華東理工大學学报 自然科学版 Natural Science Edition
JOURNAL OF EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEB-RRT:基于动态椭球体采样的改进RRT算法

邹宇辰 王学武 高永亮

DEB-RRT:An Improved RRT Algorithm Based on Dynamic Ellipsoid Sampling

ZOU Yuchen, WANG Xuewu, GAO Yongliang

在线阅读 View online: http://doi.org/10.14135/j.cnki.1006-3080.20260203001

您可能感兴趣的其他文章

Articles you may be interested in

多工位、多机器人焊接任务分配与路径规划

Multi-Station Multi-Robot Welding Task Assignment and Path Planning

华东理工大学学报（自然科学版）. 2025, 51(5): 633-644 http://doi.org/10.14135/j.cnki.1006-3080.20241225001

基于立方Farrow结构的插值细搜频偏估计算法

Frequency Offset Estimation Algorithm of Interpolation Detail Search Based on Cubic Farrow Structure

华东理工大学学报（自然科学版）. 2024, 50(1): 121-128 http://doi.org/10.14135/j.cnki.1006-3080.20221109001

基于多尺度WideResNet的铁轨缺陷小样本检测算法

A Few-Shot Rail Defect Detection Algorithm Based on Multi-Scale WideResNet Model

华东理工大学学报（自然科学版）. 2025, 51(6): 843-849 http://doi.org/10.14135/j.cnki.1006-3080.20250214001

基于贪婪策略的低复杂度功率分配算法

A Low Computational Complexity Power Allocation Algorithm Based on Greedy Policy

华东理工大学学报（自然科学版）. 2021, 47(3): 340-347 http://doi.org/10.14135/j.cnki.1006-3080.20200119002

基于Pareto支配的两阶段多目标优化算法

Two Stage Multi-Objective Optimization Algorithm Based on Pareto Dominance

华东理工大学学报（自然科学版）. 2022, 48(6): 806-815 http://doi.org/10.14135/j.cnki.1006-3080.20210530001

基于网格拥挤度的自适应参考点多目标优化算法

An Adaptive Multi-Objective Optimization Algorithm with Reference Point Based on Grid Congestion Degree

华东理工大学学报（自然科学版）. 2025, 51(4): 522-537 http://doi.org/10.14135/j.cnki.1006-3080.20241007001

扫码关注公众号，获取更多信息！

文章编号：1006-3080(2026)05-0695-10 DOI：10.14135/j.cnki.1006-3080.20260203001

DEB-RRT:基于动态椭球体采样的改进 RRT 算法

邹宇辰， 王学武， 高永亮
（华东理工大学能源化工过程智能制造教育部重点实验室, 上海 200237）

摘要: 针对双向快速扩展随机树算法采样点随机性大、采样点利用率低、算法收敛速度缓慢等问题，本文基于二维和三维环境提出了改进的 DEB-RRT(Dynamic Ellipsoid Bidirectional-Rapidly-Exploring Random Tree)算法。在保证概率完备性前提下，本文算法依据碰撞检测的失败概率动态控制随机采样点的目标导向性，在获得可行路径解后对其进行路径二次优化处理，使得最终路径更短、转角处更平滑。在 MATLAB 平台上进行算法仿真，结果表明相较于其他传统的路径规划算法，该算法路径搜索时间更短，所得路径长度更短，更具有可行性。

关键词: 路径规划；自适应采样；曲线优化处理；贪心算法；双向 RRT 算法

中图分类号: TP301 **文献标志码:** A

RRT(Rapidly-Exploring Random Tree)算法作为一种基于采样的路径规划算法，具有快速性和随机性优势，其自身具有算法架构简单无需建模、可拓展性强、方便与其他算法相结合以及概率完备性等优点，在路径规划问题中得到了广泛的关注与研究。然而 RRT 算法在路径探索过程中存在的目标导向性差、收敛速度缓慢、生成的冗余节点较多、在狭窄环境下搜索效率差等缺点。

为了解决 RRT 算法在路径规划中随机搜索性大的缺点，Lavalle 等[1] 首次在典型 RRT 算法采样点的选取上加入了概率为 P 的引导值，显著提高了随机树的生成速度，但是由于 P 值不稳定，导致其时间成本增加。Kuffner等[2] 提出了双向快速扩展随机树算法(Bidirectional RRT, BI-RRT)，构造两个分别以起始点和目标点为原点的扩展树(分别以对方为目标点进行扩展)，使得扩展效率得到提升，但同时也存在目标导向性差的问题。Wang 等[3] 提出利用神经网络预测最优路径区域来指导采样，提高采样效率；Zhang 等[4] 提出了基于目标导向采样策略的概率优化 P-RRT*算法，提升了基础 RRT*算法的收敛速度以及路径质量；Li 等[5] 提出了 Q-RRT*算法，有效解决了收敛速度问题，优化了路径代价。Wang 等[6] 用非均匀采样引导 RRT*算法并与人工势场法相结合，同时加入深度学习机制，使该算法的综合性能得到提升。Gammell 等[7] 提出了一种基于启发式引导的采样改进方案，对最优路径的存在空间进一步缩小，使得算法在高纬度采样空间中的采样效率得到提高。Wang 等[8] 为解决具有凹、凸、窄通道、迷宫和多障碍物等复杂环境下的路径规划问题，提出了一种自适应扩展双向 RRT* (AEB-RRT*) 算法，结果表明 AEB-RRT*算法在搜索能力上优于其他路径规划算法，且成功应用于六自由度弧焊机器人的路径规划问题。改进算法的设计思路也借鉴了上述基于自适应调整采样概率的思想提升了路径搜索过程的效率。

综上，现有对 RRT 算法的改进工作，主要围绕该算法本身存在的采样点随机性强、采样点利用率低以及收敛速度缓慢等核心问题展开。但目前多数基于固定概率的目标偏置改进策略，在应用于复杂地图环境时，仍存在易陷入局部最优无法跳脱，甚至无法搜索到可行路径的局限性。此外，在路径二次

收稿日期: 2026-02-03
基金项目: 国家自然科学基金项目(62076095)
作者简介: 邹宇辰(2001—)，男，上海人，硕士生，主要研究方向为工业机器人优化与调度。E-mail: y20250066@mail.ecust.edu.cn
通信联系人: 王学武, E-mail: wangxuew@ecust.edu.cn
引用本文: 邹宇辰, 王学武, 高永亮. DEB-RRT:基于动态椭球体采样的改进 RRT 算法 [J]. 华东理工大学学报(自然科学版), 2026, 52(5): 695-704.
Citation: ZOU Yuchen, WANG Xuewu, GAO Yongliang. DEB-RRT:An Improved RRT Algorithm Based on Dynamic Ellipsoid Sampling[J]. Journal of East China University of Science and Technology, 2026, 52(5): 695-704.

优化环节,如何同时保障路径平滑性与采样节点分布合理性,仍是当前该领域需要进一步探索的主要研究方向。

本文改进了算法易陷入局部最优的问题,主要贡献包括:基于BI-RRT算法优化采样偏置区间,提出动态椭球体采样,依据采样失败概率动态调整范围,提升收敛速度;基于贪心策略优化路径节点以缩短路径,并采用三阶B样条平滑路径转角;在狭长通道障碍物环境下开展对比实验,验证改进算法在路径长度、运行效率等指标上优于传统算法。

1 路径规划问题描述

对于一个给定的障碍物空间 $X \subset Rd$,其中 d 表示障碍物空间的维度,R 表示采样空间,该障碍物空间可以被进一步细分为:障碍空间 $X_{\text{obs}} \subset X$ 和自由空间 $X_{\text{free}} = X/X_{\text{obs}}$,本文中所采用的距离计算方式为Euclidean距离(欧氏距离),可以表示为EucDis(X_1, X_2),其中 X_1 和 X_2 分别表示自由空间中的两个节点($X_1 \in X_{\text{free}}$ 和 $X_2 \in X_{\text{free}}$),设起始点和目标点分别为 X_{init} 和 X_{goal}。由于基于采样的路径规划算法最终所达到的目标点与给定的目标点可能无法做到完全相同,因此为了确保能够找到可行的路径解,需要设置距离阈值DisTh。当采样节点 X_n 与目标点之间的欧氏距离小于距离阈值时,即 $\text{EucDis}(X_n, X_{\text{goal}}) < \text{DisTh}$ 时,视为找到可行路径解,通常情况下DisTh与节点扩展的步长 Δq 相等。而可行路径解则可以定义为:在自由空间 X_{free} 中存在节点集合 $\sigma f(1:n) \subset X_{\text{free}}$,其中 $\sigma f(1) = X_{\text{init}}$,$\sigma f(n) = X_{\text{goal}}$,任取 $X_i \in \sigma f$,均有与 X_i 相连的边为无碰撞路径。RRT算法中生长树被定义为 $T = (V, E) \subset X$,其中 V 和 E 分别表示所有节点和节点连接边的集合。

由此,路径规划问题可以被概述为:在障碍物空间中寻找一条包含起始点 X_{init} 和目标点 X_{goal} 的无碰撞最优路径解 $\sigma f \subset X_{\text{free}}$,其中 $\sigma f(1) = X_{\text{init}}$,$\sigma f(n) = X_{\text{goal}}$,最优路径即所搜索到的路径 σf 的欧氏距离最短。若在有限时间内无法得到可行路径解或算法迭代次数超过设定值,则路径解输出为空,即无法找到可行路径解。

由于在解决上述路径规划问题时,BI-RRT算法相较于单个树扩展具有更快的扩展速率,因此本文基于BI-RRT算法,重新设计针对采样失败概率而动态变化的目标偏置采样空间,进而改善RRT算法中存在的目标导向性差、收敛速度缓慢等问题。同时受贪心思想启发,对搜索得到的路径进行二次优化处理,使其在实际搜索过程中转角幅度和采样点间隔合理,防止实际过程产生跳步的风险。

2 RRT算法改进

2.1 二维环境自适应节点采样算法

基于双向扩展策略,改进算法在设置导向点时加入了自适应的思想,且并非固定地将各个生长树的目标点作为随机采样点,在地图初始化过程的同时,对概率导向点进行初始化,具体操作如下:对于2D环境路径规划问题,取起始点和目标点之间的中垂线作为导向性辅助线,以其中点为起点向两侧对称平均分为若干份作为偏置点。

在算法完成初始化阶段后,开始进行双树随机探索过程,并同步进行自适应计算,在双树各自的扩展过程中分别计算近20次扩展失败的概率,具体公式如式(1)所示。

$$p_a = (p_1 + p_2 + p_3 + \cdots + p_{20})/20 \tag{1}$$

其中,p_1~p_{20} 为近20次扩展的失败率,若本次扩展成功则记为0;扩展失败则记为1,所以 p_a 是一个取值范围在0~1之间的值。在得到失败概率 p_a 后,为了辨别当前扩展步骤中是否采用自适应节点进行扩展,还需要设置一个失败概率 p_a 的阈值 p_n,若 $p_a < p_n$,则采用自适应采样节点策略;若 $p_a > p_n$,则表示当前扩展步骤中最近的失败概率过高,不适合再额外添加辅助方向的偏置,因此需要采用传统的随机节点采样。

需要指出的是,采样失败概率的阈值 p_n 对于算法的运行时间性能有显著的影响,若 p_n 设置过小,算法可能会过快进入随机探索过程,进而降低算法运行效率和延长算法的运行时间;若 p_n 设置过大,则算法目标偏置效果明显,但算法也可能因在复杂地图中极易陷入局部最优解而无法跳脱。针对简单环境,p_n 值可设置在0.6~0.8之间,而复杂环境中 p_n 尽可能不要超过0.5。针对本文后续所用地图测试用例图算法,在运行时 p_n 取值为0.5。

自适应采样策略公式如式(2)所示。

$$\text{samplepoint} = \text{centerpoint} \pm \boldsymbol{x} p_a/p_n \tag{2}$$

其中,samplepoint为所求的自适应随机采样点,centerpoint为目标点与起始点的中点,也是之前所求中垂线的中点,$\boldsymbol{x}$ 为从中点到中垂线一端的单位向量,在自适应采样的环节中,p_a 小于 p_n,又由于 p_a 是由近20次的失败概率计算而来,所以最终的 p_a/p_n 值是一个在0~1之间且间隔为0.1的数,正好对应初始化阶段所均分的10个点,平均分布在中垂线上,而正

负号则代表随机从中点向中垂线端点的两侧取点。基于正负号决定的中垂线两侧取点过程完全是随机选取过程, 因此在公式中正负号的选取概率均为 50%。

改进的 BI-RRT 算法流程图如图 1 所示。

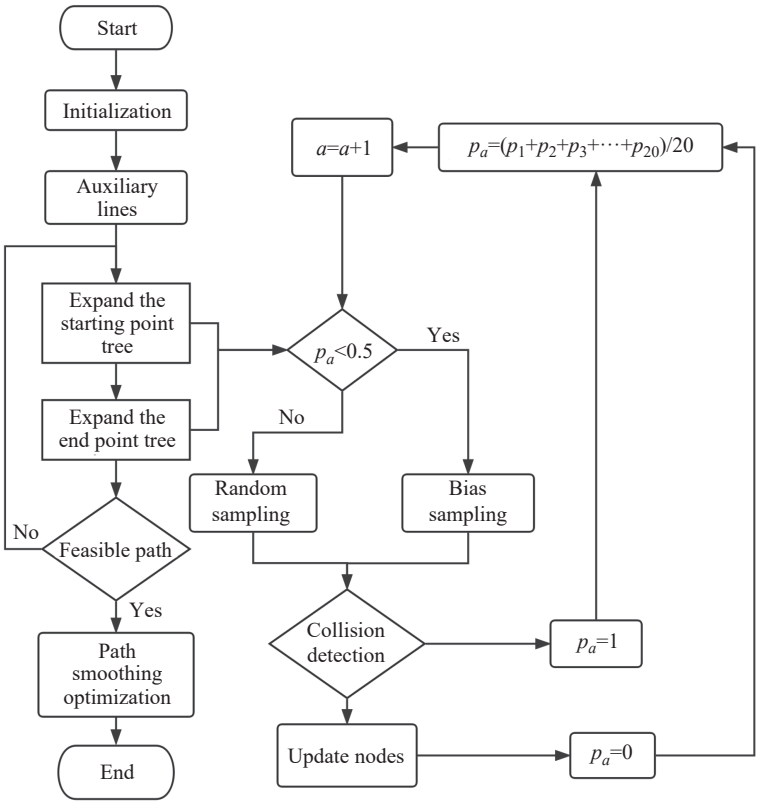


图 1　改进算法流程图

Fig. 1　Improved algorithm flow chart

以单棵树的扩展过程为例, 改进自适应采样节点策略的扩展如图 2 所示, 其中蓝色点为目标点, 0 号绿色节点为起始点, 红色点为中点, 红色线即为根据起始点和目标点的位置所求得的目标导向辅助线。

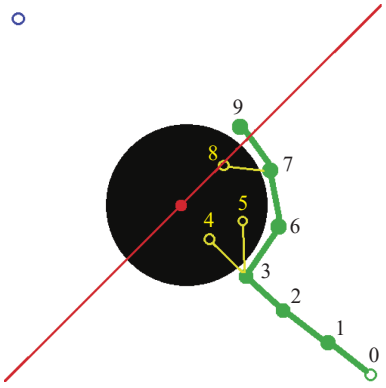


图 2　自适应采样节点扩展示意图

Fig. 2　Adaptive sampling node expansion

在节点 1~3 的扩展过程中, 由于初期过程中没有节点出现扩展失败, 所以 $p_a = 0$; 又由式(2)计算当前采样点位置可以得出, 在节点 1~3 的扩展过程中, 都是以红色点为扩展方向进行扩展的, 结合双向扩展的思想, 单树向中垂线中点进行扩展也可以等效为向目标点 X_{goal} 进行扩展, 由此确保了该改进算法具有一定的目标导向性。

在节点 3 向下一节点扩展的过程中, 受自适应采样策略的影响, 首先会考虑向红色点方向发起扩展, 从而产生了节点 4, 但由于碰撞检测不通过, 节点扩展失败后进行下一轮扩展尝试。再由式(2)获得新一轮扩展的节点方向, 即向红色点两侧开始偏移产生节点 5, 若仍碰撞失败则持续循环。

在保证 $p_a < p_n$ 的前提下, 通过自适应采样找到可行的扩展节点(即节点 6)并进行一次扩展。若在此过程中 $p_a > p_n$, 由式(2)可以得出, 在 p_a 增大时采样节点逐渐向辅助线两端延伸, 若该过程不能找到无碰撞节点, 则在此过程中自适应采样策略失效, 转换为传统的随机采样点。

在节点 6 扩展完成后, 由式(1)可得采样 p_a 会相对减小, 因此相较于节点 3~6 的扩展, 节点 6~7 的扩展方向会更靠近红色中心点。在节点 7 成功扩展后, 随着 p_a 进一步减小, 下一次扩展方向则为更靠近红色点的节点 8, 但由于碰撞检测失败, 最终向节点

9 进行扩展,自适应算法以此类推伴随该算法节点扩展过程同步进行。

2.2 三维环境自适应椭球采样算法

在三维空间中,起点和终点的连线从一条中垂线变为一个中垂面,若以整个中垂面作为目标偏置,则搜索面过于狭窄容易陷入局部最优。因此基于三维障碍物环境,在 2.1 节的自适应采样策略上进行改进,将目标偏置区间改为一个以起始点和目标点为长轴、起始点和目标点的中点作为圆心的动态椭圆体。

设起始点和目标点之间的距离为 c,则动态椭球体的长轴 a 可由式(3)表示:

$$a \geqslant c/2 \tag{3}$$

与改进的二维采样策略类似,算法在初始阶段进行扩展时,采样失败概率为 0,扩展方向为目标点,此时的椭球体短轴 b =0、$a = c/2$,即采样空间暂时性呈一条直线。当采样失败概率不为 0 时,短轴 b 的值上升,此时长轴 $a > c/2$ 。

由于扩展策略在初始阶段以目标点作为直接目标偏置,此时椭球体的两个短轴 b 均为 0,即目标偏置在起点和终点的连线。随着扩展进行,采样的失败概率上升,动态椭球体的短轴 b 根据式(4)进行动态变化,其中 $b_{\max}$ 为三维障碍物环境中短轴的最大值。

$$b = p_a \times b_{\max} \tag{4}$$

改进算法的流程图如图 3 所示。其中动态椭球体探索环节分为 3 种不同的随机采样模式:

(1)在采样过程起始阶段时,p_a=0,此时可得当前的椭球体短半轴为 0,采样点全部在起始点和目标点的连线上,即将目标点视作偏置方向。此时的目标偏置区间如图 4所示,其中彩色区域为目标偏置区间,起始点和目标点连线的中点为(0, 0, 0)。

(2)随着 p_a 增加,动态椭球体的短半轴 b 变长,以起始点和目标点连线的中点为圆心的椭球体逐渐变大,目标偏置区域逐渐从中心点附近向整个中垂面扩散进而实现更大范围的探索,此时的目标偏置区间如图 5 所示。

当 p_a 过大,即动态椭球体的短轴半径为 $b_{\max}$,此时动态椭球体策略失效,采用传统的随机采样策略帮助脱离当前探索区域。该采样策略的采样范围从窄到宽,能够兼顾探索前期的快速收敛与后期的全局完备性。本文结合 BIRRT 算法的优势,在从起始点和目标点双向扩展树时,采用相互独立的椭圆短轴半径,解决了单一尺度参数难以适配起点、终点两端障碍物环境差异的难题,更适用于复杂障碍物环

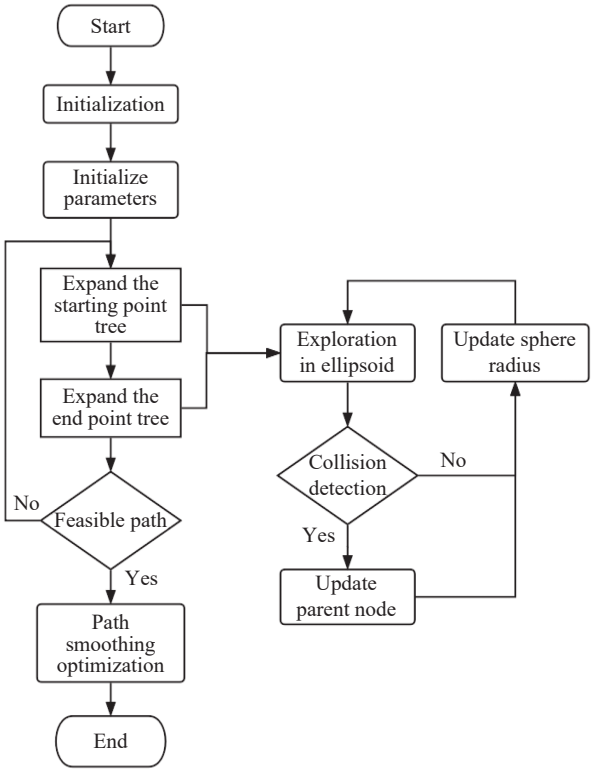


图 3 三维障碍物环境改进算法流程图

Fig. 3 3D Obstacle environment improved algorithm flow chart

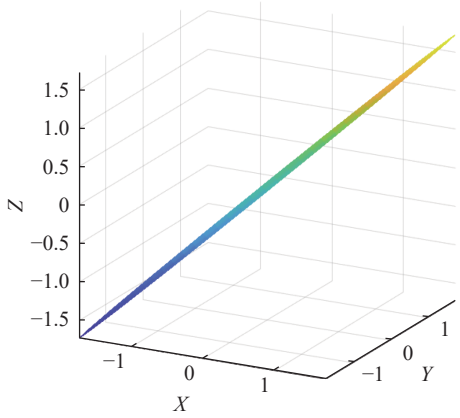


图 4 初始阶段目标偏置区间

Fig. 4 Initial phase target bias interval

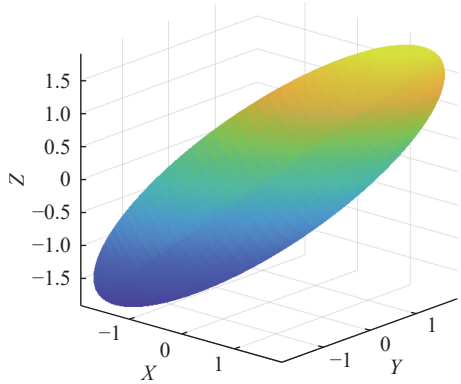


图 5 失败概率上升后目标偏置区间

Fig. 5 Target bias interval with increased failure probability

境。同时, 动态椭球体采样策略利用采样失败概率动态调整目标偏置区域, 实现对探索收敛速度与过程鲁棒性的同步优化。

3　路径二次优化策略

3.1　路径节点优化策略和线性插值处理

改进的路径节点优化策略是基于贪心算法的思想, 从起点开始遍历当前路径解中所有的节点, 判断该节点与下一节点之间的连线是否为无碰撞路径, 若是则继续向下一节点寻找, 直至寻找到一个会发生碰撞的节点, 则将最后一个无碰撞路径的节点与当前节点直接相连, 移除它们之间的其他节点, 并从新的节点开始再次循环查找直至目标点。该策略的优势在于可以去掉可行解中的无用拐点和节点, 使路径距离更短, 所得路径解也更接近最优解。

该策略所得到路径解中的节点很少, 且由于改进算法的设计当中采用了三阶 B 样条曲线优化策略, 曲线的参考点过少会导致最终生成的曲线曲率过小, 使曲线的路径长度成本太高, 甚至出现曲线优化后的路径解与障碍物环境发生碰撞的可能。

因此在简化路径节点后引入线性插值策略, 完成节点优化处理后, 不直接进行三阶 B 样条曲线优化处理, 而是对节点优化后的路径进行线性插值处理, 如式(5)所示, 其中 q_k 为节点优化后的当前节点, q_{k+1} 为节点优化后 q_k 的下一节点。由于节点优化可能导致 q_k 和 q_{k+1} 间距过大, 因此需要在 q_k 和 q_{k+1} 之间生成 t_n 个线性插值点 r_k。

$$r_{k,\ell}=q_k+\frac{\ell}{t_n+1}(q_{k+1}-q_k),\quad \ell=1,\cdots,t_n \tag{5}$$

生成后的新路径如式(6)所示, 最后再进行三阶 B 样条曲线优化处理得到最终路径。

$$\mathcal{P}_2=(q_1,r_{1,1},\cdots,r_{1,t_n},q_2,q_3,\cdots,q_{\mathrm{N}}) \tag{6}$$

3.2　三阶 B 样条曲线优化处理

B 样条曲线是一种数学工具, 用于创建一组指定控制点的平滑曲线, 它结合了样条曲线的平滑特性和 B 样条基函数的局部控制特性[9]。B 样条曲线是 B 样条基函数的线性组合, 基函数在给定的区间上构成线性空间, 曲线由一组控制点定义, 控制点依次相连形成控制多边形。通过调整这些控制点的位置, 可以局部地控制曲线的形状[10]。

B 样条曲线上的点可由式(7)表示[11]:

$$P_{(d)}=\sum_{b=0}^{n}P_bF_{b,k}(t) \tag{7}$$

其中: $P_{(d)}$为曲线上点的坐标, n 为控制点数量, P_b 是曲线上特征点的坐标, $F_{b,k}(t)$ 为样条曲线的基函数[12], b 为坐标索引, k 为多项式最高次幂[13], t 为绘制曲线时的取值, 三阶 B 样条曲线方程的基函数可由式(8)表示[14], 其中 j 为控制点的序号, m 为累加变量, k+1 为元素的总数。

$$F_{b,k}(t)=\frac{1}{k!}\sum_{m=0}^{k-b}(-1)^m\binom{m}{k+1}(t+k-m-j)^k \tag{8}$$

经过推导可得:

$$F_{0,3}(t)=\frac{1}{6}(1-t)^3 \tag{9}$$

$$F_{1,3}(t)=\frac{1}{6}\left(3t^3-6t+4\right) \tag{10}$$

$$F_{2,3}(t)=\frac{1}{6}\left(-3t^3+3t^2+3t+1\right) \tag{11}$$

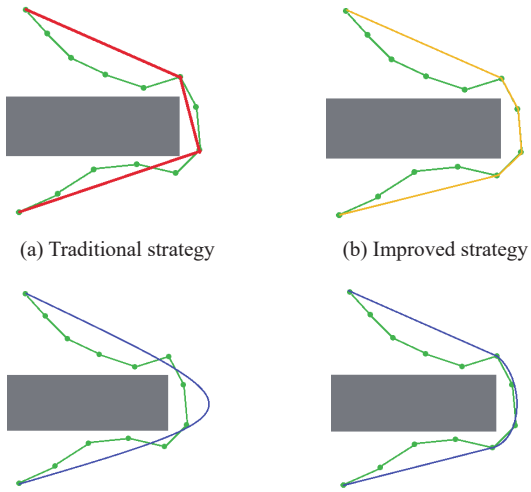
$$F_{3,3}(t)=\frac{1}{6}t^3 \tag{12}$$

将式(9)~(12)代入到式(8)可以得到三阶 B 样条曲线方程[15], 如式(13)所示:

$$P_{(\mathrm{d})}=p_0\cdot F_{0,3}(t)+p_1\cdot F_{1,3}(t)+p_2\cdot F_{2,3}(t)+p_3\cdot F_{3,3}(t) \tag{13}$$

其中: p_0、p_1、p_2、p_3 分别为所求点 P_d 的各级控制点。由于三阶 B 样条曲线拟合过程中无需控制特征点坐标, 在求得待拟合的点位坐标后即可进行近似拟合[16], 因此选择了三阶 B 样条曲线的拟合方法, 以便简化算法的复杂度。

路径节点优化的对比示意图如图 6 所示。其中绿色路径为最初路径解, 图 6(a)所示为一般节点优化策略下所生成的路径解, 图 6(b)所示为经过节点优化和线性插值后的路径解, 图 6(c)所示为一般优



(a) Traditional strategy　(b) Improved strategy

(c) Traditional curve optimization　(d) Improved curve optimization

图 6　节点优化策略效果对比图

Fig. 6　Optimization effect comparison chart

化策略下所生成的三阶B样条曲线优化结果,图6(d)所示为改进优化策略下所生成的三阶B样条曲线优化结果。

路径二次优化环节总体流程如图7所示。基于MATLAB平台实现的路径二次优化效果如图8所示,其中蓝色线段为进行路径节点优化策略后的路径,红色线段为该路径的三阶B样条曲线优化结果。

3.3 算法可行性分析

3.3.1 概率完备性分析 概率完备性是指在给定充足的失败次数、足够长的时间且该问题客观存在可行路径解的前提下,该算法一定会找到一条可行路径解。对于RRT算法来说,其自身作为一种基于采样的路径规划算法,且依赖随机点进行采样,只要采样点足够多、算法运行时间足够长就一定能找到可行路径解。即当时间趋于无穷时,总能找到一个以起始点 X_{init} 为首个节点,以目标点 X_{goal} 作为最后一个节点的集合 V,使得集合 $V=\{X_{\text{init}},\cdots,X_{\text{goal}}\}$,且顺次连接节点的边的集合 E 都有 $V\subset X_{\text{free}}$,$E\subset X_{\text{free}}$。

改进的自适应采样节点策略,随着算法运算时间的推移,在障碍物环境中生成的节点会逐渐增加,进而提升自适应策略下节点生成的效率。而当 P_a 超过设定的阈值 p_n 时,节点采样将不会再有任何自适应和目标导向性,从而进行传统的全地图随机节点采样,因此也可以保证在时间足够的情况下,该算法可以拥有充足的采样节点,进而该算法仍然具有概率完备性。

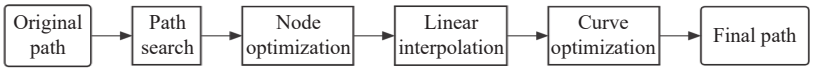


图7 路径二次优化流程图

Fig. 7 Path secondary optimization flow chart

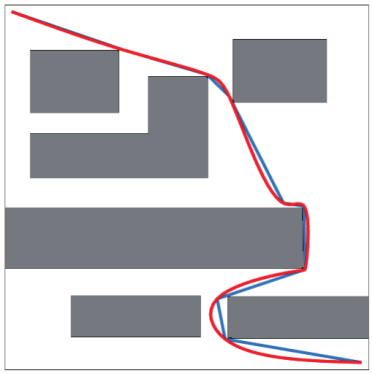


图8 路径二次优化效果图

Fig. 8 Path secondary optimization rendering

3.3.2 时间复杂度分析 时间复杂度可以从整体上反映算法的时间成本,一般情况下可以使用大O表示法(Big O Notation)来计算时间复杂度。在路径规划算法中相较于代码中各个环节的迭代、循环次数,算法代码中的主要步骤更能体现该算法的时间复杂度,由文献[17]可知RRT算法和RRT*算法都具有相同的时间复杂度 $O(N\lg N)$。

改进的BI-RRT算法的主要步骤包括:初始化、失败率求和计算、采样点获取、获取最近点、获取扩张点、碰撞检测、更新节点、更新失败概率、结束判断、路径二次优化处理,设上述步骤的时间复杂度分别为:T_{Init}、$T_{\text{Sumfailtimes}}$、$T_{\text{Samplepoint}}$、T_{Nearest}、T_{New}、$T_{\text{Collisionfree}}$、T_{Rewire}、T_{Resum}、T_{End}、T_{Path};总时间复杂度为 T,接下来深入分析上述步骤各自的时间复杂度。

由于算法计算过程简单,$T_{\text{Sumfailtimes}}$、$T_{\text{Samplepoint}}$、T_{Rewire}、T_{Resum}、T_{End} 以及 T_{Path} 通过基础运算法则以及集合运算规律在线性时间内即可完成,所以这些步骤在算法中的时间复杂度都为 $O(N)$。已知 T_{Nearest} 和 T_{New} 的时间复杂度与RRT算法相同,都为 $O(N\lg N)$[18]。$T_{\text{Collisionfree}}$ 包括检测节点 X_{New} 是否落在自由空间 X_{Free} 中,以及节点 X_{New} 与 X_{Nearest} 之间的连线不与障碍物空间 X_{Obstacle} 发生碰撞,因此其时间复杂度为 $O(N\lg N)$[19]。

综上所述,该改进BI-RRT算法的时间复杂度可计算为:

$$\begin{aligned}T=&T_{\text{Init}}+T_{\text{Sumfailtimes}}+T_{\text{Samplepoint}}+T_{\text{Nearest}}+T_{\text{New}}+\\&T_{\text{Collisionfree}}+T_{\text{Rewire}}+T_{\text{Resum}}+T_{\text{End}}+T_{\text{Path}}=\\&O(N)+O(N)+O(N)+O(N\lg N)+O(N\lg N)+\\&O(N\lg N)+O(N)+O(N)+O(N)+O(N)=\\&7O(N)+3O(N\lg N)\approx O(N\lg N)\end{aligned}\tag{14}$$

因此根据该时间复杂度计算规则改进的BI-RRT算法的时间复杂度应为 $O(N\lg N)$,与典型RRT、RRT*算法的时间复杂度相同。

3.3.3 空间复杂度分析 空间复杂度指该算法在运行期间所需要的内存空间,同样也可以用大O表示法来进行分析计算。与典型RRT算法相同,设改进的算法所需要的采样点总数为 N,则总体空间复杂度为所有节点的集合 V 及这些节点相邻连接的边的集合 E 所占用的存储空间之和,即:

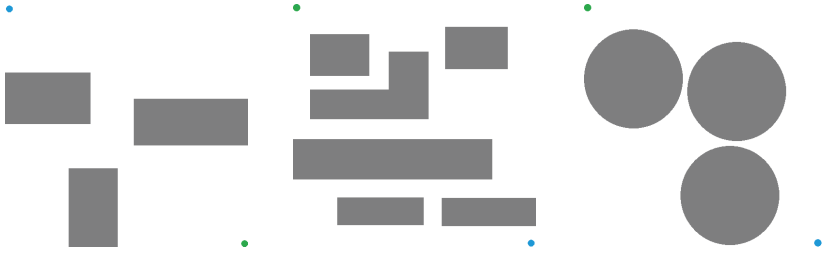
$$|V|+|E|=N+(N-1)=2N-1\tag{15}$$

综上,改进算法的空间复杂度与典型RRT算法相同,都可以表示为 $O(N)$。

4　算法测试分析

4.1　测试用例分析

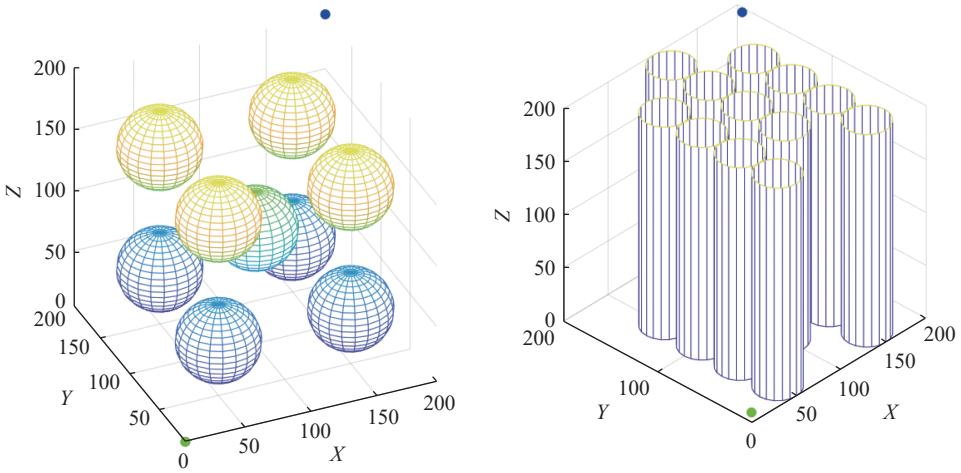
二维的路径规划结果对比部分共采用 3 张不同的障碍物环境地图，所有二维障碍物地图均由 500×500 个像素点组成，起始点位于左上角，目标点位于右下角，样例图如图 9 所示。三维环境对比部分共采用了 2 张障碍物环境地图，所有障碍环境地图均由 200×200×200的单元组成，测试样例图如图 10 所示，其中起始点（绿色点）位于坐标位置（5, 5, 5），目标点（蓝色点）位于坐标位置（195, 195, 195）。本文后续算法中使用 2D-1、2D-2、2D-3、3D-1、3D-2 分别表示图 9（a）、9（b）、9（c）、10（a）、10（b）所示的测试例图。



(a) Example diagram 1　(b) Example diagram 2　(c) Example diagram 3

图 9　二维测试样例图

Fig. 9　2D test sample diagram



(a) Example diagram 1　(b) Example diagram 2

图 10　三维测试样例图

Fig. 10　3D test sample diagram

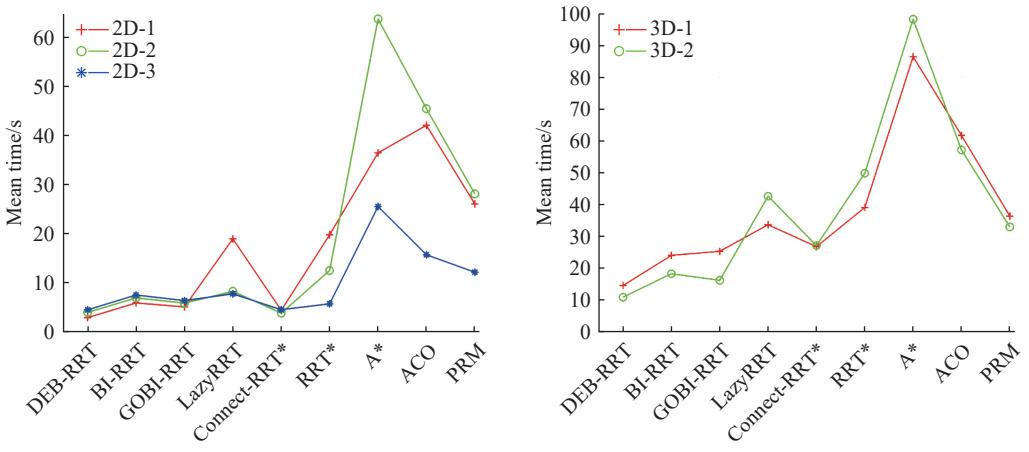
4.2　测试结果分析

基于 MATLAB 平台的改进算法仿真实验在联想 LEGION 电脑上进行，处理器为 12thGen Intel(R) Core(TM) i7-12700H 2.30 GHz, 内存 24.0 GB, GPU 为 NVIDIA RTX 3070 Laptop。设备采用基于 x64 处理器的 64 位 Windows11 操作系统，软件版本为 MATLAB R2020b。进行对比的典型路径规划算法包括：A*、ACO、PRM、LazyRRT、RRT*、Connect-RRT*、BI-RRT、GOBI-RRT。

每种算法在相同条件下，对每个障碍物环境独立运行并得到运行时间以及路径长度两个参考指标。所有算法均取最优数据，RRT 算法判断的结束条件均为 5 个像素格内。

4.2.1　算法运行时间分析　二维环境的算法运行时间数据结果如图 11（a）所示，相较于典型 BI-RRT 算法，改进算法在二维测试环境中的时间效率提升可达到 20%~50%，而相较于加强了目标导向的 GOBI-RRT 算法，改进算法仍然具有更短的运算时间，尤其是在复杂地图中时间提升更明显，这表明了改进算法的自适应采样概率策略有更加出色的目标导向性。

三维环境的算法运行时间数据结果如图 11（b）所示，从数据图中可以看出，随着测试环境的维数增加，改进算法的时间效率优势相较于传统 BI-RRT 算法愈发凸显，尤其是在复杂环境中，传统的目标导向策略并不能使算法快速脱离当前障碍物环境，而改进算法的目标导向性区域遍布整个平面，根据失败概率不同，可以动态调整策略，快速帮助其脱离障碍物区域，因此有更好的目标导向性和时间效率。



(a) 2D environment (b) 3D environment

图 11 平均运行时间对比图

Fig. 11 Runtime comparison chart

4.2.2 路径长度计算分析 二维环境样例图 9(a)的路径长度数据结果如图 12 所示,其中黄色、蓝色、红色线段所示分别为 10 次运行结果中所有环境的平均值、最大值和最小值。借助路径平滑策略,改进算法的路径长度在所有对比的 RRT 算法中最短,在部分地图中优于其他算法。

三维环境样例图 10(a)的路径长度数据结果如图 13 所示。改进算法受维度变化的影响,路径长度结果的浮动变化幅度明显增加,在三维环境中以目标点作为偏置的 RRT 算法往往更具有优势,但改进算法除 GO-BIRRT 外相较于其他 RRT 算法仍具有更短的路径结果。其余样例图的路径平均长度与路径长度方差数据如表 1 所示,从表中数据可知改进算法在其余样例图中平均路径长度结果普遍优于其他算

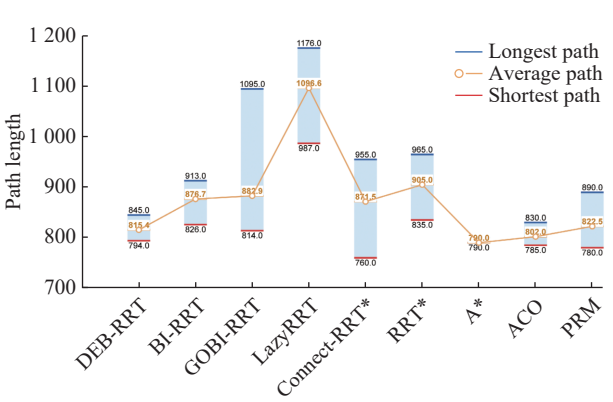


图 12 二维样例图 9(a)路径长度对比图

Fig. 12 Comparison of path length for 2D samples Fig. 9(a)

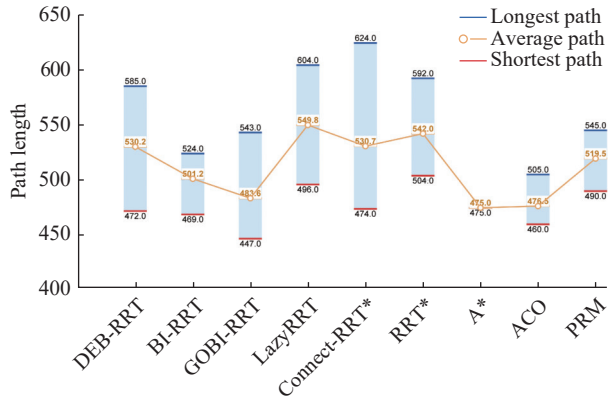


图 13 三维样例图 10(a)路径长度对比图

Fig. 13 Comparison of 3D sample path lengths Fig. 10(a)

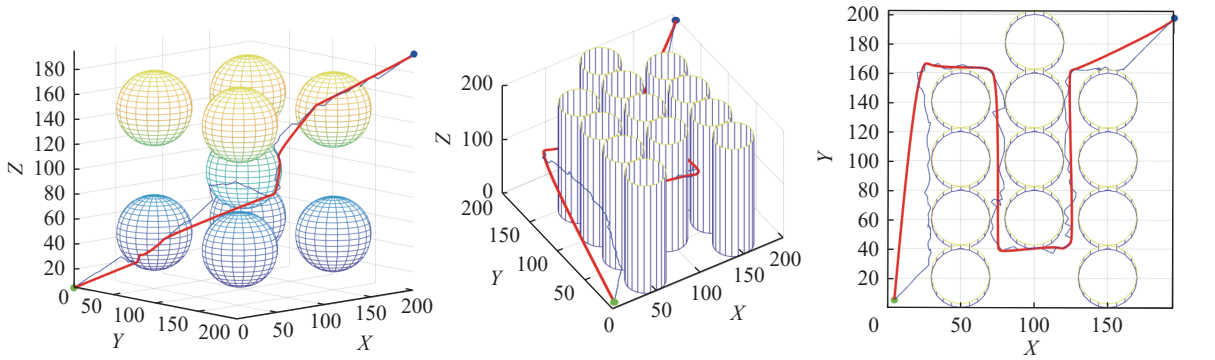
表 1 不同样例图算法的平均路径长度及路径长度方差

Table 1 Average path length and path length variance of different sample graphs

Algorithm	Average path length			Path length variance				
	2D-2	2D-3	3D-2	2D-1	2D-2	2D-3	3D-1	3D-2
DEB-RRT	856.9	844.2	506.8	72.3	26.7	205.4	354.6	306.3
BI-RRT	914.5	968.3	491.9	210.2	850.7	1 248.4	87.1	128.4
GOBI-RRT	892.5	908.8	465.9	2 193.3	870.3	1 456.7	256.0	156.3
LazyRRT	1 127.4	987.1	531.9	1 013.3	1 178.8	2 177.8	324.0	441.0
Connect-RRT*	998.5	1 016.0	557.5	1 024.0	2 756.3	1 736.1	625.0	1 284.0
RRT*	894.0	923.5	513.5	469.4	277.8	711.1	215.1	100.0
A*	835.0	990.0	463.0	0	0	0	0	0
ACO	846.0	836.5	499.7	64.0	44.4	56.3	56.25	34.0
PRM	861.0	947.5	513.3	330.0	306.3	2 417.4	84.0	164.7

法, 而从方差数据可知改进算法相较于其他算法方差较小, 说明算法扩展效率更高、收敛性更好。

三维环境的路径对比图如图 14 所示, 其中红色线为改进的动态椭球体采样策略结果, 蓝色线为传统目标偏置策略结果。



(a) Sample1 comparison diagram　(b) Sample2 comparison diagram　(c) Sample2 overlook comparison diagram

图 14　三维样例图路径对比图

Fig. 14　3D sample path comparison

5　结　语

本文针对 RRT 算法分别基于二维和三维环境提出不同的自适应采样节点策略, 并从基于贪心思想简化节点、线性插值算法保证路径完整度、三阶 B 样条曲线平滑路径策略三个方面进行路径二次优化。基于 MATLAB 平台仿真结果表明改进算法在运行时间和路径长度上相较于传统 RRT 算法及其他路径规划算法具有一定的竞争力。但同时本文所改进的算法在动态障碍物环境下具有一定的局限性, 采样失败概率 p_a 在动态空间中的增长速度明显变快, 甚至可能出现通过对失败阈值 p_n 的改变也无法解决的情况。因此后续算法改进的方向一方面是基于现有改进方案提出对动态障碍物环境的改善策略, 另一方面是在极高维空间中算法的更多可行性改进的探索。

本文提出的改进算法计划应用于多机器人焊接系统, 自主生成焊接路径。后续研究将进一步在真实工业场景中, 系统对比分析本算法的规划结果、人工示教方案以及现有先进路径规划算法的性能差异。

参考文献:

[1] LAVALLE S M, KUFFNER J J. Rapidly random-exploring trees: Progress and prospects[J]. Intelligent Robots and Systems, 2001, 20(5): 347-400.

[2] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[C]// IEEE International Conference on Robotics and Automation. San Francisco: IEEE, 2000: 995- 1001.

[3] WANG J, CHI W, LI C, *et al*. Neural RRT*: Learning-based optimal path planning[J]. IEEE Transactions on Automation Science and Engineering, 2023, 20(3): 1645-1656.

[4] ZHANG H, WANG Y, ZHENG J, *et al*. P-RRT*: Potential-guided RRT* for motion planning in cluttered environments[J]. IEEE Transactions on Industrial Electronics, 2023, 70(10): 10363-10372.

[5] LI Y, WEI W, ZHANG Y. Q-RRT*: A quick RRT* algorithm based on ancestor optimization and triangular inequality[J]. IEEE Robotics and Automation Letters, 2023, 8(5): 2845-2852.

[6] WANG J, CHIOU W, HSU M, *et al*. Neural RRT*: Learning-based optimal path planning[J]. IEEE Transactions on Automation Science and Engineering, 2022, 19(2): 1168-1181.

[7] 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[J]. The International Journal of Robotics Research, 2018, 37(13/14): 1542-1559.

[8] WANG X, WEI J, ZHOU X, *et al*. AEB-RRT*: An adaptive extension bidirectional RRT* algorithm[J]. Autonomous Robots, 2022(6): 46.

[9] WANG L, GUO Y. Speed adaptive robot trajectory generation based on derivative property of B-Spline curve[J]. IEEE Robotics and Automation Letters, 2023, 8(4): 1905-1911.

[10] XU Z, XIU Y, ZHAN X, *et al*. Vision-aided UAV navigation and dynamic obstacle avoidance using gradient-based B-spline trajectory optimization[C]//2023 IEEE Internation-

al Conference on Robotics and Automation. London, United Kingdom: IEEE, 2023: 1905-1911.

[11] HONIG W, PREISS J A, KUMAR V, *et al*. Trajectory planning for quadrotor swarms[J]. IEEE Transactions on Robotics, 2018, 34(4): 856-869.

[12] SUN W, SUN P X , DING W, *et al*. Gradient-based autonomous obstacle avoidance trajectory planning for B-spline UAVs[J]. Scientific Reports, 2024, 14(1): 14458

[13] 赵卫东, 唐顾杰, 宋江一. 基于改进 JPS 与三次 B 样条插值的路径规划算法 [J]. 安徽工业大学学报 (自然科学版), 2022, 39(2): 189-195.

[14] 刘梦奇, 王维强, 田良宇. 基于 B 样条曲线的无人驾驶车辆 Informed RRT*算法研究 [J]. 智能计算机与应用, 2022, 12(4): 25-29.

[15] 张跃明, 薛奇, 纪姝婷. 满足曲率约束的 B 样条曲线连续路径平滑方法 [J]. 华中科技大学学报 (自然科学版), 2022, 50(5): 59-65,72.

[16] 钱东海, 孙林林, 赵伟. 基于三次 B 样条曲线的叉车型 AGV 路径规划研究 [J]. 计算机测量与控制, 2022, 30(4): 177-181,189.

[17] JEONG I B, LEE S J, KIM J H. Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate[J]. Expert Systems with Applications, 2019, 123: 82-90.

[18] KARAMAN S, FRAZZOLI E. Incremental Sampling-based algorithms for optimal motion planning[C]//Robotics: Science and Systems (RSS 2010). Zürich, Switzerland: [s.n.], 2010.

[19] ZHANG Z, ZHANG J, ZHAO Y. A novel hexagonal grid map model and regenerated heuristic factor based strategy for intelligent manufacturing system's AGV path planning problem solving[J]. Computers & Industrial Engineering, 2024,192: 110154.

DEB-RRT:An Improved RRT Algorithm Based on Dynamic Ellipsoid Sampling

ZOU Yuchen, WANG Xuewu, GAO Yongliang

(Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China)

Abstract: To address the problems of the strong randomness of sampling points, low sampling point utilization, and slow convergence of the bidirectional rapidly-exploring random tree (Bi-RRT) algorithm, this paper proposes an improved DEB-RRT (Dynamic Ellipsoid Bidirectional RRT) algorithm applicable to both 2D and 3D environments. While guaranteeing probabilistic completeness, the improved algorithm dynamically adjusts the goal bias of random sampling points based on the collision check failure rate. Once a feasible path is found, a secondary path optimization step is executed to shorten the final path and smooth its sharp corners. Simulations of the algorithm are carried out on the MATLAB platform. The experimental results demonstrate that, compared with conventional path planning algorithms, the proposed method requires less path search time, yields shorter path length, and delivers better overall performance.

Key words: path planning; adaptive sampling; curve optimization processing; greedy algorithm; BI-RRT

（责任编辑：王晓丽）