文章编号:1674-2974(2026)02-0026-11 DOI:10.16339/j.cnki.hdxbzkb.2026153

基于空间探索引导双树RRT*的路径规划

秦晓辉[1,2],郝中华[1],张润邦[1],刘硕[1],黄圣杰[1],龙承启[1†]
(1. 湖南大学 机械与运载工程学院,湖南 长沙 410082;
2. 湖南大学无锡智能控制研究院,江苏 无锡 214115)

摘 要:针对基于采样的RRT*路径规划算法存在采样盲目性、引导策略单一和运动学约束等问题,提出了一种空间探索引导双树RRT*(space exploration guided bidirectional RRT*,SGB-RRT*)路径规划算法.首先,在采样阶段采用空间探索采样策略,探索当前节点周围环境并建立障碍物视角,剔除遮挡区域,降低了采样盲目性.其次,在引导阶段采用衰减探索与权重控制策略,先依据采样次数更新衰减探索式,渐进限制探索能力,引导采样向相对目标点倾向,再通过距离权重控制倾向程度,完善了引导逻辑.再次,在扩展阶段将剔除CCC类型的Dubins曲线与双树融合,解决运动学约束问题,然后以逐级回溯父节点代替Rnear回溯方式,最终快速规划出一条低代价的运动学路径.仿真实验在不同地图中将SGB-RRT*与RRT、RRT*、RRT*-connect进行对比,验证了提出的SGB-RRT*算法在计算速度、路径质量方面的优越性和可行性.

关键词:RRT;路径规划;探索采样;衰减探索;Dubins曲线;运动学约束
中图分类号:TP273 **文献标志码**:A

Path planning based on space exploration guided bidirectional RRT*

QIN Xiaohui[1,2],HAO Zhonghua[1],ZHANG Runbang[1],LIU Shuo[1],HUANG Shengjie[1],LONG Chengqi[1†]
(1. College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China;
2. Wuxi Intelligent Control Research Institute of Hunan University, Wuxi 214115, China)

Abstract: To tackle challenges such as sampling blindness, limited guidance strategies, and kinematic constraints inherent in the sampling-based RRT* path planning algorithm, this paper introduces an enhanced Space Exploration Guided Bidirectional RRT* (SGB-RRT*) path planning algorithm. The proposed method innovates by implementing a spatial exploration sampling strategy during the initial sampling phase. This approach actively investigates the environment surrounding the current node to establish comprehensive obstacle perspectives, thereby mitigating occlusion issues. This diminishes the blindness associated with sampling. In the subsequent guidance phase, the SGB-RRT* adopts a decay exploration mechanism coupled with a weight control strategy. The decay exploration dynamically adjusts its parameters based on the cumulative number of samples, progressively narrowing the exploration scope to concentrate efforts on guiding the sampling process toward the target region. Concurrently,

* **收稿日期**:2025-01-02
基金项目:国家重点研发计划项目(2022YFB4700503), National Key Research and Development Program of China (2022YFB4700503)
作者简介:秦晓辉(1988—),男,湖南张家界人,湖南大学副研究员,博士生导师
† 通信联系人,E-mail:whkxa_618_ybz@163.com

distance weight is used to regulate the tendency degree. During the extension phase, the Dubins curves with the CCC type excluded are integrated with the bidirectional tree structure to address the kinematic constraints. In lieu of the conventional Rnear-based backtracking, it employs parent node backtracking. As a result, the algorithm efficiently generates low-cost, feasible paths that respect the kinematic constraints. Simulation experiments conducted across diverse maps compare the performance of SGB-RRT* against RRT, RRT*, and RRT*-connect algorithms. These tests confirm the superior computational efficiency, improved path quality feasibility of the proposed SGB-RRT* approach.

Key words: RRT; path planning; exploration sampling; decay exploration; Dubins curve; kinematic constraints

---

无人车的路径规划在多种复杂应用场景中具有重要作用,包括城市交通、物流配送、园区巡逻以及灾害救援等领域[1].路径规划旨在尽可能快地找到一条从起点到终点的运动学无碰撞路径[2].

常用的无人车路径规划方法主要有基于图搜索的方法和基于采样的方法[3].图搜索方法主要包括Dijkstra算法[4]、A*算法[5]等,这些算法通过对地图进行栅格化处理并遍历栅格节点来找到路径[6].然而,在高维环境中,图搜索方法的计算复杂度会剧增[7],且路径的精度高度依赖于栅格分辨率[8].Ziegler等人[9]将A*算法与两种启发式算法结合并将其用于DARPA挑战赛.为了提升A*算法对无人车的适应能力,Dolgov等人[10]提出了一种Hybrid-A*算法,将车辆前轮转角进行离散,拓展得到满足运动学约束的路径.Chu等人[11]提出了构建运动学路径的A*变体,忽略栅格分辨率构建出平滑的路径.此类算法能够找到全局最优路径,但在无人车的实时路径规划中,由于计算量大,实时性较差,难以满足实际需求.采样方法主要有RRT[12]、RRT*[13]、RRT-connect[14]等算法.这类算法在搜索空间内随机采样生成节点,并逐步构建连接路径,无需栅格化地图且在高维空间中优势明显,很好地适应了无人车路径规划的实时性要求[15].RRT规划的路径基于概率随机产生,实时性较好但鲁棒性较差,不能完全满足要求.RRT*通过节点重连实现了渐进优化,在采样次数为无穷大时获得最优路径.RRT*-connect[16]是RRT*的双树形式,分别在起终点生长出节点树进行扩展,提升了连接速度但两者都存在采样盲目、渐近最优时间长的问题.Gammell等人[17]提出的Informed-RRT*算法将采样范围限制在椭圆区域来减少冗余采样,提升了收敛速度.Yuan等人[18]针对树生长无方向性问题提出了BG-RRT算法.Liao等人[19]提出了一种F-RRT*算法,通过逐级回溯父节点并创建新的父节点来优化路径,降低了路径代价,提升了收敛速度.Fan等人[20]基于PF-RRT*和APF算法,依据随机点、目标点和障碍物引力快速连通路径.Yu等人[21]提出了一种实时的RDT-RRT框架,提升了算法避障能力.王东振等人[22]将RRT与Dubins结合提出了RRT-Dubins方法,解决了运动学约束问题.葛超等人[23]改进了Informed-RRT*算法,基于人工势场法优化了点选取策略.王蔡琪等人[24]提出了一种AE-RRT*算法,基于障碍物距离自适应扩展步长.

针对现有研究在计算速度、路径质量、非完整性转向车辆的约束优化方面还难以满足诸如室内多障碍物、封闭园区、地下车库等场景的使用要求,本文重点研究已有方法存在的盲目选点采样效率低、节点选取引导逻辑单一以及双树运动学约束不完善等问题,提出了一种满足无人车运动学约束的空间探索引导双树RRT*路径规划算法:SGB-RRT*.该方法的创新点主要包括以下三个方面:

1) 提出了一种空间探索采样策略,探索当前节点周围地图环境并建立障碍物视角,据此剔除遮挡区域,实现探索采样,降低采样盲目性.

2) 提出了一种衰减探索与权重控制策略,依据最大采样次数和当前采样次数更新衰减探索式,并设置距离权重系数控制引导倾向程度,完善引导逻辑.

3) 融合了双树采样与Dubins曲线,使用逐级回溯父节点代替Rnear回溯方式,满足运动学约束并提升收敛速度.

1 相关工作

1.1 问题定义

针对规划任务要求,对自车从起点到目标点进行路径规划问题建模,建模方式类似于文献[25].

将环境地图中自车的工作空间定义为配置空间S(S空间),S空间中的元素$q \in S$用来描述自车的位置和体积.S空间中的一个子集S_{obstacle}为不可通行的障碍物空间,S_{obstacle}相对于配置空间S的补集S_{free}为无碰撞的自由移动空间.出于安全性考虑,将障碍物进行膨化处理,该部分也是不可通行空间,表示为S_{expand}.在算法层面,S_{expand}与S_{obstacle}等同.各空间关系如下:

$$S = S_{\text{free}} \cup S_{\text{obstacle}},\ S_{\text{free}} \cap S_{\text{obstacle}} = \varnothing \tag{1}$$

定义c_i: $C \to R+$表示一个连续、可微的代价函数,该函数为每段路径i分配一个正代价值.

综上,路径规划问题转化为在S空间中,给定起点配置$(q_{\text{init}}, q_{\text{goal}}) \in S_{\text{free}}$和代价函数$c$: $C \to R+$,求解一条路径π:$[0,T] \to S$.对于$\forall t \in [0,T]$,$\pi(t) \in S_{\text{free}}$,常见的代价定义方式是路径欧氏距离的积分[26],即设当前路径π,π中的节点数为n,$n \to \infty$时,路径代价近似为路径解点集间连线的距离之和.则路径代价:

$$c_\pi = \lim_{n \to +\infty} \left(c\left(\pi\left(\frac{i}{n}\right)\right) \cdot \Delta d \right) = \int c\left(\pi\left(\frac{i}{n}\right)\right) \cdot \Delta d = \frac{\text{length}(\pi)}{n} \sum_{i=1}^{n} c\left(\pi\left(\frac{i}{n}\right)\right) \tag{2}$$

式中:Δd为路点间距的微分;$\text{length}(\pi)$是路径π的总长度;$\pi(i/n)$为路径π的第i个节点;$c(\pi(i/n))$为第$(i-1)$个节点到第i个节点的代数值.空间中的可行路径不唯一,需要依据路径代价进行最优化筛选,本文同样使用欧氏距离积分作为筛选指标.令Π表示S_{free}中所有可行路径解的集合,则最优路径解为在Π中取最小的c_{π^*}满足:

$$c_{\pi^*} = \min\{c_\pi | \pi \in S_{\text{free}}\} \tag{3}$$

1.2 单树RRT*算法

RRT*算法作为单树采样算法的代表,是对RRT算法的改进.RRT算法以起点为根节点在空间中随机采样点q_{rand}及其在树中的最近点q_{nearest},若从q_{rand}到q_{nearest}的连线无碰撞,则进行扩展得到新节点q_{new},并将其添加到树中,完成一次迭代.算法重复此过程,直到起点与终点连通,完成路径规划.在此基础上,RRT算法增加了最大迭代次数n和回溯半径R_{near}.在每次迭代时回溯R_{near}范围内的所有节点,通过重连操作找到代价更小的局部路径,逐步优化,直至达到n次迭代,最终实现渐进最优.RRT*算法原理流程见图1.

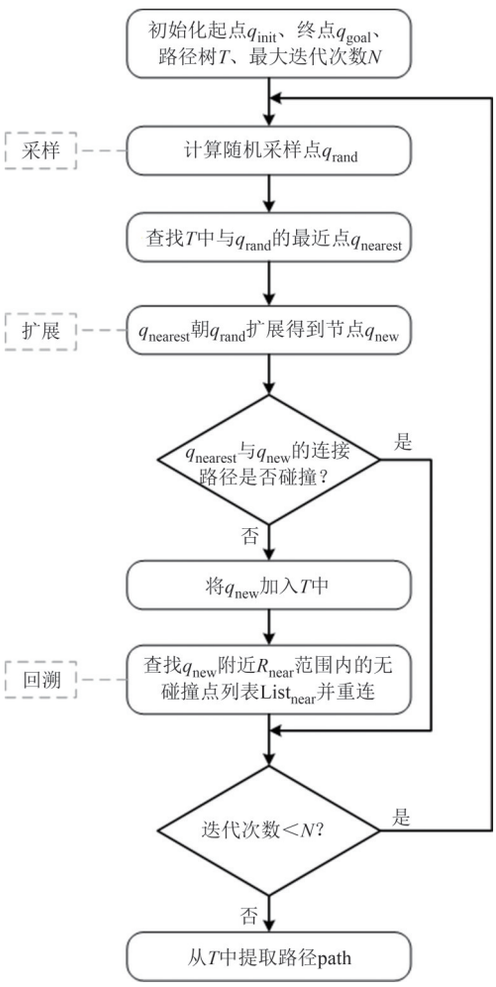


图1 RRT*流程

Fig.1 RRT* process

1.3 双树RRT*-connect算法

作为双树采样算法的典型代表,RRT*-connect是对RRT-connect算法的改进版本.而RRT-connect算法基于RRT算法设计,以起点和终点为根节点,分别向外扩展两棵树,并在每次成功扩展后尝试直接连通两棵树;一旦成功连通,连接点即成为两棵树的公共节点,从该节点回溯至起点和终点即可快速生成路径解.双树采样策略还增强了算法的避障能力,使其能够快速逃离狭窄空间.RRT*-connect结合了RRT和RRT-connect的优势,在首次连通后继续迭代,通过回溯父节点和重连树结构来加快收敛速度,实现渐进最优.RRT*-connect算法的伪代码如伪代

码 1 所示.

伪代码 1 RRT*−connect

输入: q_{init}, q_{goal}, N

输出: path

1. T_{init} ←初始化起点树
2. T_{goal} ←初始化终点树
3. **for** i ← 1 to N **do**
4. q_{rand} ←随机采样
5. T ←定义较长的树
6. **if** 扩展成功 **then**
7. 持续朝 q_{rand} 扩展
8. **if** 两棵树具有公共点 **then**
9. 提取路径
10. **end if**
11. **end if**
12. path ←选取最短路径
13. **return** path

2　SGB−RRT*算法

2.1　空间探索采样

传统采样算法在工作空间内均匀采样,其效果见图 2(a).可以看出,采样点在整个采样区域内均匀分布,其中有相当数量的点落在了障碍物内部,这部分点为无效采样点.由于该方法未能充分利用地图中的障碍物信息,因此产生大量无效采样点,从而显著降低了算法的整体效率.这种不足在复杂环境下尤为突出,制约了算法在实际应用中的性能表现.

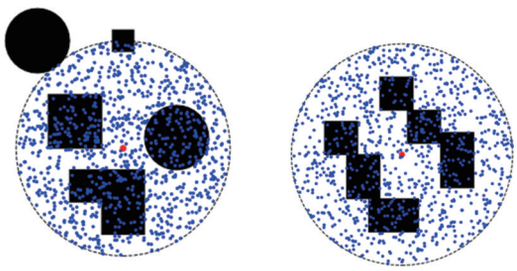
为了降低采样点落在障碍物的概率,提高采样成功率,提出了空间探索采样策略,其具体逻辑见图 2(c).图中红点为当前选中的探索点 P,以 P 为圆心探索半径 r 内的障碍物及其数量 n,再计算与障碍物 i 边界的最短距离 $d_i(i = 1, 2, 3, \cdots, n)$,令计算距离:

$$D_i = d_i + \Delta d(i = 1, 2, 3, \cdots, n) \tag{4}$$

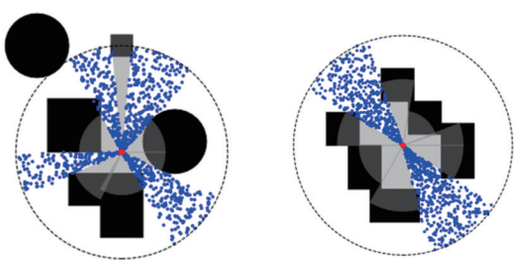
式中:Δd 为预定义的采样半径增量.然后分别以 D_i 作圆交障碍物 i 于 P_{i1}、P_{i2},依据 P、P_{i1}、P_{i2} 计算出 n 个被障碍物遮挡的扇形障碍物视角 α_i,由 α_i 组成的角度集合 U_{obs} 即为应剔除的采样区域,U_{obs} 的补集 U_{free} 为可行区域,两者关系如下:

$$U_{obs} \cup U_{free} = 360^\circ \tag{5}$$

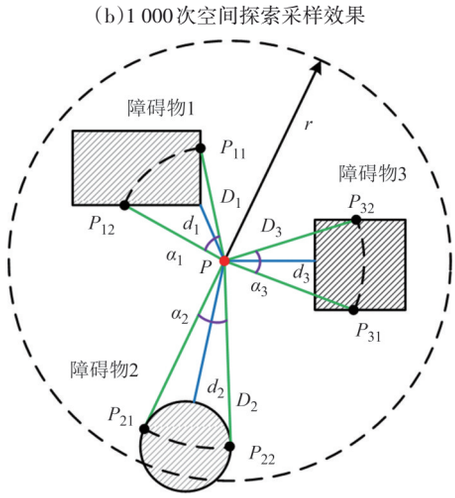
后续分两步得到采样点 P_{rand}.先随机采样得到极坐标的极角 $\theta(\theta \in U_{free})$ 和极径 γ,γ 满足下式:



(a)1 000次均匀采样效果



(b)1 000次空间探索采样效果



(c)空间探索采样逻辑

图 2　采样图示

Fig.2　Sampling diagram

$$\gamma = \text{random}(0\sim\min(d_{boundary}, r)) \tag{6}$$

式中:$d_{boundary}$ 为 P 到边界的最短距离,保证采样点不超出地图边界.然后进行相对坐标转换得到最终的采样点 P_{rand}:

$$\begin{aligned} P_{randX} &= P_X + \gamma \cdot \cos\theta \\ P_{randY} &= P_Y + \gamma \cdot \sin\theta \end{aligned} \tag{7}$$

另外,静态地图中,随机树节点的障碍物视角恒定,需要剔除的遮挡区域也不变.因此记录已计算包络圆节点的可行区域,以便再次访问该节点时只需在此基础上直接采样极角 θ 和极径 γ 即可.

图 2(b)展示了空间探索采样的效果.可以看出在多障碍物和狭窄区域,采样点基本都落在了可行区域,显著提高了采样成功率.空间探索采样的伪代码如伪代码 2 所示,其中:obs 为障碍物集合;yaw 为朝向角;涉及的参数 σ 和 w 将在 2.2 节介绍.

伪代码2 SpaceGuideSample

输入：T, T_{other}, P_{goal}, obs, σ, w, r, Δd

输出：$q_{rand}=(x, y, yaw)$

1. n_{rand} ←从0~1采样数值
2. **if** $n_{rand}>P_{goal}$ **then**
3. **if** $n_{rand}<\sigma$ **then**
4. P ←从当前树中纯随机选基准点
5. **else**
6. P ←从当前树中依据距离权重选基准点
7. **end if**
8. **else**
9. q_{rand} ←从对面树依据距离权重直接选节点
10. **return** q_{rand}
11. **end if**
12. θ ←采样极角
13. $d_{boundary}$ ←计算基准点P到边界的距离
14. γ ←采样极径
15. q_{rand}←极坐标转换为笛卡尔坐标
16. **return** q_{rand}

2.2 衰减探索与权重控制

概率目标引导策略的设计主要基于两方面的考虑：第一，在规划初期，地图中未访问区域较多而已访问区域较少；随着算法的迭代，未访问区域的比例逐渐降低，而已访问区域的比例逐渐增加.这表明从规划开始到结束，算法对探索的需求递减，而对目标引导的需求递增.第二，传统双树的贪婪搜索启发式直接以距离为考量，选择相对最近点进行引导，这容易受遮挡区域最近点的误导，分析见图3.红色点为起点树一点，蓝色点为终点树的点，绿色点为终点树中相对于红点的最近点.显然最近点达不到引导连接的效果.

基于第一点考虑，设计衰减参数σ并每隔k次更新σ以控制探索能力递减.σ和k的表达式如下：

$$\sigma=\max\left(\varepsilon^{[j/k]}, \varepsilon_0\right),\ \{\varepsilon, \varepsilon_0\in(0,1)\} \tag{8}$$

$$k=\mathrm{int}\left(\frac{N\cdot p}{\lg\varepsilon_0/\lg\varepsilon}\right)+1 \tag{9}$$

式中：j为当前迭代次数；ε为衰减因子；ε_0为衰减因子最小值；N为最大迭代次数；p为衰减比例.

基于第二点考虑，设计距离权重参数δ以调节贪婪搜索启发式的距离倾向程度.假设i为终点树中的点(起点树时同理)，每次选择引导点时以δ_i作为点i的选择概率，δ_i的表达式如下：

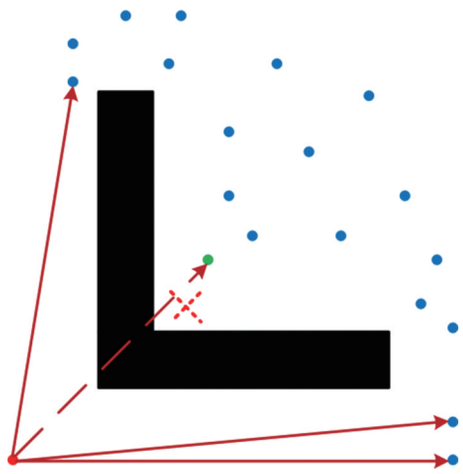


图3 最近点误导

Fig.3 The closest point misleading

$$\delta_i=\frac{1}{\sqrt[w]{d_i}+\varphi}\Big/\sum_{i=1}^{N}\frac{1}{\sqrt[w]{d_i}+\varphi} \tag{10}$$

式中：d_i为起点树中的当前点相对于i点的欧氏距离；N为终点树的点个数；φ为一个保证分母不为0的小正数；w为距离权重系数，$0<w<1$时，相当于增大距离的影响程度，$w=1$时为传统贪婪搜索以最近点为引导点的情况，$w>1$时，相当于降低距离影响程度.

概率目标引导策略结合以上两方面的优化，充分利用了环境信息，提升了收敛速度.

2.3 运动学扩展与逐级回溯

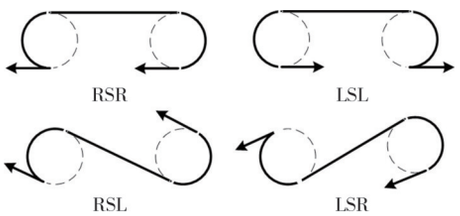
传统采样算法未考虑运动学约束，规划出的路径不连续，仅适用于差速机器人.然而现有对运动学的改进局限于单树采样算法，而双树的运动学适配亟待优化.本算法使用Dubins曲线与双树采样算法融合，以满足运动学约束.

此外，融合Dubins曲线的算法在扩展阶段会增大计算量[27].为了保障算法实时性，本文采取两项措施：一方面，用逐级回溯父节点代替传统的R_{near}回溯重连方式；另一方面，考虑到Dubins的2种CCC类型曲线包含三段圆弧，其图形计算和曲线插值阶段的计算量均大于CSC(圆弧+直线+圆弧)曲线，且CCC类型用在起终点距离较近、方向变化较大的情况，而这类情况会在回溯阶段被优化掉，因此本方法只使用4种CSC类型曲线，从而有效降低了计算复杂度.

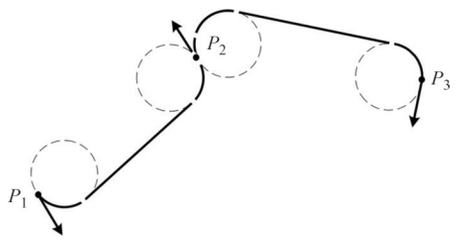
2.3.1 双树融合Dubins曲线

Dubins曲线由圆弧或直线拼接而成，共6种：{RSL, LSR, LSL, RSR, RLR, LRL}，前四种由两段圆弧和一段直线构成，统称CSC类型，后两种由三段圆

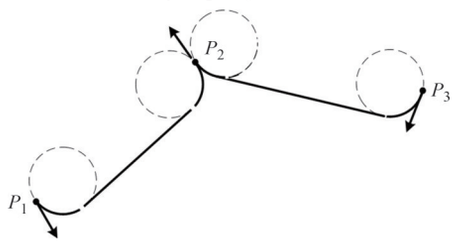
弧构成,称为 CCC 类型.CSC 类型曲线如图 4(a)所示.Dubins 曲线已有成熟的证明和计算方法,本文不做再推导,仅阐述其在双树采样方法中的代价计算方式和融合逻辑.



(a)Dubins-CSC 曲线



(b)$P_1 \rightarrow P_2 \rightarrow P_3$(无尖点)



(c)$P_1 \rightarrow P_2+P_3 \rightarrow P_2$($P_2$处有尖点)

图 4　CSC 曲线

Fig.4　CSC curve

任意一段 CSC 类型曲线的路径代价 c 表达式如下:

$$c = l_{C1} + l_S + l_{C2} = \alpha_{C1} \cdot r + l_S + \alpha_{C2} \cdot r \tag{11}$$

式中:l_{C1}、l_S 和 l_{C2} 分别为起始圆弧段、直线段和终止圆弧段长度;α_{C1}、α_{C2} 分别为起、止圆弧角度;r 为最小转弯半径.

不涉及双树公共节点时,Dubins 曲线与双树之一的融合逻辑如下:Dubins 曲线的输入起点为最近点坐标 P_{nearX}、P_{nearY} 和朝向角 θ_{near},输入终点为随机点坐标 P_{randX}、P_{randY} 和朝向角 θ_{rand} 以及圆弧曲率 K.θ_{near}、θ_{rand} 和 K 计算如下:

$$\theta_{near} = \tan^{-1} \frac{P_{parentY} - P_{nearY}}{P_{parentX} - P_{nearX}} \tag{12}$$

$$\theta_{rand} = \tan^{-1} \frac{P_{randY} - P_{nearY}}{P_{randX} - P_{nearX}} \tag{13}$$

$$K = 1/R \tag{14}$$

式中:$P_{parentX}$、$P_{parentY}$ 为最近点的父节点坐标;P_{randX}、P_{randY} 为 2.1 节空间探索采样得到的随机点 P_{rand} 坐标;R 为无人车最小转弯半径.

另外,Dubins 曲线交换起止点得到的曲线不一定相同,而双树算法求解成功判定逻辑为两棵树具有公共节点.因此,假设 P_1 为起点树节点,P_3 为终点树节点,P_2 为公共节点,若公共节点处的 Dubins 曲线计算沿用式(12)、(13)会产生图 4(c)所示尖点:$P_1 \rightarrow P_2+P_3 \rightarrow P_2$ 的两段路径不能平滑连接.如果将终点树的 P_2、P_3 交换顺序后再计算 Dubins 曲线,则可以实现图 4(b)所示 $P_1 \rightarrow P_2 \rightarrow P_3$ 的两段平滑连接的 Dubins 路径.针对可能出现尖点的连接情况,设计了双树与 Dubins 曲线的连接策略,具体如下:

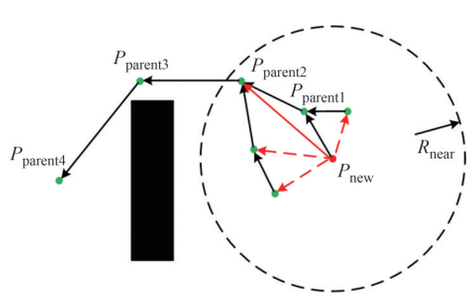
step1:判断当前树是起点树还是终点树.

step2:若是起点树,则起点为 P_1,终点为 P_2(即未交换顺序:树节点指向公共节点).

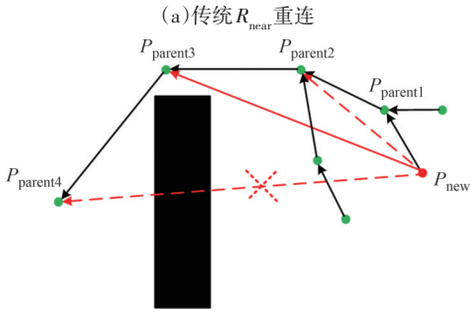
step3:若是终点树,则起点为 P_2,终点为 P_3(即交换顺序:公共节点指向树节点).

2.3.2　逐级回溯父节点重连

图 5 展示了传统 R_{near} 重连与逐级回溯父节点重连.图 5(a)中,每次迭代都会尝试将 P_{new} 与半径内的所有节点重连,随着迭代进行,则圆内的节点数会急剧增加,算法时间成本也会显著增加;图 5(b)只回溯当前点的父节点,当发生碰撞或回溯到起点时结束重连,需要回溯的节点数成线性增长并趋于稳定,该方法在文献[19]中提出并已成功应用于单树 RRT*,本文将其应用到双树算法中.



(a)传统 R_{near} 重连



(b)逐级回溯父节点重连

图 5　R_{near} 重连与逐级回溯父节点重连

Fig.5　R_{near} reconnection and step-by-step tracing back to the parent node reconnection

2.4 SGB-RRT*算法实现

基于双树采样算法进行空间探索采样、衰减探索与权重控制、运动学扩展和逐级回溯优化逻辑的SGB-RRT*算法整体流程如图6所示.

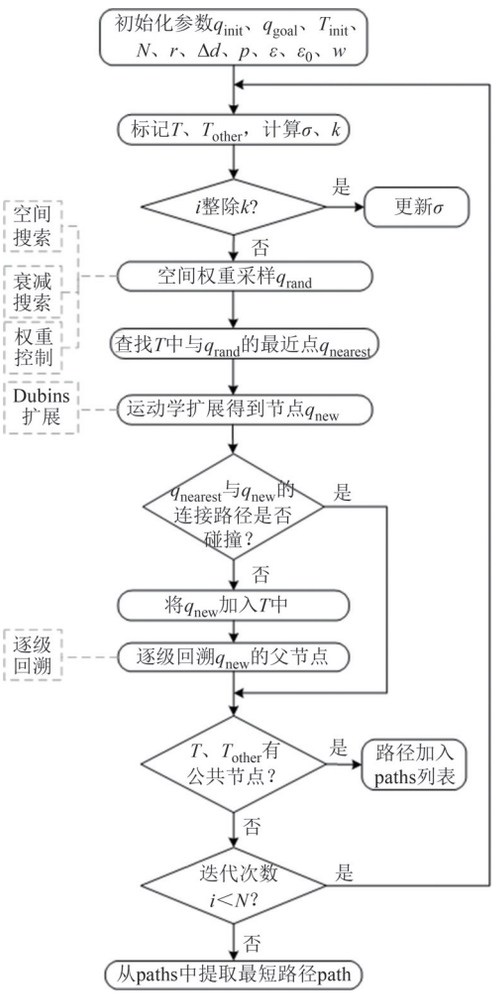


图6 SGB-RRT*流程

Fig.6 SGB-RRT* process

3 算法验证

为了验证所提算法的有效性,将SGB-RRT*算法与RRT、RRT*、RRT*-connect算法分别在三种仿真地图中进行对比实验.实验环境为Windows11操作系统,硬件配置为Intel-Core-i7-10750H@2.60 GHz处理器、16 GB内存的笔记本电脑.所有算法均基于Python实现,并在各算法的公共部分采用相同的函数,以确保实验结果的公平性与准确性.

3.1 实验参数设置

实验所用的仿真地图尺寸均为60×40.起点以蓝点标识,终点以绿点标识,初始路径以红色曲线标识,黑色区域表示障碍物.仿真实验的评价指标包括生成的初始解路径代价和计算时间.此外,由于RRT*、RRT*-connect和SGB-RRT*方法均为渐进最优求解算法,通过对这三种方法在迭代100 s过程中生成的所有解的路径代价和计算时间进行图示化处理,以直观反映算法的收敛速度.

为消除随机采样方法可能带来的偶然性,四种算法均进行了50次独立实验.实验中,各算法的参数设置详见表1.

表1 各算法参数设置

Tab.1 Parameter settings for each algorithm

算法	步长	目标采样概率	回溯半径
RRT	10.0	0.1	无回溯
RRT*	10.0	0.1	5.0
RRT*-connect	10.0	0.1	5.0
SGB-RRT*	10.0	0.1	逐级回溯

注:步长、回溯半径无量纲.

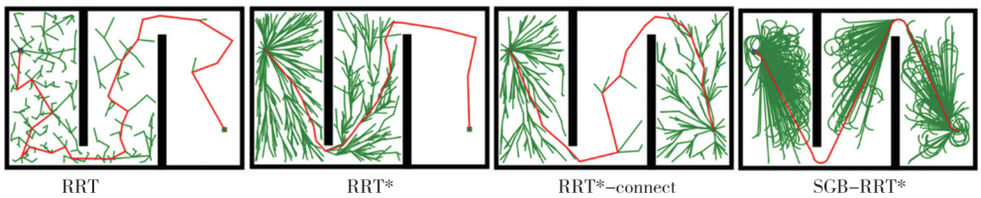
3.2 实验结果对比

仿真实验在三种地图中进行:地图1为简单环境,其反映算法在空旷场景的规划能力;地图2为狭窄环境,其设计为地下停车场等,反映算法在起止点都较狭窄场景的规划能力;地图3为多障碍物环境,其模拟了室内多障碍物环境,反映算法在杂乱障碍物场景的规划能力.各地图实验结果如下.

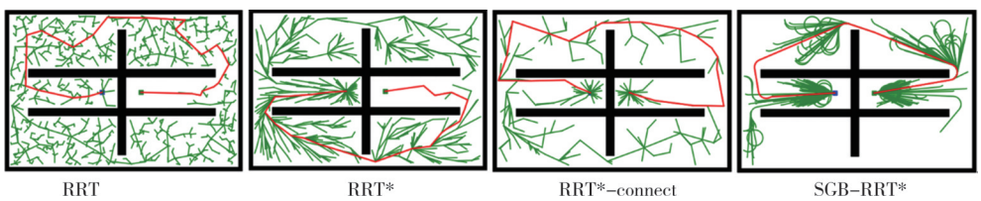
3.2.1 地图1结果

地图1中起点坐标为(4,30),终点坐标为(56,10),初始解的规划结果见图7(a).RRT算法单向搜索,终点处探索效果差,另外由于没有渐进优化,路径更曲折;RRT*算法进行了渐进优化,路径曲折程度下降,但终点搜索效果没有得到改善;RRT*-connect算法双向搜索并进行了渐进优化,起终点搜索效果和路径曲折程度都有所改善;SGB-RRT*算法进一步改进了采样和引导策略,然后使用CSC曲线连接节点和逐级回溯父节点的重连方式,搜索效果和路径质量均较好.

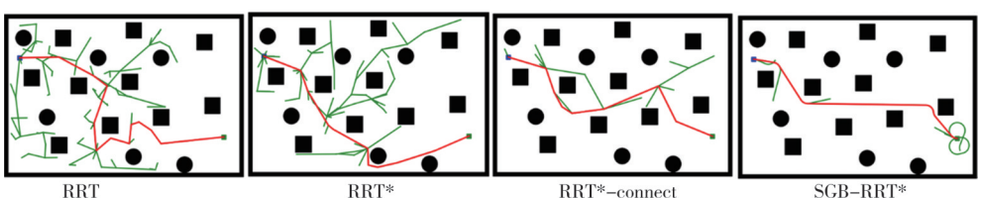
地图1下50次独立实验结果展示在表2中.由表可知,路径代价方面,SGB-RRT*路径代价最低,其余三种对比算法代价均较高;求解耗时方面,RRT无优化、逻辑单一使计算最快,RRT*的优化策略付出了较大的时间成本,RRT*-connect双树逻辑在一定程度上优化了时间成本,而SGB-RRT*算法的时间成本得到进一步优化,基本与RRT算法保持在一个水平;百次迭代耗时反映了算法的计算速度,与初解耗时变化情况一致,同样展示了本算法的优越性.



(a)地图 1 初解



(b)地图 2 初解



(c)地图 3 初解

图 7　3 种地图下算法初解结果

Fig.7　Preliminary results of the algorithms under three maps

表 2　地图 1 算法性能指标(50 次均值)

Tab.2　Map 1 algorithm performance indicators (50 times average)

算法	路径代价	求解耗时/s	迭代次数	每次运行中迭代 100 次耗时/s
RRT	131.616 7	**0.761 9**	**426.06**	0.178 8
RRT*	128.721 9	3.396 0	456.64	0.743 7
RRT*-connect	134.530 7	2.445 3	838.02	0.291 8
SGB-RRT*	**111.613 6**	0.800 1	499.90	**0.160 1**

注:最小值加粗,次小值加下画线,路径代价无量纲.

图 8(a)展示了地图 1 下渐进最优算法的收敛情况.可以看出,SGB-RRT*算法的初解代价最低,收敛速度和终解都最好;RRT*-connect 算法初解和终解与 RRT*算法差别不大,但起始阶段收敛速度优于 RRT*算法.

3.2.2　地图 2 结果

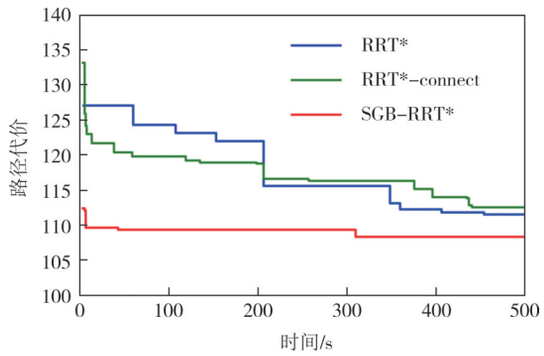
地图 2 中起点坐标为(25,20),终点坐标为(35,20),其初始解的规划结果见图 7(b).其中单树 RRT 和 RRT*算法因环境复杂,几乎探索了整个地图后才找到初解,RRT*-connect 得益于双树对环境的适应能力好而更快地找到了初解,然而这三种方法初解曲折程度仍较高.相比之下,SGB-RRT*算法在复杂环境中的搜索能力最好且路径曲折程度最低.

地图 2 下 50 次独立实验结果展示在表 3 中,与地图 1 相比,地图 2 更加复杂且起终点均处于狭窄区域,RRT 和 RRT*算法计算耗时均大幅增加,RRT*-connect 双树更好地应对了起终点狭窄的场景,耗时明显增加但优于单树算法,而 SGB-RRT*耗时仅略有增加且路径代价仍处于最低,优化效果明显.百次迭代耗时在该地图中同样表明了算法的优势.

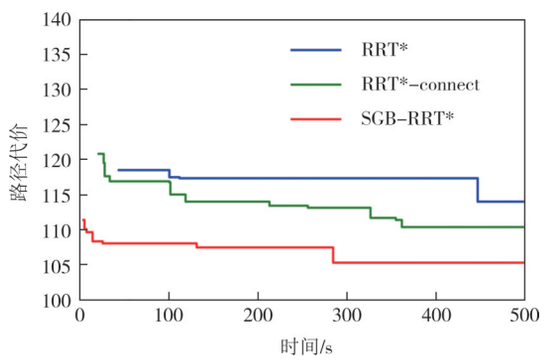
图 8(b)展示了地图 2 下渐进最优算法的收敛情况.RRT*算法对复杂地图的适应能力不足导致初解、终解及收敛速度表现均不好;RRT*-connect 效果相比单树算法有所提升;SGB-RRT*初解、终解和收敛速度仍表现最优.综合初解效果,SGB-RRT*算法效果最好.

3.2.3　地图 3 结果

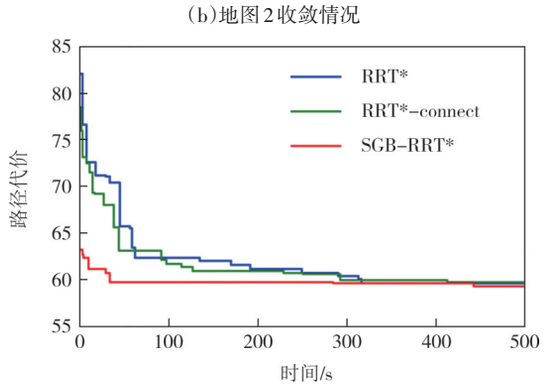
地图 3 中起点坐标为(4,30),终点坐标为(56,10).该地图为多障碍物场景,其初始解的规划结果见图 7(c).由于理论可行解增多,RRT 初解过于曲折;RRT*算法有所优化但因求解时间短而效果不明显;RRT*-connect 加快了求解速度而路径曲折问题仍明显存在;SGB-RRT*算法求解速度快且曲折程度较低.



(a)地图1收敛情况



(b)地图2收敛情况



(c)地图3收敛情况

图8 3种地图下渐进优化算法收敛结果

Fig.8 Convergence results of the progressive optimization algorithms under three maps

地图3下50次独立实验结果展示在表4中,由于可行解数目多、搜索难度下降,各算法初解计算耗时均不高且路径代价相差不大.然而,双树算法在该场景中展现了优势,求解耗时比单树算法降低了一个数量级.整体而言SGB-RRT*算法表现优异.

图8(c)展示了地图3下渐进最优算法的收敛情况.由于该地图理论解更多,RRT*和RRT*-connect算法初解代价均有所增加,而SGB-RRT*算法略有增加,但最终三种方法终解基本收敛至同一水平,其中SGB-RRT*收敛速度最快.

3.3 综合性能分析

综合三种地图表现,本算法在路径代价、路径质量、计算耗时、收敛速度方面均表现优异,以下将详细分析其优越性.

路径代价方面:综合表2~表4数据,RRT、RRT*、RRT*-connect初始解路径代价总体上依次降低,但基本处在同一层次,而SGB-RRT*在三种地图中路径代价均最优且明显低于其他方法.

路径质量方面:由图8可知,RRT、RRT*、RRT*-connect路径曲折且含有大量尖点,无法直接供非完整性转向车辆使用,而结合了运动学曲线的SGB-RRT*算法路径平滑度高且初步满足了非完整性转向车辆的需求.

计算耗时方面:由表2、表3可知,RRT由于逻辑简单无须回溯,计算耗时明显较低,RRT*加入渐进最优设计后,求解耗时和每次运行中百次迭代耗时大幅增加,RRT*-connect进一步设计了双树逻辑,计算耗时较RRT*明显降低,但仍比RRT方法高一个数量级,而SGB-RRT*将初始路径求解耗时和每次运行中百次平均耗时均降低至与RRT同等数量级,甚至在地图2(复杂环境)表现更优.由图8(c)和表4可知,由于地图3可行解增多,四种方法计算耗时无大幅差异,然而SGB-RRT*仍保持了明显优势.

收敛速度方面:综合图8中三种回溯算法运行100 s过程中得到所有路径解的收敛结果,SGB-RRT*算法的收敛速度优势明显,收敛值也表现优异.

表3 地图2算法性能指标(50次均值)

Tab.3 Map 2 algorithm performance indicators (50 times average)

算法	路径代价	求解耗时/s	迭代次数	每次运行中迭代100次耗时/s
RRT	131.219 0	5.354 5	2 000.30	0.267 7
RRT*	123.087 8	49.336 5	1 849.28	2.667 9
RRT*-connect	125.593 2	11.289 9	1 635.74	0.690 2
SGB-RRT*	**116.360 5**	**1.597 1**	**1 216.84**	**0.131 2**

注:最小值加粗,次小值加下画线,路径代价无量纲.

表4　地图3算法性能指标(50次均值)

Tab.4　Map 3 algorithm performance indicators (50 times average)

算法	路径代价	求解耗时/s	迭代次数	每次运行中迭代100次耗时/s
RRT	71.203 2	0.457 3	95.70	0.477 8
RRT*	72.098 1	0.787 9	98.84	0.797 1
RRT*-connect	72.527 4	**0.029 9**	**42.98**	**0.069 6**
SGB-RRT*	**67.524 4**	0.056 6	57.18	0.099 0

注:最小值加粗,次小值加下画线,路径代价无量纲.

4　结 论

为克服RRT及其改进算法存在的盲目采样、引导单一、不满足运动学约束以及重连速度慢等问题,本文提出了一种改进的SGB-RRT*算法.该算法引入了空间探索采样策略、衰减探索与权重控制策略、CSC曲线以及逐级回溯策略.在三种不同类型的地图中对各算法进行了仿真实验,实验数据的分析结果表明,所提出的SGB-RRT*算法在初解路径质量、计算速度以及对运动学约束的满足方面均有较好的表现,验证了该算法的有效性.

然而,本方法基于先验地图信息输入,未深入探索其在动态环境下的表现.Dubins曲线也未考虑倒车因素且曲线平滑度仍有提升空间.未来侧重于动态场景的实时规划和乘坐舒适性等方面的工作.

参考文献

[1] BADUE C, GUIDOLINI R, CARNEIRO R V, et al. Self-driving cars: a survey[J]. Expert Systems with Applications, 2021, 165: 113816.

[2] ZHAO J Y, ZHAO W Y, DENG B, et al. Autonomous driving system: a comprehensive survey [J]. Expert Systems with Applications, 2024, 242: 122836.

[3] REDA M, ONSY A, HAIKAL A Y, et al. Path planning algorithms in the autonomous driving system: a comprehensive review[J]. Robotics and Autonomous Systems, 2024, 174: 104630.

[4] DANIEL K, NASH A, KOENIG S, et al. Theta*: any-angle path planning on grids[J]. Journal of Artificial Intelligence Research, 2010, 39(1): 533-579.

[5] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[6] 秦洪懋,金英杰,杨泽宇,等. 融合模式决策的4WIS车辆路径规划方法[J]. 湖南大学学报(自然科学版),2024,51(8):176-184.
QIN H M, JIN Y J, YANG Z Y, et al. Path planning method integrated with mode decision for 4WIS vehicles[J]. Journal of Hunan University (Natural Sciences), 2024, 51(8): 176-184. (in Chinese)

[7] 周兵,黄治坤,柴天,等.基于图搜索与数值优化方法的分层轨迹规划方法[J].湖南大学学报(自然科学版),2022,49(12):1-10.
ZHOU B, HUANG Z K, CHAI T, et al. Hierarchical trajectory planning method based on graph search and optimization method [J]. Journal of Hunan University (Natural Sciences), 2022, 49(12): 1-10. (in Chinese)

[8] ELBANHAWI M, SIMIC M. Sampling-based robot motion planning: a review[J]. IEEE Access, 2014, 2: 56-77.

[9] ZIEGLER J, WERLING M, SCHRODER J. Navigating car-like robots in unstructured environments using an obstacle sensitive cost function [C]//2008 IEEE Intelligent Vehicles Symposium. June 4-6, 2008, Eindhoven, Netherlands. IEEE, 2008: 787-791.

[10] DOLGOV D, THRUN S, MONTEMERLO M, et al. Path planning for autonomous vehicles in unknown semi-structured environments [J]. International Journal of Robotics Research, 2010, 29(5): 485-501.

[11] CHU K, LEE M, SUNWOO M. Local path planning for off-road autonomous driving with avoidance of static obstacles[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4): 1599-1616.

[12] LAVALLE S M, KUFFNER JR J J. Randomized kinodynamic planning [J]. The International Journal of Robotics Research, 2001, 20(5): 378-400.

[13] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning [J]. International Journal of Robotics Research, 2011, 30(7): 846-894.

[14] KUFFNER J J, LAVALLE S M. RRT-connect: an efficient approach to single-query path planning [C]//IEEE International Conference on Robotics and Automation. Symposia Proceedings. April 24-28, 2000, San Francisco, USA. IEEE, 2002: 995-1001.

[15] 刘依卓. 无人车全局路径规划与局部避障方法研究[D]. 哈尔滨:哈尔滨工程大学,2021.
LIU Y Z. Research on global path planning and local obstacle avoidance methods for unmanned vehicles[D]. Harbin: Harbin Engineering University, 2021. (in Chinese)

[16] KLEMM S, OBERLÄNDER J, HERMANN A, et al. RRT*-connect: faster, asymptotically optimal motion planning [C]// 2015 IEEE International Conference on Robotics and Biomimetics

(ROBIO). December 6-9, 2015, Zhuhai, China. IEEE, 2016: 1670-1677.

[17] 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. September 14-18, 2014, Chicago, USA. IEEE, 2014: 2997-3004.

[18] YUAN C R, ZHANG W Q, LIU G F, et al. A heuristic rapidly-exploring random trees method for manipulator motion planning [J]. IEEE Access, 2020, 8: 900-910.

[19] LIAO B, WAN F Y, HUA Y, et al. F-RRT*: an improved path planning algorithm with improved initial solution and convergence rate[J]. Expert Systems with Applications, 2021, 184: 115457.

[20] FAN J M, CHEN X, WANG Y, et al. UAV trajectory planning in cluttered environments based on PF-RRT* algorithm with goal-biased strategy [J]. Engineering Applications of Artificial Intelligence, 2022, 114: 105182.

[21] YU J X, CHEN C, ARAB A, et al. RDT-RRT: real-time double-tree rapidly-exploring random tree path planning for autonomous vehicles [J]. Expert Systems with Applications, 2024, 240: 122510.

[22] 王东振, 张岳, 赵宇, 等. 基于RRT-Dubins的无人机航迹优化方法[J]. 兵工学报, 2024, 45(8): 2761-2773.
WANG D Z, ZHANG Y, ZHAO Y, et al. A UAV trajectory optimization method based on RRT-Dubins [J]. Acta Armamentarii, 2024, 45(8): 2761-2773. (in Chinese)

[23] 葛超, 张鑫源, 王红, 等. 改进informed-RRT*算法的移动机器人路径规划[J]. 电光与控制, 2025, 32(1): 48-53.
GE C, ZHANG X Y, WANG H, et al. An improved informed-RRT* algorithm for mobile robot path planning [J]. Electronics Optics & Control, 2025, 32(1): 48-53. (in Chinese)

[24] 王蔡琪, 崔西宁, 熊毅, 等. 基于节点到障碍物距离的自适应扩展RRT*路径规划算法[J]. 计算机应用, 2025, 45(3): 920-927.
WANG C Q, CUI X N, XIONG Y, et al. Adaptive extended RRT* path planning algorithm based on node-to-obstacle distance [J]. Journal of Computer Applications, 2025, 45(3): 920-927. (in Chinese)

[25] MASHAYEKHI R, IDRIS M Y I, ANISI M H, et al. Informed RRT* -connect: an asymptotically optimal single-query path planning method[J]. IEEE Access, 2020, 8: 19842-19852.

[26] SALZMAN O, HALPERIN D. Asymptotically near-optimal RRT for fast, high-quality motion planning [J]. IEEE Transactions on Robotics, 2016, 32(3): 473-483.

[27] ZHANG J, ZHANG G Y, PENG Z H, et al. Position-based Dubins-RRT* path planning algorithm for autonomous surface vehicles[J]. Ocean Engineering, 2025, 324: 120702.