

一种基于多步竞争网络的多智能体协作方法

厉子凡, 王 浩, 方宝富

(合肥工业大学 计算机与信息学院, 合肥 230601)

摘 要: 多智能体高效协作是多智能体深度强化学习的重要目标,然而多智能体决策系统中存在的环境非平稳、维数灾难等问题使得这一目标难以实现。现有值分解方法可在环境平稳性和智能体拓展性之间取得较好平衡,但忽视了智能体策略网络的重要性,并且在学习联合动作值函数时未充分利用经验池中保存的完整历史轨迹。提出一种基于多智能体多步竞争网络的多智能体协作方法,在训练过程中使用智能体网络和价值网络对智能体动作评估和环境状态评估进行解耦,同时针对整条历史轨迹完成多步学习以估计时间差分目标,通过优化近似联合动作值函数的混合网络集中且端到端地训练分散的多智能体协作策略。实验结果表明,该方法在6种场景中的平均胜率均优于基于值分解网络、单调值函数分解、值函数变换分解、反事实多智能体策略梯度的多智能体协作方法,并且具有较快的收敛速度和较好的稳定性。

关键词: 多智能体协作;深度强化学习;值分解;多步竞争网络;动作值函数

开放科学(资源服务)标志码(OSID):



中文引用格式: 厉子凡,王浩,方宝富.一种基于多步竞争网络的多智能体协作方法[J].计算机工程,2022,48(5):74-81.

英文引用格式: LI Z F, WANG H, FANG B F. A method for multi-agent cooperation based on multi-step dueling network[J]. Computer Engineering, 2022, 48(5): 74-81.

A Method for Multi-Agent Cooperation Based on Multi-Step Dueling Network

LI Zifan, WANG Hao, FANG Baofu

(School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China)

[Abstract] Multi-agent efficient cooperation is an important goal in Multi-Agent Deep Reinforcement Learning (MADRL); however, environmental non-stationarity and dimensionality disasters in multi-agent decision-making systems render it difficult to achieve this goal. Existing value-decomposition methods can achieve a good balance between environment stationarity and agent scalability. Nevertheless, some value-decomposition methods disregard the importance of the agent-policy network and do not fully utilize the full historical trajectories saved in the experience pool when learning joint action-value functions. Hence, a method for multi-agent cooperation based on Multi-agent Multi-step Dueling Network (MMDN) is proposed herein. First, action estimation and state estimation are decoupled through an independent agent network and a value network during training; additionally, the temporal-difference target is estimated via multistep learning for the entire history trajectory. Second, decentralized multi-agent cooperation policies are trained via a centralized end-to-end mode by optimizing a mixing network that approximates the joint action-value function. Experimental results show that the average winning rate of this method in six scenarios is better than those of multi-agent cooperative methods based on the Value-Decomposition Network (VDN), QMIX, QTRAN, and Counterfactual Multi-Agent (COMA) policy gradient. Additionally, it offers a higher convergence speed and better stability.

[Key words] multi-agent cooperation; Deep Reinforcement Learning (DRL); value-decomposition; multi-step dueling network; action value function

DOI: 10.19678/j.issn.1000-3428.0061437

0 概述

多智能体协作是指多个智能体之间相互合作完

成一项任务或者分别完成复杂任务的某项子任务。

目前,基于深度强化学习(Deep Reinforcement Learning, DRL)^[1]的多智能体协作成为研究热点,已

基金项目: 国家自然科学基金(61876206);中央高校基本科研业务费专项资金(ACAIM190102);安徽省自然科学基金(1708085MF146);民航飞行技术与飞行安全重点实验室开放基金(FZ2020KF15)。

作者简介: 厉子凡(1996—),男,硕士研究生,主研方向为多智能体深度强化学习;王 浩,教授、博士、博士生导师;方宝富,副教授、博士。

收稿日期: 2021-04-25 **修回日期:** 2021-05-31 **E-mail:** fangbf@hfut.edu.cn

在多智能体协同控制^[2]、交通控制^[3]、资源调度^[4]、自动驾驶^[5-6]、游戏AI^[7]等领域得到广泛应用。将DRL与多智能体系统(Multi-Agent System, MAS)相结合,称为多智能体深度强化学习(Multi-Agent Deep Reinforcement Learning, MADRL)^[8]。

多智能体决策系统中主要存在环境非平稳、智能体数量增加导致的维数灾难和多智能体信用分配等问题,这些问题对MADRL而言是巨大的挑战。分散式方法^[9-11]令每个智能体只学习自己的个体动作值函数,并将其他智能体看作是环境的一部分,然后直接应用单智能体深度强化学习(Single-Agent Deep Reinforcement Learning, SADRL)算法学习策略。这样可以避免维数灾难,但由于其他智能体的策略在不断变化,智能体学习到的策略也会随之不断变化,从而出现非平稳特性。集中式方法^[12-14]考虑所有智能体信息直接学习联合动作值函数,可以减轻非平稳性带来的不利影响,但随着智能体数量的增加,参数空间会呈指数级增长,联合动作值函数将难以有效学习并用于智能体数量较多的环境,导致拓展性较差。

近些年来,结合了分散式方法和集中式方法各自优势的值分解方法^[15-17]成为主流方法。值分解方法先分散地学习每个智能体的个体动作值函数,然后集中利用个体动作值拟合联合动作值函数。在此方法框架下,联合动作值函数的计算复杂度随智能体数量呈线性增长,同时也考虑了所有智能体的信息,在环境平稳性和智能体拓展性之间取得了较好的平衡。然而,现有的一些值分解方法忽视了智能体策略网络的重要性,而将研究的重点集中到了联合动作值函数的学习上。此外,在学习联合动作值函数时也没有充分利用经验池中保存的完整历史轨迹,仍然以单智能体常用的单步更新方式学习。

本文提出基于多智能体多步竞争网络(Multi-agent Multi-step Dueling Network, MMDN)的多智能体协作方法,借鉴值分解思想,在集中式训练分散式执行(Centralized Training with Decentralized Execution, CTDE)^[18]框架的基础上,将动作评估与状态估计解耦,利用整条历史轨迹估计时间差分目标,以集中式端到端的方式训练智能体分散策略。

1 相关工作

基于MADRL的多智能体协作方法大致可以分为分散式方法、集中式方法、值分解方法3类。

分散式方法直接应用SADRL算法建模智能体,每个智能体仅学习个体动作值函数,将其他智能体看作是环境的一部分。2017年,TAMPUU等^[9]将深度Q网络(Deep Q Network, DQN)^[19]应用到多智能体环境。同年,GUPTA等^[11]进一步将异步优势行动者-评论家(Asynchronous Advantage Actor-Critic, A3C)算法^[20]、深度确定性策略梯度(Deep

Deterministic Policy Gradient, DDPG)算法^[21]、置信域策略优化(Trust Region Policy Optimization, TRPO)算法^[22]应用到多智能体环境。由于无法解决非平稳性的问题,分散式方法在复杂协作场景中往往无法发挥作用。

集中式方法中每个智能体均利用所有智能体的信息学习联合动作值函数,这样可以减轻非平稳性带来的不利影响,但是存在拓展性的问题,难以用于智能体数量较多的环境。反事实多智能体(Counterfactual Multi-Agent, COMA)策略梯度算法^[13]是基于行动者-评论家(Actor-Critic, AC)框架的算法,所有智能体的Actor网络与一个中心化Critic网络连接。中心化Critic网络使用特殊的反事实模块输出联合优势函数值。由于只有一个中心化的Critic,因此COMA在异构智能体场景中往往无效。多智能体深度确定性策略梯度(Multi-Agent Deep Deterministic Policy Gradient, MADDPG)算法^[12]在DDPG的基础上为每个智能体建立一个中心化的Critic,在训练阶段使用所有智能体的信息而非个体信息以缓解非平稳性,并为每个智能体保留多个子策略。MADDPG不能直接应用于具有离散动作空间的环境。

值分解方法兼具分散式方法和集中式方法的优点,可在环境平稳性和智能体拓展性之间取得平衡。但是值分解方法基于一定的限制条件,多用于完全协作的多智能体任务。值分解网络(Value-Decomposition Network, VDN)算法^[15]将联合动作值函数分解为每个智能体个体动作值函数的简单和,从而将一个复杂的学习问题分解为多个局部的更易学习的子问题。单调值函数分解(QMIX)算法^[16]引入超网络^[23]来学习联合动作值函数与个体动作值函数之间的非线性关系,并限制联合动作值函数和个体动作值函数满足单调约束。值函数变换分解算法(QTRAN)^[17]直接学习联合动作值函数,并构造了多个损失函数用于优化,但该方式难以求解优化问题,并且在复杂任务中很难取得较好的效果。

2 基于MMDN的多智能体协作

2.1 去中心化部分可观察马尔科夫决策过程

完全合作的多智能体任务可以被描述为去中心化部分可观察马尔科夫决策过程(Decentralized Partially Observable Markov Decision Process, Dec-POMDP)^[24]。Dec-POMDP可以定义为一个九元组 $G = \langle N, S, U, P, r, O, Z, n, \gamma \rangle$,其中, $N = \{1, 2, \dots, n\}$ 表示有限数量智能体集合, S 表示环境状态集合, $s \in S$ 表示环境真实状态, U 表示联合动作空间, O 表示联合观察集合, Z 表示观察概率函数, $\gamma \in [0, 1]$ 表示折扣因子。在每一个时间步内,每个智能体 $i \in N = \{1, 2, \dots, n\}$ 选择一个动作 $u^i \in U^i$ 组成联合动作 $u \in U$,环境通过状态转移方程 $P(s'|s, u): S \times U \times S \rightarrow [0, 1]$ 得到

下一步状态 $s', r(s, \mathbf{u})$: $S \times U \rightarrow \mathbb{R}$ 表示奖励函数。

在一个部分可观察的环境中,每个智能体仅能根据观察函数 $Z(s, \mathbf{u})$: $S \times U \rightarrow O$ 得到自己的观察信息 $o^i \in O$ 。每个智能体有自己的动作-观察历史 $\tau^i \in T \equiv (O \times U^i)^*$, 并以此遵循随机策略 $\pi^i(u^i | \tau^i)$: $T \times U \rightarrow [0, 1]$ 。联合策略 π 拥有一个联合动作值函数:

$$Q_{\text{tot}}^{\pi}(\tau, \mathbf{u}) = E_{\tau \in T^N, \mathbf{u} \in U} \left[\sum_{t=0}^{\infty} \gamma^t r(s, \mathbf{u}) \right], \tau \in T^N \text{ 表示一个联合}$$

动作观察历史。

2.2 个体全局最大条件

值分解的核心是将联合动作值函数 Q_{tot} 看作是由每个智能体的个体动作值函数 Q^i 线性或非线性组合而成的(如式(1)所示),直接对联合动作值函数进行优化,通过梯度传播端到端地更新个体动作值函数。

$$Q_{\text{tot}}(\tau, u^1, u^2, \dots, u^n) \approx Q_{\text{tot}}(s, Q^1, Q^2, \dots, Q^n) \quad (1)$$

在值分解框架下,通常将个体全局最大(Individual Global Max, IGM)^[17]作为智能体执行分散策略的条件。该条件确保了对联合动作值函数和个体动作值函数的动作选择保持一致,遵循CTDE框架。

定义 1 对于一个联合动作值函数 $Q_{\text{tot}}(\tau, \mathbf{u})$: $T^N \times U \rightarrow \mathbb{R}$, 如果存在个体动作值函数 $[Q^i(\tau^i, u^i): T \times U^i \rightarrow \mathbb{R}]_{i=1}^n$ 满足式(2), 那么在 τ 下 $[Q^i]$ 对 Q_{tot} 满足 IGM 条件^[17]。在这种情况下, $Q_{\text{tot}}(\tau, \mathbf{u})$ 可以分解为 $[Q^i(\tau^i, u^i)]$ 。

$$\arg\max_{\mathbf{u}} Q_{\text{tot}}(\tau, \mathbf{u}) = \begin{pmatrix} \arg\max_{u^1} Q^1(\tau^1, u^1) \\ \arg\max_{u^2} Q^2(\tau^2, u^2) \\ \vdots \\ \arg\max_{u^n} Q^n(\tau^n, u^n) \end{pmatrix} \quad (2)$$

2.3 多智能体多步竞争网络

在强化学习中,时间差分学习直接从历史经验中学习而无需学习环境的完整知识,可以基于其他状态的估计值来更新当前状态的价值函数,更新规则如下:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_t \quad (3)$$

其中: $\alpha \in [0, 1]$ 表示学习步长; t 表示时间步; δ_t 表示 t 时刻的时间差分误差。

智能体决策具有一定的连续性,若要估计当前决策对所有未来决策的影响,需要对较长的决策序列进行整体考虑,即从经验池中取出整条轨迹时,可以利用当前时间步及之后 n 步的数据进行学习。动作值函数可以用来评估当前的决策对未来的效益,联合动作值函数 Q_{tot} 由个体动作值函数 Q^i 构建,对 Q_{tot} 进行更新可以端到端地训练 Q^i 。基于Q学习的 n 步回报^[25]可表示如下:

$$G_{t:t+n} = \sum_{k=0}^{n-1} \gamma^k r_{t+k+1} + \gamma^n \max_{\mathbf{u}} Q_{\text{tot}}(\tau_{t+n}, \mathbf{u}) \quad (4)$$

值得注意的是, n 步学习可以减少更新目标时的偏差,但会引入高方差^[26]。

为了缓解这一问题,本文引入 λ -回报^[27]作为时间差分目标的估计。 λ -回报可以平均不同 n 的 n 步回报,同时通过调节参数 λ 可以权衡方差和偏差^[28]。 λ -回报定义如下:

$$G_t^{\lambda} = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_{t:t+n} \quad (5)$$

其中: $\lambda \in [0, 1]$ 是调节平均程度的参数。当 $\lambda = 1$ 时,退化到蒙特卡洛方法;当 $\lambda = 0$ 时,退化到一步时间差分方法。换言之, λ 越大,考虑的轨迹越长; λ 越小,考虑的轨迹越短。式(6)的等价写法更能体现这一性质。

$$G_t^{\lambda} = (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t \quad (6)$$

将 λ -回报代入时间差分学习的更新规则后,可以推导出如式(7)所示的联合动作值函数更新规则:

$$Q_{\text{tot}}(\tau, \mathbf{u}) \leftarrow Q_{\text{tot}}(\tau, \mathbf{u}) + \sum_{k=t}^{\min(t+n, T)-1} (\lambda \gamma)^{k-t} \delta_k \quad (7)$$

$$\delta_k = r_{k+1} + \gamma \max_{\mathbf{u}} Q_{\text{tot}}(\tau_{k+1}, \mathbf{u}) - Q_{\text{tot}}(\tau_k, \mathbf{u}_k)$$

为了实现上述更新过程,本文设计如图1所示的MMDN结构,其由3个部分组成:1)估计优势函数的智能体网络;2)估计状态值函数的价值网络;3)估计联合动作值函数的混合网络。

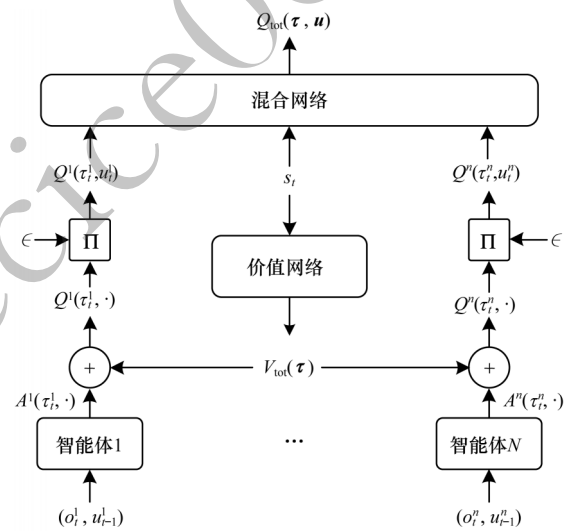


图1 MMDN结构

Fig.1 Structure of MMDN

智能体网络用于学习策略,对动作优劣进行评估,将智能体的观察信息 o^i_t 和上一时刻的动作 u^i_{t-1} 作为输入,将动作优势值 $A^i(\tau^i, u^i_t)$ 作为输出。优势函数定义如下:

$$A(\tau, \mathbf{u}) = Q(\tau, \mathbf{u}) - V(\tau) \quad (8)$$

其中: $Q(\tau, \mathbf{u})$ 表示动作值函数; $V(\tau)$ 表示状态值函数。优势函数用于衡量动作的优劣,因为在同一状态下状态值 $V(\tau)$ 是一个固定的值,所以在优势函数和动作值函数上根据贪婪策略选取最优动作是等价的,

即 $u^* = \underset{u \in U}{\operatorname{argmax}} Q(\tau, u) = \underset{u \in U}{\operatorname{argmax}} A(\tau, u)$ 。智能体网络结构如图2所示,由2个多层感知机(Multilayer Perceptron, MLP)和1个门控循环单元(Gated Recurrent Unit, GRU)组成,激活函数为ReLU,其中 h_t^i 表示智能体 i 在 t 时刻由GRU产生的隐藏状态。

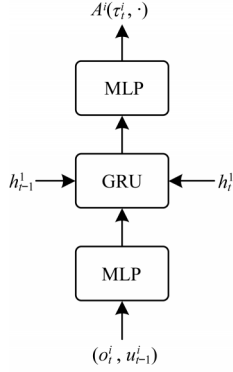


图2 智能体网络结构

Fig.2 Structure of agent network

价值网络用于估计全局状态的优劣,将全局状态 s_t 作为网络输入,将全局状态值 $V(\tau_t)$ 作为输出。MMDN使用智能体网络和价值网络共同估计动作值函数,因此式(8)存在不可辨识的问题^[20],即当 V 和 A 加减同一个常数时, Q 是不变的,但 V 和 A 却可能发生很大的变化。为缓解这一问题,本文在实施过程中采用式(9)计算动作值:

$$Q^i(\tau^i, u^i; \alpha, \beta) = V(\tau; \alpha) + \left(A^i(\tau^i, u^i; \beta) - \frac{1}{|U|} \sum_{a \in U} A^i(\tau^i, a^i; \beta) \right) \quad (9)$$

其中: α 是价值网络的权重参数; β 是智能体网络的权重参数; $Q^i(\tau^i, u^i; \alpha, \beta)$ 表示智能体个体动作值函数的参数化估计; $|U|$ 表示动作空间大小。将使用全局信息估计的全局状态值函数 $V(\tau; \alpha)$ 代替使用每个智能体观察估计的局部状态值函数 $V^i(\tau^i; \alpha)$, 这样做可使 $Q^i(\tau^i, u^i; \alpha, \beta)$ 在训练过程中聚合全局信息,帮助智能体更快更好地学习策略。在测试过程中,价值网络不参与决策,以满足集中式训练分散式执行框架。价值网络结构如图3所示,其由3个MLP组成,激活函数为ReLU。

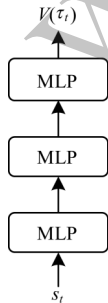


图3 价值网络结构

Fig.3 Structure of value network

混合网络用于拟合联合动作值函数,将每个智能体的个体动作值作为输入,将联合动作值作为输出。本文同样利用超网络训练混合网络的权重参数。混合网络结构如图4所示。超网络是学习神经网络权重参数的神经网络, w_1 和 w_2 即超网络利用全局状态 s 学习的权重, w_1 和 w_2 之间的激活函数为ELU,两个MLP之间的激活函数为ReLU。个体动作值经由两层权重层非线性计算得到联合动作值。

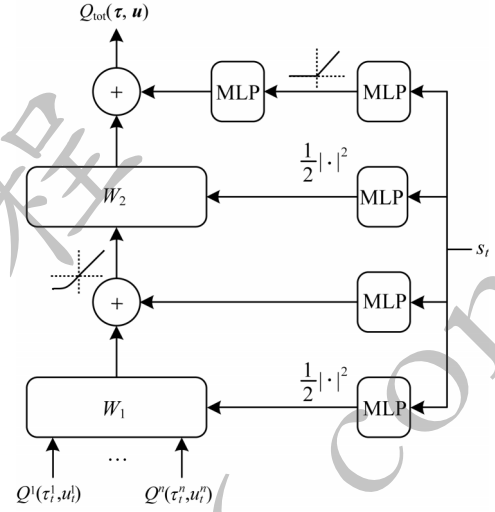


图4 混合网络结构

Fig.4 Structure of mixing network

为了有足够多的训练数据,本文设置一个额外的经验池存储一个情节中所有智能体的历史轨迹。一个情节指从任务开始到任务结束或达到终止条件的整个过程,因此每个情节中的轨迹都是连续的,以方便GRU的训练。

MMDN通过最小化如式(10)所示的损失函数端到端地更新所有模块的网络权重参数:

$$L(\theta) = \sum_{i=1}^b \sum_{t=1}^{T-1} [(y_{i,t}^{\text{tot}} - Q_{\text{tot}}(\tau_{i,t}, u_{i,t}; \theta))^2] \quad (10)$$

其中:时间差分目标 $y_{i,t}^{\text{tot}} = G_t^i$, G_t^i 由权重参数为 θ 的混合目标网络参与计算; b 是从经验池中采样的批大小(batch size); T 是每个情节的最大时间步。

算法1 MMDN训练算法

输入 智能体观察信息 $\{o_t^i\}_{i=1}^n$, 全局状态 s , 经验池大小 d , 输入批大小 b , 训练总情节数 M , 目标网络更新周期 K

输出 智能体目标网络的权重参数 β^-

初始化 经验池 D , 智能体网络的权重参数 β , 价值网络的权重参数 α , 混合网络的权重参数 θ , 对应目标网络的权重参数 $\beta^- = \beta, \alpha^- = \alpha, \theta^- = \theta$

1. for episode = 1 to M do

2. for time step $t = 1$ to $T - 1$ do

3. 得到每个智能体观察信息 o_t^i 和全局状态 s_t

4. 根据每个智能体的 Q^i 利用 ϵ -贪婪策略为每个智能体选择动作 u_t^i 并执行

5. 得到每个智能体的下一步观察信息 o_{t+1}^i 、下一步全局状态 s_{t+1} 和全局奖励 r_{t+1}

```

6.end for
7.将整条历史轨迹存储到D中
8.从D中以均匀分布方式采样b条轨迹作为训练样本,
输入混合网络 $\theta$ 得到 $Q_{tot}$ 
9.使用目标混合网络 $\theta^-$ 计算时间差分目标 $y^{tot}$ 
10.最小化目标函数式(10)以更新网络权重参数
11.在每K个情节后更新目标网络的权重参数 $\beta^-=\beta$ 、
 $\alpha^-=\alpha$ 、 $\theta^-=\theta$ 
12.end for

```

3 实验与结果分析

3.1 实验设置

选择聚焦多智能体微观管理的对抗场景的SMAC^[30]作为基准测试环境,所有方法需要对每个智能体进行细粒度控制,以评估单个智能体能否学会同其他智能体协作完成对战任务。

基于MMDN的多智能体协作方法与基于COMA^[13]、VDN^[15]、QMIX^[16]、QTRAN-base^[17]、QTRAN-alt^[17]的多智能体协作基线方法在8m、2s3z、2s_vs_1sc、MMM、3s5z、1c3s5z等6个场景中进行性能评估。所有方法均与内建的启发式游戏AI进行对抗并计算胜率,内建的AI的难度等级设置为非常困难。实验环境的详细信息和算法复现可以参考文献[30]。

在所有场景的训练过程中,每个智能体分散地使用 ϵ -贪婪策略选择执行动作。随着训练过程的进行, ϵ 在50 000个时间步中从1.0线性衰减到0.05,并在以后的训练过程保持不变。折扣因子 λ 设置为0.99。优化器选用RMSprop,学习率设置为0.000 5。当一方获胜或达到最大时间步后,一个情节终止。所有场景中的最大情节数为20 000,经验池包含最近的4 000条完整历史轨迹。每次更新过程从经验池中均匀地采样32个批量样本,并在完整的历史轨迹上训练。每次训练完100个情节后暂停训练并独立地运行20个情节进行评估,每个智能体分散地使用贪婪策略选择目标动作。测试胜率指算法控制的智能体在一定时间内击败所有敌方单位的情节数占总测试情节数的百分比。目标网络的权重参数为每200个情节更新一次。

3.2 实验结果与消融研究

图5给出了6种方法在6个场景中的评估结果。在每个场景中每种方法按照不同的随机种子运行5次,取5次结果的均值。从图5的实验结果可以看出:

1)COMA的表现相对而言劣于其他基于值函数的方法,在异构环境中均为最差。这也许与其仅有一个中心化Critic不能很好地处理异构智能体信息有关。

2)QTRAN-alt综合而言是基于值函数的方法中性能表现最差的,明显劣于其他方法,在3s5z和1c3s5z两个复杂场景中完全无效。

3)QTRAN-base相比QTRAN-alt表现较好,但在两个复杂场景中也是几乎失效的,原因在于QTRAN相比于VDN和QMIX额外增加了两个损失函数以确保联合动作值函数和个体动作值函数满足文献[17]中定理1或定理2的条件,这样做使得优化问题的复杂度也增加到 $O(|S| \cdot |U|^n)$,其中, $|S|$ 表示状态空间数量, $|U|$ 表示动作空间数量。相比之下VDN和QMIX的优化复杂度从 $O(|U|^n)$ 降低到 $O(n|U|)^{[31]}$ 。SMAC尽管是一个离散动作空间的环境,但是其状态空间是非常大的,这就造成了QTRAN的优化复杂度远高于其他方法,可能出现在计算上难以解决该优化问题,而且场景越复杂,算法性能表现越差。

4)QMIX和VDN的表现接近,在比较复杂的场景中QMIX表现更好,在比较简单的场景中VDN表现更好。笔者认为这是由线性分解和非线性分解的表征能力导致的差异。在复杂场景中,线性分解表征能力受限,不足以很好地学习联合动作值函数,而在简单场景中,线性分解和非线性分解的表征能力没有较大区别,但是非线性分解使用神经网络需要额外的训练,线性分解只需直接进行计算。

5)MMDN相比于基线方法,获得了最好的性能表现,尤其是在复杂的场景中性能提升非常明显。

对MMDN进行进一步的消融研究,在场景2s_vs_1sc和3s5z中验证每个模块的有效性。

消融实验1 探究将动作评估与状态估计解耦的有效性。MMDN取消多智能体竞争网络结构后可被视为引入 λ -回报的QMIX,记作QMIX(λ)。VDN+DN可以视为使用线性混合网络且 $\lambda=0$ 的MMDN。图6结果表明,采用多智能体竞争网络后方法的性能均得到了提升,这说明将动作评估与状态估计解耦有利于智能体做出更好的决策,提升方法的性能表现。

消融实验2 探究不同的 λ 值对MMDN性能的影响。分别选取 λ 等于0、0.4、0.8和0.99进行测试。图7的结果表明,不同的 λ 值会对算法产生不同的影响。 λ 具有平衡偏差和方差的作用,若 λ 值取得太大,则多步估计的权重较高,不能缓解高方差;若 λ 值取得太小,则多步估计的权重较低,具有较大偏差。因此,在本文中选取 $\lambda=0.8$ 。

消融实验3 探究 V 和 A 的不可辨识问题。图8的结果表明,采用式(9)近似动作值函数比直接采用式(8)效果更好,场景越复杂性能差异越明显,说明式(9)确实可以缓解因不可辨识导致的训练不稳定问题。

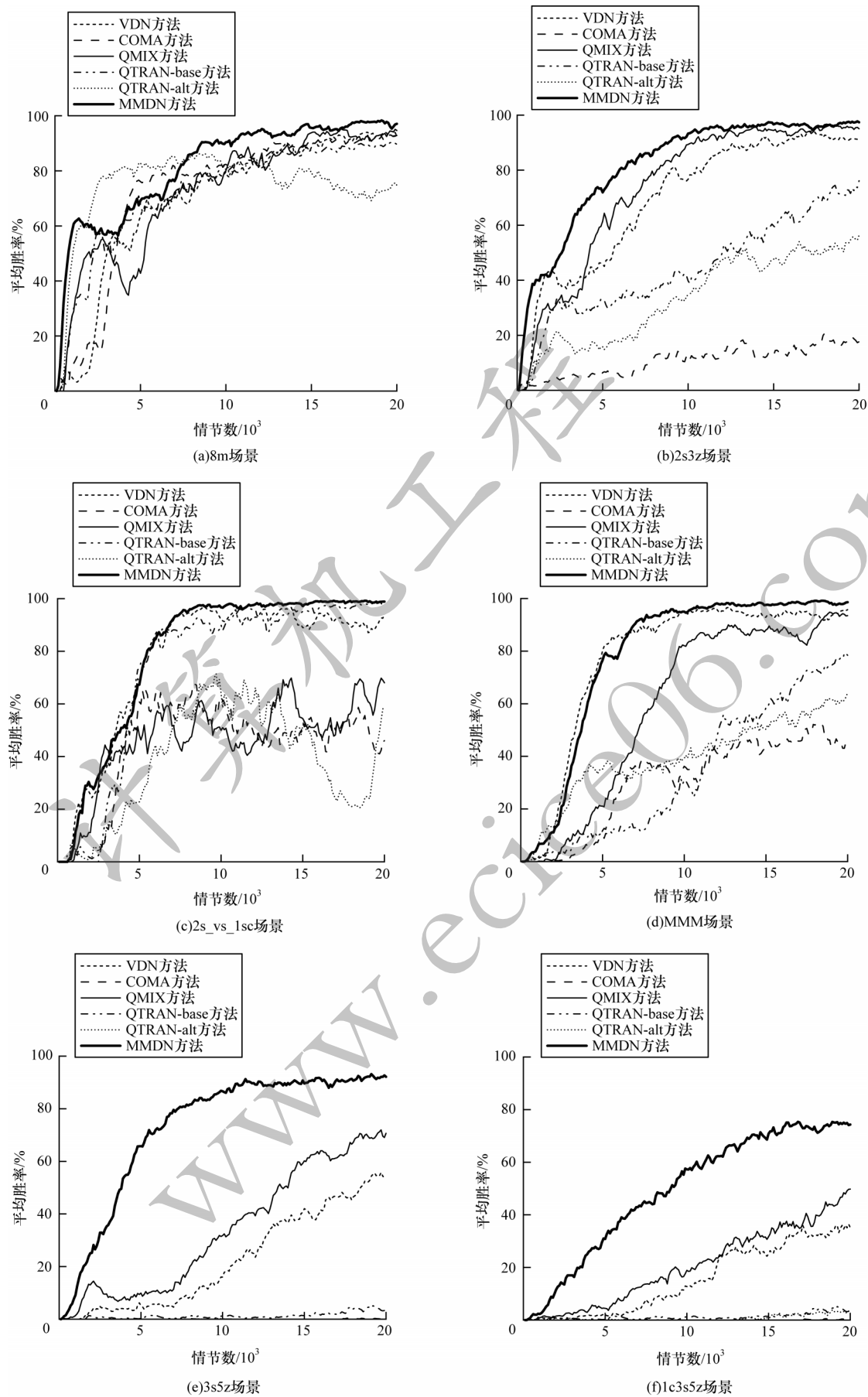


图5 在6种不同场景中的胜率结果

Fig.5 Results of win rates in six different scenarios

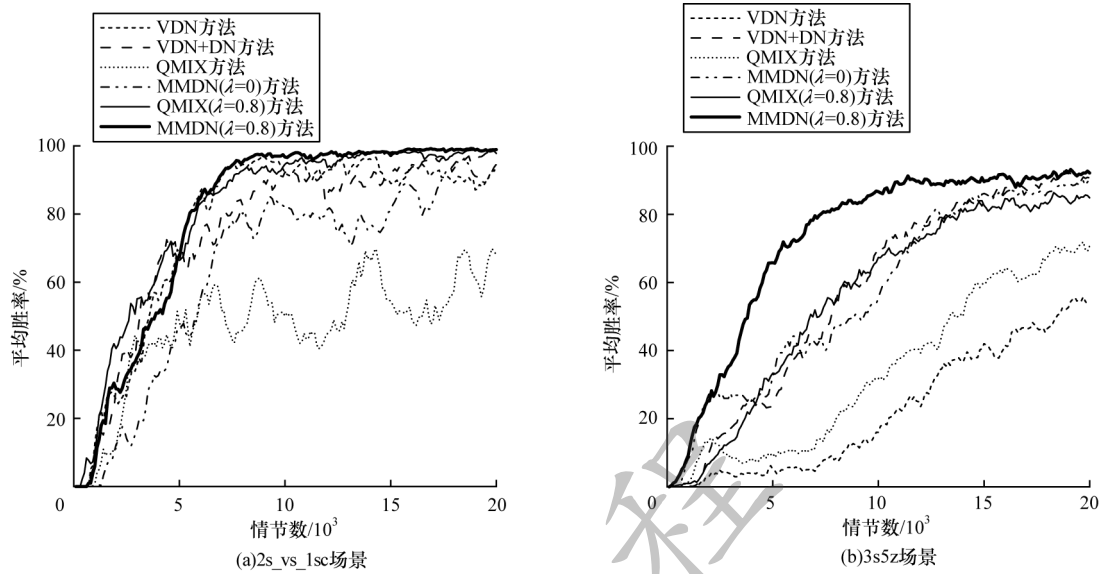


图6 消融实验1的结果

Fig.6 Results of ablation experiment 1

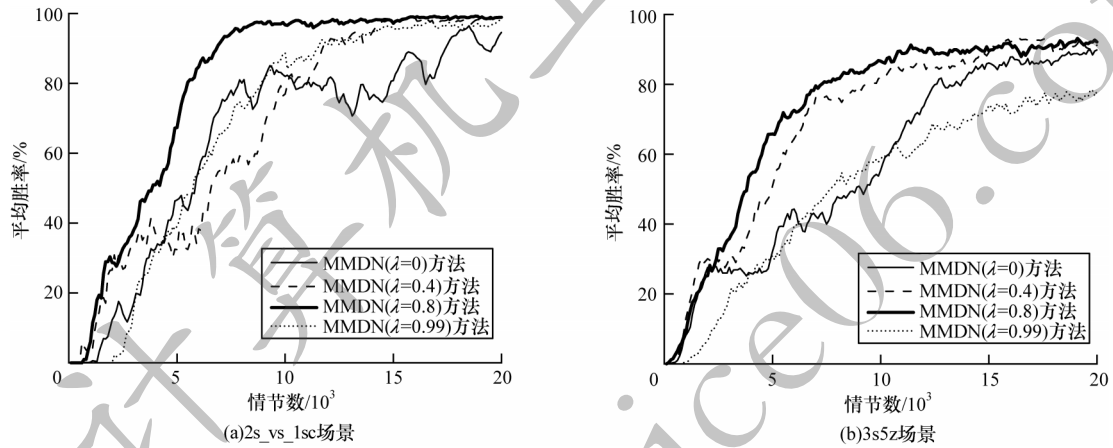


图7 消融实验2的结果

Fig.7 Results of ablation experiment 2

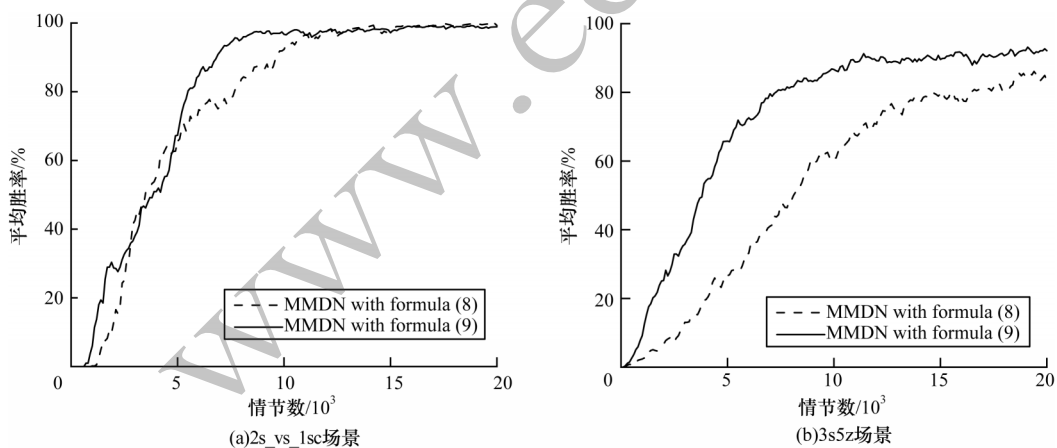


图8 消融实验3的结果

Fig.8 Results of ablation experiment 3

4 结束语

本文提出一个基于MMDN的多智能体协作方法,融合多智能体竞争网络结构、值分解思想和多步时间差分学习,将动作评估与状态估计解耦,充分利

用整条历史轨迹学习联合动作值函数,权衡估计偏差与多步采样带来的方差,并且在环境平稳性和智能体拓展性之间取得较好的平衡,有利于训练与学习多智能体协作策略。实验结果验证了该方法的有效性。下一步将对多智能体竞争网络结构做进一步

改进并拓展到连续动作空间,同时引入协作图、智能体通信等机制,提升其在更为复杂的多智能体协作任务中的性能表现。

参考文献

- [1] ARULKUMARAN K, DEISENROTH M P. Deep reinforcement learning: a brief survey [J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.
- [2] HÜTTENRAUCH M, ŠOŠIĆ A, NEUMANN G. Guided deep reinforcement learning for swarm systems [EB/OL]. [2021-03-17]. <https://arxiv.org/abs/1709.06011>.
- [3] CHU T S, WANG J, CODECÀ L, et al. Multi-agent deep reinforcement learning for large-scale traffic signal control [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(3): 1086-1095.
- [4] 徐西建,王子磊,奚宏生. 基于深度强化学习的流媒体边缘云会话调度策略[J]. 计算机工程, 2019, 45(5): 237-242, 248. XU X J, WANG Z L, XI H S. Session scheduling strategy for streaming media edge cloud based on deep reinforcement learning [J]. Computer Engineering, 2019, 45(5): 237-242, 248. (in Chinese)
- [5] SALLAB A E, ABDOL M, PEROT E, et al. Deep reinforcement learning framework for autonomous driving [J]. Electronic Imaging, 2017, 29(19): 70-76.
- [6] 韩向敏,鲍泓,梁军,等. 一种基于深度强化学习的自适应巡航控制算法[J]. 计算机工程, 2018, 44(7): 32-35, 41. HAN X M, BAO H, LIANG J, et al. An adaptive cruise control algorithm based on deep reinforcement learning [J]. Computer Engineering, 2018, 44(7): 32-35, 41. (in Chinese)
- [7] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning [J]. Nature, 2019, 575(7782): 350-354.
- [8] HERNANDEZ-LEAL P, KARTAL B, TAYLOR M E. A survey and critique of multiagent deep reinforcement learning [J]. Autonomous Agents and Multi-Agent Systems, 2019, 33(6): 750-797.
- [9] TAMPUU A, MATHISEN T, KODELJA D, et al. Multiagent cooperation and competition with deep reinforcement learning [J]. PLoS One, 2017, 12(4): 17-23.
- [10] DE WITT C S, GUPTA T, MAKOVICHUK D, et al. Is independent learning all you need in the starcraft multi-agent challenge? [EB/OL]. [2021-03-17]. <http://arxiv.org/abs/2011.09533>.
- [11] GUPTA J K, EGOROV M, KOCHENDERFER M. Cooperative multi-agent control using deep reinforcement learning [M]. Berlin, Germany: Springer, 2017: 66-83.
- [12] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2017: 6382-6393.
- [13] FOERSTER J, FARQUHAR G, AFOURAS T, et al. Counterfactual multi-agent policy gradients [C]// Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 2974-2982.
- [14] IQBAL S, SHA F. Actor-attention-critic for multi-agent reinforcement learning [C]// Proceedings of the 36th International Conference on Machine Learning. New York, USA: ACM Press, 2019: 2961-2970.
- [15] SUNEHAG P, LEVER G, GRUSLYS A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward [C]// Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. Berlin, Germany: Springer, 2018: 2085-2087.
- [16] RASHID T, SAMVELYAN M, WITT C S, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning [C]// Proceedings of the 35th International Conference on Machine Learning. New York, USA: ACM Press, 2018: 4292-4301.
- [17] SON K, KIM D, KANG W J, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning [C]// Proceedings of the 36th International Conference on Machine Learning. New York, USA: ACM Press, 2019: 5887-5896.
- [18] OLIEHOEK F A, SPAAN M T J, VLASSIS N. Optimal and approximate Q-value functions for decentralized POMDPs [J]. Journal of Artificial Intelligence Research, 2008, 32: 289-353.
- [19] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.
- [20] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning [C]// Proceedings of the 33rd International Conference on Machine Learning. New York, USA: ACM Press, 2016: 1928-1937.
- [21] LILICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning [EB/OL]. [2021-03-17]. <https://arxiv.org/abs/1509.02971>.
- [22] SCHULMAN J, LEVINE S, MORITZ P, et al. Trust region policy optimization [C]// Proceedings of the 32nd International Conference on Machine Learning. New York, USA: ACM Press, 2015: 1889-1897.
- [23] HA D, DAI A M, LE Q V. Hypernetworks [C]// Proceedings of the 5th International Conference on Learning Representations. Amherst, USA: [s. n.], 2017: 1-8.
- [24] OLIEHOEK F A, AMATO C. A concise introduction to decentralized POMDPs [M]. Berlin, Germany: Springer, 2016.
- [25] HESSEL M, MODAYIL J, VAN HASSELT H, et al. Rainbow: combining improvements in deep reinforcement learning [C]// Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 3215-3222.
- [26] JAAKKOLA T, JORDAN M I, SINGH S P. On the convergence of stochastic iterative dynamic programming algorithms [J]. Neural Computation, 1994, 6(6): 1185-1201.
- [27] SUTTON R S, BARTO A G. Reinforcement learning: an introduction [J]. IEEE Transactions on Neural Networks, 2005, 16(1): 285-286.
- [28] KEARNS M J, SINGH S P. Bias-variance error bounds for temporal difference updates [C]// Proceedings of the 13th Annual Conference on Computational Learning Theory. San Francisco, USA: Morgan Kaufmann, 2000: 142-147.
- [29] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning [C]// Proceedings of the 33rd International Conference on Machine Learning. New York, USA: ACM Press, 2016: 1995-2003.
- [30] SAMVELYAN M, RASHID T, DE WITT C S, et al. The starcraft multi-agent challenge [EB/OL]. [2021-03-17]. <http://arxiv.org/abs/1902.04043>.
- [31] MAHAJAN A, RASHID T, SAMVELYAN M, et al. MAVEN: multi-agent variational exploration [C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2019: 7611-7622.