start}))\| \} \quad (4)$$

式中: i 为可行路径上节点的下标, t 为可行路径上 X_i 节点在起、终点连线上的映射点到 X_{start} 的距离与起、终点连线的距离之比。

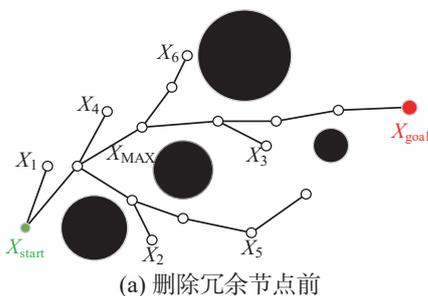
2) 遍历树上所有节点, 随机取出节点计算其与起终点连线的距离为

$$d_{Tj} = \|X_j(X_{start} + t \cdot (X_{goal} - X_{start}))\| \quad (5)$$

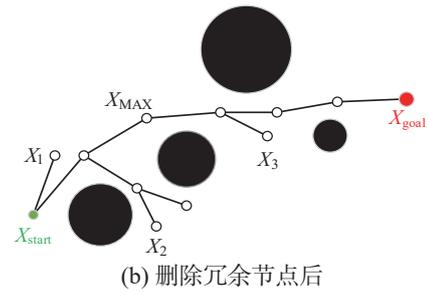
式中 j 为树上其余节点下标。

3) 将式 (5) 中计算出的 d_{Tj} 与式 (4) 的 d_{MAX} 进行对比, 若 $d_{Tj} > d_{MAX}$, 删除此节点及其附属枝干, 反之保留此节点。

4) 不断重复步骤 2) 和 3), 直至所有节点均计算完毕, 如图 2 所示, 其中, X_{MAX} 为最优路径节点垂直于起终点连线的最大值节点。



(a) 删除冗余节点前

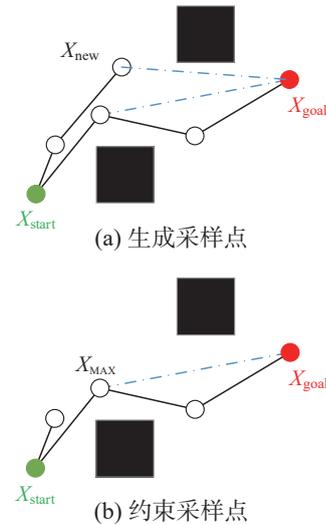


(b) 删除冗余节点后

图 2 删除冗余节点过程示意

Fig. 2 Schematic of the process of removing redundant nodes

剔除冗余节点, 选择关键节点, 从而提升算法运行速率。与此同时, 对采样过程进行二次约束。改进 RRT* 算法的采样过程中, 通过判断新生成的节点与目标点之间的距离来确定是否有效, 生成的新节点离目标点会更近的同时可能会造成路径总长度的增加。而删除冗余节点过程中计算出的 d_{MAX} 同样对采样生成的点有约束作用, 相当于变相拉伸可行路径, 使得路径长度减小, 提高了算法的容错率, 如图 3 所示。



(a) 生成采样点

(b) 约束采样点

图 3 二次约束示意

Fig. 3 Schematic of secondary constraint

2.3 DWA 算法

本文引入 DWA 算法沿全局路径进行局部路径规划。DWA 算法将机器人速度选择问题转化到其线速度 v 和角速度 ω 组成的速度矢量空间 (v, ω) 中, 对于速度矢量空间 (v, ω) 的约束条件主要有 3 点, 具体如下^[30]:

1) 由机器人自身限制的线速度和角速度约束。机器人的采样速度需控制在自身的最大速度和最小速度之间, 约束公式为

$$V_s = \{(v, \omega) | v_{min} \leq v \leq v_{max}, \omega_{min} \leq \omega \leq \omega_{max}\}$$

式中: v 为机器人的线速度, ω 为其角速度。

2) 由电机性能限制的加速度与减速度约束。机器人的加速度会受到电机输出扭矩的限制, 因此线速度 v 和角速度 ω 都存在最大加速度及最大减速度, 约束公式为

$$V_d = \{(v, \omega) | v \in [v_t - \alpha_v \Delta t, v_t + \alpha_v \Delta t], \omega \in [\omega_t - \alpha_\omega \Delta t, \omega_t + \alpha_\omega \Delta t]\}$$

式中: (v_t, ω_t) 分别为当前时刻机器人的线速度和角速度, α_v 为最大线加速度, α_ω 为最大角加速度, Δt 为变化时间间隔。

3) 由环境因素限制的制动距离约束。为防止机器人因为速度原因碰撞到障碍物, 提出满足机器人安全性的速度约束空间, 约束公式为

$$V_a = \{(v, \omega) | v \leq \sqrt{2d_s(v_t, \omega_t)\alpha_v}, \omega \leq \sqrt{2d_s(v_t, \omega_t)\alpha_\omega}\}$$

$$V_m = \{(v, \omega) | v \leq \sqrt{2d_s(v_t, w_t)\alpha_v}, \omega \leq \sqrt{2d_s(v_t, w_t)\alpha_\omega}\}$$

式中: $d_s(v, \omega)$ 为机器人在 t 时刻的速度 (v, ω) 下到静态障碍物的最近距离, $d_m(v, \omega)$ 为机器人在 t 时刻的速度 (v, ω) 下到动态障碍物的最近距离。

因此机器人的速度空间由上述 3 条约束条件共同约束, 最终机器人的速度空间满足:

$$V_r = V_s \cap V_d \cap V_a \cap V_m$$

此外, DWA 算法在上述机器人速度空间 V_r 的约束下进行速度的采样以及离散化, 得到可行的速度集合, 并根据速度集合和轨迹预测时间段生成预测轨迹, 之后通过评价函数对预测的轨迹集进行评价, 选取最优路径, 具体评价函数如下:

$$G(v, \omega) = \{\alpha \cdot \text{head}(v, \omega) + \beta \cdot \text{vel}(v, \omega) + \gamma \cdot d_s(v, \omega) + \eta \cdot d_m(v, \omega)\}$$

式中: $G(v, \omega)$ 为机器人在保证不碰到障碍物的情况下的最大速度; $\text{head}(v, \omega)$ 为方位角评价函数, 用来评估机器人在当前速度下模拟轨迹的末端方向与目标点方向之间的角度差; α 为其权重系数; $\text{vel}(v, \omega)$ 为速度评价函数, 表示机器人当前的运行速度; β 为其权重系数; $d_s(v, \omega)$ 、 $d_m(v, \omega)$ 为距离评价函数, 前者表示机器人位置与最近静态障碍物的关系, 后者表示机器人与最近动态障碍物的关系, 若当前轨迹无障碍物, 记 $d_s(v, \omega)$ 、 $d_m(v, \omega)$ 为常数; γ 、 η 为其各自对应权重系数。

2.4 融合算法实现

融合算法首先利用改进 RRT* 算法提升采样性能, 能够在短时间内规划出一条起点到终点的可行路径, 并且在之后的迭代中删除冗余节点, 减少计算复杂度, 其次结合 DWA 局部路径规划算法实现动态避障, 融合算法具体实现流程如图 4 所示。

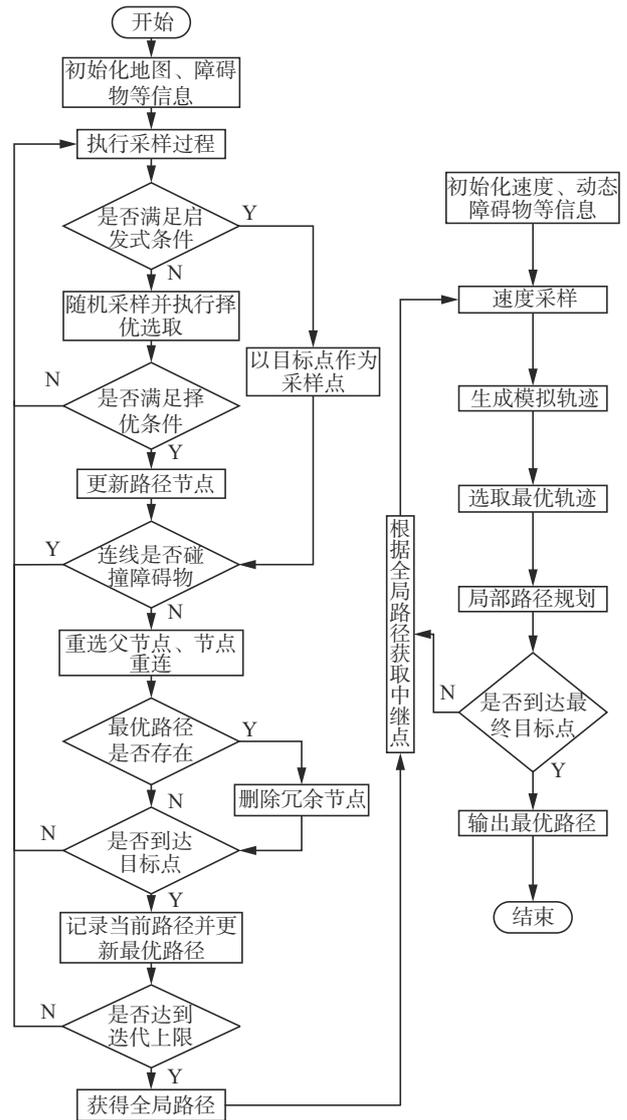


图 4 融合算法流程

Fig. 4 Fusion algorithm flow

3 仿真实验与分析

为验证改进算法的有效性, 将改进 RRT* 算法与其他算法在简单、复杂和存在移动障碍物 3 种地图下进行对比, 并评价其性能指标。本文将不规则障碍物膨胀化为规则图形, 并将机器人看作质点, 为减小 RRT 类算法随机采样带来的不稳定性, 分别对每种地图设置 50 次独立仿真实验, 最终取平均值进行对比分析。本次实验采用处理器为 Intel(R)Core(TM)i5-4460 CPU @ 3.20 GHz、内存为 8 GB 的 64 位 WIN7 操作系统, 实验平台为 Python3.6.8。

3.1 简单地图仿真

简单地图下各算法路径规划结果示意如图 5 所示, 地图大小设置为 20 m × 17 m, 横向和纵向坐标向负扩展 2 m, 移动机器人的起点终点位置为 [0,0] m、[15,12] m, 分别用黄、红圆点表示, 黑色实

心圆表示障碍物, 绿色线表示搜索树枝干, 红色线表示搜索到的最优路径。

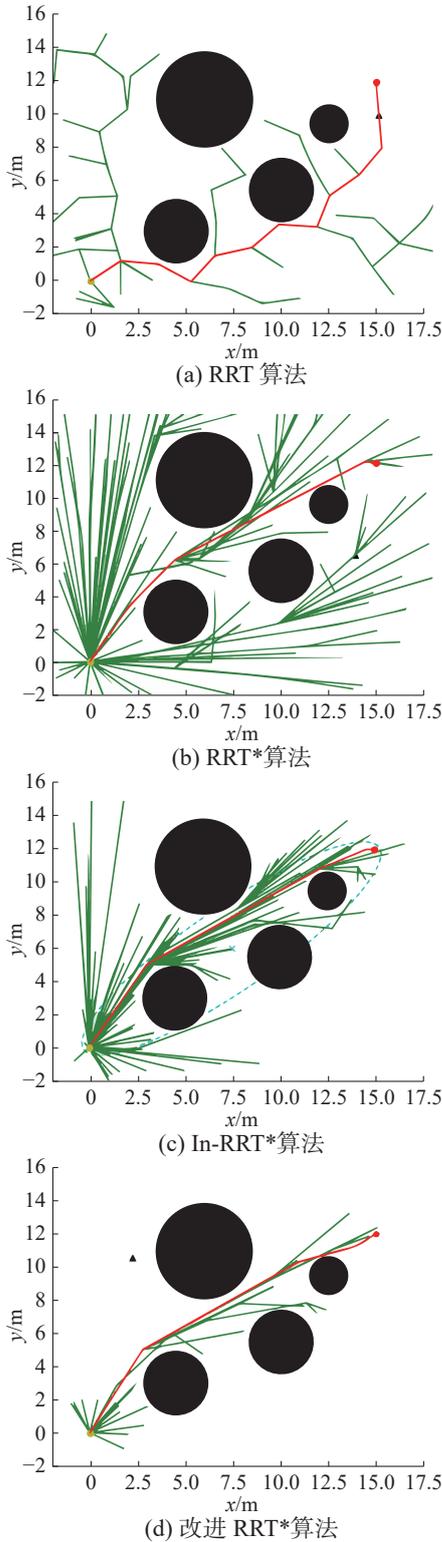


图 5 简单地图下各算法路径规划结果示意

Fig. 5 Schematic of path planning results of algorithms under simple map

分别用 RRT、RRT*、In-RRT*、改进 RRT*算法分别在简单地图中进行路径规划, 迭代次数均为 200, 将 50 次独立仿真实验记录并求得平均值列于表 1。

表 1 简单地图各算法数据

Table 1 Algorithm data of simple map

对比算法	平均路径长度/m	平均运行时间/s	平均迭代次数
RRT	24.398	3.700	85
RRT*	20.192	9.171	82
In-RRT*	19.923	17.861	118
改进RRT*	19.860	5.417	79

从表 1 中可看出, 虽然 RRT 算法运行时间最小, 但其搜索的路径最差, 改进 RRT*算法所求得路径最优, 且平均运行时间和迭代次数小于 RRT*、In-RRT*寻优算法。相比 RRT*算法, 改进 RRT*所的路径长度减小了 1.644%, 时间缩短了 40.933%, 而相比 In-RRT*算法, 改进 RRT*所得路径长度缩短较小, 但其运行时间缩短了 69.671%。由此可得改进 RRT*算法通过对采样点的择优和冗余节点的删除, 有效降低路径长度和计算复杂度的同时提升算法效率。

此外, 将迭代次数提升至 2 000, 简单地图各算法性能对比如图 6 所示, 改进 RRT*算法搜索到的路径优于 RRT*并且与 In-RRT*在迭代 1 600 次以后基本持平(如图 6(a)), 但由于 In-RRT*算法存在大量冗余节点, 运行时间随迭代次数的增加快速增大, 2 000 次迭代后共花费 980.774 s, 而改进 RRT*上升曲线相对平缓, 用时 347.413 s(如图 6(b))。

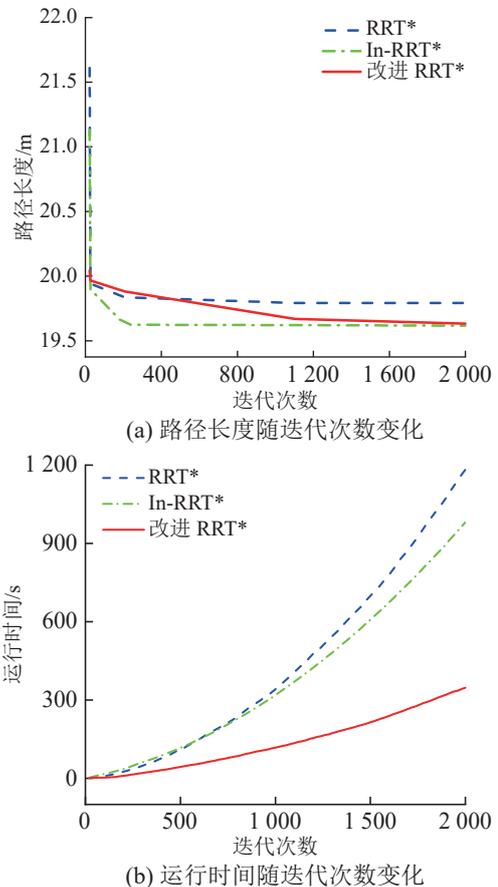


图 6 简单地图各算法性能对比

Fig. 6 Performance comparison of simple map algorithms

3.2 复杂地图仿真

复杂地图下各算法路径规划结果示意如图 7 所示,除障碍物设置外,其余设置如同 3.1 节。从图 7 中可看出,在存在大量障碍物的复杂地图下,RRT 算法获取的路径转折点多,路径长度大,不利于移动机器人的运动,RRT*和 In-RRT*算法获取的路径转折量少,路径也更优,但存在许多不必要的搜索路径,改进 RRT*获得的路径接近上述两类算法,但减少了不必要的搜索路径,即删除了过多冗余节点。分别将各类算法的 50 次独立仿真记录并求得平均值列于表 2。

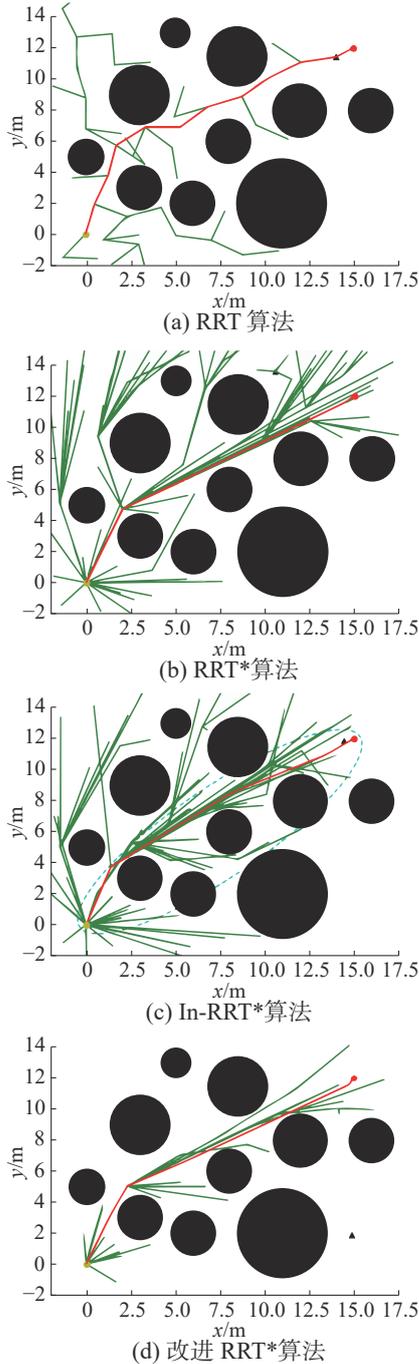


图 7 复杂地图下各算法路径规划结果示意

Fig. 7 Schematic of path planning results of algorithms under complex map

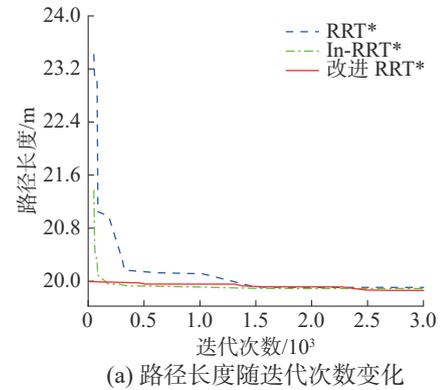
表 2 复杂地图各算法数据

Table 2 Algorithm data of complex map

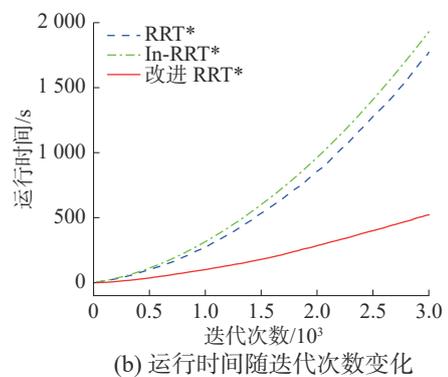
对比算法	平均路径长度/m	平均运行时间/s	平均迭代次数	节点利用率/%
RRT	26.156	4.722	76	37.651
RRT*	20.782	9.087	122	4.467
In-RRT*	20.552	12.861	136	5.686
改进RRT*	20.165	2.954	72	7.051

从表 2 中可看出,RRT 的路径长度远大于其他 3 种算法,因此不作考虑;而 RRT*、In-RRT* 的路径虽短,但由于其冗余节点的存在加大了算法运行时间;改进 RRT* 算法相较 RRT* 算法,获取的路径长度减少 2.969%,运行时间降低 67.492%,相较于 In-RRT* 算法,获取的路径长度减少 1.883%,运行时间降低 77.031%,同时将节点占用率提升到 7.051%,节点利用率是指路径上的节点数目与搜索树上的节点数目之比,节点利用率越大,算法搜寻最优路径的效率越高。

将迭代次数提升至 3 000,复杂地图各算法性能对比如图 8 所示,由于采样点在多障碍物的情况下难以选取,导致 RRT* 和 In-RRT* 算法的路径搜索效率降低,花费大量的运行时间,而改进 RRT* 算法最优路径收敛较快,且迭代次数到达 3 000 时的运行时间为 522.019 s,远小于其他两种算法。



(a) 路径长度随迭代次数变化



(b) 运行时间随迭代次数变化

图 8 复杂地图各算法性能对比

Fig. 8 Performance comparison of algorithms for complex maps

3.3 动态地图仿真

为进一步探究改进 RRT* 算法在动态环境下

的性能, 本文设置了含有动态障碍物、大小为 $20\text{ m} \times 17\text{ m}$ 的动态地图, 引入 DWA 局部路径规划算法, 将移动机器人的最大线速度、最大角速度、最大线加速度、最大角加速度、线速度分辨率、角速度分辨率、时间间隔、模拟轨迹持续时间分别为 3 m/s 、 $50\text{ (}^\circ\text{)/s}$ 、 0.5 m/s^2 、 $30\text{ (}^\circ\text{)/s}^2$ 、 0.01 m/s 、 $0.1\text{ (}^\circ\text{)/s}$ 、 0.1 s 、 4 s 。动态障碍物地图下各算法路径规划结果示意如图 9 所示。

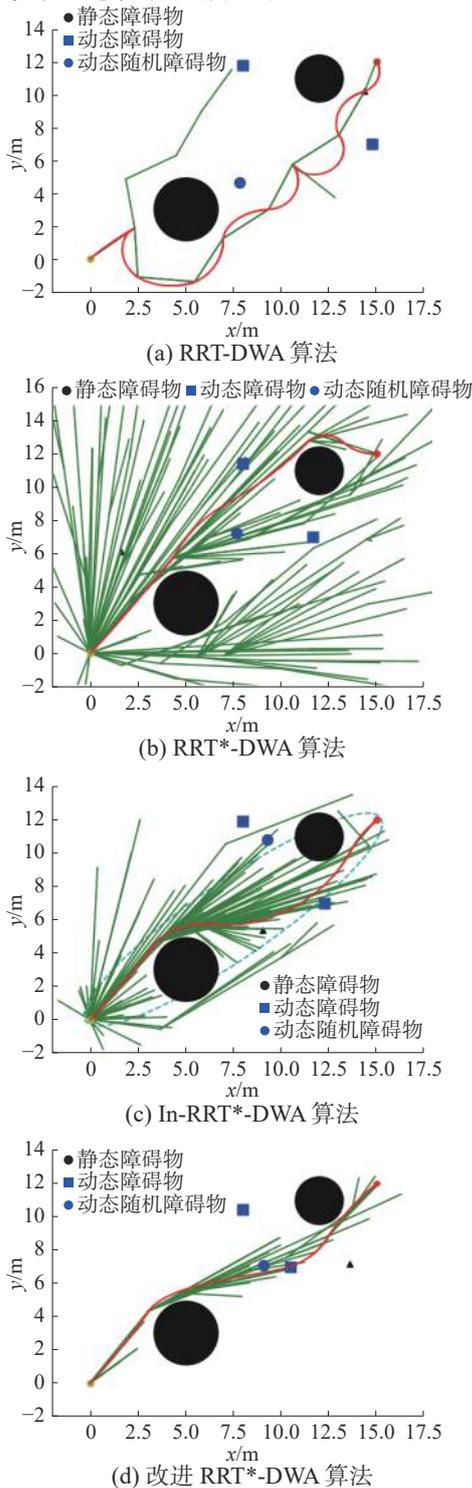


图 9 动态地图下各算法路径规划结果示意

Fig. 9 Schematic of path planning results of each algorithm under dynamic map

从图 9 中可看出, RRT-DWA 算法因路径节点过多而产生较大转折, 其他 3 类算法规划出的路径较为平滑, 将 50 次独立仿真记录并求得平均值列于表 3。

表 3 动态地图各算法数据
Table 3 Algorithm data of dynamic map

对比算法	平均路径长度/m	平均运行时间/s
RRT-DWA	29.344	717.503
RRT*-DWA	21.406	367.994
In-RRT*-DWA	20.809	402.117
改进RRT*-DWA	20.414	294.721

从表 3 中可看出, RRT-DWA 算法引入局部规划算法后得到的路径长度最大, 运行时间也最久; 其他 3 类算法规划出的路径都接近最优路径, 其中利用改进 RRT*-DWA 算法得到的路径平均长度比 RRT*-DWA 算法、In-RRT*-DWA 算法分别减少 4.634 %、1.898 %, 平均运行时间与之对比分别降低 19.911 %、26.708 %, 由此可得, 本文算法在动态环境下具有较好的动态规划能力。

4 结束语

针对传统 RRT*算法收敛速度慢、无法应用于动态环境等问题, 提出了一种改进 RRT*算法结合 DWA 算法的路径规划方法。该算法首先对随机树的采样点进行择优选取, 避免无效搜寻; 其次对树上冗余节点进行删除, 提高算法运算速率, 同时对采样点进行二次择优; 最后引入 DWA 算法, 实现动态环境下的避障功能。通过仿真实验与各搜索算法对比, 验证所提融合算法在二维平面路径规划中能够有效提升路径质量, 大幅减少算法运行时间, 并提升在动态障碍物环境下的路径规划性能。

参考文献:

- [1] GUL F, MIR I, ABUALIGAH L, et al. A consolidated review of path planning and optimization techniques: technical perspectives and future directions[J]. *Electronics*, 2021, 10(18): 2250.
- [2] WU Xiaojing, XU Lei, ZHEN Ran, et al. Biased sampling potentially guided intelligent bidirectional RRT algorithm for UAV path planning in 3D environment[J]. *Mathematical problems in engineering*, 2019, 2019: 5157403.
- [3] LYU Desheng, CHEN Ziwei, CAI Zesu, et al. Robot path planning by leveraging the graph-encoded Floyd algorithm[J]. *Future generation computer systems*, 2021, 122: 204–208.

- [4] 张子迎, 宫思远, 徐东, 等. 多机器人任务分配与路径规划算法 [J]. 哈尔滨工程大学学报, 2019, 40(10): 1753–1759.
ZHANG Ziying, GONG Siyuan, XU Dong, et al. Research on multi-robot task assignment and path planning algorithm[J]. Journal of Harbin Engineering University, 2019, 40(10): 1753–1759.
- [5] 崔永杰, 王寅初, 何智, 等. 基于改进 RRT 算法的猕猴桃采摘机器人全局路径规划 [J]. 农业机械学报, 2022, 53(6): 151–158.
CUI Yongjie, WANG Yinchu, HE Zhi, et al. Global path planning of kiwifruit harvesting robot based on improved RRT algorithm[J]. Transactions of the Chinese society for agricultural machinery, 2022, 53(6): 151–158.
- [6] 张飞, 白伟, 乔耀华, 等. 基于改进 D* 算法的无人机室内路径规划 [J]. 智能系统学报, 2019, 14(4): 662–669.
ZHANG Fei, BAI Wei, QIAO Yaohua, et al. UAV indoor path planning based on improved D* algorithm[J]. CAAI transactions on intelligent systems, 2019, 14(4): 662–669.
- [7] 李颀, 汪伟. 多全向轮协同分拣平台的路径规划 [J]. 系统仿真学报, 2021, 33(3): 698–709.
LI Qi, WANG Wei. Path designing of multi-omnidirectional wheel collaborative sorting platform[J]. Journal of system simulation, 2021, 33(3): 698–709.
- [8] 黄鑫, 董红斌. 面向无人机路径规划的双层多目标粒子群算法 [J]. 应用科技, 2023, 50(3): 1–10.
HUANG Xin, DONG Hongbin. Double-layer multi-objective particle swarm algorithm for UAV path planning [J]. Applied science and technology, 2023, 50(3): 1–10.
- [9] 申铨京, 施英杰, 黄永平, 等. 基于双向蚁群算法的路径规划研究 [J]. 哈尔滨工程大学学报, 2023, 44(5): 865–875.
SHEN Xuanjing, SHI Yingjie, HUANG Yongping, et al. Path planning optimization using the bidirectional ant colony algorithm[J]. Journal of Harbin Engineering University, 2023, 44(5): 865–875.
- [10] 成怡, 肖宏图. 融合改进 A* 算法和 Morphin 算法的机器人动态路径规划 [J]. 智能系统学报, 2020, 15(3): 546–552.
CHENG Yi, XIAO Hongtu. Mobile-robot dynamic path planning based on improved A* and Morphin algorithms [J]. CAAI transactions on intelligent systems, 2020, 15(3): 546–552.
- [11] 邹宇星, 李立君, 高自成. 基于改进 PRM 的采摘机器人机械臂避障路径规划 [J]. 传感器与微系统, 2019, 38(1): 52–56.
ZOU Yuxing, LI Lijun, GAO Zicheng. Obstacle avoidance path planning for harvesting robot arm based on improved PRM[J]. Transducer and microsystem technologies, 2019, 38(1): 52–56.
- [12] WANG Hao, LI Guoqing, HOU Jie, et al. A path planning method for underground intelligent vehicles based on an improved RRT* algorithm[J]. *Electronics*, 2022, 11(3): 294.
- [13] 许万, 杨晔, 余磊涛, 等. 一种基于改进 RRT* 的全局路径规划算法 [J]. 控制与决策, 2022, 37(4): 829–838.
XU Wan, YANG Ye, YU Leitao, et al. A global path planning algorithm based on improved RRT*[J]. Control and decision, 2022, 37(4): 829–838.
- [14] 夏炎, 隋岩. PRM 路径规划算法优化研究 [J]. 应用科技, 2010, 37(10): 1–5.
XIA Yan, SUI Yan. Research on optimization of PRM path planning algorithm[J]. Applied science and technology, 2010, 37(10): 1–5.
- [15] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. *International journal of robotics research*, 2011, 30(7): 846–894.
- [16] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago: IEEE, 2014: 2997–3004.
- [17] 李娟, 张韵, 陈涛. 改进 RRT 算法在未知三维环境下 AUV 目标搜索中的应用 [J]. 智能系统学报, 2022, 17(2): 368–375.
LI Juan, ZHANG Yun, CHEN Tao. Application of the improved RRT algorithm to AUV target search in an unknown 3D environment[J]. CAAI transactions on intelligent systems, 2022, 17(2): 368–375.
- [18] SPANOGLANOPOULOS S, SIRLANTZIS K. Non-holonomic path planning of car-like robot using RRT*FN[C]//2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence. Goyangi: IEEE, 2015: 53–57.
- [19] 张腾龙, 李擎. 基于 B-RRT*FND 算法的机器人路径规划 [J]. 控制与决策, 2023, 38(11): 3121–3127.
ZHANG Tenglong, LI Qing. Path planning of AGV based on B-RRT*FND algorithm[J]. Control and decision, 2023, 38(11): 3121–3127.
- [20] QIN Hongwei, SHAO Shiliang, WANG Ting, et al. Review of autonomous path planning algorithms for mobile robots[J]. *Drones*, 2023, 7(3): 211.
- [21] PEREIRA N, RIBEIRO A F, LOPES G, et al. Path planning towards non-compulsory multiple targets using TWIN-RRT[J]. *Industrial robot*, 2016, 43(4): 370–379.
- [22] 刘成菊, 韩俊强, 安康. 基于改进 RRT 算法的 RoboCup 机器人动态路径规划 [J]. 机器人, 2017, 39(1):

- 8–15.
LIU Chengju, HAN Junqiang, AN Kang. Dynamic path planning based on an improved RRT algorithm for RoboCup robot[J]. *Robot*, 2017, 39(1): 8–15.
- [23] CHAI Qisen, WANG Yujun. RJ-RRT: improved RRT for path planning in narrow passages[J]. *Applied sciences*, 2022, 12(23): 12033.
- [24] WANG Jiankun, CHI Wenzheng, LI Chenming, et al. Neural RRT*: learning-based optimal path planning[J]. *IEEE transactions on automation science and engineering*, 2020, 17(4): 1748–1758.
- [25] ZHOU Lintao, WU Nanpeng, CHEN Hu, et al. RRT*-fuzzy dynamic window approach (RRT*-FDWA) for collision-free path planning[J]. *Applied sciences*, 2023, 13(9): 5234.
- [26] DAM T, CHALVATZAKI G, PETERS J, et al. Monte-Carlo robot path planning[J]. *IEEE robotics and automation letters*, 2022, 7(4): 11213–11220.
- [27] FRANSEN K, EEKELEN J V. Efficient path planning for automated guided vehicles using A* (Astar) algorithm incorporating turning costs in search heuristic[J]. *International journal of production research*, 2021, 61: 707–725.
- [28] MENG B H, GODAGE I S, KANJ I. RRT*-based path planning for continuum arms[J]. *IEEE robotics and automation letters*, 2022, 7(3): 6830–6837.
- [29] DING Jun, ZHOU Yinxuan, HUANG Xia, et al. An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling[J]. *Journal of computational science*, 2023, 67: 101937.
- [30] 蒲兴成, 谭令. 基于自适应动态窗口改进细菌算法与移动机器人路径规划 [J]. *智能系统学报*, 2023, 18(2): 314–324.
PU Xingcheng, TAN Ling. A mobile robot path planning method based on adaptive DWA and an improved bacteria algorithm[J]. *CAAI transactions on intelligent systems*, 2023, 18(2): 314–324.

作者简介:



郭利进, 教授, 博士, 主要研究方向为机器人信号获取与处理、移动机器人导航与控制。主持和参与市级以上科研项目 20 余项, 发表学术论文 60 余篇。E-mail: Doctor_guo@tian-gong.edu.cn。



李强, 硕士研究生, 主要研究方向为动态环境下移动机器人路径规划。E-mail: winkli_tgu04@163.com。

[责任编辑: 刘冰洁]