计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(7):37−47 [doi: 10.15888/j.cnki.csa.009939] [CSTR: 32024.14.csa.009939]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进深度 Q 网络的移动机器人路径规划算法①

谢　天[1], 周　毅[1], 邱宇峰[2]

[1](武汉科技大学 人工智能与自动化学院, 武汉 430081)
[2](宝信软件武汉有限公司, 武汉 430080)
通信作者: 周　毅, E-mail: zhouyi83@wust.edu.cn

摘　要: 随着自动化技术和机器人领域的快速发展, 移动机器人路径规划的精确性要求日益提高. 针对深度强化学习在复杂环境下路径规划存在的收敛稳定性差、样本效率低及环境适应性不足等问题, 提出了一种改进的基于决斗深度双 Q 网络的路径规划算法 (R-D3QN). 通过构建双网络架构解耦动作选择与价值估计过程, 有效缓解 Q 值过估计问题, 提高收敛稳定性; 设计时序优先经验回放机制, 结合长短期记忆网络 (LSTM) 的时空特征提取能力, 改进样本利用效率; 提出基于模拟退火的多阶段探索策略, 平衡了探索与利用, 增强环境适应性. 实验结果表明, 与传统 DQN 算法相比, R-D3QN 算法在简单环境下平均奖励值提高了 9.25%, 收敛次数减少了 24.39%, 碰撞次数减少了 41.20%; 在复杂环境下, 平均奖励值提升了 12.98%, 收敛次数减少了 11.86%, 碰撞次数减少了 42.14%. 同时与其他改进的 DQN 算法对比也具有明显的优势, 验证了所提算法的有效性.

关键词: 移动机器人; 路径规划; 深度 Q 网络; 强化学习

引用格式: 谢天,周毅,邱宇峰.基于改进深度 Q 网络的移动机器人路径规划算法.计算机系统应用,2025,34(7):37–47. http://www.c-s-a.org.cn/1003-3254/9939.html

Mobile Robot Path Planning Algorithm Based on Improved Deep Q-network

XIE Tian[1], ZHOU Yi[1], QIU Yu-Feng[2]

[1](School of Artificial Intelligence and Automation, Wuhan University of Science and Technology, Wuhan 430081, China)
[2](Baosight Software (Wuhan) Co. Ltd., Wuhan 430080, China)

Abstract: The rapid advancement of automation technology and robotics requires more precision in mobile robot path planning. To address the problems of poor convergence stability, low sample efficiency, and insufficient environmental adaptability in deep reinforcement learning for path planning in complex environments, this study proposes an enhanced path planning algorithm based on dueling double deep Q-network (R-D3QN). By constructing a dual-network architecture to decouple the action selection and value estimation processes, this method effectively alleviates the Q-value over-estimation problem, thereby improving convergence stability. In addition, this method designs a temporal-prioritized experience replay mechanism combined with the spatiotemporal feature extraction capabilities of long short-term memory (LSTM) networks to improve sample utilization efficiency. Finally, this method proposes a multi-stage exploration strategy based on simulated annealing to balance exploration and exploitation, thereby enhancing environmental adaptability. Experimental results demonstrate that, compared to the traditional DQN algorithm, the R-D3QN algorithm achieves a 9.25% increase in average reward value, a 24.39% reduction in convergence iterations, and a 41.20% decrease in collision frequency in simple environments. In complex environments, it shows a 12.98% increase in average reward value, an 11.86% reduction in convergence iterations, and a 42.14% decrease in collision frequency. Furthermore, the effectiveness of the proposed algorithm is validated when compared with other enhanced DQN algorithms.

Key words: mobile robot; path planning; deep Q-network (DQN); reinforcement learning

① 基金项目: 国家自然科学基金 (62372343)
收稿时间: 2024-11-18; 修改时间: 2025-02-11; 采用时间: 2025-03-06; csa 在线出版时间: 2025-05-23
CNKI 网络首发时间: 2025-05-26

移动机器人自主导航是人工智能领域的一个重要研究方向, 而路径规划作为其核心技术之一, 在工业自动化、仓储物流、医疗服务、智能家居以及灾难救援等多个领域具有广泛的应用前景[1]. 在工业场景中, 移动机器人需要实现物料的高效精准运输; 在仓储物流领域, 机器人需在复杂的货架环境中完成货物的自动分拣与配送; 在医疗服务中, 机器人需在医院走廊和病房之间安全导航, 实现药品和医疗器械的精准配送; 在灾难救援中, 机器人则需要在非结构化环境中进行自主探索与搜救[2]. 这些应用场景对路径规划技术提出了更高的要求.

目前, 路径规划算法种类繁多, 依据其原理和应用场景, 大致可以分为 3 类: 传统算法、智能仿生算法和基于强化学习的算法. 常见的传统算法包括 Dijkstra 算法[3]、A*算法[4]和人工势场法[5]; 智能仿生算法如蚁群算法[6,7]、遗传算法[8]以及粒子群算法[9]. 这些仿生算法通过模拟自然界的生物行为, 在一定程度上提高了路径规划的适应性, 但仍面临收敛速度慢、易陷入局部最优等问题.

近年来, 深度强化学习 (deep reinforcement learning, DRL) 在路径规划领域取得了显著进展. 深度 Q 网络 (deep Q-network, DQN) 作为 DRL 的典型代表, 通过将深度神经网络与 Q-learning 相结合, 有效解决了高维状态空间下的决策问题[10]. 然而, 传统 DQN 存在明显过估计问题, 导致路径规划精度不足. 为此, van Hasselt 等人[11]提出深度双 Q 网络 (double DQN, DDQN), 通过解耦动作选择与价值评估, 显著降低了过估计的影响. 随后, Wang 等人[12]引入决斗网络结构 (dueling network), 将状态价值函数和优势函数分离, 进一步提升了算法的稳定性和泛化能力.

尽管这些改进在一定程度上提升了 DQN 的性能, 但在实际应用中仍面临诸多挑战, 传统 DQN 及其改进版本难以快速适应环境变化, 导致路径规划失败率较高. 针对这些问题, 国内外学者进行了深入研究. 其次, 在处理复杂障碍物时, 现有算法的路径平滑度不足, 影响了机器人的运动稳定性. Yang 等人[13]提出了一种基于深度双 Q 网络 (DDQN) 的两栖无人艇 (USV) 全局路径规划算法, 利用电子海图和高程图构建环境模型, 结合动作掩码和路径平滑方法, 生成合理且高效的路径. 马天等人[14]基于深度强化学习提出了一种在有限观测空间下优化移动机器人三维路径规划的方法, 能够满足线性时序逻辑约束下的多目标路径规划任务. 倪培龙等人[15]进一步对 DDQN 进行了优化, 通过最大化目标网络预测的 Q 值, 提升了路径规划的效率, 并实现了更优的规划效果. 董永峰等人[16]将 DDQN 算法与平均 DQN 算法融合, 提升了算法的避障性能与路径规划的效率. 然而, 这些改进的 DQN 算法依然存在过估计现象, 导致在复杂环境下的路径规划精度以及平滑度不足. 同时学习效果以及收敛性较差, 尤其在应对复杂障碍物或未知环境时, 模型的学习容易遇到瓶颈或者存在不稳定性.

基于上述问题, 本文针对决斗深度双 Q 网络 (dueling double DQN) 进行改进, 主要基于以下考虑: 首先, 决斗网络结构虽然能够有效分离状态价值和动作优势, 但是在处理高维状态空间时仍存在估计偏差问题; 其次, DDQN 网络结构在复杂环境中的探索效率较低, 难以平衡探索与利用的关系; 最后, 现有算法的奖励函数设计较为简单, 无法准确反映机器人在不同状态下的决策价值. 因此, 本文提出了一种改进的决斗深度双 Q 网络算法, 结合长短期记忆网络 (LSTM) 的时空特征提取能力, 显著提升算法对复杂障碍物分布及环境不确定性的适应能力, 通过引入 Dropout 层增强模型泛化能力, 同时引入优先经验回放机制, 采用基于模拟退火的动态 epsilon-greedy 策略优化探索效率, 并设计了一种动态奖励值函数, 根据机器人距离目标的远近实时调整奖励值, 从而提升路径规划的精度和效率. 实验结果表明, 本文提出的算法在复杂地图环境中的表现显著优于现有方法, 为移动机器人自主导航提供了更为可靠的解决方案.

1 强化学习框架

1.1 Q-learning 算法

Q 学习是一种经典的强化学习算法, 其起源可以追溯到 20 世纪 80 年代初期. 它最初由 Watkins[17]在他的博士论文中提出, 是一种无模型的强化学习算法. 这意味着它不需要环境的先验模型或明确的状态转移概率, 而是通过与环境的直接交互来学习策略. Q 学习基于马尔可夫决策过程 (Markov decision process, MDP) 的框架, 通过维护一个 Q 表格 (状态-动作值表) 来学习每个状态-动作对的期望价值. 这个 Q 表记录了在特定状态下执行某一动作后所能获得的预期回报[18].

在 Q 学习中, 智能体通过与环境互动获得反馈, 并

不断更新 Q 值以逐渐逼近最优策略. 该更新过程遵循贝尔曼方程, 利用即时奖励 (即在当前步骤获得的奖励) 和后续状态的折扣回报来更新 Q 值. 随着时间的推移, Q 学习算法通过多次迭代逐渐逼近最优策略, 即在每个状态下选择能获得最大累积回报的动作. 值更新公式为:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha[r_{t+1}+\gamma \max Q(s_{t+1},a')-Q(s_t,a_t)] \tag{1}$$

其中, α是学习率, γ是折扣因子, r_{t+1}是执行动作a_t后从状态s_t转移到状态s_{t+1}所获得的即时奖励. Q 学习中常使用ε-greedy 策略来平衡探索和利用. 以ε概率选择一个随机动作 (探索). 以$1-\varepsilon$的概率选择使$Q(s,a)$最大的动作 (利用). 其公式如下:

$$\pi(a|s)=\begin{cases} \varepsilon/|A|+1-\varepsilon, & a^*=\underset{a\in A}{\operatorname{argmax}} Q(s,a) \\ \varepsilon/|A|, & \text{其他} \end{cases} \tag{2}$$

1.2 DQN 算法

深度 Q 网络 (deep Q-network, DQN) 由 Mnih 等人[10]于 2015 年提出来解决高维状态空间中的决策问题. 该方法结合了传统 Q 学习算法和深度神经网络, 通过使用深度神经网络来近似动作值函数, 从而有效应对高维状态空间的挑战. 与传统 Q 学习直接更新 Q 值表不同, DQN 通过最小化损失函数来更新网络参数:

$$L(\theta)=E[(r(s,a,s')+\gamma \max_{a'\in A} Q(s',a';\theta^-)-Q(s,a,;\theta))^2] \tag{3}$$

其中, s和s'分别表示当前状态和后继状态, a是在状态s下采取的动作, r是奖励, θ^-表示目标网络的参数. DQN 的关键创新之一是引入目标网络[19], 该网络的参数θ^-定期从主网络复制而来, 以稳定学习过程. 其最优 Q 函数为:

$$Q^*(S,A,\theta)=E_{s'}[r(s,a,s')+\gamma \max_{a'\in A} Q^*(s',a')|s,a] \tag{4}$$

如图 1 所示, DQN 算法的架构主要包括以下几个阶段[20]. 在 DQN 算法的初始阶段, 智能体通过与环境的交互探索并积累经验, 构建经验回放池. 该回放池存储了大量的状态-动作-奖励-新状态序列, 用于训练深度神经网络, 从而逐步逼近 Q 值函数并近似状态-动作值函数. 在此过程中, 智能体根据当前网络参数评估动作值 (Q 值), 并选择对应的最优行为执行. 通过这些行为, 智能体获取新的状态和奖励, 不断更新经验回放池. 在训练过程中, 智能体从经验回放池中随机抽取样本, 以更新 Q 网络. 同时, 为了提高学习的稳定性, DQN 引入了一个目标 Q 网络 (target Q-network), 用于计算目标 Q 值. 通过最小化预测 Q 值与目标 Q 值之间的差异, 智能体优化了策略, 逐渐提升在复杂环境中的决策能力, 进而实现更高的长期累积奖励. 这种机制能够有效缓解策略更新中的不稳定性, 确保智能体逐步收敛到最优行为策略[21].

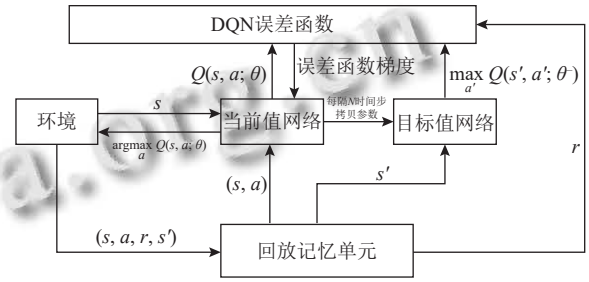

图 1 DQN 运行结构图

1.3 DDQN 和 Dueling DQN

深度双 Q 网络 (DDQN) 是一种改进的深度强化学习算法, 旨在解决传统 DQN 算法中存在的 Q 值高估问题. 在传统 DQN 算法中, 动作选择与目标 Q 值的计算均依赖于同一个 Q 网络, 这容易导致 Q 值被系统性地高估. 对于某些动作, DQN 会产生不均衡的过高估计, 从而影响智能体的决策过程.

为了解决这一问题, DDQN 通过将动作选择与目标 Q 值的计算解耦, 显著减少了 Q 值的高估偏差. 具体而言, DDQN 引入了两个独立的神经网络: 一个是主 Q 网络, 负责选择最优动作; 另一个是目标 Q 网络, 专门用于计算目标 Q 值. 在每一步更新中, 主 Q 网络首先选择动作, 然后目标 Q 网络根据该动作计算目标 Q 值. 这种机制有效避免了 Q 值的过高估计问题, 提升算法在复杂决策环境中的性能. 具体的更新规则如下:

$$y_t^{\text{DDQN}}=r_t+\gamma Q(s_{t+1},\arg\max_a Q(s_{t+1},a;\theta_t);\theta_t^-) \tag{5}$$

由参数θ_t的主网络选择 Q 值最大对应的动作, 然后使用参数θ_t^-的目标网络计算该动作对应的目标值, 并对所选动作进行评估. 通过这种方式, DDQN 有效地解耦了动作选择和目标 Q 值计算, 避免了 DQN 中可能出现的高估问题. 这一改进使得算法在复杂环境下能够做出更准确的动作选择, 提高了决策的稳定性和策略优化效果.

Dueling DQN 通过优化 Q 值函数的结构来提升学

习效率和策略质量[22]. 该算法引入了“决斗”网络结构, 如图 2 所示, 将 Q 值分解为状态值函数和优势函数的和, 更精细地评估状态和动作的价值. Q 值的计算公式如下:

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\beta)+A(s,a;\theta,a) \tag{6}$$

由于网络直接输出 Q 值, 所以状态值V和动作A是未知的, 为了体现这种可辨识性, 在保证性能的同时, 对动作优势进行集中处理. 具体来说, 使用所有可能动作的平均优势值来修正 Q 值.

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\beta)+(A(s,a;\theta,a)-\frac{1}{|A|}\sum_{a'}A(s,a';\theta,\alpha)) \tag{7}$$

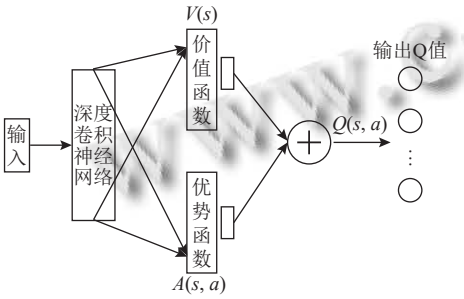

图 2 Dueling DQN 模型结构

2 改进深度强化学习算法

2.1 R-D3QN 算法

D3QN (dueling double deep Q-network) 算法的原理为在 Dueling DQN 算法的基础上融入了 Double DQN 算法的思想, 即通过双 Q 网络降低过估计, 又通过决斗网络提升决策精度, 从而在路径规划的平滑性和收敛性方面表现更优.

算法首先引入状态值函数V_π和动作优势函数A_π, 其中状态值函数表示在状态s下采取最优策略时, 期望的累计回报, 公式如下:

$$V_\pi=E_{a\sim\pi(s)}[Q_\pi(s,a)] \tag{8}$$

动作优势函数表示在状态s下, 选择动作a相对于在该状态下的平均动作值的相对优势. 具体公式如下:

$$A_\pi(s,a)=Q_\pi(s,a)-V_\pi(s) \tag{9}$$

R-D3QN 为了突出关键动作, 在优势函数的均值计算中引入一个加权参数, 使得重要的动作可以被赋予更大的权重, 从而在 Q 值计算时对关键动作有更精确的估计. 具体公式可以改写为:

$$Q(s,a;\alpha,\beta,\theta)=V(s;a,\theta)+A(s,a;\beta,\theta)-\frac{1}{\sum_{a'}w_{a'}}\sum_{a'}w_{a'}A(s,a';\beta,\theta) \tag{10}$$

其中, s表示状态, a表示动作, α是价值函数支路的网络参数, β是优势函数支路的网络参数, θ为公共部分的网络参数. $w_{a'}$是加权参数, 计算方法如下:

$$w_{a'}=w_{a'}+\eta\cdot\Delta Loss \tag{11}$$

其中, η为调整步长, $\Delta Loss$表示损失函数的变化量. 当某个动作的 Q 值逐渐增大时, 相应的权重$w_{a'}$也随之增加; 反之, 当某个动作的效果不理想时, 可以适当降低该动作的权重.

R-D3QN 使用两个具有相同结构和初始参数的网络, 即估计网络和目标网络, 根据所选动作计算出目标网络中的目标 Q 值, 即:

$$y_{\text{target}}^{\text{R-D3QN}}=r+\gamma Q(s_{t+1},\arg\max Q(s_{t+1},a;\theta);\theta^{-}) \tag{12}$$

其损失函数为:

$$Loss=E[(y_{\text{target}}^{\text{R-D3QN}}-Q(s,a;\theta))^2] \tag{13}$$

其中, $Q(s_{t+1},a_{\max};\theta^{-})$为目标网络的值函数, $Q(s,a;\theta)$为估计网络值函数.

在传统的深度 Q 网络 (DQN) 算法中, 简单的 3 层全连接神经网络结构在处理低维度任务时表现尚可, 但在面对大规模长序列图像问题时效果不尽人意. 为了提升模型的表达能力, R-D3QN 引入了二维卷积神经网络 (2D CNN)、长短期记忆网络 (long short-term memory, LSTM)、Dropout 层与全连接层的组合架构, LSTM 通过引入具有门控机制的单元 (包括输入门、遗忘门和输出门) 来有效地控制信息的存储与更新, 从而能够捕捉时间序列中的长期依赖关系. 在移动机器人路径规划任务中, LSTM 能够通过建模机器人与环境之间的复杂动态交互, 学习并记忆历史运动轨迹及环境变化的相关特征. 特别是在大空间区域环境中, LSTM 通过其门控机制有效地调整机器人路径决策过程, 适应环境中多障碍物的出现以及路径规划中的不确定性. 具体而言, 输入门决定哪些新信息应被存储, 遗忘门则控制哪些过时的信息应被丢弃, 输出门则基于当前状态和历史信息生成合理的动作决策. 通过这些机制, LSTM 显著提升了移动机器人在复杂环境下的路径规划能力.

2.2 改进探索策略

为优化智能体的动作选择策略, 减少无效探索并规避局部最优解问题, 本节对传统策略进行了改进, 提出了引入模拟退火算法中的控制温度机制的方法. 该机制通过在动作选择过程中引入随机性, 平衡了策略利用与探索之间的关系: 一方面, 能够充分利用智能体的已有知识; 另一方面, 在局部最优解附近保留适度的探索能力. 其数学表达式如下, 旨在提升策略在复杂环境中的适应性和学习效率.

$$\varepsilon_t = \frac{1}{1+\exp\left(\dfrac{Q(s_t,a)}{T}\right)} \tag{14}$$

其中, T 是模拟退火的当前温度. 随着温度 T 的逐渐降低, ε_t 逐渐减小, 从而使得选择最优动作的概率逐步增大, 减少探索, 更多依赖已经学到的 Q 值. 温度更新公式如下:

$$T_{t+1} = 0.98T_t \tag{15}$$

因此, 在路径规划过程中, 智能体从起点出发时, 温度参数处于较高水平, 探索率 ε 也相对较大. 这使得改进后的 R-D3QN 算法在初期能够更积极地探索未知区域, 从而积累丰富的环境经验. 随着算法的学习过程逐步推进, 温度参数逐渐降低, 探索概率相应减小, 系统对 Q 值的依赖程度逐渐增加, 策略倾向于选择当前已知的最优路径. 这种温度递减机制不仅有效平衡了探索与利用的关系, 还能够提高算法在复杂环境中的学习效率和决策能力.

2.3 改进奖励函数

在现有的强化学习算法中, 过于简单的奖励函数常导致移动机器人在执行避障与路径规划任务时缺乏明确的目标方向, 长时间停留于探索阶段, 从而降低了学习效率. 因此提出了一种改进的奖励机制, 根据智能体所处的不同环境和状态动态调整奖励函数, 以加速学习过程, 帮助智能体更有效地找到迷宫的出口, 并避免与障碍物发生碰撞. 该设计能够优化探索策略, 使得智能体在复杂环境中的表现更加稳健.

奖励机制如式 (16) 所示. 当智能体成功到达迷宫的目标位置时, 给予其一个高额奖励 (+100), 以强烈激励智能体尽快找到出口. 若智能体发生碰撞, 则施加负奖励 (−2), 惩罚不安全行为. 此外, 在智能体每一步移动中, 若未发生碰撞且未到达目标位置, 给予一个小的奖励 (+1), 鼓励智能体尽快完成任务, 避免无效的长时间探索.

$$R(s_t,a_t) = \begin{cases} 100, & \text{达到终点} \\ -2, & \text{碰撞到障碍物} \\ 1, & \text{正常移动} \\ 0.5, & \text{向目标靠近} \\ -0.5, & \text{向目标远离} \end{cases} \tag{16}$$

每次移动后, 智能体与目标位置之间的距离通过曼哈顿距离公式进行计算. 此距离的动态反馈有助于指导智能体不断调整策略, 优化路径规划. 公式如下:

$$d(s,goal) = |x_s - x_{goal}| + |y_s - y_{goal}| \tag{17}$$

其中, (x_s,y_s) 为智能体坐标, (x_{goal},y_{goal}) 为目标终点坐标. 当智能体向终点靠近时, 即 $d(s_{t+1},goal) < d(s_t,goal)$, 设置奖励为 0.5, 反之亦然.

2.4 改进算法流程及伪代码

R-D3QN 算法的训练模型如图 3 所示, 移动机器人通过与环境交互, 可以得到序列样本 (s,a,r,s'), 并存放到经验回放池中, 随后从经验池中使用优先经验回放的采样策略抽取经验数据样本进行训练. 估计网络和目标网络分别从数据样本中提取特征, 并将这些特征分流到两个不同的函数中: 状态值函数 $V(s)$ 和动作优势函数 $A(s,a;\theta)$. 状态值函数用于评估当前状态 s 的整体价值, 而动作优势函数用于评估在特定状态 s 下, 选择动作 a 相对于其他可能动作的优越性.

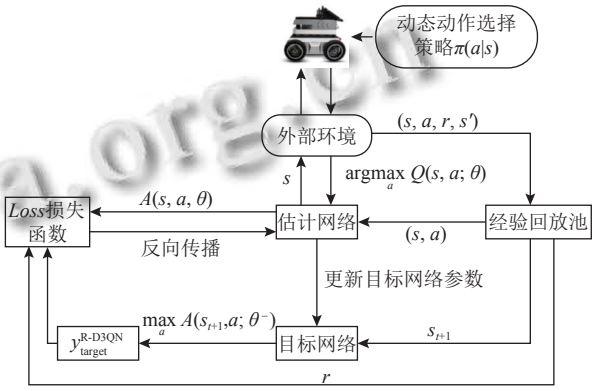

图 3 R-D3QN 训练流程

目标网络首先对动作优势函数 $A(s,a)$ 的输出进行处理, 并以此为基础计算目标值函数 $y_{\text{target}}^{\text{R-D3QN}}$. 然后通过计算目标值函数与估计网络预测的值函数的差异, 定义损失函数 *Loss*. 该损失函数通过优化器进行反向传播, 进而更新估计网络的权重参数. 这一过程不断迭代, 直到机器人成功到达目标位置, 或者训练达到预设的最大迭代次数. 通过上述训练机制, 算法有效提高了移动机器人在复杂环境中的路径规划能力.

综上所述, 结合前述的理论分析和算法改进过程, 经过对各个环节的优化与调整, 最终得出了改进后的算法. 算法 1 是基于这些改进所设计的算法伪代码, 展示了改进后的算法在具体实现过程中的步骤和逻辑.

算法 1. R-D3QN 算法

输入: 训练次数 N, 状态集S, 动作集A, 学习率α, 折扣因子γ, 初始温度T_0, 最低探索率$\varepsilon_{\min}$, 加权参数调整步长η.
输出: 最优动作价值函数$Q(s,a)$.

1) 初始化价值网络$V(S,\alpha,\theta)$和优势网络$A(s,a;\beta,\theta)$, 将 Q 值函数按式 (10) 进行初始化.
2) 初始化目标网络参数$\theta^-=\theta$, 探索率$\varepsilon=\varepsilon_0$, 所有动作的加权参数$w_a=1$.
3) 在网络结构中加 LSTM 结构和 Dropout 机制, 以提高网络的泛化能力.
4) **for** i=1:N
5) 初始化状态环境s.
6) **While** s不是终止状态
7) 根据动态ε-greedy 策略选择动作a:

$$a=\begin{cases}\text{随机选择一个动作}, & \text{以概率}\varepsilon\\ \arg\max\limits_a Q(s,a;\theta), & \text{以概率}1-\varepsilon\end{cases}$$

8) 执行动作a, 获得奖励R和下一个状态s'.
9) 根据策略选择下一步动作a'.
10) 根据式 (12) 计算目标 Q 值y.
11) 根据式 (13) 更新 Q 值网络的损失函数.
12) 更新网络参数θ以最小化损失.
13) 根据式 (11) 动态调整动作的加权参数w_a.
14) 将目标网络参数更新为主网络参数: $\theta^-=\theta$.
15) 根据式 (14) 动态调整探索率.
16) 更新状态$s=s'$.

3 实验仿真

3.1 仿真环境

实验环境的配置如下: 硬件方面, 实验所使用的服务器配备了主频为 3.2 GHz 的中央处理器 (CPU) 以及 32 GB 的内存. 软件框架采用了 TensorFlow 2.6.0, 编程语言使用 Python 3.10.9. 仿真地图环境的构建遵循以下规则: 黑色方块代表障碍物, 白色方块代表自由栅格, 机器人可以在自由栅格上自由移动. 机器人可通过上、下、左、右这 4 个方向进行移动. 当机器人遇到障碍物、超出地图边界或成功到达终点时, 当前回合结束. 仿真环境的具体设置如图 4 所示.

3.2 参数设置

在进行深度强化学习算法仿真时, 超参数的设置对于算法性能具有关键影响. 在本研究中, 采用 Adam 优化器, 卷积层使用 ReLU 作为激活函数. 学习率设定为 0.001, 折扣因子设置为 0.99, 模拟退火的初始温度设置为 2.0, 动作探索策略最低值为 0.01. 此设置的目的是在训练初期确保移动机器人能够充分探索未知环境, 随着训练的深入, 逐步提升机器人路径规划的效率. 具体算法的超参数设置见表 1.

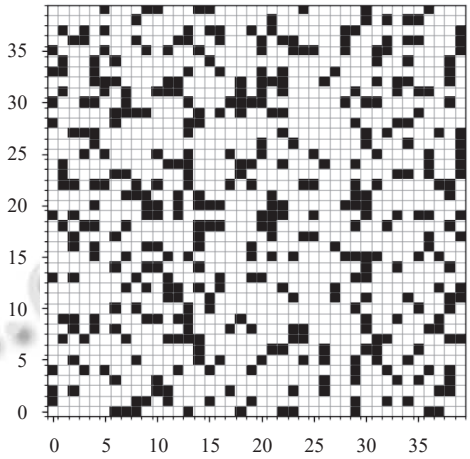

图 4 栅格仿真环境

表 1 参数设置

参数	值
优化器	Adam
学习率	0.001
折扣因子	0.99
经验回放缓冲区大小	10 000
初始温度	2.0
初始加权参数	1.0
探索率衰减系数	0.999 95
Dropout层丢弃率	0.5
网络激活函数	ReLU
最低探索率	0.01

本文设置在复杂环境下进行实验时所采用的对比指标, 包括平均奖励值和综合性能指标的对比分析.

(1) 平均奖励值

在实验过程中, 通过对比分析每次训练迭代后的平均奖励值, 以评估学习策略的优化效果. 该值越高, 意味着智能体的策略越为有效, 能够在环境中获得更多的正向反馈, 进而反映出策略优化的成功程度和智能体在任务执行中的表现优劣. 平均奖励值R_{avg}的计算公式如下:

$$R_{\text{avg}}=\frac{\sum_{t=1}^{T}R_t}{T} \tag{18}$$

其中, R_t代表在一次训练迭代的第t步中, 移动机器人执行特定动作所获得的即时奖励值; 而T则指在该训练迭代过程中, 机器人所经历的总步数.

(2) 综合性能指标

路径规划的综合性能指标包括算法的收敛的迭代次数, 路径的拐点数, 算法训练中的碰撞次数以及上述的平均奖励值. 其中, 路径拐点数反映了规划路径的复杂度和直线化程度, 较少的拐点通常表明路径更加简洁高效, 能够减少不必要的路径调整. 碰撞次数衡量了智能体在训练过程中与障碍物发生碰撞的频率, 较低的碰撞次数表明算法能够有效规避障碍物, 从而确保路径的安全性与可行性.

3.3 40×40 栅格环境

本节将在一个 40×40 的大区域环境中进行路径规划实验. 在栅格法实验设置中, 初期的探索因子被设定为 1, 这意味着机器人在初期阶段完全依赖随机选择的动作. 然而, 如果将碰撞作为训练结束的条件, 机器人可能会过早终止探索过程, 从而未能充分了解和适应环境, 进而影响学习效果. 因此本文将机器人成功到达目标点作为路径规划训练迭代的结束标志. 在该实验场景中, 比较了 4 种路径规划算法的性能: 深度 Q 网络 (DQN) 算法、深度双 Q 网络 (DDQN) 算法、文献[23]中的 PMR-Dueling DQN 算法以及 R-D3QN 算法. 实验结果如图 5 所示, 清晰地展示了各算法在复杂环境中路径规划的效果及其差异.

从路径规划的角度分析, 在 40×40 的大区域障碍物环境中, 4 种算法 (DQN、DDQN、PMR-Dueling DQN 和 R-D3QN) 均能够成功找到目标路径. 然而, 不同算法在路径质量和规划效率上存在显著差异.

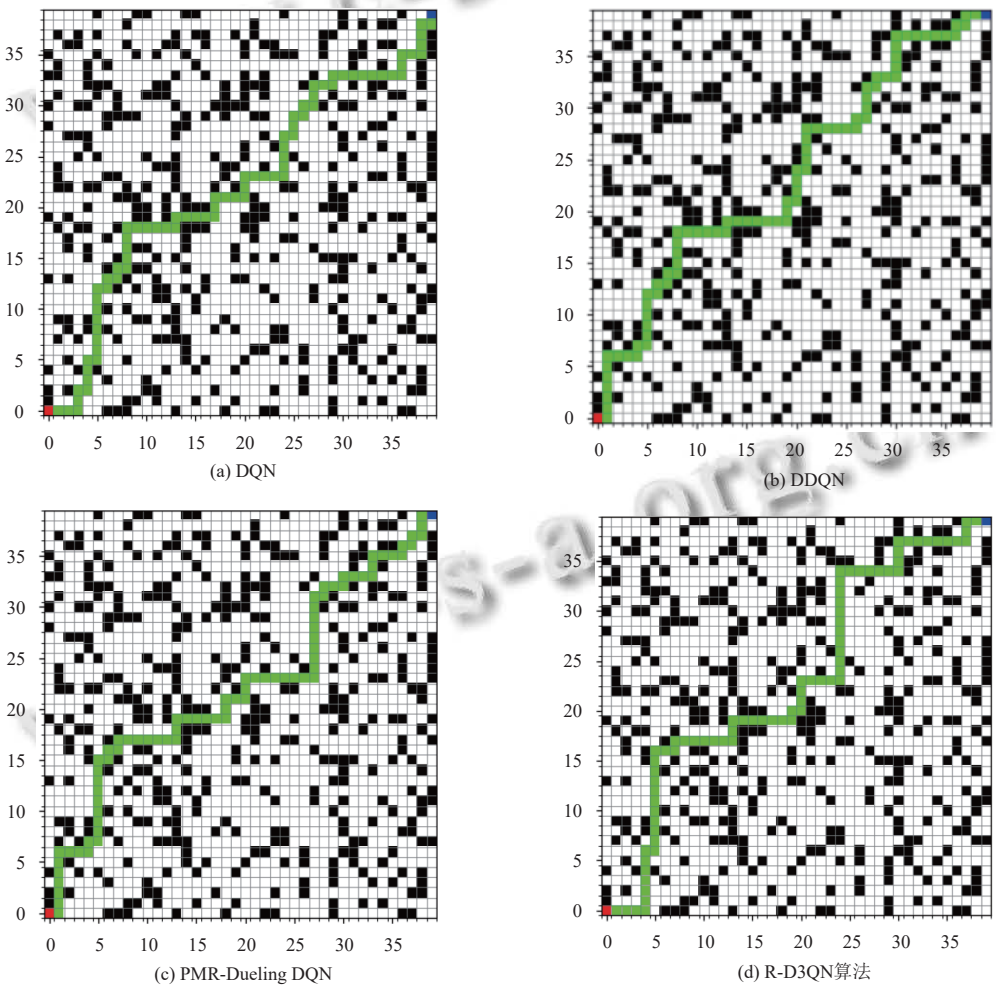

图 5 路径规划结果可视图

R-D3QN 算法在路径表现上优于其他算法, 其规划路径的拐点数为 18 个, 路径更加平滑, 冗余移动显著减少. 这表明, R-D3QN 算法在适应性和路径效率方面具有明显优势. 相比之下, DQN 和 DDQN 算法在障

碍物较密集区域的路径规划效率略显不足, 其中 DQN 算法出现 33 个拐点, DDQN 算法出现 32 个拐点, 均存在一定程度的路径冗余问题. 尤其是 DQN 算法, 由于过估计的问题, 路径效率表现出明显的缺陷. 而 PMR-Dueling DQN 算法的表现则与 DDQN 算法较为接近, 虽然在稳定性上具有一定优势, 但在路径平滑性和效率上未能超越 R-D3QN.

由于训练过程中涉及较多迭代次数, 为了更加全面和准确地评估算法性能, 并减小随机波动对结果的影响, 数据分析采用每 100 次迭代的平均奖励值进行比较. 所有算法的训练迭代次数统一设置为 5 000 次, 并以从起点到目标点的完整路径规划作为一次训练迭代的结束标准.

通过对 4 种深度强化学习算法 (DQN、DDQN、PMR-Dueling DQN 和 R-D3QN) 的性能差异进行分析, 基于平均奖励值的曲线图更直观地展现了不同算法在训练过程中的性能变化趋势. 图 6 展示了这些算法在训练过程中平均奖励值的变化情况, 为深入比较各算法在路径规划任务中的表现提供了直观依据.

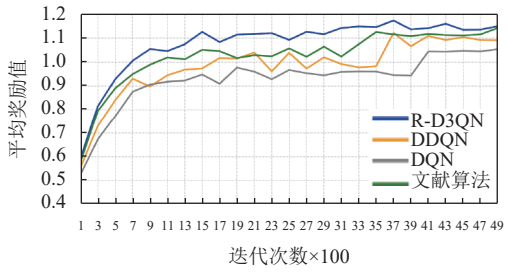

图 6 平均奖励值对比图

从实验结果分析可知, R-D3QN 算法的平均奖励值整体上明显高于 DQN 算法、DDQN 算法和 PMR-Dueling DQN, 充分表明 R-D3QN 在路径规划效率和环境适应性方面表现出显著的优越性. 具体而言, R-D3QN 通过改进的网络结构和优化的奖励机制, 有效增强了大空间区域环境中障碍物分布的适应能力. 与之相比, DQN 和 DDQN 在初始阶段的奖励值增长速度相对较慢, 且在 1 500–3 000 代时存在较大的波动性, 这是由于 DDQN 算法虽然解决了 DQN 算法中的高估问题, 但由于它使用随机经验回放, 并且仅依赖于一步的有效信息, 因此在训练的后期 (第 3 700 代后), DDQN 算法的训练曲线仍然存在一定波动, 最终得到的策略也未必是全局最优. PMR-Dueling DQN 在算法前期增长较快, 但在 2 300–3 500 代时波动较大, 其最终奖励值略低于 R-D3QN, 说明其在稳定性上还略有不足. 曲线结果表明, R-D3QN 的学习稳定性更强, 路径规划总体表现优于其他算法.

表 2 中记录了本次实验的综合性能指标. 从结果可以看出, R-D3QN 算法的平均奖励值为 1.146, 显著高于 DQN、DDQN 和 PMR-Dueling DQN, 分别提升了 9.25%、4.46% 和 2.23%. 在收敛效率方面, R-D3QN 算法的收敛次数相较于 DQN、DDQN 和 PMR-Dueling DQN 分别减少了 24.39%、23.46% 和 8.82%, 表明其在训练效率上具有更强的优势.

表 2 算法综合性能指标

算法名称	路径拐点数	碰撞次数	平均奖励值	收敛次数
DQN	33	24 956	1.049	4 100
DDQN[11]	32	22 945	1.097	4 050
PMR-Dueling DQN[23]	30	17 357	1.121	3 400
R-D3QN	**18**	**14 674**	**1.146**	**3 100**

针对障碍物碰撞次数的评估, R-D3QN 算法展现了显著的性能改进. 在学习交互的过程中, R-D3QN 算法的障碍物碰撞次数为 14 674 次, 而 DQN、DDQN 和 PMR-Dueling DQN 的碰撞次数分别为 24 956 次、22 945 次和 17 357 次. 与 DQN 和 DDQN 及 PMR-Dueling DQN 相比, R-D3QN 的碰撞次数分别减少了 41.20%、36.05% 和 15.46%. 结果表明, R-D3QN 算法能够更加准确地估计机器人与障碍物的相对位置, 在路径规划任务中显著降低碰撞风险. 障碍物碰撞次数的显著减少不仅验证了 R-D3QN 算法在避障性能上的优化, 也体现了其在目标路径规划和策略探索能力方面的显著提升.

3.4 60×60 栅格环境

本节将对一个 60×60 的大区域环境进行路径规划实验. 实验中, 比较了 4 种路径规划算法的性能, 实验结果如图 7 所示.

从路径规划的角度分析, 在 60×60 的大区域障碍物环境中, 4 种算法 (DQN、DDQN、PMR-Dueling DQN 和 R-D3QN) 均能够成功找到目标路径. 然而, 在路径质量和规划效率方面, 它们存在显著差异.

R-D3QN 算法的路径表现最佳, 其规划路径中的拐点数量为 26 个, 路径十分平滑, 冗余移动显著减少, 充分展现了其在路径规划任务中的适应性和效率优势. 相比之下, DQN 和 DDQN 算法的拐点数量分别为 54 和 48 个, 路径中存在较多冗余, 尤其是 DQN 算法, 其

路径冗余更加明显, 暴露出其在复杂环境中效率不足的问题. PMR-Dueling DQN 算法的拐点数量为 44 个, 虽然相较于 DQN 和 DDQN 算法的路径平滑度有所提升, 但与 R-D3QN 相比仍有明显差距.

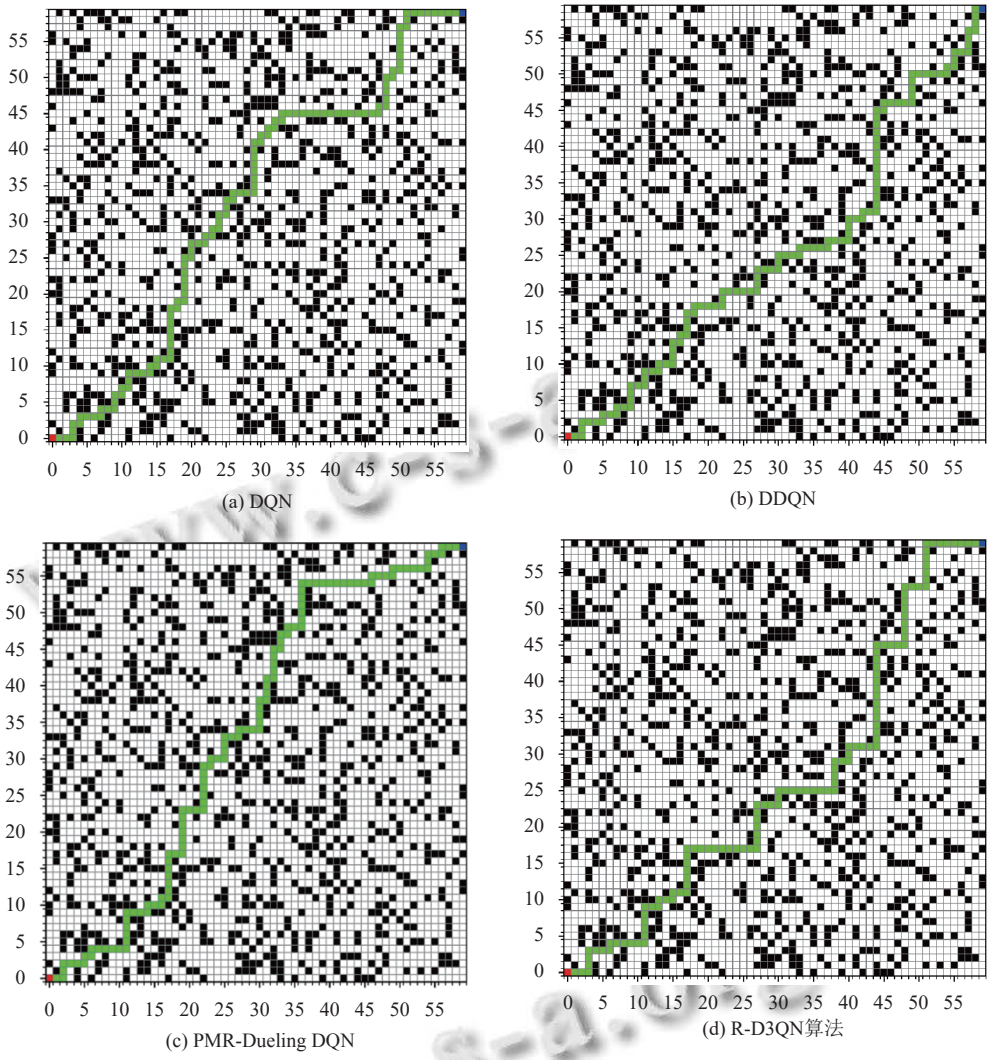

图 7 路径规划结果可视图

为了进一步比较这些算法的性能, 本次实验采用每 100 次迭代的平均奖励值进行分析, 训练迭代次数统一设置为 7 000 次, 并以从起点到目标点的完整路径规划完成作为一次训练迭代的结束标准. 基于平均奖励值的曲线图, 直观地展示了 4 种深度强化学习算法 (DQN、DDQN、PMR-Dueling DQN 算法和 R-D3QN) 在训练过程中的表现差异. 图 8 清晰地反映了各算法在训练过程中平均奖励的变化趋势, 有助于深入观察和理解它们在路径规划任务中的性能特性.

二次实验的结果进一步验证了前述结论. R-D3QN 算法在整个训练过程中获得的平均奖励值始终优于其他算法, 表现出更高的稳定性和效率. 尤其在迭代初期, R-D3QN 算法的奖励值增长速度最快, 尽管在第 3 900 代时其平均奖励值稍微落后于 PMR-Dueling DQN 算法, 但在随后的训练中, R-D3QN 算法迅速恢复并展示出较高的路径规划效率. 约在第 5 200 代时, 其平均奖励值逐渐趋于稳定, 表明该算法在路径规划中的长期优势. PMR-Dueling DQN 算法的整体曲线表现较为平稳, 波动较小, 展现了良好的适应性, 并且其在 5 000 代时就完成收敛, 优于 R-D3QN 算法. 然而其最终的平均奖励值仍略低于 R-D3QN 算法. 相比之下, DDQN 算法在第 2 100–3 900 代之间表现出了较大的波动性, 这一现象反映了其在动作选择过程中较强的随机性, 导致了路径规划的不稳定性. DQN 算法则在训练初期即

出现了较大的波动, 并且其最终的平均奖励值最低, 表明 DQN 算法的收敛速度较慢, 且在路径规划任务中的整体性能较差.

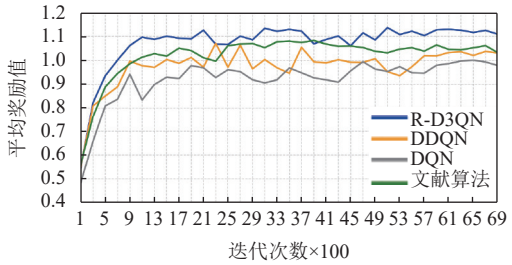

图 8 平均奖励值对比图

表 3 中记录了第 2 次实验的综合性能指标. 从实验结果可以看出, R-D3QN 算法的平均奖励值为 1.124, 显著高于 DQN、DDQN 及 PMR-Dueling DQN, 分别提升了 12.98%、9.46% 和 7.15%. 在收敛效率方面, R-D3QN 算法的收敛次数较 DQN 和 DDQN 分别减少了 11.86% 和 8.77%, 但与 PMR-Dueling DQN 算法相比, 增加了 4%. 尽管如此, R-D3QN 依然表现出较为优越的训练效率.

表 3 算法综合性能指标

算法名称	路径拐点数	碰撞次数	平均奖励值	收敛次数
DQN	54	35938	0.995	5900
DDQN[11]	48	33457	1.027	5700
PMR-Dueling DQN[23]	44	26956	1.049	**5000**
R-D3QN	**26**	**20798**	**1.124**	5200

在障碍物碰撞次数的评估中, R-D3QN 算法展现出显著的性能提升. 在学习交互过程中, R-D3QN 的障碍物碰撞次数为 20798 次, 而 DQN、DDQN 和 PMR-Dueling DQN 中的算法的碰撞次数分别为 35938 次、33457 次和 26956 次. 与 DQN、DDQN 及 PMR-Dueling DQN 相比, R-D3QN 的碰撞次数分别减少了 42.14%、37.84% 和 22.85%. 这一结果表明, 无论环境的复杂度如何, R-D3QN 算法在障碍物碰撞避免方面均表现出显著的优势. R-D3QN 算法能够有效地应对不同环境条件下的路径规划挑战, 通过更加精准的相对位置估计与策略优化, 显著降低了机器人与障碍物之间的碰撞风险. 这一优异表现在不同的大区域复杂环境中得到了验证, 证明了该算法具有较强的鲁棒性和广泛的适用性.

4 结论与展望

本文针对复杂环境下的机器人避障与路径规划问题, 提出了一种基于深度强化学习的改进算法. 在传统深度 Q 网络算法的基础上, 融合双 Q 网络和决斗网络架构, 并在网络结构中引入了 LSTM 架构. 此外, 通过引入基于模拟退火的动态ε-greedy策略, 有效避免了后期训练中对无效状态的过度探索, 从而加快了算法的收敛速度. 为了进一步提升算法的路径规划效率, 本文设计了动态奖励函数, 在不同场景下对路径质量进行自适应评价.

实验结果验证了本文方法的优越性: 在 40×40 栅格环境中, 改进算法的平均奖励值较传统 DQN 算法提升了 9.25%, 收敛次数减少了 24.39%, 碰撞次数减少了 41.20%; 在 60×60 栅格环境中, 改进算法的平均奖励值较传统 DQN 算法提升了 12.98%, 收敛次数减少了 11.86%, 碰撞次数减少了 42.14%. 并且与 DQN 的衍生算法相比, 改进后的 R-D3QN 算法在平均奖励值、收敛效率和障碍物碰撞避免方面也表现出显著优势. 这些结果表明, 所提出的 R-D3QN 算法在路径平滑度和避障能力上均有显著提高. 未来研究方向将专注于进一步提升算法的计算效率, 并引入动态地图进行实时路径规划, 以应对环境变化对路径规划的挑战.

参考文献

1 闫皎洁, 张锲石, 胡希平. 基于强化学习的路径规划技术综述. 计算机工程, 2021, 47(10): 16–25. [doi: 10.19678/j.issn.1000-3428.0060683]

2 唐伟祥. 大规模三维环境下的自主探索与搜索救援应用研究 [硕士学位论文]. 合肥: 中国科学技术大学, 2023. [doi: 10.27517/d.cnki.gzkju.2023.001838]

3 郭建, 杨朋, 曾志豪, 等. 融合改进 Dijkstra 算法和动态窗口法的移动机器人路径规划. 组合机床与自动化加工技术, 2024(3): 36–40. [doi: 10.13462/j.cnki.mmtamt.2024.03.008]

4 喻蝶, 鲍柏仲, 司言, 等. 基于搜索步优化 A*算法的移动机器人路径规划. 系统仿真学报, 2025, 37(4): 1041–1050. (2024-04-12)[2024-04-26]. [doi: 10.16182/j.issn1004731x.joss.23-1574]

5 鲜斌, 宋宁. 基于模型预测控制与改进人工势场法的多无人机路径规划. 控制与决策, 2024, 39(7): 2133–2141. [doi: 10.13195/j.kzyjc.2023.0892]

6 曾胜, 王兵, 戴贤君. 基于改进的蚁群算法的目标物流车辆路径优化. 现代电子技术, 2024, 47(7): 181–186. [doi: 10.16652/j.issn.1004-373x.2024.07.032]

7 肖金壮, 余雪乐, 周刚, 等. 一种面向室内 AGV 路径规划的

改进蚁群算法. 仪器仪表学报, 2022, 43(3): 277–285. [doi: 10.19650/j.cnki.cjsi.J2109071]
8 黄荣杰, 王亚刚. 基于可视图与改进遗传算法的机器人平滑路径规划. 控制工程, 2024, 31(4): 678–686. [doi: 10.14107/j.cnki.kzgc.20210973]
9 赵迪, 何克勤, 赵祖高. 基于改进粒子群优化算法的移动机器人路径规划. 传感器与微系统, 2023, 42(6): 150–153. [doi: 10.13873/J.1000-9787(2023)06-0150-04]
10 Mnih V, Kavukcuoglu K, Silver D, *et al*. Playing Atari with deep reinforcement learning. arXiv:1312.5602, 2013.
11 van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix: AAAI Press, 2016. [doi: 10.1609/aaai.v30i1.10295]
12 Wang Z, Schaul T, Hessel M, *et al*. Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: JMLR.org, 2016. 1995–2003.
13 Yang AF, Shi YL, Liu W, *et al*. Global path planning algorithm based on double DQN for multi-tasks amphibious unmanned surface vehicle. Ocean Engineering, 2022, 266: 112809. [doi: 10.1016/J.OCEANENG.2022.112809]
14 马天, 席润韬, 吕佳豪, 等. 基于深度强化学习的移动机器人三维路径规划方法. 计算机应用, 2024, 44(7): 2055–2064. [doi: 10.11772/j.issn.1001-9081.2023060749]
15 倪培龙, 毛鹏军, 王宁, 等. 基于改进 A-DDQN 算法的机器人路径规划. 系统仿真学报, 2024: 1–10. https://link.cnki.net/urlid/11.3092.V.20240517.1848.001. (2024-05-20)[2024-07-22]. [doi: 10.16182/j.issn1004731x.joss.24-0369]
16 董永峰, 杨琛, 董瑶, 等. 基于改进的 DQN 机器人路径规划. 计算机工程与设计, 2021, 42(2): 552–558. [doi: 10.16208/j.issn1000-7024.2021.02.037]
17 Watkins CJCH. Learning from delayed rewards [Ph.D. Thesis]. Cambridge: University of Cambridge, 1989.
18 王慧, 秦广义, 夏鹏, 等. 基于改进强化学习算法的移动机器人路径规划研究. 计算机应用与软件, 2022, 39(7): 269–274. [doi: 10.3969/j.issn.1000-386x.2022.07.041]
19 康振兴. 基于路径规划和深度强化学习的机器人避障导航研究. 计算机应用与软件, 2024, 41(1): 297–303. [doi: 10.3969/j.issn.1000-386x.2024.01.043]
20 田丽蓉. 基于深度 Q 网络的移动机器人路径规划算法研究 [硕士学位论文]. 重庆: 重庆大学, 2022. [doi: 10.27670/d.cnki.gcqdu.2022.003579]
21 王小康, 冀杰, 刘洋, 等. 基于改进 Q 学习算法的无人物流配送车路径规划. 系统仿真学报, 2024, 36(5): 1211–1221. [doi: 10.16182/j.issn1004731x.joss.23-0051]
22 赵恬恬, 孔建国, 梁海军, 等. 未知环境下基于 Dueling DQN 的无人机路径规划研究. 现代计算机, 2024, 30(5): 37–43. [doi: 10.3969/j.issn.1007-1423.2024.05.006]
23 Deguale DA, Yu LL, Sinishaw ML, *et al*. Enhancing stability and performance in mobile robot path planning with PMR-Dueling DQN algorithm. Sensors, 2024, 24(5): 1523. [doi: 10.3390/S24051523]

(校对责编: 张重毅)