

基于优先级重规划的通信多智能体路径规划

李心怡, 李衍杰[†]

(哈尔滨工业大学(深圳) 机电工程与自动化学院 广东省智能变形机构与自适应机器人重点实验室, 广东 深圳 518055)

摘要: 多智能体强化学习(MARL)因其出色的泛化能力和快速的计算速度, 已成为解决实时性要求高任务的有效工具, 并在多智能体路径规划(MAPF)问题中得到了广泛的研究和应用. 尽管如此, 当智能体密度极高时, 基于强化学习的MAPF求解器仍面临为智能体规划无碰撞路径的难题. 现有的多数学习型MAPF求解器在预测到潜在碰撞时, 倾向于将智能体的移动暂停作为替代动作, 而在智能体之间缺乏有效的协同机制, 这可能引发死锁. 为应对这一挑战, 本研究提出了一种结合了强化学习和带注意力机制的通信网络的方法, 旨在优化MAPF问题的求解. 此外, 为降低高密度智能体环境中死锁的发生率, 本研究设计了一套动作检测与重规划策略. 研究中还引入了一种内部奖励机制, 旨在激励智能体探索环境并加速其达到目标位置的过程. 经过实验验证, 所提出的方法在准确率方面显著优于现有先进的学习型MAPF方法, 并在多个环境中的表现与接近最优解的求解器相媲美.

关键词: 重规划; 优先级; 动作检测; 内部奖励机制; 路径规划

引用格式: 李心怡, 李衍杰. 基于优先级重规划的通信多智能体路径规划. 控制理论与应用, 2026, 43(4): 765 – 773

DOI: 10.7641/CTA.2024.40112

Priority-based replanning for multi-agent pathfinding with communication

LI Xin-yi, LI Yan-jie[†]

(Guangdong Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics, School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen Guangdong 518055, China)

Abstract: Multi-agent reinforcement learning (MARL), with its outstanding generalization capabilities and rapid computation speed, has become an effective tool for solving tasks with high real-time requirements and has been extensively researched and applied in multi-agent path finding (MAPF) problems. Nevertheless, MARL-based MAPF solvers still face the challenge of planning collision-free paths for agents when the agent density is extremely high. Most existing learning-based MAPF solvers tend to pause the movement of an agent as an alternative action when a potential collision is predicted, lacking effective coordination mechanisms among agents, which may lead to deadlocks. To address this challenge, this study proposes a method that combines reinforcement learning with a communication network equipped with an attention mechanism, aimed at optimizing the solution of the MAPF problem. Moreover, to reduce the occurrence of deadlocks in high-density agent environments, this study has designed a set of action detection and replanning strategies. An internal reward mechanism has also been introduced to encourage agents to explore the environment and accelerate the process of reaching their target locations. Experimental validation shows that the proposed method significantly outperforms existing advanced learning-based MAPF methods in terms of accuracy and performs comparably to near-optimal solvers in multiple environments.

Key words: replanning; priority; action detecting; intrinsically motivated; pathfinding

Citation: LI Xinyi, LI Yanjie. Priority-based replanning for multi-agent pathfinding with communication. *Control Theory & Applications*, 2026, 4(4): 765 – 773

收稿日期: 2024-02-25; 录用日期: 2024-11-20.

[†]通信作者. E-mail: autolyj@hit.edu.cn; Tel.: +86 13622316687.

本文责任编辑: 施阳.

深圳市基础研究计划项目(JCYJ20180507183837726, JCYJ20220818102415033, JSGG20201103093802006)资助.

Supported by the Shenzhen Fundamental Research Program (JCYJ20180507183837726, JCYJ20220818102415033, JSGG20201103093802006).

1 引言

多智能体路径规划(multi-agent path finding, MAPF)^[1]的目标是在特定环境中, 给定起始点和目标点, 为多智能体系统规划一组既无碰撞又接近最优的路径集合. MAPF在无人机控制、智能仓储、无人驾驶等多个领域展现出巨大的应用潜力.

鉴于MAPF属于NP难问题^[2-3], 研究者面临若干挑战, 包括复杂约束的处理、多智能体间的互相影响、环境的动态变化等. 目前, 解决MAPF的方法主要分为中心式和分布式两种. 中心式方法能保障最优和完整的结果, 但其计算代价较高, 因为它需要联合所有智能体的状态空间, 并由中心控制器规划最优路径. 随着智能体数量的增长, 中心式求解器的计算量呈指数级增加, 难以应对大规模多智能体系统的MAPF问题. 多智能体强化学习(multi-agent reinforcement learning, MARL)采用分布式强化学习策略, 指导智能体在避免碰撞的同时迅速达到目标^[4-5]. 相对而言, 分布式求解器因其能够允许每个智能体基于局部观察独立预测下一步动作而备受瞩目, 从而能迅速获得次优解, 即以牺牲结果最优性为代价换取速度.

在智能体密度高的环境中, 智能体间的冲突频发, 而目前大部分MAPF求解器^[6-7]在检测到冲突时, 往往仅仅是使智能体选择静止的动作, 而不是采取有效行动, 这种做法容易引发死锁, 导致路径规划失败. 此外, 在分布式系统中, 大部分MAPF求解器缺少优先级规划, 这可能会在智能体相遇时导致缺乏协作的动作决策^[8], 智能体各自追求尽快到达目标, 增加了死锁发生的可能性. 并且在分布式MAPF求解器中, 由于智能体的信息受限, 智能体之间无法进行有效的协同控制.

本研究提出了一种基于优先级的多智能体通信强化学习网络及其重规划机制, 旨在应对MAPF问题. 假设智能体仅能观测到局部环境, 研究采用强化学习方法训练分布式框架以解决MAPF问题. 智能体通过带有注意力机制的通信网络与邻近智能体交换特征向量, 而非简单信息融合^[9], 通过此通信网络促进分布式系统中智能体的协同控制, 并且注意力机制使智能体能够关注对其更有用的邻居信息, 以避免信息过载. 智能体从Q网络中获取各个动作的Q值, 并选择最大Q值的动作执行, 以达到目标点. 训练过程中, 在智能体执行动作之前, 将检测智能体是否选择了无效动作, 如导致碰撞或死锁的动作. 检测到无效动作, 特别是可能导致死锁的动作时, 将施加相应的惩罚措施以减少无效动作的选择. 检测后, 为每个陷入死锁的智能体分配不同的优先级^[10], 并根据优先级顺序进行动作重规划以解锁. 在本研究中, 智能体的优先级是动态

分配的, 取决于智能体完成任务的进度. 此外, 研究团队还设计了内部奖励机制, 鼓励智能体探索环境并加快完成任务. 在该机制中, 智能体若远离起点且靠近目标点, 表明其正在进行有效探索, 因此将获得内部奖励, 以加速任务完成.

本研究的主要贡献如下:

- 1) 设计了动作检测机制, 识别并惩罚可能导致智能体碰撞或死锁的无效动作, 以减少这些动作的选择;
- 2) 提出了基于优先级的动作重规划策略, 为陷入死锁的智能体动态分配优先级, 减少死锁发生并提高高密度智能体环境下的模型性能;
- 3) 引入内部奖励机制, 激励智能体迅速离开起点并向目标点移动, 而非在无效位置徘徊.

2 相关工作

2.1 多智能体路径规划

MAPF求解器可根据解决方法大致分为基于搜索、基于规则、基于优先级和基于学习这4类. 基于搜索的方法将所有智能体耦合起来寻找可行解, 如算子分解^[11]、逐步增长代价树搜索^[12]、增强部分展开A*^[13]、基于冲突的搜索^[14]和M*^[15]. 基于规则的方法将MAPF抽象为图上的石头移动问题, 文献[16]探讨了在图上协调鹅卵石运动的问题, 排列群的直径以及相关应用. Push and swap^[17]方法解决了图上的石头移动问题, 可以在不依赖于图的拓扑结构的情况下保证结果的完整性. Parallel push and swap(PPS)^[18]方法允许所有智能体并行移动. Push and rotate^[19]方法首先将图分割成子图, 然后使用推动、交换和旋转操作来寻找解决方案. MAPP^[20]是在无向图上的可处理的求解器, 其运行时间、内存需求和解决方案长度具有较低的多项式最坏情况上界. PERR^[21]假设当存在路径冲突时, 智能体可以互相交换角色.

基于优先级的方法, 如层次协作A*^[22]和带有时间窗口约束的窗口式WHCA*^[22], 通过为智能体分配优先级来规划路径, 这些方法简单且实用. 优先级的定义是优先级规划的关键, 一些研究采用启发式算法来确定优先级排序^[8, 23]. 然而, Ma等人^[24]指出, 在某些情况下静态优先级无法解决问题, 因此Okumura等人^[10]提出了动态优先级分配方法. 最近, Wang等人^[25]提出了一种基于学习的方法来定义智能体的优先级, 该方法在实验中效果显著, 尽管缺乏理论依据.

2.2 多智能体强化学习(MARL)

基于学习的多智能体路径规划求解器在涉及大量智能体的场景中展现出卓越性能. 它们将在线计算的重负担转移到了离线学习过程中, 并且智能体可以根据局部环境信息做出独立的动作决策, 因此具有高速计算的能力和强实时性. Sartoretti等人^[6]结合了强化

学习和基于ODrM*的模仿学习, 其中使用了异步优势演员-评论家(A3C)算法来训练强化学习的部分, 模仿学习则使用了动态耦合的规划器ODrM*[15]生成的数据进行行为克隆. 然而, 智能体间的通信对于分布式系统来说至关重要. Li等人[19]引入了图神经网络来促进智能体间的通信. 为了进一步优化通信机制, Li等人[26]和Ma等人[7]采用了图注意力网络来加权智能体间的通信内容. DCC[27]中的中央智能体会根据邻居智能体的存在调整决策, 以确定与哪些智能体进行通信. PICO[28]将隐式规划的优先级融入到通信拓扑学习中, 旨在建立动态通信网络并减少冲突. 尽管如此, 智能体间的通信方法可能会影响分布式方法的泛化能力, 特别是当智能体数量增多时, 结果的质量可能会急剧下降[29]. 这表明在设计MARL系统时, 需要特别注意通信机制的设计, 以保持系统在不同规模下的鲁棒性和有效性.

3 基于强化学习的MAPF求解器

3.1 MAPF问题定义

一个MAPF问题可以表示为一个元组 $\langle G, S, T \rangle$, 包含了一个由 n 个智能体组成的集合 $A = \{a_1, \dots, a_n\}$, 其中: $G = (V, E)$ 为一个无向图, V 是无向图中顶点的集合, E 是无向图中边的集合; $S = \{s_1, \dots, s_n\} \subset V$ 表示 n 个智能体的起始点; $T = \{t_1, \dots, t_n\} \subset V$ 表示 n 个智能体的目标点. $\pi_i[t] \in V$ 表示智能体 a_i 在时间步 $t \in \mathbb{N}$ 的位置. 在本研究中, 时间是离散的, 每个时间步智能体只能执行动作集合中的一个动作, 其中包括静止、向上、向下、向左和向右5种动作. 如果两个智能体选择的动作会导致它们在下一个时间步处于同一个栅格, 则称发生了点冲突; 如果它们会交换位置, 则称发生了边冲突. MAPF求解器的目标是在有限的时间步内, 为 n 个智能体找到一条从起始点到目标点的无碰撞、接近最优的路径.

基于强化学习的MAPF求解器可以将MAPF问题抽象为一个序列决策问题. 本研究的目的是训练一个映射 \mathcal{F} , 该映射的输入为智能体的局部观测地图和通信信息, 输出为智能体的动作的 Q 值. 智能体选择具有最大 Q 值的动作, 从而无碰撞地从起始点到达目标点.

3.2 环境设置

MAPF问题被建模为部分可观测的马尔可夫决策过程 (partially observable Markov decision process, POMDP), 表示为元组 $\langle A, S, \{A_i\}, \{O_i\}, \{R_i\}, \mathcal{P}, \gamma \rangle$. S 为状态集合, $\{A_i\}$ 为智能体 a_i 的动作集合, $\{O_i\}$ 为观测集合, $R_i: S \times A \rightarrow \mathbb{R}$ 为奖励函数, \mathcal{P} 为状态转移和观测概率函数, γ 为折扣因子. 每个智能体观察当前的环境, 智能体通过动作策略根据掌握的信息选择动作后, 所有智能体同时执行动作. 然后得到新的奖

励和环境的局部观察, 它们的目标是要最大化各自奖励的期望 $R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots$, 其中 r_t 是时间步 t 智能体接收到的奖励.

在本研究中, 智能体置于大小为 $M \times N$ 的栅格地图中, 地图上随机分布不规则障碍. 环境可用二进制矩阵表示, 0代表可占领位置, 1代表障碍. 每回合开始时, 从有效位置中随机选择 n 个起始点和目标点分配给智能体, 并确保它们之间可达. 与通过强化学习和模仿学习的多智能体路径规划 (pathfinding via reinforcement and imitation multi-agent learning, PRIMAL) 中的单智能体移动不同, 本研究中所有智能体同步执行动作. 此外, 本研究不假设智能体到达目标点后消失.

智能体仅能获取局部环境信息, 观测到的是 $l \times l$ 大小的局部视野. 局部可观测性赋予模型强大的泛化能力, 智能体仅需局部地图即可做出决策. 局部观测地图分为3个通道: 障碍物位置、其他智能体位置和智能体目标点位置, 以3个二进制矩阵表示. 目标点如不在视野内, 以其在视野边界的投影表示. 为指导智能体移向目标, 引入4个启发式通道[7], 分别对应4个移动方向, 若动作使智能体离目标近, 则相应位置置1. 多通道输入降低网络计算量, 提高信息效率.

奖励函数如表1所示. 每时间步, 执行动作后未到达目标点的智能体受到较小的惩罚, 若智能体到达目标点会受到奖励, 以促进其快速到达目标. 而对于发生点冲突、边冲突或者是陷入死锁的智能体施加较大的惩罚, 以避免发生碰撞和死锁. 回合结束条件为所有智能体到达目标或时间达阈值. 为减少死锁, 检测到死锁时对智能体施加额外惩罚, 详见第4.1节.

表1 奖励函数

Table 1 Reward function

动作	奖励
移动(上/下/左/右)	-0.075
停留(在目标上/远离目标)	0.0, -0.075
碰撞	-0.5
死锁	-0.5
完成	3

3.3 Q-网络搭建

本研究采用循环单元增强的Dueling深度Q网络 (Dueling deep Q-network, Dueling DQN) 优化全局信息受限的智能体行为策略, 构建端到端模型, 将局部观测地图和通信信息映射为动作的 Q 值. 在训练分布式模型时, 考虑智能体间通信, 解决局部信息限制和促进无通信模型中缺失的协同合作. 如图1所示, 网络包含特征处理、通信、Q网络3个模块.

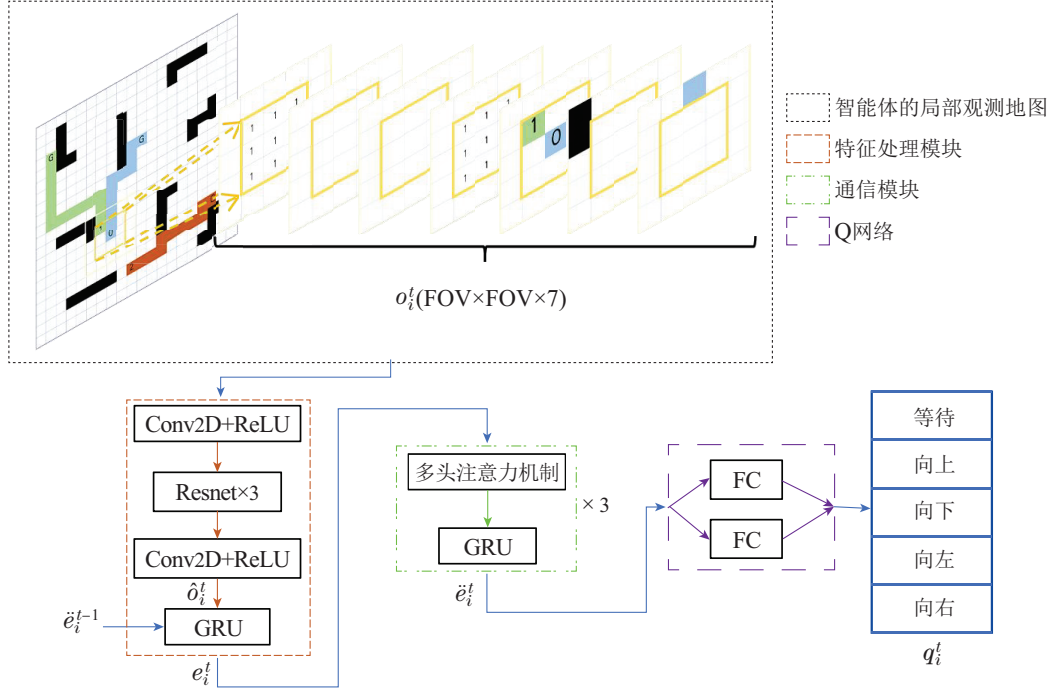


图1 PBRC的网络结构

Fig. 1 Network structure of PBRC

特征处理模块包含8个卷积层和一个门控循环单元(gated recurrent unit, GRU), 卷积层构成3个残差模块和两个独立卷积模块, 每个残差模块包含两个卷积层. 智能体 a_i 将当前时间步局部观测地图 o_i^t 输入特征处理模块, 经过卷积处理后得到 δ_i^t . GRU单元接收 δ_i^t 和上一时间步交流得到的隐藏状态 \hat{e}_i^{t-1} , 输出特征向量 e_i^t . GRU单元能使智能体整合时间序列信息, 优化动作策略.

通信模块结合图卷积网络和GRU单元, 智能体 a_i 的特征向量 e_i^t 与邻居智能体的特征向量 $e_{\mathcal{N}_i}^t$ 经图卷积聚合为中间向量 \hat{e}_i^t , 再与 e_i^t 一同输入GRU单元. 经循环处理后得到信息特征 \hat{e}_i^t , 供下一时刻特征处理模块使用. 信息特征输入Q网络计算动作Q值, 智能体根据最大Q值选择动作.

3.4 基于注意力的通信机制

在分布式系统中, 智能体受限于信息可获得性, 难以实现有效协同. 引入通信机制能够显著改善此限制. 然而, 在高密度智能体环境中, 通信机制可能导致信息过剩, 从而阻碍智能体作出准确决策. 本研究构建了基于Transformer模型^[30]的通信网络, 该模型已在多个人工智能领域展现出卓越性能. 模型运用自注意力机制, 专注于节点的邻居, 以计算节点的隐藏状态. 该策略允许节点对邻居的信息进行加权平均处理, 侧重选择与自身最为相关的节点信息. 研究中的通信模块使智能体能够集中于关键信息, 降低信息噪声的干扰. 智能体被视作图中的节点, 而通信则构成节点间

的边. 若两个智能体的距离落在通信范围之内, 则它们可以进行信息交换. 图注意力机制通过计算智能体信息的相关性来实现, 确保通信网络不仅仅是简单地汇总智能体间的信息, 而是强调信息间的动态相关性.

智能体 a_i 当前时间步的观测地图通过特征处理模块得到的特征向量 e_i^t 被输入到通信模块中, e_i^t 在每个注意力头 h 中通过权重 W_Q 映射为Query, 其他智能体信息通过权重 W_K, W_V 映射为Key和Value. 智能体 a_i 与邻居 $a_j \in \mathcal{N}_i$ 的信息相关性由注意力头 h 计算, 即

$$\alpha_{ij}^h = \text{softmax}\left(\frac{W_Q^h e_i^t \cdot (W_K^h e_j^t)^T}{\sqrt{d_k}}\right), \quad (1)$$

其中 $\sqrt{d_k}$ 为Key维度, 缓解Softmax函数梯度消失问题的缩放因子. 智能体 a_i 的每个注意力头输出 head_h 是基于 α_{ij}^h 的邻居智能体Values的加权和, 即

$$\text{head}_h = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^h W_V^h e_j^t, \quad (2)$$

所有注意力头输出连接后输入图卷积的最后一层, 得到中间向量 \hat{e}_i^t .

智能体通过注意力头 h 计算的到与邻居 $a_j \in \mathcal{N}_i$ 的信息相关性后再与邻居的信息进行聚合, 从而得到更加与自身相关的信息, 避免了信息的冗余.

4 防死锁机制

4.1 无效动作检测

在强化学习方法的训练过程中, 智能体可能会产生无效动作, 这些动作有潜力影响整个系统的性能. 由于智能体独立学习并尝试达到各自的目标点, 循环

死锁(智能体在几个栅格间循环移动)和静止死锁(智能体保持不动)的情况可能发生. 这两种情况都被认为是智能体执行的无效动作. 因此, 模型在输出预测动作后, 需要进行无效动作的检测, 并对执行无效动作的智能体实施动作重规划.

在动作检测环节, 系统首先判定智能体 a_i 是否陷入死锁. 本研究使用一个固定容量的缓存器记录智能体最近5个时间步的位置. 若智能体执行预测动作后, 下一时间步的位置 $\pi_i[t + 1]$ 与缓存器中的某位置重合, 并且缓存器中存在两对以上重合坐标, 则系统判断发生死锁. 对于死锁的智能体, 系统会采取随机策略重新选择动作, 并更新 $\pi_i[t + 1]$. 接着, 系统检查 $\pi_i[t + 1]$ 是否为无效位置, 即检查是否越界或 $\pi_i[t + 1]$ 位置存在障碍物. 若 $\pi_i[t + 1]$ 为无效动作, 则将其设为 $\pi_i[t]$. 若 a_i 与其他智能体发生边冲突或点冲突, 将 $\pi_i[t + 1]$ 标记为 \bigcirc . 最后, 对于 $\pi_i[t + 1] = \bigcirc$ 的智能体, 将执行后续节中描述的重规划(Replan)操作.

4.2 基于优先级的重规划(Replan)机制

随着环境中智能体数量的增加, 动作冲突的概率也相应提高. 大多数基于强化学习的多智能体路径规划(MAPF)求解器在检测到智能体间冲突后, 会用静止动作替代引起冲突的动作, 并对冲突智能体施加严重惩罚, 这可能导致死锁情况增多, 在处理高密度智能体的MAPF问题时, 影响结果质量.

为降低死锁发生率, 引入了基于优先级的重规划(Replan)机制. 每个时间步, 为每个智能体分配唯一优先级. 若两智能体预计进入同一栅格, 优先级高者优先移动, 低者等待. 然而, 仅按优先级顺序处理冲突可能导致死锁, 如图2所示, 优先级最高的智能体 a_1 被优先级最低的智能体 a_2 阻碍, 且 a_2 下移会与 a_3 冲突, 造成系统死锁.

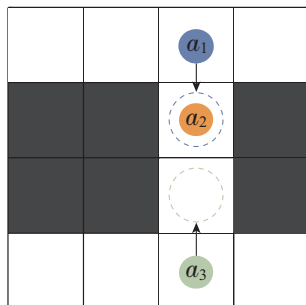


图 2 优先级堵塞示例

Fig. 2 Example of priority blocking

为解决优先级阻塞问题, 引入优先级继承(priority inheritance, PI)机制^[31]. 当优先级低的智能体 a_x 阻碍高优先级智能体 a_y 移动时, a_x 暂时继承 a_y 的优先级. 如图3所示, a_2 继承 a_1 优先级后下移, a_3 因优先级低于 a_2 而等待, a_1 随后下移, 解决死锁. 优先级继承有效解

决优先级阻塞问题, 但不能完全排除死锁.

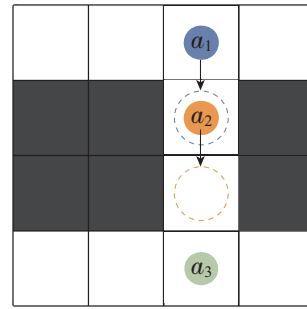
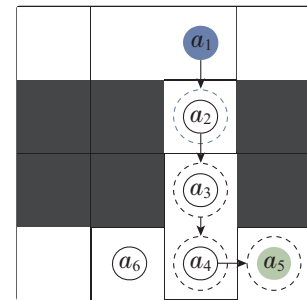


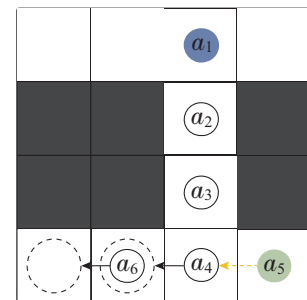
图 3 优先级继承示例

Fig. 3 Example of priority inheritance

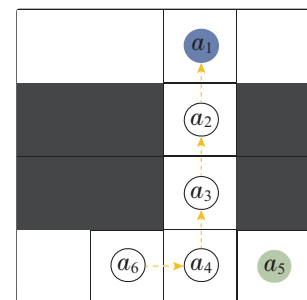
因此, 本研究结合优先级继承与回溯规则^[10]来解决MAPF问题, 以确保智能体路径有效规划, 避免静止死锁. 按回溯规则, 智能体在执行优先级继承时需等待反馈信号. 若反馈表明继承有效, 则移至目标位置; 否则, 寻找其他节点. 如图4所示, 智能体 a_5 给 a_4 发送继承失败信号, a_4 寻找新节点 a_6 , a_6 发回成功信号, a_1 最终移至目标位置.



(a) 智能体 a_5 给 a_4 发送继承失败信号



(b) a_4 寻找新节点 a_6



(c) a_6 发回成功信号

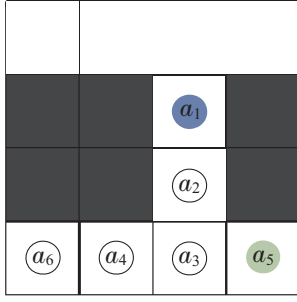
(d) a_1 最终移至目标位置

图4 回溯示例

Fig. 4 Backtracking example

在所有智能体接收到模型预测动作后, 先进行动作检测. 如图5中的检测流程图所示, 首先按照智能体距离目标位置的远近为每个智能体分配优先级 π , 距离目标位置越近的智能体优先级越大. 然后按照优先级降序的顺序对智能体的下一步位置 $\pi_i[t+1]$ 进行检测, 其中检测结果为 $\pi_i[t+1] = \bigcirc$ 的智能体 a_i , 执行 $\text{Replan}(a_i)$.

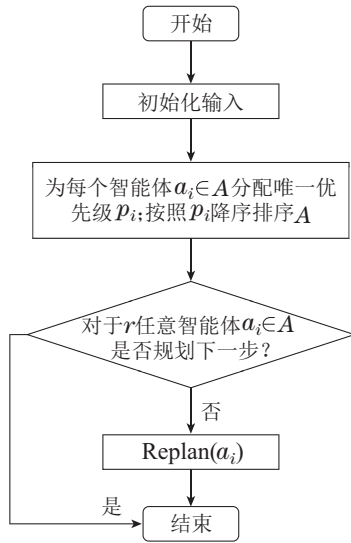


图5 检测流程图

Fig. 5 Detection Flowchart

在 $\text{Replan}(a_i)$ 过程中, 先搜索邻接位置 $C_i = \text{Neigh}(a_i)$, 并对 C_i 节点按距离目标点 g_i 的远近排序, 依次搜索. 若节点 $v \in C_i$ 存在冲突智能体 a_j , 则跳过该节点. 若 v 位置存在优先级低于 a_i 的智能体 a_j , 则调用 $\text{Replan}(a_j)$. 若 $\text{Replan}(a_j)$ 返回有效, 则移动 a_i ; 否则继续搜索. 若成功移动 a_i , 则 $\text{Replan}(a_i)$ 返回有效; 否则返回无效, 并将 $\pi_i[t+1]$ 设为 $\pi_i[t]$. 检测-重规划伪代码如表2所示.

4.3 奖励设置

相较于其他监督学习或非监督学习方法, 强化学习中的智能体最初并不了解所要解决的任务. 智能体必须通过执行动作、观察结果(表现为奖励及状态转

移)与环境互动, 并利用这些信息改进其行为以获取更多奖励. 因此, 如果智能体仅探索环境的一小部分, 则其对环境的认知将受限, 这影响其决策的准确性. MAPF中, 若智能体起始位置与目标点距离较远, 过于谨慎的探索可能导致智能体无法到达目标点.

表2 算法1: 检测-重规划伪代码

Table 2 Algorithm 1: Pseudocode for the detect-replan algorithm

```

Function Replan( $a_i$ )
 $C \leftarrow \text{Neigh}(a_i)$ ;
按照 $\text{dist}(v, t_i)$ 升序排序 $C$ , 其中 $v \in C$ ;
for 每个 $v \in C$  do
  if 存在 $a_k \in A$ 使得 $\pi_k[t+1] = v$  then
    continue;
  end if
  if 存在 $a_k \in A$ 使得 $\pi_k[t] = v \wedge \pi_k[t+1] = \pi_i[t]$  then
    continue;
  end if
   $\pi_i[t+1] \leftarrow v$ ;
  if 存在 $a_k \in A$ 使得
     $\pi_k[t] = v \wedge \pi_k[t+1] = \emptyset$  then
    if  $\text{Replan}(a_k)$ 是无效的then
      continue;
    end if
  end if
  return 有效;
end for
 $\pi_i[t+1] \leftarrow \pi_i[t]$ ;
return 无效;
end function

```

当前许多MARL方法采用随机探索策略, 但此策略常导致训练时间过长, 并可能因频繁选择无效动作而降低结果质量. 在MAPF问题中, 智能体经常接收到稀疏甚至误导性的奖励, 依赖外部奖励的探索型智能体可能因奖励梯度为零而陷入死锁, 或因奖励的误导性而陷入局部最优解.

为激励智能体朝目标点方向探索, 设置了内部奖励机制如下:

$$r_i = \frac{\text{dist}(\pi_i[t-1], t_i) - \text{dist}(\pi_i[t], t_i)}{\text{dist}(s_i, t_i)} \delta, \quad (3)$$

其中

$$\delta = \begin{cases} 1, & \text{dist}(\pi_i[t-1], t_i) > \text{dist}(\pi_i[t], t_i), \\ 0, & \text{其他}, \end{cases} \quad (4)$$

$\text{dist}(a, b)$ 表示 a, b 两点之间的距离. 此奖励机制通过比较智能体在连续两步中至目标点距离的变化来激励智能体向目标点前进. 若智能体到目标点的距离减少, 则奖励为正; 反之, 奖励为零. 该机制亦旨在引导智能体避免在有效位置附近徘徊, 鼓励其探索新状态.

5 实验

在强化学习中, 由于学习过程本质上涉及不断的尝试和优化, 因此一开始就在面积较大且智能体密度高的环境中进行训练是极具挑战性的. 因此, 本研究采用了课程学习(curriculum learning)^[32]的策略, 使得智能体接触的任务难度可以从简单逐步过渡到复杂. 初始环境的地图大小设置为 10×10 , 包含两个智能体. 当训练成功率超过设定的阈值(本研究设定为0.9)则地图大小增加5个单位, 或者增加1个智能体. 训练的最终阶段, 地图大小达到 40×40 , 包含12个智能体. 环境中障碍物的密度是从一个峰值为0.33的三角分布中随机抽取, 该分布的定义域范围为0到0.5. 智能体的局部观测范围和通信范围均被设置为 9×9 .

在智能体密度较高的场景中, 即便是采用注意力机制的通信网络, 信息冗余也是不可避免的. 为了提高通信的有效性, 将智能体的最大通信对象数量限制为5个. 当智能体的通信范围内邻居数量超过5个时, 将根据距离由近及远选择前5个邻居进行通信.

5.1 与其他MAPF求解器比较

本研究将基于优先级重规划的MAPF算法(priority-based replanning for multi-agent pathfinding with communication, PBRC)与4种先进的MAPF求解器进

行比较, 这些求解器包括: 去中心化启发式通信算法(decentralized heuristics communication, DHC)、基于强化和模仿学习的MAPF的可扩展通信(scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding, SCRIMP)、最优降维M*算法(optimal dimensionality reduction M*, ODrM*)^[15]以及为带有回溯的优先级继承MAPF算法(priority inheritance with backtracking for iterative multi-agent path finding, PIBT). 前3种基于学习的算法的训练超参数均与其原始论文中描述一致, 其中DHC的观测范围为 9×9 , SCRIMP的观测范围为 3×3 . 将成功率和平均步长作为评价算法性能的指标, 其中, 若所有智能体在256个时间步内均到达目标点, 则视该案例为成功. 平均步长定义为完成任务所需的每个智能体步长的平均值. 此外, 对所有求解器的求解时间限制为5 min, 超出时间限制则视为规划失败.

表3中的实验结果表明, PBRC算法在平均步长方面表现出色, 尤其是在智能体密度较高的大规模环境中更是如此. 这表明本研究提出的算法能够生成高质量的解决方案, 使得智能体能够在更短的时间内到达目标位置. 在某些情况下, PBRC算法的质量甚至接近于最优的集中式算法ODrM*.

表3 智能体在障碍物密度设置为0.3的两种不同大小的环境中完成任务的平均步数

Table 3 The average number of steps for agents to complete tasks in two environments of different sizes with obstacle density set to 0.3

智能体	地图尺寸 40×40					地图尺寸 80×80				
	ODrM*	DHC	SCRIMP	PIBT	PBRC	ODrM*	DHC	SCRIMP	PIBT	PBRC
4	50.00	54.85	49.93	50.59	50.10	93.40	98.20	94.14	93.67	94.48
8	52.17	61.92	56.35	55.67	55.83	104.92	109.01	108.59	105.31	106.54
16	59.78	72.00	65.15	61.98	63.31	114.75	118.99	116.20	114.04	116.23
32	67.39	81.03	73.92	66.50	71.76	121.31	127.92	129.27	121.72	126.61
64	82.60	117.21	101.28	72.53	87.04	134.42	130.80	144.58	128.80	136.91
128	96.14	—	199.56	87.18	137.46	141.53	209.38	169.00	133.42	145.71

图6展示了3种基于学习的算法及PIBT算法的成功率对比. 橙色线条代表SCRIMP的结果, 蓝色线条代表DHC的结果, 绿色线条代表PBRC的结果, 红色线条代表PIBT的结果.

在智能体数量较少的情况下, 4种算法的成功率均较高. 当智能体密度增加时, DHC和PIBT的成功率急剧下降, SCRIMP的成功率减半, 而PBRC维持了较高的成功率. 这说明PBRC在处理高密度智能体任务方面的性能优于其他3种算法, 并且随着智能体密度及环境规模的增加, PBRC的成功率仍保持在较高水平, 显示了算法具有较强的泛化能力.

5.2 动作滤波与重规划消融

在消融研究中, 本研究将训练过程中集成了死锁检测的PBRC模型与SCRIMP以及未集成死锁检测的PBRC-raw模型在求解过程中产生的死锁数量进行了比较, 结果如图7所示.

随着环境中智能体数量的增加, 3种算法在求解过程中出现的死锁数量也相应增加. 这是因为智能体密度越高, 智能体间发生冲突的概率越大, 从而导致死锁的概率上升. PBRC在3种算法中死锁数量最少, 明显低于未集成动作滤波与重规划机制的PBRC-raw和SCRIMP. 这表明在训练过程中加入动作滤波与重规

划机制能够有效地引导智能体学习避免选择可能引起死锁的动作,并尽快无碰撞地到达目标位置. PB-RC在减少死锁现象方面的显著优势,能够极大提升多智能体系统的工作效率,并确保路径规划的成功率.

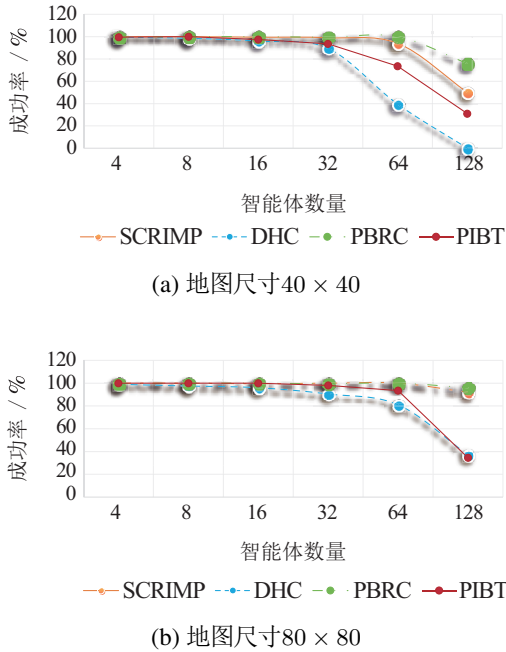


图6 4个算法的成功率对比

Fig. 6 The success rates of four algorithms

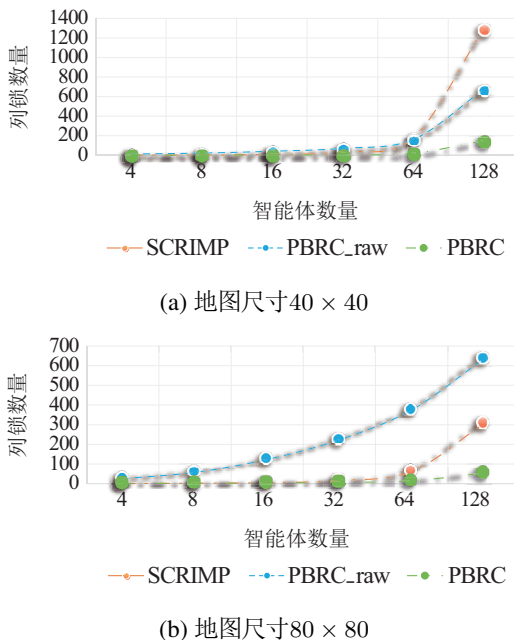


图7 3种算法路径规划过程中的死锁数量

Fig. 7 Number of deadlocks in the path planning process of three algorithms

6 结论

本文针对带通信网络的多智能体路径规划问题(MAPF)引入了动作过滤与基于优先级的重规划机制,旨在提高高智能体密度场景下MAPF求解器的效能.

研究采用了强化学习方法应对MAPF问题,并考虑到智能体间的通信,以解决分布式框架中智能体局限于局部环境信息的问题.在智能体数量众多的环境中,引入了带注意力机制的通信网络,以便智能体更有效地关注对决策有益的信息,避免因信息冗余导致的决策失误.为降低智能体密集环境中死锁的发生概率,提出了动作过滤与基于优先级的重规划机制,首先消除智能体的无效动作,随后通过动态优先级为陷入死锁的智能体重新规划动作策略.此外,为了激励未抵达目标点的智能体积极探索,设置了内部奖励机制,引导智能体快速向目标点移动.实验结果表明,本文提出的模型在性能上优于现有的先进算法,并且接近集中式算法计算得出的最优解.然而,需注意的是,本研究假设智能体间通信无延迟,并且智能体与障碍物均为规则矩形.未来工作将探讨在智能体间通信存在延迟以及智能体与障碍物形状不规则的情况下,如何高效解决MAPF问题,以使研究成果更接近实际应用场景.

参考文献:

- [1] STERN R, STURTEVANT N, FELNER A, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Proceedings of the International Symposium on Combinatorial Search*, 2019, 10(1): 151 – 158.
- [2] YU J. Intractability of optimal multirobot path planning on planar graphs. *IEEE Robotics and Automation Letters*, 2015, 1(1): 33 – 40.
- [3] BANFI J, BASILICO N, AMIGONI F. Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. *IEEE Robotics and Automation Letters*, 2017, 2(4): 1941 – 1947.
- [4] CHEN Y F, LIU M, EVERETT M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. *IEEE International Conference on Robotics and Automation (ICRA)*. Singapore: IEEE, 2017: 285 – 292.
- [5] LONG P, FAN T, LIAO X, et al. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. *IEEE International Conference on Robotics and Automation (ICRA)*. Los Angeles, California, USA: IEEE, 2018: 6252 – 6259.
- [6] SARTORETTI G, KERR J, SHI Y, et al. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters*, 2019, 4(3): 2378 – 2385.
- [7] MA Z, LUO Y, MA H. Distributed heuristic multi-agent path finding with communication. *IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, 2021: 8699 – 8705.
- [8] VAN DEN BERG, JUR P AND OVERMARS, MARK H. Prioritized motion planning for multiple robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, Alberta, Canada: IEEE, 2005: 430 – 435.
- [9] LI Q, GAMA F, RIBEIRO A, et al. Graph neural networks for decentralized multi-robot path planning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, 2020: 11785 – 11792.
- [10] OKUMURA K, MACHIDA M, DÉFAGO X, et al. Priority inheritance with backtracking for iterative multi-agent path finding. *Artificial Intelligence*, 2022, 310: 103752.
- [11] STANDLEY T. Finding optimal solutions to cooperative pathfinding problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010, 24(1): 173 – 178.

- [12] SHARON G, STERN R, GOLDENBERG M, et al. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial Intelligence*, 2013, 195: 470 – 495.
- [13] GOLDENBERG M, FELNER A, STERN R, et al. Enhanced partial expansion A. *Journal of Artificial Intelligence Research*, 2014, 50: 141 – 187.
- [14] SHARON G, STERN R, FELNER A, et al. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 2015, 219: 40 – 66.
- [15] WAGNER G, CHOSSET H. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 2015, 219: 1 – 24.
- [16] KORNHAUSER D M. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. *The 25th Annual Symposium on Foundations of Computer Science*. Singer Island, FL, USA: IEEE, 1984: 241 – 250.
- [17] LUNA R, BEKRIS K E. Push and swap: Fast cooperative pathfinding with completeness guarantees. *International Joint Conference on Artificial Intelligence*. Barcelona, Catalonia, Spain: AAAI, 2011: 294 – 300.
- [18] SAJID Q, LUNA R, BEKRIS K. Multi-agent pathfinding with simultaneous execution of single-agent primitives. *Proceedings of the International Symposium on Combinatorial Search*, 2012, 3(1): 88 – 96.
- [19] DEWILDE B, MORS A W T, WITTEVEEN C. Push and rotate: A complete multi-agent pathfinding algorithm. *Journal of Artificial Intelligence Research*, 2014, 51: 443 – 492.
- [20] WANG K C, BOTEVA A. MAPP: A scalable multi-agent path planning algorithm with tractability and completeness guarantees. *Journal of Artificial Intelligence Research*, 2011, 42: 55 – 90.
- [21] MA H, TOVEY C, SHARON G, et al. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. *AAAI Conference on Artificial Intelligence*, 2016, DOI: 10.1007/10692710_11.
- [22] SILVER D. Cooperative pathfinding. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2005, DOI: 10.1609/aiide.v1i1.18726.
- [23] BENNEWITZ M, BURGARD W, THRUN S. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 2002, 41(2/3): 89 – 99.
- [24] MA H, HARABOR D, STUCKEY P J, et al. Searching with consistent prioritization for multi-agent path finding. *ArXiv Preprint*, 2019, arXiv: 1812.06356.
- [25] WANG Y, XIANG B, HUANG S, et al. SCRIMP: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding. *ArXiv Preprint*, 2023, arXiv: 2303.00605.
- [26] LI Q, LIN W, LIU Z, et al. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters*, 2021, 6(3): 5533 – 5540.
- [27] MA Z, LUO Y, PAN J. Learning selective communication for multi-agent path finding. *IEEE Robotics and Automation Letters*, 2021, 7(2): 1455 – 1462.
- [28] LI W, CHEN H, JIN B, et al. Multi-agent path finding with prioritized communication learning. *International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, 2022: 10695 – 10701.
- [29] GAO J, LI Y, LI X, et al. A review of graph-based multi-agent pathfinding solvers: From classical to beyond classical. *Knowledge-Based Systems*, 2023, 283(11): 111121.
- [30] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, 30: 6000 – 6010.
- [31] ERDMANN M, LOZANO-PEREZ T. On multiple moving objects. *Algorithmica*, 1987, 2: 477 – 521.
- [32] BENGIO Y, LOURADOUR J, COLLOBERT R, et al. Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada: Association for Computing Machinery, 2009: 41 – 48.

作者简介:

李心怡 硕士研究生, 目前研究方向为考虑通信的多智能体路径规划问题, E-mail: 22S153132@stu.hit.edu.cn;

李衍杰 博士, 教授, 目前研究方向为强化学习、逆向强化学习、随机决策与优化、离散事件动态系统、无人机控制、智能自主无人系统, E-mail: autolyj@hit.edu.cn.