

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2210120111



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 基于深度强化学习与蜉蝣算法
的移动机器人路径规划研究

论 文 作 者	邹阿威
校 内 导 师	王雷(教授)
校 外 导 师	张茂杉(高级工程师)
专业学位类别	机械
研究方向(领域)	智能优化算法及应用

论文提交日期: 2024 年 5 月 20 日

分类号：TP242.6

学校代码：10363

密 级：公开

学 号：2210120111

基于深度强化学习与蜉蝣算法的移动机器人路径规划研究

Research on Mobile Robot Path Planning Based on Deep Reinforcement Learning and Mayfly Algorithm

学生姓名：_____邹阿威_____

指导教师：_____王雷(教授)_____

校外导师：_____张茂杉(高级工程师)_____

专 业：_____机械_____

研究方向：_____智能优化算法及应用_____

论文答辩日期：2024年5月20日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：邹阿威

日期：2024年5月20日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在___年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名：

邹阿威

指导教师签名：

2/27

日期：2024年5月20日

日期：2024年5月20日

基于深度强化学习与蜉蝣算法的移动机器人路径规划研究

摘要

优秀的路径规划技术可以使得移动机器人在从事各种工作时提高其工作效率与安全性，节约更多能源，具有重要的研究意义。路径规划的研究近年来不断发展，有许多优秀的路径规划方法都被用于解决路径规划问题。然而其中深度强化学习算法只能解决简单环境下局部路径规划问题，在环境复杂度持续增高后可能无法规划出一条合适的路径，而蜉蝣算法也存在迭代速度不足，易陷入局部最优等问题，且不适用于动态环境。因此，本文考虑到将蜉蝣算法、深度强化学习算法和动态窗口法改进后相结合，经验证对解决复杂环境下的动态路径规划问题具有不错的效果。本文的主要工作如下：

首先针对传统深度强化学习算法(Deep Q-learning,DQN)在解决路径规划问题时存在的不足，提出一种改进的 DQN 算法，主要改进方法为改进探索策略，引入人工势场初始化 Q 值与跳点策略，并进行仿真验证其求解性能。结果表明本章改进算法在路径长度上明显表现更优。

其次针对传统蜉蝣算法在路径规划问题中的不足，提出一种改进蜉蝣算法，采用自适应动态参数调整策略对蜉蝣算法多个参数进行调节，将能量消耗加入适应度函数中，并引入莱维飞行策略对蜉蝣飞行过程进行优化。最后对算法进行路径规划的仿真验证其有效性，结果表明本章改进算法所得路径更短，收敛速度更快。

然后针对蜉蝣算法在动态路径规划中存在的不足，提出一种分阶段的动态路径规划方法。第一阶段执行全局规划任务，第二阶段执行局部再规划任务。在全局规划中将深度强化学习中 DQN 算法与蜉蝣算法相结合，并引入记忆机制，进行全局路径规划。局部再规划任务采用改进的动态窗口法进行实时避障。最后对算法进行仿真验证，仿真结果表明，本章改进算法能够有效完成动态路径规划任务。

最后对融合深度强化学习的改进蜉蝣算法进行基于 ROS 平台的仿真测试，对得到的数据进行分析，并对算法进行现实实验测试，对结果进行分析。

关键词：移动机器人；改进蜉蝣算法；路径规划；融合算法；
DQN 算法

Research on Mobile Robot Path Planning Based on Deep Reinforcement Learning and Mayfly Algorithm

ABSTRACT

Advanced path planning technology can enhance the efficiency and safety of mobile robots in diverse tasks, leading to significant energy savings and possessing considerable research importance. Path planning research has evolved in recent years, with numerous sophisticated methods addressing various planning challenges. However, deep reinforcement learning algorithms typically address only local path planning in simple environments and may struggle as environmental complexity increases. Similarly, the mayfly algorithm suffers from slow iteration speeds, a tendency to converge on local optima, and is ill-suited for dynamic settings. Consequently, this paper posits that integrating the mayfly algorithm with deep reinforcement learning and the dynamic window approach can effectively tackle dynamic path planning in complex environments. The principal contributions of this study are:

First, to address the limitations of the traditional Deep Q-learning (DQN) algorithm in path planning, an enhanced DQN algorithm is introduced. Key improvements include an optimized exploration strategy, the use of an artificial potential field for Q-value initialization, and a jump point strategy, all validated through simulation. The findings indicate superior path length performance by the algorithm presented in this section.

Second, in response to the deficiencies of the traditional mayfly algorithm in path planning, a refined ephemerfly algorithm is developed. This version employs an adaptive dynamic parameter adjustment strategy, incorporates energy consumption into the fitness function, and leverages the Levy flight strategy to refine the ephemerfly's trajectory. Simulation-based path planning tests confirm the improved algorithm's efficacy,

demonstrating shorter paths and accelerated convergence.

Next, to overcome the mayfly algorithm's drawbacks in dynamic path planning, a phased dynamic path planning method is proposed. The initial phase focuses on global planning, while the subsequent phase addresses local re-planning. Global planning integrates the DQN and mayfly algorithms within a deep reinforcement learning framework, enhanced by a memory mechanism. For local re-planning, an advanced dynamic window approach facilitates real-time obstacle avoidance. Simulation results affirm the improved algorithm's capability to efficiently execute dynamic path planning tasks.

Lastly, the improved mayfly algorithm undergoes simulation testing on the ROS platform, with data analysis and real-world experimental validation to assess performance.

Keywords: Mobile robots; improved mayfly algorithm; path planning; fusion algorithm; deep reinforcement learning

目录

第 1 章 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	3
1.2.1 路径规划方法研究现状	3
1.2.2 蜚螭算法研究现状	5
1.2.3 深度强化学习在路径规划中研究现状	5
1.3 本文研究内容与组织结构	6
第 2 章 蜚螭算法与 DQN 算法的路径规划理论	9
2.1 路径规划问题描述	9
2.2 环境建模	9
2.3 基本蜚螭算法	10
2.3.1 基本蜚螭算法思想	10
2.3.2 计算过程	10
2.3.3 算法流程	12
2.3.4 蜚螭算法存在问题	13
2.4 DQN 算法	14
2.4.1 深度学习	14
2.4.2 强化学习	16
2.4.3 DQN 算法	17
2.4.4 DQN 算法存在问题	18
第 3 章 基于改进深度强化学习的移动机器人路径规划研究	19
3.1 改进的探索策略	19
3.2 引力势场	20
3.3 基于双指针的路径优化策略	21
3.4 算法流程	22
3.5 仿真测试	22
第 4 章 基于改进蜚螭算法的移动机器人路径规划研究	28
4.1 自适应参数调整策略	28
4.2 莱维飞行	30
4.3 改进的适应度函数	30
4.4 算法运行步骤	31
4.5 仿真分析	33
第 5 章 基于深度强化学习和蜚螭算法的移动机器人动态路径规划研究	37

5.1 结合 DQN 的自适应参数控制策略	37
5.2 记忆机制	39
5.3 改进的动态窗口法	39
5.3.1 运动模型	39
5.3.2 改进的速度模型	40
5.3.3 评价函数	41
5.4 算法运行步骤	41
5.5 测试与仿真	42
5.5.1 改进蜉蝣算法仿真验证	42
5.5.2 动态路径规划仿真	45
第 6 章 基于 ROS 平台的仿真实验	48
6.1 ROS 平台及仿真软件简介	48
6.1.1 ROS 简介	48
6.1.2 Gazebo 简介	48
6.1.3 Rviz 简介	49
6.2 环境建模与仿真实验	49
6.2.1 移动机器人建模与环境建模	49
6.2.2 仿真实验与结果分析	51
6.2 实物实验验证	53
6.2.1 机器人平台搭建	53
6.2.2 现实实验验证	53
第 7 章 总结与展望	57
7.1 工作总结	57
7.2 研究展望	57
参考文献	59

第 1 章 绪论

1.1 研究背景与意义

机器人是指可以帮助人类或独立完成某些重复性工作以及恶劣环境下工作的人造装置^[1]。如今，机器人已广泛应用于汽车制造、快递物流、航空航天、船舶制造等各个工业领域。机器人辅助制造由此得到广泛应用，提高了工艺效率和产品质量^[2]。

图 1-1 所示的财务数据显示，在 2021 年，全球机器人市场规模达到了令人瞩目的 175 亿美元，超越了 2018 年创下的历史新高 165 亿美元。市场需求的不断增长以及机器人技术的广泛应用预示着这一市场的持续扩张。预计到 2024 年，机器人市场的规模将进一步攀升至约 230 亿美元，展现了该行业的强劲发展势头^[3]。

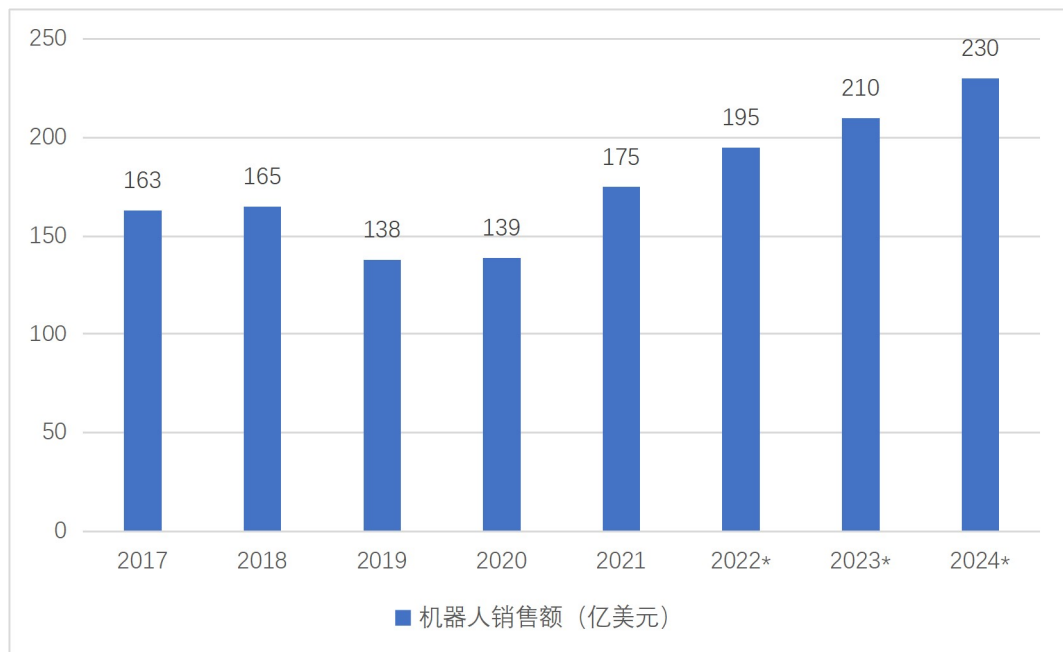


图 1-1 机器人销售额(亿美元)

在过去五年中，机器人市场规模持续增长。正如图 1-2 所示，预计到了 2024 年，这一市场将进一步壮大，市场规模有望突破 110 亿美元，显示出中国在机器人及工业自动化领域的迅速发展^[3]。

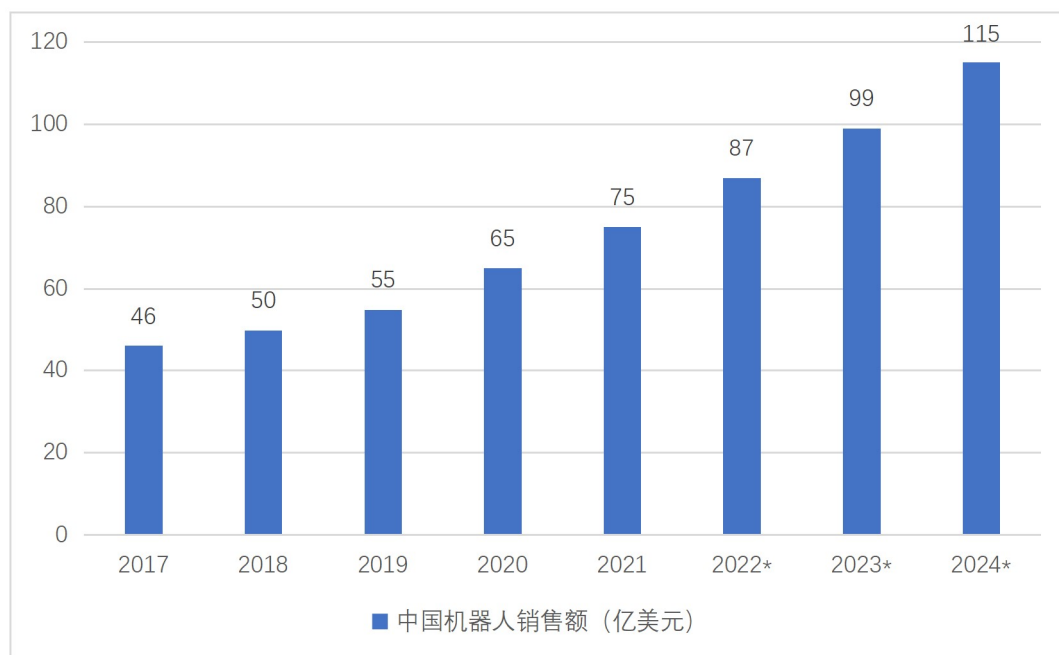


图 2-2 中国机器人销售额(亿美元)

移动机器人是机器人的一个重要分支，移动机器人是现代智能制造领域的核心技术之一，它不仅拥有广阔的市场潜力，更承载着深远的战略价值。在即将到来的工业 4.0 时代，移动机器人的角色将变得更加关键，其独特功能和灵活性使其成为不可或缺的力量。

移动机器人的不仅广泛应用于制造业当中，还普遍应用于工业、农业、航天业、海洋业、医疗行业以及军工业等领域^[4]，为各行各业注入了新鲜的活力。在“十四五”规划中强调了机器人产业的重要性，并指出其发展将主要集中在采矿、制造、建筑和农业等核心行业。同时，机器人技术也将扩展到家庭服务、公共服务、医疗保健、养老助残以及特殊环境作业等多元化需求，以实现行业的全面升级和社会服务的优化。这一规划突显了机器人技术在未来社会经济发展中的重要战略地位^[5]。而上述这些应用领域的实现都离不开路径规划技术，由此可以看出路径规划技术的研究对国家与社会具有重要意义。

移动机器人的路径规划是指在某种环境中规划出一条满足某种条件的由起始点至目标点的安全路径^[6]。优秀的路径规划技术可以使得移动机器人在从事各种工作时提高其工作效率与安全性，节约更多能源，具有重要的研究意义^[7]。对该技术的深入研究，不仅可以有效的提高国家生产制造方面的能力，增强国家可持续发展能力，还能促进航天、医学、军事、农业等各领域的发展。

1.2 国内外研究现状

1.2.1 路径规划技术研究现状

路径规划的研究近年来不断发展，有许多优秀的路径规划方法都被用于解决路径规划问题^[8]。根据算法的特点可以将路径规划算法分为三类，即：经典算法、仿生算法和人工智能算法^[9]，如图 1-3 所示。经典算法包括 A*算法^[10]、快速拓展随机树算法 (rapidly exploring random tree, RRT)^[11]、Dijkstra 算法^[12]、人工势场法^[13,14]、动态窗口法(Dynamic Window Approaches, DWA)^[15]等。其中仿生算法包括蚁群优化算法^[16-19]、遗传算法^[20-22]、粒子群算法 (Particle swarm optimization, PSO)^[23-26]、蝙蝠算法^[27, 28]、蜉蝣算法、萤火虫算法^[29]、布谷鸟搜索算法^[30]和人工蜂群算法^[31]等。人工智能算法主要包括模糊控制算法^[32]和神经网络算法，深度强化学习算法等。

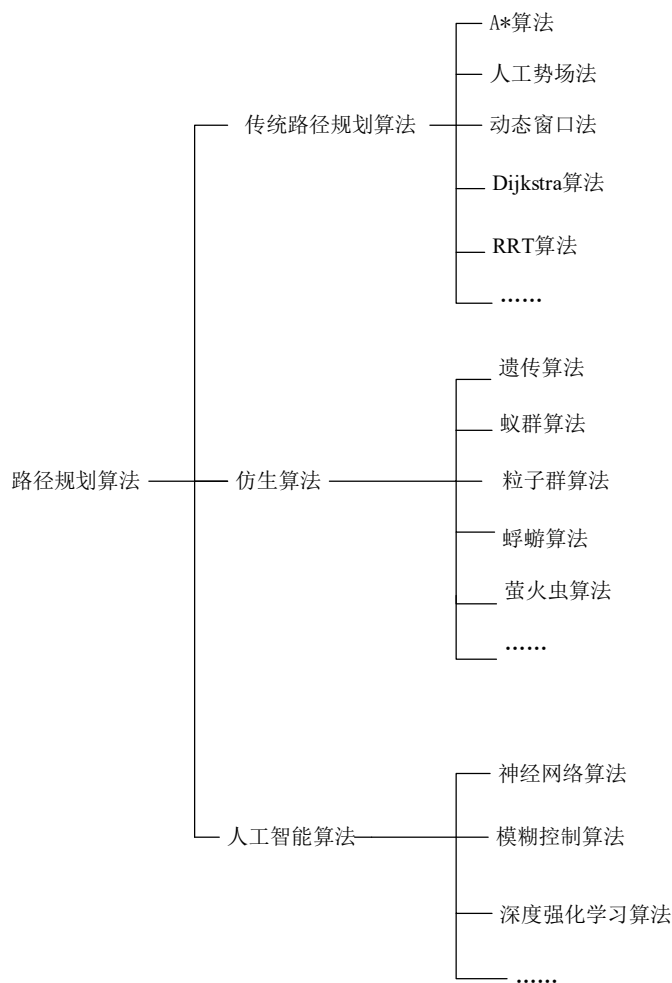


图 1-3 常用路径规划算法

其中 A*算法、Dijkstra 算法等传统方法因其可以在简单环境中快速求解路

径规划问题，求解能力稳定而在路径规划问题中被广泛应用。但其在实际应用时仍存在一定的不足之处，RRT 算法很难找到最优解，而 A* 算法虽然理论上能够找到最优路径，但在复杂环境下的路径规划时，效率明显不足。DWA 虽然实时性不错，算法结构也比较简单，但是对于除两轮差速模型以外的机器人并不提供支持，且只能解决简单环境下路径规划问题。针对这些问题，许多学者也对其进行了改进。

文献[33]提出一种变权重 A* 算法，可以根据机器人当前位置动态调整 A* 算法中启发函数与代价函数权重，有效提高了算法的求解效率。文献[34]提出了一种改进 RRT 算法。在随机树算法中，在待扩展节点的选取上引入了目标导向的启发式信息，有效提高了收敛速度。在新节点的生成及其加入过程中，考虑了无人机的动力学约束条件，确保规划出的航线在实际应用中的可行性。并采用动态扩展随机树技术对原有随机树进行剪枝和优化重构，以便更有效地规避潜在威胁，增强了路径规划的安全性和实用性。文献[35]中提出了一种改良的动态窗口法，该算法采用多目标粒子群优化（MOPSO）方法来优化 DWA 算法的子评价函数。通过 MOPSO 算法来动态调整 DWA 的权重系数，将权重系数的自适应调整问题转换为多目标优化问题，从而在提高算法安全性的同时，也增加了运行速度。

这些改进取得了一定成果，但传统路径规划算法在面对复杂问题时仍然很难成功规划出一条最优路径。因此，学者们将目光转向了在处理复杂的组合优化问题时具有出色表现的仿生智能算法，并对其进行了广泛的研究与应用。

文献[36]中，研究者提出了一种结合混沌理论的粒子群优化算法，提升了路径规划的效率和质量。文献[37]通过引入中间值插入法和混合选择策略，优化了遗传算法，增强了种群的多样性并减少了早熟收敛的问题。文献[38]提出了一种改进的蚁群算法，通过奖励惩罚机制和信息素自适应挥发因子，提高了算法的全局搜索能力和收敛速度。

这些算法都在解决路径规划问题时取得了不错的成果，但单一智能优化算法在解决复杂优化问题时未能将群体智能和进化算法的优势相结合，仍有些力不从心，因此融合算法成为了路径规划算法发展的一种趋势。

1.2.2 蜉蝣算法研究现状

蜉蝣算法^[39]，由 Konstantinos Zervoudakis 于 2020 年提出，这种算法通过模拟蜉蝣的飞行与交配过程，以解决复杂的优化问题，融合了粒子群算法、遗传算法与萤火虫算法，集中了群体智能和进化算法的优势。蜉蝣算法因其在收敛精度、速度和利用方面的特殊优势而受到越来越多的关注，但在收敛效果，搜索精度等也存在一些问题，因此国内外许多学者也对其进行了一系列的改进。

在文献[40]中，研究者结合了模拟退火算法和蜉蝣优化算法，以确定支持向量回归的最优超参数，解决了蜉蝣算法在探索性方面的不足，并提高了迭代效率。文献[41]引入了反吸引速度更新机制到蜉蝣算法中，平衡了全局搜索与局部寻优的能力，增强了算法的收敛性；同时，通过对全局最优蜉蝣进行逐维的重心反向学习变异，减少了维度间的干扰，帮助算法避免局部最优，加快了收敛速度。文献[42]使用 Tent 映射产生混沌序列作为蜉蝣算法的初始种群，为全局搜索奠定了基础，并在迭代过程中对全局最优个体施加混沌扰动，增强了跳出局部最优的能力。文献[43]采用了多种策略改进蜉蝣算法，包括使用 Sin 混沌映射初始化种群，引入 Tent 混沌映射、高斯变异和不完全伽玛函数，有效提升了全局搜索能力，加快了收敛速度，并增强了稳定性。文献[44]引入了非均匀高斯变异策略来更新蜉蝣的位置，提高了全局搜索能力和种群多样性，并通过局部停滞对抗策略进一步增强了全局搜索能力。

随着研究的深入，学者对其进行的改进都取得了一定的成果，在某一特定场景的效果也越来越强，但当这些改进算法应用于复杂路径规划问题时，惯性权重与正吸引因子都为固定值，不具有平衡全局搜索与局部寻优的能力，有时为了优化效果也牺牲了迭代速度，且实时性较低，因此本文考虑对这些方面进行研究。

1.2.3 深度强化学习在路径规划中研究现状

随着近几年深度强化学习的兴起，因其无需依赖数据且拥有自主学习能力，越来越多的国内外学者也开始研究其在求解路径规划问题中的应用。

文献[45]中提出了一种新的方向惩罚避障函数，优化了路径规划的长度，还通过引入人工势场法的思想来建立奖惩函数，有效提高了避障成功率。文献[46]则对深度学习网络的输入输出进行了改进，并设计了一个新的奖惩函数，有效

缓解了奖励稀疏的问题。文献[47]采用 A* 算法进行全局路径规划，结合深度强化学习进行局部路径规划，并设计了新的动作空间与状态空间，有效提升了路径规划的性能。

通过上述文献可以看出，深度强化学习算法正在解决路径规划问题时取得了越来越好的成绩，但其在 Q 表初始化阶段，通常采用均等分布或随机分布的方法，这在大型未知环境中可能导致大量无效迭代。且最终所优化路径常常难以达到最优。因此本文考虑将针对这些问题进行一些改进及优化。

综上可知，国内外学者对蜉蝣算法和深度强化学习算法进行了一系列改进，这些改进方法都取得了不错的效果，并在各领域解决了一些问题，但当其应用于路径规划问题时，深度强化学习算法只能解决简单环境下局部路径规划问题，在环境复杂度持续增高后可能无法规划出一条合适的路径，而蜉蝣算法也存在迭代速度不足，易陷入局部最优等问题，且不适用于动态环境。

因此，本文考虑到将深度强化学习算法、动态窗口法与蜉蝣算法改进后相结合，经验证对解决复杂环境下的动态路径规划问题具有不错的效果。

1.3 本文研究内容与组织结构

本文主要包括 7 个章节：

第 1 章：绪论。简述移动机器人路径规划的研究背景与研究意义，对路径规划算法国内外研究进展、研究方法以及不足进行了分析，并提出了本文的主要研究内容。

第 2 章：蜉蝣算法与 DQN 算法的路径规划理论。主要简述了蜉蝣算法与深度强化学习算法中的 DQN 算法的实现及理论分析。给出了路径规划任务的数学描述与评价指标，进行了环境建模的描述，然后对比分析了蜉蝣算法与 DQN 算法的路径规划实现过程与存在的问题。

第 3 章：基于改进深度强化学习的移动机器人路径规划研究。针对传统 DQN 算法在解决路径规划问题时存在的不足，提出一种改进的 DQN 算法，主要改进方法为改进探索策略，引入人工势场初始化 Q 值与跳点策略，并进行仿真验证其求解性能。

第 4 章：基于改进蜉蝣算法的移动机器人路径规划研究。针对传统蜉蝣算法在路径规划问题中的不足，提出一种改进蜉蝣算法，采用自适应动态参数调

整策略对蜉蝣算法多个参数进行调节，将能量消耗加入适应度函数中，并引入莱维飞行策略对蜉蝣飞行过程进行优化。最后对算法进行路径规划的仿真验证其有效性。

第 5 章：基于深度强化学习和蜉蝣算法的移动机器人动态路径规划研究。针对蜉蝣算法在动态路径规划中存在的不足，提出一种分阶段的动态路径规划方法。第一阶段执行全局规划任务，第二阶段执行局部再规划任务。在全局规划中将深度强化学习中 DQN 算法与蜉蝣算法相结合，并引入记忆机制，进行全局路径规划。局部再规划任务采用改进的动态窗口法进行实时避障。最后对算法进行仿真验证。

第 6 章：基于 ROS 平台的仿真实验与现实实验。对改进的融合深度强化学习的改进蜉蝣算法进行基于 ROS 平台的仿真测试，对得到的数据进行分析，并对算法进行现实实验测试，对结果进行分析。

第 7 章：总结。对本文的研究结论进行了归纳总结，同时也指出了研究过程中存在的局限性，并对未来的研究方向提出了展望。

图 1-4 为本文技术路线总图。

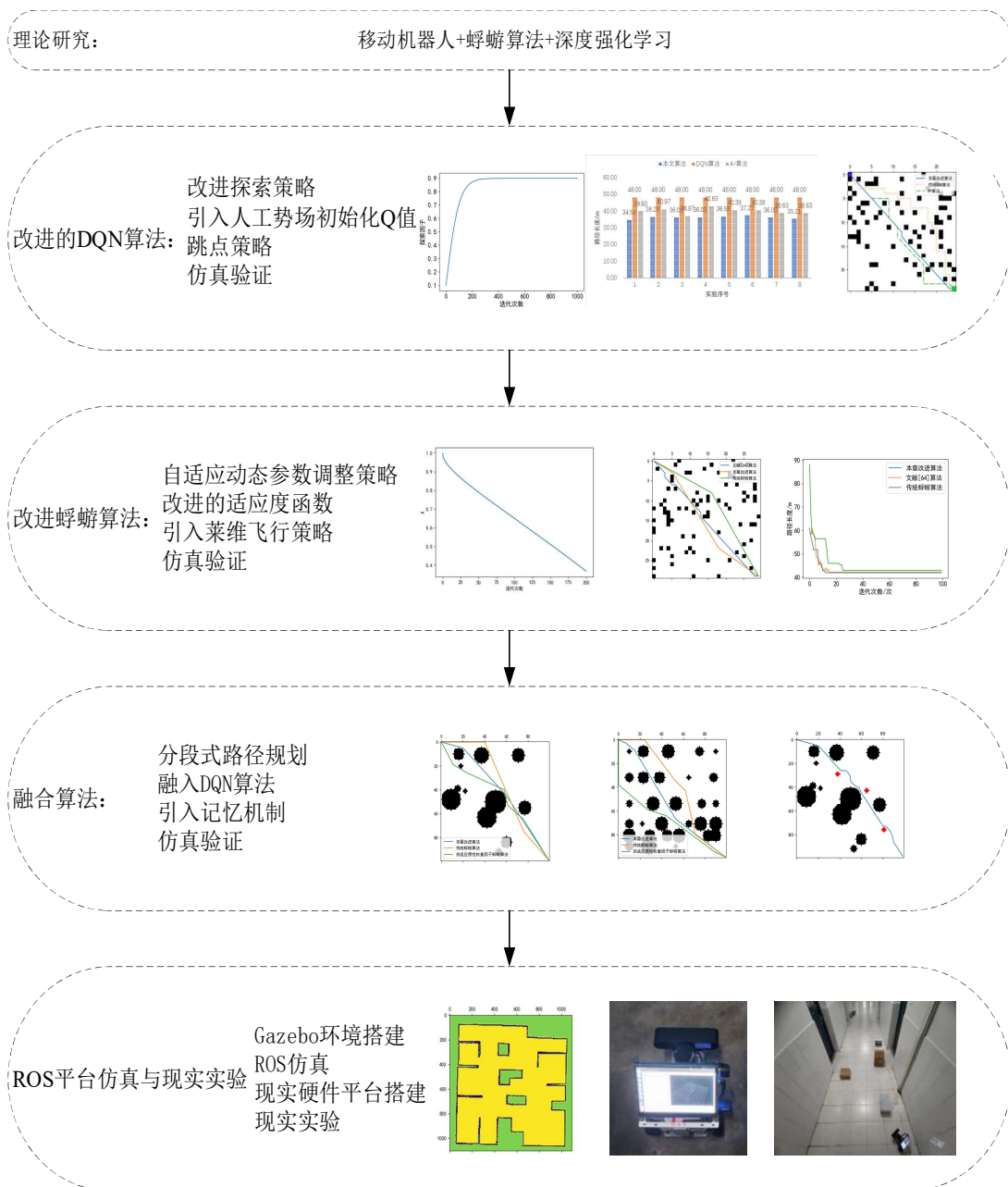


图 1-4 技术路线总图

第 2 章 蜚螭算法与 DQN 算法的路径规划理论

本章首先对路径规划问题进行了描述，选取了本文的环境建模方式，并对蜚螭算法与强化学习算法进行了逐一的理论介绍，为之后对蜚螭算法与强化学习算法的改进与融合做出了理论基础准备。

2.1 路径规划问题描述

路径规划是移动机器人导航研究中的关键环节，它确保机器人能够在各种环境中有效地导航和执行任务，是移动机器人执行各项任务的基础。其主要目标是基于既定的优化准则（例如最短行进距离、最快规划时间以及最低能量开销等），在模型空间内寻找到一条从初始点至目标点的最优路径，同时确保该路径的安全性，避免任何碰撞事件。

2.2 环境建模

路径规划的环境建模方法有很多，如栅格法、几何图法、拓扑图法等。其中，栅格法^[48]因其简单有效，可以有效降低计算量，所以得到广泛应用，本文拟采用栅格法进行环境建模。

其具体实施方法是：在二维平面的运动空间中，地图被划分为矩阵，障碍物区域以黑色标记，意味着不可通行；非障碍物区域则以白色标记，表示可以通行。若障碍物未完全占据一个栅格，该栅格仍视为障碍物区域。环境栅格图模型如图 2-1 所示。

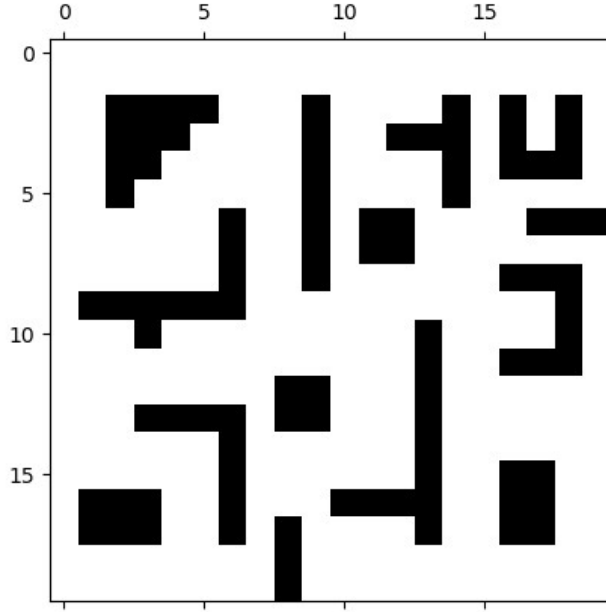


图 2-1 栅格地图

2.3 基本蜉蝣算法

蜉蝣算法，由 Konstantinos Zervoudakis 在 2020 年提出，是一种模拟蜉蝣飞行和交配行为的优化算法。它结合了群体智能和进化算法的特点，通过模拟自然界中蜉蝣的行为来寻找最优解。

2.3.1 基本蜉蝣算法思想

蜉蝣算法通过模拟自然界中蜉蝣的行为来寻找最优解。在这个过程中，每个解都被视为一只在搜索空间中飞行的蜉蝣，它们根据经验不断调整飞行方向。雄性和雌性蜉蝣根据适应度进行排名，按照排名交配产生后代。算法保持蜉蝣总数恒定，确保最优秀的蜉蝣能够存活下来。

2.3.2 计算过程

在 D 维搜索空间中，第 i 只雄性蜉蝣的位置可以用 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ 表示，其中包含了该蜉蝣在每个维度上的位置信息。

同时它具有一个速度，可以表示为：

$$v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$$

所以位置更新公式为：

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2-1)$$

每只雄性蜉蝣都会在水面上不停的飞行，它的速度更新公式为：

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 * e^{-\beta r_p^2} * (pbest_{ij} - x_{ij}^t) + a_2 * e^{-\beta r_g^2} * (gbest_{ij} - x_{ij}^t) \quad (2-2)$$

其中 v_{ij}^{t+1} 是 $t+1$ 时雄性蜉蝣 i 在维度 $j = 1, \dots, n$ 上的速度， x_{ij}^t 是 t 时雄性蜉蝣 i 在维度 $j = 1, \dots, n$ 上的位置； g 是惯性权重； a_1, a_2 是雄性蜉蝣的正吸引系数，用来缩放从自身和社会所获取经验对行为的影响； β 是视力系数，用来限制蜉蝣的行动，而 r_p, r_g 是 x_i 和局部最优和全局最优间的笛卡尔距离。 $gbest_{ij}, pbest_{ij}$ 分别代表当前种群中的局部最优与全局最优，其计算方法可表示为：

$$pbest_i = \begin{cases} x_i^{t+1}, f(x_i^{t+1}) < f(pbest_i) \\ pbest_i, f(x_i^{t+1}) \geq f(pbest_i) \end{cases} \quad (2-3)$$

$$gbest_i = \begin{cases} pbest_i, f(pbest_i) < f(gbest_i) \\ gbest_i, others \end{cases} \quad (2-4)$$

其中 $f(x)$ 表示适应度计算公式，在本文中为欧几里得距离计算公式。

当前种群中的全局最优仍会不断更新它的速度，更新公式为：

$$v_{ij}^{t+1} = g * v_{ij}^t + dance * e \quad (2-5)$$

其中， $dance$ 是舞蹈系数， e 是 $[-1, 1]$ 的随机数。

在 D 维的空间内，第 i 只雌性蜉蝣的位置可表示为：

$$y_i = (y_{i1}, y_{i2}, \dots, y_{iD})^T$$

它的速度表示为：

$$v_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$$

位置更新公式为：

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (2-6)$$

在蜉蝣算法中，雌性蜉蝣的行为是基于雄性蜉蝣的位置来调整飞行路线的。虽然实际过程可能包含随机性，但在算法中通常会简化这一过程，使其变得确定性。这样，通过适应度值进行排序，形成一种配对机制。这种方法有助于算法高效地找到解决方案，同时保持种群的多样性。则速度更新公式可以表示为：

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^t + a_3 * e^{-\beta r_{mf}^2} * (x_{ij}^t - y_{ij}^t), f(y_i) > f(x_i) \\ g * v_{ij}^t + f_i * e, f(y_i) \leq f(x_i) \end{cases} \quad (2-7)$$

其中 v_{ij}^{t+1} 是时间步 $t+1$ 时雌性蜉蝣 i 在维度 $j = 1, \dots, n$ 上的速度, y_{ij}^t 是时间步 t 时雌性蜉蝣 i 在维度 $j = 1, \dots, n$ 上的位置; g 是惯性权重; β 是固定可见度常数; r_{mf} 是雄性和雌性之间的笛卡尔距离, a_3 是雌性蜉蝣的正吸引系数, f_i 是随机游走系数, e 是 $[-1,1]$ 间的随机值, $f(x_i), f(y_i)$ 分别表示第 i 只雄性蜉蝣与第 i 只雌性蜉蝣的适应度。

雄雌蜉蝣互相吸引后, 将会产生新的蜉蝣, 交配公式如下:

$$offspring1 = L * male + (1 - L) * female \quad (2-8)$$

$$offspring2 = L * female + (1 - L) * male \quad (2-9)$$

其中 $male$ 是雄性父代, $female$ 是雌性父代, L 是 $[0,1]$ 的随机值。

2.3.3 算法流程

传统蜉蝣算法流程如图 2-2 所示。

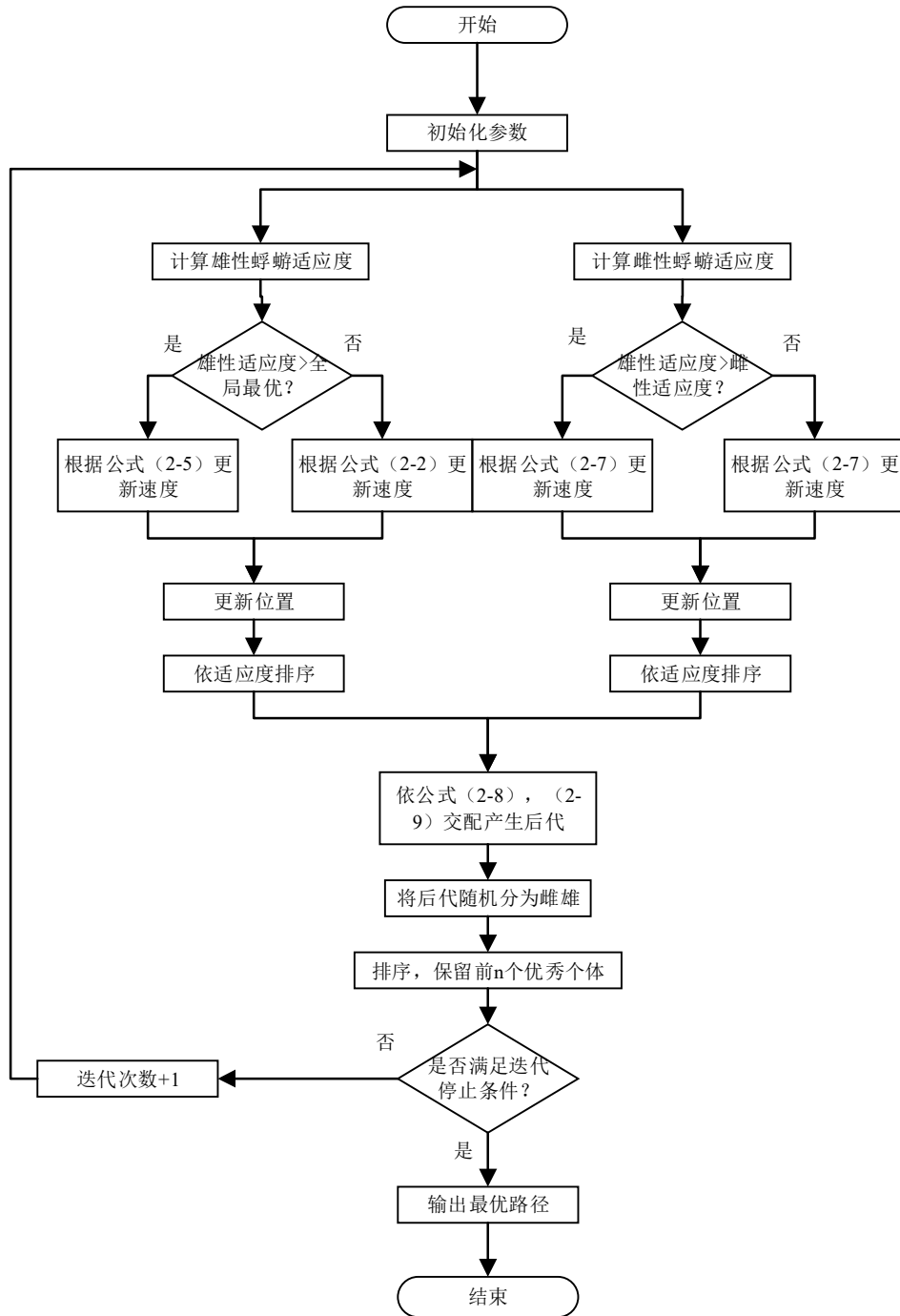


图 2-2 传统萤火虫算法流程图

2.3.4 萤火虫算法存在问题

传统萤火虫算法也存在一些缺点，如收敛速度较慢、精度差，易陷入局部最优，且算法实时性较差等问题，分析问题原因为：

- (1) 搜索能力差。因各参数都为固定值，无法平衡全局搜索与局部搜索，易导致算法搜索能力差。
- (2) 因算法无记忆能力，将会对已搜寻过区域进行重复搜索，导致产生大量

无用迭代。

2.4 DQN 算法

2.4.1 深度学习

深度学习是机器学习的一个重要分支，它通过多层感知器模型来模拟人脑处理信息的层级化方式，从而识别数据中的复杂模式和特征。Torsten Wiesel 和 David Hubel 的研究为深度学习提供了生物学上的启示，他们的发现强调了层级化处理在视觉系统中的重要性。他们的发现暗示了大脑处理信息的过程可能是一个连续的迭代与抽象化过程。这一突破性的研究为理解神经系统的复杂性提供了重要的视角^[49]。人的大脑可以通过迭代抽象对信息进行处理，来进行信息的分辨等功能。深度学习就是通过模拟这一过程来处理数据的。

深度学习近年来不断发展，解决了医学诊断、自动驾驶、自然语言处理等各领域的问题。传统机器学习与深度学习的流程区别如图 2-3 所示。在众多场景中，鉴于问题本身的复杂性，深度学习已被证明能够突破早期浅层网络的限制。这些浅层网络往往限制了有效训练的进行以及多维训练数据的分层抽象表示。相比之下，深度神经网络（DNN）能够凭借其多层次、高度优化的算法和结构有效解决多维度的复杂问题。

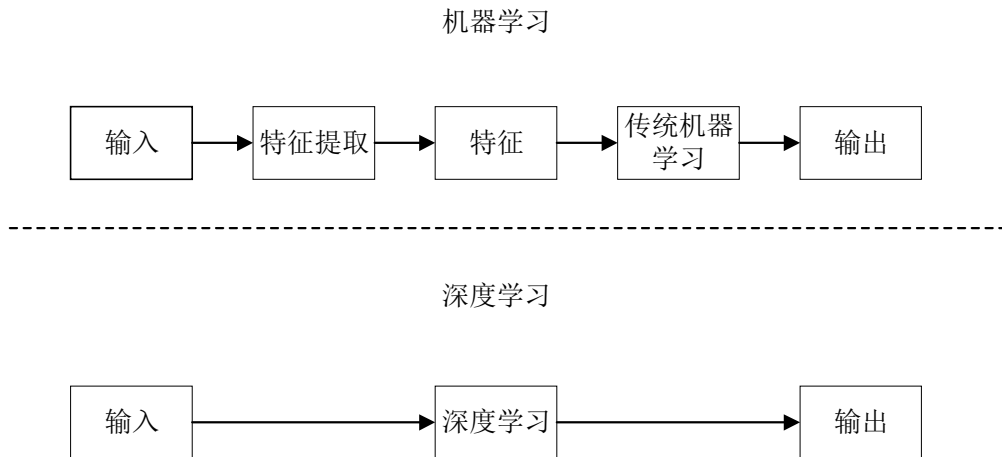


图 2-3 深度学习与传统机器学习流程比较

1998 年，Yann LeCun 提出了 LeNet-5，这是第一个成功的 CNN 架构，通过局部感受野和权重共享有效提高了网络性能^[50]。这一突破性的设计为后续深度学习的发展奠定了基础，从而令网络更易于优化，其次，模型的复杂度得到降低，因此也在一定程度上缓解了过拟合的风险^[51]。

LeNet-5 是一种经典的卷积神经网络架构，它由两个卷积层组成，这些层通过两个池化层进行特征降维。在经过几个全连接层处理后，最终通过 softmax 回归函数实现多类别的输出分类。因此，一个基本的 CNN 的结构包括卷积层、池化层、全连接层^[52]。

（1）卷积层

在卷积层中，卷积核会在输入图像上滑动，并与图像的局部区域进行内积运算，以此产生新的特征图。这个过程会在整个输入特征图上重复进行，直到卷积核覆盖了所有可能的位置。这些卷积核构成了卷积层，是 CNN 能够执行高效图像分析的关键。

卷积计算公式定义为式(2-10)：

$$s(i, j) = (x \times w)(i, j) = \sum_m \sum_n x(i+m, j+n)w(m, n) \quad (2-10)$$

式(2-5)中， $x(i+m, j+n)$ 是输入， $w(m, n)$ 是卷积核。卷积层的参数是由一组可学习的卷积核构成的，这些卷积核通过对输入图像进行卷积操作来提取特征。一个 3×3 的卷积核可以表示为一个包含不同数字的 3 行 3 列的矩阵，如图 2-4 所示。

1	1	1
1	2	3
4	1	2

图 2-4 卷积核

（2）池化层

卷积层是一种线性操作，具有平移不变性，单纯的线性操作有可能会引起过拟合的问题。因此，需要卷积层之后加入池化层来减少这一问题的影响。池化层不同于卷积层，是一种非线性采样，主要目的是减少特征图像的参数从而提高运算速度，其也可以防止过拟合。池化层在 CNN 中具有三个主要功能：增强特征的不变性，减少数据维度以简化模型，以及帮助防止模型过拟合。常用的池化方法包括随机池化、平均池化和最大池化等^[53]，其过程就是取一定范围的像素值的随机值平均值或者最大值来作为新的特征图的像素值。

（3）全连接层

全连接层通常位于卷积神经网络的末端，充当分类器角色。它的主要功能是整合前面卷积层和池化层提取的局部特征信息，通过权重参数融合这些信息，以形成完整的特征表示。

(4) 损失函数

损失函数评估模型预测的准确性，对优化模型至关重要。选择合适的损失函数能显著影响训练速度和性能。回归和分类任务常用的损失函数包括 Softmax 交叉熵、平均绝对误差（MAE）、均方误差（MSE）其计算公式分别如式(2-11)，(2-12)和(2-13)所示。

$$L = -\sum y_i \ln p_i \quad (2-11)$$

$$L1 = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2-12)$$

$$L2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2-13)$$

2.4.2 强化学习

强化学习是机器学习的一种，已在智能机器人中得到广泛应用^[54]。与带有标签数据的“开环”类型的监督学习相比，强化学习是一种“闭环”的且带反馈的机器学习方法，如图 2-5 所示。在这个过程中，智能体必须不断地从与动态且复杂环境的互动中学习，以优化其行为策略。强化学习可以依据智能体与环境交互的经验数据进行学习，而无需人为进行数据标注工作。

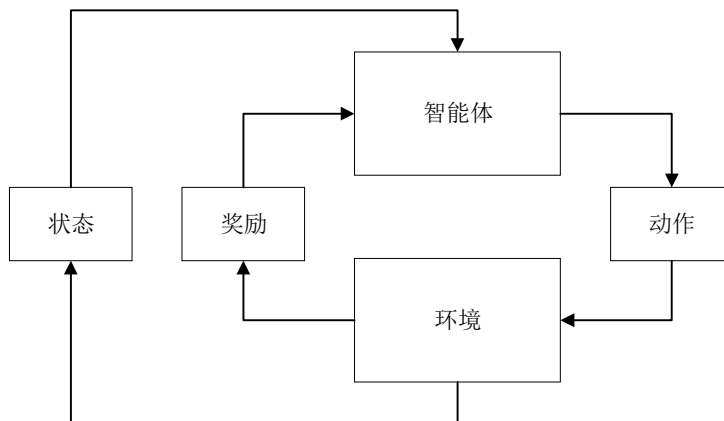


图 2-5 强化学习流程

Q-learning 是由 Watkins 和 Dayan 提出的一种经典强化学习算法，它基于价值来进行动作选择^[55]，涉及智能体如何在特定状态下与环境交互^[56]，Q 学习主

要包括 4 个重要的部分：状态、动作、Q 表和奖励。在某一时间点， $Q(s,a)$ 代表着在特定状态 s ($s \in S$) 下执行动作 a ($a \in A$) 所能带来的预期收益。环境会根据智能体的动作反馈相应的奖励 r 。因此，该算法的核心理念是建立一个 Q 表，将状态和动作对应的 Q 值存储其中，并据此选择能够获得最大收益的行动。智能体在每个状态下的对于每个动作都进行不断试错和学习，在一定的学习后，得到最优动作选择策略。

Q-Learning 算法在处理大规模状态空间时会遇到价值表过大的问题。为了解决这一问题，研究者们将深度学习技术与 Q-Learning 结合，创建了 Deep Q Network(DQN)^[57]。DQN 利用深度神经网络来近似 Q 值，有效处理了传统 Q-Learning 算法的限制。

2.4.3 DQN 算法

Q-Learning 算法在处理大规模状态空间时会遇到价值表过大的问题。针对这一问题，DeepMind 团队在 2015 年提出了 DQN 算法^[58]，DQN 算法将可以提取图片中特征的 CNN 与 Q 学习相结合，集合了深度学习与强化学习的优点，如图 2-6 所示。

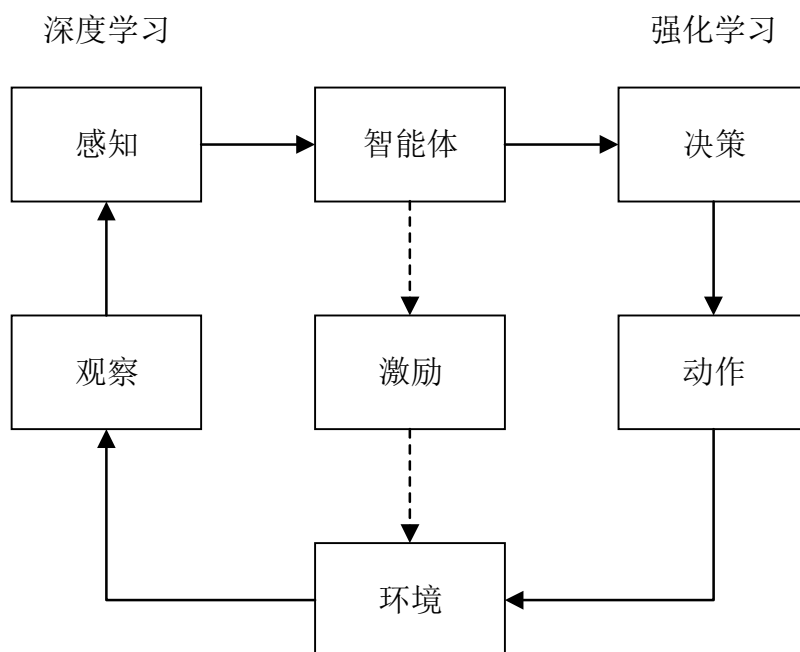


图 2-6 深度强化学习

DQN 算法结合了深度学习和强化学习的优势。它使用 RGB 图像作为输入，这样可以处理高维度的状态空间。Q-learning 部分则负责根据奖励来更新策略，以优化未来的行为。DQN 的关键之一是经验回放，它存储过去的转换，随机抽

取小批量进行学习，这减少了样本间的相关性并提高了学习效率。另一个重要特点是目标网络，它帮助算法稳定学习过程，防止目标值快速变化导致的不稳定。总的来说，DQN 算法通过这些创新步骤，有效地解决了强化学习中的一些关键挑战。

(1) 目标函数。DQN 的目标函数是基于 Q 学习所构建的，其公式如下：

$$Q'(s, a) \leftarrow Q(s, a) + \alpha[r + \max_{a'} Q(s', a') - Q(s, a)] \quad (2-10)$$

损失函数设置为均方误差损失函数，公式为：

$$L(\theta) = E[(y' - Q(s, a, \theta))^2] \quad (2-11)$$

(2) 目标网络。DQN 利用目标网络和预测网络来准确评估当前状态下的动作价值函数。它通过神经网络计算出目标 Q 值，并以此为基础预测下一状态的 Q 值，有效克服了 Q-learning 中因状态-动作空间过大而引发的“维数灾难”。

(3) 经验回放。经验回放机制通过解决数据样本间的关联性问题 and 提高数据利用效率，优化了学习过程。在此过程中，机器人从环境交互中获得的样本被存储在经验池里，训练时从经验池中随机选取一批样本进行训练。通过这一策略，不仅增强了样本的多样性，还使得同一样本得以在多次训练中重复使用，极大提升了学习效率。

2.4.4 DQN 算法存在问题

尽管 DQN 算法在路径规划等诸多领域已展现出其广泛应用价值，但其仍有待改进之处。例如，该算法可能会陷入局部最优解，且在学习效率、稳定性和探索效率方面表现不佳。这些不足可能源于以下几个因素：

(1) 迭代效率低

传统的 DQN 算法在 Q 表的初始化时通常将其设为均等或完全随机，在一些状态下执行某种动作是可以达到全局最优，但在初始化时概率设置过低，使得它被访问的概率很低。这使得在尺度较大的未知环境中，容易产生大量的无效迭代。

(2) 数据探索效率低下

在路径规划应用中，DQN 算法有时会过分强调最大奖励，忽略探索，导致效率下降。为提升性能，应设定合理探索策略，确保机器人全面学习环境^[59]。

第3章 基于改进深度强化学习的移动机器人路径规划研究

传统路径规划算法依赖已知环境信息，限制了其在未知领域的应用。但移动机器人的自主探索和学习能力对于有效的路径规划至关重要，这也是深度强化学习受到重视的原因。

而传统 DQN 算法在使用时探索与收敛速度时常较慢，且所规划路径无法满足路径长度最短的优化指标，这就需要对这些问题进行改进。针对传统 DQN 算法在解决路径规划问题时存在的不足，本章提出一种改进的 DQN 算法，主要改进方法为改进探索策略，引入人工势场初始化 Q 值与基于双指针的路径优化策略，并进行仿真验证其求解性能。

3.1 改进的探索策略

在解决问题的过程中，智能体必须在探索新行为和利用已知信息之间找到平衡。这是强化学习中的核心挑战之一。在学习初期，智能体对环境知之甚少，因此必须进行探索。随着时间的推移，智能体会逐渐了解环境，并能够预测不同行为的潜在回报。为了解决这一问题，一些学者尝试在算法的探索策略中加入 ϵ 贪婪策略。

ϵ -贪婪策略是强化学习中常用的一种方法，用于在探索和利用之间取得平衡。在这种策略下，智能体以一定的概率 ϵ ($0 < \epsilon < 1$) 选择一个随机动作，以 $1 - \epsilon$ 的概率选择当前估计最优的动作。这样既保证了对环境的充分探索，又能利用已有的知识来做出决策。

但这一策略仍存在其局限性，虽然理论上每个动作都有被选择的可能，但是随机性过高，有一些状态下执行某种动作时可以达到全局最优，但这一动作在初始化时概率设置过低，使得它被访问的概率很低。且固定的 ϵ 值并不能很好平衡探索与利用，前期过高的 ϵ 值使得前期训练重心分出过多在了利用历史信息，后期 ϵ 不足又会使利用历史信息部分过少，训练不平衡。

针对这一问题，本章提出一种自适应线性贪婪策略，将 ϵ 值设定为一个随迭代次数而逐渐减小的值，这样可以使得智能体在前期将训练重心放在探索新的动作上，在后期将训练重心放在利用历史信息上，有效地保持了探索新的动作和利用历史信息的平衡。本章将 ϵ 的值设定为：

$$\varepsilon = \varepsilon_0 + \xi \times \left(\frac{e^{iteration/0.1iteration_max} - e^{-iteration/0.1iteration_max}}{e^{iteration/0.1iteration_max} + e^{-iteration/0.1iteration_max}} \right) \quad (3-1)$$

ε_0 表示探索因子的初始值, $iteration$ 代表迭代次数, $iteration$ 的取值范围设定为 $[0, iteration_max]$, 其中 $iteration_max$ 是最大迭代次数, ξ 是一个实数, 代表动态系数, 其取值范围为 $0 < \xi < 1$ 。在本文中取 $\varepsilon_0 = 0.1$, $iteration_max = 1000$, $\xi = 0.8$, 因此 ε 的值变化范围为 $[0.1, 0.9]$, 探索因子 ε 的值会随迭代次数的增加而调整, 其变化曲线如图 3-1 所示。

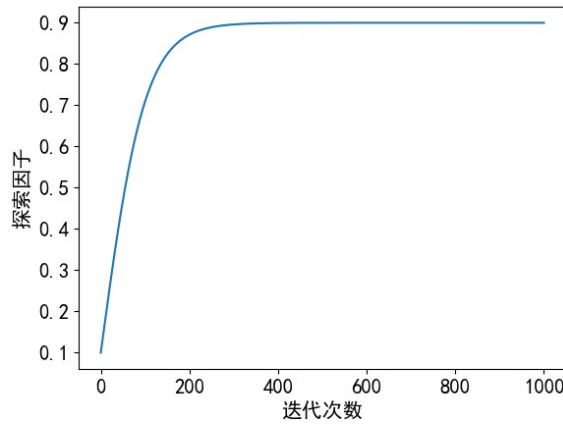


图 3-1 探索因子

3.2 引力势场

在 DQN 算法的传统应用中, Q 表初始通常随机, 导致动作选择具随机性。智能体在新状态基于即时奖励更新价值函数。尽管某些动作可能理想, 但初始概率低, 选中几率也低。在大型未知环境的路径规划初期, 这可能导致无效迭代, 影响效率。

在本章中, 对 DQN 算法进行了改进, 引入了人工势场理念以初始化 Q 值。由于地图上障碍物众多, 直接计算每个障碍物的斥力会增加计算负担。因此, 我们在初始化 Q 表时, 没有考虑障碍物的斥力, 而是只考虑了目标点的引力。为了引导动作选择向 Q 值增长的方向发展, 本章设计了一个与距离成反比的引力势函数, 建立了引力势场。具体的引力势场函数表达式如下所示, 见公式(3-2)。

$$U = \frac{\zeta(l_{aim})^2}{l_{max} - l_{aim}} \quad (3-2)$$

其中 ζ 设为引力因子, l_{max} 是地图最大长度, l_{aim} 是距目标点长度。

通过公式 (3-3) 来实现 Q 值的初始化。

$$Q(s,a) = r + \omega U(s_i) \quad (3-3)$$

r 是当前状态下的奖励, ω 为 [0,1] 的随机数。

3.3 基于双指针的路径优化策略

通过对环境的持续探索, 智能体能够在环境中找到由初始点到目标点的可行路径。这些路径确保了智能体能够从起点到达终点完成路径规划任务。然而, 这些路径并不一定是最优的, 因为它们可能不满足转折点最少或路径最短等优化条件。

为了实现路径转折点最少、路径最短等优化指标, 国内外一些学者对智能体的探索步长进行了改进, 如文献[60]中, 将每个栅格由 1 个状态扩展为 8 个状态, 在动作集中选择只包含相对于当前方向的前、后、左前和右前四个方向。该文献的方法虽然取得了进展, 但增加的运动方向同时也导致了 Q 表维度的增加, 使得算法的求解速度变低。

基于此, 本章提出一种基于双指针的路径优化策略来对路径进行优化, 消除路径冗余部分并减少路径转折。该方法主要由起始指针确定, 目标指针确定, 安全判断三个步骤组成。

- (1) 由起始点至目标点的路径坐标点依次作为起始指针;
- (2) 由目标点至起始点的路径坐标点依次作为目标指针;
- (3) 采用公式计算由起始指针至目标指针形成的矩形区域内的各障碍物到起始指针至目标指针所连接直线的距离, 并判断距离是否大于安全距离, 即障碍物到直线的距离 d_i 是否满足式(3-4)。

$$d_i \geq \frac{\sqrt{2}}{2} d_0 \quad (3-4)$$

其中 d_0 为安全距离, 即移动机器人半径, 本章在仿真时将其设定为 1。

3.4 算法流程

本章改进 DQN 算法伪代码如下所示。

改进的 DQN 算法

输入：状态空间 S ，动作空间 A ，折扣因子 β ，学习率 α ，探索因子 ε

输出：行动策略

初始化回放经验池 D 的容量为 M ；

随机初始化动作值函数 Q 的参数 θ ；

随机初始化目标动作值函数 Q 的参数 $\theta^- = \theta$

For iterations=1,M

 初始化起始序列和预处理序列

 For t=1,T

 计算自适应值

 以概率选择动作 a

 执行动作 a ，得到新状态 s' 和奖励 r ；

 将 (s, a, r, s') 放入 D 中；

 从 D 中采样多组数据

$$y_j = \begin{cases} r_j, & \text{if end at } j+1 \\ r_j + \gamma \max_a \hat{Q}(s_j', a_j'; \theta^-), & \text{otherwise} \end{cases}$$

 以损失函数为 $((y_i - Q(s_j, a_j; \theta))^2)$ 应用梯度下降法进行训练；

$S \leftarrow s'$ ；

 每隔 C 步， $Q \leftarrow Q$ ；

End For

End for

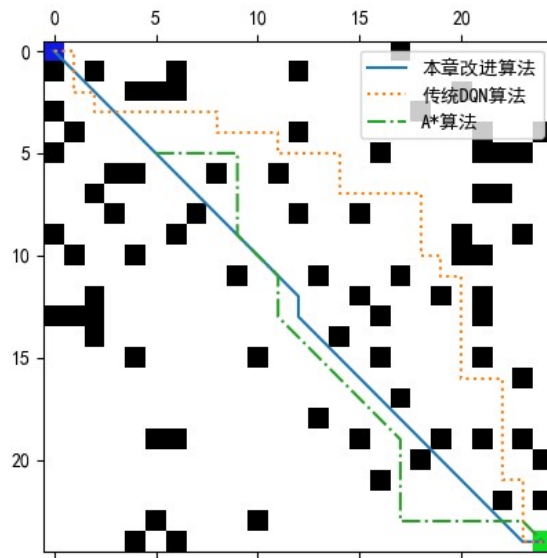
对所得路径进行优化

3.5 仿真测试

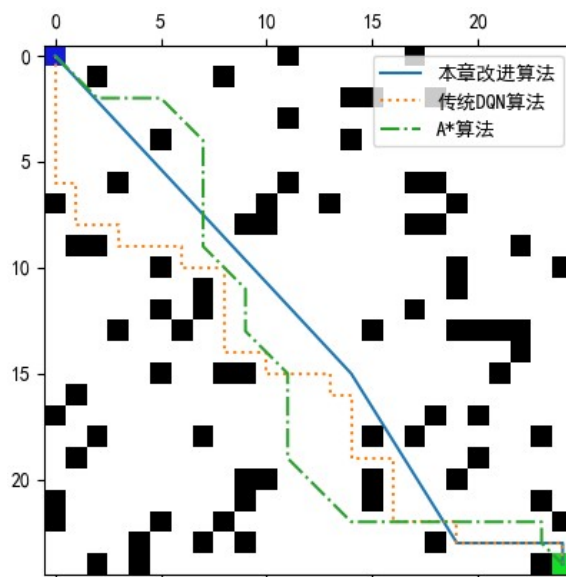
为验证此改进算法的有效性，首先在静态 25×25 环境下运用 python 语言编程进行路径规划仿真测试，而后分别在文献[61]中的 11×11 与 15×15 的环境下与文献[61]算法进行比较。本章改进算法参数设置为：学习率 $\alpha=0.3$ ，折扣因

子 $\beta = 0.95$ ，经验池（Memory Size）最大尺寸为 500，探索因子 ε 的初始值为 0.1，引力因子 ζ 为 0.5。

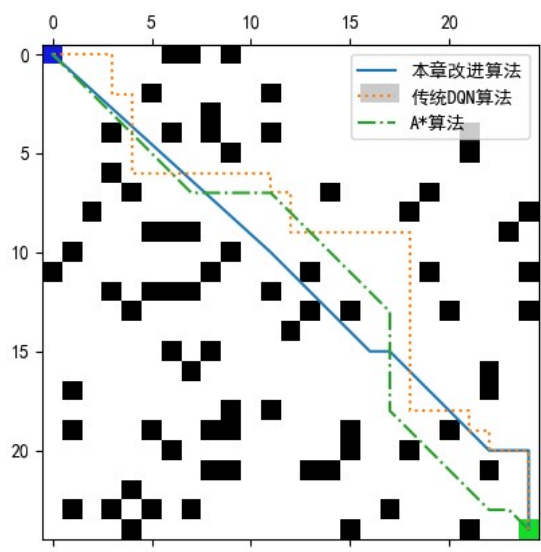
为了验证本章改进算法的有效性和稳定性，首先采用类似图 2-1 中 25×25 的栅格环境，在该环境下对该算法进行仿真，并与传统 DQN 算法和 A* 算法进行比较，为了保证实验结果的可靠性，对三种算法分别在障碍物随机的 8 个不同的 25×25 的栅格环境模型下进行 8 次独立实验，并计算每次实验的路径长度与路径长度的平均值，各次实验路径优化对比结果如图 3-2 所示，每次实验的路径长度对比如图 3-3 所示，多次独立实验平均值统计结果如表 3-1 所示。



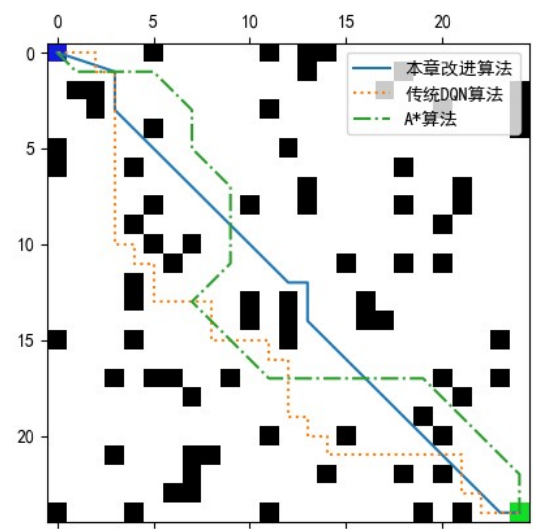
环境 1 下的路径对比(a)



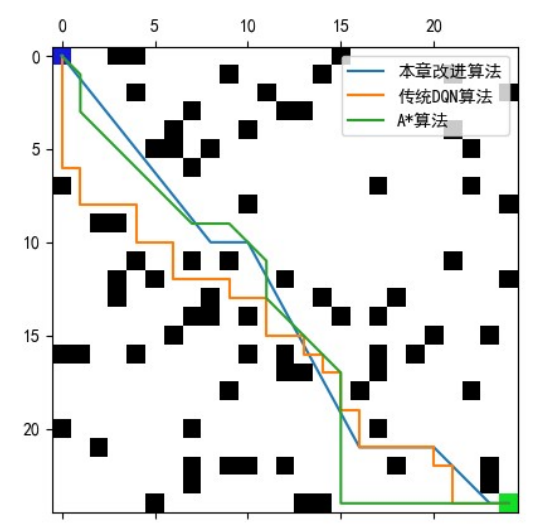
环境 2 下的路径对比(b)



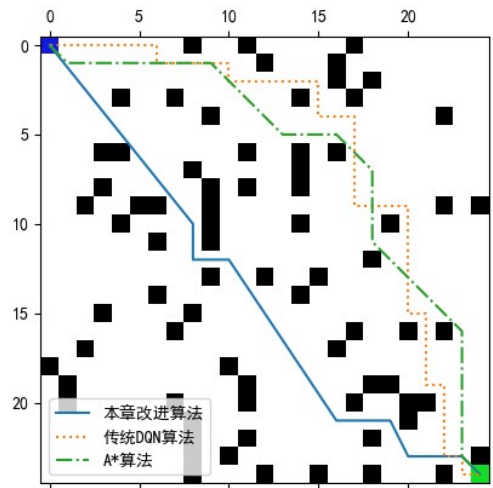
环境 3 下的路径对比(c)



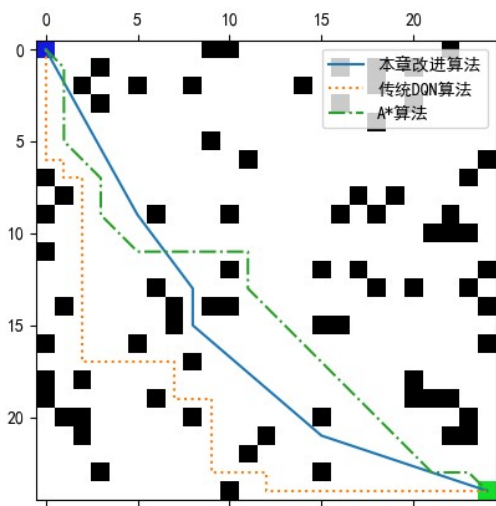
环境 4 下的路径对比(d)



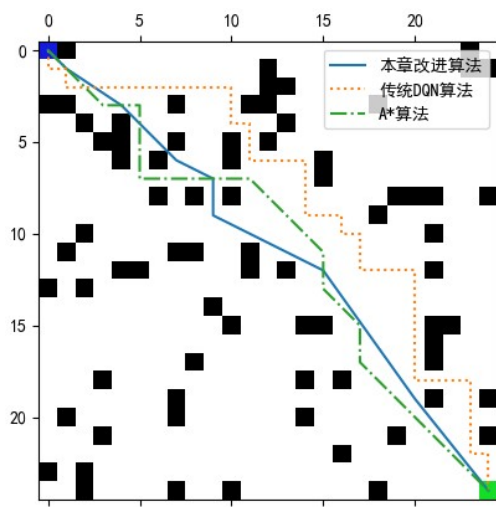
环境 5 下的路径对比(e)



环境 6 下的路径对比(f)



环境 7 下的路径对比(g)



环境 8 下的路径对比(h)

图 3-2 路径对比图

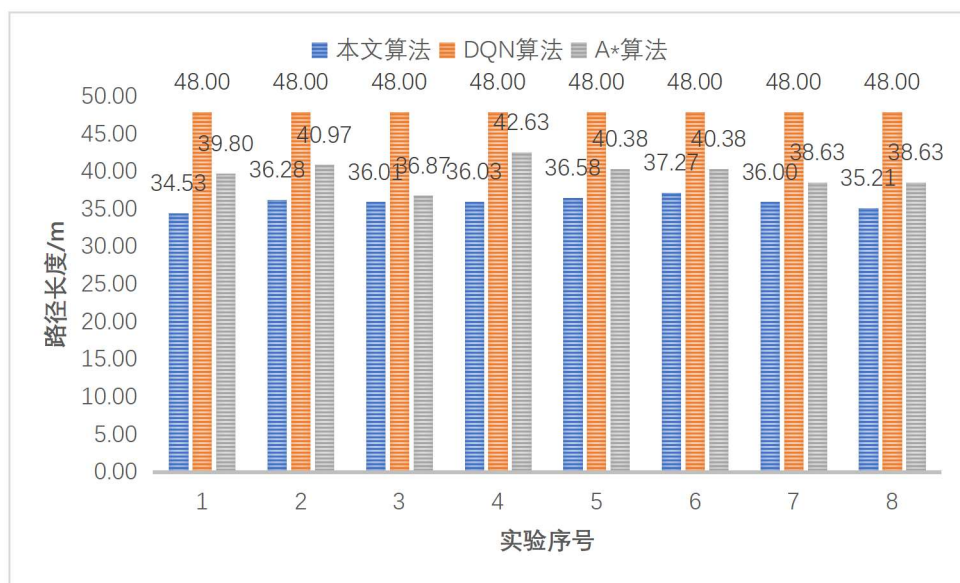


图 3-3 各环境下实验路径长度对比

表 3-1 算法对比结果

	路径长度平均值/m
传统 DQN 算法	48
A*算法	39.786426740
改进 DQN 算法	35.989629704

由图 3-2, 图 3-3 可知, 本章改进算法在环境中规划出了平滑且距离短的一条路径, 传统 DQN 算法因只有 4 个方向路径长度较长且转折多, A*算法虽因方向设置为 8 个方向路径长度有所缩短但仍具有较长的路径长度与较多的转折, 而本章改进算法路径长度明显较传统 DQN 算法与 A*算法更短, 且转折更少, 安全性更好, 能耗更低。由表 3-1 可以看出, 本章改进算法所得路径长度平均值明显更短, 传统 DQN 算法路径规划路径长度平均值为 48, A*算法所规划路径长度平均值为 39.786426740, 而本章改进算法仅为 35.989629704, 较传统 DQN 算法与 A*算法分别提升了 33.4%与 10.6%。

为了进一步验证本章改进算法的有效性, 故使用本章改进算法在文献[61]中的 11×11 与 15×15 的环境下并与其 AGR-DDQN 算法进行比较, 仿真对比结果如图 3-4, 图 3-5 所示。

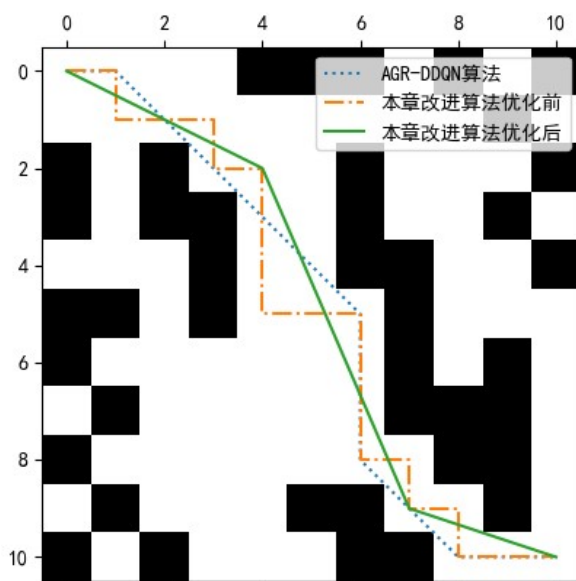


图 3-4 路径对比图

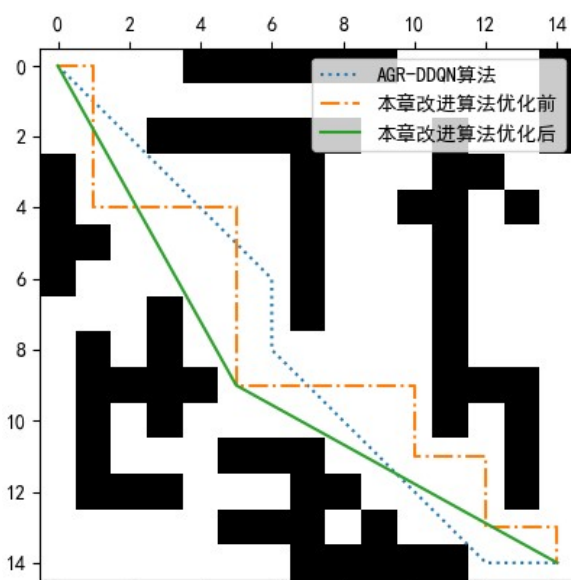


图 3-5 路径对比图

分析图 3-4，图 3-5 的仿真结果可知，本章改进算法所得最终路径长度分别为 15.250 与 20.591，文献[61]中 AGR-DDQN 算法所得最终路径长度为 15.9005 与 20.971，本章改进算法与文献[61]中 AGR-DDQN 算法相比，在路径长度上分别缩短了 4.26%与 1.85%，转折点数量也减少了 50%与 66.7%。由此可知，本章改进算法最终规划出的路径长度更短，更平滑。

本章改进算法经多次试验结果都较为稳定，由此可得，本章改进算法可以有效稳定地解决移动机器人在各种环境中的路径规划问题。

第 4 章 基于改进蜚蜉算法的移动机器人路径规划研究

DQN 算法在解决简单环境下的路径规划问题时取得了一定的成果，然而当环境的复杂度持续增加后，与环境的交互中所的交互数据将会大量增加，无法有效完成路径规划任务。

蜚蜉算法一种新型智能优化算法，在面对高维复杂优化问题时因其收敛精度、速度的优势而备受青睐。但蜚蜉算法应用于路径规划问题时，常常出现局部最优探索能力不足，迭代速度慢等问题。针对传统蜚蜉算法在路径规划问题中的不足，本文提出一种改进蜚蜉算法，采用自适应动态参数调整策略对蜚蜉算法多个参数进行调节，将能量消耗加入适应度函数中，并引入莱维飞行策略对蜚蜉飞行过程进行优化^[62]。

4.1 自适应参数调整策略

参数的设置对算法的结果是影响巨大的，参数的合理设置可以使算法能力得到充分发挥，在本章改进蜚蜉算法中重要参数有：惯性权重 g ，正吸引系数 a_1 、 a_2 和 a_3 。由雄性蜚蜉进行主要寻优工作，雌性蜚蜉进行辅助，故将 g 、 a_1 和 a_2 设为自适应非线性变化， a_3 设置为一固定值。

在传统蜚蜉算法中，惯性权重，正吸引系数通常被设置为一固定值，这使得局部搜索能力与全局搜索能力不平衡，易陷入局部最优，也限制了迭代速度。在将惯性权重设置为自适应参数后，可以在前期使蜚蜉拥有较高的速度，提高前期探索效率，而在后期使蜚蜉速度降低，增加蜚蜉寻优能力。

在本改进算法中将惯性权重设置为：

$$g = \frac{-0.55}{e^{1-(t/\frac{t_{\max}}{2})^{0.05}}} + 1.2 \quad (4-1)$$

若只将惯性权重设为自适应，则在寻优过程中脱离局部最优的能力仍旧不足，故将吸引系数也设为自适应值，这样使得算法局部寻优和全局寻优保持平衡，在迭代前期更多依靠自身经验，减小群体影响，增加解的多样性，在后期多依赖群体经验以避免局部最优。

将正吸引系数设置为：

$$a_1 = \frac{3}{e^{1-(t/\frac{t_{\max}}{2})^{0.05}}} \quad (4-2)$$

$$a_2 = \frac{-3}{e^{1-(t/\frac{t_{\max}}{2})^{0.05}}} + 4 \quad (4-3)$$

当最大迭代次数 $t_{\max}=200$ 时，各参数变化曲线如图 4-1，图 4-2 和图 4-3 所示。

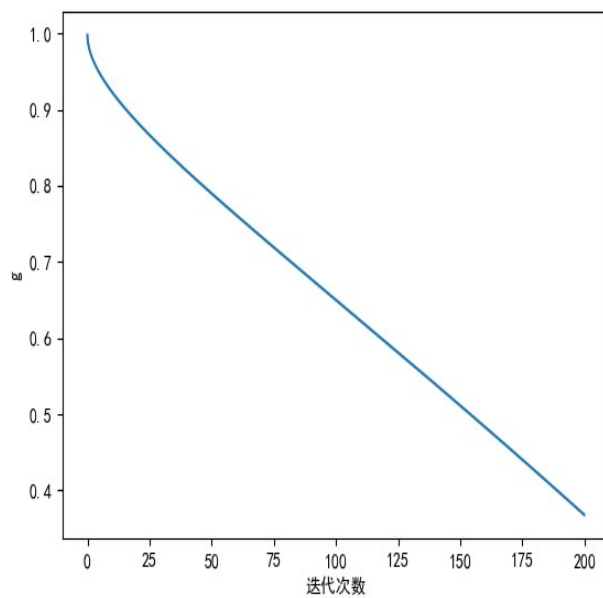


图 4-1 惯性权重 g 变化曲线

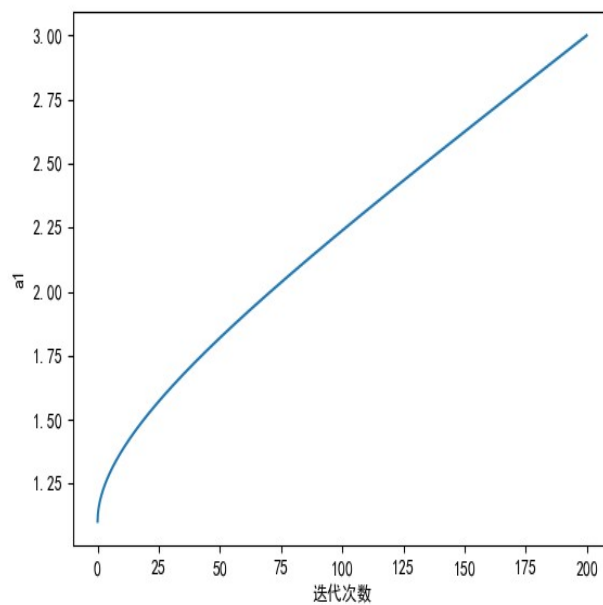
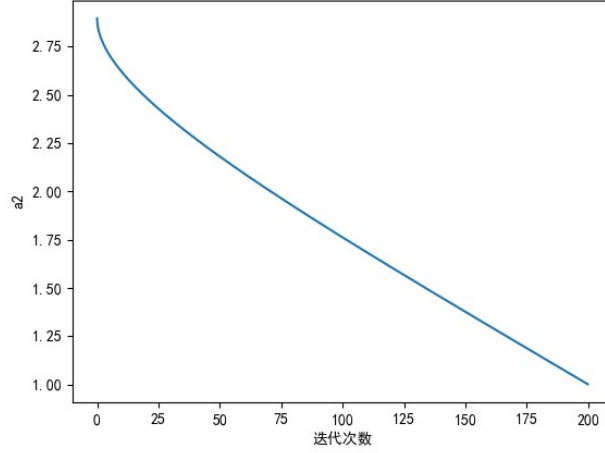


图 4-2 正吸引系数 a_1 变化曲线

图 4-3 正吸引系数 a_2 变化曲线

4.2 莱维飞行

莱维飞行是指在飞行中有一定几率出现随机步长^[63]。许多鸟类的飞行过程都符合莱维飞行策略，蜉蝣在飞行过程中的步长也应是符合这一策略的，故在本章改进蜉蝣算法中引入莱维飞行策略。通过对上次迭代的全局最优位置进行扰动，使得解更具有多样性，帮助其跳出局部最优，提升了解的搜索效率。引入莱维飞行策略后，雄性蜉蝣的速度更新公式为：

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 * e^{-\beta r_p^2} * (pbest_{ij} - x_{ij}^t) + a_2 * e^{-\beta r_g^2} * (gbest_{ij} \oplus Levy(\beta) - x_{ij}^t) \quad (4-4)$$

其中， $\oplus$ 为对点乘法， $Levy$ 是莱维飞行步长，本文使用 Mantegna 方法生成服从莱维分布的随机步长，表达式为：

$$Levy(\beta) = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (4-5)$$

其中 $u \sim N(0, \sigma^2)$, $v \sim N(0, 1)$ 。

$$\sigma = \left\{ \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\beta \Gamma\left(\frac{1+\beta}{2}\right) 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}} \quad (4-6)$$

通常， $1 \leq \beta \leq 3$ ，在本文中取 $\beta=1.5$ 。

4.3 改进的适应度函数

在移动机器人运行中，影响能量的消耗的主要有两个因素，一是路径的长度，二是转弯的次数与平滑度，移动机器人在每次转弯时都要进行一个减速-转

弯-加速的过程，这一过程将会比直行消耗更多的能量，在一般的路径规划研究中常常只考虑路径的长度而忽略了转弯对能耗的影响。本文将转弯次数纳入适应度函数计算中，为了方便计算，本文忽略转弯的平滑度对能耗的影响，将每次转弯的额外消耗视为定值，改进后的适应度函数如下。

$$L = \sum_{i=1}^D \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} + D \times \psi, i = 1, 2, \dots, D \quad (4-7)$$

其中， x, y 是节点所在位置对应的栅格地图中位置， D 是路径节点数量， ψ 为一常数，表示转弯额外消耗。

4.4 算法运行步骤

本章改进蜉蝣算法的流程图如图 4-4 所示。

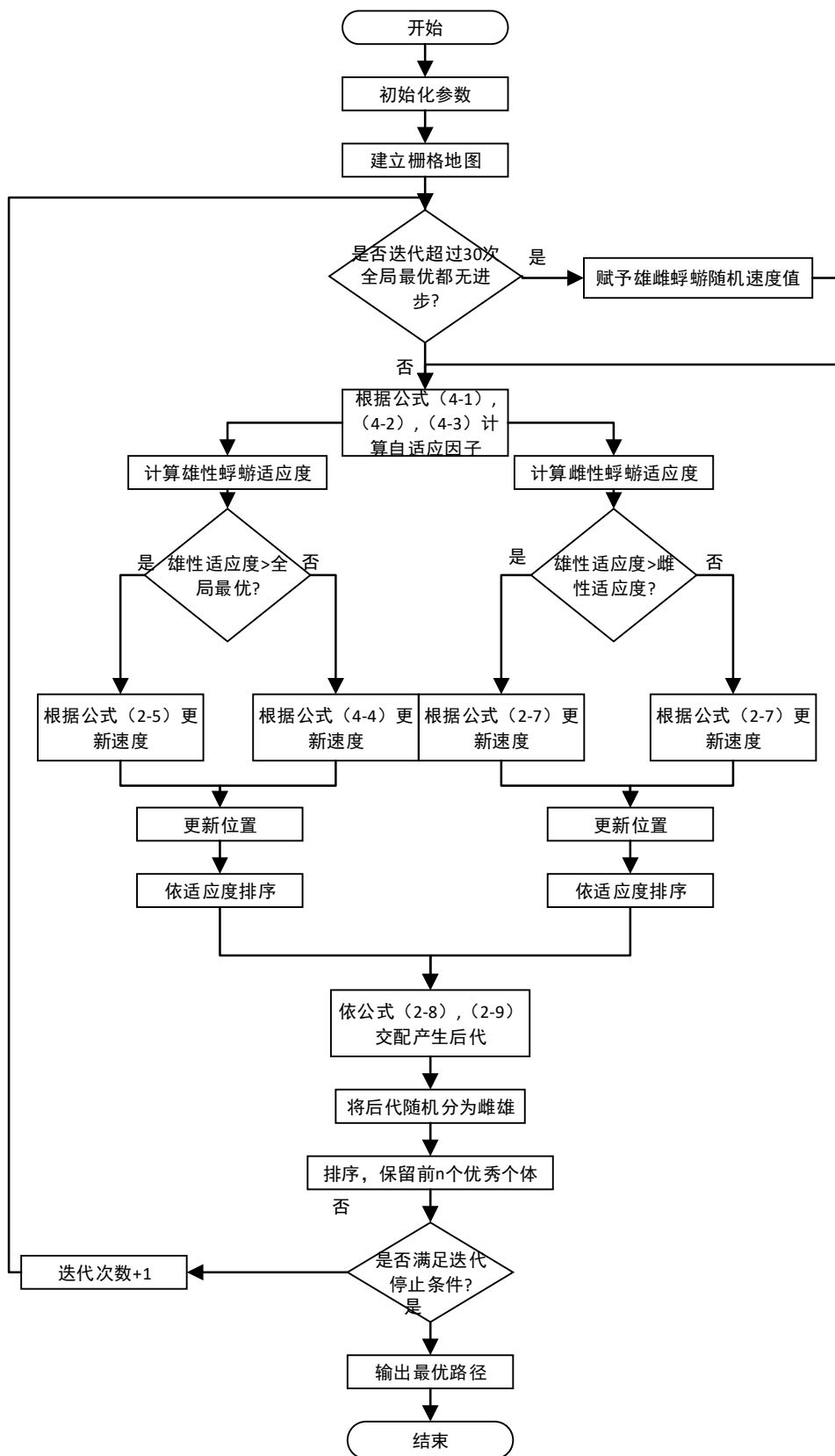


图 4-4 算法流程图

4.5 仿真分析

为验证本章改进蜉蝣算法的优越性和合理性,运用 python 语言编程,选取文献[64]中的 20×20 的栅格环境模型,在此模型下对该算法进行仿真,并与传统蜉蝣算法进行比较。传统蜉蝣算法中惯性权重及正吸引系数依文献[39]设定为:惯性权重 $g=0.8$,正吸引系数 $a_1=1.0$, $a_2=1.5$, $a_3=1.5$,其余参数为:雄雌蜉蝣种群规模均设 10,随机游走系数 $fl=0.5$,速度最大值 $Vel_{max}=20$,速度最小值 $Vel_{min}=-20$,最大迭代次数 $n=100$,舞蹈系数 $dance=0.5$,本章改进蜉蝣算法中 g 、 a_1 和 a_2 设置为自适应参数,其余参数同上。

仿真结果如图 4-5 和图 4-6 所示。

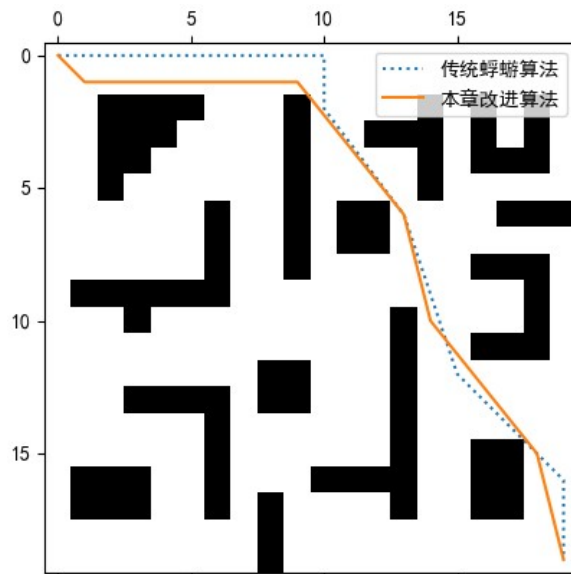


图 4-5 仿真路径图

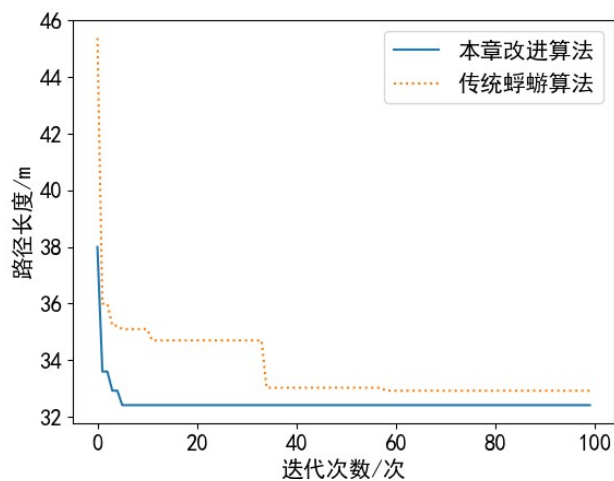


图 4-6 算法收敛图

分析仿真结果可以看出，传统蜉蝣算法所得最优路径长度为 32.9224 长度单位，最小迭代次数为 58 代，而本章改进算法所得路径长度为 32.4137，最小迭代次数为 6 代，因此，本章改进算法相对比于传统蜉蝣算法，其能以更快的速度找到更优的路径。

为了验证改进蜉蝣算法的稳定性，将算法独立进行 20 次运算，对各数据取平均值，并与传统蜉蝣算法和文献[64]算法进行对比，数据如表 4-1 所示。

表 4-1 运行 20 次仿真结果

算法	路径长度	最短路径拐点数量（向上取整）	算法收敛迭代次数（向上取整）
传统蜉蝣算法	33.4578	6	63
文献[64]算法	34.7154	12	30
改进蜉蝣算法	32.5365	5	8

通过表 4-1 的对比仿真结果可以看出，本章的改进蜉蝣算法在迭代过程中路径长度与平均值均较优，且迭代速度快，路径转折少。

本章改进蜉蝣算法有效改善了传统蜉蝣算法求解精度差，收敛速度慢等缺点，使得算法能够有效脱离局部最优，且加快了迭代速度。

为了进一步验证本章改进蜉蝣算法的优越性，并对比在复杂环境下与其他算法的表现，选取文献[65]中的 30×30 栅格环境地图进行仿真并与其进行对比。传统蜉蝣算法中惯性权重及正吸引系数依文献[39]设定为：惯性权重 $g=0.8$ ，正吸引系数 $a_1=1.0$ ， $a_2=1.5$ ， $a_3=1.5$ ，其余参数为：雄雌种群规模均设 10，随机游走系数 $fl=0.5$ ，速度最大值 $Velmax=20$ ，速度最小值 $Velmin=-20$ ，最大迭代次数 $n=100$ ，舞蹈系数 $dance=0.5$ 。本章改进蜉蝣算法中 g 、 a_1 和 a_2 设置为自适应参数，其余参数同上。文献[65] 算法参数设置为：蚂蚁个数为 50，最大迭代次数为 100，信息启发式因子 $\alpha=1$ ，期望启发式因子 $\beta=7$ ，信息素总量 $Q=1$ ，信息素残留系数 $\rho=0.9$ 。

仿真结果如图 4-7 和图 4-8 所示。

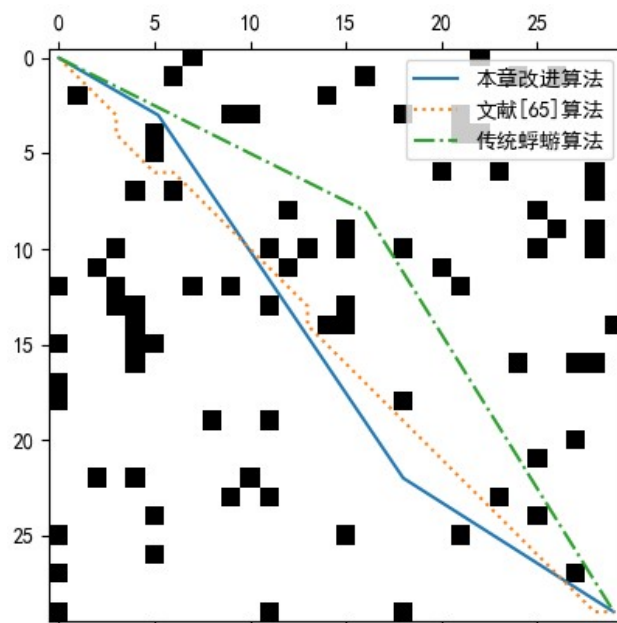


图 4-7 仿真路径图

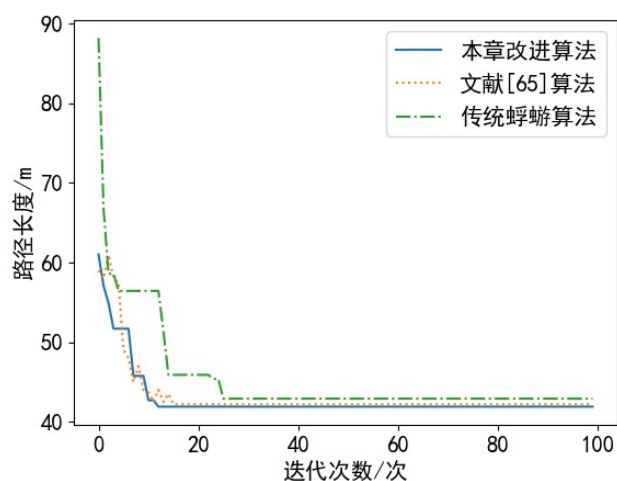


图 4-8 算法收敛图

分析图 4-8 中的仿真结果可得，文献[65]中的改进算法经 15 次迭代得到最优路径长度 42.18；而本章改进蜉蝣算法在 13 次迭代后得到最优解 41.89，此外，两个算法所得最优路径的转角次数分别为 7 个和 2 个。明显可知，本章改进蜉蝣算法不但收敛速度较快，距离更短，且路径更加平滑。为验证算法稳定性，将算法独立运行 20 次，对各数据取平均值，并于传统蜉蝣算法和文献[65]算法进行对比，数据如表 4-2 所示。

表 4-2 运行 20 次仿真结果

算法	路径长度/m	最短路径拐点数量（向上取整）	算法收敛迭代次数（向上取整）
----	--------	----------------	----------------

传统蜉蝣算法	43.37	5	30
文献[65]算法	42.24	12	16
本章改进蜉蝣算法	42.06	3	14

通过表 4-2 的对比仿真结果可以看出，在复杂环境下，传统蜉蝣算法表现能力较差，迭代速度较慢，且最终结果不稳定，而文献[65]算法虽然收敛速度快，但是路径拐点过多，能耗较高，而本章的改进蜉蝣算法，能有效提高提高算法的收敛速度，寻优结果和平滑度相对比于传统蜉蝣算法以及文献改进算法更好。

本章改进蜉蝣算法在复杂环境下也能快速找到最优路径，进一步证明了该改进算法的优越性和稳定性。

第 5 章 基于深度强化学习和蜉蝣算法的移动机器人动态路径规划研究

深度强化学习算法和蜉蝣算法单独应用于路径规划问题时，深度强化学习算法只能解决简单环境下局部路径规划问题，在环境复杂度持续增高后可能无法规划出一条合适的路径。蜉蝣算法也存在迭代速度不足，不能平衡局部搜索能力与全局搜索能力，易陷入局部最优等问题，且不适用于动态环境。

因此，本章考虑到将深度强化学习算法、动态窗口法与蜉蝣算法改进后相结合，可能会对解决复杂环境下的动态路径规划问题具有不错的效果。

5.1 结合 DQN 的自适应参数控制策略

参数的设置对算法的结果是影响巨大的，参数的合理设置可以使算法能力得到充分发挥，在本章改进蜉蝣算法中重要参数有：惯性权重 g ，正吸引系数 a_1 、 a_2 和 a_3 。

在传统蜉蝣算法中，惯性权重，正吸引系数通被设置为一固定值，这一设置并不合理，将会使得局部搜索能力与全局搜索能力不平衡，易陷入局部最优，也限制了迭代速度。在本章中融合 DQN 算法对参数进行自适应调整，根据每一蜉蝣的适应度与迭代次数和当前平均适应度选取合适的参数。本章将适应度，迭代次数，当前平均适应度视为 DQN 的状态，将参数的选取作为 DQN 的动作。将惯性权重设置为自适应参数后，可以赋予适应度低的蜉蝣较大的速度，提高了搜索全局最优解的能力，而对于适应度较高的蜉蝣，则降低速度不使蜉蝣速度过大而偏离最优位置区域，错失全局最优解而陷入局部最优。将正吸引系数设置为自适应后，使得算法局部寻优和全局寻优保持平衡，可以使得适应度低的蜉蝣从自身获取经验增多，增大算法结果多样性，避免了陷入局部最优，并使适应度高的蜉蝣经验更多来自于群体，增大了结果的精度。

本章将适应度，迭代次数，当前平均适应度视为 DQN 的状态，为简化计算，根据当前蜉蝣与全局最优蜉蝣的适应度、全局蜉蝣的平均适应度之间的相对性能，将蜉蝣的状态进行分级，标记为 s ($s=1, 2, 3, 4$)，如表 5-1 所示。

在表 5-1 中， f_i 表示蜉蝣 i 的适应度与全局最优蜉蝣的适应度之差； Δf 为蜉蝣的平均适应度与全局最优蜉蝣的适应度之差。

表 5-1 蜉蝣状态

适应度	$s_{i,t}$
$0 \leq f_i < \Delta f / 10$	1
$\Delta f / 10 \leq f_i < \Delta f$	2
$\Delta f \leq f_i < \Delta f * 1.5$	3
$\Delta f * 1.5 \leq f_i$	4

本章将参数的选取作为 DQN 的动作，据当前蜉蝣的状态以及 DQN 中的 Q 函数，蜉蝣将执行相应的动作 $A_{i,t}$ ($A_{i,t}=1,2,3,4$)，对应动作与参数取值结果如表 5-2 所示。

表 5-2 蜉蝣动作与参数取值的关系

$A_{i,t}$	g	a1	a2	a3
1	1	3	1	1
2	0.8	2	2	2
3	0.6	1	3	3
4	0.4	0.5	3	3

同时，DQN 需要根据蜉蝣的行为，通过相应的奖励和惩罚，对目标函数进行更新，本章对奖励的设定为：若蜉蝣 i 经此次更新后适应值变得更优，则奖励 R_t 设为 5，若适应值变差，则奖励 R_t 设为 -5，更新公式为：

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \times (R_t + \beta \max_{a'} Q(s_{t+1}, \varepsilon) - Q(s_t, a_t)) \quad (5-1)$$

其中， α 为学习率，表示学习到的知识所占权重， β 是折扣因子，表示未来奖励对现状影响， R_t 表示智能体在状态 s_t 下执行动作 a_t 所得奖励， $Q(s_t, a_t)$ 表示智能体在状态 s_t 下执行动作 a_t 所能获得的总奖励， s_{t+1} 表示下一状态， a' 为下一状态的最大 Q 值动作。

5.2 记忆机制

免疫记忆机制是免疫算法中的一个核心组成部分，它能够显著提升搜索过程的速度，从而提高整体的优化性能。通过将经过筛选的优质抗体存储在记忆细胞中，当面对相似的问题时，免疫算法能够迅速地调用这些记忆细胞，以产生更加适合当前问题的优秀解决方案，甚至是最佳解决方案。这种机制不仅加快了问题解决的速度，而且还提高了解决方案的质量和效率。为加快算法迭代速度，本章引入免疫算法中的记忆机制，建立记忆库，将每一次迭代中的每个蜉蝣的位置更新后进行检验该位置是否已储存于记忆库中，若已被储存，则将蜉蝣位置进行再次更新，并再次进行检验，直至更新后位置未被储存，则将该新位置赋予蜉蝣并存储于记忆库中。通过记忆机制去除大量无用迭代，加快了算法的效率与迭代速度。

5.3 改进的动态窗口法

动态窗口法通过当前状态的采样计算下一步的合理轨迹，有效避开障碍物，但缺乏全局视野可能导致局部最优。本文提出融合蜉蝣算法的路径规划，先全局规划再局部优化，确保全局最优与高效率。

5.3.1 运动模型

动态窗口法通过速度空间 (v_t, ω_t) 采样和轨迹模拟来确定最优路径。线速度和角速度变化揭示了机器人的运动状态。假设机器人在时间 Δt 内移动，其运动模型表示为：

$$\begin{cases} x = x + v_x \Delta t \cos \theta_t - v_y \Delta t \sin \theta_t \\ y = y + v_x \Delta t \sin \theta_t + v_y \Delta t \cos \theta_t \\ \theta_t = \theta_t + \omega_t \Delta t \end{cases} \quad (5-2)$$

其中 x 表示机器人的 x 坐标， y 表示机器人的 y 坐标， v_x ， v_y 表示机器人在 x, y 方向上的线速度， ω 表示角速度， θ 表示机器人的航向角。

在本章中存在多个目标点，为了加快规划效率，将每一目标点的初始航向角设定为当前起始点与当前目标点的直线连接的夹角，即在第 i 个起始点处初始航向角可表示为：

$$\theta_i = a \tan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \quad (5-3)$$

其中, x_{i+1} , y_{i+1} 表示当前目标点的 x, y 值, x_i , y_i 表示当前起始点的 x, y 值。

5.3.2 改进的速度模型

在速度空间中, 存在无数可能的 (v, ω) 速度组合。但在实际应用中, 移动机器人的速度范围受到其物理能力和环境限制的约束。通常, 这些约束表现为机器人的最大和最小速度限制。这些速度限制确保机器人在安全和实际的参数内操作。

$$V_m = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (5-4)$$

移动机器人在动态窗口法中的移动时间间隔内, 必须考虑到电机的加速度和减速度约束。这些约束影响机器人在给定时间内能够达到的速度范围设定为:

$$V_d = \{(v, \omega) | v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t], \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\} \quad (5-5)$$

其中, v_c 为当前线速度, ω_c 为当前角速度, $\dot{v}_a$ 为最大线加速度, $\dot{\omega}_a$ 为最大角加速度, $\dot{v}_b$ 为最大线减速度, $\dot{\omega}_b$ 为最大角减速度。

确保机器人在遇到随机障碍物时能够安全停下是至关重要的。机器人的制动约束表示为:

$$V_a = \{(v, \omega) | v \leq (2 \text{dist}(v, \omega) \dot{v}_b)^{1/2}, \omega \leq (2 \text{dist}(v, \omega) \dot{\omega}_b)^{1/2}\} \quad (5-6)$$

其中 $\text{dist}(v, \omega)$ 为 (v, ω) 对应轨迹上离障碍物的最近距离。

上述三种约束共同限定了机器人的运动速度范围。如果我们用 V 来表示速度集合, 那么机器人的运动速度范围可以表示为:

$$V = V_m \cap V_d \cap V_a$$

在本章改进算法中存在多个目标点, 若仍采用原速度模型, 则机器人在运行过程中可能会出现运行过程不平稳、规划时间较长, 效率低等问题, 为解决这一问题, 应将每一目标点联系起来, 建立新的速度模型。

在机器人路径规划中, 如果临时目标是全局目标或与机器人有一定距离, 速度模型按原约束运行。若非全局目标且距离小于安全距离, 线速度不变, 角

速度按约束调整。

则此时机器人最大、最小速度约束为：

$$V'_m = \{(v, \omega) | v = v_{t-\Delta t}, \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (5-7)$$

电机的加速度和减速度约束所产生的速度约束为：

$$V'_d = \{(v, \omega) | v = v_{t-\Delta t}, \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\} \quad (5-8)$$

制动约束表示为：

$$V'_a = \{(v, \omega) | v = v_{t-\Delta t}, \omega \leq (2dist(v, \omega)\dot{\omega}_b)^{1/2}\} \quad (5-9)$$

则此时机器人运动速度范围为：

$$\begin{cases} V' = V'_m \cap V'_d \cap V'_a, p \neq E \cap dist(p) \leq 3 \\ V = V_m \cap V_d \cap V_a, others \end{cases} \quad (5-10)$$

其中， p 为当前目标点， E 为总目标点， $dist(p)$ 为当前位置与当前目标点之间的距离。

5.3.3 评价函数

在速度空间中，有许多可行的速度组合。为了从中挑选出最佳轨迹，需要一个有效的评价函数。本章对评价函数进行改进，本章改进的评价函数如式(5-11)所示。

$$G(v, \omega) = \sigma(\alpha head(v, \omega) + \beta dist(v, \omega) + \gamma vel(v, \omega)) \quad (5-11)$$

其中， $head(v, \omega)$ 为轨迹终点方向与当前目标点之间角度差， $vel(v, \omega)$ 为当前速度评价函数， α, β, γ 为各函数的权重系数。

5.4 算法运行步骤

融合算法流程图如图 5-1 所示。

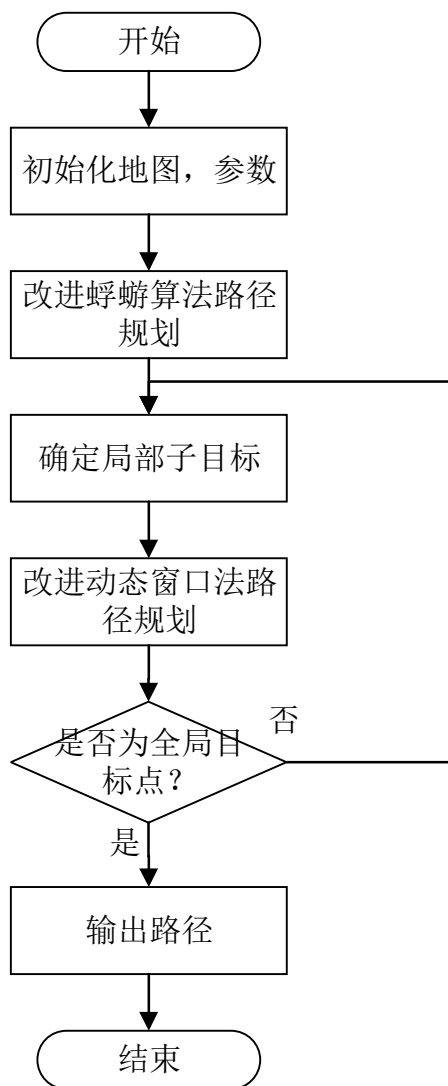


图 5-1 融合算法流程图

5.5 测试与仿真

为验证本改进算法的有效性, 本章在静态环境和动态环境下运用 python 语言编程进行路径规划仿真验证。

5.5.1 改进蜚螭算法仿真验证

为了验证本章改进算法的有效性和稳定性, 采用 100×100 的栅格环境模型, 在此模型下对该算法进行仿真, 并与传统蜚螭算法和线性自适应惯性权重因子蜚螭算法进行比较, 传统蜚螭算法中惯性权重及正吸引系数设定为: 惯性权重 $g=0.8$, 正吸引系数 $a1=1.0$, $a2=1.5$, $a3=1.5$, 雄雌蜚螭种群规模均设 10, 随机游走系数 $fl=0.5$, 速度最大值 $Velmax=20$, 速度最小值 $Velmin=-20$, 最大迭代次数 $n=100$, 舞蹈系数 $dance=0.5$, 线性自适应惯性权重因子蜚螭算法惯性权重设置

为线性下降，其余参数同上，本章改进蜉蝣算法中 g 、 $a1$ 和 $a2$ 依强化学习设置为自适应参数，学习率 $\alpha=0.3$ ，折扣因子 $\beta=0.95$ ，其余参数同上。为确保实验结果的可靠性，对三种算法各自进行 20 次独立实验，并计算出路径长度的方差和平均值，可将这些统计数据作为评估算法性能的指标。其中一次实验结果如图 5-2 所示，平均值、方差数据统计结果如表 5-3 所示，每次实验所得最优解如图 5-3 所示。

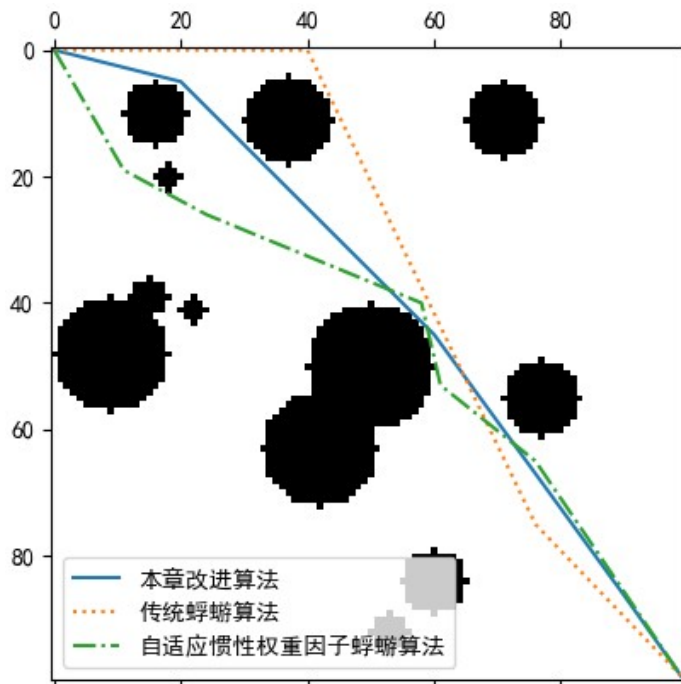


图 5-2 路径对比图

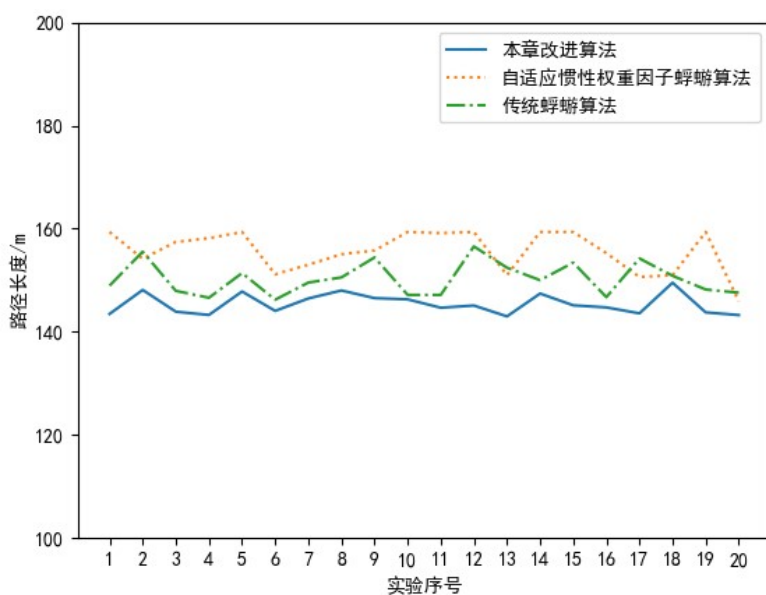


图 5-3 最优解对比

表 5-3 平均值与方差对比

	本章改进算法	蜉蝣算法	线性自适应
路径长度平均值/m	145.6862	152.5130	148.9185
方差	4.6761	16.7891	11.0460

由图 5-2，图 5-3 中可以明显看出，本章改进算法所规划路径更为平滑，且路径长度更短，方差更小，求解性能更稳定。而传统蜉蝣算法和线性自适应惯性权重因子蜉蝣算法虽然偶尔也能获得较优的路径，但常常被困在局部最优解中，算法波动大，稳定性较差。

为进一步验证本章改进蜉蝣算法在复杂环境下的表现，选取 100×100 的障碍物更多的复杂栅格环境模型，在此模型下对该算法进行仿真，并与传统蜉蝣算法和线性自适应惯性权重因子蜉蝣算法进行比较，参数设置同上。为确保实验结果的可靠性，对三种算法各自进行 20 次独立实验，并计算出路径长度的方差和平均值，可将这些统计数据作为评估算法性能的指标。其中一次实验结果如图 5-4 所示，平均值、方差数据统计结果如表 5-4 所示，每次实验所得最优解如图 5-5 所示。

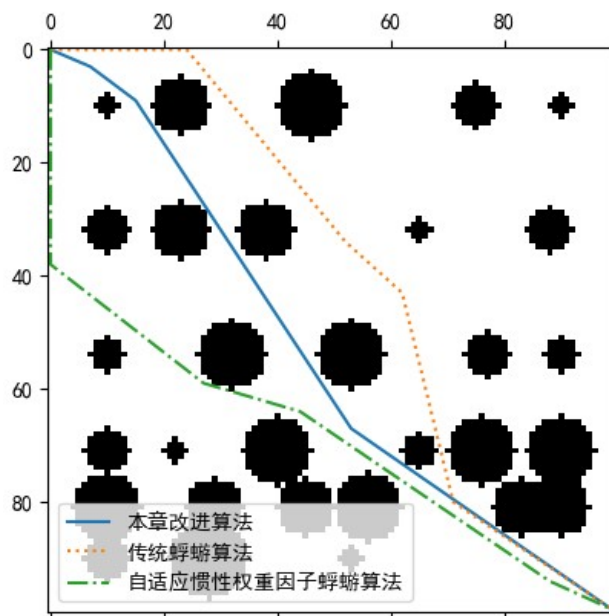


图 5-4 路径对比图

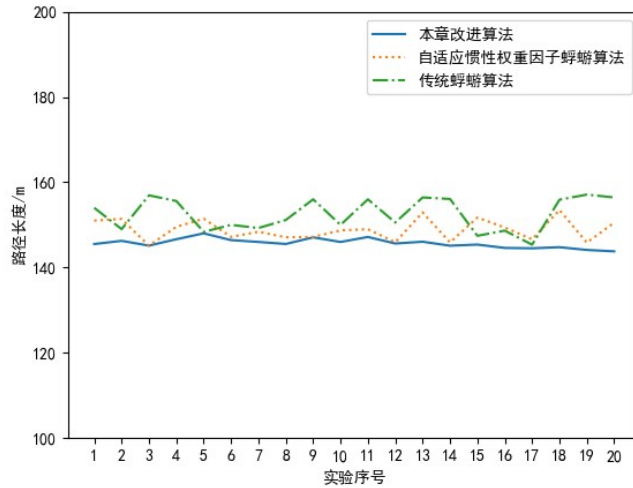


图 5-5 最优解对比

表 5-4 平均值与方差对比

	本章改进算法	蟾蜍算法	线性自适应
路径长度平均值/m	145.40628	155.6552	150.2747
方差	8.6149	17.6595	14.0461

由图 5-4，图 5-5 中可以明显看出，本章改进算法在复杂环境中得到了一条短且较平滑的路径。由表 5-4 可以看出，本章改进算法所得路径长度更短，方差更小，求解性能更稳定。而传统蟾蜍算法和线性自适应惯性权重因子蟾蜍算法虽然有时也能获得较优的路径，但算法波动大，稳定性较差，所得路径长度平均值与方差都逊色于本章改进蟾蜍算法。由此可得，本章改进算法鲁棒性好，可以有效地解决移动机器人的路径规划问题。

5.5.2 动态路径规划仿真

为了验证融合算法在动态环境中的有效性，增加了动态规划实验。在实验中，随机栅格图环境增加了 3 个沿 x 轴往复运动的动态障碍物（红色圆形），同时保留了静态障碍物（黑色圆形）。通过仿真，将融合算法的表现与传统动态窗口法进行了对比分析。本章改进算法参数设置为：移动机器人的最大线速度 2m/s、最小线速度 0m/s、最大角速度 50/s、最大线加速度 0.5、最大角加速度 30、线速度分辨 0.1s,向前预估时间 4s, $\alpha = 0.05, \beta = 0.5, \gamma = 0.1$ ，动态窗口法参数同上。本章改进算法避障过程如图 5-6 所示。

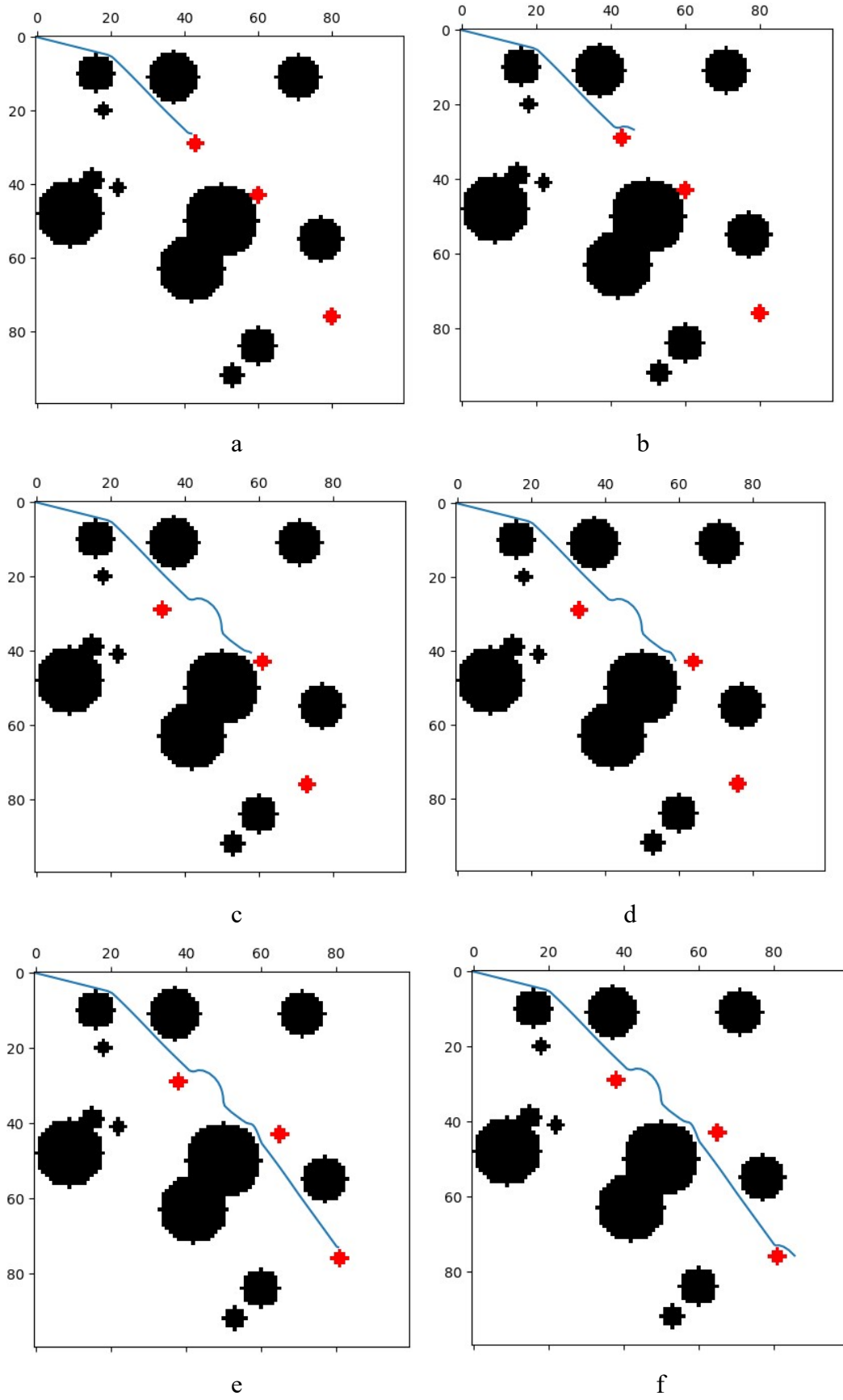


图 5-6 本章改进算法避障过程

由图 5-6 可知，文中算法能够在动态环境下有效完成路径规划任务，且能得到一条较优路径。最终所规划路径如图 5-7 所示。

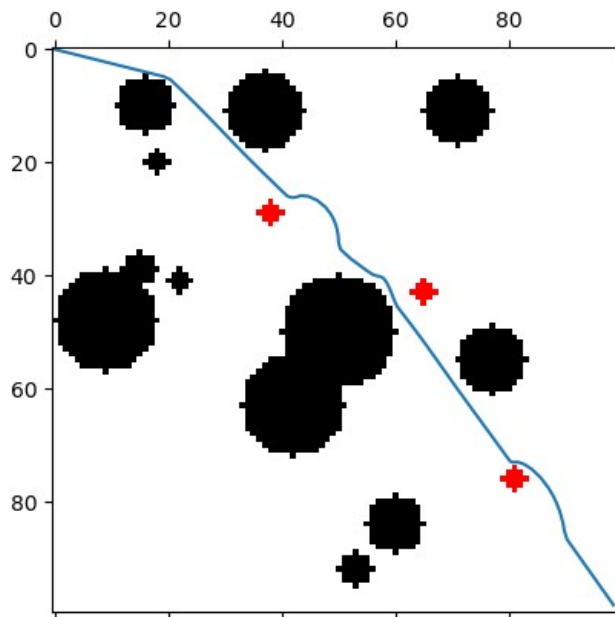


图 5-7 最终规划路径结果

传统动态窗口法最终规划路径如图 5-8 所示。

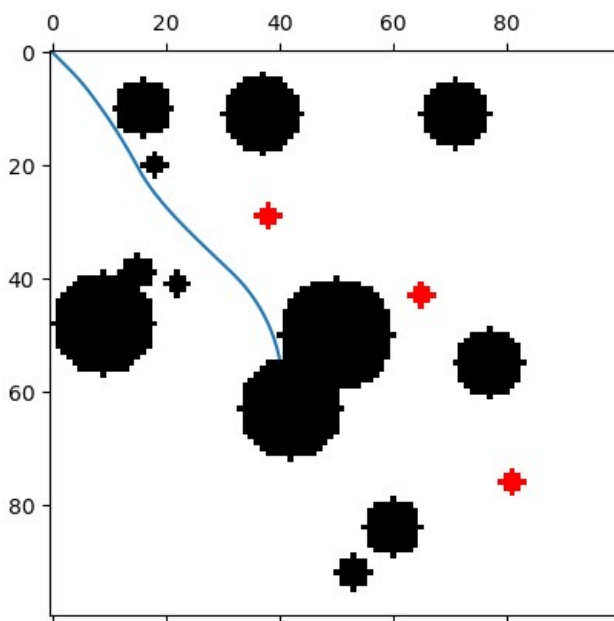


图 5-8 最终规划路径

由图 5-7，图 5-8 可知，传统动态窗口法因缺乏全局信息指引，最终无法完成路径规划任务。而本章算法可在全局最优路径信息的指引下有效完成路径规划任务，有效躲避动态障碍物，规划出一条短且安全的路径。

第 6 章 基于 ROS 平台的仿真实验

为了进一步的验证本文融合深度强化学习的改进蜉蝣算法的有效性，在本章进行基于 ROS 平台的仿真实验与现实实验。首先，基于 ROS 平台构建了仿真环境，并在 Gazebo 中建立了仿真实验环境和机器人模型。在简单和复杂两种不同环境下进行了仿真测试，验证了本文算法。最后在现实环境下进行了路径规划实验，进一步验证了算法的有效性。

6.1 ROS 平台及仿真软件简介

6.1.1 ROS 简介

Robot Operating System (ROS) 是由 Willow Garage 公司在 2010 年发布的一个开源的机器人软件平台。它为不同类型的计算机集群提供了类似操作系统的功能，非常适合于机器人的大规模研发^[66]。作为一个元操作系统，ROS 提供了一系列工具、库和通信框架，以帮助开发者构建复杂且可靠的机器人应用程序^[67]。它的基本目标是提供一个协作环境，为每个特定任务组合最合适的部分。通过这种方式，特定领域（例如计算机视觉或运动规划）的专家可以开发特定功能并将它们作为扩展 ROS 的软件库分发，同时使用其他人实现的功能来完成他们的设计^[68]。

ROS 不是一个真正的操作系统，而是一组运行在现有操作系统上的程序和库。它整合了机器人的资源框架和接口，使代码模块化，大大地提高了资源的共享和复用，重要的是这些全部都是开源的。ROS 采用了一种分布式处理框架 (Nodes)，由大量的节点组成，每个节点都可以与其他节点互相通讯，具有很好的灵活性。

6.1.2 Gazebo 简介

Gazebo 是由 ROS 提供的三维环境模拟软件，可以搭建移动机器人的实验环境，模拟重力摩擦力等参数，被广泛地应用于机器人开发领域^[69]。Gazebo 功能强大，可能够提供高逼真度的机器人模型和仿真环境，包含多个物理引擎，可模拟真实场景来训练人工智能系统，能够完成 SLAM 建图、路径导航等任务。Gazebo 的仿真环境如图 6-1 所示。

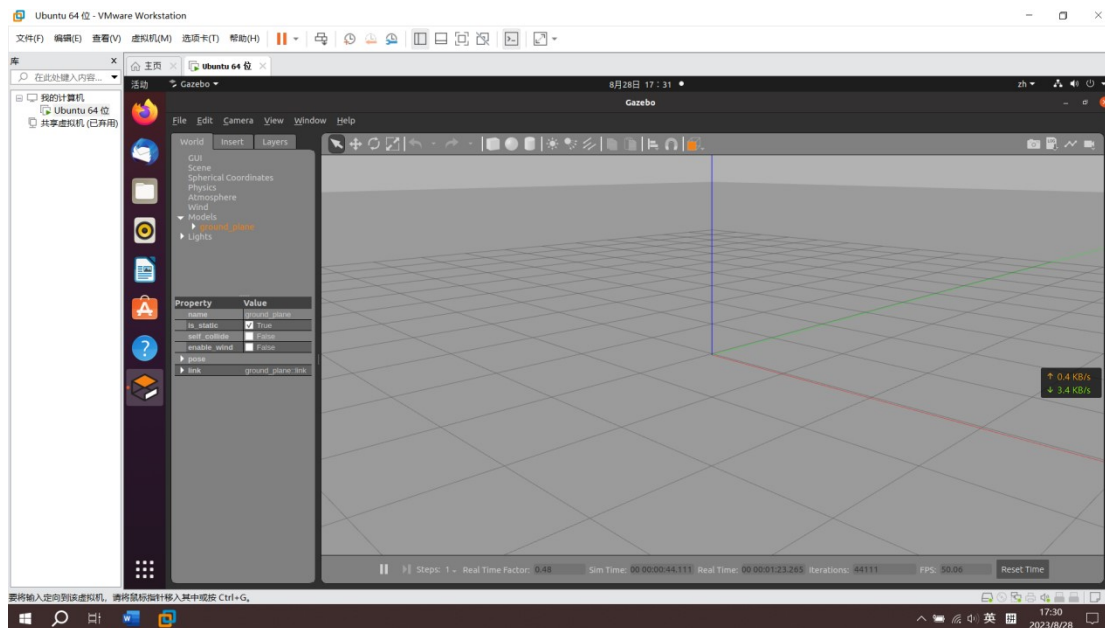


图 6-1 Gazebo 仿真环境

6.1.3 Rviz 简介

Rviz 是基于 ROS 的可视化平台，能够向机器人发布控制指令，将传感器采集到的环境信息数据实时显示^[70]。在机器人运动的过程中，能够将移动机器人规划的路径数据以曲线的方式显示出来^[71]。其界面如图 6-2 所示。

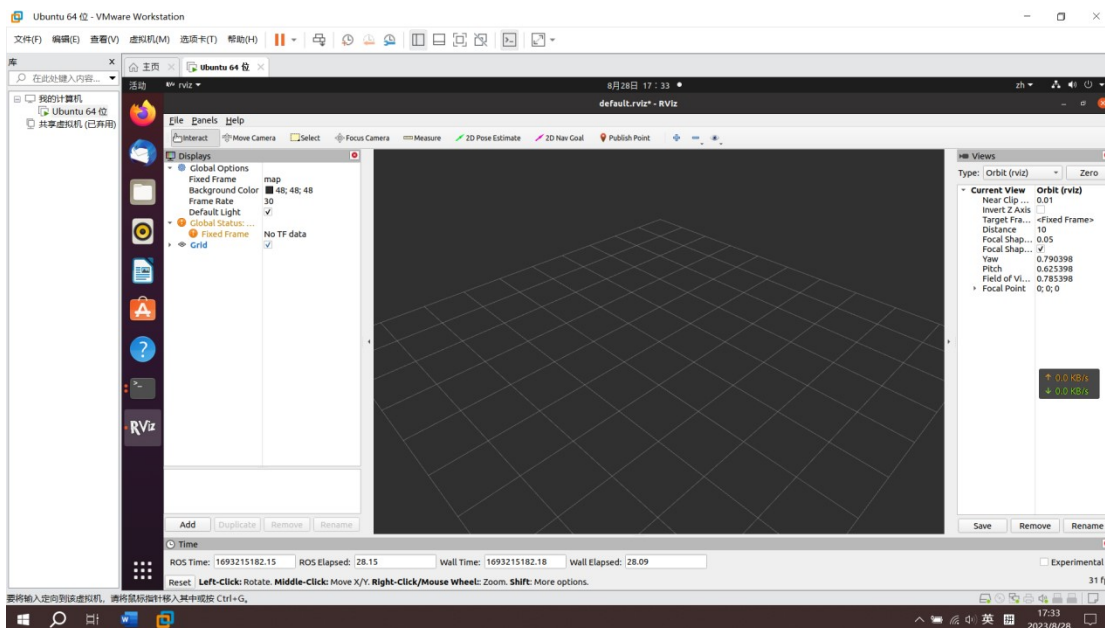


图 6-2 Rviz 可视化平台界面

6.2 环境建模与仿真实验

6.2.1 移动机器人建模与环境建模

本文使用的移动机器人为差速移动机器人，该机器人由其底部后方两个驱

动轮的转动为其提供动力，两个轮子分别由两个独立的电机控制，前方的随动轮起支撑作用并不推动其运动。

在 ROS 中，有众多软件包可用于建模各种机器人。统一机器人描述格式（URDF）是描述机器人模型的标准方法，它使用 XML 格式定义机器人的运动学、动力学、可视化和碰撞模型。URDF 通过连接件、部件和关节的组合来描述机器人的结构，使得模型创建和管理更加高效和标准化。

在机器人的基本连接件和关节描述完成后，还需要为机器人添加一些额外插件来实现所需功能，如悬崖传感器插件，用来可以探测悬崖，帮助机器人避开台阶以及防止机器人坠落；接触传感器插件，它将在机器人与其他物体发生碰撞的时候触发，用来检测碰撞；陀螺仪插件，用来测量机器人的角速度，通过角速度，可以计算出机器人的方向，机器人的方向在机器人驱动控制器中可用于计算机器人的位姿；差分驱动插件，该插件模拟机器人中的差分驱动行为，当它接收到以 ROS Twist 消息（`geometry_msgs/Twist`）形式出现的命令速度（线速度和角速度）时，它将移动机器人模型，该插件还计算机器人的里程计数，给出机器人的本地位置；深度摄像头插件，可以模拟拥有不同特征的各种深度传感器。

至此，所需的机器人模型已搭载完成，其外观如图 6-3 所示。

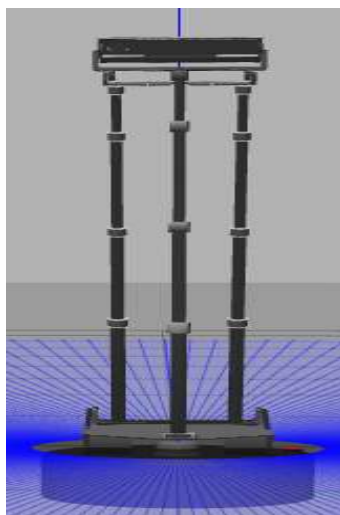


图 6-3 移动机器人外观

接下来进行仿真实验环境的构造。根据本文中移动机器人路径规划及避障的需要，本文将四周使用障碍物进行封闭，并在其中设置了一定的静态障碍物及动态障碍物。共构造有简单与复杂两种仿真环境，环境及其 SLAM 建图所得

栅格图如图 6-4，6-5 所示。

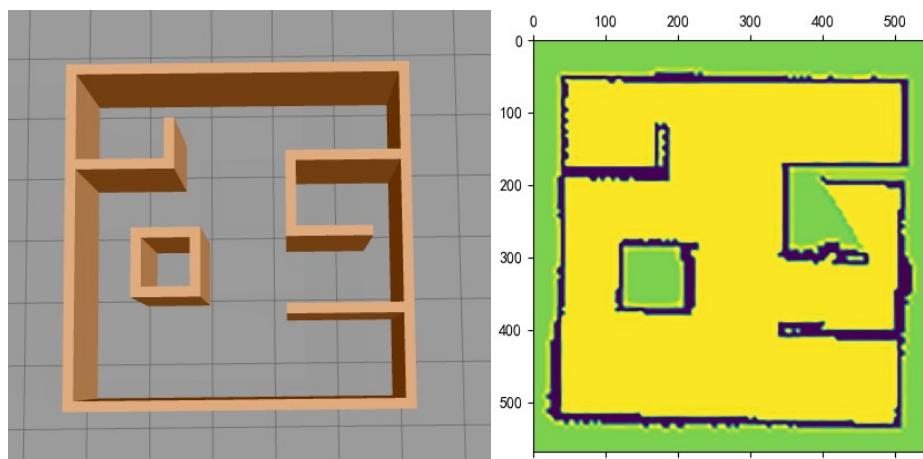


图 6-4 简单环境

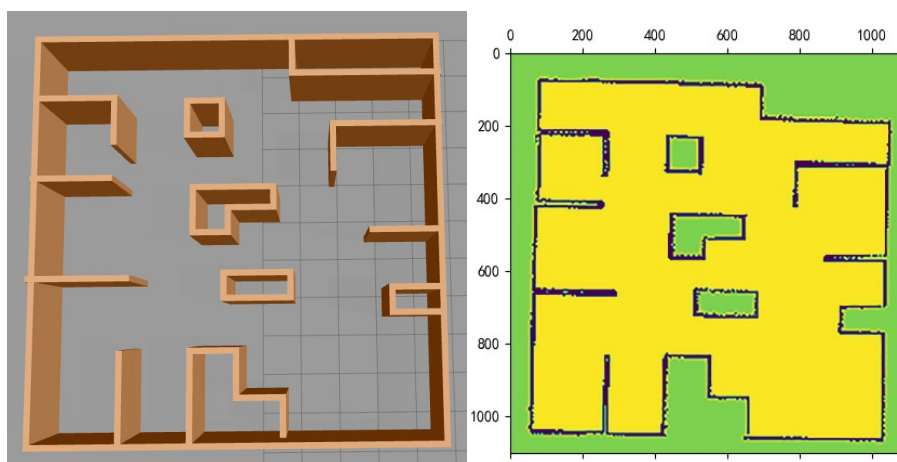


图 6-5 复杂环境

6.2.2 仿真实验与结果分析

本次仿真实验参数设置为：雄雌蜉蝣种群规模均设 10，随机游走系数 $fl=0.5$ ，速度最大值 $Velmax=20$ ，速度最小值 $Velmin=-20$ ，最大迭代次数 $n=100$ ，舞蹈系数 $dance=0.5$ ，学习率 $\alpha=0.3$ ，折扣因子 $\beta=0.95$ 。

为了验证改进算法的有效性和稳定性，采用图 6-4 中简单栅格环境模型进行仿真实验，结果如图 6-6 所示。

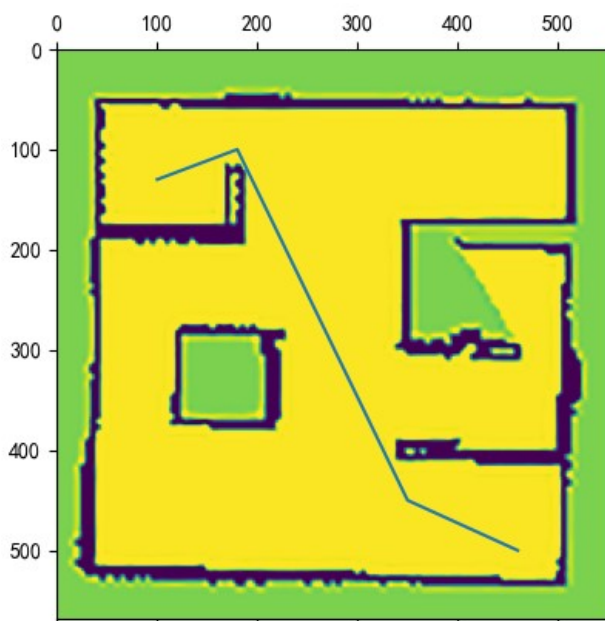


图 6-6 简单环境结果

仿真实验结果表明，本文提出的算法可以在简单环境中成功规划出一条较优路线，有效完成了路径规划任务。为了全面评估算法的性能，在更复杂的环境中进行仿真实验，并放置 2 个移动障碍物，仿真结果如图 6-7 所示。

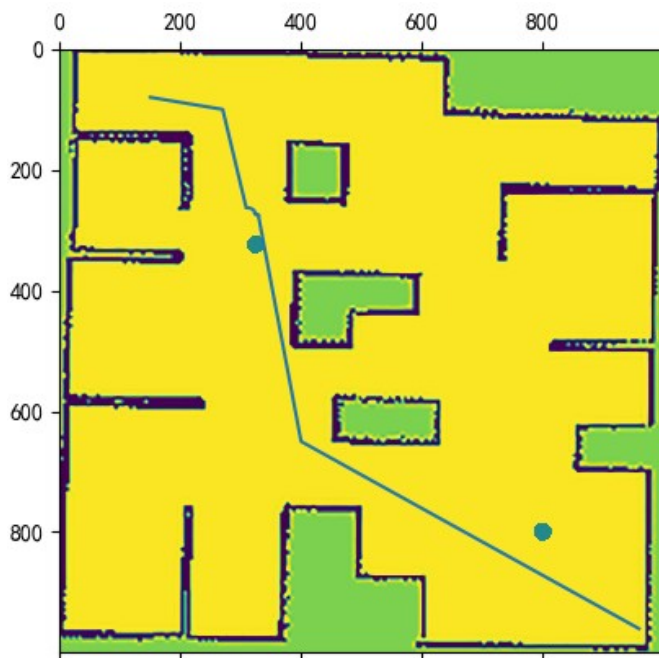


图 6-7 复杂环境结果

仿真结果表明，本文算法在复杂动态环境中也可以有效完成路径规划任务。综上，本文改进深度强化学习和蜚蜥算法融合算法可以有效完成路径规划任务。

6.2 实物实验验证

6.2.1 机器人平台搭建

在仿真实验对本文算法验证后，将通过实验以测试本文算法在现实中的有效性。

运动控制设备为搭载了 Jutson nano 主板的差速式移动机器人软硬件平台，中控板为嵌入式开发板 stm32，感知设备为奥比中光深度相机与镭神 N10P 雷达。

6.2.2 现实实验验证

控制机器人在环境中移动，进行地图的构建，现实环境及所构建地图如图 6-8 所示。

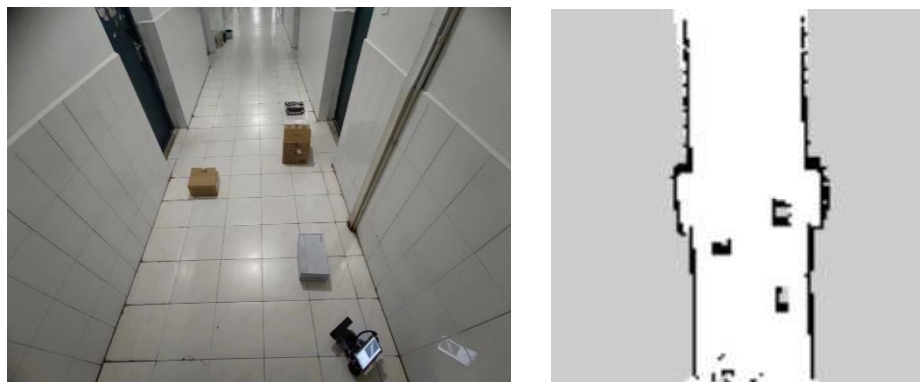


图 6-8 环境地图

设置起点与终点后，移动机器人规划路径如图 6-9 所示。

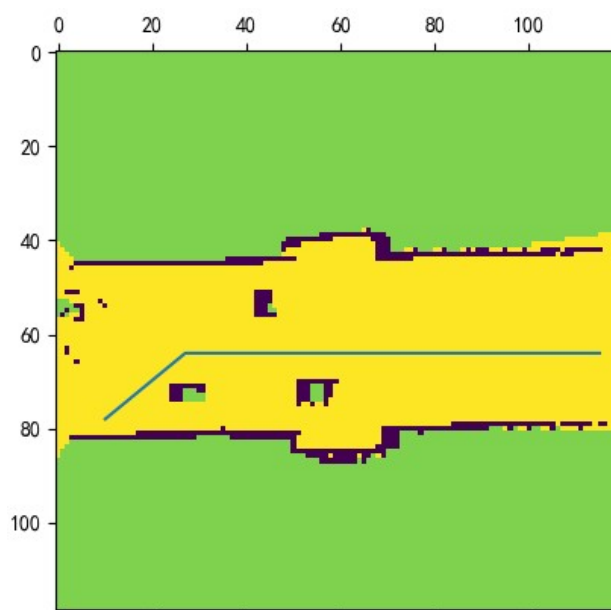


图 6-9 全局路径

在该环境下放置一个动态障碍物，再次进行路径规划实验，移动机器人的起始状态如图 6-10 所示。



图 6-10 初始环境

避障过程如图 6-11 所示。



规避过程 1



规避过程 2

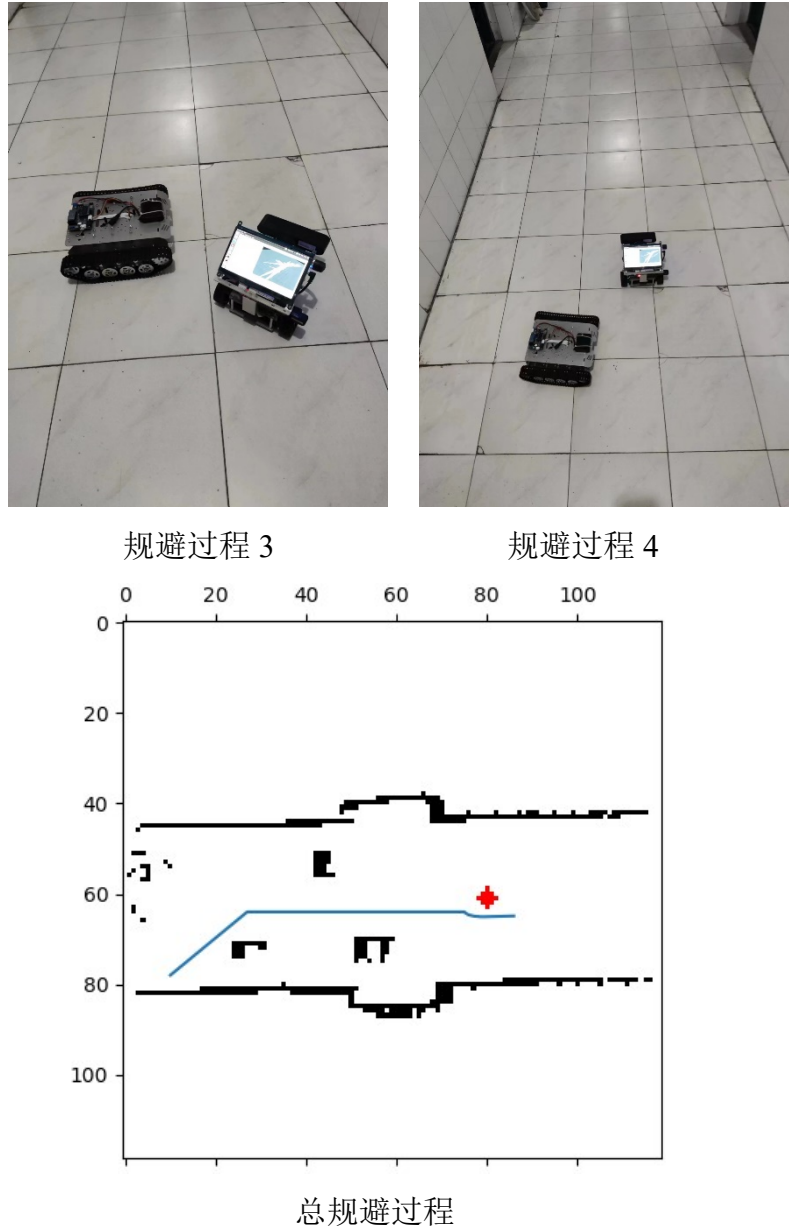


图 6-11 规避过程

移动机器人最终行进路线如图 6-12 所示。

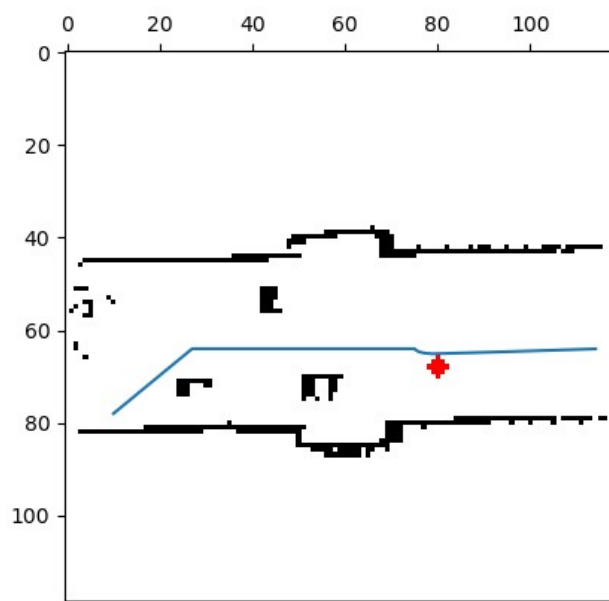


图 6-12 最终路径

由图 6-11, 6-12 可得, 改进后的算法能够使机器人在检测到障碍物时及时调整路径, 安全绕行障碍物, 并成功返回全局最优路径上, 最终抵达目标点。这证明了本文改进深度强化学习和蜚螭算法融合算法可以有效在现实中完成路径规划任务。

第 7 章 总结与展望

7.1 工作总结

本文的研究主体主要分为三个部分，包括改进深度强化学习算法，改进蜉蝣算法，与深度强化学习、动态窗口法结合蜉蝣算法以及这些算法在路径规划中的应用。本文的主要工作包括以下几个方面：

(1) 对路径规划技术的相关理论以及方法进行总结。首先对路径规划问题进行简单的概述，包括常见的几种环境建模方法、常用的路径规划算法。并对下文要研究的蜉蝣算法以及深度强化学习算法进行了叙述。

(2) 针对传统的 DQN 算法在 Q 表的初始化时通常将其设为均等或完全随机的，这使得某种最优动作因初始化时概率设置过低，使得它被访问的概率很低，尤其在大规模未知环境中，可能导致大量无效迭代。本文提出一种基于人工势场的探索策略。而且传统 DQN 算法在处理路径规划问题时探索效率低下。本文提出一种自适应线性贪婪策略，提高了机器人的探索能力。仿真实验表明，本章改进算法相较于传统算法路径规划长度明显更短。

(3) 针对传统蜉蝣算法因各参数都为固定值，无法平衡全局搜索与局部搜索，易导致算法搜索能力差，且易陷入局部最优的问题。本文提出一种自适应参数调整策略，平衡了算法局部搜索能力与全局搜索能力，并且引入莱维飞行策略，进行最优扰动，帮助其跳出局部最优，提升了解的多样性。针对路径规划问题中常常只考虑路径的长度而忽略了转弯对能耗的影响，本文将转弯次数纳入适应度函数计算中，在能量消耗上更贴近现实。仿真实验表明，本章改进算法相较于传统算法在处理路径规划问题时表现更优。

(4) 针对动态路径规划问题，提出一种基于深度强化学习、蜉蝣算法与动态窗口法的融合算法，并引入记忆机制，去除大量无用迭代，加快了算法的效率与迭代速度。仿真实验表明，本章改进算法能有效完成动态避障功能。最后，通过搭建基于 ROS 系统的三轮全向移动机器人，在实际环境下对本文所提算法进行实验测试，验证了本文算法的有效性和可行性。

7.2 研究展望

本文通过上述的研究工作，对蜉蝣算法、深度强化学习算法在路径规划任

务上存在的一些缺陷进行了改进，取得了一定的成果。但依旧存在一些缺陷值得去解决，

主要包括以下几点：

(1) 在本文中，环境建模所用栅格图法仅包含了通行与否信息，与现实环境中复杂多样环境信息不同，未涉及如环境坡度等问题，考虑在后续研究中对更多环境信息进行模拟。

(2) 在实际生产与应用时，单一机器人路径规划不足以满足需求，可以考虑在后续研究中进行多机器人的路径规划。

参考文献

- [1]张松灿.基于蚁群算法的移动机器人路径规划研究[D].洛阳:河南科技大学,2021.
- [2]Ke X L,Yu Y H, Li K S.Review on robot-assisted polishing: Status and future trends[J]. Robotics and Computer-Integrated Manufacturing, 2023, 80:102482.
- [3]陈英.中国电子学会.中国机器人产业发展报告(2022年)[R].世界机器人大会,北京,2022,8.
- [4]Liang X X, Liu C Y, Song X L, et al. Research on improved artificial potential field approach in local path planning for mobile robot[J]. Computer simulation, 2018, 35(4): 291-361.
- [5]中国社会科学院工业经济研究所课题组,史丹."十四五"时期中国工业发展战略研究[J].中国工业经济, 2020(2).
- [6]Zhang D, Luo R, Yin Y B, et al. Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A*[J]. Nuclear Engineering and Technology, 2023, 55: 1738-5733,
- [7]Zhang L, Mou J M, Chen P F, et al. Path planning for autonomous ships: a hybrid approach based on improved APF and modified VO methods[J]. Journal of Marine Science and Engineering. 2021; 9(7):761.
- [8]Wen J B, Yang J C, Wang T Y. Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm[J]. IEEE Transactions on Vehicular Technology., 2021,70: 8529-8544.
- [9]Liu L X, Wang X, Yang X. Path planning techniques for mobile robots: Review and prospect[J]. Expert Systems with Applications. 2023,227:120254.
- [10]槐创锋,郭龙,贾雪艳,等.改进 A*算法与动态窗口法的机器人动态路径规划[J].计算机工程与应用,2021, 57(8):244-248.
- [11]顾子倡,刘宇,孙文邦,等.基于 RRT 的无人机动态航路规划算法[J].计算机科学,2023,50(S1):65-69.
- [12]王彬,聂建军,李海洋,等.优化 A~*与动态窗口法的移动机器人路径规划[J].计

计算机集成制造系统,2024,30(04):1353-1363.

[13]徐小强,王明勇,冒燕.基于改进人工势场法的移动机器人路径规划[J].计算机应用,2020,40(12):3508-3512.

[14] Min H S, Lin Y H, Wang S J, et al. Path planning of mobile robot by mixing experience with modified artificial potential field method[J]. Advances in Mechanical Engineering, 2015,12:1-17.

[15]赵亮,李春轩,张玮奇,等.基于融合引-斥力与动态窗口法的机器人静动态局部路径规划方法优化[J].信息与控制,2023:1-13.

[16] Ibraheem I K, Ajeil F H. Path planning of an autonomous mobile robot using swarm based optimization techniques[J]. Al-Khwarizmi Engineering Journal, 2016, 12:12-25.

[17]马飞宇,瞿中.基于异构双种群全局视野蚁群算法的移动机器人路径规划研究[J].计算机应用研究,2022,39(06):1705-1709.

[18] Li E C, Qi K K. Ant colony algorithm for path planning based on grid feature point extraction[J]. Journal of Shanghai Jiaotong University (Science), 2023, 28(01): 86-99.

[19]毛文平,李帅永,谢现乐,等.基于自适应机制改进蚁群算法的移动机器人全局路径规划[J].控制与决策,2023,38(09):2520-2528.

[20] Davoodi M, Panahi F, Mohades A, et al. Multi-objective path planning in discrete space[J]. Applied Soft Computing, 2013, 13:709-720.

[21]牛秦玉,李博.基于模拟退火遗传算法的全向 AGV 路径规划[J/OL].计算机集成制造系统:1-21[2023-12-04].

[22]魏彤,龙琛.基于改进遗传算法的移动机器人路径规划[J].北京航空航天大学学报,2020,46(4):703-711.

[23] Song B, Wang Z, Zou L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve[J]. Applied Soft Computing, 2021, 100:1-11.

[24]陈秋莲,郑以君,蒋环宇,等.基于神经网络改进粒子群算法的动态路径规划[J].华中科技大学学报(自然科学版),2021,49(02):51-55.

- [25] 贾会群,魏仲慧,何昕,等.基于改进粒子群算法的路径规划[J].农业机械学报,2018,49(12):371-377.
- [26] 吴大飞,杨光永,樊康生,等.多策略融合的改进粒子群优化算法[J].计算机应用研究,2022,39(11):3358-3364.
- [27] Guo J, Gao Y, Cui G. The path planning for mobile robot based on bat algorithm[J]. International Journal of Control, Automation and Systems, 2015 ,9 : 50-61.
- [28] Ajeil F H, Ibraheem I, Humaidi A, et al. A novel path planning algorithm for mobile robot in dynamic environments using modified bat swarm optimization[J]. Engineering, 2021 ,2021: 37-48.
- [29] 吕莉,潘宁康,肖人彬,等.求解带约束优化问题的混合式多策略萤火虫算法[J/OL].控制与决策:1-9[2023-12-04].
- [30] 李小牧,田妙苗,马广彬,等.基于改进布谷鸟搜索的卫星成像规划方法[J].中国科学院大学学报,2023,40(01):101-108.
- [31] 周新宇,刘颖,吴艳林,等.基于多元信息引导的人工蜂群算法[J/OL].电子学报:1-14[2023-07-18].
- [32] 徐淑静.基于 FPGA 的并行模糊控制算法的研究与实现[D].中国科学院大学(中国科学院光电技术研究所),2022.
- [33] 周惠珂.低空无人机路径规划算法研究[D].西安邮电大学,2023.
- [34] 顾子倡,刘宇,孙文邦,等.基于 RRT 的无人机动态航路规划算法[J].计算机科学,2023,50(S1):65-69.
- [35] 李薪颖,单梁,常路,等.复杂环境下基于多目标粒子群的 DWA 路径规划算法[J].国防科技大学学报,2022,44(04):52-59.
- [36] 褚宏悦,易军凯.无人机安全路径规划的混沌粒子群优化研究[J/OL].控制工程:1-8[2023-12-04].
- [37] 杨博,刘树东,鲁维佳,等.改进遗传算法在机器人路径规划中的应用[J].现代制造工程,2022,06:9-16.
- [38] 毛文平,李帅永,谢现乐,等.基于自适应机制改进蚁群算法的移动机器人全局路径规划[J].控制与决策,2023,38(09):2520-2528.

- [39] Konstantinos Z, Stelios T. A mayfly optimization algorithm[J]. Computers & Industrial Engineering,2020,145:106559.
- [40] Adnan R M, Kisi O, Mostafa R R, et al, The potential of a novel support vector machine trained with modified mayfly optimization algorithm for streamflow prediction[J], Hydrological Sciences Journal, 2022, 67(2):161-174.
- [41] 毛清华,王迎港,牛晓辉.基于反吸引速度更新机制的改进蜉蝣算法[J/OL].北京航空航天大学学报:1-16[2024-02-04].
- [42] Gupta J, Nijhawan P, Ganguli S. Parameter estimation of fuel cell using chaotic mayflies optimization algorithm[J]. Advanced Theory and Simulations, 2021, 4(12):2100183.
- [43] 蒋宇飞,许贤泽,徐逢秋,等.多策略融合改进的自适应蜉蝣算法[J].北京航空航天大学学报,2024,50(04):1416-1426.
- [44] 王义,张达敏,邹诚诚.增强全局搜索和自适应蜉蝣算法[J].哈尔滨工业大学学报,2022,54(11):137-150.
- [45] 熊俊涛,李中行,陈淑绵,等.基于深度强化学习的虚拟机器人采摘路径避障规划[J].农业机械学报,2020,51(S2):1-10.
- [46] 成怡,郝密密.改进深度强化学习的室内移动机器人路径规划[J].计算机工程与应用,2021,57(21):256-262.
- [47] 朱少凯,孟庆浩,金晟,等.基于深度强化学习的室内视觉局部路径规划[J].智能系统学报,2022,17(05):908-918.
- [48] Han S, Wang L, Wang Y T, et al. An efficient motion planning based on grid map: predicted trajectory approach with global path guiding[J]. Ocean Engineering, 2021,238:109696.
- [49] 张会丽.地面自主移动机器人路径规划算法研究[D].浙江大学,2013.
- [50] Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
- [51] Gu J X, Wang Z H, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77:354-377.
- [52] 郭心德. 基于深度强化学习的 AGV 路径规划[D].广东工业大学,2022.

- [53]Zavlangas P G, Tzafestas S G. Industrial robot navigation and obstacle avoidance employing fuzzy logic[J]. Journal of Intelligent and Robotic Systems, 2000, 27(1-2): 85-97.
- [54] Jiang W Y, Liu Y Q, Fang G H, et al, Research on short-term optimal scheduling of hydro-wind-solar multi-energy power system based on deep reinforcement learning[J], Journal of Cleaner Production, 2023 ,385:135704.
- [55]崔浩岩,张震,赵德京,等.一种基于一致性的多智能体 Q 学习算法[J/OL].控制工程:1-11[2023-09-04].
- [56]李贤. 基于强化学习的移动机器人路径规划研究[D]. 哈尔滨: 哈尔滨工程大学, 2019.
- [57]孟祺. 室内环境下基于深度强化学习的路径规划方法研究[D].天津工业大学,2022.
- [58] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning, 2015, 518: 529–533.
- [59]张荣霞,武长旭,孙同超,等.深度强化学习及在路径规划中的研究进展[J].计算机工程与应用,2021,57(19):44-56.
- [60]徐晓苏,袁杰.基于改进强化学习的移动机器人路径规划方法[J].中国惯性技术学报,2019,27(03):314-320.
- [61]邓修朋,崔建明,李敏,等.深度强化学习在机器人路径规划中的应用[J].电子测量技术,2023,46(06):1-8.
- [62] Essam H H, Mohammed R S, Fatma A H, et al. Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems[J]. Engineering Applications of Artificial Intelligence, 2020, 94:103731.
- [63]邹阿威,王雷,李伟民,等.改进蜚螭算法的移动机器人路径规划研究[J/OL].机械科学与技术:1-8[2024-04-07].
- [64] 王豪,赵学军,袁修久.基于改进自适应遗传算法的机器人路径规划[J].电光与控制,2022,29(5):72-76.
- [65]汤云峰.改进智能算法在移动机器人路径规划中的研究[D].南京邮电大学,2021.

- [66] 王颖,王冰青,关永,等.面向 ROS 的差分模糊测试方法[J].软件学报,2021,32(06):1867-1881.
- [67] Gayathri R, Uma V, Bettina O B,Unified robot task and motion planning with extended planner using ROS simulator[J].Journal of King Saud University - Computer and Information Sciences,2022,34(9), 7468-7481.
- [68] Enrique M M, Manuel M, Adrián R, et al. Verification of the ROS NavFn planner using executable specification languages[J], Journal of Logical and Algebraic Methods in Programming,2023,132:100860
- [69] 王翠芝.多传感器融合技术在移动机器人中的应用[D].北方工业大学.2010.186-190.
- [70] 朱伟达.基于改进型人工势场法的车辆避障路径规划研究[D].江苏大学,2017.
- [71] Qi B K, Li M Q, Yang Y, et al. Research on UAV path planning obstacle avoidance algorithm based on improved artificial potential field method[J]. Journal of Physics: Conference Series,2021,1948(1): 14-16.

硕士学习期间获得的科研成果

一、发表学术论文情况

- [1]Zou A W, Wang L, Li W M, et al. Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach[J]. The Journal of Supercomputing, 2023,79:8340–8367. (SCI, EI 收录)
- [2]Li W M, Wang L, Zou A W, et al. Path Planning for UAV Based on Improved PRM[J].Energies,2020,15(19):7267. (SCI 收录)
- [3]邹阿威,王雷,李伟民等.改进蜉蝣算法的移动机器人路径规划研究[J/OL].机械科学与技术:1-8[2023-12-08]. (CSCD 核心库收录)
- [4]李伟民,王雷,邹阿威等.PRM 蚁群融合算法的 AGV 路径规划研究[J].井冈山大学学报(自然科学版),2023,44(02):65-70.

二、专利授权情况

- [1]邹阿威,王雷,蔡劲草等.一种基于动态自适应参数调整的蜉蝣算法的移动机器人路径规划方法[P]. 安徽省: CN114995390A,2022-09-02. (发明实审)
- [2]李伟民,王雷,蔡劲草,邹阿威等.一种改进 A*算法的 4 阶 B 样条曲线路径规划方法[P]. 安徽省: CN114995391A,2022-09-02. (发明实审)
- [3]蔡劲草,王雷,王安恒,邹阿威等.一种解决分布式混合流水车间总延迟时间优化的强化学习蛙跳算法[P]. 安徽省: CN114912676A,2022-08-16. (发明实审)
- [4]蔡劲草,王雷,王安恒,邹阿威等.一种解决分布式混合流水车间的双目标调度优化方法[P]. 安徽省: CN114819558A,2022-07-29. (发明实审)
- [5]王雷,李东东,蔡劲草,邹阿威等.一种基于动态终点策略的 IA-Star 算法的机器人路径规划方法[P]. 安徽省: CN115016461A,2022-09-06. (发明实审)
- [6]朱廷焕,徐刚,石建德,朱兴博,王雷,邹阿威等.一种空气弹簧[P]. 安徽省: CN219013249U,2023-05-12. (实用新型授权)

尚德敏学 唯实惟新

