多智能体强化学习下连续泊位与岸电协同分配

期刊:	*自动化学报*
稿件 ID	AAS-CN-2025-0504
稿件类型:	论文
中文关键词:	自动化集装箱码头, 泊位分配, 深度强化学习, 多智能体, 实时调度
英文关键词:	Automated container terminal, berth allocation, deep reinforcement learning, Multi-agent, real-time scheduling
学科方向:	控制理论与控制工程

多智能体强化学习下连续泊位与岸电协同分配

摘　要：近年来，多智能体强化学习（Multi-agent reinforcement learning， MARL）在复杂协同决策问题中展现出巨大潜力，故针对自动化集装箱码头连续泊位分配与岸电协同分配问题，提出一种基于多智能体强化学习的实时动态调度框架，以克服传统启发式与数学规划方法在求解耗时、对历史分配方案学习乏力方面的不足。该框架将问题建模为部分可观测马尔可夫决策过程（Partially Observable Markov Decision Process, POMDP），基于集中训练-分散执行（Centralized Training with Decentralized Execution, CTDE）架构，采用多智能体双延迟深度确定性策略梯度算法（Multi-Agent Twin Delayed Deep Deterministic Policy Gradient, MATD3）训练协同优化策略。设计5类奖励函数以克服奖励稀疏问题，并引入混合探索机制以平衡探索与利用。实验结果表明，MATD3算法在多项关键性能指标上均显著优于主流MADDPG与MAAC算法：在65艘船舶的高负载场景下，泊位利用率提升约2-5%；总碳排放量降低最多达17.4%（与MAAC相比）；船舶平均等待时间减少37%（相较于MADDPG）。同时训练后的策略网络可在1秒内生成实时调度方案，不仅兼顾了经济性与环保性，还验证了奖励函数设计与双Critic机制在抑制Q值高估、加速训练过程中的有效性。

关键字：自动化集装箱码头；泊位分配；深度强化学习；多智能体；实时调度
中图分类号：U 691

Collaborative Allocation of Continuous Berth and Shore Power under Multi-Agent Reinforcement Learning

Abstract: In recent years, Multi-Agent Reinforcement Learning (MARL) has demonstrated significant potential in complex collaborative decision-making problems. To address the issue of integrated continuous berth and shore power allocation in automated container terminals, this paper proposes a real-time dynamic scheduling framework based on MARL, aiming to overcome the shortcomings of heuristic and mathematical programming methods, such as long computation times and the inability to learn from historical allocation schemes. The framework models the problem as a Partially Observable Markov Decision Process (POMDP). Based on the Centralized Training with Decentralized Execution (CTDE) architecture, the Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (MATD3) algorithm trains the collaborative optimization policy. Five reward functions are designed to mitigate the sparse reward problem, and a hybrid exploration mechanism is introduced to balance exploration and exploitation. Experimental results show that the proposed MATD3 algorithm significantly outperforms mainstream MADDPG and MAAC algorithms across multiple key performance indicators: under a high-load scenario with 65 vessels, berth utilization is improved by approximately 2–5%; total carbon emissions are reduced by up to 17.4% (compared to MAAC); and average vessel waiting time is reduced by 37% (compared to MADDPG). Furthermore, the trained policy network can generate real-time scheduling plans within one second, effectively balancing economic efficiency and environmental sustainability, while also validating the effectiveness of the reward function design and the double-critic mechanism in mitigating Q-value overestimation and accelerating the training process.

Key words：Automated container terminal, berth allocation, deep reinforcement learning, multi-agent, real-time scheduling

0 引　言

自动化集装箱码头（图1 a）作为全球供应链的核心枢纽，其连续平直的岸线（图1 b）在提升船舶靠泊效率与泊位利用率方面具有显著优势（图1 c）。依托智能TOS系统（Terminal Operation System, TOS），码头将泊位分配、船舶配积载等关键决策时间从“数天”压缩至“数小时”；而随着深度强化学习技术的成熟，将其融入TOS系统有望进一步缩短决策时间并增强方案的鲁棒性，成为当前研究热点。另一方面，为应对航运业的碳排放问题，码头普遍引入岸电系统，如何协同“连续泊位分配”与“岸电分配”以降低能耗，成为亟待突破的研究方向。

当前泊位分配与岸电协同优化的研究思路主要有两种：第一种思路是对泊位进行离散化，离散后的泊位只允许有一艘船舶停靠，根据泊位是否安装有岸电、船舶是否可用岸电，确定是否使用岸电[1-4]。第二种思路是船舶可停靠在泊位上满足停靠要求的任意位置，根据停靠位置和岸电覆盖范围是否有交叉、船舶是否可用岸电，确定船舶是否使用岸电。该思路需引入连续决策变量和复杂的几何约束（如船舶间安全距离），这将增加模型求解难度，故仅有少量学者采用第二种思路对该问题进行研究，其中，Zhen等[5]、Iris等[6]、Mauri等[7]分别使用混合整数非线性规划（MIP）、时空网络建模、元启发式算法等方式处理连续变量：由于

MIP和时空网络建模需将时间、空间离散化为足够小的间隔，会导致变量和约束数量剧增，求解时间呈指数增长；元启发式算法采用实数编码刻画靠泊位置与时间，需设计修复策略修复迭代中不可行解，结果不稳定且易陷入局部最优。

求解算法是决策泊位分配计划的关键。针对泊位分配问题，当前求解算法主要有三种：第一种算法是数学规划法。泊位分配问题是NP-hard问题，该方法处理的规模相当有限，如Wu等[9]考虑潮汐、进出港时段交替与偏好泊位的影响，建立整数线性规划模型，将数学模型通过Dantzig-Wolfe分成主问题和子问题，设计分支定价算法求解。Jia等[10]考虑内锚地影响，以到港船舶时间惩罚成本之和最小为目标函数，构建了数学规划模型，并结合拉格朗日松弛来求解。第二种算法是启发式算法，该算法能在短时间内求出大规模问题的近似解，如Park等[11]研究泊位分配的鲁棒问题时，在粒子群算法中融入时间缓冲区的自适应过程。第三种算法是强化学习算法，该类算法能够结合过往的分配数据学习分配经验，最终使算法具有推理能力，并能随分配方案持续学习，可兼顾求解时间和规模，如Ai等[12]通过分析散货港口进口业务的装卸过程，构建了一个马尔科夫决策模型，并采用了PS-D3QN的强化学习算法求解。Li等[13]考虑了船舶需求信息，将码头作业时间等特征加载至状态集，并将调度过程离散化为动作，并采用双DQN算法求解。Cervellera等[14]为解决多式联运码头泊位分配问题，将问题转化为马尔可夫决策过程。Li等[15]建立了泊位分配问题的马尔可夫决策过程模型，通过DQN算法对问题求解。

通过梳理文献，泊位与岸电分配相关研究仍有以下不足：（1）以数学建模方式刻画问题，不能反映泊位环境的动态性。（2）泊位分配计划求解算法多以启发式算法为主，该算法存在重复决策问题，不能据过往分配方案学习分配经验。（3）连续泊位是码头的真实环境，岸电是减少碳排放的有效手段，而将连续泊位分配与岸电分配协同研究时会引入连续变量（靠泊位置和时间）和0-1变量（是否连接岸电），这将使问题求解变得更为困难，需对求解算法做进一步研究。

本文针对多智能体强化学习下码头连续泊位与岸电协同分配优化问题，提出了一种基于CTDE范式的实时动态调度框架，使其能够自主推演泊位分配过程，最终实现各船舶在竞争有限泊位资源的同时，降低碳排放、减少靠泊等待时间、最大化泊位利用率。

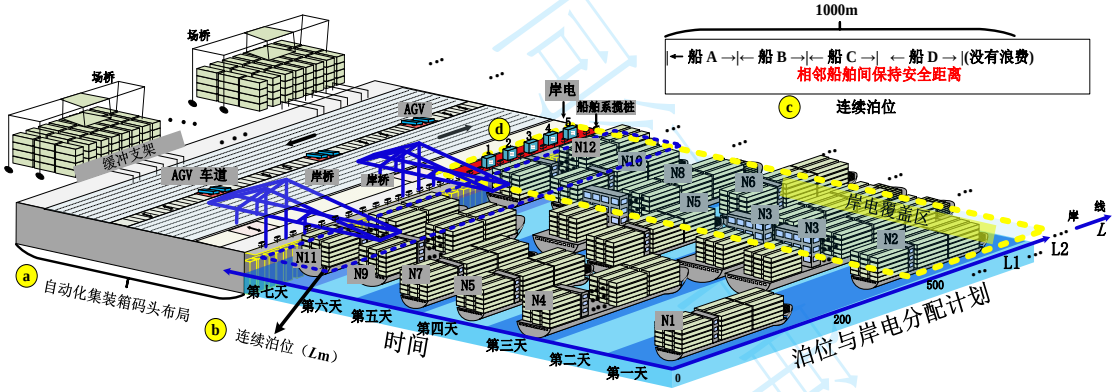


图1 自动化集装箱码头连续泊位与岸电协同分配

Fig.1 Automated container terminal with continuous berth and shore power

1 多智能体连续泊位分配模型构建

1.1 问题描述

图1是自动化集装箱码头 L 米岸线的泊位与岸电分配计划。码头在岸线每隔10米设置一个系揽桩，岸电在系揽桩后，只能为泊位某个区段提供岸电。图1 d是岸电覆盖区，当船舶靠泊至该区域时，船舶可用岸电。

泊位分配计划通常以7天为基准制定，并在船舶实际抵港前1-3天通过动态滚动调整分配计划（Giallombardo et al 2010）。码头需综合考虑泊位利用率、碳排放量、船舶等待时间等因素，制定在时间和空间上不重叠的泊位与岸电分配计划，制定计划时需遵循以下三类约束：

（1）空间物理约束（两条）：①船舶间需保持安全距离；②高优先级船舶需优先安排靠泊。

（2）时间约束（三条）：①船舶需在到达后才能靠泊；②抵港船舶在锚地等待时间不能超过最长等待时间。③泊位在任何时刻只能停靠一艘船。

（3）岸电使用约束（两条）：①岸电覆盖区只能同时为有限艘船提供岸电。②只有船舶靠泊在岸电覆盖区才能使用岸电。

综上，本文问题可表述为：针对连续泊位长度为 L 的集装箱码头，在已知船舶到港时间、船舶作业时长、优先级、是否可用岸电等信息下，以最小化船舶等待时间、碳排放，最大化泊位利用率为目标，制定

在时空不重叠的泊位与岸电分配计划。

1.2 符号说明

集合	
N	船舶集合， n 表示任一船舶
S	船舶状态集合， s_{nt} 表示船舶 n 在时间 t 下的状态
A	船舶动作结合， a_{nt} 表示船舶 n 在时间 t 下的动作
H	岸电覆盖段集合， h 表示任意段岸电
输入变量(泊位、船舶、岸电相关)	
T	总时长
L	连续泊位长度
T_n^a	船舶 n 到达港口的时刻
q_n	0-1常数。船舶 n 若能用岸电则 $q_n=1$ ，否则为0
L_n	船舶 n 的长度
d_n	船舶 n 的吃水深度
t_n^b	船舶 n 的靠泊时长（含岸电连接时间（若 $q_n=1$ ）、离泊时间）
p_n	船舶 n 的优先级
u_n	船舶 n 的岸电接口距离船尾的距离
v_h	第 h 段岸电的容量
$[s_h, f_h]$	第 h 段岸电的可服务的开始位置和结束位置
输入变量（船舶经济相关参数）	
e_n	船舶 n 每小时所需电量（百万瓦时）
c_n	船舶 n 辅机生产百万瓦时电量的碳排放
c_h	岸电生产百万瓦电量的碳排放
t_n^w	船舶最长等待时间
c_1	船舶单位等待时间惩罚成本
c_2	船舶优先级奖励
c_3	单位碳排放惩罚成本
c_4	泊位利用率奖励
c_5	无效动作单位惩罚成本
c_6	岸电使用奖励
c_7	泊位分配计划分散奖励
c_8	船舶 n 具备使用岸电条件，且分配至岸电覆盖区所给予的奖励
马尔科夫变量	
Δt	相邻两个决策时间之间的间隔
P_{nt}	表示船舶 n 在时间 t 做出动作 a_{nt} 后，转态转变为 $s_{n(t+\Delta t)}$ ， $P_{nt}=P(s_{n(t+\Delta t)}\mid s_{nt},a_{nt})$
R_{nt}	表示船舶 n 在时间 t 做出动作 a_{nt} 后，所获得的奖励， $R_{nt}=R(s_{nt},a_{nt})$
$done_{nt}$	若船舶 n 在时间 t 完成泊位分配，则 $done_{nt}=1$ ，否则为0
中间变量	
ε_t	在时间 t 下的泊位利用率
q_{ht}	第 h 段岸电在时间 t 下的使用量
k_{nt}	船舶 n 在时间 t 下的尝试次数
h_{nt}	在时间 t 下与船舶 n 靠泊位置最近的岸电段 h
c_{nt}	船舶 n 截至时间 t 下等待产生的碳排放
g_{nt}	船舶 n 截至时间 t 下靠泊产生的碳排放
神经网络变量	
γ	折扣因子， $\gamma\in[0,1]$

lr_a	Actor学习率
lr_c	Critic学习率
τ	软更新参数更新幅度
θ	Actor参数
θ^-	Target Actor参数
ω	Critic参数
ω^-	Targert Critic参数
μ	神经网络。μ_θ表示Actor，μ_{θ^-}表示Target Actor，μ_ω表示Critc，μ_{ω^-}表示Target Critic
决策变量	
l_n	连续变量。船舶n船头的位置，$l_n \in [0, L]$
t_n	连续变量。船舶n到达锚地后至开始靠泊的等待时长，$t_n \in [0, t_n^w]$
π_n	连续变量。船舶n使用岸电的概率，$\pi_n \in [0,1]$

1.3 部分可观测马尔可夫决策过程

泊位和岸电分配是一个动态过程，船舶陆续到达，码头方需要根据当前已靠泊船舶、空闲泊位、岸电使用等情况为新来船舶分配泊位和岸电，决策会影响未来的状态（剩余泊位资源、后续船舶的等待时间等）。这完美符合马尔可夫决策过程（Markov Decision Problem，MDP）。同时，在真实码头环境中，每一个艘船很难拥有全局的、完全精确的信息。因此，每艘船拥有的只是全局状态的一个局部观测（Partial Observation，PO），故采用POMDP模型比完全信息的MDP模型更符合实际。POMDP由以下因素组成：

$$POMDP = (S, A, P(s_{t+1} \mid s_t, a_t), R(s_t, a_t), \gamma) \tag{1}$$

多艘船舶竞争有限的泊位和岸电资源时，每艘船的决策都会改变环境，影响其他船舶。MARL是解决这类问题的前沿方法，而POMDP是MARL中最常用和有效的建模框架之一。

1.3.1 状态空间

为准确表达船舶、岸电、泊位的全局信息，将全局状态s设计为一个$|N|\times 1\times 18$维的矩阵，其中$|N|$表示船舶数量，1×18表示每个船舶的特征维度。每个船舶的1×18特征，包含：

（1）5个船舶静态特征$(L_n, t_n^b, p_n, q_n, t_n^b e_n)$。

（2）7个船舶动态特征$(\max(0, t-T_n^a), \max(0, T_n^a - t), h_{nt}, c_{nt}, g_{nt}, k_{nt}, R_{nt}^{'})$。$R_{nt}^{'}$表示在时间$t$因无效动作获得的负奖励。

（3）1个岸电静态特征v_h。

（4）1个岸电动态特征q_{ht}。

（5）3个泊位动态特征$(\varepsilon, t, \sum_{n\in N} done_{nt}, \sum_{h\in H} q_{ht})$。

（6）噪音，目的是增强智能体的探索。

图2 a是对船舶n的18个特征数字化得到的特征矩阵。由于18个特征的单位标准不一致，故分别对各特征进行归一化，如图2 b所示。局部特征矩阵将用于计算在时间t下的决策变量$[l_{nt}, t_{nt}, \pi_{nt}]$，$|N|$个船舶的特征矩阵组成$N\times 1\times 18$维的全局特征矩阵，如图2 c所示。全局特征矩阵将用于评估动作价值$Q(s_{nt}, a_{nt})$。

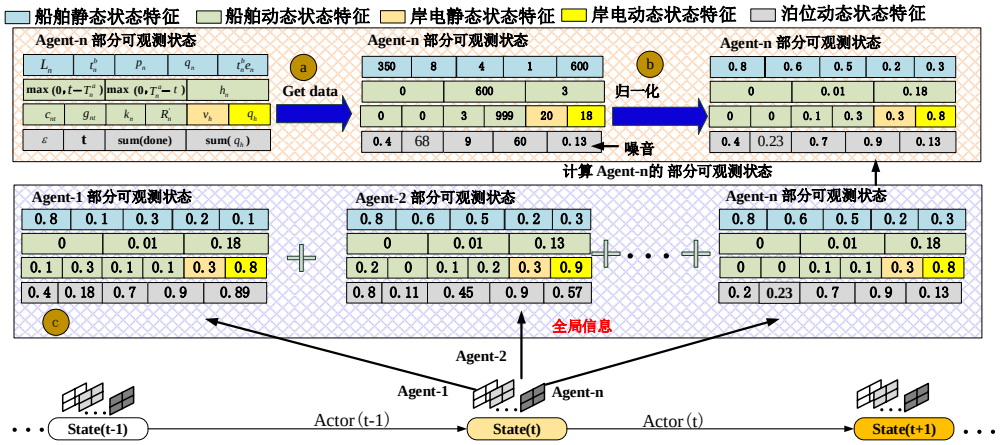


图2 局部和全局状态特征

Fig.2 The global feature matrix and local feature matrix

1.3.2 动作空间

$$l_{nt} = \mu_\theta(s_{nt})_{11} \times L \in [0, L] \tag{2}$$

$$t_{nt} = \mu_\theta(s_{nt})_{12} \times t_n^w \in [0, t_n^w] \tag{3}$$

$$\pi_{nt} = \begin{cases} 1, & \mu_\theta(s_{nt})_{13} > m \\ 0, & \mu_\theta(s_{nt})_{13} \le m \end{cases} \tag{4}$$

式（2）表示船头位置。式（3）表示船舶等待时长。式（4）表示船舶使用岸电的概率，其中m表示决策阈值。

1.3.3 奖励函数

奖励函数设计重点是解决奖励稀疏问题。由于船舶只有完成泊位分配和未完成两种状态，属于二元奖励[17]问题，对此，本文设计了5个层次的奖励函数，确保智能体训练能够收敛。

奖励函数遵循了多目标优化问题的分解原则，将码头运营的经济性（等待成本）、环保性（碳排放）和效率（泊位利用率）三大目标，映射为智能体可感知的即时奖励信号。五类奖励函数设计目的及对应目标如表1所示。

表1 奖励函数设计目的及对应目标

Table 1 Purpose of reward function design and corresponding objectives

奖励类型	设计理由	对应目标
基础奖励 (成功靠泊)	鼓励船舶完成首要任务，即成功靠泊。没有此正向奖励，船舶倾向于不采取任何行动。	最大化成功靠泊船舶数量
基础惩罚 (等待时间、碳排放)	将运营成本（船舶等待耗油）和环保成本（碳排放）直接量化为负奖励。船舶为最大化累积奖励，必须最小化这些成本。	最小化总等待时间 最小化总碳排放
附加奖励 (使用岸电)	对使用清洁能源-岸电的行为给予额外激励，引导系统向环保方向优化。	最大化岸电使用率
全局奖励 (泊位利用率)	引入全局信息，避免船舶自私行为（如过度分散靠泊），鼓励采取能提高整体资源利用率的策略。	最大化泊位利用率
无效动作惩罚	为无效动作（如分配已被占用的泊位）提供即时负反馈，极大地加速训练收敛，避免船舶在无效动作上浪费探索时间。	提高算法效率

（1）基础奖励

$$R_1 = \begin{cases} c_2 p_n, & done_{nt} = 1 \\ 0, & done_{nt} = 0 \end{cases} \tag{5}$$

$$R_2 = c_1 t_{nt}, \quad done_{nt} = 1 \tag{6}$$

$$R_3 = \begin{cases} c_3(c_n t_n e_n + c_h t_n^d e_n), & q_n = 1, \pi_{nt} = 1, done_{nt} = 1, s_h \le l_{nt} + L_n - u_n \le f_h, \forall h \in H \\ c_3(c_n t_n e_n + c_n \cdot t_n^d e_n), & q_n = 1, done_{nt} = 1, \pi_{nt} = 0 \\ c_3 c_n e_n \max(0, (t - T_n^a)), & done_{nt} = 0 \end{cases} \tag{7}$$

奖励（5）奖励表示船舶因成功靠泊获得的正奖励。奖励（6）表示船舶等待时长负奖励。奖励（7）表示船舶的碳排放负奖励。

（2）附加奖励

$$R_4 = \begin{cases} c_6, & q_n = 1, \pi_{nt} = 1 \\ 0, & others \end{cases} \tag{8}$$

$$R_5 = c_7 (1 - \frac{|l_n - 0.5L|}{0.5L}) \tag{9}$$

奖励（8）表示船舶具备岸电使用条件，且做出了使用岸电动作所获得的正奖励。奖励（9）表示船舶靠泊分散获得的正奖励。

（3）全局奖励

$$R_6 = c_4 \frac{\sum_{n \in N} L_n t_n^d}{\max TL}, \quad done_{nt} = 1 \tag{10}$$

奖励（10）表示泊位利用率为各成功靠泊船舶带来的正奖励，鼓励采用泊位利用率高的方案。

（4）无效动作惩罚

$$R_7 = c_5 p_n, \quad done_{nt} = 0 \tag{11}$$

$$R_8 = 1.5 c_5 p_n, \quad l_{nt} + L_n > L, done_{nt} = 0 \tag{12}$$

$$R_9 = 1.5^{\min(k_n, 8)} c_5 p_n, \quad done_{nt} = 0 \tag{13}$$

$$R_{10} = 0.5 c_5 \Delta t, \quad t > T_n^a, \quad done_{nt} = 0 \tag{14}$$

奖励（11）表示船舶船首位置无效失败获得的负奖励。奖励（12）表示因船尾靠泊位置无效获得的负

奖励。奖励（13）表示船舶尝试无效次数获得负奖励，鼓励船舶尽快制定出泊位分配方案。奖励（14）表示船舶动作无效，截至时间t船舶获得的等待负奖励。

（5）岸电分配奖励

$$R_{11}=\begin{cases}c_8, & q_n=1, done_{nt}=1, s_h\le l_{nt}+L_n-u_n\le f_h, \forall h\in H\\ 0, & ohters\end{cases} \tag{15}$$

奖励（15）表示具备岸电使用条件的船舶分配至具有岸电覆盖泊位所获得的奖励。

（6）总奖励函数

$$R(s_{n\,t},a_{n\,t})=\sum_{i=1}^{11}R_i \tag{16}$$

奖励（16）表示船舶 n 在时间t中获得的总奖励。

2 算法设计

本文选择MATD3算法作为基础框架，因其具备处理连续动作空间的能力，采用CTDE范式解决环境非稳定问题，并通过双Critic网络和延迟策略更新机制有效缓解Q值高估问题。以下是本文对MATD3算法的三项理论扩展。

（1）奖励函数的结构化设计：通过五类奖励函数的层次化组合，将多目标优化问题转化为可训练的奖励信号，克服了传统二元奖励的稀疏性问题。

（2）维度感知的探索策略：针对动作空间的不同物理含义（位置、时间、概率），分别引入高斯噪声、指数分布噪声和均匀噪声，提升了探索的针对性与效率。

（3）全局-局部状态表示机制：通过构建全局特征矩阵并融合局部观测，增强了智能体对系统资源的全局感知能力，改善了多智能体策略协调。

2.1 基于MATD3算法的调度框架

图3是算法流程图，MATD3中的每艘船舶由6个神经网络组成，即1个Actor、1个Target Actor、2个Critic和2个Target Critic网络。该框架分两个阶段：

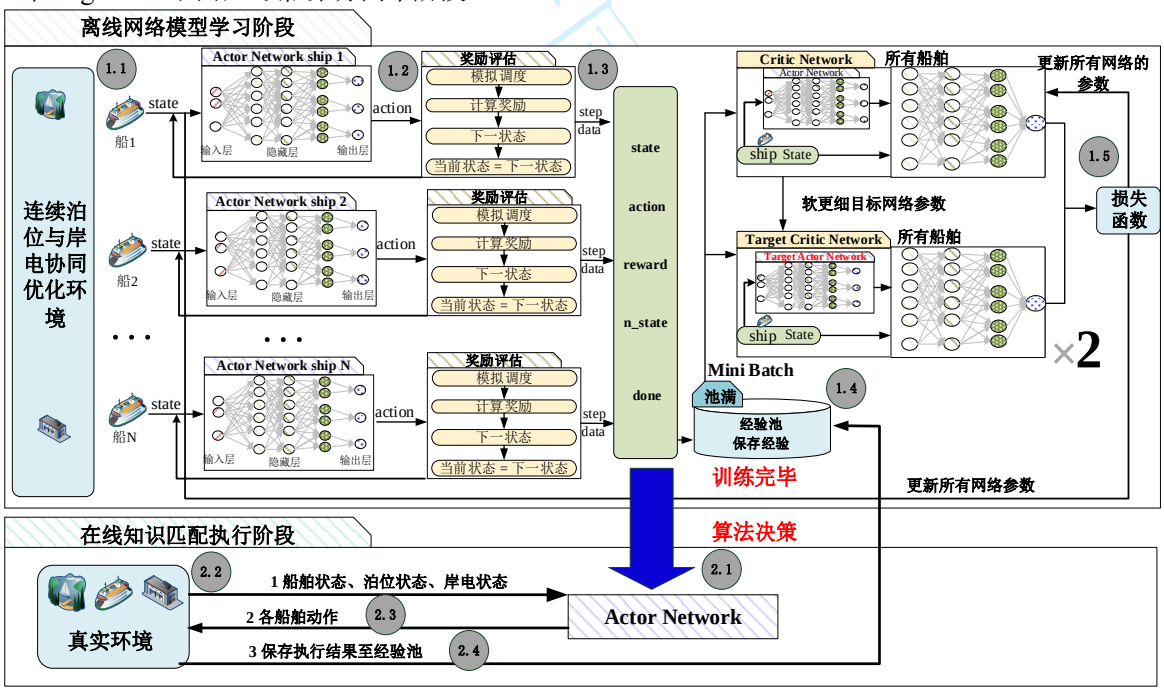


图3 MATD3算法框架

Fig.3 Framework of the MATD3 algorithm

（1）离线网络模型学习阶段

步骤 1.1 状态采集。实时获取船舶、岸电、泊位状态信息，并将其转化为状态特征。

步骤 1.2 动作决策。每艘船舶根据状态特征，用Actor网络计算动作。

步骤 1.3 奖励评估。根据每艘船舶的动作、环境信息进行模拟调度，根据调度结果计算智能体所获得的奖励。

步骤 1.4 经验储存。将船舶的(state、action、reward、next state、done)保存至经验池。

步骤 1.5 经验回放。1）若经验池未蓄满则，返回Step1.1；2）经验池蓄满后，采用经验回放，更新网

络参数，并返回Step1.1

（2）在线知识匹配执行阶段

步骤 2.1 将船舶后的局部状态输入Actor，决策该状态下的价值最高的动作。

步骤 2.2 根据决策结果进行靠泊。

步骤 2.3 将实际决策后的（state、action、reward、next state、done）保存至经验池。

2.2 Critic和Target Critic

2.2.1 Critic和Target Critic网络结构

图4是Critic网络结构，由1个输入层、2个隐藏层和1个输出层组成。

Critic和Target Critic的核心任务是学习1.3.3的多目标、多层次的奖励函数的组合，并准确评估动作的长期价值。

输入层：涵盖所有智能体的状态s_t和动作a_t，具体包括：①所有船舶、岸电、泊位的18维特征向量；②所有船舶的靠泊位置、等待时间和岸电使用概率。

输出层：是一个标量，代表了在当前全局泊位-岸电资源状态下，执行这一组联合动作所能带来的总回报，用于衡量整个调度方案的好坏。

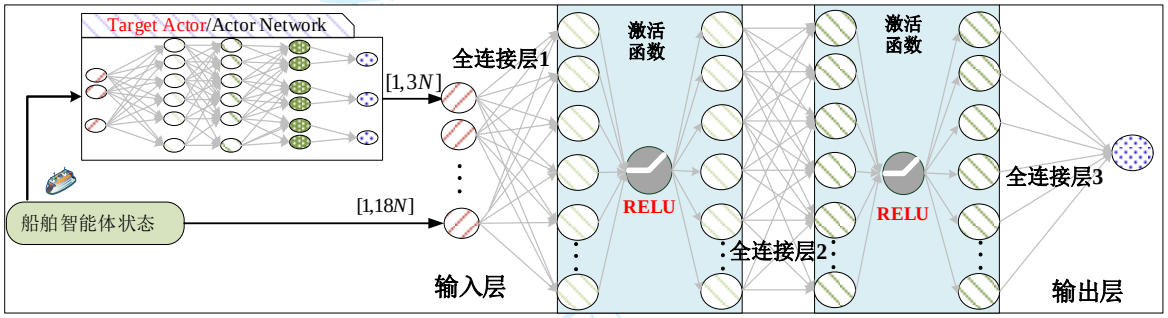


图4 Critic Nertwork和Target Critic Network

Fig.4 The Critic and Target Critic Network

2.2.2 动作价值高估

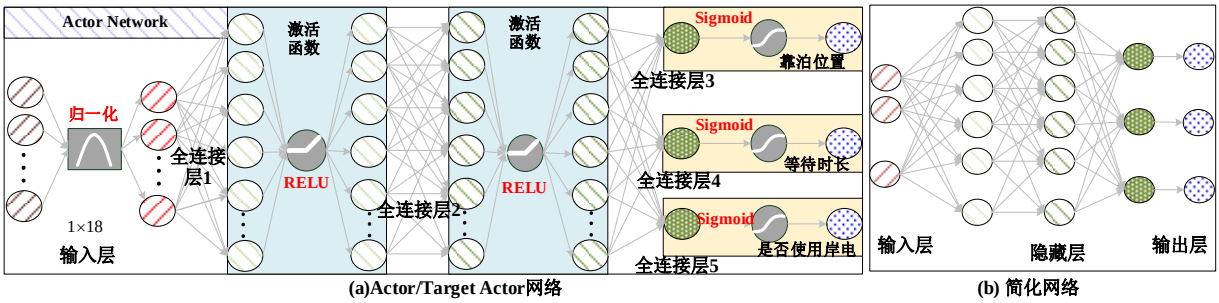
泊位与岸电协同分配的动作空间大，单Critic存在动作价值高估问题，在该情形下的时序差分误差目标为$r+\gamma\max_{a_{n(t+1)}}(Q_\omega(s_{n(t+\Delta t)},a_{n(t+\Delta t)})$；max操作被拆解为两部分：①选取$s_{n(t+\Delta t)}$下的动作价值最高的动作$a^*=\arg\max Q_\omega(s_{n(t+\Delta t)},a_{n(t+\Delta t)})$，②计算该动作的价值$Q_\omega(s_{n(t+\Delta t)},a^*)$。

以上两部分用一个Critic计算，每次得到的都是神经网络估算的所有动作价值中的最大值。由于神经网络估算动作价值会产生正向或负向的误差，在后续神经网络参数更新会累积正向误差，产生动作价值高估的问题，这种问题将影响算法的收敛。为了解决这一问题，利用两个Critic独立估算$\max_{a_{n(t+1)}}(Q_\omega(s_{n(t+\Delta t)},a_{n(t+\Delta t)})$将有效解决以上问题。

2.3 Actor和Target Actor

2.3.1 Actor和Target Actor网络结构

如图5所示，Actor和Target Actor网络由1个输入层、3个隐藏层、1个输出层组成。



(a)Actor/Target Actor网络 (b) 简化网络

图5 Actor Network

Fig.5 The Actor and Target Actor Network

输入层需进行归一化处理，第1、2层隐藏层是共享特征提取层，用RELU激活函数进行非线性化映射，第3层隐藏层是分头动作特征提取层，分别对应靠泊位置l_{nt}、锚地等待时长t_{nt}和是否使用岸电π_{nt}，最后使

用sigmoid函数将动作特征压缩至[0,1]。

输入层：单个智能体的局部观测 s_{nt} ，包含它所能感知到的泊位和岸电信息。

输出层：是智能体的动作 a_{nt} ，代表了它基于局部信息所提出的个体调度方案。

2.3.2 动作合探索策略

Actor输出确定性动作，容易对极端动作过拟合，故加入噪声，以增强探索。

$$l_{nt}=clip(l_{nt}+\eta_1,0,L),\quad \eta_1\sim(0,500^2) \tag{17}$$

$$t_{nt}=clip(t_{nt}+\eta_2,0,t_n^w),\quad \eta_2\sim Exp(1) \tag{18}$$

$$\pi_{nt}=clip(\pi_{nt}+\eta_3,0,1),\quad \eta_3\sim U(-0.1,0.1) \tag{19}$$

式（17）是泊位位置的探索机制，采用零均值高斯噪声 $\eta_1\sim(0,500^2)$ ，以适应大范围的连续岸线搜索。式（18）是等待时间的探索机制，采用指数分布噪声 $\eta_2\sim Exp(1)$ 非零的特性，用于在时间窗口内探索不同的等待策略。式（19）是岸电探索机制，采用均匀噪声 $\eta_3\sim U(-0.1,0.1)$ ，以在二进制决策（使用/不使用）附近进行平滑概率探索。

2.4 经验回放

经验回放是更新6个网络参数的关键，算法细节见Algorithm1。

Algorithm 1: Experence Replay

从经验池随机选择1024个数据(s_{nt}, a_{nt}, r_{nt}, $s_{n(t+1)}$, $done_{nt}$)

$a_{n(t+\Delta t)}=\mu_{n\theta^-}(s_{n(t+\Delta t)})$

*** 使用Target Critic 1和Target Critic 2分别估算计算Q值 ***

$Q_{n(t+\Delta t)\omega_1^-}(s_{n(t+\Delta t)},a_{n(t+\Delta t)})=\mu_{n\omega_1^-}(torch.cat((*s_{n(t+\Delta t)},\ *a_{n(t+\Delta t)}),\ dim=1))$

$Q_{n(t+\Delta t)\omega_2^-}(s_{n(t+\Delta t)},a_{n(t+\Delta t)})=\mu_{n\omega_2^-}(torch.cat((*s_{n(t+\Delta t)},\ *a_{n(t+\Delta t)}),\ dim=1))$

$Q_{n(t+\Delta t)}(s_{n(t+\Delta t)},a_{n(t+\Delta t)})=r_n+\gamma\min\{Q_{n(t+\Delta t)\omega_1^-},\ Q_{n(t+\Delta t)\omega_2^-}\}(1-done_{nt})$

*** 使用Critic 1 和 Critic 2 分别计算损失函数 *********

$Q_{nt\omega_1}(s_{nt},a_{nt})=\mu_{n\omega_1}(torch.cat((*s_{nt},\ *a_{nt}),\ dim=1))$

$Q_{nt\omega_2}(s_{nt},a_{nt})=\mu_{n\omega_2}(torch.cat((*s_{nt},\ *a_{nt}),\ dim=1))$

$Loss_{nt\omega_1}=\frac{1}{N}\sum_{n=1}^{N}(Q_{nt\omega_{\ i}}(s_{nt},a_{nt})-Q_{n(t+\Delta t)}(s_{n(t+\Delta t)},a_{n(t+\Delta t)}))^2$ *for i in N*

*** 更新Critic 1和 Critic 2的网络参数 ***************

$\nabla_{\omega_i}J(s_{n\omega},a_{n\omega})=\frac{1}{N}\sum_{n=1}^{N}\nabla_{\omega_i}(Q_{nt\omega_i}(s_{nt},a_{nt})-Q_{n(t+\Delta t)}(s_{n(t+\Delta t)},a_{n(t+\Delta t)}))^2$ *for i in N*

*** 计算Actor的损失函数 *****************

$a_{nt}=\mu_\theta(s_{nt})$

$Loss_{nt\theta}=-\frac{1}{N}\sum_{n}\mu_{n\omega_1}(torch.cat((*s_{nt},\ *a_{nt}),\ dim=1))$

*** 更新Actor的网络参数**************************

$\nabla_\theta J(a_{n\theta})=-\frac{1}{N}\sum_{n=1}^{N}\nabla_\theta\mu_\theta(s_{nt})\nabla_a Q_\omega(s_{nt},a_{nt})|_{a_{nt}=\mu_\theta(s_{nt})}$

2.5 MATD3算法流程

MATD3算法流程见Algorithm 2：

Algorithm 2: MATD3

初始化：N个智能体，每个智能体构建6个网络 $\mu_\theta,\mu_{\theta^-},\mu_{\omega_1},\mu_{\omega_2},\mu_{\omega_1^-},\mu_{\omega_2^-}$ ，并初始化参数 $\theta,\theta^-,\omega_1,\omega_2,\omega_1^-,\omega_2^-$ ，令 $\omega_1^-=\omega_1,\omega_2^-=\omega_2$

创建：环境*Env*，最大尝试数K，经验池D，批次数量 b_{num} ，训练回合epoches，软更新因子 τ ，软更新间隔G

for epoch in epoches:

  $s_0=\{o_0^1,o_0^2,...,o_0^N\}$

  count = 0

  t = 0

  Done = False

  while not Done or t > T:

    count = count+1

    $a_t=\{\mu_{n\theta}(o_{nt}),i=n,...,|N|\}$

    $s_{t+1},r_t,done_t=Env.step(a_t)$

save $(s_t, a_t, s_{t+1}, r_t, done_t)$ *in D*
$s_t = s_{t+1}$
$t = t + \Delta t$
$Done = True,\ if\ \ sum(done_t) = N$
if D.size() >= 5000:
    sample = D.sample(b_{num}*)*
    Algorithm 1: Experience Replay
    if count / G == 0:
        $\omega_1^- = \tau\omega_1 + (1-\tau)\omega_1^-$
        $\omega_2^- = \tau\omega_2 + (1-\tau)\omega_2^-$
        $\theta^- = \tau\theta + (1-\tau)\theta^-$

2.6 生成船舶

为了模拟真实港口环境中船舶的动态到达，需要一种可靠的方法来生成不同的船舶实例。Algorithm 3是详细的随机生成过程，以确保训练场景的广泛覆盖。

Algorithm 3: Generate Vessels

Vessels = []
for n in N:
    L_n =random.uniform(150, 400)
    p_n = random.randint(1, 5)
    q_n =1 if random.rand() > 0.3 else 0
    u_n =random.uniform(20, 80) if q_n =1
    d_n =10+(L_n /400)× 10+random.uniform(-1, 1)
    T_n^a =random.uniform(0, *T*/6)
    cargo=(L_n /200)^3 × 5000 × random.uniform(0.8, 1.2)
    t_n^b =360+(cargo/15000)× 240
    e_n =random.uniform(0.05, 1.2)
    c_n = random.randint(2450, 2500)
    Vessels.append($(L_n, p_n, q_n, u_n, d_n, T_n^a, t_n^b, e_n, c_n)$)
Return: Vessels

3 算例实验

实验环境为Python3.10，Pytorch2.5，Windows11系统，硬件配置为Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz 3.70 GHz；32.0 GB RAM；4GB NVIDIA Quadro P1000工作站。

3.1 参数设置

3.1.1 环境参数设置

本文创建了一个集装箱码头模拟环境：模拟了2000米长的连续海岸线，规划周期为一周（168小时）。具体参数值如表2所示。这些参数构成了多智能体强化学习训练的物理基础。

表2 泊位、岸电以及船舶经济参数

Table 2 Berths, shore power and vessel economic parameters

泊位分配参数及取值				船舶经济相关参数			
参数	取值	参数	取值	参数	取值	参数	取值
L	2000m	v_2	50MWh	c_1	-500	c_7	1000
T	7天	$[s_2, f_2]$	[1500, 2000]	c_2	300	c_8	10000
Δt	10min	m	0.5	c_3	-0.002	c_n	2500
t_n^w	4天	船舶安全距离	(1+5%) L_n	c_4	100000	c_h	800
v_1	50MWh			c_5	20		
$[s_1, f_1]$	[500,1000]			c_6	3000		

3.1.2 神经网络超参数设置

表3是神经网络的超参数设置，其中最重要的参数是 lr_a 、 lr_c 以及软更新间隔，调参细节见3.2节，其

余参仅列出取值。**船舶在环境学习会变得越来越智能，故后续称在船舶为“智能体”。**

表3 神经网络参数

Table 3 Neural network parameters

神经网络参数				经验池参数		训练参数	
参数	取值	参数	取值	参数	取值	参数	取值
γ	0.9	τ	0.01	经验池容量	100000	批次数量	1024
lr_a	0.01	G	2	经验回放启用数量	5000	软更新间隔	2
lr_c	0.01	隐藏层神经元数量	128			训练次数	10000
						测试间隔	5

3.2 超参数取值分析

超参数取值对算法训练具有重要影响，若设置不当将对训练的稳定性、智能体行为、收益将产生重要影响。本节以7艘船为例，对Actor和Critic的学习率、软更新间隔进行灵敏度分析，以探索超参数取值。

3.2.1 Actor学习率取值

固定Critic学习率为10^{-2}，$\tau=10^{-2}$，软更新间隔为2，探究Actor学习率为10^{-2}、10^{-3}、10^{-4}对智能体行为的影响。

通过图6可以看出，7个智能体的在三种学习率下的收益值出现了明显的差异，在学习率为10^{-2}时智能体行为最稳定，且能在短时间内收敛。学习率为10^{-3}时，收益曲线缓慢上升，虽然能最终实现和10^{-2}的效果一样，但有陷入局部最优的风险。学习率为10^{-4}时，收益曲线呈现下降趋势，这说明Actor无法根据智能体的实时状态做出恰当的反馈。结合三种结果，学习率应选用10^{-2}。

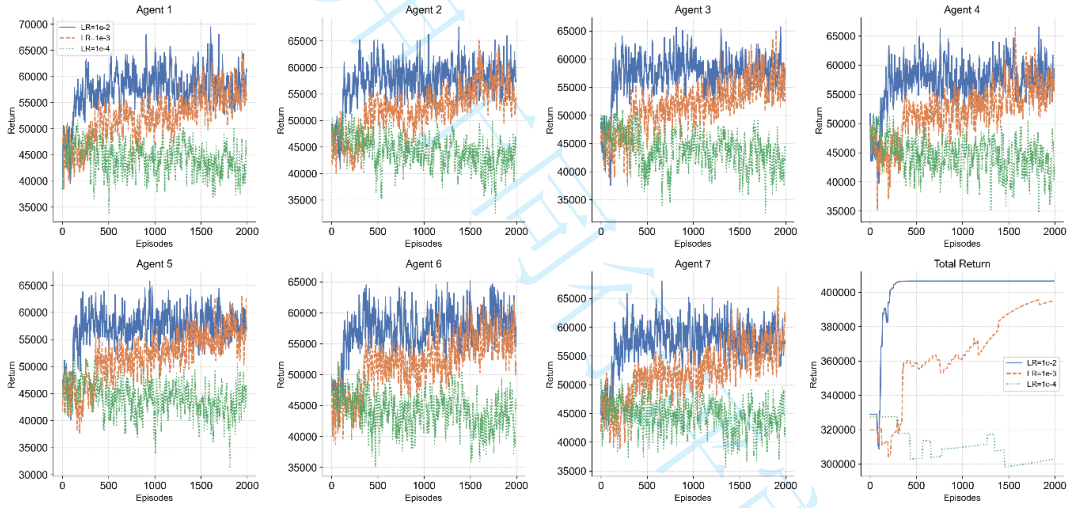


图6 Actork学习率对智能体的影响

Fig.6 Impact of the Actor network learning rate

3.2.2 Critic学习率取值

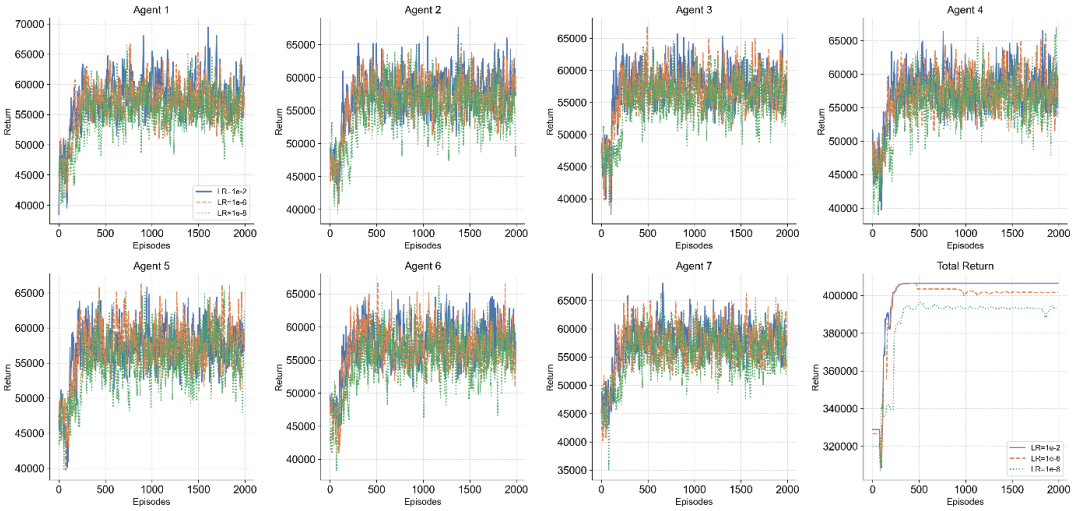


图7 Crtic学习率对智能体的影响

Fig.7 Impact of Critic network learning rate

固定Actor学习率为10^{-2}、$\tau=10^{-2}$，软更新间隔为2，探究Critic学习率为10^{-2}、10^{-6}、10^{-8}对多智能体价值估计的影响。

通过图7可以看出：（1）通过7个智能体的收益可以看出，7个智能体的收益值差距较小，这说明了Critic学习率对智能体的行为影响较小。（2）在总体收益值图中不同的Critic学习率下总收益值出现了较大差距，这说明了Actor网络在训练前期有较多无效决策，无效动作惩罚奖励导致了总体收益下降。（3）Critic学习率取值为10^{-6}时，总体收益先达到了和10^{-2}一样的收益，然后又出现了下降，这说明了虽然双Critic网络可以解决动作价值高估问题，但这里出现的下降是动作价值过于低估导致的。综上，Critic学习率为10^{-2}时，可有效减少无效决策次数，预防动作价值过于低估的问题。

3.2.3 软更新间隔取值

固定Actor学习率为10^{-2}，$\tau=1e-2$，Critic学习率为10^{-2}，探究软更新间隔为1、2、5对算法的影响。

通过图8可以看出：（1）通过7个智能体的收益值可以发现，训练完毕后7个智能体的收益值几乎一致，这说明了软更新间隔取值对智能体的行为几乎没有影响。（2）总体收益值却出现了较大差距，这是Actor网络无效决策次数使负奖励增多导致的。

综上，软更新间隔设置不会影响最终的训练效果，但设置次数大则会增加智能体的探索次数，探索次数越多可能带来的无效惩罚越多，收敛速度更慢，故软更新间隔设定为2。

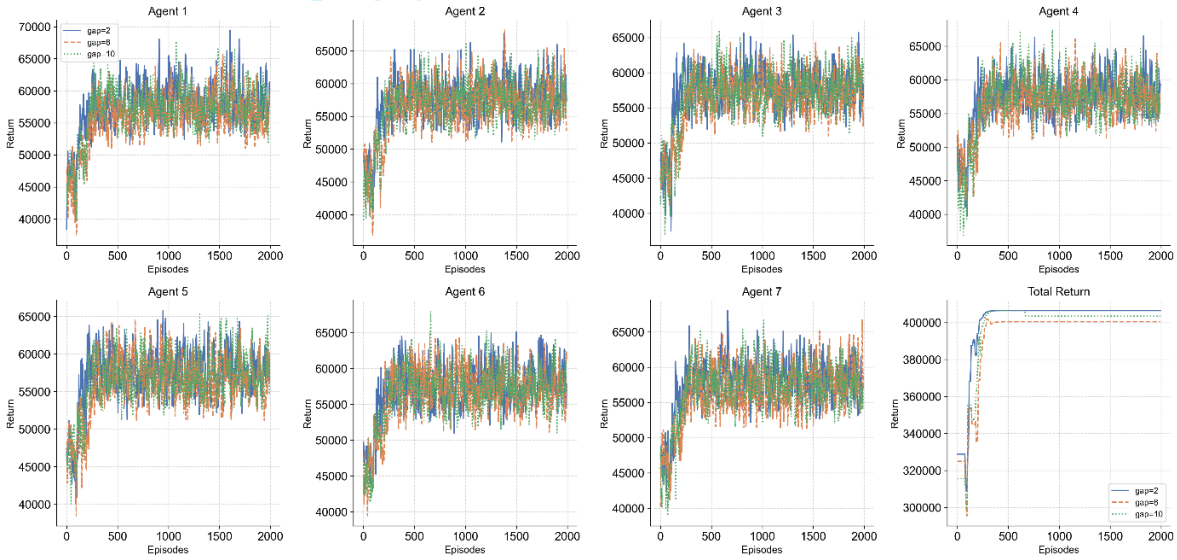


图8 软更新间隔取值对智能体的影响

Fig.8 Impact of Soft Update Gap

3.3 泊位环境稳定性以及算法智能性测试

3.3.1 环境稳定性测试

通过随机数种子，保证在重置环境时固定生成三艘船舶，用于测试环境的稳定性，相关数据见表4。

表4 3艘船舶信息

Table 4 Information of three vessels

	T_n^a	p_n	L_n	t_n^w	q_n	e_n	u_n
船舶 1	700min	3	243.63m	446.75min	1	0.094	30
船舶 2	700min	3	233.42m	538.81min	0	0.10	-
船舶 3	700min	2	195.85m	403.25min	0	0.19	-

岸线长度为2000米，因此三艘船舶的最优分配方案可以直接算出来，故用其检测算法能否在环境中收敛，以及能否实现目标。表5是训练后产生的调度计划，显然该调度方案是最优解。实验结果证明了泊位分配环境稳定、算法结构正确、奖励函数设计合理。

表5 智能体泊位分配及岸电分配计划

Table 5 Berth and shore power allocation plan for three vessels

	船尾位置	船头位置	靠泊时间	离泊时间	是否使用岸电	锚地等待时间	奖励值
船舶 1	588.73	832.36	700	1245.20	是	0.00	17165.13
船舶 2	297.39	530.82	700	1325.38	否	0.00	14655.42
船舶 3	0	195.85	700	1191.41	否	0000	15748.62

图9是智能体训练过程中的收益图，通过三个智能体的收益图可以看出三个智能体均实现了各自的收益最大化，且可用岸电的船舶收益值明显高；通过总体收益图可以看出，港口实现了收益最大化。算法训练时间3min18s，三个智能体均在第800轮左右实现收敛，除去经验采集阶段（约300轮），本文算法可在较低的训练代价下完成训练。

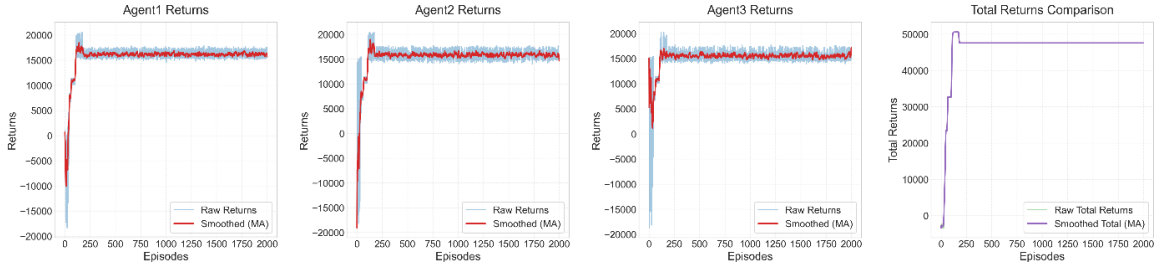


图9　环境及算法稳定性测试

Fig.9　Environmental and algorithm stability test

3.3.2 算法智能性测试

港口真实环境是每次抵港船舶的到达时间、船舶长度、靠泊时间、优先级等情况不一样，因此需对每艘船据其到达情况，制定调度方案。若MATD3算法在该情境下依然能够收敛，则说明本文算法具备具体情况具体分析的能力，即智能性。

取消随机种子，每次重置环境时只保证依据表3规则生成三个智能体。图10是实验结果，可以看出，三个智能体均实现了利益最大化（收敛），这说明了算法可根据不同状态的船舶实现科学调度。

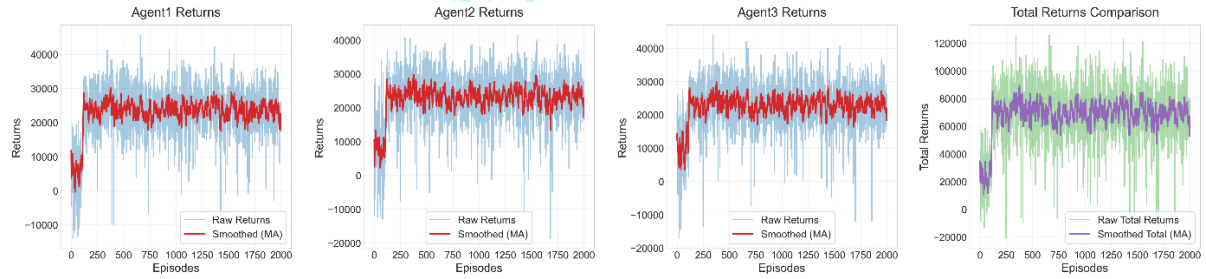


图10　算法智能性测试

Fig.10　Algorithm intelligence test

3.3.3 抑制Q值高估

为评估双Critic与单Critic方法的性能，呈现3.2.3节对Q值数据的三组对比图。

在图11 a中，单Critic（虚线）在训练初期的Q值估计波动大且存在明显高估，而双Critic（实线）对Q值的估计更为平稳，且能更精准地趋近目标Q值（点线），体现出其在Q值估计上的优势。

图11 b中，单Critic（虚线）初始高估程度高，后续虽有下降但仍存高估；双Critic（实线）的高估程度显著更低，随训练回合增加快速降低并趋于稳定，验证了其抑制Q值高估的有效性。

图11 c中，单Critic（虚线）初始时序差分误差极大，后续虽快速下降但整体误差水平仍较高；双Critic（实线）初始误差更低，下降更快，最终稳定在更低误差水平，表明其学习过程更稳定。

图11 d中，双Critic高估程度的分布（由较小的标准差体现）相对单Critic方法具有更小的离散程度。

据以上分析可得出：双Critic在Q值估计准确性、Q值高估抑制及时序差分误差降低方面更好。

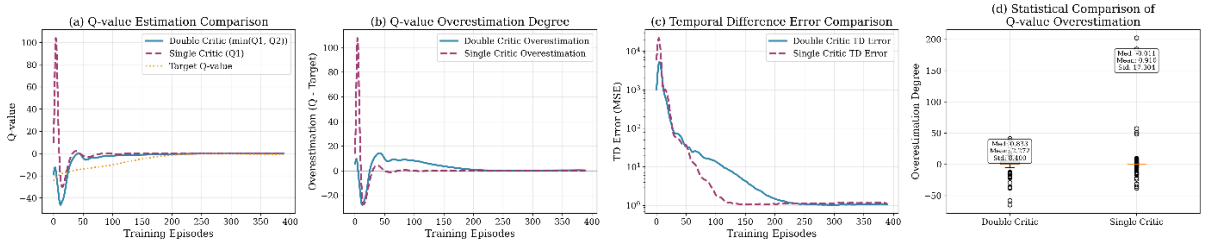


图11　Q值估计对比

Fig.11　Q-value comparison between the single Critic and double Critic

3.4 强化学习算法对比

MADDPG、MAAC算法也是基于Actor-Critic构建的算法，故将该算法和MADDPG、MAAC算法进行对比。为保障对比实验的公平性，将MAAC、MADDPG算法的Critic结构、Actor结构分别与MATD3算法的Critic结构(图4)、Actor结构（图5）设置一致，超参数均按表5设置。两类算法的更新规则等可参考文献[18][19]。

3.4.1 算法训练对比

表6是算法实验结果对比。可以看出：（1）虽然MATD3算法使用了6套网络、MADDPG算法使用4套网络，但MATD3算法训练时间更短。（2）MATD3算法制定的泊位分配方案时可以兼顾船舶等待时间、碳排放和船舶优先级。（4）通过对比泊位利用率、成功靠泊的数量，MATD3算法可以使更多的船舶靠泊。（5）通过对比总收益可以发现，MATD3算法的决策效果更好，可以用少量的探索次数，生成更好的泊位分配方案。

表6 MATD3 MADDPG MAAC实验对比结果

Table 6 Results of MATD3, MADDPG, MAAC

	智能体数量	成功靠泊数量	碳排放量 (kg)	等待时间 p_4 (min)	等待时间 p_3 (min)	等待时间 p_2 (min)	等待时间 p_1 (min)	平均等待时间（min）	总奖励值	泊位利用率	训练时间（min）
MATD3		15	**26389.26**	0.00	0.00	0.00	0.00	0.00	**584435.94**	10.91%	13.17
MADDPG	15	15	29986.58	0.00	0.00	0.00	0.00	0.00	409069.56	10.91%	18.96
MAAC		15	34641.53	0.00	0.00	0.00	0.00	0.00	199956.61	10.91%	**11.69**
MATD3		25	**36053.66**	0.00	0.00	0.00	**0.00**	**0.00**	**2093900.62**	18.51%	1001.60
MADDPG	25	25	38155.35	0.00	0.00	0.00	72.33	17.35	1928906.81	18.51%	1873.35
MAAC		25	39322.53	9.00	0.00	0.00	84.00	22.67	1775348.25	18.51%	**682.06**
MATD3		35	**33154.49**	0.00	0.00	56.50	84.85	36.34	**5790578.41**	26.12%	2963.10
MADDPG	35	35	34927.98	194.81	0.00	0.00	0.00	61.22	5613897.71	26.12%	4314.22
MAAC		35	35934.32	0.00	0.00	18.58	96.14	**25.59**	5532152.47	26.12%	**1211.92**
MATD3		45	**38272.89**	175.42	0.00	**0.00**	**293.33**	**113.24**	**10357169.53**	34.41%	6353.73
MADDPG	45	45	38687.92	**115.07**	0.00.	23.70	581.77	161.11	10349136.17	34.41%	7429.53
MAAC		45	39655.16	246.00	0.00	37.88	377.22	166.28	9853930.08	34.41%	**3019.98**
MATD3		**55**	**43080.70**	**240.93**	55.33	198.38	666.60	**263.90**	**15764665.93**	**42.35%**	8021.78
MADDPG	55	54	44776.63	333.07	**22.66**	327.16	413.00	276.92	14661856.86	41.37%	10425.31
MAAC		54	45418.86	384.26	341.25	**141.50**	**234.00**	268.74	14284950.97	41.12%	**4234.16**
MATD3		**61**	**62853.10**	**271.12**	395.07	**0.00**	765.76	**323.81**	**17830778.79**	**46.72%**	9468.70
MADDPG	65	59	63321.74	412.12	333.58	182.84	818.83	405.03	15868706.46	44.54%	11346.28
MAAC		60	76092.44	840.13	**194.90**	329.94	**655.40**	514.09	15454115.73	45.71%	**5798.93**

3.4.2 实验结果可视化

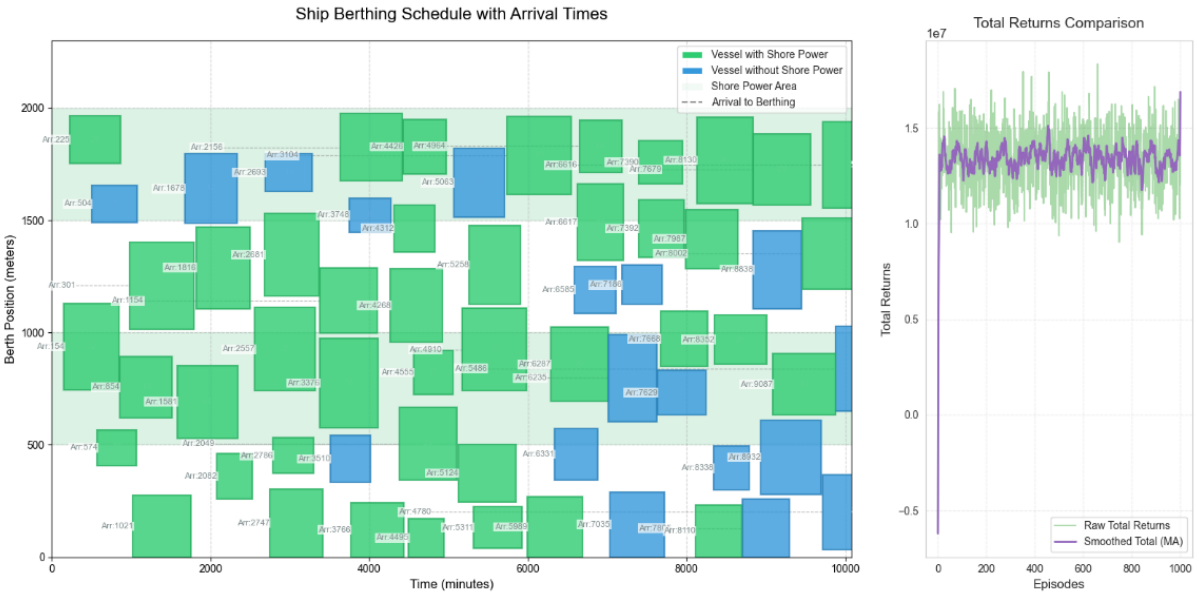


图12 65艘船舶智能体泊位及岸电分配方案

Fig.12 Berth and shore power allocation plan of 65 agents and total rewards

图12是65艘船的泊位分配方案，可以发现：（1）MATD3算法将可使用岸电的船舶分配至岸电覆盖区。（2）船舶在整个岸线实现了从中间（1000m）向两侧（0m和2000m）均匀分布，以最大化泊位利用率。（3）调度方案最小化船舶等待时间。（4）调度方案中无时空重叠问题。（5）图中只有61艘船，有4艘船在该周期内不能靠泊，这四艘船的信息为：

（id=36，$T_{36}^{a}=8124\,\text{min}$，$q_{36}=1$，$L_{36}=165.10m$，$t_{36}^{b}=608.89\,\text{min}$，$p_{36}=1$）

（id=33，$T_{33}^{a}=5986\,\text{min}$，$q_{33}=1$，$L_{33}=262.16m$，$t_{33}^{b}=551.16\,\text{min}$，$p_{33}=2$）

（id=52，$T_{52}^{a}=6731\,\text{min}$，$q_{52}=0$，$L_{52}=298.98m$，$t_{52}^{b}=649.39\,\text{min}$，$p_{52}=2$）

（id=63，$T_{63}^{a}=6970\,\mathrm{min}$，$q_{63}=0$，$L_{63}=152.63m$，$t_{63}^{b}=571.41\,\mathrm{min}$，$p_{63}=2$）。

结合图9可以看出，无法靠泊的四艘船的到港时间有较多船舶同时到达，又因为该四艘船优先级较低，故在该周期内不靠泊。

3.4.3 算法响应时间对比

表7是MATD3 MADDPG MAAC算法前向传播一次的时间的对比表。可以看出，三个算法前向传播的响应时间都在“秒”级，这说明了当算法训练完毕后，将每个周期内的船舶状态信息按1.3.1节输入，船舶调度方案可以在“1秒”内生成。

表7 MATD3 MADDPG MAAC决策时间对比

Table 7 Response time of MATD3, MADDPG, and MAAC

算法	智能体数量	决策时间(s)	算法	智能体数量	决策时间(s)
MATD3		0.099	MATD3		0.1023
MADDPG	15	0.079	MADDPG	25	0.1154
MAAC		0.091	MAAC		0.0913
MATD3		0.1136	MATD3		0.1354
MADDPG	35	0.1399	MADDPG	65	0.2078
MAAC		0.0998	MAAC		0.1380
MATD3		0.1182	MATD3		0.1343
MADDPG	45	0.1489	MADDPG	55	0.1610
MAAC		0.1059	MAAC		0.1073

4 结 论

本文针对连续泊位和岸电协同分配问题，将该问题转化为部分可观测的马尔科夫决策过程，并设计了基于MATD3算法的实时调度框架，求解泊位利用率最高、船舶等待时间最短、碳排放最小的调度方案。本文研究结论如下：

（1）本文构建的“连续泊位与岸电协同优化环境”是稳定的，智能体在此训练可以实现收敛。

（2）本文所设计的5类奖励函数，可有效解决智能体训练中奖励稀疏问题。

（3）MATD3算法具有智能性，即它可定向追踪每个周期内不同属性的船舶，并根据各船舶的相关信息，制定兼顾碳排放、船舶等待时间的泊位分配方案。

（4）通过对比MADDPG、MAAC算法，可以得出采用双Critic可有效解决动作价值高估的问题，且对Actor的延迟更新策略会加速算法训练，可有效节约训练成本。

（5）MATD3算法训练完毕后，可根据港口周期内的船舶状态信息，在充分考虑各船舶优先级、船长、吃水深度等因素下，实现在“1秒”内完成调度方案的制定，且可以根据实时情况动态调整泊位分配方案。这解决了启发式算法、精确算法求解时间长、无法学习过往分配方案经验的问题。

本文研究还有一些不足，未考虑潮汐对泊位分配的影响，未来将在此基础上继续深入探究。

References

[1] Yu J, Voss S, Song X. Multi-objective optimization of daily use of shore side electricity integrated with quayside operation[J]. Journal of Cleaner Production, 2022, 351:131406.

[2] Wang Z , Hu H , Zhen L. Berth and quay cranes allocation problem with on-shore power supply assignment in container terminals[J]. Computers and Industrial Engineering, 2024, 188(02):1-18.

[3] Yu J, Tang G, Voss S, et al. Berth allocation and quay crane assignment considering the adoption of different green technologies[J]. Transportation Research Part E: Logistics and Transportation Review, 2023, 176:103185.

[4] Peng Y, Dong M, Li X, et al. Cooperative optimization of shore power and berth allocation: A balance between cost and environmental benefit[J]. Journal of Cleaner Production, 2020, 279:123816.

[5] Zhen L, Liang Z, Zhuge D, et al. Daily berth planning in a tidal port with channel flow control[J]. Transportation Research Part B: Methodological, 2017, 106:193-217.

[6] Iris C, Lam J S L. A review of energy efficiency in ports: Operational strategies, technologies and energy management systems[J]. Renewable and Sustainable Energy Reviews, 2019, 112:170-182.

[7] Mauri G R, Ribeiro G M, Lorena L A N, et al. An adaptive large neighborhood search for the discrete and continuous Berth allocation problem[J].Computers and Operations Research, 2016, 70:140-154.

[8] Hou, Jue. Dynamic berth allocation problem with two types of shore power for containership based on rolling horizon strategy[J]. IEEE International Conference on Intelligent Transportation Engineering, 2017, 15:144-149.

[9] Wu Y Q, Zhang R. Integrated Optimization of Continuous Berth Allocation and Ship Scheduling Under One-Way Channel[J]. Computer Engineering and Applications, 2022, 58(09):246-255

[10] Jia S, Li C L, Xu Z. Managing navigation channel traffic and anchorage area utilization of a container port [J]. Transportation Science, 2019, 53(3):728-745.

[11] Park, H J, Cho, S W, Lee, C. Particle swarm optimization algorithm with time buffer insertion for robust berth scheduling[J]. Computers and Industrial Engineering, 2021, 160:107585.

[12] Ai T, Huang L, Song R J, et al. An improved deep reinforcement learning approach: A case study for optimization of berth and yard

scheduling for bulk cargo terminal[J]. Advances in Production Engineering and Management, 2023, 18:145-163.
[13] Li C, Wu S, Li Z, Zhang Y, Zhang L, Gomes L. Intelligent scheduling method for bulk cargo terminal loading process based on deep reinforcement learning[J]. Electronics 2022, 11:1390.
[14] Cervellera C, Gaggero M, Macciò D. Policy optimization for berth allocation problems[J]. In Proceedings of the 2021 International Joint Conference on Neural Networks, 2021, 21:1-6.
[15] Lv Y, Zou M, Li J, et al. Dynamic berth allocation under uncertainties based on deep reinforcement learning towards resilient ports[J]. Ocean and Coastal Management, 2024, 252:13-28.
[16] Zhang H, Zhang X H, Feng Z, Xiao X H. Heterogeneous multi-robot cooperation with asynchronous multi-agent reinforcement learning[J]. IEEE Robotics and Automation Letters, 2024, 9(1): 159−166
[17] Li C, Wang R X, Huang J Z, et al. Autonomous Decision－making and Intelligent Collaboration of UAV Swarms Based on Reinforcement Learning with Sparse Rewards[J]. ACTA ARMAMENTARII, 2023,44(06):1537-1546.
[18] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor [J]. International Conference on Machine Learning, 2018, 80:56-69.
[19] Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for mixed cooperative-competitive environments[J]. 31st Annual Conference on Neural Information Processing Systems, 2017, 30:42-61.
[20] Giallombardo G, Moccia L, Salani M, et al. Modeling and solving the tactical berth allocation problem[J]. Transportation Research Part B Methodological, 2010, 44(2):232-245.
[21] Tseng I F, Hsu C H, Yeh P H, et al. Physical mechanism for seabed scouring around a breakwater—a case study in Mailiao port[J]. Journal of Marine Science and Engineering, 2022, 10(10):37-52.
[22] Silver, D., Lever, G., Heess, et al. Deterministic policy gradient algorithms[J]. PMLR, 2014, 52(08):11040