

引文格式：

张可, 宋呈群, 程俊, 等. 基于多目标快速探索随机树的移动机器人巡检路径优化方法 [J]. 集成技术, 2023, 12(4): 32-41.
Zhang K, Song CQ, Cheng J, et al. Multi-objective rapidly-exploring random tree robot patrol path optimization method [J]. Journal of Integration Technology, 2023, 12(4): 32-41.

基于多目标快速探索随机树的移动机器人巡检路径优化方法

张可^{1,2} 宋呈群^{1,2*} 程俊^{1,2*} 张锲石^{1,2} 曾驳^{1,2}

¹(中国科学院深圳先进技术研究院 中国科学院人机智能协同系统重点实验室 深圳 518055)

²(香港中文大学 香港 999077)

摘要 针对移动机器人需要访问多目标的巡检路径规划问题, 该文提出一种多目标快速探索随机树路径优化方法。首先, 根据提供的环境地图与巡检目标点, 该文采用一种 RRT-Connect-ACO 算法得到目标点的巡检顺序和可行路径; 然后, 通过引入信息子集, 对路径进行优化, 得到最终的最优路径。实验结果表明, 与现有的多目标路径规划算法相比, 该方法考虑了地形的影响, 得到的最优路径更符合实际情况。

关键词 多目标路径规划; 快速探索随机树; 旅行商; 蚁群算法; 信息子集; 移动机器人
中图分类号 TP 242 文献标志码 A doi: 10.12146/j.issn.2095-3135.20220721001

Multi-objective Rapidly-Exploring Random Tree Robot Patrol Path Optimization Method

ZHANG Ke^{1,2} SONG Chengqun^{1,2*} CHENG Jun^{1,2*} ZHANG Qieshi^{1,2} ZENG Bo^{1,2}

¹(CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China)

²(The Chinese University of Hong Kong, Hong Kong 999077, China)

*Corresponding Authors: cq.song@siat.ac.cn; jun.cheng@siat.ac.cn

Abstract A multi-objective rapidly-exploring random tree path optimization method is proposed for the multi-objective patrol path planning of mobile robots. According to the provided environment map and patrol target points, a new method RRT-Connect-ACO is used to obtain the patrol sequence and feasible path of the target points. Then the optimal path is obtained by introducing informed subset to optimize the path. The experiment

收稿日期: 2022-07-21 修回日期: 2022-09-15

基金项目: 国家自然科学基金项目(U21A20487); 深圳市科技计划项目(JCYJ20180507182610734, KCXFZ20201221173411032); 中国科学院关键技术人才项目

作者简介: 张可, 硕士研究生, 研究方向为机器人路径规划等; 宋呈群(通讯作者), 副教授, 研究方向为计算机视觉、机器人等, E-mail: cq.song@siat.ac.cn; 程俊(通讯作者), 教授, 研究方向为计算机视觉、机器人等, E-mail: jun.cheng@siat.ac.cn; 张锲石, 高级工程师, 研究方向为智能驾驶与机器人、行为识别与分析等; 曾驳, 硕士研究生, 研究方向为计算机视觉、同时定位与建图等。

results show that the method considers the influence of terrain and obtains an optimal path that is more consistent with the actual situation, which is different from the existing multi-objective path planning algorithms.

Keywords multi-objective path planning; rapidly-exploring random tree; travelling salesman problem; ant colony optimization; informed subset; mobile robots

Funding This work is supported by National Natural Science Foundation of China (U21A20487), Shenzhen Technology Project (JCYJ20180507182610734, KCXFZ20201221173411032), CAS Key Technology Talent Program

1 引 言

随着计算机技术、传感器技术的飞速发展, 移动机器人已广泛应用于日常生产生活, 并迎来了发展高峰期。面向巡检任务^[1]的移动机器人在巡检过程中的路径规划极为重要, 最优的路径可以及时发现隐患和安全事故^[2]。本文将巡检问题视作旅行商问题(travelling salesman problem, TSP)^[3]和路径规划(path planning, PP)问题^[4-5]的结合, 根据给定的环境地图和任务需求访问的多个点, 确定访问顺序, 并得到最优的访问路径。

巡检机器人的路径规划可以被简单描述为 TSP。TSP 是组合优化问题中的一个著名难题, 传统的 TSP 可以叙述为: 需要推销员去若干城市推销产品, 规定推销员必须到达每个城市且只去一次, 最后还需回到出发城市。TSP 的目标函数是找到符合上述条件的最短访问路线。在 TSP 中, 每个顶点即为需要访问的城市, 边为两个城市之间的距离。此类问题用一般的算法很难得到最优解, 早期使用分支界定^[6]、动态规划法^[7]等精确算法^[8-9]求解, 但随着解决问题规模的扩大, 将难以获得精确结果, 因而启发式算法被提出。启发式算法得到的结果虽然不能证明是最优解, 但可以有效地提供次优或者可能最优的解, 其分为传统启发式算法和元启发式算法^[3,8]。传统启发式算法^[7]包括最近邻算法(nearest neighbor, NN)、爬山法等, 此类算

法原理简单, 但优化结果与全局最优解差距明显; 元启发算法是基于启发式算法的改良, 不依赖特定问题, 采用通用启发式策略寻优, 如模拟退火(simulated annealing, SA)^[8,10]、遗传算法(genetic algorithm, GA)^[11-12]、蚁群算法(ant colony optimization, ACO)^[13-15]等。

ACO 是一种群智能算法, 由一群无智能或有轻微智能的个体(Agent)通过相互协作而表现出智能行为, 从而为求解复杂问题提供了一个新的可能性。ACO 最早由意大利学者 Colomi 等^[13]提出。经过 30 多年的发展, ACO 在理论以及应用研究上取得了巨大的进步。ACO 最早用于求解 TSP, 其具有分布式特性, 鲁棒性强且易与其他算法结合, 表现出了较强的优越性, 但同时存在收敛速度慢、容易陷入局部最优等缺点。

路径规划是机器人学的一个基础研究领域, 其目的是为机器人找到一条可行的路径, 同时避免环境中的障碍物。其主要分为基于网格和基于采样这两种方法。

基于网格的算法, 如 A*^[16-17]、D*^[18]可以找到最佳路径, 但时间成本和计算量会随着规模增加而呈指数增长。

基于采样的算法, 如概率路线图^[19-20]和快速探索随机树(rapidly-exploring random tree, RRT)^[21], 因其能够有效解决多约束和高维路径规划问题而被广泛应用。特别是 RRT 可有效处理具有微分约束的系统, 因此, RRT 成为许多机器人平台上

流行的一种路径规划算法^[15]。RRT 算法在规划路径之前必须获取状态空间的全部信息，然后以起始点为根节点，在状态空间内进行随机抽样，通过特定的约束条件逐步扩展随机树。当随机树上的点到达目标点的邻域范围内时，即可找到一条从起始点到目标点的有效路径。

RRT 算法的双向变体已成功应用于复杂的路径规划，Kuffner 等^[22]提出一种快速 RRT 算法，即在起始点和目标点同时构建 RRT，并交替展开两棵随机树。与单向 RRT 算法相比，该算法可以更快地找到一条可行路径。

尽管 RRT 算法相对高效，可以较好地处理带有非完整约束的路径规划问题，但不能保证得到的可行路径是相对优化的。为进一步改进 RRT 算法，Karaman 等^[23]提出了 RRT*算法，通过重新连接节点优化生成路径。与 RRT 算法相比，RRT*算法的主要优化过程为：重新选择父节点和重布线随机树。RRT*具有渐进最优性，且不能在有限时间内得到最优路径，故其收敛时间是一个较为突出的问题，表现为收敛速度缓慢。

基于 RRT*的框架，相关学者提出了许多路径规划的变体，如 Gammell 等^[24]提出一种 informed-RRT*方法，将采样范围限制在椭圆内，对规划问题的状态维度和范围的依赖性较小，可提高收敛速度。

目前，针对多目标路径规划的研究较少，大部分多目标路径规划的方法是直接采用两点间的欧氏距离，如 Wang 等^[15]的研究，其应用场景是较为空旷的机场，因此可以忽略障碍物的影响，直接采用欧氏距离作为任意两目标点之间的路径代价，但对于略微复杂的环境，该方法将可能引起较大的路径代价。

为解决上述移动机器人访问多目标的巡检路径规划问题，本文提出一种多目标快速探索随机树巡检路径优化方法。与已有的多目标路径规划策略不同，本文利用 RRT-Connect-ACO 算法得

到各重点监测点间的可行路径，并以路径长度作为各点之间的路径代价，得到更具可靠性的巡检顺序，同时，大大降低了由于迭代次数不足导致无法生成路径的风险。此外，本文还引入信息子集，缩小采样空间，提高了路径优化效率。

2 方法

本文方法的流程如图 1 所示，根据提供的环境地图与指定的巡检多目标点，利用 RRT-Connect-ACO 算法得到目标点的巡检顺序和可行路径，通过引入信息子集对路径进行优化，最终得到最优路径。

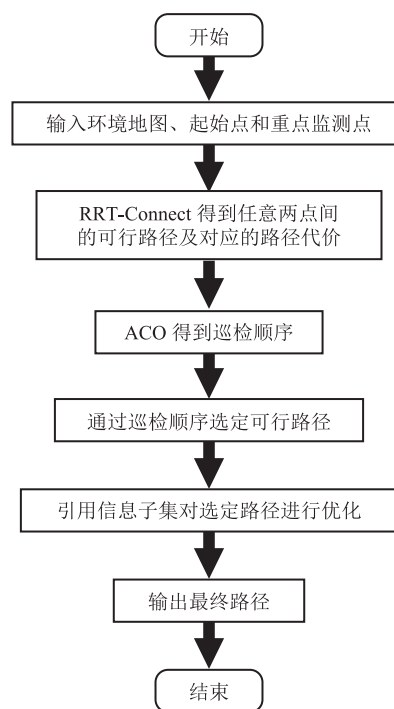


图 1 方法流程图

Fig. 1 Method flow chart

2.1 RRT-Connect-ACO 算法

假设一个具有 n 个需要重点监测节点 x 的环境地图 X ，每对节点通过 $\text{Edge}(x_i, x_j)$ 连接，由二进制变量 x_{ij} 判定每条边是否属于巡检路径，当 $x_{ij}=1$ 时， $\text{Edge}(x_i, x_j)$ 属于巡检路径，否则不属于。每个 $\text{Edge}(x_i, x_j)$ 包含信息素追踪强度 τ_{ij} 和常

量启发值 η_{ij} , τ_{ij} 在每次迭代的时候由蚂蚁更新。其中, i 和 j 为重点监测点对应的节点序号。算法的目标是使最终巡检路径的总路径代价最小化, 如公式(1)所示。

$$\min \sum_{i=1}^n \sum_{j \neq i, j=2}^n c_{ij} x_{ij} \quad (1)$$

s.t.

$$x_{ij} \in \{0, 1\} \quad i, j=1, 2, \dots, n \quad i \neq j \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j=1, 2, \dots, n \quad i \neq j \quad (3)$$

如公式(2)所示, x_{ij} 取值 0 或 1, 表示节点 i 和 j 之间的路径是否被选择; 公式(3)表示任意节点为端点的路径只有一条被选中, 保证每个节点只经过一次。

将 m 个蚂蚁放置在随机选择的节点上, 信息素 τ_0 的初始值赋予在每条边上: $\tau_{ij} = \tau_0$, 启发值 η_{ij} 采用节点 x_i 和 x_j 之间路径代价 c_{ij} 的倒数进行初始化, 如公式(4)所示。

$$\eta_{ij} = \frac{1}{c_{ij}} \quad (4)$$

节点 x_i 上的蚂蚁 k 移动到下一个节点 x_j 的概率计算公式如下:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in N_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad (5)$$

其中, α 为信息启发因子; β 为期望启发因子; N_i^k 为节点 x_i 上蚂蚁 k 尚未访问的节点集。 α 和 β 是两个正参数, 分别用于确定 τ_i 和 η_{ij} 的权重, α 相比于 β 的值越大, 蚁群选择之前走过的路径可能性越大, 值越小, 蚁群容易选择局部最短路径, 减少搜索范围, 容易陷入局部最优。

当所有的蚂蚁根据以上约束完成一次旅行后, 根据公式(6)更新信息素。

$$\tau_{ij} = \rho \tau_{ij} + \Delta \tau_{ij} \quad (6)$$

其中, ρ 为系数, 取值范围为 $0 \leq \rho < 1$; $1 - \rho$ 为信息素的蒸发率; $\rho \tau_{ij}$ 为保留的上一次旅行的信息

素浓度; $\Delta \tau_{ij}$ 的计算公式如下。

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (7)$$

$\Delta \tau_{ij}^k$ 为第 k 只蚂蚁从节点 x_i 移动到 x_j 留下的信息素, 计算公式如下:

$$\Delta \tau_{ij}^k = \begin{cases} \frac{1}{c_{ij}}, & \text{if the } k\text{th ant traverses } (x_i, x_j) \\ 0, & \text{others} \end{cases} \quad (8)$$

传统方法大多将巡检问题视作旅行商问题, 但在一些复杂环境中, 仅以两点间的欧氏距离作为两点间的路径代价是不合理的, 易导致实际路径无法达到最优, 故本文将 RRT-Connect 得到的路径长度作为蚁群算法的代价输入。RRT-Connect 采用双向拓展平衡的连结型双树 RRT 算法, 该算法基于双向 RRT 算法并加入了贪心启发函数, 一定程度上提升了算法的收敛速度, 有利于快速得到路径代价。此外, 双向算法避免了终点与起点的不同选择导致路径存在差异的问题, 使路径代价更可靠。基本步骤如下:

(1) 在已知的空间环境 X 中, 定义起始点 x_{start} 、目标区域点 x_{goal} 和额定步长 λ , 初始化两棵扩展树 T_a 和 T_b 。

(2) 在空间中随机生成一个点 x_{rand} , 遍历扩展树 T_a , 找到距离 x_{rand} 最近的节点 x_{near} , 选择两点间连线上距离 x_{near} 为额定步长 λ 的点作为新节点 x_{new} 。

(3) 监测 x_{new} 与 x_{near} 之间是否与障碍物发生碰撞, 若未发生碰撞则将 x_{new} 加入扩展树 T_a 中, 否则重新执行步骤(2)。遍历扩展树 T_b , 找到距离 x_{new} 最近的节点 x'_{near} , 选择两点间连线上距离 x'_{near} 为额定步长 λ 的点作为新节点 x'_{new} 。

(4) 监测 x'_{new} 与 x'_{near} 之间是否与障碍物发生碰撞, 若未发生碰撞则将 x'_{new} 加入扩展树 T_b 中, 否则经过步骤(6)的判断后, 重新执行步骤(2)。

(5) 选择 x'_{new} 和 x_{new} 之间连线上距离 x'_{new} 为额定步长 λ 的点作为新节点 x''_{new} , 监测 x''_{new} 与

x'_{new} 之间是否与障碍物发生碰撞, 若未发生碰撞, 则将 x''_{new} 加入拓展树 T_b 中, 将 x''_{new} 重新定义为新的 x'_{new} , 重复执行步骤(5); 若监测到 x''_{new} 与 x'_{new} 之间与障碍物发生碰撞, 则跳出循环, 执行下一步。若 x'_{new} 和 x''_{new} 是相同节点, 则结束流程, 依次寻找父节点并进行连接, 即完成起始点到目标点的路径规划; 否则, 重复当前步骤。

(6) 若 T_b 的节点数小于 T_a 的节点数, 则交换两棵扩展树 T_a 和 T_b , 并重新执行步骤(2)。

RRT-Connect-ACO 算法的伪代码如算法 1 所示。其中, x 为目标点列表, Γ_f 为经过所有目标点的可行路径, 将任意两点间的路径和代价进行初始化, RRT_Connect(\cdot)可得到 x_i, x_j 两点之间的可行路径 Γ_{ij} , Cost(\cdot)返回生成的路径的长度 c_{ij} , 将点列表 x 和代价矩阵 c 作为输入, 通过 ACO(\cdot)可以得到目标点的巡检顺序 o_x , 使用 Path(\cdot)根据巡检顺序和已经生成的路径可得到一条可行的巡检路径。

算法 1 RRT-Connect-ACO Algorithm

```

输入:  $x \leftarrow$  goal point list
输出: A feasible path  $\Gamma_f$  through all the goal points;
1: init:  $\Gamma_{\text{init}}(x), c_{\text{init}}(x)$ 
2: for  $i=1$  to number of  $x$  do:
3:   for  $j=i+1$  to number of  $x$  do:
4:      $\Gamma_{ij} = \Gamma_{ji} = \text{RRT\_Connect}(x_i, x_j)$ 
5:      $c_{ij} = c_{ji} = \text{Cost}(\Gamma_{ij})$ 
6:   end
7: end
8:  $o_x = \text{ACO}(x, c)$ 
9:  $\Gamma_f = \text{Path}(\Gamma, o_x)$ 

```

由于 RRT-Connect-ACO 算法具备快速性和低成本性, 本文采用这种算法对任意两中间监测位置的路径代价进行粗略估计, 与传统方法中直接采用两点间欧氏距离作为路径代价相比, 该方法更具可靠性。此外, 由于生成路径的可行性, 提高了优化效率。

2.2 Informed RRT*-Connect 算法

RRT-Connect-ACO 算法可用于确定机器人访

问重点监测地区的顺序和可行路径, 将该条路径进行优化可得到一条最优路径。

针对提供环境地图较大、排序后路径经过区域较小的问题, 本文采用 RRT* 算法对自由空间 X_{free} 进行均匀采样, 但导致搜索树上生成很多冗余的分支, 造成 RRT* 收敛速度相对较慢。本文引用信息子集^[25]的概念, 它是状态空间 X 的子集, 利用椭圆采样代替了全局均匀采样。

代价函数 $f(x)$ 可分为两个部分: 从 x_{start} 到 x 的路径代价 $g(x)$ 以及从 x 到 x_{goal} 的路径代价 $h(\cdot)$ 。为得到估计代价 $\hat{f}(\cdot)$, 需要定义从起始点到 x 的代价估算 $\hat{g}(\cdot)$ 和 x 到终止点的代价估算 $\hat{h}(\cdot)$ 。

欧几里得距离是寻找路径最小长度问题的启发式方法, 因此, 改善当前解决方案最优代价 c_{best} 的椭圆信息子集可被定义如公式(9)所示, 即椭圆的一般方程。

$$X_f = \left\{ x \in X \mid \|x_{\text{start}} - x\|_2 + \|x - x_{\text{goal}}\|_2 \leq c_{\text{best}} \right\} \quad (9)$$

图 2 是以 x_{start} 和 x_{goal} 为焦点的椭圆信息子集, 长轴的长度为 c_{best} , 短轴的长度为 $\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}$ 。其中, c_{min} 为 x_{start} 和 x_{goal} 之间的欧几里得距离。

每次迭代过程均在椭圆信息子集 X_f 内均匀采样, 通过重新选择父节点和重布线进行优化, 计算优化后的估计代价 $\hat{f}(\cdot)$, 取其中最优化代价定义为 c_{best} 。若 $c_{\text{best}} < c_{\text{min}}$, 则更新最优路径信息和椭圆采样区域。

搜索新生成的节点 x_{new} 额定半径内的节点,

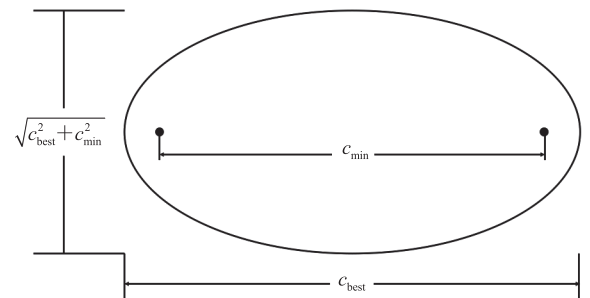


图 2 椭圆信息子集示例图

Fig. 2 Sample diagram of elliptic informed subset

排除其中经过障碍物的节点, 将符合要求的节点加入节点列表 X_{near} 。

重新选择父节点的过程为: 计算节点 x_{new} 到起始点 x_{start} 的路径代价 c_{min} ; 依次选择 X_{near} 中的节点作为节点 x_{new} 的父节点, 计算对应的路径代价, 取其中最小值 c_{new} , 若 $c_{new} < c_{min}$, 那么 $c_{min} = c_{new}$, x_{new} 的父节点更改为 X_{near} 中的对应节点 x_{near} 。

重布线随机树的过程为: 计算 X_{near} 中每个节

算法 2 Informed RRT*-Connect Algorithm

输入: $X \leftarrow$ Environmental Map, $T \leftarrow \Gamma$, $X_{soln} \leftarrow \Gamma$,

$\lambda \leftarrow$ Step Length, $n \leftarrow$ Number of Iterations

输出: An optimal path Γ_o from x_{start} to x_{goal}

```

1: init:  $x_{start} \leftarrow T_{first}$ ,
2: init:  $x_{goal} \leftarrow T_{last}$ 
3: init:  $c_{min} \leftarrow Cost(x_{goal}, x_{start})$ 
4: for  $i=1$  to  $n$  do:
5:    $c_{best} = \min_{x_{soln} \in X_{soln}} \{Cost(x_{soln})\}$ 
6:    $x_{best} \leftarrow x_{c_{best}}$ 
7:    $x_{rand} \leftarrow Sample(X, c_{best})$ 
8:    $x_{nearest} \leftarrow Nearest(x_{rand}, T)$ 
9:    $x_{new} \leftarrow Steer(x_{rand}, x_{nearest}, \lambda)$ 
10:  if CollisionFree( $x_{new}, x_{nearest}$ ) then:
11:     $T.addNode(x_{new})$ 
12:     $X_{near} \leftarrow Near(x_{new}, T)$ 
13:     $c_{min} = Cost(x_{nearest}) + Line(x_{nearest}, x_{new})$ 
14:    for  $x_{near} : X_{near}$  do:
15:       $c_{new} = Cost(x_{near}) + Line(x_{near}, x_{new})$ 
16:      if  $c_{new} < c_{min}$  then:
17:         $x_{new} \cdot parent \leftarrow x_{near}$ 
18:         $c_{min} \leftarrow c_{new}$ 
19:    end
20:  end
21:  for  $x_{near} : X_{near}$  do:
22:     $c_{near} = Cost(x_{near})$ 
23:     $c_{new} = Cost(x_{new}) + Line(x_{new}, x_{near})$ 
24:    if  $c_{new} < c_{near}$  then:
25:       $x_{near} \cdot parent \leftarrow x_{new}$ 
26:    end
27:  end
28: end
29: end
30:  $\Gamma_o \leftarrow ExtractPath(x_{best})$ 

```

点到起始点的代价 c_{near} , x_{new} 到起始点的代价与到 x_{near} 的代价之和 c_{new} , 若 $c_{near} > c_{new}$, 那么对应节点 x_{near} 的父节点将更改为 x_{new} 。

Informed RRT*-Connect 算法的伪代码如算法 2 所示。

3 仿真与分析

为验证本算法性能, 本实验通过两个方案进行仿真分析。由于机器人存在自身半径, 无法直接视作一个质点, 仿真实验中对障碍物进行了膨化处理, 进而可将机器人看作一个质点。

仿真地图如图 3 所示, 所有环境的地图规模均为 $500 \text{ m} \times 300 \text{ m}$, 以地图左下角为坐标原点, 最优路径长度为 l_{opt} , 最大迭代次数为 n 。

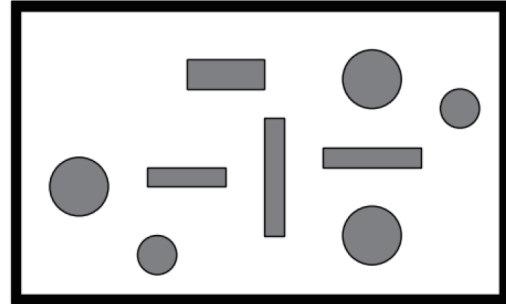


图 3 仿真地图

Fig. 3 The simulation map

首先通过 RRT-Connect-ACO 得到一条粗略的可行巡检路径, 如图 4 所示。

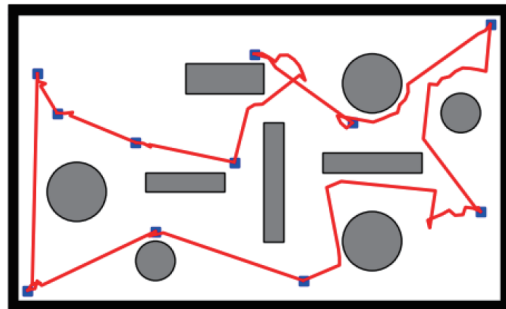


图 4 使用 RRT-Connect-ACO 得到可行路径

Fig. 4 Feasible path obtained by RRT-Connect-ACO

利用 Informed RRT*-Connect 算法对该路径进

行优化, 最终得到一条最优路径, 如图 5 所示。

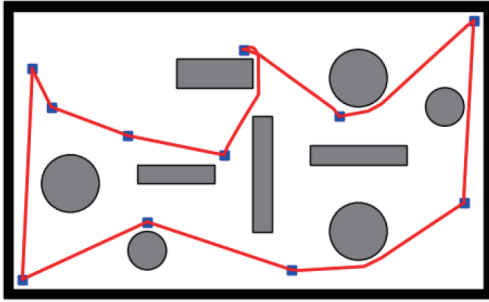


图 5 优化后的巡检路径

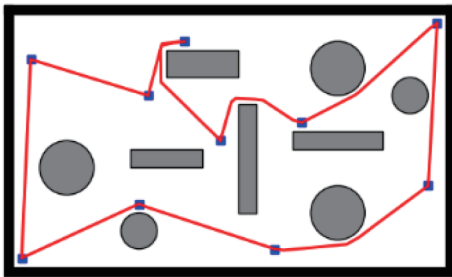
Fig. 5 Optimized inspection path

Wang 等^[15]研究的应用场景为较空旷的地方, 障碍物较少, 两点之间的欧氏距离与实际路径代价基本一致, 所以其采用欧氏距离作为路径预估代价, 对实验结果影响不大。对于如图 3 所示地图, 采用欧氏距离和使用 RRT-Connect 实际可行路径长度作为预估代价的结果也相差较小, 如图 6 所示。

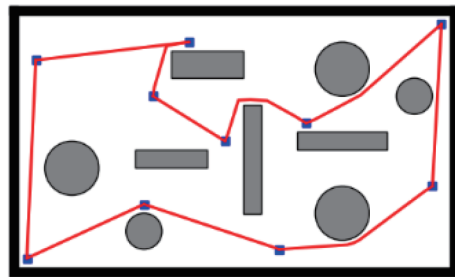
图 6(a) 的路径成本 $l_{opt}=1\ 590.2\ \text{m}$, 图 6(b) 的路径成本 $l_{opt}=1\ 558.4\ \text{m}$, 与直接采用欧氏距离的方案相比, 采用 RRT-Connect 实际路径作为预估代价的路径成本略低, 但优势不明显。

但对于一些特殊地形(图 7), 最终得到的路径结果差异明显。图 7(a) 为采用欧氏距离作为预估代价最终得到的结果 $l_{opt}=1\ 909.7\ \text{m}$, 忽略了地形对巡检路径的影响, 导致大量重复路径, 增加了路径代价。图 7(b) 为采用 RRT-Connect 实际路径长度作为预估代价最终得到的结果 $l_{opt}=1\ 728.7\ \text{m}$, 该结果明显优于采用欧氏距离得到的结果。两幅图的差异主要集中在矩形框标注部分, 本文算法考虑了地形对巡检路径的影响, 得到了路径更短的巡检顺序。

由于直接采用欧氏距离作为预估代价, 没有生成可行路径的过程, 需直接采用 Informed RRT* 进行路径规划, 即根据欧氏距离得到的路



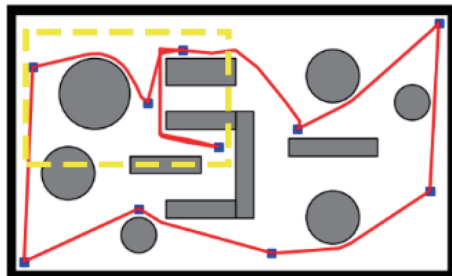
(a) 采用欧氏距离作为预估代价的最优巡检路径



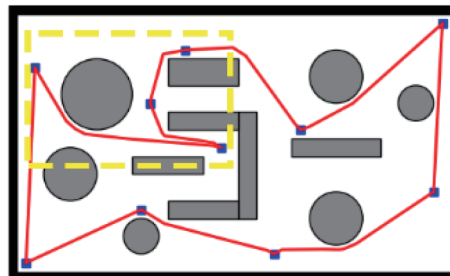
(b) 采用 RRT-Connect 实际路径长度作为预估代价的最优巡检路径

图 6 分别以欧氏距离与实际路径长度作为代价所生成的最终路径

Fig. 6 The final paths generated at the cost of Euclidean distance and actual path length, respectively



(a) 采用欧氏距离作为预估代价的最优巡检路径



(b) 采用 RRT-Connect 实际路径长度作为预估代价的最优巡检路径

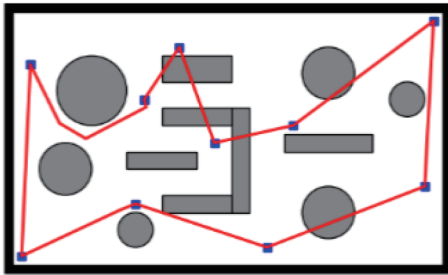
图 7 特殊地形下, 分别以欧氏距离与实际路径长度作为代价所生成的最终路径

Fig. 7 Under the special map, the final paths generated at the cost of Euclidean distance and actual path length, respectively

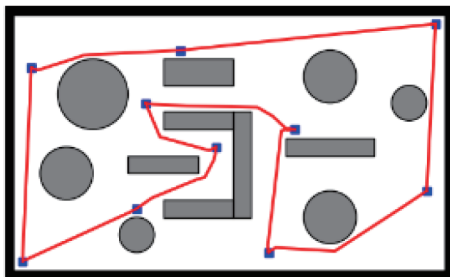
径巡检顺序, 采用 Informed RRT* 进行两点之间的路径规划。在采用相同的迭代次数时, 则更易出现规划路径失败的结果。

在较少迭代次数下, 以欧氏距离与实际路径长度为预估代价生成的最终路径对比图如图 8 所示。其中, 图 8(a) 为采用欧氏距离作为预估代价的最优巡检路径, 图 8(b) 为采用 RRT-Connect 实际路径长度作为预估代价的最优巡检路径。

如图 8 所示, 当 $n=500$ 时, 图 8(a) 存在 5 条未规划出的路径, 由于其 Informed RRT* 属于单树的 RRT 算法, 一边扩展一边优化, 提高了优化的效率, 但极有可能在迭代次数较小的情况下不能生成一条可行路径。故相同迭代次数的情况下, 本文提出的算法更易得到较优路径。



(a) 采用欧氏距离作为代价的最优巡检路径



(b) 采用 RRT-Connect 实际路径长度作为代价的最优巡检路径

图 8 在较少迭代次数的情况下, 以欧氏距离与实际路径长度作为代价所生成的最终路径

Fig. 8 In the case of fewer iterations, the final paths generated at the cost of Euclidean distance and actual path length, respectively

此外, 对优化过程进行对比, 采用图 7 中的地图, 对于生成的同一条可行路径(路径长度为

$l=2\ 108.4\text{ m}$), 都采用重新选择父节点与重布线随机树的方式进行路径优化, 在相同的迭代次数 $n=5\ 000$ 的条件下, 将引入信息子集的路径优化结果与未引入信息子集的结果进行对比实验, 每种方法运行 50 次, 实验结果如图 9 所示。

如图 9 所示, 其中, 左侧为采用信息子集的数据结果, 右侧为未采用的数据结果, 两种方法的平均长度分别为 $1\ 775.6\text{ m}$ 和 $1\ 795.2\text{ m}$ 。采用信息子集的方法虽然异常值较多, 但优化结果优于未采用信息子集的方法, 因此, 采用信息子集的算法得到的结果更优。

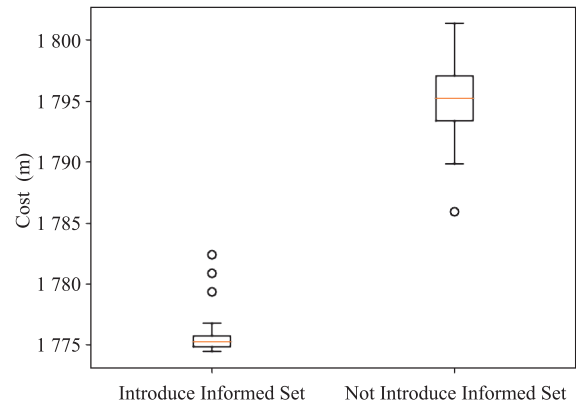


图 9 是否引入信息子集的路径优化结果对比

Fig. 9 Comparison of path optimization results wheater to introduce informed subset

综上所述, 本文方法可以生成一条最优巡检路径。此外, 与其他方法对比, 本文方法对巡检顺序的优化得益于可行路径的生成, 考虑到了障碍物对计算两点间路径代价的影响, 得到了可以生成更短路径的巡检顺序。而信息子集的引入, 克服了基本的 RRT 优化方法随机性强、采样效率低的缺点, 在保证较少迭代次数的前提下, 不仅生成了可行路径, 还进一步提高了优化效率。

4 结 论

针对巡检机器人多目标复杂路径规划难题, 本文提出一种多目标快速探索随机树巡检路径优

化方法。首先利用 RRT-Connect-ACO 算法快速得到一条粗略的巡检路径,提高了巡检顺序的可靠性;然后利用 Informed RRT*-Connect 算法优化路径,以获取最优路径。对于大的子集,当解决方案接近理论最小值时,通过全局采样或均匀采样改善解决方案的可行性较小。本文采用椭圆信息子集,将子集范围缩小,提高了优化效率。值得注意的是,本文方法仅针对静态情况,对环境信息发生的变化无法动态改变路径,未来可根据巡检结果进行进一步动态巡检,使巡检过程更加智能高效。

参 考 文 献

- [1] Hoshino S, Ishiwata T, Ueda R. Optimal patrolling methodology of mobile robot for unknown visitors [J]. *Advanced Robotics*, 2016, 30(16): 1072-1085.
- [2] Wang JH, Fu G, Yan MW. Comparative analysis of two catastrophic hazardous chemical accidents in China [J]. *Process Safety Progress*, 2020, 39(1): e12137.
- [3] Halim AH, Ismail I. Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem [J]. *Archives of Computational Methods in Engineering*, 2019, 26(2): 367-380.
- [4] Wang JK, Meng QH. Optimal path planning using generalized voronoi graph and multiple potential functions [J]. *IEEE Transactions on Industrial Electronics*, 2020, 67(12): 10621-10630.
- [5] Khanmirza E, Haghbeigi M, Nazarahari M, et al. A comparative study of deterministic and probabilistic mobile robot path planning algorithms [C] // *Proceedings of the 2017 5th RSI International Conference on Robotics and Mechatronics*, 2017: 534-539.
- [6] Poikonen S, Golden B, Wasil EA. A branch-and-bound approach to the traveling salesman problem with a drone [J]. *INFORMS Journal on Computing*, 2019, 31(2): 193-410.
- [7] Chauhan C, Gupta R, Pathak K. Survey of methods of solving TSP along with its implementation using dynamic programming approach [J]. *International Journal of Computer Applications*, 2012, 52(4): 12-19.
- [8] İlhan İ, Gökmen G. A list-based simulated annealing algorithm with crossover operator for the traveling salesman problem [J]. *Neural Computing and Applications*, 2022, 34(10): 7627-7652.
- [9] Behmanesh R, Rahimi I, Gandomi AH. Evolutionary many-objective algorithms for combinatorial optimization problems: a comparative study [J]. *Archives of Computational Methods in Engineering*, 2021, 28(2): 673-688.
- [10] Zhou YM, Xu WQ, Fu ZH, et al. Multi-neighborhood simulated annealing-based iterated local search for colored traveling salesman problems [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(9): 16072-16082.
- [11] Liu JJ, Li WZ. Greedy permuting method for genetic algorithm on traveling salesman problem [C] // *Proceedings of the 2018 8th International Conference on Electronics Information and Emergency Communication*, 2018: 47-51.
- [12] Hussain A, Muhammad YS, Nauman SM, et al. Genetic algorithm for traveling salesman problem with modified cycle crossover operator [J]. *Computational Intelligence and Neuroscience*, 2017, 2017: 7430125.
- [13] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [C] // *Proceedings of the First European Conference on Artificial Life*, 1991: 134-142.
- [14] Du PZ, Liu N, Zhang HF, et al. An improved ant colony optimization based on an adaptive heuristic factor for the traveling salesman problem [J]. *Journal of Advanced Transportation*, 2021, 2021: 6642009.
- [15] Wang JK, Meng QH. Real-time decision making and path planning for robotic autonomous luggage trolley collection at airports [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 52(4): 2174-2173.

- [16] 郭耕辰, 冯良炳, 邓亮, 等. 基于A*算法与自适应分片的大规模最优路径规划 [J]. 集成技术, 2014, 3(2): 68-77.
Guo GC, Feng LB, Deng L, et al. Large scale route planning A* algorithm based on self-adaptive hierarchy method [J]. Journal of Integration Technology, 2014, 3(2): 68-77.
- [17] Zhang J, Wu J, Shen X, et al. Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star [J]. International Journal of Advanced Robotic Systems, 2021, 18(5). <https://doi.org/10.1177/17298814211042730>.
- [18] Liu LS, Lin JF, Yao JX, et al. Path planning for smart car based on Dijkstra algorithm and dynamic window approach [J]. Wireless Communications and Mobile Computing, 2021, 2021(4): 1-12.
- [19] Chen G, Luo N, Liu D, et al. Path planning for manipulators based on an improved probabilistic roadmap method [J]. Robotics and Computer-Integrated Manufacturing, 2021, 72: 102196.
- [20] Alarabi S, Luo CM, Santora M. A PRM approach to path planning with obstacle avoidance of an autonomous robot [C] // Proceedings of the 2022 8th International Conference on Automation, Robotics and Applications, 2022: 76-80.
- [21] Kuwata Y, Karaman S, Teo J, et al. Real-time motion planning with applications to autonomous urban driving [J]. IEEE Transactions on Control Systems Technology, 2009, 17(5): 1105-1118.
- [22] Kuffner JJ, Lavalley SM. RRT-Connect: an efficient approach to single-query path planning [C] // Proceedings of the 2000 IEEE International Conference on Robotics and Automation, 2000: 995-1001.
- [23] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning [J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.
- [24] Gammell JD, Srinivasa SS, Barfoot TD. Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic [C] // Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014: 2997-3004.
- [25] Mashayekhi R, Idris MYI, Anisi MH, et al. Informed RRT*-Connect: an asymptotically optimal single-query path planning method [J]. IEEE Access, 2020, 8: 19842-19852.