

# 应用改进 RRT\_Connect 算法的移动机器人路径规划

史小旭<sup>1</sup>, 张烈平<sup>1\*</sup>, 唐柳<sup>2</sup>, 董路熙<sup>1</sup>, 彭建盛<sup>3</sup>

1. 广西高校先进制造与自动化技术重点实验室, 桂林 541006

2. 贺州学院人工智能学院, 贺州 542899

3. 广西高校人工智能与信息处理重点实验室, 河池 546300

**摘要** 针对复杂未知环境下的移动机器人路径规划问题, 提出一种优化已搜索节点和已规划部分路径的改进 RRT\_Connect 算法。算法引入了终点和已搜索节点目标偏向策略, 该策略在随机采样函数中引入终点和已搜索节点偏向概率基准值, 使随机采样点按随机概率设定为终点或已搜索节点; 通过筛选有效新节点和一定范围内邻节点的父节点, 优化路径规划成本, 使规划路径趋于平滑。仿真结果表明, 提出的改进 RRT\_Connect 算法所规划的路径在平均转弯次数、平均规划路径长度和平均规划成功率等指标较改进前更优。

**关键词** RRT\_Connect 算法; 移动机器人; 路径规划; 目标偏向策略

近年来, 路径规划技术正逐渐被应用到智能物流、智能家居、海上搜救、邮轮等领域<sup>[1]</sup>。路径规划就是在遵循一定的评价指标函数的前提下, 让机器人在充满障碍物的环境中顺利避开障碍物, 从而规划出一条从起始位置到目标位置的最优路径<sup>[2]</sup>。根据目标范围的不同, 规划可以分为全局路径规划和局部路径规划<sup>[3]</sup>。全局路径规划是指依据已有的全局环境信息设计一个由起始点至终点的可行路线<sup>[4]</sup>。而局部路径规划则是指在部分环境信息未知

的情形下, 使用当前环境信息设计一条无碰撞路线<sup>[5]</sup>。全局路径规划主要有基于图搜索的 Dijkstra、A\*、D\* 算法和基于采样的概率地图法 (probabilistic roadmap method, PRM)、快速探索随机树 (rapidly-exploring random tree, RRT) 算法等<sup>[6]</sup>。RRT 算法与 A\* 算法相比, 不需要对空间进行栅格化处理, 无需对环境进行结构化建模, 适合在高维空间进行路径规划, 缺点是搜索效率不高、规划路径容易曲折且不是最优路径, 通常需要路径优化<sup>[7-8]</sup>。

收稿日期: 2022-10-09; 修回日期: 2022-10-31

基金项目: 国家自然科学基金项目 (61741303); 广西空间信息与测绘重点实验室基金项目 (19-185-10-08)

作者简介: 史小旭, 硕士研究生, 研究方向为智能信息处理技术, 电子信箱: 2975868194@qq.com; 张烈平 (通信作者), 教授, 研究方向为传感器与智能信息处理技术, 电子信箱: zlp\_gx\_gl@163.com

引用格式: 史小旭, 张烈平, 唐柳, 等. 应用改进 RRT\_Connect 算法的移动机器人路径规划[J]. 科技导报, 2024, 42(8): 111-119;

doi: 10.3981/j.issn.1000-7857.2022.10.01517

针对 RRT 算法的缺点, Lavelle 等<sup>[9]</sup>提出了 RRT\_Connect 算法。该算法在开始和结束配置中, 递增地生长 2 个快速探索随机树, 找到有效路径的时间显著减少, 但依然没有解决采样范围大的问题<sup>[10]</sup>。近年来, 许多研究人员对其进行了改进。王海芳等<sup>[11]</sup>提出一种基于贝塞尔插值方法的改进固定点数渐近最优快速扩展随机数 (rapidly-exploring random trees star fixed nodes, RRT\*FN) 算法, 改进算法采用一定概率进行随机点的目标偏置选择, 并利用贝塞尔插值方法平滑路径。赵文龙等<sup>[12]</sup>引入一种全新的目标偏向采样策略, 在均匀采样得到的随机点与目标点的连线上随机取一个点作为新的随机点进行树的扩展, 极大地提升了原算法的规划效率。Fan 等<sup>[13]</sup>提出了一种改进的 RRT 和人工势场法 (artificial potential field, APF) 融合算法。首先用改进的 RRT 算法规划全局路径, 然后用 APF 算法对每条路径进行优化, 提高了实时性, 解决了局部最优问题。Mei 等<sup>[14]</sup>提出了优化的快速探索随机树 A\* (optimized rapidly exploring random trees A\*, ORRT-A\*) 方法, 通过引入额外的步长来加速朝向目标的树的生成, 然后应用 A\* 来获得最短路径。Sun 等<sup>[15]</sup>提出了一种基于广义 Voronoi 图和 RRT\_Connect 的安全连接 RRT (safe-Connect RRT, SC-RRT) 算法。SC-RRT 算法改进了 RRT\_Connect 算法的分支增长, 减少了采样空间, 从而可以快速获得远离障碍物的路径。Li 等<sup>[16]</sup>提出了一种自适应 RRT\_Connect 方法, 它可以很好地处理狭窄通道环境。

综上所述, 目前基于 RRT\_Connect 的改进路径规划算法都取得了一定的成果, 但仍存在随机采样搜索平均、效率低下、规划路径质量不高等问题。针对这些问题, 本研究在算法中引入终点和已搜索节点偏向策略, 结合已搜索节点和已规划部分路径的优化提出一种改进 RRT\_Connect 路径规划算法, 并仿真实验验证改进 RRT\_Connect 算法的有效性。

## 1 RRT\_Connect 算法

由于 RRT 算法以随机方式生成采样点, 搜索

过程盲目性较大, 算法收敛速度慢<sup>[17]</sup>。为了提高规划效率, Lavelle 等<sup>[9]</sup>提出的 RRT\_Connect 算法利用双向搜索策略替代传统 RRT 算法的完全随机探索。当起始点和终点都处于被障碍物包围的约束空间时, 这种具有启发性的扩展方式使得两棵搜索树更快地朝对方生长, 从而远离各自的约束空间, 极大地缩短了算法收敛时间<sup>[18]</sup>。RRT\_Connect 算法扩展过程如图 1 所示。

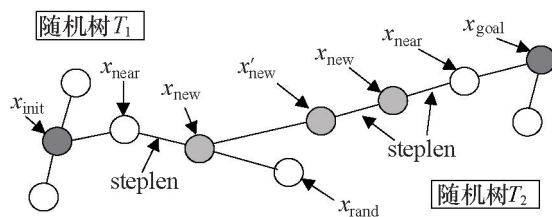


图1 RRT\_Connect 算法扩展过程

图 1 中,  $x_{init}$  为路径规划起始点;  $x_{rand}$  为随机采样点;  $x_{near}$  为搜索树中与距离最近的节点;  $x_{new}$  为以  $x_{near}$  为父节点沿着固定搜索步长  $steplen$  扩展的新节点。一次搜索扩展包括 2 个步骤: 随机扩展和贪婪扩展。随机扩展在随机树  $T_1$  进行, 贪婪扩展在随机树  $T_2$  进行。随机扩展与 RRT 算法扩展的过程相同。在贪婪扩展中, 设计了启发式搜索函数。  $T_2$  以终点  $x_{goal}$  和  $T_1$  生成的  $x_{new}$  新节点之间的角度为方向, 用与  $T_1$  一样的固定步长  $steplen$  生成自己的新节点  $x'_{new}$ ,  $T_2$  将  $x'_{new}$  节点作为父节点, 不停地朝着  $T_1$  的  $x_{new}$  节点生成新节点, 并把新节点的数据覆写至  $x'_{new}$  节点里, 直到节点碰到障碍物或成功连接另一棵树的节点,  $T_2$  扩展结束。如果 2 个随机树在贪婪扩展后未成功连接, 则转换 2 棵随机树的扩展方式, 进行下一次的搜索扩展。

## 2 RRT\_Connect 算法的改进

### 2.1 随机采样函数改进

在路径规划中, 随机采样方法存在搜索平均、效率低下的问题。RRT\_Connect 算法尽管采样 2 棵随机搜索树双向生长策略提高搜索效率, 但是易导致不必要节点的生成, 使规划路径绕远。针对上述

RRT\_Connect算法在随机采样过程中所面临的问题,在目标偏向思想的启发下,采用终点和已搜索节点偏向策略,充分利用搜索树随机扩展过程中的节点信息和已经规划的部分路径。通过引入终点偏向概率基准值 Goalprob 和已搜索节点偏向概率基准值 OldNodeprob,修改 SampleNode()随机采样函数,提升算法收敛性能。基于终点和已搜索节点偏向策略的随机采样流程如下。

Step 1: 函数开始运行时,生成一个[0,1]随机实数并赋值给概率 $p$ 。标签的值为[-1,0]范围内的一个随机实数。

Step 2: 对概率 $p$ 进行判断,如果概率 $p$ 大于0且小于等于终点偏向概率基准值 Goalprob,则随机采样点 $x_{rand}$ 设定为终点 $x_{goal}$ ,搜索树选择距离终点最近的 $x_{near}$ 为父节点来生成 $x_{new}$ 。

Step 3: 如果概率 $p$ 大于终点偏向概率基准值且小于等于已搜索节点偏向概率基准值 OldNodeprob 与 Goalprob 的和,同时已搜索节点缓存集不为空,则随机采样点 $x_{rand}$ 设定为已搜索节点缓存集里对应标签 $i$ 的节点, $x_{new}$ 的父节点为距离已搜索节点最近的 $x_{near}$ 。如果已搜索节点缓存集为空,说明算法正在初始化或者尚未进行随机扩展,随机采样点依然设定为终点 $x_{goal}$ 。

Step 4: 如果概率 $p$ 大于 OldNodeprob 与 Goalprob 的和且小于等于1,则随机采样点 $x_{rand}$ 通过 SampleNode()函数在有限状态空间中随机产生。

## 2.2 已搜索节点和已规划部分路径的改进

在对随机采样函数的改进中,因为采样随机性,得到的规划路径不是最优的。根据 RRT\_Connect算法原理可知,算法规划的路径是相邻节点之间连线构成的边集,这些节点就是算法迭代生成的点集,也是改进随机采样函数利用的已搜索节点。为了提高已搜索节点和已规划部分路径的质量,对父节点的选取进行优化并重新布线,使其逐渐趋于渐进最优。改进 RRT\_Connect算法以路径代价之和作为父节点选取流程的判断条件<sup>[19]</sup>,若路径代价之和小于 $Cost(x_{nebr})$ ,则说明 $x_{nebr}$ 现在的父节点与 $x_{new}$ 作为父节点相比,所规划路径存在绕远现象,删除 $x_{nebr}$ 与其父节点连接的边,并将 $x_{nebr}$ 的父节点替换为

$x_{new}$ ,然后为随机树重新布线,减小路径代价。 $x_{new}$ 父节点优选思路和 $x_{nebr}$ 类似,都是为了优化选择父节点,减少冗余节点的产生,使路径趋于渐进最优。 $x_{nebr}$ 为 $x_{new}$ 附近的所有已搜索节点,优化已搜索节点 $x_{nebr}$ 父节点选取流程如图2所示。优化 $x_{new}$ 父节点选取流程如图3所示。

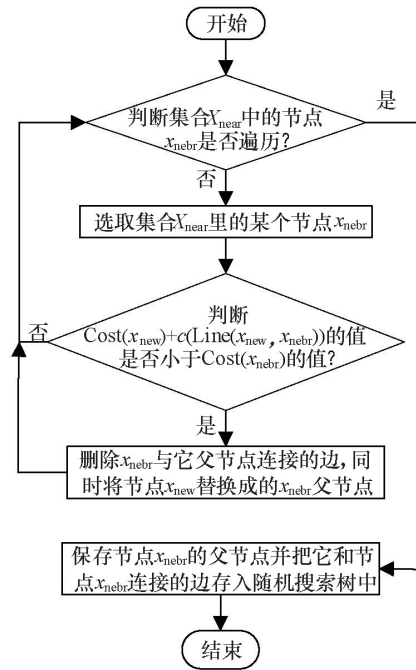
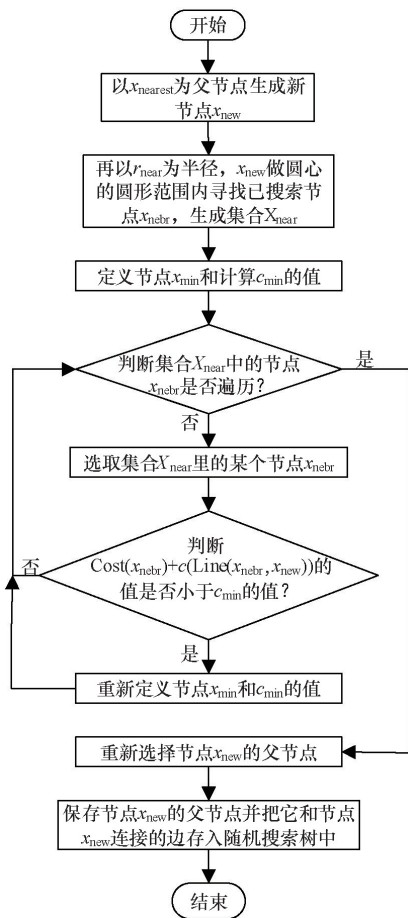


图2 优化已搜索节点 $x_{nebr}$ 父节点选取流程

其中, $x_{nearest}$ 是距离采样点 $x_{rand}$ 最近的节点, $r_{near}$ 是以 $x_{new}$ 为圆心的圆的半径, $X_{near}$ 是 $x_{nebr}$ 的集合。路径代价函数 $Cost()$ 的返回值为设定节点 $x$ 从父节点一直回溯到根节点的路径长度,距离计算函数 $Line()$ 的返回值为2个设定节点之间的欧几里得距离,代价函数 $c$ 代表所有非平凡无碰撞路径的严格正代价。 $x_{min}$ 为初始定义为节点 $x_{nearest}$ , $c_{min}$ 为节点 $x_{nearest}$ 到节点 $x_{new}$ 的欧式距离以及节点 $x_{nearest}$ 回溯到根节点的路径长度之和。

## 2.3 改进 RRT\_Connect算法

尽管在许多问题上,RRT\_Connect以更少的迭代来寻找解决方案,但它的输出仍然像RRT一样不是最优的。为了使路径趋于渐进最优解,提出一种优化已搜索节点和已规划部分路径的改进 RRT\_Connect算法。改进 RRT\_Connect算法步骤如下。

图3 优化 $x_{new}$ 父节点选取流程

Step 1: 对算法参数进行初始化,将路径规划起始点 $x_{init}$ 作为随机搜索树 $T_1$ 的根节点,随机搜索树 $T_2$ 的根节点设为 $x_{goal}$ 终点,然后进行第 $k$ 次迭代,由改进随机采样函数生成随机采样点 $x_{rand}$ 。

Step 2: 在随机树 $T_1$ 上选择距离采样点 $x_{rand}$ 最近的 $x_{nearest}$ 节点,并以 $x_{nearest}$ 为父节点生成新节点 $x_{new}$ 。如果 $x_{new}$ 节点存在且不与有限状态空间里的障碍物发生碰撞,则以 $x_{new}$ 为圆心取一个半径为 $r_{near}$ 的圆。在这个圆形区域内,寻找 $x_{new}$ 附近的所有已搜索节点 $x_{nebr}$ ,并将这些点的集合存放于 $X_{near}$ 中。重选父节点圆形区域的半径 $r_{near}$ 在算法刚开始时为固定搜索步长 $steplen$ ,随着算法不断地进行迭代扩展,半径 $r_{near}$ 的值随着随机树已搜索节点 $x_{nebr}$ 的增多而逐渐减小。

Step 3: 生成集合 $X_{near}$ 之后,在集合 $X_{near}$ 中选

取 $x_{new}$ 的最优父节点。首先选取集合 $X_{near}$ 里的某一个已搜索节点 $x_{nebr}$ ,如果已搜索节点 $x_{nebr}$ 与 $x_{new}$ 之间的连线不与障碍物发生碰撞,则计算从节点 $x_{nebr}$ 回溯到根节点的路径长度以及节点 $x_{nebr}$ 到 $x_{new}$ 的欧式距离之和,并将其与 $c_{min}$ 进行大小比较。如果比 $c_{min}$ 小,就把这个已搜索节点 $x_{nebr}$ 定义为 $x_{new}$ ,从根节点到已搜索节点 $x_{nebr}$ 与节点 $x_{nebr}$ 到 $x_{new}$ 的路径代价之和定义为 $c_{min}$ 。随后选取集合 $X_{near}$ 里其他已搜索节点进行同样的碰撞检测和路径代价之和计算,如果比 $c_{min}$ 小,一样进行 $c_{min}$ 和 $x_{min}$ 的重定义。最终,路径代价之和最小的 $x_{nebr}$ 会取代 $x_{nearest}$ 成为 $x_{new}$ 的父节点。如果遍历集合 $X_{near}$ 内所有的已搜索节点 $x_{nebr}$ ,都没有找到通过碰撞检测或者路径代价之和比初始定义的 $c_{min}$ 还要小的节点,则 $x_{new}$ 的父节点依然是 $x_{nearest}$ ,没有发生改变。

Step 4: 找到新节点 $x_{new}$ 的最优父节点 $x_{parent}$ 之后,将更换后最优父节点 $x_{parent}$ 的新节点 $x_{new}$ 加入到随机搜索树 $T_1$ 中,并把最优父节点 $x_{parent}$ 和新节点 $x_{new}$ 连接起来的边存入随机搜索树 $T_1$ 中。

Step 5: 在之前选取 $x_{new}$ 最优父节点的圆形区域内,实现对已规划部分路径的优化。遍历选取集合 $X_{near}$ 里所有的已搜索节点 $x_{nebr}$ 并进行以下操作:如果新节点 $x_{new}$ 与选择的某一个已搜索节点 $x_{nebr}$ 之间的路径没有与障碍物碰撞,则计算从根节点到 $x_{new}$ 与 $x_{new}$ 到这个已搜索节点 $x_{nebr}$ 之间的路径代价之和,并与从根节点到这个已搜索节点 $x_{nebr}$ 的路径代价相比较。如果从根节点到 $x_{new}$ 再到 $x_{nebr}$ 的路径长度小于根节点到 $x_{nebr}$ 的路径长度,则找到这个已搜索节点 $x_{nebr}$ 的父节点,把随机搜索树 $T_1$ 边集 $E$ 中 $x_{nebr}$ 父节点与 $x_{nebr}$ 连接的边删除,再把节点 $x_{new}$ 与 $x_{nebr}$ 重新连接成边,并将修改好的路径保存于边集 $E$ 中,完成 $x_{new}$ 替换 $x_{nebr}$ 父节点的步骤。

Step 6: 若随机搜索树 $T_1$ 和随机搜索树 $T_2$ 的树枝可以连接起来,则路径规划成功。否则就比较搜索树 $T_1$ 和搜索树 $T_2$ 的路径长度,然后交换2棵树 $T_1$ 、 $T_2$ 的扩展方式,选择路径长度较短的那一棵搜索树进行贪婪扩展,另外一棵搜索树进行有改进随机采样和优化父节点选取步骤的随机扩展。

### 3 仿真实验及结果分析

#### 3.1 参数设置

仿真平台为 PyCharm Community Edition。仿真实验过程中,结合文献[20]实验参数的选取方法以及实验法对相关参数进行微调,最终算法参数采用以下常量和变量设置:实验开始时,移动机器人对周围环境处于陌生状态,为了增加随机搜索树初步探索的随机性,设置终点偏向概率 $P_g=0.2$ 。经过多次仿真实验,设置已搜索节点偏向概率 $P_r=0.2$ ,随机搜索树扩展步长 $u=2$ ,重选父节点圆形区域半径 $r_{near}$ 初始值为2。连接型双向随机搜索树中,为了避免陷入局部最优,以贪婪扩展方式生长的随机树朝着另一棵随机搜索树新节点生长的偏向概率 $P_d$ 为0.8。因为在矩形中,位于对角线上的两点相隔距离最远,能更好地检测算法性能<sup>[21]</sup>,本研究仿真环境起始点坐标位置和目标点坐标位置设定在矩形环境的对角线上。

#### 3.2 实验及结果分析

设计多种类型障碍物环境和狭长通道障碍物

环境2种仿真实验环境,用于对比验证RRT路径规划算法、RRT\_Connect路径规划算法、RRT\*(rapidly-exploring random trees\*)<sup>[22]</sup>路径规划算法和本研究提出的改进RRT\_Connect算法在路径规划仿真实验中的性能。

##### 3.2.1 实验环境1:多种类型障碍物环境

多种类型障碍物环境分辨率为 $56 \times 36$ ,主体为6个灰色圆形障碍物和2个灰色矩形障碍物组成的对称结构,移动机器人X型红色起始点坐标为(0, 0),X型蓝色目标点坐标为(50, 30)。在本地图中,设置最大迭代次数为 $T_{max}=750$ 。在多种类型障碍物环境中,使用上述4种不同的算法进行路径规划,规划路线如图4所示。

在图4中,单向随机搜索树扩展的枝条为绿色细线,双向随机搜索树中2棵树扩展的枝条分别为绿色和黄色细线,4种算法最终搜索到的路径用红色线条表示。由于RRT\*算法和改进RRT\_Connect算法都会在搜索过程中通过重选父节点操作来优化规划路线长度,次优规划路径用蓝色细线表示,红色线条为算法满足最大迭代次数下或者无法生

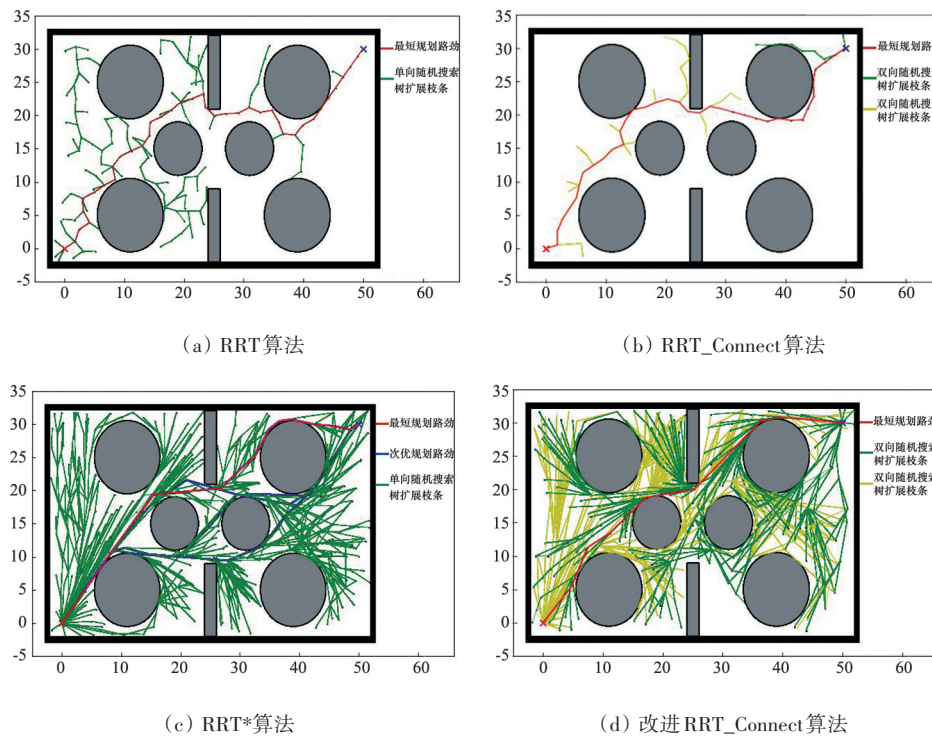


图4 多种类型障碍物环境下4种不同算法的规划路线

成有效新节点和无法进行重选父节点操作的最短规划路径。

从图4可见,RRT算法因其探索的随机性,规划的路线弯曲严重,转弯次数最多,RRT\_Connect算法除了贪婪扩展时路线比较笔直,其他部分路线也存在弯曲现象。RRT\*算法对规划路线进行优化之后,转弯次数大量减少。相比之下,改进RRT\_Connect算法优化的规划路线,转弯次数最少,路径笔直且规划合理,紧贴障碍物内侧安全通过。

同时对4种算法运行100次的寻路时间、规划路径长度、搜索迭代次数、搜索迭代节点个数等指标求取平均值进行性能对比,如表1所示。从表1中可以看出,在多种类型障碍物环境下,改进

RRT\_Connect算法寻路时间最短。从规划路径长度来看,改进RRT\_Connect算法和RRT\*算法的结果相差不大,平均值低于65 m,而另外2种算法规划的路线长度均值达到75 m以上。从搜索迭代次数来看,RRT算法和RRT\*算法都迭代200次以上,其次是RRT\_Connect算法迭代127次,改进RRT\_Connect算法迭代次数最少。从搜索迭代节点个数来看,改进RRT\_Connect算法生成新节点个数也是最少,RRT算法和RRT\*算法生成了100个左右的新节点,说明改进RRT\_Connect算法搜索效率高,能减少系统内存的损耗。同时,改进RRT\_Connect算法转弯次数最少,规划成功率最高,路径规划性能总体上比其他3种算法更好。

表1 实验环境1中4种不同算法性能对比

算法	运行次数	平均寻路时间/s	平均规划路径长度/m	平均搜索迭代次数	平均搜索迭代节点数	平均转弯次数	平均规划成功率/%
RRT算法	100	28.35	76.41	287.80	107.95	24.00	90.00
RRT_Connect算法	100	12.40	80.57	126.70	77.15	17.00	95.00
RRT*算法	100	25.66	64.50	230.60	98.80	10.00	92.00
改进RRT_Connect算法	100	13.89	64.83	118.20	75.30	7.00	96.00

### 3.2.2 实验环境2:狭长通道障碍物环境

狭长通道障碍物环境分辨率为 $56 \times 36$ ,主体为4个灰色矩形障碍物围成的狭窄通道,移动机器人X型红色起始点坐标为(0,0),X型蓝色目标点坐标为(50,30)。在本地图中,设置最大迭代次数为 $T_{max}=1400$ 。在狭长通道障碍物环境中,使用上述4种不同的算法进行路径规划,4种算法规划的路线如图5所示。从图5可以明显看出,RRT\*算法和改进RRT\_Connect算法对规划路线进行优化之后,与RRT算法和RRT\_Connect算法相比,极大地减少移动机器人转弯次数,从而缩短规划路径长度,提高规划路线质量。

本次实验将4种不同算法运行100次的性能指标数据记录下来,求取平均值进行性能对比,如表2所示。从表2中可以看出,在狭长通道环境下,改进RRT\_Connect算法规划次优路径的时间最少,比RRT算法快了49.17%,比RRT\*算法快了45.72%,

比RRT\_Connect算法快了29.53%。从平均规划路径长度来看,改进RRT\_Connect算法和RRT\*算法的结果都低于90 m,两者之间仅相差0.36 m,而另外2种算法规划的路线都达到100 m以上。从搜索迭代次数来看,RRT算法和RRT\*算法都迭代了700次以上,其次是RRT\_Connect算法迭代了500次以上,改进RRT\_Connect算法迭代次数最少。从搜索迭代节点个数来看,改进RRT\_Connect算法生成新节点个数最少,比其他3种算法所生成的160个新节点少了12.5%。同时,改进RRT\_Connect算法平均转弯次数最少,平均规划成功率最高。

结合2个实验环境的实验结果来看,随着障碍物的变化和规划路径长度的增加,4种算法的路径规划性能都有一定的下降,并且所提方法相较于其他方法的效率不同。RRT算法生成的节点比较随机,导致随机树扩展时没有方向。RRT\_Connect是从起点和终点同时递增地生长两个快速探索随机

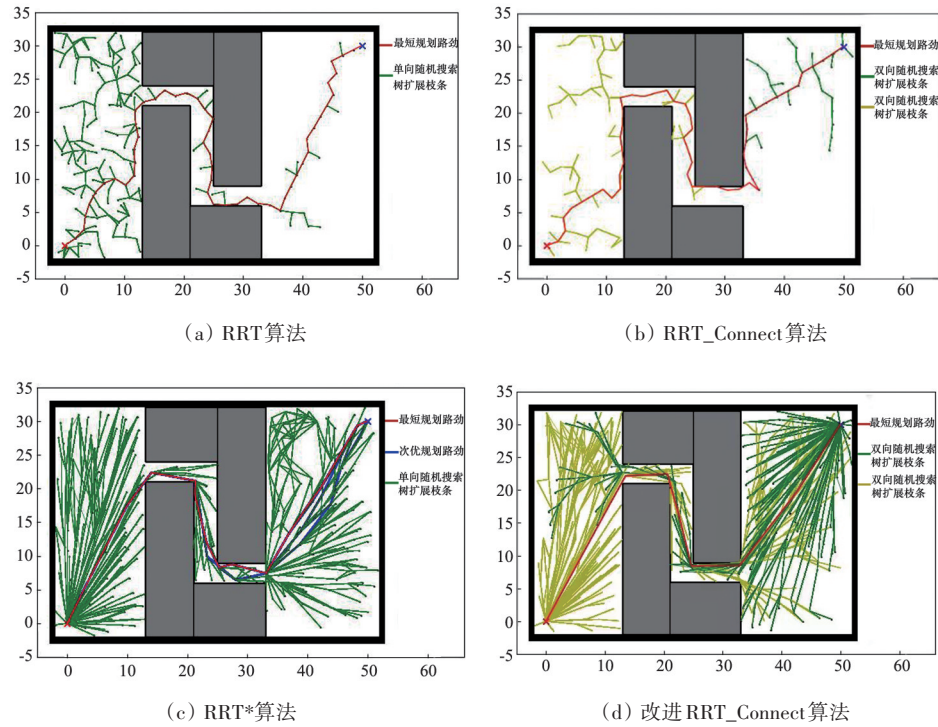


图5 狭长通道障碍物环境下4种不同算法的规划路线

表2 实验环境1中4种不同算法性能对比

算法	运算次数	平均寻路 时间/s	平均规划 路径长度/m	平均搜索 迭代次数	平均搜索 迭代节点数	平均转弯 次数	平均规划 成功率/%
RRT算法	100	67.46	101.55	723.50	161.30	27.00	82.00
RRT_Connect算法	100	48.66	107.16	550.05	159.90	20.00	85.00
RRT*算法	100	63.17	86.54	764.35	163.80	10.00	84.00
改进RRT_Connect算法	100	34.29	86.90	430.10	141.95	6.00	88.00

树,能够显著减少找到有效路径的时间,但仍然没有解决随机取样的问题。RRT\*算法流程与RRT算法流程基本相同,不同之处在于优化选择 $x_{new}$ 的父节点。而改进RRT\_Connect算法引入了目标偏向思想,使随机树的生长具有一定的方向性,减少因随机采样生成的无用节点,同时继承了RRT\_Connect算法双向搜索的特性,进一步加快搜索效率,并且通过利用已搜索节点和已规划部分路径不仅优化选择 $x_{new}$ 的父节点,还优化选择 $x_{near}$ 的父节点,优化规划路线长度,比RRT\_Connect算法平均少了20%。与同样具有优化搜索路径步骤的RRT\*算法相比,改进RRT\_Connect算法搜索迭代次数最少,生成有效新节点个数也最少,说明该算法搜索效率

高,运行占用内存少,能有效减少机器人控制器的损耗。并且平均转弯次数最少,平均规划成功率最高,这说明改进RRT\_Connect算法优化规划路径的能力优于RRT\*算法。实验证明了改进RRT\_Connect算法在路径规划中具有优越的性能。

## 4 结论

针对传统RRT\_Connect算法存在的规划路径曲折、绕远等问题,提供了一种优化已搜索节点和已规划部分路径的改进RRT\_Connect算法。该算法引入终点和已搜索节点偏向策略,以一定的概率实现对随机点的目标偏置选择,引导随机树向目标

位置延伸,使随机树的扩展过程带有一定的方向性,从而减少了冗余节点的产生,加快路径搜索效率。另外,改进 RRT\_Connect 算法对父节点的选取进行优化并重新布线,优选父节点将使得新扩展节点的路径代价尽可能小,而重新布线则使扩展新节点后的随机树减少了冗余路径,优化了路径规划成本。通过仿真实验,证明了改进 RRT\_Connect 算法耗时更短,规划路径更平稳,系统内存能耗更低。目前该研究还处于模拟实验阶段,后期将搭建物理实验平台对提出的算法进一步验证。虽然改进 RRT\_Connect 算法的实际适用性尚未得到验证,但改进 RRT\_Connect 算法用于路径规划,能够有效降低机器人的路径代价和成本,可广泛应用在无人车、无人船、无人机等自动驾驶运输工具中,以达到控制其沿最优路径行驶的目的。

#### 参考文献(References)

- [1] Zhu Z X, Wang F X, He S, et al. Global path planning of mobile robots using a memetic algorithm[J]. *International Journal of Systems Science*, 2015, 46(11): 1982–1993.
- [2] Sivaranjani A, Vinod B. Artificial potential field incorporated deep-Q-network algorithm for mobile robot path prediction[J]. *Intelligent Automation & Soft Computing*, 2023, 35(1): 1135–1150.
- [3] Liu L S, Yao J X, He D W, et al. Global dynamic path planning fusion algorithm combining jump-A\* algorithm and dynamic window approach[J]. *IEEE Access*, 2021, 9: 19632–19638.
- [4] 杨韵, 王成彦, 巫凯旋, 等. 移动机器人全局路径规划算法综述[J]. *信息记录材料*, 2022, 23(3): 29–32.
- [5] 宋永杰, 孟祥印, 翟守才, 等. 改进 Bi-RRT 算法的 AGV 全局路径规划[J]. *机械设计与制造*, 2022(8): 287–291, 296.
- [6] Wang X Y, Li X J, Guan Y, et al. Bidirectional potential guided RRT\* for motion planning[J]. *IEEE Access*, 2019, 7: 95046–95057.
- [7] Kurenkov M, Potapov A, Savinykh A, et al. NFOMP: Neural field for optimal motion planner of differential drive robots with nonholonomic constraints[J]. *IEEE Robotics and Automation Letters*, 2022, 7(4): 10991–10998.
- [8] Ma N C, Wang J K, Liu J B, et al. Conditional generative adversarial networks for optimal path planning[J]. *IEEE Transactions on Cognitive and Developmental Systems*, 2022, 14(2): 662–671.
- [9] LaValle S M, Kuffner J J Jr. Randomized kinodynamic planning[J]. *The International Journal of Robotics Research*, 2001, 20(5): 378–400.
- [10] Chen Y Y, Fu Y X, Zhang B, et al. Path planning of the fruit tree pruning manipulator based on improved RRT-Connect algorithm[J]. *International Journal of Agricultural and Biological Engineering*, 2022, 15(2): 177–188.
- [11] 王海芳, 崔阳阳, 李鸣飞, 等. 基于改进 RRT\*FN 的移动机器人路径规划算法[J]. *东北大学学报(自然科学版)*, 2022, 43(9): 1217–1224, 1249.
- [12] 赵文龙, Abdou Y M S. 基于改进 RRT 算法的移动机器人路径规划方法[J]. *计算机与数字工程*, 2022, 50(8): 1733–1738.
- [13] Fan Q G, Cui G M, Zhao Z Q, et al. Obstacle avoidance for microrobots in simulated vascular environment based on combined path planning[J]. *IEEE Robotics and Automation Letters*, 2022, 7(4): 9794–9801.
- [14] Ayawli B B K, Mei X, Shen M Q, et al. Optimized RRT-A\* path planning method for mobile robots in partially known environment[J]. *Information Technology and Control*, 2019, 48(2): 179–194.
- [15] Sun Y X, Zhang C R, Sun P C, et al. Safe and smooth motion planning for mecanum-wheeled robot using improved RRT and cubic spline[J]. *Arabian Journal for Science and Engineering*, 2020, 45(4): 3075–3090.
- [16] Li B H, Chen B D. An adaptive rapidly-exploring random tree[J]. *CAA Journal of Automatica Sinica*, 2022, 9(2): 283–294.
- [17] Ganesan S, Natarajan S K, Srinivasan J. A global path planning algorithm for mobile robot in cluttered environments with an improved initial cost solution and convergence rate[J]. *Arabian Journal for Science and Engineering*, 2022, 47(3): 3633–3647.
- [18] 韦玉海, 张辉, 刘理, 等. 基于 AMRRT-Connect 算法的移动机器人路径规划[J]. *武汉大学学报(工学版)*, 2022, 55(5): 531–538.
- [19] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. *The International Journal of Robotics Research*, 2011, 30(7): 846–894.
- [20] 王赟. 基于 RRT 轮式机器人路径规划方法研究[D]. 天津: 天津工业大学, 2019.
- [21] Qi J, Yang H, Sun H X. MOD-RRT\*: A sampling-based algorithm for robot path planning in dynamic environment[J]. *IEEE Transactions on Industrial Electronics*,

2021, 68(8): 7244–7251.  
[22] Ryu H, Park Y. Improved informed RRT\* using gridmap skeletonization for mobile robot path planning[J]. Inter-

national Journal of Precision Engineering and Manufacturing, 2019, 20(11): 2033–2039. rong-about-china-and-technical-standards-pub-89110.

## Path planning of mobile robot using improved RRT\_Connect algorithm

SHI Xiaoxu<sup>1</sup>, ZHANG Lieping<sup>1\*</sup>, TANG Liu<sup>2</sup>, DONG Luxi<sup>1</sup>, PENG Jiansheng<sup>3</sup>

1. Key Laboratory of Advanced Manufacturing and Automation Technology (Guilin University of Technology), Education Department of Guangxi Zhuang Autonomous Region, Guilin 541006, China
2. School of Artificial Intelligence, Hezhou University, Hezhou 542899, China
3. Key Laboratory of AI and Information Processing Hechi University, Education Department of Guangxi Zhuang Autonomous Region, Hechi 546300, China

**Abstract** Path planning is an indispensable technology in the field of robot research. Aiming at the path planning problem of mobile robot in complex unknown environment, an improved RRT\_Connect algorithm is proposed to optimize the searched nodes and planned partial paths. Firstly, the algorithm introduces bias strategy of endpoint and searched node, which introduces bias probability reference value of endpoint and searched node in random sampling function, so that random sampling point is set as endpoint or searched node according to random probability. Then, by screening the effective new nodes and the parents of neighboring nodes in a certain range, the path planning cost is optimized to make the planned path tend to be smooth. The simulation results show that the path planned by the proposed improved RRT\_Connect algorithm is better than before in terms of the average number of turns, average planned path length and average planning success rate.

**Keywords** RRT\_Connect algorithm; mobile robot; path planning; goal biased strategy ●



(责任编辑 刘志远)