

基于改进 Bi-RRT 的移动机器人路径规划算法

崔春雷, 陈诗豪, 沈超航, 李 锋

(广东交通职业技术学院, 广州 510650)

摘要: 双向快速扩展随机树 (Bi-RRT) 算法因采样点的随机性导致在复杂环境中的路径规划存在搜索时间长、采样效率低等问题, 为此提出了一种改进 Bi-RRT 的移动机器人路径规划算法; 算法引入启发式搜索策略, 分别以机器人的起点和终点为中心, 构造了二维高斯分布函数, 并用该概率密度函数约束采样点的生成, 使得越接近目标点的空间采样点出现概率越大, 同时保留部分均匀分布的采样点, 这样采样过程既可以利用目标点的位置信息又保证了算法的概率完备性; 通过算法设计的启发式采样点的引导, 两棵随机树可以快速向着目标区域生长, 降低了搜索的盲目性, 提高了搜索的效率; 仿真结果: 相比于基本 Bi-RRT 算法, 改进算法在复杂环境下规划时间缩短了 43.9%, 扩展节点数目减少了 41.4%, 路径长度优化了 8.1%, 并分析了高斯分布采样点占采样点总数的比值对算法性能的影响。

关键词: 机器人; 路径规划; 双向快速扩展随机树 (Bi-RRT); 目标偏向性; 高斯分布

Path Planning of Mobile Robots Based on Improved Bi-RRT Algorithm

CUI Chunlei, CHEN Shihao, SHEN Chaohang, LI Feng

(Guangdong Communication Polytechnic, Guangzhou 51065, China)

Abstract: Because of the randomness of sampling points, the bidirectional rapidly-exploring random tree algorithm (Bi-RRT) has the problems of long search time and low sampling efficiency of path planning in complex environments. For this reason, an improved Bi-RRT path planning algorithm of mobile robots is proposed. The algorithm introduces a heuristic search strategy to construct a two-dimensional Gaussian distribution density function with the start and end points of robot as the center, and this function is used to constrain the generation of sampling points, so that the closer the target point is, the greater the probability of spatial sampling point occurs. At the same time, some uniformly distributed sampling points are retained, so that the sampling process can not only use the location information of target point, but also ensure the probabilistic completeness of algorithm. Guided by the heuristic sampling points designed by the algorithm, two random trees can quickly grow toward the target area, thereby reduces the search blindness and improves the search efficiency. The simulation results show that, compared with the basic Bi-RRT algorithm, the planning time of improved algorithm in complex environments is shortened by 43.9%, the number of extended nodes is reduced by 41.4%, and the path length is optimized by 8.1%. Finally, the influence on the ratio of the Gaussian distribution sampling points to the total number of sampling points on the performance of algorithm is analyzed.

Keywords: robot; path planning; bidirectional rapid expansion random tree (Bi-RRT); target bias; gaussian distribution

0 引言

随着科技水平的进步, 移动机器人越来越多的被应用到日常生活和工作的多种场景中, 如目标的移动、探测和清洁等。在移动机器人的研究领域中, 路径规划算法属于重点研究方向^[1-3]。移动机器人路径规划问题可定义为: 在一个存在障碍物的空间里, 给定移动机器人的起点和终点, 在遵循时间最短、路径最优以及机器人运动学规律等一系列约束条件下, 找到一条与障碍物无碰撞的路径^[4]。其中环境完全已知的规划问题属于全局路径规划, 环境部分已知的规划问题为局部路径规划^[5]。常见的路径规划算法包括: A* 算法、人工势场法、可视图法、概率路图算法、模拟退火算法、粒子群算法、蚁群算法、遗传算法等^[6-11]。以

上算法往往需要事先对状态空间内的障碍物进行环境建模, 导致计算复杂高, 收敛速度过于缓慢^[12-13]。

LaValle 等人在 1998 年提出了经典的快速搜索随机树 (RRT, rapid-exploring random tree) 算法。RRT 算法的搜索过程通过随机采样的方式把搜索树导向空白区, 通过对空间中的采样点进行碰撞检测, 从而避免了对空间的建模, 该算法具有概率完备、计算量小、效率高等优点, 能有效地解决高维空间以及复杂环境下的路径规划^[14]。然而 RRT 算法也存在一些不足, 如需要在全图进行采样与搜索, 会产生较多不相关的节点, 从而增加了算法对内存的需求, 并且也增加了相应的搜索时间, 导致收敛速率较缓慢^[15]。针对 RRT 算法存在的缺陷, 国内外众多学者纷纷进行了研究, 提出了多种改进的 RRT 算法, 如 C. Urmson 等在文献

收稿日期: 2021-11-05; 修回日期: 2021-12-02。

基金项目: 2020 年广东省科技创新战略专项资金 (pdjh2020b0978); 广东交通职业技术学院大学生科技创新项目 (QKYB0716119); 教育部职业院校信息化教学研究课题 (2018LXA0006)。

作者简介: 崔春雷 (1978-), 男, 河南焦作人, 硕士, 讲师, 主要从事人工智能、通信技术方向的研究。

引用格式: 崔春雷, 陈诗豪, 沈超航, 等. 基于改进 Bi-RRT 的移动机器人路径规划算法 [J]. 计算机测量与控制, 2022, 30(5): 181-185.

[16] 发表了一个基于启发式的偏向算法，它将搜索树向目标点的位置进行了偏移，引导搜索树向目标区域生长，使路径搜索时间进一步优化。针对原始 RRT 算法中搜索得到的路径并非最优的问题，Karaman 等在文献 [17] 中提出了具有渐进最优性的 RRT* 算法，该算法增加了父节点的重选和剪枝两个优化过程，但此算法也存在收敛速度较慢的缺点。为了提高搜索速度，Kuffne 等人先后提出了 RRT-connect 算法^[18]和双向搜索树 (Bi-RRT) 算法^[19]。Bi-RRT 算法从起点和终点同时出发，并行生成两棵 RRT 树，直至两棵树相遇，相较于 RRT 算法，Bi-RRT 算法的收敛速度更快，但 Bi-RRT 算法采用的仍是 RRT 算法的随机节点扩展思想，导致路径搜索过程存在无目标导向性等缺点。

为了解决双向搜索树 (Bi-RRT) 算法在路径搜索时无目标导向性所导致的搜索效率过低的问题，本文提出了一种基于高斯采样的改进 Bi-RRT 算法。该算法在双向搜索的基础上，引入启发式搜索思想，采样点以一定概率以高斯分布的方式被约束在起点与终点周边，从而降低搜索的盲目性，提高了搜索的效率。通过仿真实验表明：在多种类型的复杂环境中，相对于基本的 Bi-RRT 算法，该算法搜索过程中扩展节点更少、收敛速度更快、路径相对更优。

1 Bi-RRT 算法

RRT 算法中随机树的生长缺乏目标导向性，导致大量冗余节点出现，算法的收敛速度较慢，Kuffner 提出的双向 RRT 算法 (Bi-RRT) 则较好的解决了这个问题^[20]。Bi-RRT 算法的主要过程为：以起始点 q_{init} 和目标点 q_{goal} 为根节点分别构建两棵随机搜索树 T_a 和 T_b ，树 T_a 以 q_{init} 为树的根节点进行扩展，树 T_b 以 q_{goal} 为树的根节点进行扩展，直到两棵树相遇。

具体过程如图 1 所示，算法首先在整个搜索空间中生成随机扩展节点 g_{rand} ，然后遍历随机树 T_a 和 T_b ，找出两棵树中距离 q_{rand} 最近的节点并记为 q_{near} ，接着由 q_{near} 出发向 q_{rand} 延伸步长 δ_{step} 得到新节点 q_{new} ，之后对新生成的节点 q_{new} 进行碰撞检测，若检测到 q_{new} 与障碍物碰撞则舍弃该节点，反之将 q_{new} 添加到树中，此时 q_{near} 是 q_{new} 父节点；通过对上述过程进行迭代，使两棵搜索树不断向着对方扩展，直到两棵树各自的 q_{new} 之间的距离小于设定的阈值，此时认为两棵树相遇，路径规划成功。

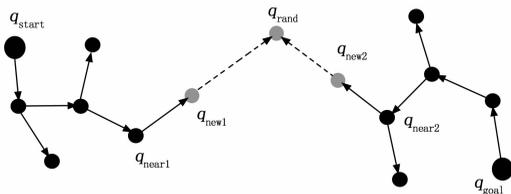


图 1 Bi-RRT 算法节点扩展过程

相对于 RRT 算法，Bi-RRT 算法效率更高搜索速度更快，但是 Bi-RRT 算法采用的仍然是 RRT 算法的随机扩展节点思想，这也导致 Bi-RRT 算法和 RRT 算法一样存在着构型无目标导向性的缺点。

2 基于高斯采样的改进 Bi-RRT 算法

由于 RRT 算法和 Bi-RRT 算法在路径搜索过程中缺乏目标导向性，导致搜索过程的随机性较大，搜索树往往会扩展到远离我们所期待的目标区域的‘无用区域’，浪费了大量的计算资源，需要较长时间才能找到可行路径，且路径的代价往往较大。为此，本文提出了一种基于高斯采样的 Bi-RRT 算法，该算法的核心思想在于引入启发式搜索思想，随机扩展节点 q_{rand} 不再以均匀分布的形式在搜索空间内随机出现，而是以一定概率以高斯分布的方式出现在目标点周边，这样搜索过程不但可以引导搜索树尽可能朝着目标区域前进，同时也保留了 RRT 算法的空间搜索能力，算法不仅提高了搜索效率，得到的路径也相对更优。

2.1 二维高斯分布性质分析

若二维随机变量 (X, Y) 服从二维正态 (高斯) 分布，即： $(X, Y) \sim N(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$ ，则其概率密度函数为^[21]：

$$f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2(1-\rho^2)} \left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right]} \quad (1)$$

其中： μ_1, σ_1 为分量 X 的期望值与标准差， μ_2, σ_2 为分量 Y 的期望值与标准差， ρ 值决定了 X 和 Y 的线性相关程度。这里假定 $\mu_1 = \mu_2 = 0, \sigma_1 = \sigma_2 = 5$ ，来观察 ρ 取不同值时二维随机变量 (X, Y) 概率密度函数的图像。

由图 2 可以看出，在限定 $\sigma_1 = \sigma_2$ 的前提下，二维高斯分布的概率密度函数图像的形状主要和 ρ 值有关，当 ρ 取值较小，如 $\rho = 0.4$ 时，随机变量 X 与 Y 的相关度较小，此时概率密度函数图像的等高线较圆；而当 ρ 取值较大，如 $\rho = 0.8$ 时，随机变量 X 与 Y 的相关度较大，此时概率密度函数图像的等高线较扁长。另外，当 $\sigma_1 = \sigma_2$ 时，二维高斯分布的概率密度图像的等高线的长轴方向显然和坐标系的横轴方向成 45° 夹角，设这个角度为 θ_1 ，即 $\theta_1 = \frac{\pi}{4}$ 。

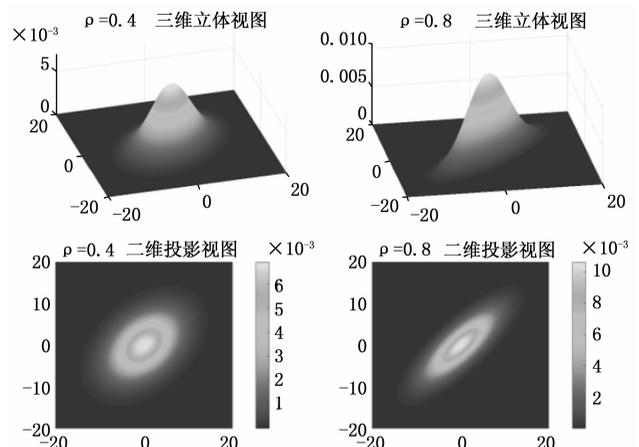


图 2 ρ 值与二维高斯分布概率密度函数图像关系

利用二维高斯分布的上述特性，我们引入启发式搜索策略，随机扩展采样点 q_{rand} 不再以均匀分布的形式出现搜索空间内，而是被二维高斯分布函数约束在起点与终点周边区域，并引导搜索树朝着该区域方向生长。采用这种策略，在越接近目标点的空间，采样点 q_{rand} 的出现概率越大，但是

又不会把概率完全锁死在目标点本身, 这样不但可以启发随机搜索树向着目标区域生长, 提高搜索的效率, 同时又能以一定的概率绕过障碍物。

2.2 高斯分布随机采样点的生成方法

本文算法采用双采样点策略, 即用采样点 q_{rand1} 来引导随机树 T_b 的生长, 用采样点 q_{rand2} 来引导随机树 T_a 的生长, 而 q_{rand1} 和 q_{rand2} 则分别由以起点 q_{init} 和终点 q_{goal} 为中心的两个二维高斯分布函数来分别约束其生成。

假定机器人起点位于搜索空间中的点 $q_{init}(x_1, y_1)$, 目标点为 $q_{goal}(x_2, y_2)$, 则这两个点的长度为 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。设向量 $\vec{q_{init}q_{goal}}$ 与坐标系横轴的夹角为 θ_2 , 这里 $\theta_2 = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$ 。

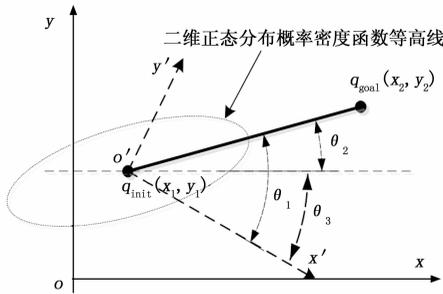


图 3 坐标系转换示意图

对于随机扩展点 q_{rand1} , 为了方便分析, 我们对坐标系进行变换, 如图 3 所示把原坐标系 OXY 变换为新的坐标系 $O'X'Y'$, 此时 q_{init} 与原点 O' 重合, 设 $\vec{q_{init}q_{goal}}$ 与 $\vec{O'X'}$ 的夹角 $\theta_1 = \frac{\pi}{4}$, 由前述分析可知, 若 $\rho > 0, \mu_1 = \mu_2 = 0, \sigma_1 = \sigma_2$, 则二维正态 (高斯) 分布概率密度函数的等高线图像的长轴方向与 $\vec{q_{init}q_{goal}}$ 正好重合, 此时 $\theta_3 = \theta_1 - \theta_2$ 。

在坐标系 $O'X'Y'$ 中, 以 q_{init} 为中心点, 生成随机扩展点 $q_{rand1}(x'_{rand1}, y'_{rand1})$, 该点的坐标 (x'_{rand1}, y'_{rand1}) 符合以 O' 为分布中心的二维高斯分布, 即其概率由公式 (1) 确定。此时公式 (1) 中参数为 $\mu_1 = \mu_2 = 0, \sigma_1 = \sigma_2 = \lambda * d$, 这里 d 为 $\vec{q_{init}q_{goal}}$ 长度, λ 为系数。

按照公式 (2), 我们把上述生成的随机扩展点 $q_{rand1}(x'_{rand1}, y'_{rand1})$ 再变换到坐标系 OXY 中, 得到 $q_{rand1}(x_{rand1}, y_{rand1})$ 。

$$\begin{bmatrix} x_{rand1} \\ y_{rand1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) \end{bmatrix} \begin{bmatrix} x'_{rand1} \\ y'_{rand1} \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (2)$$

其中式 (2) 中的 (x_{rand1}, y_{rand1}) 为 q_{rand1} 在坐标系 OXY 中的坐标。

按相同方法, 我们可以生成以 q_{goal} 中心点随机采样点 $q_{rand2}(x_{rand2}, y_{rand2})$ 。

由图 4 可以看出随机采样点 q_{rand1} 主要分布在起点附近区域, 且越接近点, 其出现的概率越大, q_{rand2} 则主要分布在终点 q_{goal} 附近区域, 越接近点 q_{goal} , 其出现的概率也越大。

2.3 改进 Bi-RRT 算法实现

算法 1 给出了改进 Bi-RRT 算法的伪代码, 算法过程如

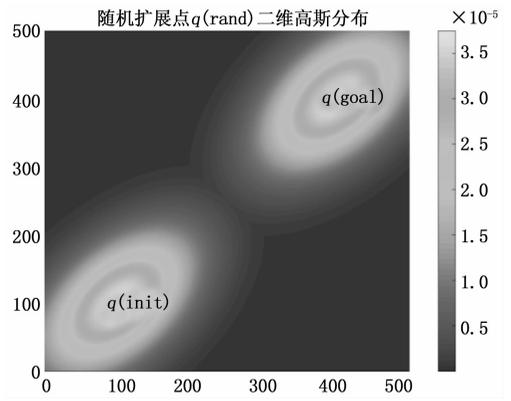


图 4 随机采样点的概率分布示意图

下: 首先以起点 q_{init} 和终点 q_{goal} 为根节点分别构建随机树 T_a 和 T_b (第 1-2 行); 然后先以 q_{goal} 为目标通过算法 2 生成随机点 q_{rand_b} 用来引导 T_a 的生长, 找到随机树 T_a 中离 q_{rand_b} 最近的点 q_{near_a} , 以 q_{near_a} 为起点向 q_{rand_b} 方向延伸步长 δ_{step} 生成叶子节点 q_{new_a} , 判断 q_{near_a} 与 q_{new_a} 之间是否存在障碍物, 若不存在障碍物则将 q_{new_a} 作为 T_a 的子节点, 并添加边 (q_{near_a}, q_{new_a}) (第 4-9 行); 接着以 q_{init} 为目标通过算法 2 生成随机点 q_{rand_a} 用来引导 T_b 的生长, 按照相同的规则生成叶子节点 q_{new_b} 和边 (q_{near_b}, q_{new_b}) (第 10-15 行); 最后如果 q_{new_a} 和 q_{new_b} 之间的距离小于设定的阈值 s , 且两者之间没有障碍物, 则判定两棵树相互连接, 路径规划完成 (第 16-17 行)。

算法 1: 改进 Bi-RRT 算法伪代码

- 1) $V_a \leftarrow \{q_{init}\}; E_a \leftarrow \Phi; T_a \leftarrow (V_a, E_a);$
- 2) $V_b \leftarrow \{q_{goal}\}; E_b \leftarrow \Phi; T_b \leftarrow (V_b, E_b);$
- 3) for $i = 1$ to N do;
- 4) $q_{rand_b} \leftarrow \text{sample}(q_{goal});$
- 5) $q_{near_a} \leftarrow \text{Nearest}(T_a, q_{rand_b});$
- 6) $q_{new_a} \leftarrow \text{Steer}(q_{near_a}, q_{rand_b});$
- 7) if $\text{ObstacleFree}(q_{new_a}, q_{near_a})$ then
- 8) $V_a \leftarrow V_a \cup \{q_{new_a}\};$
- 9) $E_a \leftarrow E_a \cup \{(q_{near_a}, q_{new_a})\};$
- 10) $q_{rand_a} \leftarrow \text{Sample}(q_{init});$
- 11) $q_{near_b} \leftarrow \text{Nearest}(T_b, q_{rand_a});$
- 12) $q_{new_b} \leftarrow \text{Steer}(q_{near_b}, q_{rand_a});$
- 13) if $\text{ObstacleFree}(q_{new_b}, q_{near_b})$ then;
- 14) $V_b \leftarrow V_b \cup \{q_{new_b}\};$
- 15) $E_b \leftarrow E_b \cup \{(q_{near_b}, q_{new_b})\};$
- 16) if $(\text{Distance}(q_{new_a}, q_{new_b}) < s \cap) \dots \text{ObstacleFree}(q_{new_a}, q_{new_b})$ then
- 17) Return (T_a, T_b) 。

为了平衡搜索过程中的随机性与目标导向性, 随机点 q_{rand} 的产生由 3 种机制共同决定, 见公式 (3)。设置两个目标偏置概率 $p_{target1}$ 和 $p_{target2}$, 且满足 $0 < p_{target1} < p_{target2} < 1$ 。

$$q_{rand} = \begin{cases} q_{\text{gaussian_sample}}, & P \leq p_{target1} \\ q_{\text{init}} \text{ (或 } q_{\text{goal}}), & p \geq p_{target2} \\ q_{\text{uniform_sample}}, & \text{else} \end{cases} \quad (3)$$

公式 (3) 中 p 为 0~1 之间的一个随机数; 当 $p \leq p_{target1}$ 时, q_{rand} 由 2.2 中的方法产生, 即随机点 q_{rand} 满足二维高斯分布特性; 当 $p \geq p_{target2}$ 时, 随机采样点 q_{rand} 为目标点 q_{init} 或 q_{goal} 本身, 这样可以充分利用目标点的信息, 使得搜索树以一定概率朝着目标点方向快速前进; 当 $p_{target1} < p < p_{target2}$ 时, q_{rand} 符合均匀分布, 这样可以一部分随机点保留采样时的随机性, 确保搜索过程在概率上的完备性, 保障搜索树能跳出局部最优; 算法 2 给出了公式 (3) 的伪代码。

算法 2: 随机点生成的伪代码

```

P_rand() //生成 0~1 之间的随机数 p if p ≤ p_target1
q_rand = q_target. //q_target 可取 q_init 或 q_goal else if p > p_target2
q_rand = q_uniform_Sample; //此时 q_rand 按均匀分布 else
q_rand = q_gaussian_Sample; //此时 q_rand 按二维高斯分布 end if return(q_rand)
    
```

3 仿真与分析

为了验证算法的有效性, 在配置 window10 系统, 主频 3.40 GHz, 内存 16 GB 的 PC 机上, 采用 Matlab 2020a 对算法进行了编程仿真实验。仿真时设置的参数如下: 仿真空间尺寸为 500×500 , 起点 q_{init} 坐标为 $[1, 1]$, 终点 q_{goal} 坐标为 $[500, 500]$, 步长 $\delta_{step} = 15$, 两树之间的距离阈值 $s = 30$, 目标偏置概率 $p_{target1} = 0.6, p_{target2} = 0.9$, 二维高斯分布的参数为 $\sigma_1 = \sigma_2 = 0.25 * d, \rho = 0.5$ 。

3.1 算法性能分析

仿真时按照障碍物的分布类型设置了 3 种有代表性的环境: 简单环境、复杂环境、迷宫环境。分别在每一种环境下对 Bi-RRT 算法和基于高斯采样的改进 Bi-RRT 算法的性能进行对比。具体过程为: 在每一种环境下, 分别对两种算法各进行 50 次仿真测试, 并使用路径长度 (L)、扩展节点数目 (n)、路径规划时间 (t) 这 3 个参数作为性能指标, 并求得这 3 个指标的 50 次仿真测试结果的均值, 最后通过这 3 个指标对算法性能进行定量分析。

图 5 为简单环境中两种算法的表现, 图 6 为充满障碍物的复杂环境中两种算法的表现, 图 7 为通道狭窄曲折的迷宫环境下两种算法的表现。从图 5~7 可以明显看出, 基本 Bi-RRT 算法采样过程因为缺乏目标导向, 采样点分布过于随机, 导致搜索树往往会在“无用区域”浪费过多资源, 产生过多的无用节点, 而本文改进算法引入启发式搜索思想, 充分利用了目标点的位置信息, 让随机点不再“盲目”出现, 而是以高斯分布的形式出现在目标点附近的区域, 搜索过程所需的随机采样点的数量更少, 效率也更高。

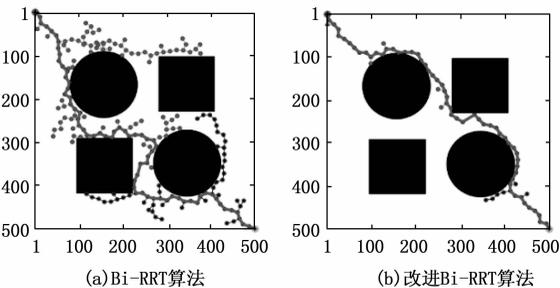


图 5 环境 1: 简单环境

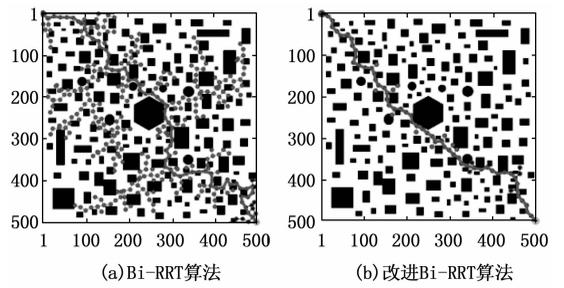


图 6 环境 2: 复杂环境

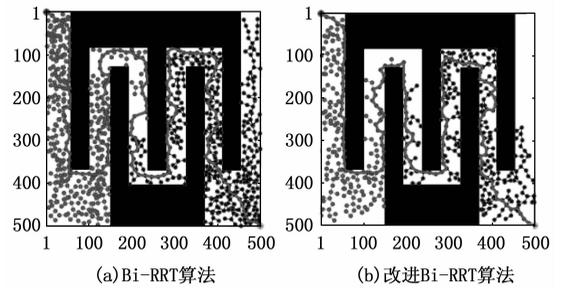


图 7 环境 3: 迷宫环境

如表 1 所示, 对图 5 这种简单环境, 本文算法的额外的扩展节点更少, 路径也更平滑, 路径长度也更小。对图 6 这种充满密集障碍物的复杂环境, 本文算法的优势更加明显, 相对于基本 Bi-RRT 算法, 平均规划时间缩短了 43.9%, 平均扩展节点数目减少了 41.4%, 路径长度优化了 8.1%。对图 7 这种富有挑战性的充满狭窄通道的迷宫环境, 相对于基本 Bi-RRT 算法, 改进算法的平均规划时间缩短了 30.9%, 平均扩展节点数目减少了 27.2%, 路径长度优化了 2%。通过定量分析可知, 本文提出的改进 Bi-RRT 算法的路径搜索时间更短、扩展节点更少、路径更优。

表 1 不同环境下算法性能对比

环境	算法	扩展节点数目	路径长度	规划时间/s
环境 1	Bi-RRT	119	955.4	7.74
	本文算法	92	910.6	6.87
环境 2	Bi-RRT	162	980.7	11.48
	本文算法	95	901.1	6.44
环境 3	Bi-RRT	788	2254.4	50.82
	本文算法	574	2211.3	35.11

3.2 目标偏置概率对算法性能的影响

改进算法中目标偏置概率 $p_{target1}, p_{target2}$ 对算法的运行效率有着重要影响。由公式 (3) 可知采样点由 3 种机制共同产生, 分别在概率 $p_1 = p_{target1}$ 下按二维高斯分布出现, 在概率 $p_2 = p_{target2} - p_{target1}$ 下按均匀分布出现, 以概率 $p_3 = 1 - p_{target2}$ 把起点 (或终点) 作为采样点, 显然 $p_1 + p_2 + p_3 = 1$ 。为了简化问题, 保持 $p_3 = 0.1$ 不变, 此时 $p_2 = 0.9 - p_1$, 这里 p_1 显然表示的是按高斯分布出现的采样点占总的采样点数的百分比。下面来分析概率 p_1 的值对算法性能的影响, 这里仍然以图 6 中的复杂环境作为测试环境, 在保持其他仿真参数不变的情况下, 测试 p_1 取不同值时算法性能的表现。

图 8 为仿真测试得到的平均节点数目随概率 p_1 变化的曲线, 从图中可以看出, 平均节点数目随着概率 p_1 的增加而减少, 当 $p_1=0.7$ 附近时达到极小值。图 9 则为平均规划时间随概率 p_1 变化的曲线, 与图 8 的规律一致, 这里规划时间也随着概率 p_1 的增加而减少, 当 $p_1=0.7$ 时规划时间达到最小值, 之后随着 p_1 的增加规划时间又有所上升。可见, 概率 p_1 较小时, 即按照高斯分布出现的采样点占比小时, 算法效率会降低; 另一方面概率 p_1 如果取到 0.9, 即所有随机点都按照高斯采样时, 则又因为缺少了自由采样带来的随机性, 算法效率也会降低, 只有当 p_1 取到合适值时, 算法的效率才最高。

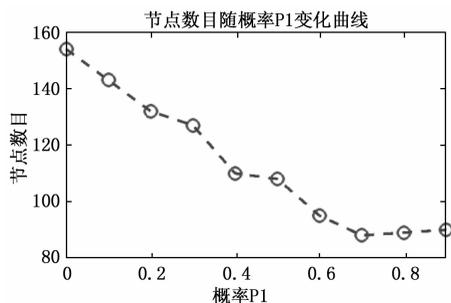


图 8 平均节点数目随概率变化曲线

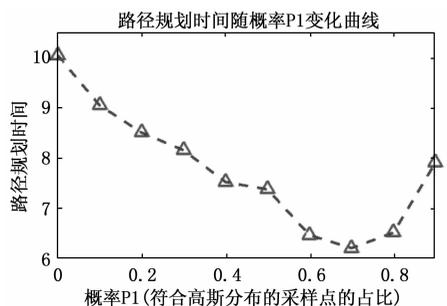


图 9 平均规划时间随概率变化曲线

4 结束语

为了改进 Bi-RRT 算法的效率, 本文提出了一种改进 Bi-RRT 算法, 该算法分别以机器人的起点和终点为概率分布的中心, 构建二维高斯分布概率密度函数, 并用该函数约束随机采样点的生成, 同时也保留一部分均匀分布的采样点, 通过这些具有目标启发性的采样点来引导两棵搜索树快速向目标点生长并相遇。该算法在保证搜索过程概率完备性的前提下, 提高了寻径的效率和质量, 相对于基本的 Bi-RRT 算法, 本文算法在应对复杂环境、迷宫环境等环境时的表现更佳, 如在复杂环境下规划时间缩短了 43.9%, 扩展节点数目减少了 41.4%, 路径长度优化了 8.1%, 最后分析了目标偏置概率对算法性能的影响, 找到了使算法效果最优时对应的高斯分布采样点的占比。未来可以考虑将本文算法结合 RRT* 算法, 并应用到 3 维以上的高维空间中的路径规划问题。

参考文献:

[1] MOHANTY P K, PARHI D R. Optimal path planning for a

mobile robot using cuckoo search algorithm [J]. Journal of experimental & theoretical artificial intelligence, 2016, 28 (1/2): 35-52.

- [2] 张启飞, 郭太良. 基于多阶段决策的机器人全局路径规划算法 [J]. 计算机工程, 2016, 42 (10): 296-302.
- [3] 王小伟. 室内自主导航轮式机器人路径规划策略研究 [D]. 合肥: 合肥工业大学, 2018.
- [4] 杨也, 倪建军, 陈一楠, 等. 改进 RRT* 的室内机器人路径规划算法 [J]. 计算机测量与控制, 2020, 28 (1): 241-245.
- [5] 魏武, 韩进, 李艳杰, 等. 基于双树 Quick-RRT 算法的移动机器人路径规划 [J]. 华南理工大学学报 (自然科学版), 2021, 49 (7): 51-58.
- [6] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots [C] // Proceedings of 1985 IEEE International Conference on Robotics and Automation, St. Louis: IEEE, 1985: 500-505.
- [7] LOZANO-PEREZ T, WESLEY M A. An algorithm for planning collision-free paths among polyhedral obstacles [J]. Communications of the ACM, 1979, 22 (10): 560-570.
- [8] KAVRAKI L E, SVETSKA P, LATOMBE J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces [J]. IEEE Transactions on Robotics and Automation, 1996, 12 (4): 566-580.
- [9] 韩明, 刘教民, 吴朔媚, 等. 粒子群优化的移动机器人路径规划算法 [J]. 计算机应用, 2017, 37 (8): 2258-2263.
- [10] HU Y M, LIU W M. Solving of optimal path problem based on improved ant colony algorithm [J]. Journal of South China University of Technology (Natural Science Edition), 2010, 38 (10): 105-110.
- [11] 张好剑, 苏婷婷, 吴少泓, 等. 基于改进遗传算法的并联机器人分枝路径优化 [J]. 华南理工大学学报 (自然科学版), 2017, 45 (10): 93-99.
- [12] 张丹萌, 甄子洋, 陈核. 基于改进 RRT-Connect 的协同航迹规划 [J]. 电光与控制, 2021, 28 (9): 25-29.
- [13] 马向华, 张谦. 改进蚁群算法在机器人路径规划上的研究 [J]. 计算机工程与应用, 2021, 57 (5): 210-215.
- [14] LAVALLE S M. Rapidly-exploring random trees: a new tool for path planning [R]. Ames: Iowa State University, 1998.
- [15] 宋林忆, 严华. 一种基于改进 RRT* 的移动机器人的路径规划算法 [J]. 现代计算机, 2020 (7): 3-8, 17.
- [16] URMSON C, SIMMONS R G. Approaches for Heuristically Biasing RRT Growth [C] // IROS, 2003, 2: 1178-1183.
- [17] KARAMAN S, FRAZZOLI E. Optimal kinodynamic motion planning using incremental sampling-based methods [C] // IEEE Conference on Decision and Control, IEEE, 2010: 7681-7687.
- [18] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query pathplanning [C] // IEEE International Conference on Robotics and Automation, 2000, Proceedings, ICRA, IEEE, 2002: 995-1001.
- [19] LAVALLE S M, KUFFNER J J. Randomized Kinodynamic Planning [J]. The International Journal of Robotics Research, 2001, 20 (5): 378.
- [20] 韩丰键, 邱书波, 冯超, 等. 基于目标导向的双向 RRT 路径规划算法 [J]. 齐鲁工业大学学报, 2021, 35 (1): 35-43.
- [21] 邹云蕾. 二维正态随机变量的线性组合的独立性 [J]. 教育教学论坛, 2020 (15): 279-280.