控制与决策
Control and Decision

基于改进经验回放策略的路径规划算法

李佩哲, 张文彪

引用本文:
李佩哲, 张文彪. 基于改进经验回放策略的路径规划算法[J]. 控制与决策, 2025, 40(8): 2545-2552.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.1138

基于改进经验回放策略的路径规划算法

李佩哲，张文彪†

(华北电力大学 控制与计算机工程学院，北京 102206)

摘　要: 移动机器人的路径规划和避障问题已成为近年来的研究热点. 现有的基于深度Q网络算法在 RPP 问题上取得了一定的效果. 然而, 该算法在训练过程中存在动作选择随机性过大、收敛速度慢等问题. 此外, 现有的算法较少涉及动态环境的定量分析. 鉴于此, 提出一种基于双深度Q网络的路径规划算法. 首先, 设计一种特别的时序输入结构, 能够采集更加丰富的动态语义信息, 可以更好地适应动态场景的路径规划; 然后, 设计一种独特的经验分配策略, 这种策略可在不同的训练阶段分配不同经验池中的经验, 以改善网络的训练效率; 最后, 在静态和动态环境中对所提出算法进行验证. 与改进前的方法相比, 所提出方法训练时间减少了 50%, 路径规划的成功率提高了 9.6%.

关键词: 路径规划；移动机器人；深度强化学习；优先经验回放；随机环境；动态环境

中图分类号: TP242　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.1138

引用格式: 李佩哲, 张文彪. 基于改进经验回放策略的路径规划算法 [J]. 控制与决策, 2025, 40(8): 2545-2552.

A path planning algorithm based on improved experience replay strategy

LI Pei-zhe，*ZHANG Wen-biao*†

(School of Control and Computer Engineering，North China Electric Power University，Beijing 102206，China)

Abstract: The path planning problem for mobile robots has garnered significant attention in recent years. While existing algorithms based on a deep Q network have exhibited a certain degree of effectiveness, they often struggle with excessive randomness in action selection and slow convergence during training. Moreover, existing algorithms are less involved in quantitative analyses of dynamic settings. Therefore, a path planning algorithm based on a double deep Q network (DDQN) is introduced, featuring a unique time-series input structure. Furthermore, a distinctive experience distribution strategy is proposed, which optimizes network training efficiency by distributing experiences from different experience pools at different training stages. The proposed algorithm is evaluated in both static and dynamic environments. Compared to the DDQN method, the proposed algorithm reduces training time by 50% and increases the success rate by 9.6%.

Keywords: robot path planning；mobile robots；deep reinforcement learning；prioritized experience replay；random environments；dynamic environments

0　引　言

移动机器人的路径规划 (RPP) 作为自主机器人以及无人驾驶系统的核心技术之一, 正面临日益严峻的挑战. 特别是在智能变电站、火力发电厂、重金属冶炼工厂等包含复杂动态环境下[1], 实现一种快速且高效的路径规划方法成为了一个亟待解决的关键问题[2-3].

路径规划的目标是生成一条从起点到目标的可行路径, 同时避开环境中的障碍物. 通常, 机器人会根据性能指标 (如最短路径长度、转弯次数等) 来寻找最优路径[4]. 在工业环境中, 由于存在复杂多变的动态障碍, 这类情形常被视为缺乏全局信息, 局部路径规划算法在此类问题中得到了广泛应用.

局部规划算法能够提供实时反馈, 并根据感知到的环境数据动态调整轨迹. 然而, 由于缺乏全面的全局环境信息, 规划的路径可能并非是最优的[5]. 目

收稿日期: 2024-09-24；**录用日期:** 2025-02-09.
基金项目: 北京市自然科学基金项目 (3242023).
责任编委: 方勇纯.
†通信作者. E-mail: wbzhang@ncepu.edu.cn.
本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

前, 常用的局部路径规划算法包括人工势场法 (APF) 以及以强化学习 (RL) 为基础的智能算法: APF 包含一个虚拟的引力场, 吸引力用于引导机器人朝着目标前进, 而排斥力用于使得机器人避开障碍物以实现路径规划, 但是, 这种方法极易陷入局部最优[6]; RL 是一种通过智能体与环境交互来学习和做出决策的方法, 智能体根据当前状态并采取行动, 通过试错, 不断迭代, 其最终目标是随着时间的推移最大化累积奖励, 最后形成一个实现避障和规划的决策系统[7].

Li 等[8] 提出了一种基于深度Q网络 (DQN) 的方法, 这种方法通过密集连接网络结构来提高学习效率和路径规划的准确性; Yi 等[9]改进了 DQN 算法, 通过对神经网络中动作选择和Q值估计这两个步骤进行解耦, 解决了Q值的高估问题; Yu 等[10] 实现了一种基于双深度Q网络 (DDQN) 的农业机器人避障方法, 避障性能良好; Lv 等[11] 提出了一种基于改进 DQN 的学习策略, 这种方法的优势在于在不同学习阶段根据经验深度和广度的不同需求, 通过建立经验价值评估网络和并行探索结构, 提高了学习效率和路径规划的准确性.

以上研究针对当前 RL 路径规划算法的不足进行了有益探索. 然而, 这些方法鲜有能够解决 RL 方法在实际场景中存在的诸多问题, 如在实际工业环境中广泛存在的众多动态障碍物的避障问题[12]. 此外, 目前的强化学习算法通常存在训练周期长, 网络收敛速度慢等问题[13-14].

针对现有 DQN 算法在应对环境模型复杂性增加时, 学习效率逐渐降低、收敛速度变慢的问题, 以及 DDQN 算法在动态场景中路径规划效率的瓶颈, 本文提出一种新的基于时序输入并融合经验池分配策略的 DDQN 路径规划算法 (TEDDQN), 其主要内容如下.

1) 提出一种新的经验回放池和经验分配策略. 该策略通过分配不同经验池中的经验, 使得智能体能够更高效地利用经验回放池中的信息. 这种新的经验分配策略能够避免智能体进行重复且低效的随机探索, 从而显著加快模型的训练速度.

2) 设计一种全新的特征输入结构. 该结构利用当前时刻以及过去两个时刻的位置信息, 通过这样一种时序机制, 使得移动机器人能够充分感知周围环境信息, 以更好地适应动态环境和陌生环境.

3) 提出一种随机动态仿真场景生成方式, 能够模拟实际工业环境中复杂多变的动态环境, 更好地评价不同算法的性能.

1 TEDDQN 算法

1.1 基于 RL 的路径规划算法

强化学习 (RL) 是一种通过反馈进行学习的机器学习方法, 主要应用于序列决策问题, 其最终目标是学习一种行动策略, 以获得最大累积奖励. 其流程示意图如图 1 所示.

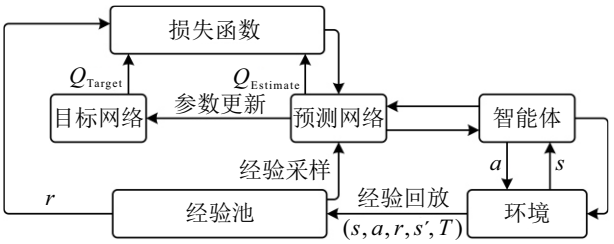


图1 基于 RL 的路径规划示意图

由图 1 可见, 每个训练周期均会生成一个新的随机模拟环境, 其中包含多个机器人, 每个机器人作为一个独立的智能体, 互不干扰地对环境进行探索. 智能体通过执行动作与环境互动, 如向不同方向移动. 新的动作以及动态障碍物的位置更新使得智能体的状态从s转换至s', 环境会反馈相应的奖励r和是否结束的状态标志T. 每次行动前后的相关信息均会被作为经验信息进行存储, 每段经验包含 5 组数据(s,a,r,s',T), 以积累各种交互经验.

在训练阶段, 从经验回放存储器中随机抽取特定批次的经验信息. 损失函数用于计算预测网络的目标网络Q值间的差异. 网络参数根据损失函数的结果进行更新. 预测网络的权重参数会在特定周期对目标网络的权重参数进行更新. 通过不断循环的经验收集和网络训练, 网络逐渐学会在环境中进行路径规划的有效策略[15].

1.2 DDQN 算法

在标准的Q-学习 (Q-Learning) 中, 低维离散状态和动作空间可使用Q表存储Q值, 但是, 面对高维视觉数据 (如图像像素) 时, Q表不再适用. 此时, Q表更新问题可转化为函数拟合问题. 深度神经网络 (CNN) 能够自动提取复杂特征, 因此可与Q-学习结合, 得到近似的Q函数, 即 DQN. 然而, 在 DQN 算法中, 动作值的随机误差会导致目标Q值被高估, 进而引发训练波动或难以收敛. 为了减少高估, DDQN 算法将目标Q值的动作选择与评估解耦, 采用两个神经网络: 一个用于估计动作值, 另一个用于选择动作. 预测网络估计目标网络的最大动作值并选择实际动作, 从而消除Q值高估问题. 目标值更新公式如下所示:

$$y_i = r + \gamma Q(s', \arg\max_{a'} Q(s', a; \theta); \theta^-). \quad (1)$$

其中: y_i为目标值, γ为学习率, $Q(s', a; \theta)$为预测网络输出的Q值, θ为预测网络的权重参数, θ^-为目标网络的参数. 方法的具体实施流程可概括如下: 每轮训练均会初始化随机环境, 再从经验回放缓冲区随机抽取一批样本, 计算由预测网络得到的Q值, 然后, 计算这些Q值与目标网络的Q值间的差异并更新神经网络的参数[16].

1.3　时序输入

目前的局部路径规划算法通常将智能体附近的栅格地图作为神经网络的输入. 相比于全局路径规划算法直接将整张栅格地图输入神经网络, 这种仅依赖于当前时刻智能体周围的环境信息往往是不够的. 为了适应复杂的动态环境, 本文提出一种时序输入的机制. 这种机制结合了当前时刻t以及过去的$t-1$和$t-2$两个时刻, 以智能体为中心, 半径为5的栅格地图, 特征图经一个卷积层, 然后按照深度进行拼接作为网络的输入. 同时, 网络的输入还结合了8个全局信息: 起点的横纵坐标、终点的横纵坐标、当前的横纵坐标、距终点的曼哈顿距离和方位角. 这些信息首先进行一维的展平和拼接操作, 然后通过2个隐藏层和1个全连接层, 最终输出一个动作. 网络的具体结构如图2所示.

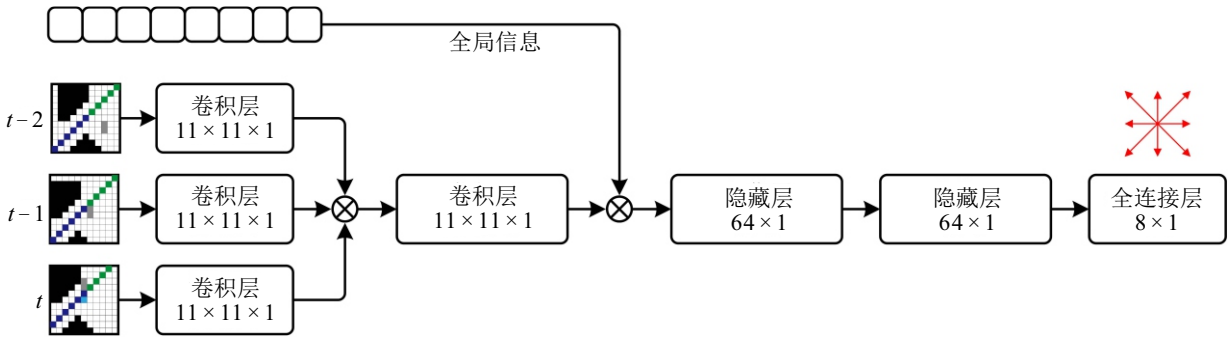


图2　TEDDQN 网络结构

这样一种时序输入的机制, 使得决策网络能够捕捉周围环境在时序上的变化, 因此, 能够更好地适应动态环境.

2　动态经验回放策略

2.1　优先经验回放

经验回放机制能够将经验元组存储在记忆库中, 并在每次学习时随机抽取一定量的记忆输入神经网络. 这种随机采样机制的引入能够在一定程度降低经验间的相关性, 有助于加速模型网络的收敛, 从而更准确地评估价值函数. 然而, RPP 的决策过程可通过如图3所示的抽象模型来描述. 这个模型将智能体状态抽象为两种情形: 一种是以“成功”作为结局, 另一种以“失败”作为结局. 当智能体采取“错误的”动作时, 探索会立即终止 (由红色虚线箭头表示); 而采取“正确的”行动则需要经历n个状态序列 (由绿色实线箭头表示). 这个抽象模型突显了智能体需要经历指数级别的随机步骤, 才能遇到第1个“成功”的奖励. 这也意味着罕见的成功隐藏于大量高度冗余的失败案例中. 因此, 传统的基于均匀随机抽样的经验回放机制虽然能够缓解经验间的相关性, 但是, 其效率是极其低下的. 为了解决这个问题, 本文使用一种优先经验回放的方法, 旨在为每个批次的经验数据用一种重要程度的参数度量, 重要程度越高的经验批次, 越能够获得更大概率的采样机会.

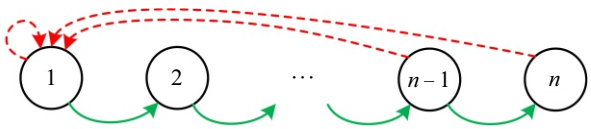


图3　RPP 决策过程的抽象模型

通常, RL 中使用一种基于 TD-error 指标 (用δ表示) 和均匀分布采样的随机采样方法来衡量每个经验样本重要性, 其中 TD-error 定义为

$$\delta = r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \theta); \theta^-) - Q(s, a; \theta). \quad (2)$$

这样可以实现使用一种指标度量每个经验批次的权重等级, 权重等级越高的经验批次越易被采样. 然而, 这种方法存在一些缺陷. 每次在网络更新后均需要更新所有经验池中每个经验批次的 TD-error, 且排序算法需要消耗大量的时间. 因此, 为了进一步提高经验样本采样的效率, 本文使用一种使用求和树 (sum tree) 的优先采样方法, 这种方法可以加快速度, 避免每次抽样均需要对每个经验批次的样本进行排序, 以实现快速地按照优先级的经验样本采样[17].

2.2　动态经验分配策略

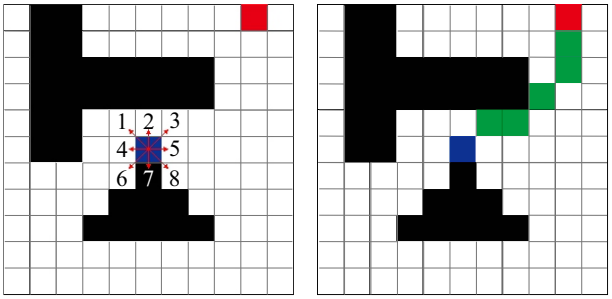
仅依靠这样一种优先经验回放策略仍然是不够

的. 在 RL 算法训练初期, 智能体通常进行大量无规律的随机探索, 这种探索虽然增强了智能体对于环境的交互感知, 但是, 这将会产生“大量”无用的经验样本, 使得网络收敛速度较慢; 在算法训练后期, 由于“探索性”经验样本的稀缺性, 大量“优质”的经验样本被反复取样, 限制了更多随机探索的可能, 网络陷入过拟合状态, 极大影响了神经网络的进一步训练, 算法陷入准确率瓶颈. 因此, 本文还提出了一种全新的经验分配策略, 这个策略将经验池划分为 3 个独立的动态经验池来优化经验采样的过程, 具体包括初始经验池、基础经验池和专家经验池. 这 3 个经验池的设计允许人类以更直观的方式来引导模型的训练过程, 从而显著加快模型的收敛速度.

1) 初始经验池: 在训练的初期阶段, 通过初始经验池存储智能体在探索环境时获得的经验. 这个阶段, 智能体的行为主要是探索性的, 因此, 这些经验对于学习基础行为模式和理解环境的基本动态至关重要. 这个阶段通常具有较高的贪心值以促进智能体进行更大范围地探索. 这样, 在训练后期, 贪心值较小时模型仍然能够具备一定的随机环境探索能力, 使得算法能够避免陷入过拟合, 突破精度瓶颈.

2) 专家经验池: 为了解决训练初期动态环境场景中失败率过高的问题, 本文提出一种基于 A*算法指导下的方法. 专家经验池储存了某一状态下依靠 A*算法指导下的一个批次的经验样本(s, a, r, s', T).

图 4 (a) 展示了一个示例状态, 在当前状态下智能体有 8 种可以选择的动作. 普通的贪心策略会随机选择这 8 种动作中的一种. 然而, 一旦选择动作 7 将会直接导致智能体与障碍物产生碰撞, 另外, 动作 1、动作 4、动作 6、动作 8 显然也不会对智能体抵达目的地产生有益影响. 此时, 假设当前环境为静态环境, 通过 A*算法建立一条当前状态到终点的最短路径, 如图 4 (b) 绿色色块所示 (这条路径不是智能体实际运动的路径). 基于 A*算法生成的结果可获知 A*算法推荐智能体选择动作 3. 此时, 智能体将选取动作 3. 环境根据选择的动作a更新s', 计算奖励函数r. 最后, 这份经验样本批次(s, a, r, s', T)将被放入专家经验池.



(a) 智能体动作选择示意图 (b) A*规划算法给出的示意路径

图4 A*算法引导生成的专家经验策略示意图

这种通过启发式搜索确定智能体的下一个动作能够避免智能体在训练初期过多的无意义探索, 极大地加快训练初期算法训练的效率.

3) 基础经验池: 随着训练的进行, 智能体也将开始学习更复杂的行为模式. 基础经验池用于存储智能体整个训练流程的经验数据, 其中每个批次的经验均按照优先经验回放的方法计算出各自的优先级. 基础经验池是训练过程中最重要的经验池, 对于网络的训练起着决定性作用.

智能体在环境中会随机选择下一个动作生成的方式, 主要包括强化学习网络推理、A*算法指导和完全随机, 其中后两种方法被选择的概率会随着网络训练周期而不断衰减. A*算法指导情形下的经验样本会放入专家经验池, 其他两种方法的经验样本会放入初始经验池和基础经验池. 初始经验池达到存储上限后, 将不再放入新的经验样本. 基础经验池中的经验样本会按照优先经验回放策略实时更新.

这 3 个经验池中的经验样本在经验回放阶段的抽样占比将随着训练周期不断变化, 其比例变化如图 5 所示. 通过调节不同经验池的经验分配策略, 智能体能够更有效地从不同经验池中的经验进行学习, 这种方法也允许人类训练者能够更好地监控和指导训练过程. 这种方法不仅提高了训练效率, 还有助于生成更鲁棒和更有效的策略.

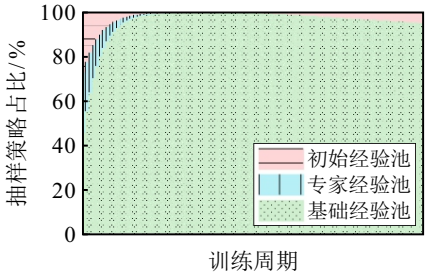


图5 经验分配策略

3 智能体与随机动态环境

3.1 状态空间

目前, 大多数随机地图的生成方法通常只包含静态随机地图环境, 然而, 这对于模拟实际复杂多变的动态工业环境是不够的. 基于此, 本文提出一种全新的动态随机地图生成方法. 这个地图以网格法作为基础, 地图内随机分布不同数量、大小的静态和动态障碍物. 动态障碍物主要包括两种随机的运动方式: 第 1 种是通过一组随机确定的起点和终点来控

制障碍物直线往复运动; 第 2 种则是通过一个随机生成的虚拟圆, 使得障碍物按照其做圆周运动.

3.2　动作空间

基于栅格地图的机器人运动状态通常采用 8 个基本动作来定义, 其动作空间包括$A=\{$右, 上, 左, 下, 左上, 右下, 右上, 左下$\}$. 这些动作定义了机器人在网格环境中的基本移动方向.

3.3　奖励函数

在 MDP 框架中, 奖励函数是评估智能体行动价值的关键组成部分. 传统的研究大多采用一种简单的奖励机制: 当 RPP 接近目标区域时, 给予大量奖励; 与障碍物碰撞时, 则给予大量惩罚. 然而, 这种机制在其他状态下奖励值均为 0, 易导致 RL 算法中奖励稀疏问题.

为了解决这一问题, 本文设计一个新的奖励函数, 其设计考虑了多个因素以更全面地评估行动的价值. 具体而言, 这个分段奖励函数包含以下几种情形:

$$r=\begin{cases}-\mathrm{d}s-\dfrac{\mathrm{step}\times\sigma}{D\times 1.5}-\mathrm{d}r\times\xi(1-\varepsilon), & \text{otherwise};\\ -100, & \text{collison};\\ -100, & \mathrm{step}>\mathrm{AL}\times 1.5;\\ +100, & \text{reach}.\end{cases}\tag{3}$$

其中: D为机器人从起点到终点的曼哈顿距离; 在此, 本文采用阈值 1.5 作为最大行动次数, 超过该运行次数的智能体将被惩罚; ε为贪心值; ξ为一个比例系数, 通常取 0.2; $\mathrm{d}s$为距终点的曼哈顿距离变化量, 其计算方法为

$$\mathrm{d}s=|x_1-x_2|+|y_1-y_2|;\tag{4}$$

$\mathrm{d}r$为机器人运动方向相对于上一时刻运动方向的角度的变化量; step为当前智能体已运动的步数; σ为路径增益, 当智能体斜向运动时, 取值为$\sqrt{2}$. 不难看出, 这个奖励函数针对超过特定步长以及碰撞存在巨大的惩罚; 对于抵达目标的智能体进行大量地奖励. 为了防止智能体陷入局部最优, 引入比例函数$\dfrac{\mathrm{step}\times\sigma}{D\times 1.5}$, 使得智能体能够更快地抵达目标位置. 同时, 为了防止智能体出现路径震荡, 使用$\mathrm{d}r$指标进行惩罚. 在训练初期过于重视$\mathrm{d}r$易导致网络陷入局部最优, 因此, 加入贪心值进行限制, 使得这一指标主要在网络训练的后期发挥作用.

通过这样一个奖励函数, 实现了同时对路径长度和方向变化次数的优化, 从而改善移动机器人的路径规划的质量.

4　网络训练

本节主要介绍所提出 TEDDQN 算法的训练方法, 同时, 比较所提出网络与其他著名 RL 网络的收敛性和学习效率. 这些算法以及所提出 TEDDQN 算法均采用相同的训练参数, 在如表 1所示的动态随机环境进行训练. 其在训练阶段成功率的变化情况如图 6 所示. 其中: 横坐标为训练周期, 纵坐标为该算法路径规划成功率, 其结果已经过平滑处理.

表1　随机环境的参数设置

参数类型	情形1	情形2	情形3
栅格地图大小	100	100	100
静态障碍物数目	自变量	20	20
静态障碍物大小	[1, 5]	[1, 5]	[1, 5]
动态障碍物数目	–	自变量	20
动态障碍物大小	–	[1, 5]	[1, 5]
动态障碍物运动速度	–	1	自变量
自变量变化范围	[0, 30]	[0, 30]	[1, 3]

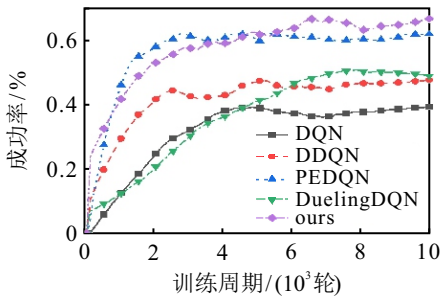


图6　训练阶段成功率变化曲线

由图 6 可知: 在训练初期, 智能体对环境进行了广泛地探索; 然后, 智能体通过利用从探索中获得的经验知识进行学习, 使得网络逐步走向稳定. 不难看出, 尽管 DQN 与 DDQN 算法趋势相似, 但是, DQN 表现出较大的训练波动, 且最终效果存在显著差距, 这也表明 DQN 算法的过分估计带来了巨大的影响. PEDQN 和所提出 TEDDQN 算法得益于优先经验存放策略, 训练效果显著优于其他算法, 且优先经验回放的方法能够更快地使得网络收敛, 没有使用这种优先经验回放的方法均需要更多的探索步骤才能使得奖励值趋于稳定. 具体而言, 其他算法几乎在训练阶段 50% 才基本稳定, 然而, 使用优先经验回放方法能够在训练阶段的前 25% 便已基本实现了收敛, 相比于 DDQN 收敛效率提升了近 50%.

值得注意的是, 所提出 TEDDQN 得益于初始阶段分配的专家经验策略, 在训练初期极大地改善了网络的收敛效率, 并在网络训练的后期增大了初始经验池中经验策略分配的比例, 使得算法能够避免陷入过拟合, 突破了算法瓶颈. TEDDQN 相比于 PEDQN 有着更高的上限. 通过对比还可以发现, 在

动态的随机环境下, 其他 4 种方法最终稳定的成功率为 39%、45%、62% 和 47%, 而 TEDDQN 算法稳定在 66%. 这些对比结果表明, TEDDQN 具有较高的学习效率、更好的收敛性.

5 路径规划算法的评价

路径规划算法的有效性通常受到多种因素的影响, 包括地图的大小、障碍物的数量和复杂程度等. 评估路径规划算法的质量也不是基于单一标准, 常见的评估标准包括平均路径长度 (AL)、平均算法运算时间 (AT)、平均路径转向角度 (AR) 以及路径规划的成功率 (ASR). 其中: AR 指的是智能体从起点到终点经过的栅格数, AT 勇于评价智能体每进行一次动作决策所需要花费的平均时间, ASR 为智能体从起点到终点运动过程中角度变化绝对值的累积值.

本文设计了 3 组不同的实验. 情形 1 设置了固定的栅格地图场景 (500 × 500), 以静态障碍物的数目作为自变量; 情形 2 以动态障碍物的数目作为自变量; 情形 3 以动态障碍物的运动速度作为自变量. 具体设置如表 1 所示, 其中动态障碍物的运动速度是指每个时刻移动的栅格长度. 另外, 当障碍物的面积占环境地图的面积超过 90% 时, 将终止随机生成新的障碍物. 每次自变量的变化均将随机生成 10 种随机场景, 测试结果取这 10 次实验的平均值. 本文对比了 DQN、DDQN、PEDQN、A*算法[18]、RRT 算法[19]、APF 算法[20]等在内的多种著名算法. 实验的结果如表 2 和表 3 所示. 表 2 为情形 1 自变量变化范围内所有指标的平均结果. 表 3 为在情形 2 和情形 3 动态环境条件下各指标的平均结果.

表2 不同算法在静态环境测试结果

算法	AL/栅格数	AT/ms	AR/(°)	ASR/%
DQN	242.46	1.20	2395.83	89
DDQN	236.76	1.31	2161.47	91
PEDQN	216.79	1.38	1887.33	96
Dueling DQN	262.43	1.14	1978.79	92
A*	179.71	7.07	882.09	100
RRT	436.43	2.71	13140.96	100
APF	225.35	1.07	1257.39	100
TEDDQN-S	223.67	1.10	1872.46	95
TEDDQN	224.14	1.12	1980.79	95

表3 不同算法在动态环境测试结果

算法	情形2				情形3			
	AL/栅格数	AT/ms	AR/(°)	ASR/%	AL/栅格数	AT/ms	AR/(°)	ASR/%
DQN	251.12	1.10	2970.12	81	242.02	1.28	1874.24	71
DDQN	241.52	1.24	2729.31	83	249.95	1.29	2823.20	72
PEDQN	252.36	1.27	2418.14	89	262.91	1.25	2316.39	87
Dueling DQN	285.73	1.15	2772.71	85	246.17	1.10	2638.83	79
A*	192.96	6.89	902.92	100	200.10	7.08	930.37	91
RRT	452.32	3.75	13580.22	100	451.32	4.68	13130.22	92
APF	230.89	1.06	1510.34	94	235.92	1.01	1564.08	87
TEDDQN-S	260.48	1.04	1964.54	88	234.16	1.17	2116.45	84
TEDDQN	261.23	1.18	2333.92	91	240.73	1.25	2322.07	88

通过对比表 2 中 DQN 算法与 DDQN 算法的各项指标, DDQN 在多个评价维度上存在显著的优越性. 采用经验回放方法的 PEDQN 和 TEDDQN 算法无论是从路径规划的成功率, 还是从减少平均路径长度和转弯角度方面, 其综合性能均优于传统的算法, 其中 PEDQN 与 TEDDQN 算法效率难分优劣. 然而, 在平均长度以及平均转弯角度的两项指标上, TEDDQN 展现出了较大优势.

表 3 进一步展示了几种算法在复杂环境中的测试结果. 情形 2 由于涉及多种状态的障碍物, 比较适合作为评价各种路径规划算法的综合环境. 表 3 结果显示, TEDDQN 在大多数条件下优于 PEDQN 算法, 其平均成功率为 91%, 高于 PEDQN 的 89%, 综合提升率达到 2.4%, 这也表明了本文对 TEDDQN 算法动态环境的优化是合理的.

情形 3 中动态障碍物的运动速度为 3, 远超智能体的移动速度. 以 A*算法和 RRT 算法为代表的传统网络, 在这种复杂的动态环境中存在被碰撞的风险, 但是, 值得注意的是, 所提出 TEDDQN 算法在这种极端复杂环境中存在一定的适应能力. 相比于情形 2 中的动态环境, 成功率仅从 91% 下降至 89%, 下降了 2%, 可以表明所提出算法针对动态环境的优化是有效的.

表 2 和表 3 中: TEDDQN 为所提出算法, TEDDQN-S 为仅以t时刻地图作为输入的方法. 由这组消融实验可知: 在静态环境中, 两者成功率均为 95%, 但是, TEDDQN-S 具备更快的推理速度; 然而, 在动态环境中, 时序机制显著提升了网络在动态环

境中的成功率, 尤其是在复杂动态环境中, 成功率从84% 提升至 88%, 这也验证了以$t-2$、$t-1$和t时刻地图作为网络输入的合理性.

图 7 为一组动态的路径规划过程. 该环境包含5 个智能体, 其位置均为随机生成, 按照所提出算法进行路径的决策规划. 这 8 张图像分别为随时间变化的 8 个特殊的时刻, 各智能体在环境中留下的轨迹.

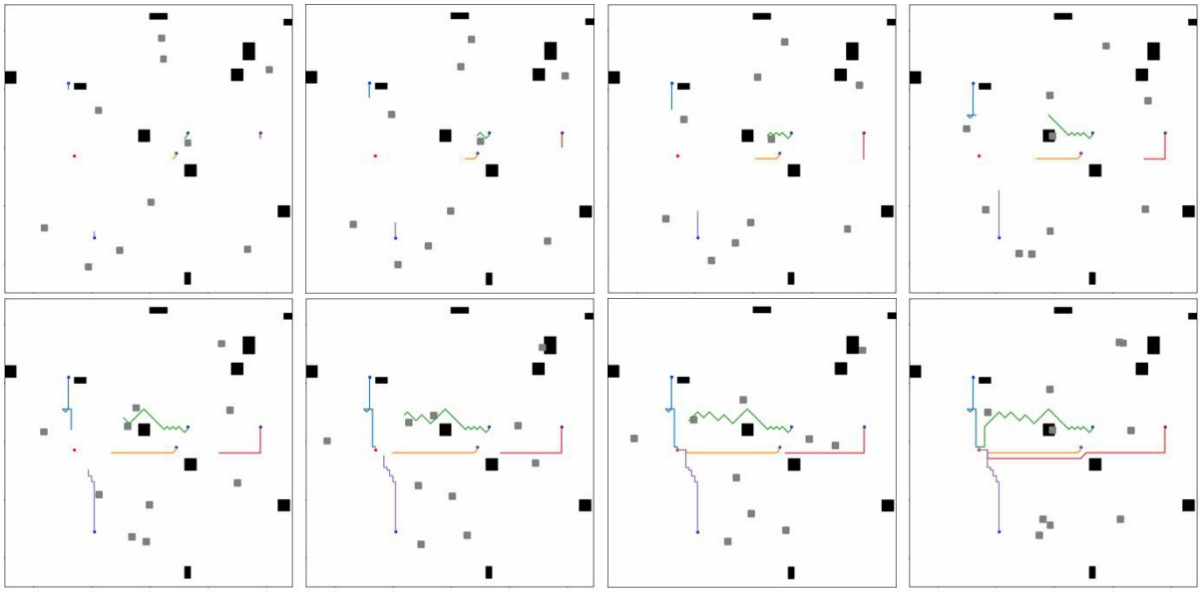
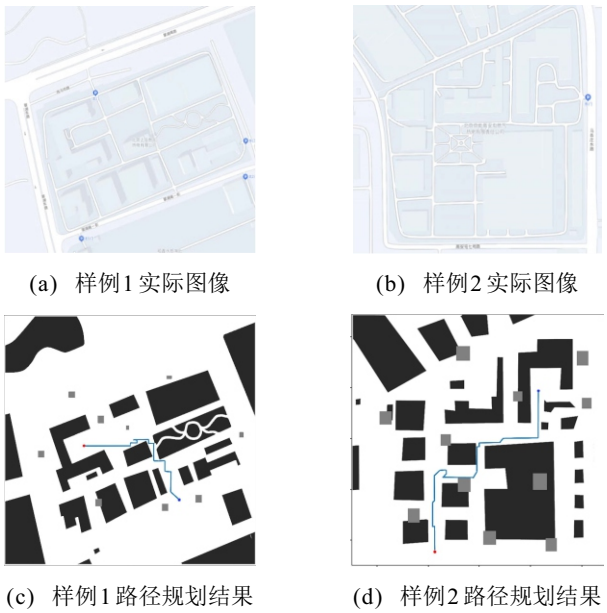


图7　动态仿真样例

由图 7 可见, 受动态障碍物影响最大的是留下绿色轨迹的智能体. 然而, 即使受到如此强烈的干扰, 智能体仍然能够抵达目标点, 验证了所提出 TEDDQN 算法的可靠性.

为了测试实际场景中的算法表现, 图 8 展示了两组实际工业环境中的地图. 其中: 灰色的色块模拟实际环境中如汽车、智能设备、操作员等动态障碍物, 蓝色的轨迹为所提出 TEDDQN 算法生成的轨迹.



(a)　样例1 实际图像　　(b)　样例2 实际图像

(c)　样例1 路径规划结果　　(d)　样例2 路径规划结果

图8　实际场景仿真样例

6　结　论

针对 RPP 问题, 本文提出了一种 TEDDQN 路径规划算法, 该方法能够实现更快速的网络训练以及更优越的动态性能. 主要内容如下.

1) 得益于一种独特的经验池分配策略, 所提出方法在复杂随机环境中对神经网络进行训练的效率优于其他各种算法. PEDDQN 在训练初期能够依靠专家经验池更快地收敛; 在训练后期能够依靠初始经验池, 改善网络的探索效率, 突破精度瓶颈, 在训练时间上相比于 DDQN 方法减少了 50%, 且最终的训练效果提高了 31%.

2) 本文设计了一种独特的时序输入结构. 该方法在动态环境中表现出较好的适应性. 动态环境实验表明, 所提出方法在平均路径长度、平均转弯角度、平均路径规划的成功率几项关键指标上均优于其他网络. 相比于 DDQN 算法, 在复杂动态环境中成功率提升了 22.2%, 综合环境中提升了 9.6%.

3) 本文提出了一种随机的动态场景生成方法, 包含随机的静态障碍物和随机运动的动态障碍物. 这种方法能够更加贴合于实际的物理场景, 使得方法能够更加真实有效地反映其在物理世界中的性能, 为路径规划算法的研究提供了一种新的思路.

参考文献 (References)

[1]　毛建旭, 贺振宇, 王耀南, 等. 电力巡检机器人路径规

划技术及应用综述[J]. 控制与决策, 2023, 38(11): 3009-3024.
(Mao J X, He Z Y, Wang Y N, et al. Review of research and applications on path planning technology for power inspection robots[J]. Control and Decision, 2023, 38(11): 3009-3024.)

[2] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.
(Zhu D Q, Yan M Z. Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7): 961-967.)

[3] Liu L X, Wang X, Yang X, et al. Path planning techniques for mobile robots: Review and prospect[J]. Expert Systems with Applications, 2023, 227: 120254.

[4] Aggarwal S, Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges[J]. Computer Communications, 2020, 149: 270-299.

[5] Lu S Y, Zhang Y, Su J J. Mobile robot for power substation inspection: A survey[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 4(4): 830-847.

[6] Chen Z Y, Xu B. AGV path planning based on improved artificial potential field method[C]. IEEE International Conference on Power Electronics, Computer Applications. Shenyang, 2021: 32-37.

[7] 孙辉辉, 胡春鹤, 张军国. 移动机器人运动规划中的深度强化学习方法[J]. 控制与决策, 2021, 36(6): 1281-1292.
(Sun H H, Hu C H, Zhang J G. Deep reinforcement learning for motion planning of mobile robots[J]. Control and Decision, 2021, 36(6): 1281-1292.)

[8] Li Z L, Luo X N. Autonomous underwater vehicles (AUVs) path planning based on deep reinforcement learning[C]. Proceedings of the 9th International Conference on Digital Home. Guangzhou, 2022: 257-262.

[9] Yi C, Qi M. Research on virtual path planning based on improved DQN[C]. IEEE International Conference on Real-Time Computing and Robotics. Asahikawa, 2020: 387-392.

[10] Yu Y, Liu Y F, Wang J C, et al. Obstacle avoidance method based on double DQN for agricultural robots[J]. Computers and Electronics in Agriculture, 2023, 204: 107546.

[11] Lv L H, Zhang S J, Ding D R, et al. Path planning via an improved DQN-based learning policy[J]. IEEE Access, 2019, 7: 67319-67330.

[12] 王贺, 许佳宁, 闫广宇. 基于深度强化学习的 AGV 行人避让策略研究[J]. 系统仿真学报, DOI: 10.16182/j.issn1004731x.joss.24-0088.
(Wang H, Xu J N, Yan G Y. Research on AGV pedestrian avoidance strategy based on deep reinforcement learning[J]. Journal of System Simulation, DOI: 10.16182/j.issn1004731x.joss.24-0088.)

[13] Xin J, Zhao H, Liu D, et al. Application of deep reinforcement learning in mobile robot path planning[C]. Chinese Automation Congress. Jinan, 2017: 7112-7116.

[14] 卢锦澎, 梁宏斌. 基于深度Q网络的机器人路径规划研究综述[J]. 传感器与微系统, 2024, 43(6): 1-5.
(Lu J P, Liang H B. Research review of robot path planning based on DQN[J]. Transducer and Microsystem Technologies, 2024, 43(6): 1-5.)

[15] de Berg M, van Kreveld M, Overmars M, et al. Computational geometry: Algorithms and applications[M]. Heidelberg: Springer Berlin Heidelberg, 2000.

[16] 董豪, 杨静, 李少波, 等. 基于深度强化学习的机器人运动控制研究进展[J]. 控制与决策, 2022, 37(2): 278-292.
(Dong H, Yang J, Li S B, et al. Research progress of robot motion control based on deep reinforcement learning[J]. Control and Decision, 2022, 37(2): 278-292.)

[17] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J/OL]. 2015, arXiv: 1511.05952.

[18] 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[J]. 系统仿真学报, DOI: 10.16182/j.issn1004731x.joss.23-1574.
(Yu D, Bao B Z, Si Y, et al. Path planning for mobile robots based on search step optimization A* algorithm[J]. Journal of System Simulation, DOI: 10.16182/j.issn1004731x.joss.23-1574.)

[19] 陈际同, 周佳加, 吴迪, 等. 基于 TD3-RRT 的特殊环境下 USV 路径规划算法研究[J]. 系统仿真学报, DOI: 10.16182/j.issn1004731x.joss.24-0622.
(Chen J T, Zhou J J, Wu D, et al. Research on USV path planning algorithm in special environment based on TD3-RRT[J]. Journal of System Simulation, DOI: 10.16182/j.issn1004731x.joss.24-0622.)

[20] 张弛, 魏巍. 基于改进人工势场法的移动机器人路径规划[J]. 系统仿真学报, DOI: 10.16182/j.issn1004731x.joss.24-0665.
(Zhang C, Wei W. Path planning for mobile robots based on improved artificial potential field method[J]. Journal of System Simulation, DOI: 10.16182/j.issn1004731x.joss.24-0665.)

作者简介

李佩哲 (2000−), 男, 硕士生, 主要研究方向为深度学习、强化学习, E-mail: 120222227224@ncepu.edu.cn;

张文彪 (1985−), 男, 副教授, 博士, 主要研究方向为智能仪表与状态监测、机器人智能感知, E-mail: wbzhang@ncepu.edu.cn.