

doi: 10.7690/bgzd.2018.09.019

基于 RRT 优化算法的移动机器人路径规划

曹 凯, 高佳佳, 李 昂

(西安工业大学电子信息工程学院, 西安 710021)

摘要: 针对快速扩展随机树算法 (rapidly-exploring random trees, RRT) 存在的不足, 以移动机器人的研究为背景, 对 RRT 算法进行改进优化。依据 RRT 算法扩展原理, 构建仿真环境地图, 通过完成基于概率 P 的 RRT 优化算法和基于双向 RRT 优化算法仿真实验, 将这 2 种优化算法结合, 提出了基于概率 P -双向的 RRT 优化算法, 并进行了仿真实验验证。仿真结果表明, 经过优化后的 RRT 算法在复杂障碍环境中有着良好的收敛效果和鲁棒性。

关键词: 移动机器人; 路径规划; RRT 算法; 双向 RRT; 概率 P -双向 RRT

中图分类号: TP24 **文献标志码:** A

Path Planning Based on RRT Optimization Algorithm for Mobile Robot

Cao Kai, Gao Jiajia, Li Ang

(School of Electronic Information Engineering, Xi'an Technology University, Xi'an 710021, China)

Abstract: Aiming at the shortcomings of the rapid exploration of random tree (RRT), the RRT algorithm is improved and optimized based on the research of mobile robot. According to the extension principle of the RRT, the simulation environment map is constructed, and the simulation experiments of the RRT optimization method based on probability P and based on bidirectional RRT are completed. A probabilistic P -bidirectional RRT optimization algorithm is proposed by combining 2 methods, and verified by simulation experiments. The simulation results show that the optimized RRT algorithm has good convergence and robustness in complex obstacle environments.

Keywords: mobile robots; path planning; RRT algorithm; bidirectional RRT; probabilistic P -Bidirectional RRT

0 引言

在移动机器人的研究领域中, 路径规划算法是最为重要而且不可缺少的组成部分, 是移动机器人在障碍物环境下实现自主移动导航的基础。随着路径规划算法研究的不断深入, 研究人员已经改善并优化了很多传统的路径规划算法, 并不断提出了很多新的路径规划算法, 使得路径规划算法的研究趋势已经逐步偏向复杂化和智能化^[1]。

常用的路径规划算法包括 A^* 、人工势场法、概率路线图 (PRM) 算法、蚁群算法、遗传算法以及诸多启发式算法等^[2-6]。这些算法的收敛速度过于缓慢, 需要提前对未知的障碍物空间环境进行建模, 当环境改变时无法在未知的障碍空间中使用。RRT 算法能有效解决这些传统路径规划算法存在的问题。与其他传统路径规划算法相比, RRT 算法不仅具有快速、高效的特点, 而且不需要依赖于环境建模就能有效地解决未知复杂障碍物空间和高维动态环境的路径规划问题。

为了提高节点的扩展效率, 2000 年, 库夫纳提出了 RRT-connect, 次年, 又提出了双向搜索树

(Bidirectional-RRT) 的算法结构优化, 从起始点和目标点同时出发并行生成 2 棵扩展树, 直至 2 棵树相遇, 加速算法收敛^[7-8]。2010 年, 卡拉曼首次提出 RRT* 算法, 用来解决由 RRT 算法存在的非概率最优解的问题^[9]。2013 年, 佩雷斯提出了 Bi-RRT* 算法, 用变异的贪心算法的改进算法作为启发式函数连接环境空间中的算法随机树^[10]。2014 年, 蒙塔兹提出了 TG-RRT*, 应用三角几何原理来筛选节点, 降低了解最优解所花费的迭代次数, 使算法收敛速度加快^[11]。2015 年, 胡赛因提出了 IB-RRT*, 利用智能样本插入的启发式函数和构造双向树的方法, 使得算法能够快速地收敛到最优路径^[12]。

笔者在理解 RRT 算法扩展原理和建模实现方法的基础上, 分别论述了基于概率 P 的 RRT 优化和双向 RRT 优化算法的基础知识, 并在不同实验场景下进行了仿真。其次, 在双向的扩展结构上引入了概率 P 的判断, 提出了一种基于概率 P -双向的 RRT 优化算法。最后, 在不同环境中通过实验进行验证。实验结果表明: 在复杂的环境中, 该方法能够以较少的扩展节点和较快的收敛速度规划出可行路径。

收稿日期: 2018-07-10; 修回日期: 2018-07-20

基金项目: 陕西省国际科技合作基地项目 (2017GHJD-009)

作者简介: 曹 凯 (1984—), 男, 陕西人, 硕士, 讲师, 从事自主无人系统与智能控制研究。

1 相关工作

1.1 仿真环境地图的建立

在不同的障碍物地图下观察算法的收敛情况可以准确判断算法有效性和鲁棒性。文中算法在仿真中用黑色像素代表环境空间的障碍物，白色像素代表障碍物空间内的自由空间区域。在2维平面内搭建的地图可分为分隔式、隔断式、横断式和结合式，如图1所示。在搭建不同的空间障碍物过程中，必须预留至少一条能够连接起始点到目标点的空白无障碍区域通道，若将起始点和目标点完全用障碍物分隔开，则导致路径规划算法失败，无法找到从起始点到目标点的路径。

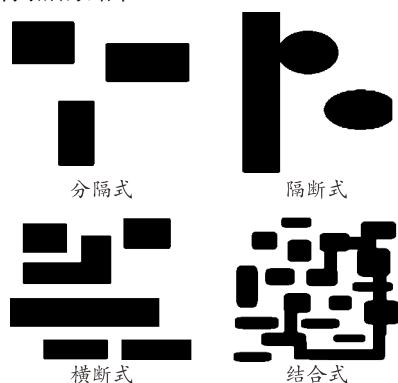


图1 路径规划仿真地图

1.2 RRT 算法

RRT是由美国爱荷华州立大学的Steven Lavalle教授在1998年提出的^[13]，是一种在多维空间中基于采样的规划方法。该方法通过随机探索自由空间，并构建一棵从初始状态开始寻找朝向目标状态的可行路径的树 $T=(V, E)$ 。在迭代的过程中，创建随机点并搜索距离该点最近的顶点，然后以固定步长得到该方向上的新节点，并检查其是否属于自由空间，直至达到目标，如图2所示。在未占用空间中均匀扩展的属性，使得其在复杂的高维环境、动态环境以及非完整和运动约束环境中依旧有效。

RRT 算法流程：

1. $T \leftarrow \text{InitializeTree}()$;
2. $T \leftarrow \text{InsertNode}(\varnothing, z_{\text{init}}, T)$;
3. for $i \leftarrow 0$ to N do
4. $z_{\text{rand}} \leftarrow \text{Sample}()$;
5. $z_{\text{nearest}} \leftarrow \text{Nearest}(T, z_{\text{rand}})$;
6. $(z_{\text{new}}, U_{\text{new}}) \leftarrow \text{Steer}(z_{\text{nearest}}, z_{\text{rand}})$;
7. if $\text{ObstacleFree}(z_{\text{new}})$ then
8. $T \leftarrow \text{InsertNode}(z_{\text{min}}, z_{\text{new}}, T)$;
9. end if

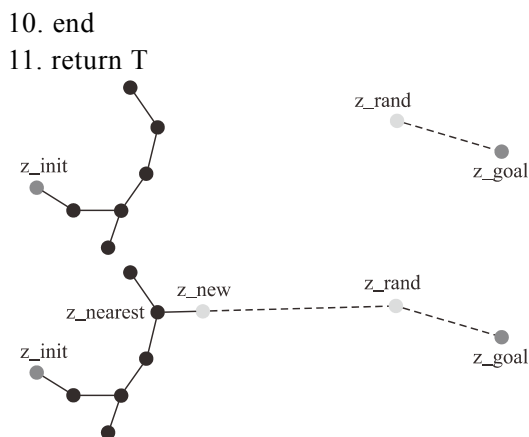


图2 RRT 节点扩展过程

1.3 基于概率 P 的 RRT 算法

1.3.1 基于概率 P 的 RRT 算法的原理

为了加快随机树到达目标点的速度，文献[14]在原始RRT算法的基础上进行改进，其改进方法：在随机树每次的生长过程中，根据随机概率来决定 q_{rand} 是目标点还是随机点。在采样判断时或者在 Sample 函数中设定参数 Prob ，每次得到一个0到 Prob 的随机值 p ，当 $0 < p < \text{Prob}$ 的时候，随机树朝向一个随机方向生长；当 $\text{Prob} < p < 1.0$ 时，随机树朝向目标点生长。其中 Prob 和 P 都在区间 $(0,1)$ 内。在本次仿真实验中，笔者把 Prob 取值 0.5 来观察算法在不同地图的收敛情况。RRT 算法每次都可以按照 0.5 的概率朝向目标点的方向扩展。与原始的RRT算法在地图自由空间内随机扩展的方式相比较，基于概率 P 的RRT算法更加具有方向性和目标性。基于概率 P 的RRT算法流程：

1. $V \leftarrow \{z_{\text{init}}\}$, $E \leftarrow \varnothing$, $i \leftarrow 0$;
2. while $i < n$ do;
3. $T \leftarrow (V, E)$;
4. if $p \in (0, \text{Prob})$
5. $Z_{\text{rand}} \leftarrow \text{sample}(i)$, $i \leftarrow i+1$;
6. else $\text{sample}(i) \leftarrow z_{\text{goal}}$;
7. $V' \leftarrow V$, $E' \leftarrow E$;
8. $Z_{\text{nearest}} \leftarrow \text{Nearest}(G, Z_{\text{rand}})$;
9. $Z_{\text{new}} \leftarrow \text{Steer}(Z_{\text{new}}, Z_{\text{rand}})$;
10. if $\text{ObstacleFree} \leftarrow (Z_{\text{near}}, Z_{\text{new}})$
11. else return $G=(V', E')$;
12. $V' \leftarrow V' \cup \{Z_{\text{new}}\}$;
13. $E' \leftarrow E' \cup \{Z_{\text{near}}, Z_{\text{new}}\}$;
14. return $G=(V', E')$;

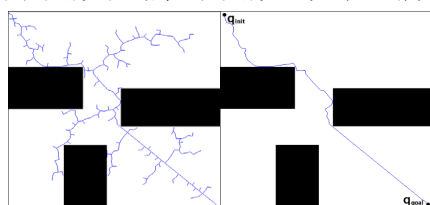
```

15. if  $Z_{\text{new}} = Z_{\text{goal}}$ ;
16. return T
17. else return  $T = (V', E')$ 

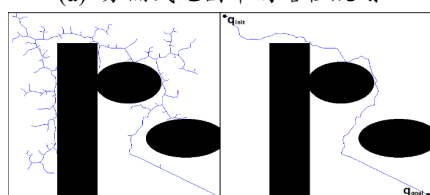
```

1.3.2 基于概率 P 的 RRT 算法的仿真实验

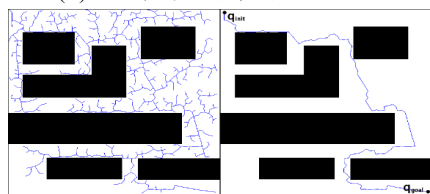
在不同地图障碍物模型下观察基于概率 P 的 RRT 算法的收敛效果,如图 3 所示。通过观察比较原始 RRT 算法的仿真实验效果图可知,基于概率 P 的 RRT 算法不仅可以在不同的障碍物地图下完成路径规划,而且其在障碍地图下自由空间区域的收敛范围与原始的 RRT 算法相比有了较为明显的改善。实验数据记录不同地图中的执行时间、路径长度、地图节点数和路径节点数,如表 1 所示。



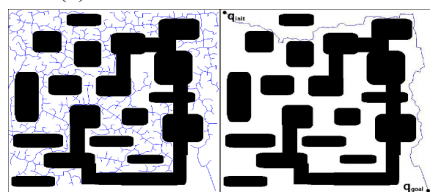
(a) 分隔式地图中的路径规划



(b) 隔断式地图中的路径规划



(c) 横断式地图中的路径规划



(d) 结合式地图中的路径规划

图 3 基于概率 P 的 RRT 算法在不同环境中的收敛图 and 路径规划

表 1 基于概率 P 的 RRT 算法的仿真实验数据

地图	执行时间/s	路径长度/m	扩展节点	路径节点
分隔式地图	23.17	785.14	332	86
隔断式地图	46.57	994.05	489	103
横断式地图	106.48	1 193.35	847	136
结合式地图	127.67	1 287.44	942	147

由表 1 可知,基于概率 P 的 RRT 算法在不同场景均可以规划出可行路径。同时,相比于原始 RRT,

其在简单地图中的收敛速度明显加快,路径长度和扩展节点数大大减少。在复杂地图中虽有改善,但是扩展节点依旧冗余,很大程度地影响了收敛速度。

1.4 双向 RRT 算法

1.4.1 双向 RRT 算法的原理

基于概率 P 的 RRT 算法的优化虽然在一定程度上加快了 RRT 算法的收敛速度,但是在一些复杂的地图下,如横断式地图和结合式地图下应用时,算法的收敛速度依然过慢,路径节点占比率依旧过低,随机树的扩展范围依然很广。针对复杂地图下收敛速度过慢的问题,双向 RRT 算法从结构上进行优化,从而加快算法的收敛速度。它在原有 RRT 算法的基础上进行了算法结构上的改变,分别从起始点和目标点同时生成了 2 棵树。从目标点生成的随机树将目标点作为根节点,起始点作为目标点来进行生长,2 棵树同时在障碍物地图的自由空间区域内以不断迭代的方式随机扩展,当在自由空间内的两棵树相互连接时,则完成路径规划。双向 RRT 算法流程:

```

1.  $V_1 \leftarrow \{Z_{\text{init}}\}, E_1 \leftarrow \emptyset, T_1 \leftarrow (V_1, E_1), V_2 \leftarrow \{q_{\text{goal}}\},$   

 $E_2 \leftarrow \emptyset, T_2 \leftarrow (V_2, E_2), i \leftarrow 0;$ 
2. while  $i < n$  do
3.  $T \leftarrow (V, E)$ 
4.  $Z_{\text{rand}} \leftarrow \text{sample}(i); i \leftarrow i + 1;$ 
5.  $V' \leftarrow V, E' \leftarrow E;$ 
6.  $Z_{\text{nearest}} \leftarrow \text{Nearest}(G, Z_{\text{rand}});$ 
7.  $Z_{\text{new}} \leftarrow \text{Steer}(Z_{\text{new}}, Z_{\text{rand}});$ 
8. if  $\text{Obstaclefree} \leftarrow (Z_{\text{near}}, Z_{\text{new}})$ 
9. else return  $T = (V', E')$ ;
10.  $V_1 \leftarrow V_1 \cup \{q_{\text{new}}\}, E_1 \leftarrow E_1 \cup \{Z_{\text{near}}, Z_{\text{new}}\};$ 
11.  $Z'_{\text{near}} \leftarrow \text{Nearst}(T_2, Z_{\text{new}});$ 
12.  $z \leftarrow \text{Steer}(Z'_{\text{near}}, Z'_{\text{new}});$ 
13. if  $\text{Obstaclefree} \leftarrow (Z'_{\text{near}}, Z'_{\text{new}})$ 
14.  $V_2 \leftarrow V_2 \cup \{Z'_{\text{new}}\}, E_2 \leftarrow E_2 \cup \{Z'_{\text{near}}, Z'_{\text{new}}\}$ 
15. else return  $T_2 = (V_2, E_2)$ 
16.  $Z'_{\text{new}} \leftarrow Z'_{\text{new}};$ 
17. if  $Z'_{\text{new}} = Z_{\text{new}}$ 
18. return  $T_1(V_1, E_1);$ 
19. if  $|V_1| < |V_2|$ 
20. Swap( $V_1, V_2$ );

```

1.4.2 双向 RRT 算法的仿真实验

通过双向 RRT 算法仿真实验,在不同地图障碍

物模型下观察基于双向 RRT 算法的收敛效果, 如图 4 所示。由图可知: 基于双向 RRT 算法可以在不同的障碍物地图下完成路径规划, 其在障碍地图的自由空间区域的收敛范围与基于概率 P 的 RRT 算法相比有了十分显著的改善。仿真实验数据记录了不同地图下的收敛时间、路径长度、地图节点数和路径节点数, 如表 2 所示。

由表 2 可知, 与基于概率 P 的 RRT 算法相比, 双向 RRT 算法不仅拥有更快的收敛速度, 更短的规划路径和在障碍地图上拥有更少节点数量, 而且在较为复杂的地图中也能拥有更快的收敛速度。通过双向 RRT 算法优化改进, 有效解决了在复杂地图下基于概率 P 的 RRT 算法收敛速度过慢的问题。

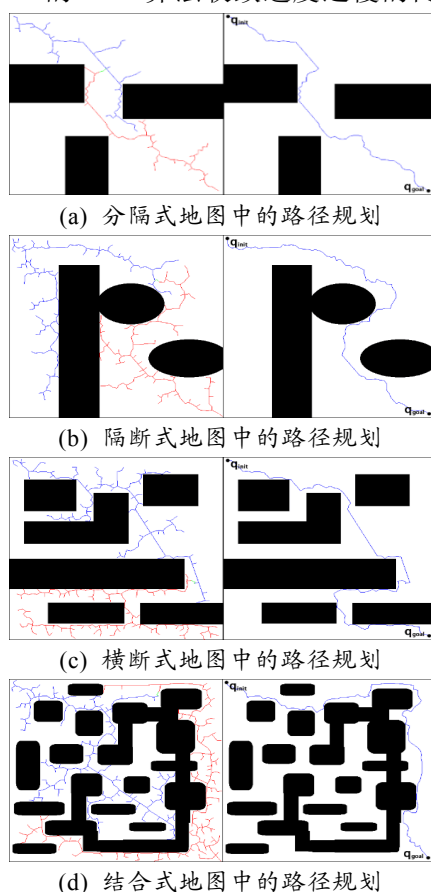


图 4 基于双向 RRT 算法在不同环境中的收敛图和路径规划

表 2 基于双向 RRT 算法的仿真实验数据

地图	收敛时间/s	路径长度/m	扩展节点	路径节点
分隔式地图	19.18	894.64	303	104
隔断式地图	20.12	994.05	288	112
横断式地图	25.30	1 163.93	364	116
结合式地图	40.23	1 223.96	488	127

2 基于概率 P-双向 RRT 算法

2.1 基于概率 P-双向 RRT 算法的原理

基于概率 P 的 RRT 算法和基于双向 RRT 算法

都能解决原始 RRT 算法搜索效率范围广以及效率过低等一系列问题, 将 2 种 RRT 算法的优化进行有机结合, 在原始 RRT 算法的基础上既引入了概率 P 的判断, 又在结构上加入了双向优化, 提出基于概率 P-双向 RRT 算法。该方法首先分别从起始点和目标点各为根节点生成一棵随机生长树, 然后在两棵随机树不断向外扩展时都分别引入了概率 P 的判断。从起始点为根节点的生长树以 P 的扩展概率在障碍物地图中的自由空间内朝向目标点进行扩展, 从目标点为根节点的生长树以 P 的扩展概率在障碍物地图中的自由空间内朝向起始点进行扩展。当两棵随机树在障碍物地图的自由空间内发生连接时, 则可以在障碍物地图中成功完成路径规划。基于概率 P-双向 RRT 算法流程:

```

1.  $V_1 \leftarrow \{Z_{init}\}, E_1 \leftarrow \emptyset, T_1 \leftarrow (V_1, E_1), V_2 \leftarrow \{q_{goal}\},$ 
 $E_2 \leftarrow \emptyset, T_2 \leftarrow (V_2, E_2), i \leftarrow 0;$ 
2. while  $i < n$  do
3.  $T \leftarrow (V, E);$ 
4. if  $P_1, P_2 \in (0, Prob)$ 
5.  $\leftarrow \text{sample}(i); i \leftarrow i+1;$ 
6. else  $\text{sample}(i)_1 \leftarrow z_{goal}; \text{sample}(i)_2 \leftarrow z_{init}$ 
7.  $V' \leftarrow V, E' \leftarrow E;$ 
8.  $z_{nearest} \leftarrow \text{Nearest}(T, z_{rand});$ 
9.  $z_{new} \leftarrow \text{Steer}(z_{new}, z_{rand});$ 
10. if  $\text{Obstaclefree} \leftarrow (z_{near}, z_{new});$ 
11. else return  $T = (V', E')$ ;
12.  $V_1 \leftarrow V_1 \cup \{q_{new}\}, E_1 \leftarrow E_1 \cup \{q_{near}, q_{new}\};$ 
13.  $z'_{near} \leftarrow \text{Nearest}(T_2, z_{new})$ 
14.  $z'_{new} \leftarrow \text{Steer}(z'_{near}, z'_{new});$ 
15. if  $\text{Obstaclefree} \leftarrow (z'_{near}, z'_{new})$ 
16.  $V_2 \leftarrow V_2 \cup \{z'_{new}\}, E_2 \leftarrow E_2 \cup \{z'_{near}, z'_{new}\}$ 
17. else return  $T_2 = (V_2, E_2);$ 
18.  $z'_{new} \leftarrow z_{new};$ 
19. if  $z'_{new} = z_{new}$ 
20. return  $T_1 = (V_1, E_1);$ 
21. if  $|V_1| < |V_2|$ 
22.  $\text{Swap}(V_1, V_2)$ 

```

2.2 基于概率 P-双向 RRT 算法的仿真实验

基于概率 P-双向 RRT 算法仿真实验, 采用与上文相同的地图障碍物模型, 在概率 Prob 的值依旧设定为 0.5 的前提下, 观察基于概率 P-双向 RRT 算法的收敛效果, 如图 5 所示。由图可知: 基于概率 P-双向 RRT 算法可以在不同的障碍物地图下完成

路径规划, 其在障碍地图的自由空间区域的收敛范围也是 3 种 RRT 优化算法中最少的, 符合预期的实验效果。仿真实验数据同样记录基于概率 P-双向 RRT 算法在不同地图下的收敛时间、路径长度、地图节点数和路径节点数, 如表 3 所示。

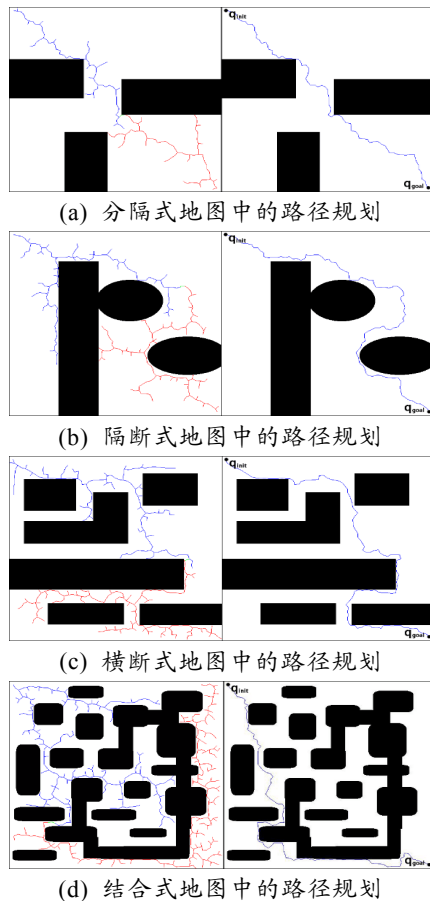


图 5 基于概率 P-双向 RRT 算法在不同环境中的收敛图和路径规划

表 3 基于概率 P-双向 RRT 算法的仿真实验数据

地图	收敛时间/s	路径长度/m	扩展节点	路径节点
分隔式地图	17.18	883.62	297	112
隔断式地图	18.21	964.75	278	147
横断式地图	20.53	1 063.17	304	163
结合式地图	36.79	1 194.72	453	166

由表 3 可知, 该算法在 5 种类型的地图下可以规划一条路径。基于概率 P 的 RRT 算法和基于双向 RRT 算法相比较, 该优化算法兼有了前 2 种优化算法的特点, 有效解决了在复杂程度不同的地图下, 基于概率 P 的 RRT 算法和基于双向 RRT 算法收敛速度过慢的问题。通过实验数据分析可知: 通过 RRT 算法的不断优化改进, 基于概率 P-双向 RRT 优化算法在 4 类地图的仿真实验中均以最少的执行时间和扩展节点收敛到最短的路径。综上所述, 通过仿真实验的比较分析, 基于概率 P-双向 RRT 优化算

法的优化效果是 3 种 RRT 优化算法中效果最好的一种算法。

3 结论

笔者主要介绍了 RRT 算法的改进算法, 针对原始 RRT 算法的收敛速度过慢、规划路径过长、节点数量过多及路径节点占比率低下的问题, 提出了 RRT 算法的改进优化。改进的 RRT 算法分别为基于概率 P 的 RRT 算法、基于双向 RRT 算法和基于概率 P-双向 RRT 算法。基于概率 P 的 RRT 算法引入了概率区间的判断, 使算法的扩展更加有方向性; 基于双向 RRT 算法在算法的结构上进行了优化, 使该算法更适合用于复杂环境下; 基于概率 P-双向 RRT 算法则将 2 种优化方式进行了有机结合, 既引入了概率区间的判断, 又在结构上加入了双向构造。经仿真实验和数据分析可知, 优化的 RRT 算法不仅收敛时间和扩展节点数量更少, 且规划路径的长度也相对更短。

4 致谢

感谢陕西省国际科技合作基地项目(2017GHJD-009)的支持以及项目前期对移动机器人路径规划基础理论知识的深入研究分析。

参考文献:

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.
- [2] SFEIR J, SAAD M, SALIAH-HASSANE H. An improved Artificial Potential Field approach to real-time mobile robot path planning in an unknown environment[C]// IEEE International Symposium on Robotic and Sensors Environments. IEEE, 2011: 208-213.
- [3] 陈辉, 张燕, 夏庆锋. 移动机器人路径规划技术[J]. 兵工自动化, 2018, 37(4): 65-68+72.
- [4] KAVRAKI L E, ŠVESTKA, PETR, LATOMBE J C. Probabilistic roadmaps for path planning in high-dimensional configuration spaces [J]. IEEE Trans on Robotics & Automation, 1994, 12(4): 566-580.
- [5] 张天赫, 彭绍雄, 罗亚民, 等. 基于蚁群算法的舰载机避障路线分析[J]. 兵工自动化, 2017, 36(10): 71-74.
- [6] 李连鹏, 苏中, 解迎刚, 等. 基于遗传算法的机器鱼水中路径规划[J]. 兵工自动化, 2015, 34(12): 93-96.
- [7] KUFFNER J J, LAVALLE S M. RRT-connect: An efficient approach to single-query path planning[C]//IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA. IEEE, 2002: 995-1001.
- [8] LAVALLE S M, KUFFNER J J J. Randomized Kinodynamic Planning[J]. International Journal of

- Robotics & Research, 2001, 15(5): 378-400.
- [9] KARAMAN S, FRAZZOLI E. Optimal kinodynamic motion planning using incremental sampling-based methods[C]//IEEE Conference on Decision and Control. IEEE, 2010: 7681-7687.
- [10] JORDAN M, PEREZ A. Optimal Bidirectional Rapidly Exploring Random Trees [Z]. 2013.
- [11] QURESHI A H, MUMTAZ S, IQBAL K F, et al. Triangular geometry based optimal motion planning using RRT*-motion planner[C]// IEEE, International Workshop on Advanced Motion Control. IEEE, 2014: 380-385.
- [12] QURESHI A H, AYA Z Y. Intelligent Bidirectional Rapidly-exploring Random Trees for Optimal Motion Planning in Complex Cluttered Environments [J]. Robotics and Autonomous Systems, 2015, 68: 1-11.
- [13] LAVALLE S. Rapidly-exploring random trees: a new tool for path planning[Z]. Research Report, 1998: 293-308.
- [14] ABBADI A, PRENOSIL V. Collided Path Replanning in Dynamic Environments Using RRT and Cell Decomposition Algorithms[J]. 2015, 9055: 131-143.

(上接第 73 页)

用经典方法评估置信度 $\gamma=0.9$ 时,求得 $R_L=0.847\ 4$,由于新研部件价值高、试验危险性较大,未进行大量试验,导致评估结果较为保守。相比于经典评估方法用 3 种共轭先验分布评估,得到的结果较为合适。对于该部件,试验之前专家建议选用 Beta(0,0)作为验前分布,求得 $R_L=0.904\ 7$,后续的正样试验与鉴定试验结果也证明了采用 Beta(0,0)的正确性与优越性。

5 结束语

笔者介绍了成败型产品可靠性评估经典方法与 Bayes 方法,指出经典评估方法的不足,并针对无验前信息的成败型弹药部件可靠性评估问题,介绍 3 种常用的验前分布,通过数值仿真研究了在无先验信息情况下, Beta(0,0)、Beta(1/2,1/2)、Beta(1,1)验前分布对可靠性置信下限的影响,得到了 3 种共轭验前分布的适用情况,最后通过算例表明共轭验前分布求无先验信息弹药部件可靠性的优越性。该研究可为弹药可靠性评估提供参考。

参考文献:

- [1] 潘勇,黄进永,胡宁. 可靠性概论[M]. 北京: 电子工业出版社, 2015: 457-458.

- [2] 王学,冯志刚,高普云,等. 寿命型到成败型数据的等效折算[J]. 机械强度, 2009, 31(2): 240-244.
- [3] 张文杰,杨华波,张士峰. 基于 Bayes 混合验前分布的成败型产品可靠性评估[J]. 兵工学报, 2016, 37(3): 505-511.
- [4] 叶豪杰,赵占伟. 采取随机数的指数型系统贮存可靠性仿真评估[J]. 电子产品可靠性与环境试验, 2012, 35(1): 27-29.
- [5] 丁湛,黄双华. 基于威布尔分布的可靠性寿命分布模型的建立[J]. 电子测量技术, 2007, 30(3): 34-35.
- [6] 张丰收,韩尚云,刘建亭,等. 基于 Bayes 理论的重型数控机床可靠性评估方法研究[J]. 现代制造工程, 2015, 36(8): 122-125.
- [7] 张士峰,樊树江,张金槐. 成败型产品的可靠性 Bayes 评估[J]. 兵工学报, 2011, 22(2): 238-240.
- [8] 赵勇,刘建新,牛青坡. 基于混合 Beta 先验分布的成败型产品的可靠性评估[J]. 航空兵器, 2014, 49(6): 59-61.
- [9] 邵宗战,马浩文. 航空深弹实航工作可靠度评定方法[J]. 鱼雷技术, 2015, 23(6): 475-478.
- [10] 王玮,蔡瑞娇,焦清介. 制导弹药可靠性评定方法的研究[J]. 兵工学报, 2007, 28(7): 800-803.
- [11] 周源泉,翁朝曦. 可靠性评定[M]. 北京: 科学出版社, 1990: 91-92.
- [12] 何劼,张彬彬. 应用 Jeffreys 方法计算可靠性参数和始发事件频率的无信息先验[J]. 原子能科学技术, 2013, 47(11): 2059-2062.