Modeling and Simulation 建模与仿真, 2023, 12(6), 5272-5283
Published Online November 2023 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2023.126479

Hans汉斯

基于强化学习的多智能体路径规划研究与应用

陈天润[1]，高大威[2]

[1]澳门科技大学创新工程学院，澳门
[2]上海理工大学机械工程学院，上海

收稿日期：2023年9月11日；录用日期：2023年11月6日；发布日期：2023年11月14日

摘　要

研究聚焦于智能仓储中AGV的路径规划问题，构建了基于强化学习的多智能体寻路算法。每个AGV能从环境和以往经验中学习，利用不同行为产生的奖励机制，训练智能体自主选择更高效策略，以达到预定目标。本研究在ACTOR-CRITIC算法基础上加入经验回放机制并采用了中心化训练和去中心化决策的方法，以提高智能体的路径规划效率。同时，将ACTOR-CRITIC算法在智能仓的环境下进行模拟训练，验证AGV的路径规划效果。

关键词

智能仓储，多智能体强化学习，路径规划

Research and Application of Multi-Agent Path Planning Based on Reinforcement Learning

Tianrun Chen[1], Dawei Gao[2]

[1]Faculty of Innovation Engineering, Macau University of Science and Technology, Macau
[2]School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Sep. 11th, 2023; accepted: Nov. 6th, 2023; published: Nov. 14th, 2023

Abstract

The research focuses on the path planning problem of AGV in intelligent warehousing, constructs a multi-agent path finding algorithm based on reinforcement learning. Each AGV can learn from the environment and previous experience, use the reward mechanism generated by different beha-

文章引用: 陈天润, 高大威. 基于强化学习的多智能体路径规划研究与应用[J]. 建模与仿真, 2023, 12(6): 5272-5283.
DOI: 10.12677/mos.2023.126479

viors to train the agent to independently choose more efficient strategies to achieve the predetermined goal. In this research, an experience playback mechanism is added to the ACTOR-CRITIC algorithm, centralized training and decentralized decision-making methods are adopted to improve the path planning efficiency of the agent. At the same time, the ACTOR-CRITIC algorithm is simulated and trained in the environment of the smart warehouse to verify the path planning effect of the AGV.

Keywords

Intelligent Warehousing, Multi-Agent Reinforcement Learning, Path Planning

1. 引言

近年来，多智能体系统由于能够提供灵活高效的解决方案而更多地应用于智能仓储管理。它们由多个可以相互通信和合作的自主智能体组成，能够共享资源和知识来解决复杂的任务[1] [2]。然而，这对智能体的有效控制和协调提出了重大挑战。智能体之间的有效沟通对于确保他们能够协作并到达预期目标至关重要，当处理在复杂环境中运行的多个智能体时尤其如此。前人提出了智能仓库中单智能体路径规划的几种方法，如 A*搜索算法和 Dijkstra 算法[3]已被广泛使用并被证明是有效的。

终身多智能体路径规划是一个过程，使一组智能体能够在环境中移动时动态生成和更新路径。通过结合人工势场和分层规划等技术，智能体可以协作实时生成和更新路径，从而更有效地对环境变化做出反应。这种方法已被证明在各种应用中都是有效的，包括机器人、自动驾驶车辆和自主探索[4] [5]。然而它具有众多限制，首先，对多个智能体寻找最佳路径所涉及的计算复杂性非常高，随着智能体数量的增加，问题的解决难度呈指数级增长。其次，智能体之间缺乏沟通和协调，智能体对环境状态缺乏整体感知，这可能会严重影响规划路径的性能和可行性。以上问题导致传统的多智能体寻路算法难以高效的应对大型仓库环境或环境频繁变化的动态场景。目前，行业迫切需要在复杂场景和易变环境下使用先进的大面积导航方法。

强化学习是控制智能仓库中多个智能体的强大工具[6] [7]。智能体能够从环境和经验中学习，以最大限度地增强智能体的路径规划效果。通过利用智能体不同行为产生的奖励和惩罚，可以训练智能体优化其行为，以达到预期目标[8]。强化学习使用试错方法，允许智能体探索不同的策略，直到找到最成功的策略。这种类型的学习为智能体提供了从错误中总结经验并根据环境反馈调整其行为的机会[9]。此外，强化学习可用于教导智能体以无监督的方式工作，使它们无需人工干预即可学习。

本研究采用基于 ACTOR-CRITIC 的强化学习技术来训练智能体就其路径规划做出智能决策，加入了经验回放机制[10] [11]并采用了中心化训练和去中心化决策[12]的方法，以提高 AGV 的路径规划效率。同时，将 ACTOR-CRITIC 算法在仓储环境下进行模拟训练，验证本次研究中 AGV 之间的协作效果。

2. 多智能体强化学习算法

2.1. 系统模型

在路径规划领域，多智能体强化学习算法(MARL)的使用在提高运行效率和优化资源分配方面具有巨

大潜力，这些算法的一个关键组成部分是马尔可夫决策过程(MDP) [13]。MDP 作为动态环境中决策问题建模的数学框架，能够捕获此类环境的不确定性和动态性质，适合智能仓储场景。

MDP 的核心由一组状态空间、动作空间、状态转移函数、奖励函数、动作价值函数等组成，记为元组(S, A, P, R, Q)。本研究主要探索智能仓储中 AGV 的路径规划问题，设仓储环境中有 n 个智能体(AGV)。在每一时刻 T，环境输出一个全局状态 $s_t \in S$，和 n 个观测状态$\left[o_t^1, o_t^2, \cdots, o_t^n\right] \in O$。其中，$S$ 代表全局状态空间，O 代表全局观测空间。每个智能体根据自己的观测 o_t^i 选择一个动作 $a_t^i \in A$，其中 A 表示每个智能体 $i \in \{1, \cdots, n\}$ 的动作空间。系统根据状态转移函数 $P\left(s_{t+1} \middle| s_t, a_t\right)$ 对所有智能体执行联合动作 $a_t \in A$。智能体得到各自的奖励 r_t^i，全局奖励 $R = \left[r_t^1, r_t^2, \cdots, r_t^n\right]$。在这个过程中，每个智能体都试图增加自己的个人奖励以使全局奖励增大。动作价值函数 $Q(s, a)$ 会根据 T 时刻的奖励 r、状态 s 和动作 a 输出一个评价智能体的动作 a 的值。

2.2. Actor-Critic

MARL 提供了一种有效的方法来解决多智能体环境中遇到的复杂控制问题。Actor-Critic [14]方法是一种常见的 MARL 算法，它汇集了基于策略和基于值的方法的优点。

Actor-Critic 算法由两个主要部分组成：Actor 和 Critic。Actor 负责根据当前情况选择动作，而 Critic 则检查所选择的动作并向 Actor 提供反馈。这种反馈使智能体能够随着时间的推移通过更新其策略来增强其决策过程。通过利用这种双组件架构，Actor-Critic 算法可以有效地平衡决策和反馈，并在多智能体环境中学习最优策略。

在智能仓储的背景下，应用多智能体 Actor-Critic 算法来协调不同智能体(例如机器人或自动车辆)的动作，以实现高效可靠的仓库运营。Actor-Critic 算法允许每个智能体学习自己的策略，同时考虑其他智能体的行为对整体系统性能的影响。这种协作学习过程使智能体能够实时适应和优化其行为，从而改善任务分配、资源利用率和整体仓库效率。总体而言，基于 Actor-Critic 的多智能体强化学习算法在彻底改变智能仓库控制机制方面具有巨大潜力。

策略网络 $\pi\left(a \middle| s; \theta\right)$ 相当于一个参与者，它根据状态 s 做出动作 a。价值网络 $q(s, a; \omega)$ 相当于法官，对行动者的表现进行打分，评价状态 s 下动作 a 的质量。Actor 和 Critic 之间的关系如图 1 所示。

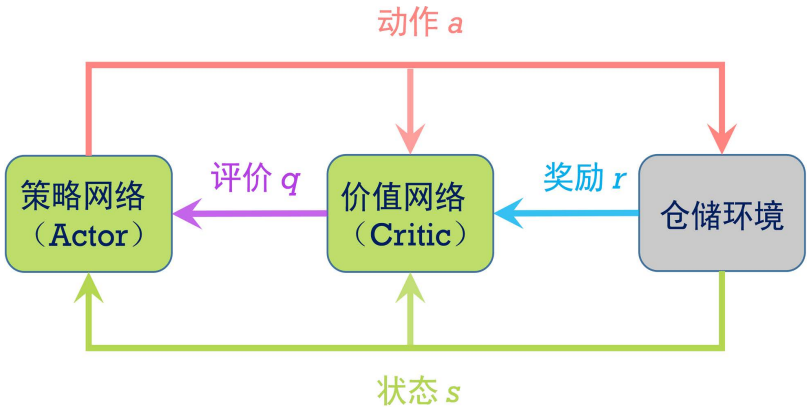


Figure 1. Actor-Critic algorithm framework
图 1. Actor-Critic 算法组织框架

Actor 使用一个神经网络近似策略函数。这种神经网络称为策略网络，记为 $\pi\left(a \middle| s; \theta\right)$，其中 θ 表示策略网络参数。策略网络的输入是当前环境的状态 s。状态 s 经过卷积层、全连接层和 softmax 后，输出智

能体将要执行的动作以及该动作的概率。这时，策略网络会随机采样一个动作并执行。图 2 描述了策略网络的结构。

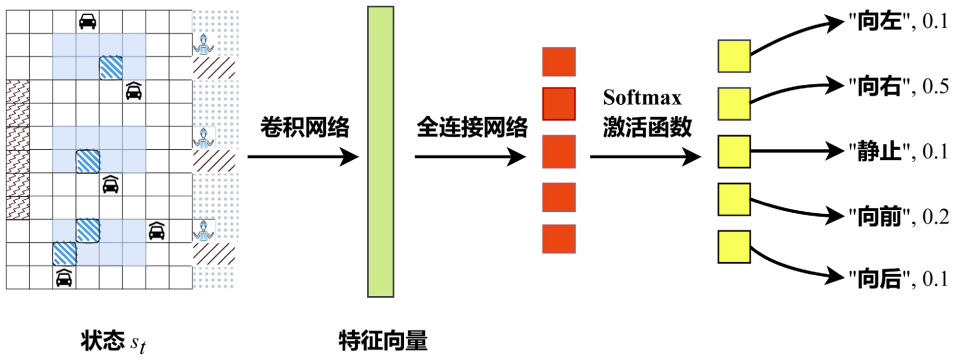


Figure 2. Policy network
图 2. 策略网络

Critic 用一个神经网络来近似动作价值函数 $Q_\pi(s,a)$。这种神经网络被称为价值网络，记为 $q(s,a;\omega)$，其中 ω 表示价值网络参数。价值网络接收状态 s 作为输入并输出与每个动作相关的值。图 3 描绘了价值网络的组织结构。

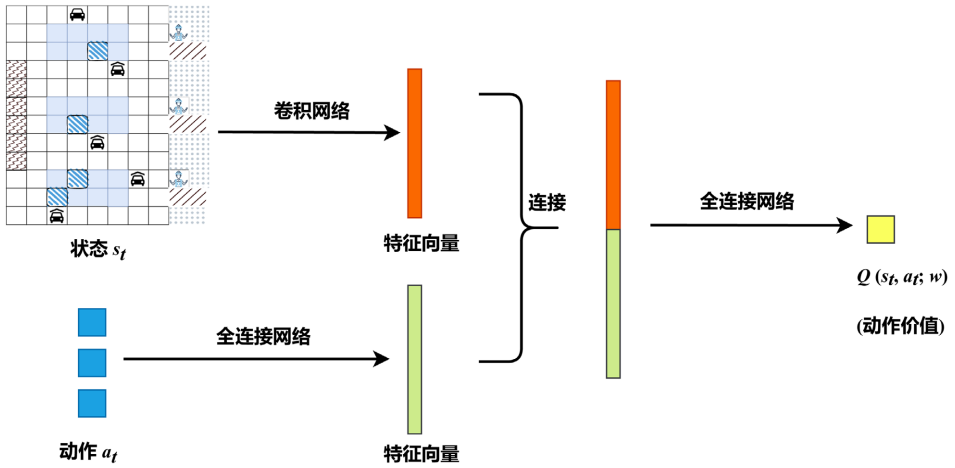


Figure 3. Value network
图 3. 价值网络

2.3. 训练

Actor-Critic 算法中的两个神经网络不同时工作。如图 4 所示，每个 Agent 上分别定义了一个策略网络。中央控制器中有多个价值网络，每个价值网络对应一个策略网络。当训练算法时，中央控制器可以查看所有智能体的观察、动作和奖励，并把每一轮的 o^i, r^i, a^i 依次存入经验回放数组。它被称为集中训练。

训练结束后，中央控制器上的价值网络被屏蔽，每个智能体根据策略网络进行决策，称为去中心化执行。如图 5 所示。在执行阶段，第一步，每个智能体从各自环境观察环境状态 s。每个智能体都有一个独特的策略网络，可以生成概率分布。第二步，每个智能体根据自己的概率分布进行采样，得到动作 a 并执行。此时环境已经发生变化，重复步骤一和二，直到智能体到达目标位置。

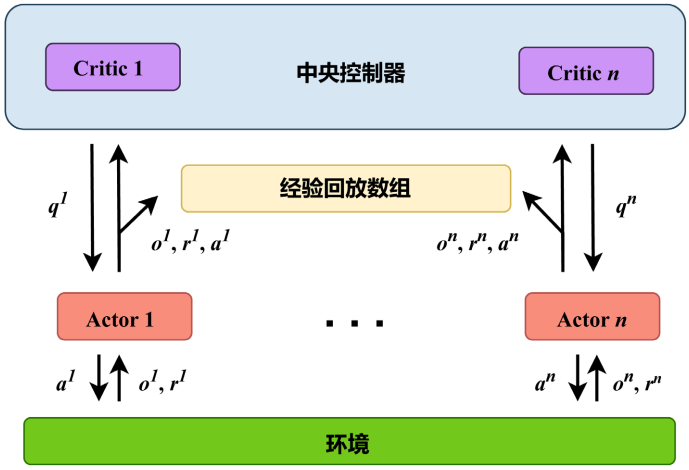


Figure 4. Centralized training
图 4. 中心化训练

Figure 5. Decentralized execution
图 5. 去中心化执行

下面概述了 Actor-Critic 算法的训练过程。中央控制器从经验回放数组中随机抽取一个四元组 $\left(s_t,r_t,a_t,s_{t+1}\right)$，令当前策略网络变量为 θ_{now}，价值网络变量为 ω_{now}。执行以下步骤并更新参数 θ_{now} 和 ω_{now}。

1) 观察当前状态 s_t，根据策略网络做决策：$a_t\sim\pi\left(\cdot\middle|s_t;\theta_{now}\right)$，并让智能体执行动作 a_t。
2) 从环境中观察到奖励 r_t 和新状态 s_{t+1}。
3) 根据策略网络做决策：$\tilde{a}_{t+1}\sim\pi\left(\cdot\middle|s_{t+1};\theta_{now}\right)$，但不允许智能体执行动作 $\tilde{a}_{t+1}$。
4) 让价值网络打分：

$$\hat{q}_t\sim q\left(s_t,a_t;\omega_{now}\right)\text{和}\hat{q}_{t+1}\sim q\left(s_{t+!},\tilde{a}_{t+1};\omega_{now}\right)$$

5) 计算 TD 目标和 TD 误差：

$$\hat{y}_t=r_t+\gamma\cdot\hat{q}_{t+1}\text{和}\delta_t=\hat{q}_t-\hat{y}_t$$

6) 更新价值网络：

$$\omega_{new}\leftarrow\omega_{now}-\alpha\cdot\delta_t\cdot\nabla_w q\left(s_t,a_t;\omega_{now}\right)$$

7) 更新策略网络：

$$\theta_{new}\leftarrow\theta_{now}-\beta\cdot\hat{q}_t\cdot\nabla_\theta\ln\pi\left(a_t\middle|s_t;\theta_{now}\right)$$

3. 仿真

3.1. 仓储模型

为了便于理解智能仓库的操作流程，本研究总结了一个典型仓库的环境配置。该仓储系统的整个过

程分为 5 个阶段。在第一阶段，当系统启动时，所有 AGV 都被分配货物的位置和拣选站的位置。第二阶段，AGV 从充电区(起点)移动到目标货架底部并将其举起。第三阶段，AGV 搬运货架至拣选工位入口。第四阶段，每位员工从货架上挑选产品。第五步，当拣选工作完成后，AGV 将货架带到可用的货架位置并将其放下。AGV 重复阶段二至五，直到系统停止。本次研究只涵盖第二阶段中的 AGV 路径规划。智能仓储系统的框图如图 6 所示。

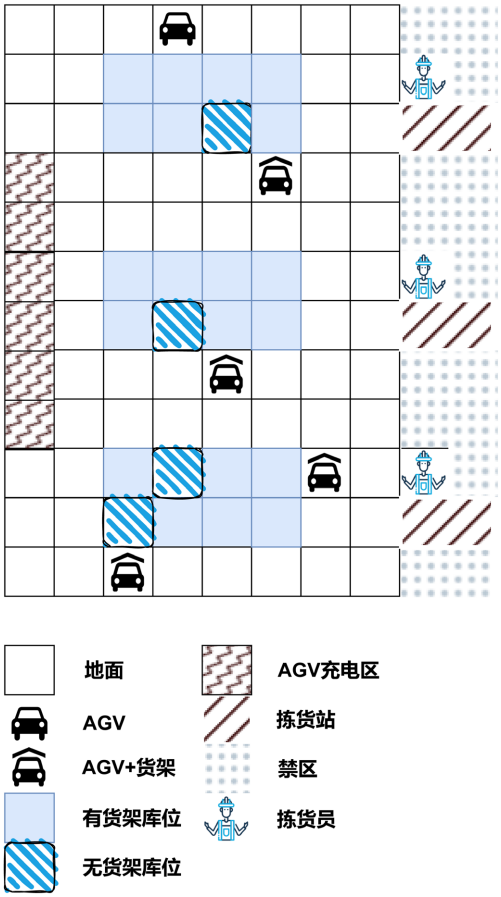


Figure 6. Intelligent warehousing environment
图 6. 智能仓储环境

本研究基于上述假设环境。使用以下四种情形进行模拟。他们是 5 个智能体和一张 2 × 2 的地图；10 个智能体和一张 2 × 2 地图；5 个智能体和一张 5 × 5 地图；10 个智能体和一张 5 × 5 地图。其中，每个智能体占用面积为 0.15 × 0.15，每个货架占用面积为 0.08 × 0.08。将仓储作 XY 平面图，使整个仓储区域位于第一象限，智能体的起始区域(AGV 充电区)为从 $x = 0$ 到 $x = 0.3$ 的所有区域内。

在训练策略网络和价值网络时，环境根据智能体在每个时间步长选择的动作返回全局奖励和单个奖励。单个奖励设定如下：

$$r_t^i = \begin{cases} -d_t^i, & d_t^i \neq 0 \\ -d_t^i + 2, & d_t^i = 0 \end{cases} \tag{1}$$

其中 d_t^i 是智能体 i 在第 t 个时间步长到目标位置的距离。为了让每个智能体在每个时间步中获得良好的奖励，智能体必须选择向目标位置移动的动作。另外，为了鼓励智能体尽可能向目标前进，设置了额外

的激励奖励，即如果智能体到达目标架，则奖励额外增加 2。全局奖励确定如下：

$$r_t = \sum_{i=0}^{n} r_t^i \tag{2}$$

其中 n 是环境中智能体的数量。全局奖励是对所有智能体动作的全面评估，它可表述为：每个智能体与目标书架之间的距离之和的负数。训练神经网时，中央控制器会根据全局奖励选择每个智能体的动作。在某些情况下，少数智能体的奖励减少会导致全局奖励增加。这种奖励设计在提高智能体的泛化性能方面是有利的。

仿真设定训练次数为 5000，神经网络隐藏层维度为 64，策略网络与价值网络的学习率均为 1e-2。算法折扣因子设为 0.95。

3.2. 仿真结果

本研究使用两个指标来衡量算法的有效性：(1) 累积奖励，即系统中所有智能体的累积奖励回报。(2) 到达目标点时，整个运动期间与目标点之间的平均距离。本研究还使用不同区域和不同智能体数量的四张地图进行仿真演示，以证明所用方法的可行性和泛化性。

3.2.1. 累积奖励

累积奖励与训练次数曲线如图 7~10 所示，随着训练次数的增加，算法的累积奖励逐渐收敛。依据本研究设置的奖励函数为每个智能体与目标书架之间的距离之和的负数。因此，当奖励收敛到设定的奖励值时，就可以确认智能体成功找到了目标。

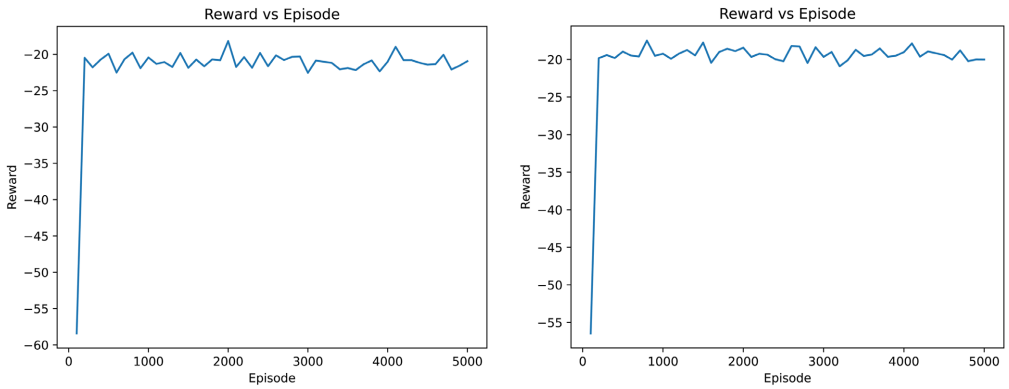
当智能体数量相同时，累积奖励在较大地图环境中比在较小地图环境中收敛得更慢。这是因为地图尺寸变大，智能体需要走得更远才能找到各自的目标，因此需要更多的训练才能使曲线收敛。

当地图大小相同时，同时训练的智能体越多，累积奖励收敛的速度就越慢。这是因为同时参与的智能体越多，智能体之间就越容易出现阻塞，从而越难达到目标。

当有更多的智能体同时运行时，最终的累积奖励更大，因为更多的智能体获得额外的奖励值。

3.2.2. 平均距离

智能体到目标的平均距离与训练次数的关系如图 11~14 所示，随着训练次数的增加，相同时间步内智能体与目标之间的平均距离逐渐趋于 0。可以证实，随着训练次数的增加，系统的性能随之提升，智能体的策略也越来越好，最终能够在规定的时间内成功找到目标。当智能体数量增加或地图尺寸增大时，需要更多的训练时间才能使智能体完美找到目标。随着地图尺寸的增大，曲线的收敛值会更大，这是因为在大地图下，智能体往往需要移动的更远才能到达目标。



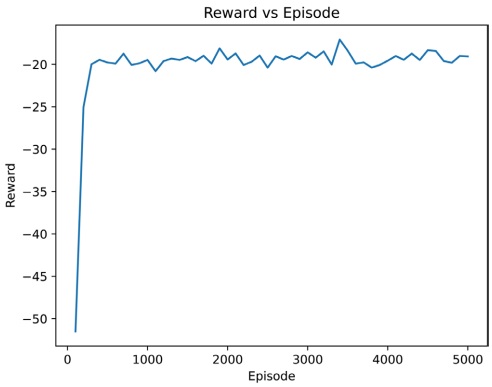


Figure 7. Small layout with 5 agents
图 7. 5 个智能体与小空间

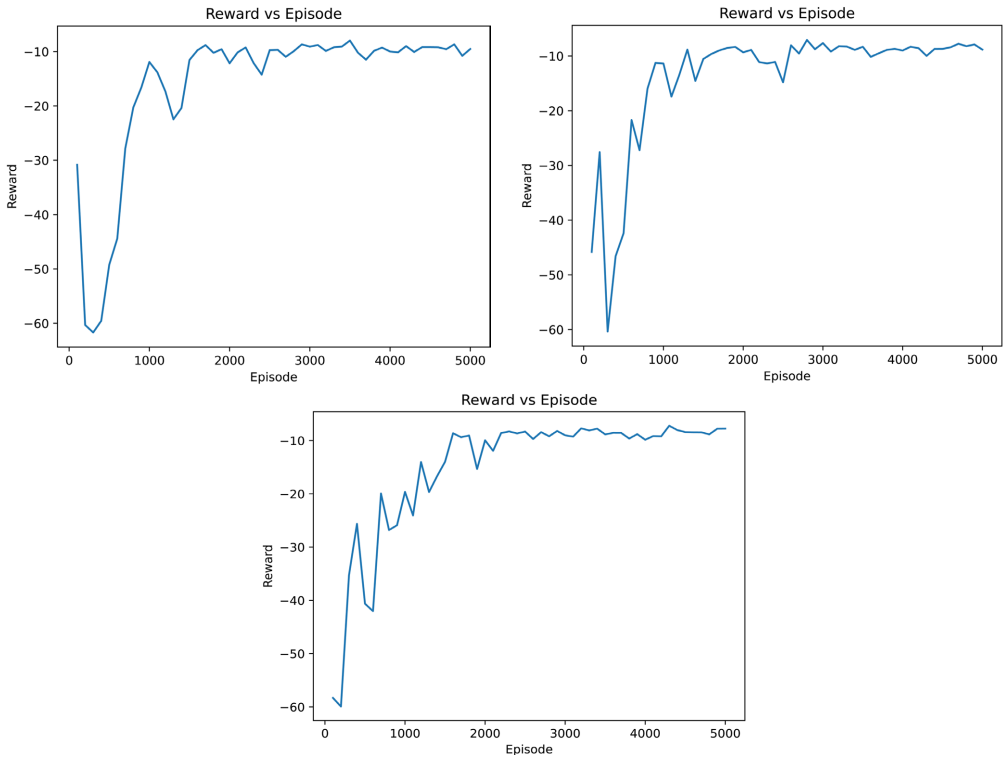
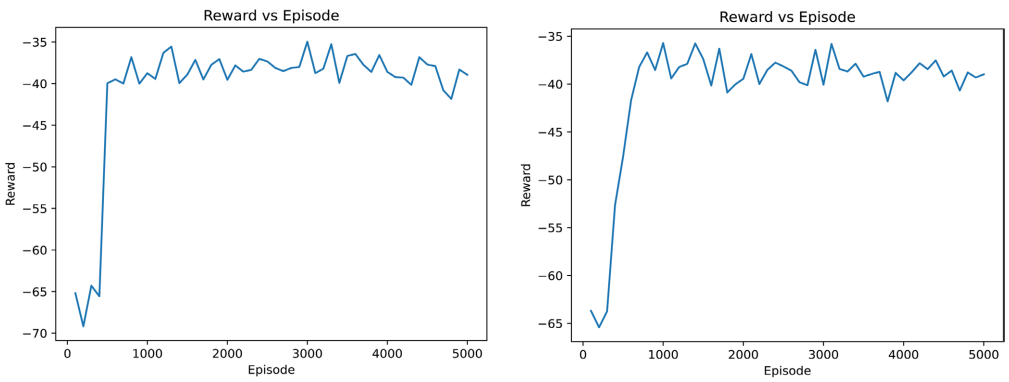


Figure 8. Small layout with 10 agents
图 8. 10 个智能体与小空间



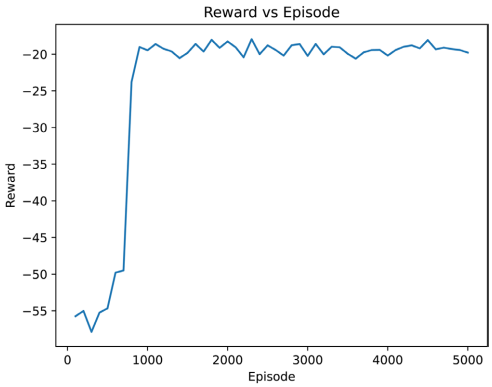


Figure 9. Large layout with 5 agents
图 9. 5 个智能体与大空间

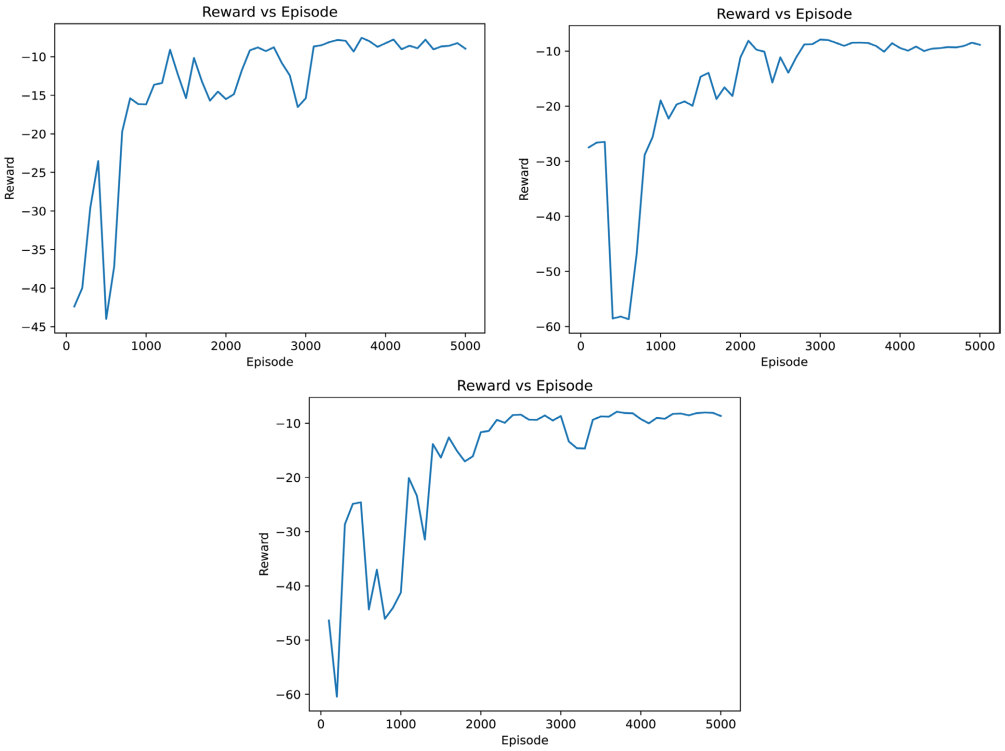
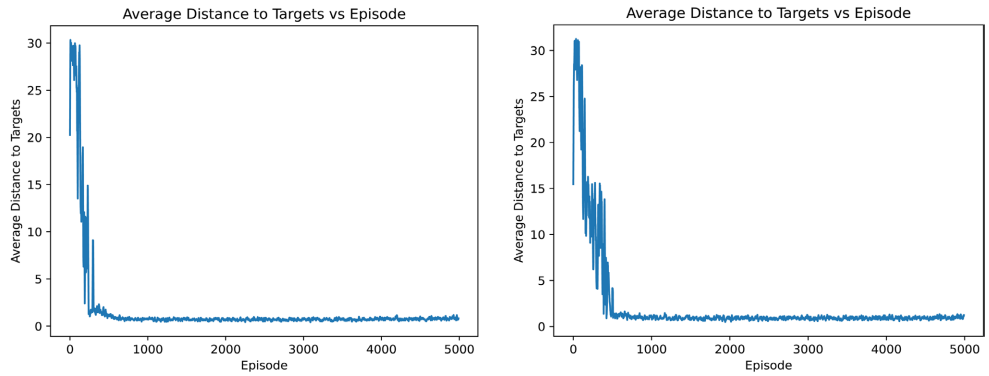


Figure 10. Large layout with 10 agents
图 10. 10 个智能体与大空间



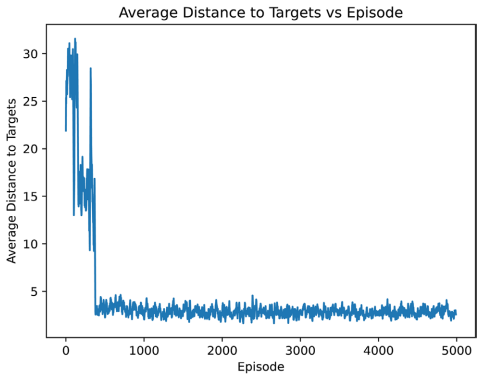


Figure 11. Small layout with 5 AGVs
图 11. 5 个 Agent 与小空间

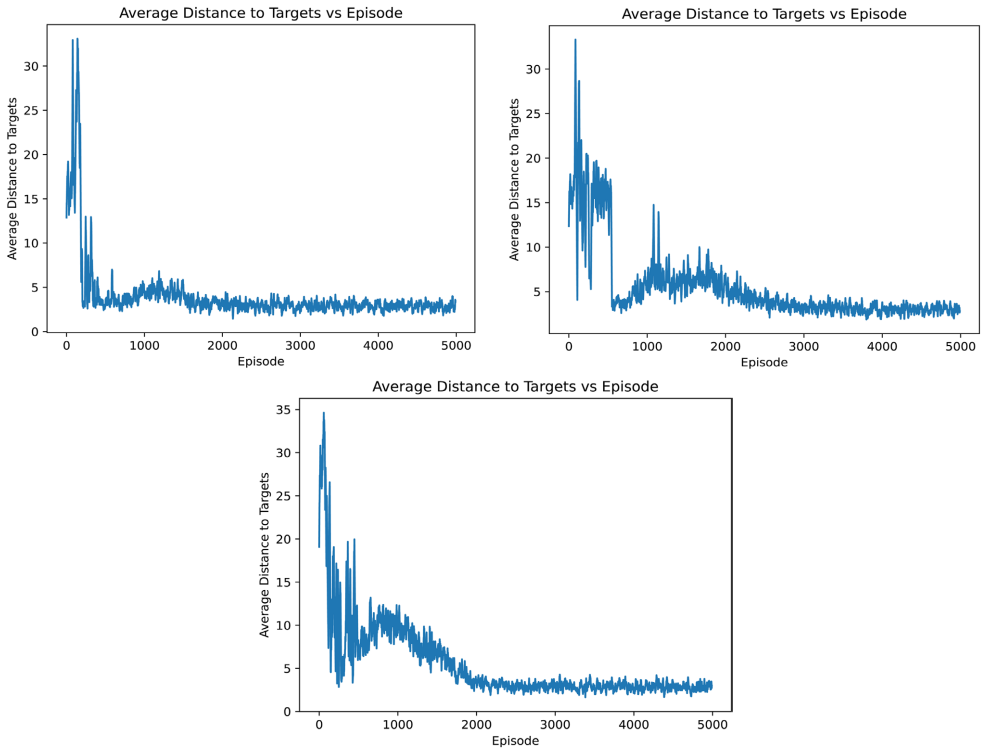
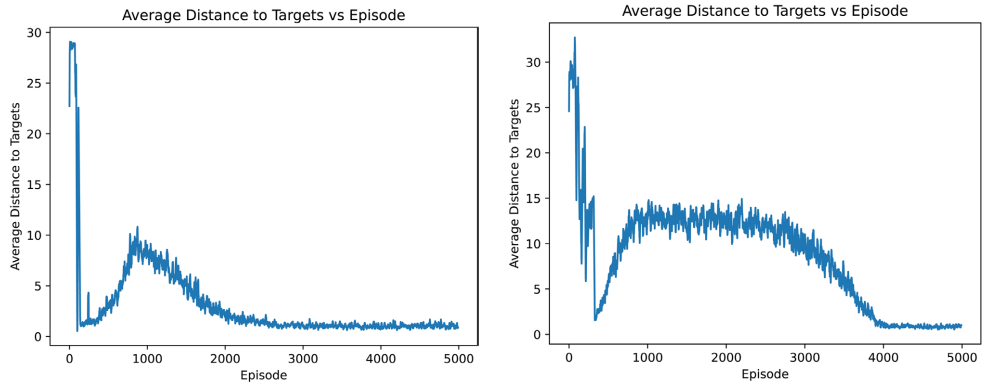


Figure 12. Small layout with 10 AGVs
图 12. 10 个 Agent 与小空间



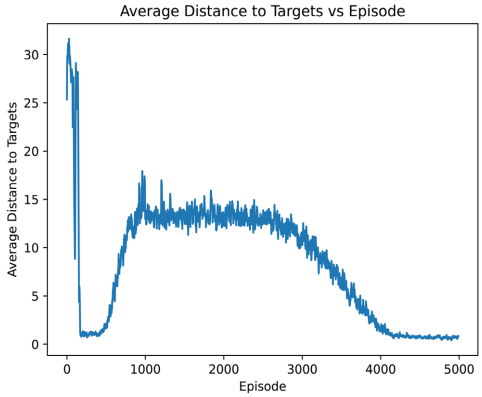


Figure 13. Small layout with 5 AGVs
图 13. 5 个 Agent 与小空间

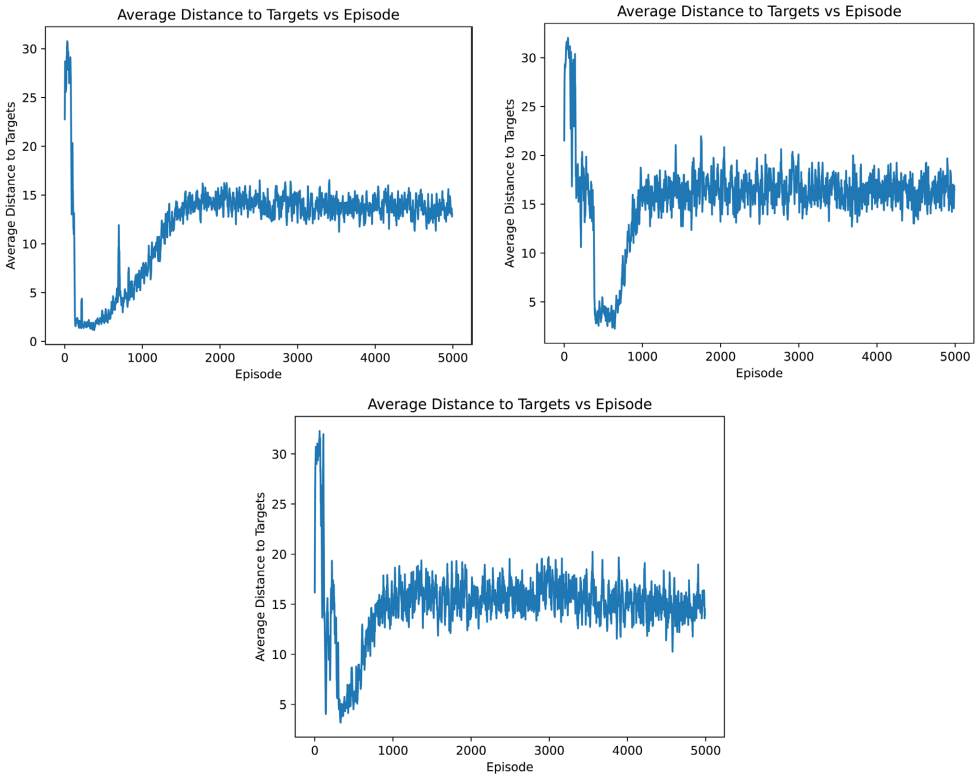


Figure 14. Large layout with 10 AGVs
图 14. 10 个 Agent 与大空间

4. 结论

本研究探讨了多智能体强化学习算法在智能仓储中的应用。得到如下结论：

(1) 通过利用强化学习技术，智能体可以从与环境的交互中学习并不断提高决策能力。在仓储环境中，区别于传统的固定路线的路径规划，智能体根据仓储环境状态而自主产生合适的路径，使仓库运营的效率、灵活性和适应性都显著提高。

(2) 模拟了基于 Actor-Critic 方法的多智能体强化学习算法，并利用该算法解决仓储环境下 AGV 的路径控制问题。仿真实验证实，在 5000 轮迭代下，10 个以内的 AGV 已经具备能在一定范围内共同寻

找目标的能力。AGV 根据策略网络输出的动作策略自主规划每一时间步的动作并记录奖励与平均距离。所有 AGV 在 3000 轮迭代后的动作策略趋于稳定。最终，仿真的奖励函数曲线收敛稳定，AGV 与目标的平均距离持续缩短并收敛。可以充分证实 AGV 能在实际仓储环境下根据自己的策略自主规划路径并找到目标。

(3) 根据真实的智能仓储场景，完成仓库环境的初步设计，为今后的算法实验提供了理想的环境。

总体而言，这项研究有助于探索 MARL 算法在提高智能仓库系统效率方面的能力。

基金项目

科学技术发展基金(0047/2021/A1)。

参考文献

[1] Foerster, J., Assael, I.A., De Freitas, N. and Whiteson, S. (2016) Learning to Communicate with Deep Multi-Agent Reinforcement Learning. arXiv: 1605.06676. https://arxiv.org/abs/1605.06676

[2] Sukhbaatar, S. and Fergus, R. (2016) Learning Multiagent Communication with Backpropagation. arXiv: 1605.07736. https://arxiv.org/abs/1605.07736

[3] Wang, C. and Mao, J. (2019) Summary of AGV Path Planning. 2019 3*rd International Conference on Electronic Information Technology and Computer Engineering* (*EITCE*), Xiamen, 18-20 October 2019, 332-335. https://doi.org/10.1109/EITCE47263.2019.9094825

[4] Bai, X., Fielbaum, A., Kronmuller, M., Knoedler, L. and Alonso-Mora, J. (2022) Group-Based Distributed Auction Algorithms for Multi-Robot Task Assignment. *IEEE Transactions on Automation Science and Engineering*, **20**, 1292-1303. https://doi.org/10.1109/TASE.2022.3175040

[5] Hu, J., Niu, H., Carrasco, J., Lennox, B. and Arvin, F. (2022) Fault-Tolerant Cooperative Navigation of Networked UAV Swarms for Forest Fire Monitoring. *Aerospace Science and Technology*, **123**, 107494. https://doi.org/10.1016/j.ast.2022.107494

[6] Chen, M., Wang, T., Ota, K., Dong, M., Zhao, M. and Liu, A. (2020) Intelligent Resource Allocation Management for Vehicles Network: An A3C Learning Approach. *Computer Communications*, **151**, 485-494. https://doi.org/10.1016/j.comcom.2019.12.054

[7] Chen, M., Liu, W., Wang, T., Liu, A. and Zeng, Z. (2021) Edge Intelligence Computing for Mobile Augmented Reality with Deep Reinforcement Learning Approach. *Computer Networks*, **195**, 108186. https://doi.org/10.1016/j.comnet.2021.108186

[8] Garaffa, L.C., Basso, M., Konzen, A.A. and de Freitas, E.P. (2021) Reinforcement Learning for Mobile Robotics Exploration: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, **34**, 3796-3810. https://doi.org/10.1109/TNNLS.2021.3124466

[9] Wei, E., Wicke, D., Freelan, D. and Luke, S. (2018) Multiagent Soft q-Learning. arXiv: 1804.09817.

[10] Foerster, J., *et al.* (2017) Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning. *Proceedings of the* 34*th International Conference on Machine Learning*, **70**, 1146-1155.

[11] Omidshafiei, S., *et al.* (2017) Deep Decentralized Multi-Task Multi-Agent Reinforcement Learning Under Partial Observability. *Proceedings of the* 34*th International Conference on Machine Learning*, **70**, 2681-2690.

[12] Oliehoek, F.A., Spaan, M.T.J. and Vlassis, N. (2008) Optimal and Approximate Q-Value Functions for Decentralized POMDPs. *Journal of Artificial Intelligence Research*, **32**, 289-353. https://doi.org/10.1613/jair.2447

[13] Oliehoek, F.A. and Amato, C. (2016) A Concise Introduction to Decentralized POMDPs. Springer International Publishing, Switzerland. https://doi.org/10.1007/978-3-319-28929-8

[14] Lowe, R., *et al.* (2017) Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv: 1706.02275. https://arxiv.org/abs/1706.02275