

基于Maklink图与Boustrophedon路径的机器人二维全覆盖路径规划算法

李伯尧, 赵斌斌, 陶明杰, 陈露

引用本文

李伯尧, 赵斌斌, 陶明杰, 陈露. 基于Maklink图与Boustrophedon路径的机器人二维全覆盖路径规划算法[J]. 计算机科学, 2026, 53(4): 78-87.

LI Boyao, ZHAO Binbin, TAO Mingjie, CHEN Lu. Mobile Robot Two-dimensional Full Coverage Path Planning Algorithm Based on MaklinkDiagram and Boustrophedon Path [J]. Computer Science, 2026, 53(4): 78-87.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于无冲突路径算法的多目标智能仓储路径规划](#)

Multi-objective Intelligent Warehousing Path Planning Based on Conflict Free Path Algorithm
计算机科学, 2026, 53(4): 88-100. <https://doi.org/10.11896/jsjcx.250200035>

[融合改进A*算法和TEB算法的AGV路径规划](#)

Path Planning for AGV Integrating Improved A* Algorithm and TEB Algorithm
计算机科学, 2025, 52(11A): 240900148-7. <https://doi.org/10.11896/jsjcx.240900148>

[基于蚁群混合势场法的无人机路径规划](#)

UAV Path Planning Method Based on Ant Colony Mixed Potential Field Method
计算机科学, 2025, 52(11A): 241100179-6. <https://doi.org/10.11896/jsjcx.241100179>

[基于改进Dueling-DQN的多无人机路径规划算法](#)

Multi-UAV Path Planning Algorithm Based on Improved Dueling-DQN
计算机科学, 2025, 52(8): 326-334. <https://doi.org/10.11896/jsjcx.240600104>

[基于拥塞感知和缓存通信的多智能体路径规划](#)

Congestion-aware and Cached Communication for Multi-agent Pathfinding
计算机科学, 2025, 52(8): 317-325. <https://doi.org/10.11896/jsjcx.240900012>

基于 Maklink 图与 Boustrophedon 路径的移动机器人二维全覆盖路径规划算法

李伯尧¹ 赵斌斌² 陶明杰³ 陈露⁴

1 中国政法大学商学院 北京 100088

2 成都理工大学数学科学学院 成都 610051

3 成都理工大学物理学院 成都 610051

4 成都理工大学计算机与网络安全学院 成都 610051

(liboyao@cupl.edu.cn)

摘要 随着复杂障碍物环境下移动机器人的全覆盖路径规划在生产巡检、家庭卫生等领域的应用越来越广泛,现有方法存在的重复覆盖率高、子区域间转换路径复杂以及对凹多边形障碍物适应性不足等问题愈发凸显。因此,提出一种融合 Maklink 图论、改进蚁群算法和 Boustrophedon 路径的移动机器人全覆盖路径规划方法。该方法首先利用 Maklink 图论构建环境模型,生成链接线,利用链接线将二维空间划分为多个凸多边形子区域并构建初步可行路径网络;其次,将子区域访问顺序建模为广义的 TSP 问题,利用一维蚁群算法获取子区域间的访问序列;然后,结合求最小函数值的蚁群算法与三角剪枝几何优化,得到子区域间最优转换路径;最后,按照访问顺序在各子区域内采用 Boustrophedon 路径进行“弓”字形路径遍历,形成全局覆盖路径。在多个不同二维复杂度环境中的仿真实验表明,所提方法可有效适应存在多种多边形障碍物的环境,覆盖率均可达 100%,重复率为 0。与传统蚁群算法和改进蚁群算法两种单一算法的对比实验表明,该算法在转换路径长度、遍历路径长度及重复率三方面均具有较好的表现;与传统利用栅格法构建环境模型的全遍历方法的对比结果表明,该算法建模精度高,存储效率优。

关键词: Maklink 图; Boustrophedon 路径; 蚁群算法; 路径规划; 全遍历

中图分类号 TP242

Mobile Robot Two-dimensional Full Coverage Path Planning Algorithm Based on Maklink Diagram and Boustrophedon Path

LI Boyao¹, ZHAO Binbin², TAO Mingjie³ and CHEN Lu⁴

1 School of Business, China University of Political Science and Law, Beijing 100088, China

2 School of Mathematical Sciences, Chengdu University of Technology, Chengdu 610051, China

3 School of Physics, Chengdu University of Technology, Chengdu 610051, China

4 School of Computer and Cyber Security, Chengdu University of Technology, Chengdu 610051, China

Abstract With the increasingly widespread application of complete coverage path planning of mobile robots in production inspection and home cleaning fields, the problems existing in the current algorithms are becoming increasingly obvious, such as high repetitive coverage rate, non-optimal transition paths between sub-regions, and insufficient adaptability to concave polygonal obstacles. Therefore, this paper proposes a two-dimensional complete coverage path planning algorithm for mobile robots, which integrates Maklink graph theory, improved ant colony algorithm and Boustrophedon path. Firstly, the method employs Maklink graph theory to construct an environmental model by generating link lines. These lines are then used to partition the two-dimensional space into multiple convex polygonal subregions and establish an initial feasible path network. Secondly, the method transforms the connection sequence of the sub-regions into a generalized “Traveling Salesman Problem” and uses one-dimensional ant colony algorithm to compute the sequence of the sub-regions that the robot visits. Then, the ant colony algorithm for minimizing results of function and triangular pruning geometric optimization algorithm is applied to obtain the optimal transition paths between sub-regions. Finally, the method performs a zigzag traversal within each sub-region through the visiting sequence by the

到稿日期:2025-07-31 返修日期:2025-12-23

基金项目:国家自然科学基金青年项目(80303-AZ20240001);中国政法大学青年教师学术创新团队支持计划(25CXTD04);成都理工大学人才建设科研启动基金(10912-KYQD2024-10450)

This work was supported by the Youth Project of the National Natural Science Foundation of China(80303-AZ20240001), Program for Young Innovative Research Team in China University of Political Science and Law(25CXTD04) and Chengdu University of Technology Talent Construction Research Start-up Fund(10912-KYQD2024-10450).

通信作者:赵斌斌(zhaobinbin@stu.cdut.edu.cn)

Boustrophedon path in order to achieve global coverage path planning. Simulation experiments conducted in multiple two-dimensional environments with varying complexities demonstrate that the proposed method can effectively adapt to scenarios containing a variety of polygonal obstacles, achieving a coverage rate of 100% with zero redundancy. Meanwhile, the comparative experiments with only using the traditional ant colony algorithm and the improved ant colony algorithm, this algorithm performs well in three aspects, which are the length of the transformation path, the length of the traversal path, and the repetition rate. The comparison with the traditional full traversal method of constructing environmental models using the raster method shows that the proposed algorithm has high modeling accuracy and excellent storage efficiency.

Keywords Maklink diagram, Boustrophedon path, Ant colony algorithm, Path planning, Full traversal

1 引言

近年来,人工智能算法与应用技术研究方兴未艾,移动机器人作为人工智能技术的代表性载体之一,已成为全球科技界竞相探索的重要前沿领域^[1]。这使得移动机器人被赋予越来越多的功能和特质,被应用于越来越多的领域,如基础设施检查^[2-3]、应急现场救援^[4-5]和特殊环境维护^[6]。在日常应用场景中也较为常见,如清洁机器人^[7]、割草机器人^[8]、农业喷洒无人机^[9]等。上述机器人正常运行的关键在于完成完整的全覆盖路径规划,以确保机器人导航并覆盖指定区域的每个部分^[10]。然而,环境的复杂性以及对不同表面和障碍物的优化和适应性需求,使其成为一项持续的研究挑战。

当前,学者针对移动机器人的路径规划问题进行了大量研究工作,提出或改进了诸多算法^[11],包括 A* 算法^[12]、Dijkstra 算法^[13]、快速扩展随机树算法^[14](RRT)、随机路标图算法^[15](PRM)、人工势场法^[16](APF)等传统算法,以及遗传算法^[17](GA)、人工鱼群算法^[18](AFS)、粒子群算法^[19](PSO)、神经网络算法^[20](NN)、蚁群算法^[21](ACO)等智能优化算法^[22]。上述算法中,蚁群算法作为一种启发式搜索算法,具有鲁棒性强,易与其他算法相结合等优势,成为当今人工智能研究的一个热点^[23]。然而,经典蚁群算法存在信息素积累和更新时间长导致其收敛速度慢、局部信息素过高导致易陷入局部最优解等问题。综合其利弊,许多源于蚁群模型设计的改进算法越来越多地被应用于各个领域的研究,尤其在移动机器人路径规划领域得到了充分的应用和发展^[24]。

实现移动机器人在二维地图中的遍历和避障路径规划的首要任务是将连续的空间离散化,建立移动机器人作业空间模型^[25]。在实现这一目的的主流方法中,可视图法(Visibility Graph)^[26]的局限性相对明显,当机器人的起始点发生变化时需重新构图,灵活性较低;而现有文献中应用最为广泛的栅格法(Grid method)^[27],对于复杂的二维环境模型,例如不规则障碍物、凹凸多边形障碍物,具有地图信息存储量大、精确度低、决策效率低、抗干扰能力弱等缺点。相比之下,将 Maklink 图论法用于复杂的二维环境建模,可有效降低信息存储量,提高精确度,提升决策效率,增强鲁棒性等^[28]。因此本文采用 Maklink 图论法构建环境模型。该方法作为一种改进的自由空间法,使用 Maklink 链接线连接不同障碍物的顶点,这些链接线不经过障碍物,与障碍物边界构成的外角不大于 180°且链接线两两不相交,将作业空间分割成多个凸多边形来构建移动机器人的可遍历子区域。

为实现移动机器人在二维地图中的全覆盖路径规划,主

要遍历方法包括矩形螺旋路径^[29]、希尔伯特曲线路径^[30]、Boustrophedon 路径^[31]等。Boustrophedon 路径在覆盖领域应用最为广泛,原因在于其可以进行模块化设计,从而达到模块间的首尾相接。矩形螺旋路径虽然在路径总长度和直角转向次数等方面均接近 Boustrophedon 路径,但是其规划路径在形成全覆盖方面的效率较低,该方法一般作为 Boustrophedon 路径的补充。希尔伯特曲线路径由于其整个覆盖路径中存在过多的直角转向,且盲区数量与“牛耕式”路径和矩形螺旋路径相近,在以上 3 种方法中性能最差,应用也最少^[32]。

基于上述研究现状,本文选择在利用 Maklink 图论法构建环境模型的基础上,提出一种基于求最小函数值的蚁群算法与 Boustrophedon(牛耕式)路径的移动机器人二维全覆盖路径规划策略。此外,子区域间转换路径还采用了三角剪枝的几何优化,使得由蚁群算法计算得到的较优转换路径为最优路径,以期最大程度地降低路径重复率、提高工作效率。本文利用上述方法进行仿真实验,结果表明:按照某些传统计算重复率的方法^[33],子区域间转换路径不计入重复路径,本文的重复率仅为 0%;若将子区域转换路径计入重复率,该方法的覆盖率可达 100%,重复率仅为 5.80%。

2 环境建模

本文假设移动机器人在静态二维空间中移动,障碍物用凹凸多边形来表示,障碍物的高度垂直于 XY 平面,将障碍物的边界依据机器人的最大尺寸和机器人正常感知所需的最小范围进行扩展,并将移动机器人简化为一个质点,对实验环境空间做以下假设:环境空间中存在 m 个障碍物,第 i 个障碍物 O_i 有 n_i ($n_i \geq 3$) 个顶点,则障碍物分别表示为:

$$O_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{in_i}, y_{in_i})\} \quad (1)$$

$$W = \{O_1, O_2, \dots, O_m\} \quad (2)$$

其中, (x_{ij}, y_{ij}) , $i=1, 2, \dots, m, j=1, 2, \dots, n_i$ 表示第 i 个障碍物第 j 个顶点的坐标。随后利用 Maklink 图论自由链接线的原理与基本要求进行 Maklink 图的构建,自由链接线的建立参考 Maki K. Habib 教授在 1991 年的相关研究,具体构造步骤如下。

1) 将所有凹凸多边形障碍物的顶点组合成集合:

$$O_i = \{(x_{11}, y_{11}), (x_{12}, y_{12}), \dots, (x_{ij}, y_{ij}), \dots, (x_{mm}, y_{mm})\} \quad (3)$$

作顶点到边界的垂线,将所有点两两相连,并按从短到长的顺序排列,存储在集合 L 中。

2) 依次判断集合 L 中的线段是否与障碍物相交。若相交,则删除该线段;若不相交,则保留该线段。

3) 初始情况令所有障碍物均无链接线。

4) 选取一个无链接线的障碍物顶点 v_0 , 从集合 L 中选取与 v_0 相链接的线段组成集合 L_0 , 并按从短到长的顺序排列。依次选取 L_0 中的一条线段, 若出现外角均不大于 180° 的某条线段, 则该线段为顶点 v_0 的链接线, 放入集合 LL 中。

5) 若步骤 4) 中不存在满足条件的线段, 则依次选取 L_0 中的两条线段; 若出现外角均不大于 180° 的某两条线段, 则该两条线段为顶点 v_0 的链接线, 放入集合 LL 中。

6) 重复步骤 4) 和步骤 5), 直至所有顶点均有链接线为止。

7) 删除障碍物顶点的冗余链接线。

8) 给出所有链接线的中点, 并与路径起点及终点组合构成顶点集合 V 。

9) 将集合 V 中的顶点两两进行连线记为集合 E , 删除集合 E 中与障碍物、链接线相交的中点连线。

10) 集合 (V, E) 记为移动机器人的初步可行空间。

按上述步骤, 即可完成对 Maklink 图的构建。进一步地, 所述方法对于子区域的划分原理为: 在 Maklink 地图上, 由 Maklink 链接线、障碍物边界和二维地图边界围成的多块凸多边形为移动机器人的可遍历子区域, 如图 1 所示。

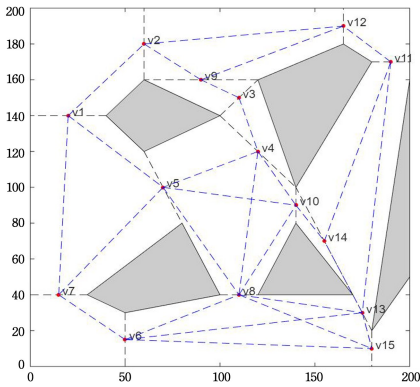


图 1 Maklink 图论对平面区域的划分(电子版为彩图)

Fig. 1 Maklink graph theory on the division of planar regions

图 1 中灰色区域表示二维地图中的障碍物, 灰色虚线为利用 Maklink 图论法构建的链接线, 红色点为各链接线上的中点, 蓝色虚线为各链接线上中点间的连接线。

3 子区域访问顺序的蚁群算法

3.1 蚁群算法

蚁群算法(Ant Colony Optimization, ACO)是一种模拟自然界蚂蚁觅食行为的优化算法, 它通过模拟蚂蚁在寻找食物过程中释放信息素并倾向于选择信息素浓度高的路径的特性, 解决如旅行商(TSP)等组合优化问题。本文主要通过蚁群算法实现两方面的功能: 1) 探究移动机器人在子区域间的访问顺序; 2) 探究移动机器人在子区域间转换的最短路径。

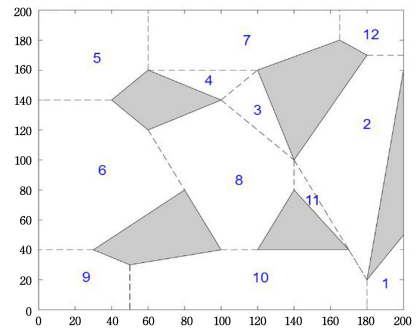
3.2 建立子区域访问顺序

如图 2 所示, 子区域共有 12 个, 记为 n 。图 2(a) 中的蓝色数字为子区域的初始编号, 子区域连接时规定它们的距离为 1, 子区域不连接时规定它们的距离为 0, 由此构建子区域之间的邻接矩阵 A 。

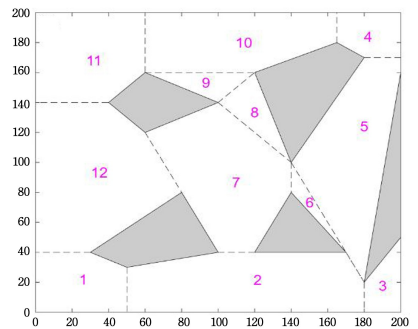
$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

利用 Dijkstra 算法(见 4.1 节)得到任意两个子区域间的最短距离 d_{ij} ($i, j=1, 2, \dots, n$), 如下所示:

$$D = \begin{bmatrix} 0 & 2 & 3 & 4 & 4 & 3 & 4 & 2 & 2 & 1 & 3 & 3 \\ 2 & 0 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 \\ 3 & 3 & 0 & 1 & 3 & 2 & 2 & 1 & 3 & 2 & 2 & 3 \\ 4 & 3 & 1 & 0 & 2 & 3 & 1 & 2 & 4 & 3 & 3 & 2 \\ 4 & 3 & 3 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & 2 \\ 3 & 3 & 2 & 3 & 1 & 0 & 2 & 1 & 1 & 2 & 2 & 3 \\ 4 & 2 & 2 & 1 & 1 & 2 & 0 & 3 & 3 & 3 & 3 & 1 \\ 2 & 2 & 1 & 2 & 2 & 1 & 3 & 0 & 2 & 1 & 1 & 3 \\ 2 & 2 & 3 & 4 & 2 & 1 & 3 & 2 & 0 & 1 & 3 & 3 \\ 1 & 1 & 2 & 3 & 3 & 2 & 3 & 1 & 1 & 0 & 2 & 2 \\ 3 & 1 & 2 & 3 & 3 & 2 & 3 & 1 & 3 & 2 & 0 & 2 \\ 3 & 1 & 3 & 2 & 2 & 3 & 1 & 3 & 3 & 2 & 2 & 0 \end{bmatrix}$$



(a) 子区域的原始编号



(b) 对子区域访问顺序进行编号

图 2 子区域间访问顺序编号变换与初始状态(电子版为彩图)

Fig. 2 Initial and final states of the sequence number transformation for the order of inter-subregion access

根据矩阵 D 可以求出子区域的访问顺序, 即将子区域的访问顺序转换为经典的旅行商问题。

设整个蚂蚁群体中蚂蚁数量为 m , t 时刻子区域 i 与子区域 j 连接路径上的信息素浓度为 $\tau_{ij}(t)$ 。初始时刻, 各个子区

域间连接路径上的信息素浓度相同,不妨设 $\tau_{ij}(t)=\tau_0$ 。

蚂蚁 $k(k=1,2,\dots,n)$ 根据各个子区域间连接路径上的信息素浓度决定下一次访问的子区域。设 $P_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 从子区域 i 转移到子区域 j 的概率,其计算式为:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & s \in allow_k \\ 0, & s \notin allow_k \end{cases} \quad (4)$$

其中, $\eta_{ij}(t)$ 为启发函数, $\eta_{ij}(t)=1/d_{ij}$ 表示蚂蚁从子区域 i 转移到子区域 j 的期望程度; $allow_k(k=1,2,\dots,m)$ 为蚂蚁 k 访问的子区域集合。开始时, $allow_k$ 中有 $n-1$ 个元素,即包括了蚂蚁 k 出发子区域的其他所有子区域;随着时间的推进, $allow_k$ 中的元素不断减少,直至为空,即表示所有的子区域均访问完毕。 α 为信息素重要程度因子,其值越大,表示信息素浓度在转移中起的作用越大; β 为启发函数重要程度因子,其值越大,表示启发函数在转移中的作用越大,即蚂蚁会以较大的概率转移到距离短的子区域。

蚁群算法使用信息素吸引蚂蚁搜索,信息素位置设定及更新方法对于蚁群算法的成功搜索具有重要的意义,各点信息素在蚂蚁经过后进行更新。信息素的更新包括局部更新与全局更新两部分。局部更新是指当蚂蚁经过该点时,该点的信息素就会减少,局部更新的目的是增加蚂蚁搜索未经过点的概率,达到全局搜索的目的。局部信息素更新随着蚂蚁的搜索进行,更新公式为:

$$\tau_{ij} = (1-\zeta)\tau_{ij} \quad (5)$$

其中, τ_{ij} 为子区域 i 与子区域 j 连接路径上所带的信息素值; ζ 为信息素的衰减系数。在蚂蚁释放信息素的同时,各个子区域间连接路径上的信息素浓度也在逐渐消失,设参数 ρ ($0<\rho<1$) 表示信息素的挥发程度。因此,当所有蚂蚁完成一次循环后,各个子区域间连接路径上的信息素浓度需进行实时更新,即:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \end{cases}, 0 < \rho < 1 \quad (6)$$

其中, $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在子区域 i 与子区域 j 连接路径上释放的信息素浓度; $\Delta\tau_{ij}$ 表示所有蚂蚁在子区域 i 与子区域

j 连接路径上释放的信息素浓度之和。路径较短的蚂蚁释放的信息素量较多,随着时间的推进,较短的路径上累积的信息素浓度逐渐增高,选择该路径的蚂蚁个数也越来越多。最终,蚁群会在正反馈的作用下集中在最佳的路径上,此时对应的便是带优化问题的最优解。在本文中,得到的子区域访问顺序如图 2(b) 所示。

4 子区域的转换路径

4.1 Dijkstra 计算初始路径

Dijkstra 算法作为一种标号法,给赋权图中的每一个顶点记一个数,称为顶点的标号,其中临时标号 T 表示从始顶点到该标点的最短路长的上界,固定标号 P 表示从始顶点到该顶点的最短路长。Dijkstra 的具体算法如下。

1) 给顶点 v_1 标 P 符号 $d(v_1)=0$, 给顶点 $v_j(j=2,3,\dots,n)$ 标符号 $d(v_j)=l_{1j}$ 。

2) 在所有 T 标号中取最小值,例如 $d(v_{j_0})=l_{1j_0}$, 则把 v_{j_0} 的 T 标号改为 P 标号,并重新计算具有 T 标号的其他各顶点的 T 标号;选顶点 v_j 的 T 标号 $d(v_j)$ 与 $d(v_{j_0})+l_{j_0j}$ 中较小者作为 v_j 的新的 T 标号。

3) 重复上述步骤,直到目标顶点的标号改为 P 标号。

基于 Dijkstra 算法的移动机器人规划路径如图 3 所示。

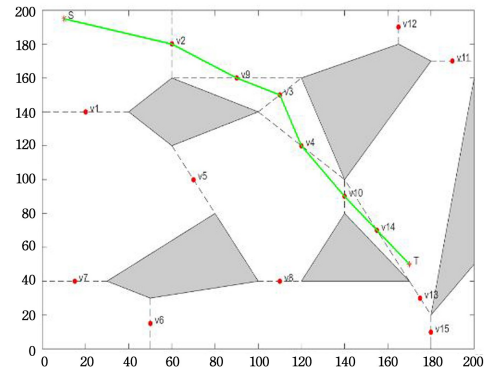


图 3 基于 Dijkstra 算法的初始路径规划

Fig. 3 Initial path planning based on Dijkstra algorithm

由图 3 得到 17×17 的邻接矩阵:

0	55.902	52.202	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55.902	0	56.569	0	0	64.031	0	100.12	0	0	0	0	0	0	0	0	0
52.202	56.569	0	0	0	0	0	0	0	36.056	0	0	105.48	0	0	0	0
0	0	0	0	31.623	0	0	0	0	22.361	0	0	0	0	0	0	0
0	0	0	31.623	0	53.852	0	0	80.623	0	36.056	0	0	0	0	0	0
0	64.031	0	0	53.852	0	0	81.394	72.111	0	70.711	0	0	0	0	0	0
0	0	0	0	0	0	0	43.012	65	0	0	0	0	125.9	0	130.1	0
0	100.12	0	0	0	81.394	43.012	0	0	0	0	0	0	0	0	0	0
0	0	0	0	80.623	72.111	65	0	0	0	58.31	0	0	65.765	0	76.158	0
0	0	36.056	22.361	0	0	0	0	0	0	0	0	80.777	0	0	0	0
0	0	0	0	36.056	70.711	0	0	58.31	0	0	0	0	0	25	0	0
0	0	0	0	0	0	0	0	0	0	0	0	32.016	140.8	105.95	0	121.66
0	0	105.48	0	0	0	0	0	80.777	0	32.016	0	0	0	0	0	0
0	0	0	0	0	0	125.9	0	67.765	0	0	140.8	0	0	44.721	20.616	20.616
0	0	0	0	0	0	0	0	0	0	25	105.95	0	44.721	0	0	25
0	0	0	0	0	0	130.1	0	76.158	0	0	0	0	20.616	0	0	0
0	0	0	0	0	0	0	0	0	0	0	121.66	0	20.616	25	0	0

4.2 建立求解最短路径的最小函数值蚁群算法

从路径起点 S 到终点 T 所经过的链接线依次是 L_1, L_2, \dots, L_d , 从链接线 L_{i-1} 到 L_i 有无数条路径选择。采用改进的蚁群算法, 给出起点 S 到链接线 L_1 , 链接线 L_1 到 L_2, \dots , 链接线 L_{i-1} 到 L_i, \dots, L_d 到 T 的最佳路径, 即为起点 S 到终点 T 的最佳路径。改进的蚁群算法是求解最小函数值的蚁群算法, 具体描述如下。

首先对每条自由链接线的长度进行归一化处理, 设定归一化参数 h_i 的取值为 $[0, 1]$, 其中 $i=1, 2, \dots, d$ 。链接线上的任意一点可以通过 $V_i(h_i) = V_i^{(0)} + h_i(V_i^{(1)} - V_i^{(0)})$ 来表示, 这里 $V_i^{(0)}$ 和 $V_i^{(1)}$ 分别代表该链接线的两个端点。利用蚁群算法求解多元函数最佳值, 得到最佳参数 (h_1, h_2, \dots, h_d) , 并依据这些参数计算相应路径的成本。

该方法使链接线上的每个点均具有被选择的可能性, 且计算量较少。最短函数值是求解两点之间的最短路径, 以三段式路径的最小函数值求解算法为例, 介绍该算法原理。

令起点坐标为 (x_s, y_s) , 终点坐标为 (x_t, y_t) , 三段式路径

$$\begin{aligned} f(t_1, t_2, t_3) &= \sqrt{[x_s - (x_{10} + t_1 a_1)]^2 + [y_s - (y_{10} + t_1 b_1)]^2} + \\ &\sqrt{[(x_{10} + t_1 a_1) - (x_{20} + t_2 a_2)]^2 + [(y_{10} + t_1 b_1) - (y_{20} + t_2 b_2)]^2} + \\ &\sqrt{[(x_{20} + t_2 a_2) - (x_{30} + t_3 a_3)]^2 + [(y_{20} + t_2 b_2) - (y_{30} + t_3 b_3)]^2} \\ &= \sqrt{(x_s - x_1)^2 + (y_s - y_1)^2} + \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} + \sqrt{(x_2 - x_t)^2 + (y_2 - y_t)^2} \end{aligned} \quad (7)$$

利用蚁群算法对上述路径长度的表达式进行迭代求解, 寻找最优值。最优值包括每一次迭代中得到的函数的最小值和其对应的自变量的取值。

针对每次迭代得到 t_1, t_2, t_3 的解及其对应的路径长度的值 f_i , 令其中的最大值为 f_{\max} , 最小值为 f_{\min} , 则信息素定义为:

$$f_{\max} - f_i = \text{Tau}_i \quad (8)$$

利用蚁群算法进行迭代的过程中, 其启发函数的表达式为:

$$\text{TMT} = |\text{Tau} - \min(\text{Tau})| \quad (9)$$

该蚂蚁的状态转移概率表达式为:

$$P_o = \text{Tau}^{\text{alpha}} \times \text{TmT}^{\text{beta}} \quad (10)$$

$$PT = P_o / \text{sum}(P_o) \quad (11)$$

局部搜索和全局搜索的概率值为 PT 的中间值。函数值 P_o 越小时, 相应的转移概率越大, 越要进行局部搜索; 函数值越大时, 相应的转移概率越小, 越要进行全局搜索。

蚁群在进行迭代优化的过程中, 若状态转移概率的值大于 P_o , 则蚁群进行局部搜索, 局部过程中对 t 的搜索范围为 $\pm 1/T$, 其中 T 为迭代次数; 若状态转移概率的值小于 P_o , 则蚁群进行全局搜索, 全局搜索过程中对 t 的搜索范围为 $[0, 1]$, 直至得到 t_1, t_2, t_3 分别对应某个值, 使得路径长度 $f(t_1, t_2, t_3)$ 最小。

在迭代求解过程中, 若出现 $t_i < 0 (i=1, 2, 3)$, 则令 $t_i = 0$; 若出现 $t_i > 1 (i=1, 2, 3)$, 则令 $t_i = 1$, 之后代入下一次迭代计算。当迭代次数用完或迭代过程中的最优值连续几代不发生

中的第一条链接线起, 终点分别为 $(x_{10}, y_{10}), (x_{1e}, y_{1e})$, 如图 4 所示, 该段路径的向量表示为 $\bar{V} = (a_1, b_1)$ 。其中 $a_1 = x_{1e} - x_{10}, b_1 = y_{1e} - y_{10}$, 则有 $(x_1, y_1) = (x_{10}, y_{10}) + t_1(a_1, b_1), t_1 \in [0, 1]$ 。

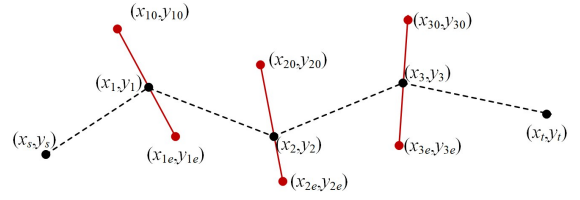


图 4 三段式路径示意图

Fig 4 Schematic diagram of the three-segment path

第二条链接线的起、终点分别为 $(x_{20}, y_{20}), (x_{2e}, y_{2e})$, 第三条链接线的起、终点分别为 $(x_{30}, y_{30}), (x_{3e}, y_{3e})$ 。同上理论可有: $a_2 = x_{2e} - x_{20}, b_2 = y_{2e} - y_{20}; a_3 = x_{3e} - x_{30}, b_3 = y_{3e} - y_{30}; (x_2, y_2) = (x_{20}, y_{20}) + t_2(a_2, b_2), t_2 \in [0, 1]; (x_3, y_3) = (x_{30}, y_{30}) + t_3(a_3, b_3), t_3 \in [0, 1]$ 。

则从起点至终点的路径长度可表示为:

变化, 即可得到基于蚁群算法的最小函数值。

当蚂蚁从初始位置 S 走到终点 E , 完成迭代搜索后, 选择最短的一条作为最优解, 并更新该条路径上的每个转折点的信息素, 如下所示:

$$\Pi_{ij}(t+1) = (1-\rho) \Pi_{ij}(t) + \rho \Delta \Pi_{ij} \quad (12)$$

其中, $\Delta \Pi_{ij} = 1/L^*$, L^* 为最短路径的长度; ρ 为信息素的挥发程度, 取值为 $0 < \rho < 1$ 。

4.3 几何优化算法

三角剪枝几何优化的基本思想是: 如果路径中的多个连续节点 $V_i, \dots, V_j, \dots, V_k$ 形成一个多段折线, 此时 V_i 与 V_k 的直接连接路径不经过任何障碍物, 则可以用 V_i 与 V_k 之间的直线段替换从 V_i 到 V_k 的多条折线路段。图 5 的蓝色实线为初始规划路径, 图 6 的红色实线为利用三角剪枝进行优化后的规划路径。

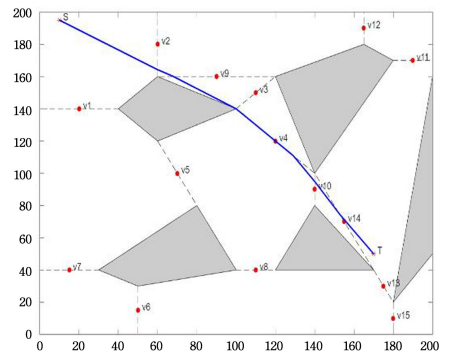


图 5 基于初始路径与蚁群算法的优化路径

Fig. 5 Optimized path based on initial path and ant colony algorithm

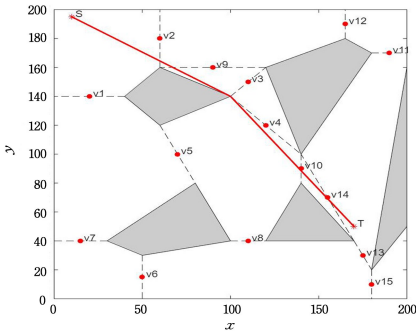


图 6 三角剪枝(几何优化)示意图(电子版为彩图)

Fig. 6 Schematic diagram of triangular pruning (geometric optimization)

5 全覆盖路径规划算法

Boustrophedon 一词最早于 1699 年在英语中使用,字面意思为“牛的方式”。通常,当牛在田地里拖犁时,它会沿着一条直线穿过整个田地,然后转身,沿着与前一条路径相邻的新直线路径前进,通过重复该操作,可保证牛的犁地过程覆盖整个田地,即实现遍历。

移动机器人以最短的路径走到需要遍历某个子区域的顶点附近,沿该子区域的最长边呈“弓”字形往复地走到子区域该条边的另一顶点处,到达这条边的另一顶点时,再沿着这条最长边往复运动,实现移动机器人全覆盖的路径规划。

6 仿真分析

针对上述研究内容,为有效说明其可行性和优势所在,本文对一系列参数进行设定,并在此基础上利用 MATLAB 软件进行仿真分析。

6.1 参数设定

地图及障碍物顶点相关参数设定如表 1 所列。

表 1 地图及障碍物顶点相关参数设定

Table 1 Parameter settings related to map and obstacle vertices

Type	Parameter
横坐标范围	[0,200]
纵坐标范围	[0,200]
第一个障碍物顶点	[40,140],[60,160],[100,140],[60,120]
第二个障碍物顶点	[50,30],[30,40],[80,80],[100,40]
第三个障碍物顶点	[120,160],[140,100],[180,170],[165,180]
第四个障碍物顶点	[120,40],[170,40],[140,80]
第五个障碍物顶点	[180,20],[220,80],[200,160]

移动机器人及蚁群算法的相关参数设定如表 2 所列。

表 2 移动机器人及蚁群算法相关参数设定

Table 2 Parameter settings related to mobile robots and ant colony algorithms

Type	Parameter
移动机器人初始位置	[-0.2,-0.2]
相邻“弓”字形路径间距	6
蚂蚁数量	50
最大迭代次数	30
信息素重要程度因子	2
启发函数重要程度因子	7
局部信息素挥发因子	0.5
全局信息素挥发因子	0.8
常系数	5

6.2 仿真实验

基于 Maklink(链接)图与 Boustrophedon(牛耕式)路径的移动机器人二维全覆盖路径规划策略的流程如图 7 所示。

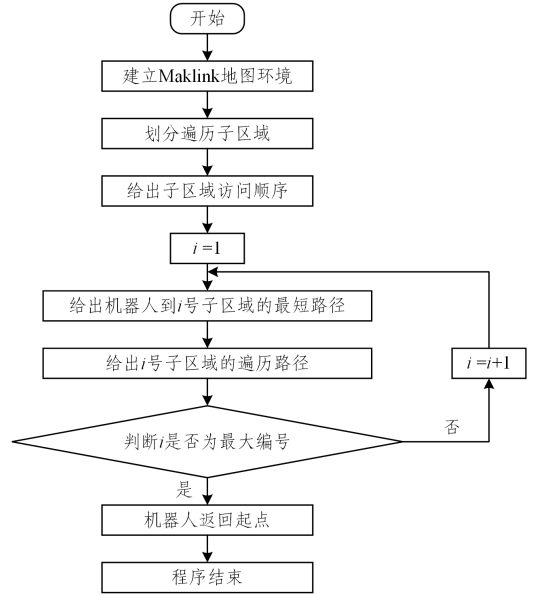


图 7 算法流程

Fig. 7 Algorithm flow

通过上述参数设定以及对算法的应用,在 MATLAB 软件中进行仿真分析即可得到移动机器人在存在障碍物的二维地图中的仿真遍历路径,如图 8 所示。

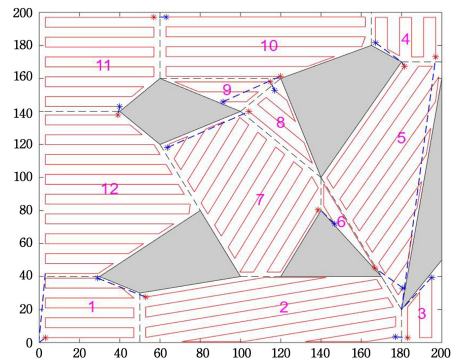


图 8 机器人全局遍历路径(电子版为彩图)

Fig. 8 Robot global traversal path

其中,红色实线即为利用 Boustrophedon 算法得到的移动机器人全局遍历路径,蓝色虚线为利用蚁群算法得到的移动机器人在可遍历子区域间的最优转换路径,子区域内数字为移动机器人的遍历顺序。上述仿真实验的结果表明:当子区域间转换路径不计入重复路径时,本文方法的规划路径覆盖率达 100%,重复率为 0%;若将子区域转换路径计入重复路径,重复覆盖面积计算方法为子区域间转换路径长度乘以路径宽度,覆盖率保持 100%,重复率仅为 5.80%。

在上述算法说明及仿真实验中,以存在凸多边形障碍物的地图为例,为验证该算法在不同障碍物地图环境下的表现,本文又分别建立了全凹多边形障碍物地图(见图 9)、凹凸多边形混合障碍物地图(见图 10)与房屋走廊仿真模拟地图(见图 11),利用该算法进行遍历的效果表现依旧良好,凸显了该算法的稳健性。

6.2.1 全凹多边形障碍物地图仿真实验

利用存在全凹多边形障碍物的二维地图进行仿真实验,地图内障碍物相关参数如表 3 所列,遍历结果如图 9 所示。

表 3 全凹多边形障碍物地图相关参数设定

Table 3 Parameter settings for map of fully concave polygon obstacles

Type	Parameter
横坐标范围	[0,20]
纵坐标范围	[0,20]
第一个障碍物顶点	[3,19],[3,11],[5,14],[6,11],[6,15],[5,17],[5,19]
第二个障碍物顶点	[0,6],[0,4],[4,4],[4,6],[2,5]
第三个障碍物顶点	[15,8],[15,5],[17,4],[17,2],[19,2],[19,8]
第四个障碍物顶点	[7,10],[10,10],[10,13],[14,13],[14,5],[12,5],[12,0],[10,0],[10,3],[6,3],[6,5],[11,5],[11,8],[7,8]
第五个障碍物顶点	[16,12],[18,12],[18,13],[17,13],[17.5,14],[16.5,15],[17,16],[16,17]

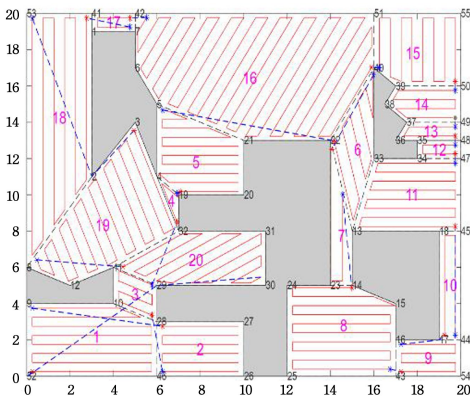


图 9 全凹多边形障碍物环境下的机器人全局遍历路径

Fig. 9 Global traversal path of robots in an environment with fully concave polygonal obstacles

6.2.2 凹凸多边形混合障碍物地图

利用存在凹凸多边形混合障碍物的二维地图进行仿真实验,地图内障碍物相关参数如表 4 所列,遍历结果如图 10 所示。

表 4 凹凸多边形混合障碍物地图相关参数设定

Table 4 Parameter settings for concave-convex polygon hybrid obstacle map

Type	Parameter
横坐标范围	[0,20]
纵坐标范围	[0,20]
第一个障碍物顶点	[1,19],[1,17],[3,17],[3,19]
第二个障碍物顶点	[1,15],[1,11],[4,11],[4,15]
第三个障碍物顶点	[0,6],[0,4],[4,4],[4,6]
第四个障碍物顶点	[6,20],[6,12],[9,12],[9,20]
第五个障碍物顶点	[15,8],[15,5],[19,5],[19,8]
第六个障碍物顶点	[17,4],[17,2],[19,2],[19,4]
第七个障碍物顶点	[7,10],[10,10],[10,13],[14,13],[14,5],[12,5],[12,0],[10,0],[10,3],[6,3],[6,5],[11,5],[11,8],[7,8]
第八个障碍物顶点	[14,2],[14,1],[15,1],[15,2]
第九个障碍物顶点	[16,12],[18,12],[18,13],[17,13],[17.5,14],[16.5,15],[17,16],[16,17]

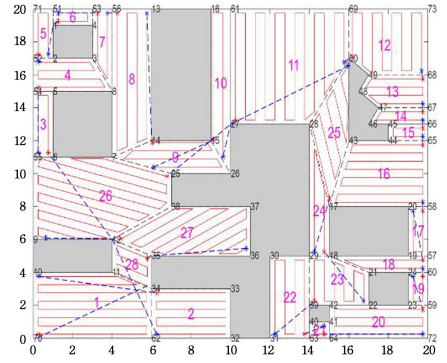


图 10 凹凸多边形混合障碍物环境下的机器人全局遍历路径

Fig. 10 Global traversal path for robots in an environment with mixed obstacles of concave and convex polygons

6.2.3 房屋走廊仿真模拟地图

利用房屋走廊仿真模拟地图进行仿真实验,地图内障碍物相关参数如表 5 所列,遍历结果如图 11 所示。

表 5 房屋走廊仿真模拟地图相关参数设定

Table 5 Parameter settings for the simulated map of the house corridor

Type	Parameter
横坐标范围	[0,200]
纵坐标范围	[0,200]
第一个障碍物顶点	[0,200],[0,170],[10,170],[10,190],[30,190],[30,200]
第二个障碍物顶点	[0,140],[0,135],[80,135],[80,110],[85,110],[85,160],[80,160],[80,140]
第三个障碍物顶点	[25,110],[25,95],[50,95],[50,110]
第四个障碍物顶点	[80,200],[80,175],[85,175],[85,200]
第五个障碍物顶点	[0,65],[0,60],[80,60],[80,35],[85,35],[85,95],[80,95],[80,65]
第六个障碍物顶点	[0,0],[0,30],[30,0]
第七个障碍物顶点	[80,20],[80,0],[85,0],[85,20]
第八个障碍物顶点	[125,200],[125,110],[130,110],[130,200]
第九个障碍物顶点	[125,95],[125,0],[130,0],[130,95]
第十个障碍物顶点	[160,200],[160,155],[165,155],[165,200]
第十一个障碍物顶点	[160,140],[160,55],[165,55],[165,95],[200,95],[200,100],[165,100],[165,140]
第十二个障碍物顶点	[150,35],[150,30],[200,30],[200,35]

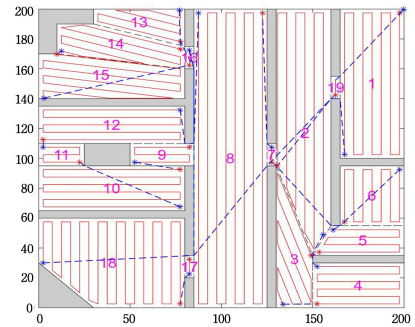


图 11 房屋走廊仿真模拟地图环境下的机器人全局遍历路径

Fig. 11 Global traversal path of a robot in a simulated house corridor map environment

在上述多种测试环境下,利用该算法进行仿真实验均具有较好的表现,可见,该算法的稳定性好,适用性广。

6.3 参数敏感性分析

为探究蚁群算法参数和 Boustrophedon 路径间距等参数

对实验结果的影响,本文增加了相关参数的敏感性分析。

在蚁群算法参数设定中,本文主要对信息素重要程度因子、启发函数重要程度因子、局部信息素挥发因子和全局信息素挥发因子 4 项参数进行分析,其参数变化通过影响子区域间转换路径,进而影响移动机器人的总遍历路径长度,实验结果如图 12 所示。

如图 12(a)和图 12(b)所示,当参数信息素重要程度因子 α 和启发函数重要程度因子 β 分别在 0 至 10 之间变化,而其他参数不变时,利用该算法得到的移动机器人总遍历路径长

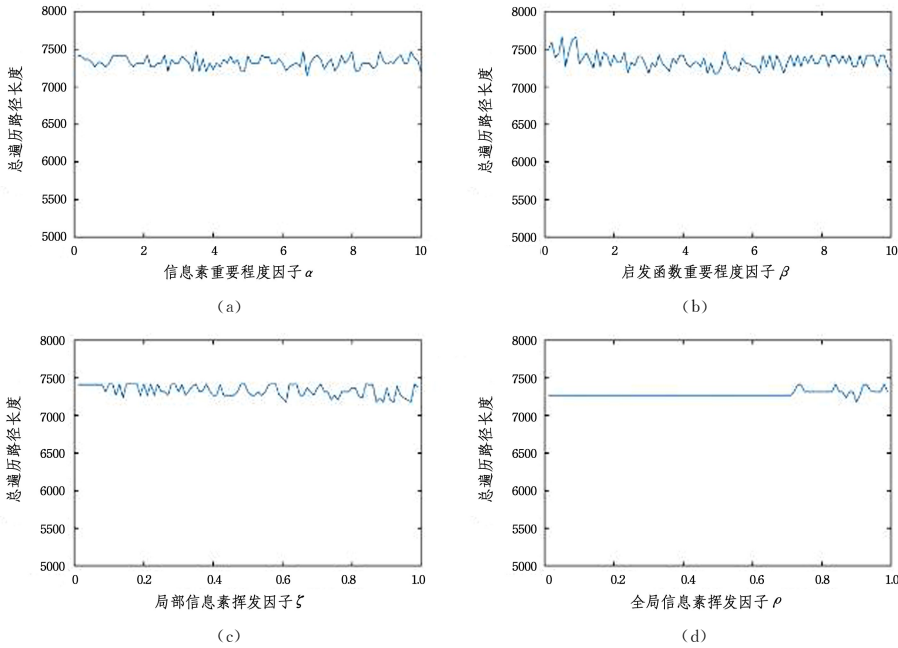


图 12 蚁群算法参数敏感性分析

Fig. 12 Sensitivity analysis of ant colony algorithm parameters

关于 Boustrophedon 路径间距的设定,其主要影响移动机器人的子区域间转换路径,进而影响全覆盖路径规划的重复率,影响机制如图 13 所示。当区域 I 遍历完成后,要进入区域 II 进行遍历,此时,在区域 II 中选择与区域 I 遍历终点最近的位置进行子区域路径转换。图 13(a)和图 13(b)中,遍历起点相同,但间距不同,因此图 13(a)中区域 I 的遍历终点与区域 II 距离较近,图 13(b)中区域 I 的遍历终点与区域 II 距离较远。

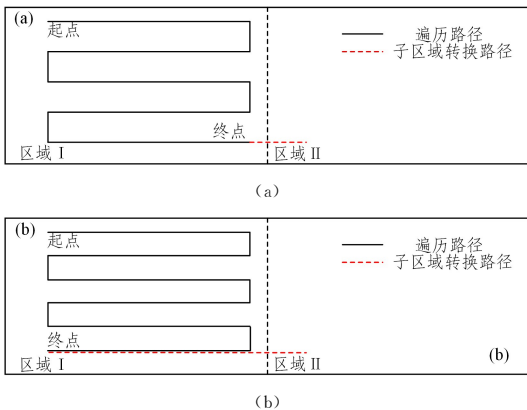


图 13 路径间距的参数影响机制

Fig. 13 Mechanism of the impact of path spacing parameters

度出现微小波动,处于误差允许范围内;如图 12(c)和图 12(d)所示,当参数局部信息素挥发因子 ζ 和全局信息素挥发因子 ρ 分别在 0 至 1 之间变化,而其他参数不变时,利用该算法得到的移动机器人总遍历路径长度同样出现微小波动,处于误差允许范围内。出现此现象的原因是蚁群算法的参数设定虽直接影响移动机器人在子区域间的转换路径,但三角剪枝几何优化算法的加入使其影响得以削减,进而对总遍历路径长度的影响减弱,同时在一定程度上说明了本文算法在全覆盖路径规划领域的先进性和稳健性。

当 Boustrophedon 路径间距在 [1, 20] 间不断变化时,子区域间转换路径长度及重复率如图 14 所示。

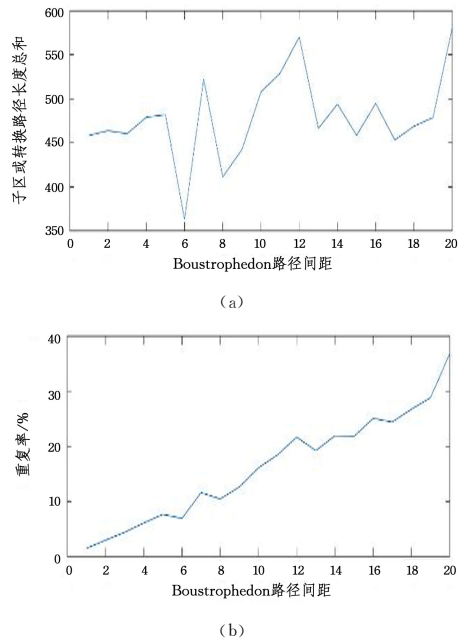


图 14 Boustrophedon 路径间距敏感度分析

Fig. 14 Sensitivity analysis of Boustrophedon path spacing

如图 14(a)所示,观察路径间距参数-子区域转换路径长度总和的变化发现:当路径间距取 6 时,转换路径长度总和最小,即转换效率最高;当重复覆盖面积计算方法按照“子区域间转换路径长度乘以路径宽度”计算时,在图 14(b)中,当路径间距取 1 至 5 时,重复率会上升;当取路径间距为 6 时,重复率首次出现下降,同时基于路径间距太小会导致遍历路径过分密集的情况,本文选择路径间距为 6 进行系列仿真实验。

6.4 仿真实验分析

为凸显本文算法与其他算法相比的遍历优势,本文分别利用传统蚁群算法、改进的蚁群算法及本文改进蚁群算法与几何优化算法的组合算法,在图 1 所示的障碍物环境下进行全覆盖路径规划,就转换路径长度、遍历路径长度及重复率三方面进行对比,结果如表 6 所列。结果表明:本文改进蚁群算法与几何优化算法的组合算法无论是转换路径长度、总路径长度还是重复率,均优于传统蚁群算法和改进蚁群算法两种单一算法。与传统蚁群算法相比,转换路径长度缩减了 4.3%,重复率从 6.04%降低至 5.80%。

表 6 3 种算法实验结果对比

Table 6 Comparison of experimental results of three algorithms

算法	转换路径长度	总路径长度	重复率/%
算法 1	368.66	6103.7	6.04
算法 2	362.86	6097.9	5.95
算法 3	352.86	6087.9	5.80

注:算法 1 为传统蚁群算法;算法 2 为改进蚁群算法;算法 3 为本文改进蚁群算法与三角剪枝几何优化算法的组合算法。

此外,目前用于全遍历路径规划的方法^[31],大多先采用栅格法进行环境建模(见图 15),然后利用人工智能与深度优先的组合算法,得到较优的全遍历路径。然而,栅格法只能将地图划分为等大小栅格单元,通过形态学膨胀处理障碍物边缘,对障碍物的规则性要求较高,不具备处理复杂多边形障碍物的优势。另外,基于栅格地图建模的全遍历算法,大多数路径遍历规划未将机器人回到原始位置纳入考虑范围,若考虑“机器人回到原始位置”的情况,则会导致重复率大幅度上升。

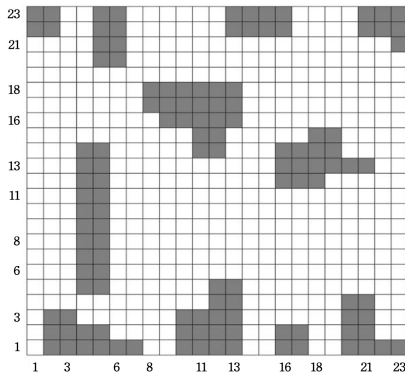


图 15 栅格地图示例

Fig. 15 Example of a grid map

本文所选的全覆盖路径规划算法在障碍物地图适用性上具有优势,不仅适用于规则障碍物地图,也适用于多种复杂多边形障碍物地图(包括凹多边形)。在图 15 所示的障碍物环

境中,将栅格地图中的障碍物提取出来后,利用本文所选算法进行全覆盖路径规划,如图 16 所示。结果表明,本文所选环境建模方法的适用性更广,建模精度更高,存储效率更优。

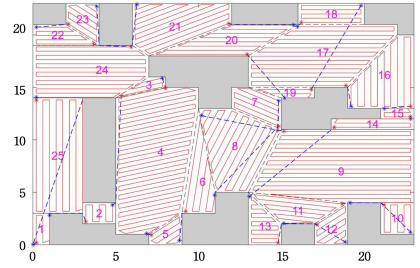


图 16 全覆盖路径规划

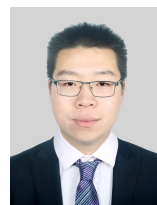
Fig. 16 Full coverage path planning

结束语 本文首先利用 Maklink 图自由链接线将整个二维平面空间划分为多个子区域(每个子区域均为凸多边形);然后将子区域的连接顺序转换为“旅行商问题”,由一维蚁群算法给出子区域的访问顺序,并由求最小函数值的蚁群算法给出机器人在子区域间转换的最优路径;最后实时规划出沿子区域内某条边平行移动的“弓”字形运动路径,实现全局遍历的路径规划。该遍历策略有效可行,且覆盖率高,遍历用时短,可为移动机器人全覆盖路径规划的相关研究提供参考。未来将关注于寻找在全遍历路径规划领域,如环境建模、规划算法等多模块的进一步创新解决方案。

参考文献

- [1] LIU Y J, ZENG G H, HUANG B. Research on path planning of mobile robots based on improved ant colony optimization algorithm [J]. Transducer and Microsystem Technologies, 2020, 39(4): 56-58, 62.
- [2] PAN Y, LI L, QIN J, et al. Unmanned aerial vehicle-human collaboration route planning for intelligent infrastructure inspection [J]. Computer-Aided Civil and Infrastructure Engineering, 2024, 39(14): 2074-2104.
- [3] MIAO S Y, ZHOU Y S. Design of motion control of tractor-trailer wheeled mobile robots with on-axle hitching based on curvature tracking method [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, 41(2): 26-33.
- [4] WU Y F, LI J, BI S, et al. Research on improved ant colony algorithm for mountain hiking emergency rescue path planning [J]. Journal of Geo-information Science, 2023, 25(1): 90-101.
- [5] TANG Y L, MU R R, ZHANG X F, et al. Research on path planning of rescue robots based on improved A* algorithm [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(6): 33-40.
- [6] MAO J X, HE Z Y, WANG Y N, et al. Review of Path Planning Technology and Applications for Electric Power Inspection Robots [J]. Control and Decision, 2023, 38(11): 3009-3024.
- [7] WU K, WU Z, LU S, et al. Full coverage path planning strategy for cleaning robots in semi-structured outdoor environments [J]. Robotics and Autonomous Systems, 2025, 192: 105050.

- [8] CHEN J Y, GUO Z J, YIN Y K. Full-traversal path planning and system design of intelligent lawn mower based on hybrid algorithm [J]. *Computer Science*, 2021, 48(S1): 633-637.
- [9] LI J, SHENG H, ZHANG J, et al. Coverage path planning method for agricultural spraying UAV in arbitrary polygon area [J]. *Aerospace*, 2023, 10(9): 755.
- [10] CHEN Y, LU Z M, CUI J L, et al. A Complete Coverage Path Planning Algorithm for Lawn Mowing Robots Based on Deep Reinforcement Learning [J]. *Sensors*, 2025, 25(2): 416.
- [11] HU Z H, YANG Z H, LIU X C, et al. Research on unmanned boat path planning algorithm based on navigation radar [J]. *Science Sinica Technologica*, 2021, 51(11): 1401-1409.
- [12] YUE G F, ZHANG M, SHEN C, et al. Bidirectional smooth A-star algorithm for mobile robot navigation planning [J]. *Science Sinica Technologica*, 2021, 51(4): 459-468.
- [13] SUN Y, FANG M, SU Y. AGV path planning based on improved Dijkstra algorithm [C]// *Journal of Physics: Conference Series*. Bristol: IOP Publishing, 2021: 012052.
- [14] WANG J, CHI W, LI C, et al. Neural RRT*: Learning-based optimal path planning [J]. *IEEE Transactions on Automation Science and Engineering*, 2020, 17(4): 1748-1758.
- [15] LI Q, XU Y, BU S, et al. Smart vehicle path planning based on modified PRM algorithm [J]. *Sensors*, 2022, 22(17): 6581.
- [16] ZHANG W, WANG N, WU W. A hybrid path planning algorithm considering AUV dynamic constraints based on improved A* algorithm and APF algorithm [J]. *Ocean Engineering*, 2023, 285: 115333.
- [17] LI Y, ZHAO J, CHEN Z, et al. A robot path planning method based on improved genetic algorithm and improved dynamic window approach [J]. *Sustainability*, 2023, 15(5): 4656.
- [18] ZHAO L, BAI Y, WANG F, et al. Path planning for autonomous surface vessels based on improved artificial fish swarm algorithm: a further study [J]. *Ships and Offshore Structures*, 2023, 18(9): 1325-1337.
- [19] PHUNG M D, HA Q P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization [J]. *Applied Soft Computing*, 2021, 107: 107376.
- [20] DE CASTRO G G R, PINTO M F, BIUNDINI I Z, et al. Dynamic path planning based on neural networks for aerial inspection [J]. *Journal of Control, Automation and Electrical Systems*, 2023, 34(1): 85-105.
- [21] LUO Q, WANG H, ZHENG Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm [J]. *Neural Computing and Applications*, 2020, 32(6): 1555-1566.
- [22] LI Z L, LUO S P, JIA Y T. A review of path planning algorithms for mobile robots. *Modern Information Technology* [J]. *Modern Information Technology*, 2024, 8(19): 184-188.
- [23] CAI Z, LIU J, XU L, et al. Cooperative path planning study of distributed multi-mobile robots based on optimised ACO algorithm [J]. *Robotics and Autonomous Systems*, 2024, 179: 104748.
- [24] HAO Z M, AN P J, LI H Y, et al. Mobile Robot Path Planning Based on Enhanced Target-Inspired Ant Colony Algorithm [J]. *Science Technology and Engineering*, 2023, 23(22): 9585-9591.
- [25] LIU Y Q, XIANG J, CAO S Q. Path planning for underwater autonomous navigation robots based on improved ant colony algorithm [J]. *Computer Engineering and Science*, 2022, 44(3): 536.
- [26] PENG X D. Improved visual path planning algorithms. *Modern Information Technology* [J]. *Modern Information Technology*, 2021, 5(3): 152-154, 158.
- [27] QIAN L X, LI H L, WANG H R, et al. Research on underwater target search and delivery strategy based on improved hidden Markov model [J]. *Scientia Sinica (Technologica)*, 2025, 55(3): 520-539.
- [28] ZHANG B F, XIE X L, HE A. Construction water path planning based on maklink diagram and cuckoo search algorithm [J]. *Journal of Shanghai Maritime University*, 2020, 41(3): 6-11, 30.
- [29] HOU T, LI J, PEI X, et al. A spiral coverage path planning algorithm for nonomnidirectional robots [J]. *Journal of Field Robotics*, 2025, 42(5): 2260-2279.
- [30] TAN C S, MOHD-MOKHTAR R, ARSHAD M R. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms [J]. *IEEE Access*, 2021, 9: 119310-119342.
- [31] NASIRIAN B, MEHRANDEZH M, JANABI-SHARIFI F. Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment [J]. *Frontiers in Robotics and AI*, 2021, 8: 624333.
- [32] MENG X G. Path planning strategy for unmanned boats oriented to observation tasks [J]. *Intelligent Computer and Applications*, 2020, 10(8): 76-83.
- [33] WANG X Y, SHENG G J, ZHANG K, et al. Path Planning for Mowing Robots Based on Improved A* Algorithm and DFS Algorithm [J]. *Journal of China Agricultural Machinery*, 2023, 44(2): 142-147.



LI Boyao, born in 1988, Ph.D, associate professor, master's supervisor. His main research interests include social computing, complex systems, financial algorithms and FinTech.



ZHAO Binbin, born in 2006, undergraduate. His main research interests include probability statistics and path planning.