# 强化学习与自适应动态规划: 从基础理论到多智能体系统中的应用进展综述

温广辉[1], 杨 涛[2†], 周佳玲[3], 付俊杰[1], 徐 磊[2]

(1. 东南大学 系统科学系，南京 211189；2. 东北大学 流程工业综合自动化国家重点实验室，沈阳 110819；
3. 北京理工大学 前沿交叉科学研究院，北京 100081)

**摘 要:** 近年来,强化学习与自适应动态规划算法的迅猛发展及其在一系列挑战性问题(如大规模多智能体系统优化决策和最优协调控制问题)中的成功应用,使其逐渐成为人工智能、系统与控制和应用数学等领域的研究热点. 鉴于此,首先简要介绍强化学习和自适应动态规划算法的基础知识和核心思想,在此基础上综述两类密切相关的算法在不同研究领域的发展历程,着重介绍其从应用于单个智能体(控制对象)序贯决策(最优控制)问题到多智能体系统序贯决策(最优协调控制)问题的发展脉络和研究进展. 进一步, 在简要介绍自适应动态规划算法的结构变化历程和由基于模型的离线规划到无模型的在线学习发展演进的基础上,综述自适应动态规划算法在多智能体系统最优协调控制问题中的研究进展. 最后,给出多智能体强化学习算法和利用自适应动态规划求解多智能体系统最优协调控制问题研究中值得关注的一些挑战性课题.

**关键词:** 强化学习；自适应动态规划；多智能体系统；马尔科夫决策过程；序贯决策；最优协调控制

**中图分类号:** TP273 **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2022.1933

**引用格式:** 温广辉,杨涛,周佳玲,等. 强化学习与自适应动态规划: 从基础理论到多智能体系统中的应用进展综述[J]. 控制与决策, 2023, 38(5): 1200-1230.

# Reinforcement learning and adaptive/approximate dynamic programming: A survey from theory to applications in multi-agent systems

*WEN Guang-hui*[1], *YANG Tao*[2†], *ZHOU Jia-ling*[3], *FU Jun-jie*[1], *XU Lei*[2]

(1. Department of Systems Science，Southeast University，Nanjing 211189，China；2. State Key Laboratory of Synthetical Automation for Process Industries，Northeastern University，Shenyang 110819，China；3. Advanced Research Institute of Multidisciplinary Sciences，Beijing Institute of Technology，Beijing 100081，China)

**Abstract:** Reinforcement learning (RL) and adaptive/approximate dynamic programming (ADP) algorithms have recently received much attention from various scientific fields (e.g., artificial intelligence, systems and control, and applied mathematics). This is partly due to their successful applications in a series of challenging problems, such as the sequential decision and optimal coordination control problems of large-scale multi-agent systems. In this paper, some preliminaries on RL and ADP algorithms are firstly introduced, and then the developments of these two closely related algorithms in different research fields are reviewed respectively, with emphasis on the developments from solving the sequential decision (optimal control) problem for single agent (control plant) to the sequential decision (optimal coordination control) problem of multi-agent systems by utilizing these two algorithms. Furthermore, after briefly surveying the structure evolution of the ADP algorithm in the last decades and the recent development of the ADP algorithm from model-based offline programming framework to model-free online learning framework, the research progress of the ADP algorithm in solving the optimal coordination control problem of multi-agent systems is reviewed. Finally, some interesting yet challenging issues on MARL algorithms and using ADP algorithms to solve optimal coordination control problem of multi-agent systems are suggested.

**Keywords:** reinforcement learning；adaptive/approximate dynamic programming；multi-agent system；Markov decision process；sequential decision；optimal coordination control

**收稿日期:** 2022-11-09；**录用日期:** 2023-03-03.

**基金项目:** 国家自然科学基金项目(U22B2046, 62073079, 62088101, 62133003, 61991403, 62173085, 62003167)；装备预研教育部联合基金项目(8091B022114).

†通讯作者. E-mail: yangtao@mail.neu.edu.cn.

## 0 引　言

强化学习(也称增强学习, reinforcement learning, RL)与自适应动态规划(也称近似动态规划, adaptive/approximate dynamic programming, ADP)算法是与现代控制理论诞生的标志之一动态规划(dynamic programming, DP)密切相关的“两类”算法. 近年来, 来自人工智能、系统与控制以及应用数学等不同学科领域的学者对RL和ADP算法的研究形成了交相辉映的成果, 且呈现出融合发展的趋势[1-4], 亦有学者直接将这类算法统称为RLADP算法, 以彰显其核心思想的一致性[3]. 事实上, 本文无意刻意区分这“两类”算法, 与本领域中不少学者具有同样的体会: 融合从不同研究领域产生的先进思想、理论和工程应用进展对于推动RLAPD算法的进一步发展是一件值得期待的事情. 本文主要从人工智能和系统与控制领域分别梳理RL和ADP的研究进展.

RL算法是基于试错学习(trial-and-error learning)思想的一种智能化方法, 通过与环境交互提高系统的决策能力[5], 其形成和发展经历了漫长的过程. “强化”一词最早出现在1927年, 用于描述巴甫洛夫的条件反射实验中动物行为模式的增强[6]. 直到20世纪60年代, “强化”和“强化学习”这些术语才开始用于描述工程应用中的试错学习[7-8]. 进入20世纪80年代, RL算法逐步在马尔科夫决策过程(Markov decision process, MDP)框架下建立起相对严密的数学基础, 并在应用中取得了突破性进展. 在随后的研究历程中, RL算法广泛用于控制和优化博弈等领域, 同时RL算法应用于解决心理学、认知科学、神经科学等学科领域相关问题时呈现出来的交叉研究亦成为另一个研究热点. 近年来, RL算法不断发展, 并在一系列具有挑战性的问题上取得了突破性进展[9-18], 如2016年, DeepMind创建的围棋程序AlphaGo战胜了世界围棋冠军[11-12], OpenAI发布的人工智能算法在游戏Dota 2中击败了人类顶尖玩家[13]. 此外, RL算法还被成功应用于解决智能控制、车间调度、工业制造和能源管理等领域中的优化与决策问题[14-21].

RL算法主要用于序贯决策问题, 可以粗略地分为单智能体强化学习(single-agent reinforcement learning, SARL)和多智能体强化学习(multi-agent reinforcement learning, MARL)算法. 与SARL相比, MARL在解决具有网络耦合(信息交互)特性的(协同)控制与优化决策问题中具有明显潜力和优势, 且广泛应用于自动驾驶[22-23]、智能电网优化调度[24-25]、网络数据传输路由优化[26-27]、多无人系统协同任务[28-30]等领域中. 然而, MARL在解决大规模多智能体系统(multi-agent system, MAS)优化与协调控制任务中仍然面临许多新的挑战. 将RL推广到多智能体形式的两种主流方法是独立学习[31]和联合学习[32]. 在独立学习框架下, 每个智能体独立地执行RL算法, 将其他智能体视为环境的一部分. 在联合学习框架下, 通常将整个MAS视为一个智能体, 所有智能体的联合动作视为一个动作, 学习联合的状态-动作值函数($Q$函数). 基于上述两种方法, 一些学者围绕降低联合学习的算法复杂度或缓解独立学习的非平稳性提出了一些改进型MARL算法. 此外, 见诸文献的还有一些学习方式介于独立学习与联合学习之间的MARL算法. 总体而言, 对MARL算法的研究当前仍处于蓬勃发展的阶段, 相关研究工作主要围绕如下问题开展: 降低通信成本和计算资源消耗; 克服局部可观测的限制; 提升算法对大规模智能体系统的可扩展性; 有效处理环境非平稳性对算法的影响; 提高算法对恶意攻击的鲁棒性和安全性; 有效保护智能体局部信息的隐私性等.

在RL算法基础理论与应用技术发展过程中, 人工智能领域以及系统与控制领域的研究人员均发挥了重要作用. 一定程度上讲, RL算法是在DP的基础上, 通过放松DP方法对环境模型的依赖和解决维数灾难问题发展而来. 事实上, 针对系统动力学模型已知的复杂系统最优控制问题, 理论上可以由DP算法求解, 但是计算量随状态变量数量增加而呈指数级增长的“维数灾难”问题导致最优控制律(策略)的精确解常常难以获得. 在这一背景下, 系统与控制领域的学者利用RL的思想和技术并结合最优控制理论寻求近似的求解方法, 即ADP方法, 以获得使闭环系统拥有足够好性能的次优策略. ADP融合了RL、神经网络和自适应控制等理论与方法, 为解决复杂系统最优控制问题提供了新的思路.

从研究历程看, 20世纪50年代Bellman[33]在研究无后效性多阶段决策过程的优化问题时, 提出了著名的贝尔曼最优性原理(即多阶段决策过程的最优策略具有如下性质: 无论过程中前序状态和决策如何, 对前序决策所形成的状态而言, 后续诸决策必须构成最优策略), 从而创立了动态规划. 基于贝尔曼最优性原理, 将多阶段决策问题转化为求解哈密顿-雅可比-贝尔曼(Hamilton-Jacobi-Bellman, HJB)方程问题, 进而得到最优控制律. 然而, 针对实际系统最优控制问题所获得的HJB方程大多不具有解析解, 采用迭代更新的方法近似寻求HJB方程的数值解成为利用

DP方法求解最优控制问题的一个主要手段. 值得一提的是,几乎所有的DP方法都可以看作是广义策略迭代(generalized policy iteration, GPI)算法[5],算法实施中策略评估与策略改进两个流程以某种粒度交替进行. 在这一框架下,策略迭代(policy iteration, PI)算法[5]和值迭代(value iteration, VI)算法[5]是最受欢迎的两种. 需要说明的是,DP方法是一种离线规划(off-line planning)的方法,需要知道完备的系统动力学信息且计算复杂度极高,面临着“维数灾难”难题. 为了减弱算法对系统模型的依赖,研究者们提出了先识别系统动力学模型或参数,后利用DP方法求解相关HJB方程的思路. 然而,以这种方式设计的最优控制律对系统模型的参数变化反应较为缓慢,且基于系统辨识得到的系统参数可能不够准确[34]. 针对上述局限,Werbos[35-38]基于神经网络等函数近似器,开创性地提出了ADP方法并逐渐成为系统与控制领域的重要研究方向. ADP方法是一种在线学习(online learning)方法,其基于沿着系统轨迹收集的数据,利用函数近似结构以在线更新的方法寻求HJB方程的近似解,随后获得系统的近似最优控制策略,能够有效解决未知系统的优化控制问题,在无人驾驶汽车、机器人、工业制造、物流交通、经济金融、智慧教育等领域具有广泛的应用潜力. 一定程度上,ADP算法的提出及其广泛应用就像是对“control is dead”这一误解的一次不屈回击.

进入21世纪,传感技术、通信技术和分布式控制技术的快速发展对系统集成与管控方式产生了深刻的影响. 在这一背景下,从大规模国家基础设施到形形色色的嵌入式集成系统日益呈现出网络化的结构特征,将工程系统(尤其是具有网络耦合特征的工程系统)建模为MAS成为系统分析与综合的先决条件. ADP在解决形形色色的MAS最优协调控制问题方面显示出巨大的潜力,如MAS最优状态一致性问题(或者状态同步问题)[39-54]、MAS输出同步和输出调节问题[55-65]、MAS $H_\infty$追踪控制问题[66-70]和跨域无人系统编队控制问题[71-79]等.

RL和ADP在人工智能领域的理论与应用发展和在系统与控制领域的理论与应用发展,共同促成了相关研究领域欣欣向荣的局面,国内外已有不少学者从不同角度对RL和ADP算法的理论研究及其在MAS中的应用开展过文献调研与综述[80-84]. 然而,现有关于RL算法研究及其在MAS中应用研究方面的综述大多针对某一具体问题/算法的研究开展综述,如MAS 深度强化学习,针对利用ADP算法求解MAS最优协调控制问题这一研究领域的进展综述还较少见诸文献. 进一步地,系统性地阐述RL和ADP这两个密切相关的算法在不同研究领域的发展历程和梳理其在MAS中的应用进展,对于RL和ADP算法的进一步融合发展以及推动相关算法在MAS任务中的应用研究大有裨益. 本文旨在系统性地综述RL和ADP从应用于单个智能体(控制对象)序贯决策(最优控制)问题到MAS序贯决策(最优协调控制)问题的发展脉络和近期研究进展(尤其是2020年至今的一些最新研究进展),以期为相关科研人员和工程技术人员提供较为系统性的参考. 具体地,从控制中心与智能体以及智能体之间信息交互模式的角度对已有合作型、竞争型和混合型MARL算法进行更为细致的分类和讨论. 进一步地,在简要介绍ADP算法的结构变化历程和由基于模型的离线规划到无模型的在线学习发展演进的基础上,综述ADP算法在MAS最优协调控制问题中的研究进展.

## 1 单智能体强化学习

RL算法是解决序贯决策问题的一种智能化方法,学习和决策者通过与环境交互来学习最优策略.

### 1.1 MDP 模型基础知识

在RL算法中,学习和决策者称为智能体(agent),智能体之外所有与其相互作用的事物称为环境. 智能体与环境的交互可以描述为如下过程:智能体在$t$时刻感知环境的状态$s_t$并选择要执行的动作$a_t$,环境从当前状态转移到一个新的状态$s_{t+1}$,然后将相应的即时奖励$r_{t+1}$反馈给智能体. 智能体的目标是学习一个策略来合理地选择动作,以实现长期的累积奖励(又称回报)最大化. RL问题的马尔科夫性是指系统在下一时刻的状态只与当前时刻的状态和动作有关,而与之前的状态无关. 对于环境(系统)状态变换和奖励更新具有马尔科夫性质的序贯决策问题,人们通常将其建模为MDP,并定义为五元组$(\mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R},\gamma)$. 其中:$\mathcal{S}$为状态集,$\mathcal{A}$为动作集,$\mathcal{T}$为状态转移概率函数,$\mathcal{R}:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\boldsymbol{R}$为奖励函数(符号$\boldsymbol{R}$代表实数域),$\gamma$为折扣因子. 在更一般情形中,执行动作所获得的奖励服从一个概率分布. 策略是从状态到每个动作选择概率之间的映射,记为$\pi(a|s)$. 特别地,确定性策略为从状态到动作的映射.

### 1.2 价值函数

智能体与环境持续进行交互,并获得如下轨迹:

$$\langle s_0,a_0,r_1,s_1,a_1,r_2,\ldots,s_{t-1},a_{t-1},r_t,\ldots\rangle. \tag{1}$$

在每个离散时刻(步数)$t = 0, 1, \ldots$, 智能体在状态$s_t$下执行动作$a_t$, 获得奖励$r_{t+1}$, 并进入一个新的状态$s_{t+1}$. 状态值函数$v^\pi(s)$为从状态$s$出发, 执行策略$\pi$所获得的期望累积折扣奖励, 可以表示为

$$v^\pi(s) = E_\pi\Big[\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|s_0 = s\Big]. \tag{2}$$

状态动作值函数$q^\pi(s,a)$为从状态$s$出发, 采取动作$a$后再遵循策略$\pi$所获得的期望累积折扣奖励, 可以表示为

$$q^\pi(s,a) = E_\pi\Big[\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|s_0 = s, a_0 = a\Big]. \tag{3}$$

最优策略$\pi^*$下的最优状态值函数和最优状态动作值函数分别记为$v^*$和$q^*$, 满足如下Bellman最优等式:

$$\begin{cases} v^*(s_t) = \max\limits_{a_t} E[r_{t+1} + \gamma v^*(s_{t+1})], \\ q^*(s_t, a_t) = E[r_{t+1} + \gamma \max\limits_{a_{t+1}} q^*(s_{t+1}, a_{t+1})]. \end{cases} \tag{4}$$

下文中, 符号$V$和$Q$分别代表算法对状态值函数和状态动作值函数的估计.

### 1.3　单智能体强化学习算法

SARL算法(即通常文献中提及的RL算法)主要分为表格型RL、基于值函数近似RL和基于策略梯度RL, 算法分类见表1. SARL从有模型的算法逐渐发展到无模型的算法, 为了进一步解决大规模和连续动作、状态空间下的优化决策问题, 又从表格型算法发展到值函数近似和策略梯度算法, 并最终发展成将二者有机融合的演员-评论家(actor-critic, AC)算法.

**表1**　SARL算法分类

| 算法类型 | 算法名称 |
|---|---|
| 表格型[85-88] | MC[5], TD[85], $Q$-learning[87], Sarsa[86] |
| 值函数近似[89-96] | DQN[90], Double DQN[91], Dueling DQN[92], Rainbow[93] |
| 策略梯度[97-108] | TRPO[100], PPO[101], SAC[102], DPG[105], TD3[107], A3C[108] |

#### 1.3.1　表格型

表格型RL算法指价值函数用表格型表示的算法, 适用于状态和动作空间较小的问题. 表格型RL算法主要包括蒙特卡罗(Monte Carlo, MC)算法、时序差分(temporal difference, TD)算法等. 这些算法均以DP思想为基础, MC思想的引入克服了环境模型未知的难点, 而TD学习则结合了DP与MC, 通过引入自举法(Bootstrapping)降低计算复杂度, 是RL核心的思想之一. 这些方法可直接从与环境互动的经验中学习策略, 无需构建表征环境动态特性的模型.

TD思想[85]基本形式如下:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]. \tag{5}$$

与MC算法在一幕结束计算回报后再进行更新不同, TD算法采用$r_{t+1} + \gamma V(s_{t+1})$近似回报, 在下一时刻即可进行更新, 具有更高的学习效率. 典型的TD算法包括Sarsa、$Q$-learning算法等.

所有的控制学习方法都面临一个困境: 在采样过程中, 为了获得更多的奖励要尽可能选择当前的最优动作, 而为了探索潜在的更优动作又要尽可能多地尝试其他动作. 探索(exploration)和利用(exploitation)不可避免地相互矛盾, 这就是RL面临的"探索-利用困境"(exploration-exploitation dilemma). 平衡探索和利用的方法包括加噪声、乐观初始化、概率分配等不同类型的方法. $\varepsilon$-贪心策略是一种加噪声的方法, 它是解决探索利用问题的一个常用方法, 基于概率平衡探索和利用, 以$\varepsilon$的概率进行探索, 以$1-\varepsilon$的概率进行利用.

为了保证算法收敛, 需保证在采样幕数趋于无穷时每个动作状态二元组都被访问无数次, 关于探索与利用的探讨也源自于此. 对该问题的解决方案, 根据执行策略、待评估和改进策略的异同, 又可分为同轨策略(on-policy)和离轨策略(off-policy)方法. 在同轨策略方法中, 用于生成采样数据序列的行动策略与用于实际决策的待评估和改进的目标策略是相同的, 而在离轨策略方法中则是不同的.

Sarsa算法[86]是同轨策略下的TD算法, 其更新规则如下:

$$\begin{aligned} &Q(s_t, a_t) \leftarrow \\ &Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \end{aligned} \tag{6}$$

具体地, 生成采样数据序列时, 采用基于当前$Q$函数的$\varepsilon$-贪心策略; 在上述评估策略的更新式中, $a_{t+1}$的选择也为基于当前$Q$函数的$\varepsilon$-贪心策略.

$Q$-learning[87]是离轨策略下的TD算法, 其更新规则如下:

$$\begin{aligned} &Q(s_t, a_t) \leftarrow \\ &Q(s_t, a_t) + \alpha[r_{t+1} + \\ &\gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \end{aligned} \tag{7}$$

具体地, 生成采样数据序列时, 采用基于当前$Q$函数的$\varepsilon$-贪心策略; 在上述评估策略的更新式中, $a_{t+1}$采用当前$Q$函数下的最优动作.

Sarsa和$Q$-learning是两种流行的TD类算法,但二者的适用场景和优缺点不同[88]. Sarsa是同轨策略算法,其目标策略与行为策略相同,因此在探索中更加保守,更适用于在线学习场景. 而$Q$-learning是离轨策略算法,其目标策略与行为策略不同,在探索中更加激进,一般仅适用于离线学习场景. 相比于$Q$-learning, Sarsa求解的是次优解决方案,但其在线性能更好.

### 1.3.2 值函数近似

在具有大规模或连续状态和动作空间的强化学习任务中,表格型方法将面临巨大计算和存储资源需求所带来的困境. 在此背景下,函数逼近方法被引入到强化学习中,其基本思想是:用一个带参数$w$的可微函数$Q(s,a;w)$逼近价值函数$Q(s,a)$,根据随机梯度下降法,沿样本均方误差对参数的负梯度方向进行如下更新:

$$w \leftarrow w + \beta_1 \cdot [U_t - Q(s_t,a_t;w)]\nabla_w Q(s_t,a_t;w). \tag{8}$$

其中:$\beta_1$为参数$w$的更新步长;$\nabla_w Q = \dfrac{\partial Q}{\partial w}$为$Q$对$w$的梯度;$U_t$为对$Q(s_t,a_t)$的估计值,这里可以取MC估计或TD估计.

值函数估计的近似模型有广义线性近似、决策树、神经网络等. 广义线性近似模型可以收敛到全局最优,但表示能力有限且需要根据先验知识选取适合任务的特征,常见的特征包括多项式基、傅里叶基、瓦片编码、粗编码等[5,89]. 神经网络模型是当前研究的热点,利用神经网络的非线性表达能力对价值函数进行拟合并取得了良好的效果. 2015年, Mnih等[90]提出的深度$Q$网络(deep $Q$-network, DQN)结合深度学习和RL,实现了直接从图像等高维感官数据中进行RL. 该算法利用神经网络对价值函数进行近似,并分别提出了经验重放和目标网络来提高数据利用率、降低样本相关性,实现了训练稳定性和估计精度的提高. 这里,RL算法稳定性的高与低刻画的是价值函数或其参数估计值方差的大与小,方差越小稳定性越高. 此外,双DQN(double DQN)[91]、决斗DQN (dueling DQN)[92]等基于DQN的改进算法也相继被提出.

Rainbow算法集成了6种DQN的改进方法,结合多种算法的优势,性能得到了大幅提高[93-96]. 深度RL的发展标志着RL的研究和应用进入了新的阶段.

### 1.3.3 策略梯度

前述算法几乎都基于动作价值函数,即先学习动作价值函数,根据估计的动作价值函数选择动作. 与基于动作价值函数的方法不同,策略梯度(policy gradient, PG)方法直接学习策略,用神经网络等参数化函数拟合最优策略. 该方法能够直接学习选择动作的概率,实现合理程度的试探并逐步接近确定性的策略,自然地处理连续状态空间.

具体地, PG方法[97]用参数化的可微函数$\pi(s,a;\theta)$直接逼近最优策略函数$\pi(s,a)$,学习的目标由性能指标$J(\theta)$描述,可定义为

$$J(\theta) = E\Big[\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|\pi_\theta\Big], \tag{9}$$

其中$\pi_\theta$为$\pi(s,a;\theta)$的简写. 为了最大化$J(\theta)$,利用梯度上升法对参数$\theta$进行更新,有

$$\theta \leftarrow \theta + \beta_2 \cdot \nabla_\theta J(\theta). \tag{10}$$

其中:$\beta_2$为参数$\theta$的更新步长,$\nabla_\theta J$为$J$对$\theta$的梯度. 根据PG定理[5],随机性策略的梯度如下:

$$\nabla_\theta J(\theta) \propto Q^{\pi_\theta}(s,a)\cdot\nabla_\theta \log\pi(s,a;\theta). \tag{11}$$

由于价值函数的真值$Q^{\pi_\theta}(s,a)$未知,用估计值$U_t$代替$Q^{\pi_\theta}(s,a)$. REINFORCE算法[98]和AC算法[99]分别使用MC估计和单步TD估计计算$U_t$.

在REINFORCE算法中,$U_t$取MC估计$G_t$. 在PG的更新中增加一个基线,不会改变更新项的期望,但可以降低估计值的方差,带基线的REINFORCE更新如下:

$$\nabla_\theta J(\theta) = [G_t - b(s)]\cdot\nabla_\theta\log\pi(s,a;\theta), \tag{12}$$

其中$b(s)$为基线,其常用取值为状态值函数的参数化估计.

Konda等[99]结合值函数近似和PG两种学习方法提出了AC学习框架,其中逼近策略函数的参数函数称为Actor,逼近价值函数的参数函数称为Critic,于是同时估计价值函数和策略函数的方法称为AC类算法. 与单纯的PG算法不同的是,Actor部分的参数更新不再利用MC方法,而是利用TD方法对价值函数进行估计.

最原始的AC类算法利用自举法估计$U_t$,相对于REINFORCE算法降低了价值函数估计值的方差. 在AC算法中,价值函数$V(s)$参数化近似为$V(s;\bar{w})$,$U_t$取单步TD估计,以$V$函数为基线的AC更新如下:

$$\nabla_\theta J(\theta) = [r + \gamma V(s_{t+1};\bar{w}) - V(s_t;\bar{w})]\cdot \nabla_\theta\log\pi(s,a;\theta). \tag{13}$$

近年来, PG算法不断发展,产生了很多新算法. 只逼近价值函数的算法包括REINFORCE[98]、TRPO[100]、PPO[101]等,同时逼近价值函数和策略函数

的算法(即AC类算法)包括AC[99]、SAC[102]、NAC[103]、GAE[104]、DPG[105]、DDPG[106]、TD3[107]、A3C[108]等.

PG算法对算法的稳定性和收敛性带来了挑战. 为了改进随机性策略梯度方法,确定性策略梯度(deterministic policy gradient, DPG)方法被提出[105],事实上它是随机策略梯度方法的一个特例. DPG 算法相对随机策略需要的样本数量更少,也更容易训练. 设确定性策略函数为$\mu : \mathcal{S} \to \mathcal{A}$并参数化为$\mu(s;\bar{\theta})$,确定性策略梯度的更新如下:

$$\nabla_{\bar{\theta}} J(\bar{\theta}) \propto \nabla_a Q(s,a;w)\cdot \nabla_{\bar{\theta}}\mu(s;\bar{\theta}). \tag{14}$$

A3C算法[108]用多个智能体并行地与环境进行交互以加快学习速度,但本质上解决的仍是SARL问题. 该算法在多个并行的线程中收集经验样本,然后在一个中心控制器中对共享的参数进行异步更新.

## 2　多智能体强化学习

### 2.1　多智能体马尔科夫决策过程

在MARL问题中,MDP被扩展为多智能体马尔科夫决策过程(multi-agent Markov decision process, MAMDP). 考虑一个包含$N$个智能体的系统,$\mathcal{N} = \{1,2,\ldots,N\}$为智能体的集合,MAMDP可以描述为$(\mathcal{S},\{\mathcal{A}^i\}_{i\in\mathcal{N}},\mathcal{T},\{\mathcal{R}^i\}_{i\in\mathcal{N}},\gamma)$[109]. 具体地,$\mathcal{S}$为所有智能体共享的全局状态空间,$\mathcal{A}_i$为智能体$i$的动作空间,$\mathcal{T}$为状态转移概率函数,$\mathcal{R}_i : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to \boldsymbol{R}$为智能体$i$的局部奖励函数,$\gamma$为折扣因子. 每个智能体观测全局状态$s$并获得局部观测值$o_i$,执行局部动作$a_i$,获得个人奖励$r_i$. 所有智能体的联合动作记为$a = [a_1,a_2,\ldots,a_N]$,联合动作空间记为$\mathcal{A} = \mathcal{A}_1\times\mathcal{A}_2\times\ldots\times\mathcal{A}_N$. 状态转移函数$\mathcal{T}:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$和每个智能体的个体奖励$r_i$都取决于当前状态和所有智能体的联合动作. 每个智能体的个体策略为$\pi_i : \mathcal{S}\times\mathcal{A}_i\to[0,1]$,它们共同构成联合策略$\pi$.

在不同的问题背景中,环境状态的可观测性不同. 在完全可观测的环境下,每个智能体可以观测到全部的环境信息,即$o_i = s$,但在部分或局部可观测的环境下,每个智能体只能获得部分环境信息.

### 2.2　多智能体强化学习算法

从任务类型角度,MARL可以分为完全合作型、完全竞争型和混合型3种. 在完全合作型任务中,所有智能体的目标完全相同,因此奖励函数也完全相同,即$\mathcal{R}_1 = \mathcal{R}_2 = \ldots = \mathcal{R}_N$;在完全竞争型任务中,智能体一般分为两个阵营,其目标完全相反;在混合型任务中,智能体之间兼有合作和竞争. 对不同类型的任务,很难定义统一的评价指标,现有的学习准则包括收敛性、合理性和安全性等.

从学习方法角度,SARL到MARL两种自然的推广是联合学习和独立学习[32]. 联合学习将所有智能体视为一个整体,所有智能体的联合动作视为一个动作,学习联合状态动作价值函数. 在独立学习中,每个智能体独立地执行RL算法,而将其他智能体视为环境的一部分.

从通信角度,根据控制中心与智能体以及智能体之间信息交互模式的角度可对已有合作型、竞争型和混合型MARL算法进行更为细致的分类,算法分类见表2. 现有的信息交互模式分为集中式学习、分散式学习和分布式学习3种类型. 集中式学习拥有一个集中的控制中心,控制中心与所有的智能体进行通信,相应的通信和计算成本高昂,同时在恶意攻击和通信受限的情况下鲁棒性较弱. 在分散式学习中,不同的智能体各自进行决策而不进行任何通信,独立学习是典型的分散式学习算法. 集中式学习大多采用联合学习的方式进行学习,对通信条件的要求高,随着智能体数量的增加还会陷入维数诅咒;而分散式学习的智能体可能受到局部观测的限制,同时可能面临着环境非平稳导致的收敛性挑战. 一些研究基于上述两种方法提出一些改进方法,以降低集中式学习的算法复杂度或缓解分散式学习的非平稳性,包括集中式训练分散式执行(centralized training with decentralized execution, CTDE)框架、基于局部通信的分布式学习框架等. 总的来说,集中程度更低的算法的可扩展性更好,但更容易遇到非平稳性挑战;相反,集中程度更高的算法以复杂度为代价缓解非平稳性

**表2　MARL算法分类**

| 算法类型 | 主要算法名称 |
|---|---|
| 集中式学习[110-114] | 联合$Q$学习[32], RIAL/DIAL[111], CommNet[112], MAPPO[110], IRAT[113] |
| 分散式学习[115-129] | IQL[32], 乐观$Q$学习[116], 阻滞$Q$学习[117-118], 宽容$Q$学习[119-121], Minimax-$Q$学习[122] |
| 集中式训练分散式执行(CTDE)[130-142] | VDN[130], QMIX[131], QTRAN[132], MADDPG[135], COMA[136], MAAC[137] |
| 集中式训练分布式执行[143-149] | TarMAC[143], BiCNet[145], ATOC[146], IC3Net[148], I2C[149] |
| 分布式训练分布式/分散式执行(DTDE)[150-167] | 基于一致性的算法[150-153], 平均场$Q$学习算法和平均场AC算法[154-156], DTDE框架[162] |

注: 集中式指具有集中的控制中心与多智能体进行全通信,分散式指智能体之间不进行任何通信,分布式指智能体之间进行局部或部分的通信.

问题. 当前,基于CTDE的RL算法从集中学习中提取分散策略还未有最佳方法,基于局部通信的通用分布式RL算法的研究也尚不充分.

### 2.2.1 集中式学习

集中式学习算法通常包含一个中央控制中心,该控制中心能与所有智能体通信,收集它们的信息并集中地进行决策. 在集中式学习算法中,智能体之间可以交互各自对环境状态的观测值、各自的动作、奖励和策略等,甚至交互和共用价值函数.

在完全合作性任务中,所有智能体共享同一个奖励值. 联合学习算法将所有智能体看作一个整体,直接学习联合$Q$函数. 通常情况下,SARL算法都可以自然地扩展为联合学习形式并应用于求解完全合作型任务问题,例如联合$Q$学习[32]和MAPPO[110]分别是$Q$-learning和PPO的联合学习形式. 联合学习是集中式学习算法的一种典型方法,但是联合学习算法的学习空间随智能体数目增加呈指数增长,算法复杂度随之急剧上升. 为了降低算法复杂度,人们进一步提出了一些高效的联合学习算法,如RIAL/DIAL[111]、CommNet[112]等. RIAL/DIAL[111]对DQN进行了通信方面的扩展,智能体在集中式训练中学习信息编码方式和策略,根据局部观测、传输及收到的信息执行动作,其中DIAL在训练中允许梯度在智能体之间传递. CommNet[112]使用通信信道连接所有智能体,每个智能体接收其他智能体发送的信息并进行信息融合以降低维度,再根据融合后的信息和自己的观测进行决策. 为了解决MARL场景中可能存在的个体奖励稀疏问题,IRAT[113]通过分别构建个体策略和团队策略为个体构建稠密的奖励. 为了改善多个智能体在复杂环境中的协调性,Yuan等[114]提出了多智能体激励沟通(multi-agent incentive communication, MAIC)框架,该框架中每个智能体利用激励通信机制权衡通信信息的重要性,并学习和预测队友的策略,从而实现有效的协调.

在完全竞争型或混合型任务中,一般不存在集中式学习算法. 因为不同智能体的利益并不是完全一致的,不会进行完全的信息共享,也不会进行集中式通信.

表3对现有主流的一些集中式训练-集中式执行类算法进行了总结和对比.

**表3 集中式训练-集中式执行类算法总结和对比**

| 算法名称 | 算法类型 | 适用任务 | 算法结果 |
|---|---|---|---|
| 联合$Q$学习 | 无模型,表格型,TD类,离轨策略 | 合作型 | 将$Q$-learning扩展为联合学习形式 |
| MAPPO | 无模型,PG,MC类,离轨策略 | 合作型 | 将PPO扩展为联合学习形式 |
| RIAL/DIAL | 无模型,值函数/PG,通用通信协议 | 合作型 | 在集中式训练中学习信息编码方式和策略 |
| CommNet | 无模型,通用通信协议 | 合作型 | 利用神经网络和均值进行信息融合,再根据融合后的信息进行决策 |
| IRAT | 无模型,PG,MC类,离轨策略 | 合作型 | 针对奖励稀疏问题,通过个体策略和团队策略构建稠密的奖励 |
| MAIC | 无模型,通用通信协议 | 合作型 | 利用激励通信机制权衡通信信息的重要性,学习和预测队友的策略 |

### 2.2.2 分散式学习

每个智能体学习各自的价值函数并独立地进行决策,多个智能体之间不进行任何信息交互的算法称为分散式算法.

为了克服集中式算法通信成本高、可扩展性差的缺陷,分散式学习算法可以在没有通信的情况下求解最优策略,每个智能体基于局部信息学习局部$q^{(i)}$函数和个体最优策略. 在分散式学习过程中,每个智能体获得其自身的观测,并调整局部策略以最大化累积奖励. 相对于集中式算法,分散式算法具有通信成本更低、计算复杂度更低、对恶意攻击和单点失效的鲁棒性更高等优点.

在完全合作型任务中,分散式学习让每个智能体根据自己的局部观测执行各自的动作,尽可能使得构成的联合动作全局性能最优. 通常情况下,SARL算法都可以自然地扩展为独立学习形式并应用到合作型任务中,例如IQL[31]和IPPO[115]. IQL[31]是$Q$-learning的独立学习形式,更新如下:

$$q^{(i)}(o_{i,t},a_{i,t})\leftarrow r_{t+1}+\gamma\max_{a_{i,t+1}}q^{(i)}(o_{i,t},a_{i,t+1}).\quad(15)$$

其中:$o_{i,t}$为智能体$i$对环境状态$s_t$的局部观测,$r_{t+1}$为合作型任务中所有智能体共享的奖励. 分散式算法常应用在环境状态完全可观测的问题中,即$o_{i,t}=s_t$. IPPO[115]是PPO的独立学习形式,每个智能体只估计其局部值函数并独立地进行学习,在星际争霸等场景中其性能可以达到甚至超越最先进的联合学习方法,如MAPPO和QMIX. 其中,MAPPO和QMIX分别是集中式学习算法和CTDE算法,在本文相关部分已有介绍,而分散式算法IPPO的计算复杂度更低.

为了克服独立学习算法面临的局部可观测性、非马尔科夫性、环境非平稳性等挑战,一些针对独立学习算法的改进算法相继被提出,包括乐观$Q$学习[116]、阻滞$Q$学习[117-188]、宽容$Q$学习[119-121]等. 非马尔科夫性指不满足马尔科夫性质,即当前状态向下一状态的概率转移可能与历史状态和历史动作有

关. 乐观$Q$学习[116]利用联合学习在乐观假设下的投影来构造分散式算法,由于在所考虑的问题中,环境状态是完全可观测的,即$o_{i,t}=s_t$,值函数更新如下:

$$q^{(i)}(s_t,a_{i,t})\leftarrow \max\{q^{(i)}(s_t,a_{i,t}),r_{t+1}+\gamma\max_{a_{i,t+1}}q^{(i)}(s_{t+1},a_{i,t+1})\},\tag{16}$$

其中$r_{t+1}$为合作型任务中所有智能体共享的奖励. 然而,当最优联合动作不唯一时,乐观$Q$学习可能因为在最佳联合动作上协调不当而无法收敛到最佳联合策略. 为了克服乐观$Q$学习的缺陷,阻滞$Q$学习[117-118]在容忍其他智能体探索的同时,对协调不当施加少量惩罚. 宽容$Q$学习[119-121]进一步改进了阻滞$Q$学习的探索效率,在早期对其他智能体的探索更加宽容,随着策略收敛逐渐降低宽容性.

在完全竞争型任务中,不同智能体的目标完全相反. 智能体学习在最坏情况下最小化可能损失的策略,于是可以利用Minimax原理刻画智能体的学习目标,得到Minimax-$Q$学习算法[122]. 由于在所考虑的问题背景中,环境状态是完全可观测的,即$o_{i,t}=s_t$,于是智能体$i$按下式更新$Q_i$函数:

$$Q_i(s_t,a_{i,t},a_{-i,t})\leftarrow (1-\alpha)Q_i(s_t,a_{i,t},a_{-i,t})+\alpha[r_i+\gamma\max_{a_{i,t+1}}\min_{a_{-i,t+1}}Q_i(s_{t+1},a_{i,t+1},a_{-i,t+1})].\tag{17}$$

其中:$r_i$为智能体$i$获得的奖励,$a_{-i,t}$为$t$时刻智能体$i$对手的动作,$\delta=r_{t+1}+\gamma\max_{a_{i,t+1}}q^{(i)}(s_{t+1},a_{i,t+1})-q^{(i)}(s_t,a_{i,t})$. Fan等[123]将Minimax原理与DQN相结合,提出了Minimax-DQN算法. M3DDPG算法[124]将Minimax原理与MADDPG相结合来提高算法在竞争型任务中的鲁棒性,从而将此类算法扩展到求解连续动作问题中.

在混合型任务中,不同智能体之间兼有合作和竞争,每个智能体的目标是在其他智能体的策略给定时,学习最佳响应策略以最大化自己的回报. 在博弈论中,满足如下条件的联合策略$(\pi_1^*,\dots,\pi_N^*)$或$(a_1^*,\dots,a_N^*)$构成均衡解[125]:

$$Q_i(s,a_i^*,a_{-i}^*)\geqslant Q_i(s,a_i,a_{-i}^*),\ \forall i\in\mathcal{N},\tag{18}$$

其中$a_{-i}$为除智能体$i$外其他智能体的动作. 均衡解的概念在一定程度上合理地刻画智能体的学习目标,针对不同系统,纳什均衡、相关均衡、贝叶斯均衡、Stacklberg均衡等不同均衡解的定义被引入来定义新的$Q$函数[125-129]. 由于在所考虑的问题背景中,环境状态是完全可观测的,即$o_{i,t}=s_t$,具体的更新形式如下:

$$Q_i(s_t,a_t)\leftarrow(1-\alpha)Q_i(s_t,a_t)+\alpha[r_i+\gamma\tilde{Q}_i^*(s_{t+1})].\tag{19}$$

其中:$r_i$为智能体$i$获得的奖励,$\tilde{Q}_i^*(s)$为所有智能体采取某种均衡策略对应的值函数. 例如,纳什均衡$\tilde{Q}_i^*(s)$定义为

$$\tilde{Q}_i^*(s)=\max_{a_i}Q_i(s,a_1^*,\dots,a_i,\dots,a_N^*).\tag{20}$$

智能体$i$通过观测到的信息学习每个智能体的$Q$表以推断他人的决策,其中$a_{j\neq i}^*$是根据表$Q_j$计算得到的其他智能体的均衡策略.

表4对现有主流的分散式训练-分散式执行类算法进行了总结和对比.

**表 4　分散式训练-分散式执行类算法的总结和对比**

| 算法名称 | 算法类型 | 适用任务 | 算法结果 |
|---|---|---|---|
| IQL[31] | 无模型,表格型,TD类,离轨策略 | 合作型 | $Q$-learning的独立学习形式 |
| IPPO[31] | 无模型,PG,MC类,离轨策略 | 合作型 | PPO的独立学习形式 |
| 乐观/阻滞/宽容$Q$学习[116-121] | 无模型,表格型,TD类,离轨策略 | 合作型 | IQL的改进方法,降低非平稳性的影响 |
| Minimax-$Q$学习[122] | 无模型,表格型,TD类,离轨策略 | 竞争型 | 利用Minimax原理刻画竞争型任务的学习目标 |
| 纳什$Q$学习[125-129] | 无模型,表格型,TD类,离轨策略 | 混合型 | 利用均衡解的概念刻画混合型任务的学习目标 |

### 2.2.3　集中式训练分散式执行

集中式算法在训练和执行时都是集中式的,分散式算法在训练和执行时都是分散式的. 为了克服完全集中式和完全分散式算法各自的缺陷,CTDE类算法被提出并不断发展. CTDE是目前常用的MARL学习框架,该框架假设在训练阶段智能体之间没有通信限制,于是每个智能体可以获得所有智能体的观测、动作以及环境的全局状态,而在执行阶段每个智能体只能根据局部观测进行决策. 通过集中$Q$函数选出的最优联合动作与通过单个智能体的$q^{(i)}$函数选出的最优动作联合之间的一致性,被称为个体-全局最大值(individual-global maximum, IGM)原则,形式如下:

$$\arg\max_a Q(\tau,a)=\begin{bmatrix}\arg\max_{a_1}q^{(1)}(\tau_1,a_1)\\ \vdots\\ \arg\max_{a_N}q^{(N)}(\tau_N,a_N)\end{bmatrix}.\tag{21}$$

CTDE框架中需要解决的问题是如何从集中式$Q$函数中提取分散式$q^{(i)}$函数,使得集中式策略与分散式策略相一致,即满足IGM原则.

典型的CTDE算法包括VDN[130]、 QMIX[131]、QTRAN[132]、Qatten[133]、Qplex[134]等基于值函数的算法,以及MADDPG[135]、COMA[136]、MAAC[137]等基于PG的算法. 除MADDPG外,现有的CTDE算法基本都是解决合作型任务的. MADDPG算法假设其他智能体的动作是可观测的,每个智能体对其他智能体的策略进行推测,所以可以同时适用于合作型、竞争型和混合型任务.

基于值函数分解的MARL算法主要包括VDN、QMIX、QTRAN、Qatten和Qplex. VDN算法[130]提出可加性假设,即联合$Q$值是每个智能体的$q^{(i)}$值的线性和,即

$$Q(\tau,a)=\sum_i q^{(i)}(\tau_i,a_i), \tag{22}$$

其中$\tau_{i,t}=(o_{i,1},o_{i,2},\dots,o_{i,t})$. 由于可加性假设不能适用于所有场景,QMIX算法[131]进一步提出单调性非线性假设,即联合$Q$值是每个智能体的$q^{(i)}$值的单调非线性组合,并借助神经网络近似任意的单调非线性关系. 考虑到可加性和单调性假设是IGM原则的充分非必要条件,相对比较严格且在一定程度上限制了对复杂非线性关系的表达,人们提出了QTRAN并对上述条件进行放松. 具体地,QTRAN[132]将原有的联合$Q$函数等价于一个易于分解的函数$Q'$来满足IGM原则. Qatten算法[133]利用注意力机制度量个体对全局的重要性,利用多头注意力机制近似分解联合价值函数. Qplex[134]将IGM原则转化为对优势函数易于实现的约束,然后对集中优势函数进行注意力分解.

基于PG的MARL算法主要包括MADDPG、COMA和MAAC等. 其中:COMA仅适用于合作型任务,MADDPG和MAAC同时适用于完全合作型、完全竞争型和混合型任务. MADDPG算法[135]将DDPG算法扩展到多智能体领域,其中每个智能体通过一个集中的Critic收集其他智能体的信息和策略,并对其他智能体策略进行建模,能够有效降低多智能体环境的非平稳性. COMA[136]利用反事实基线解决多智能体信度分配问题. MAAC[137]在计算集中式Critic时,利用共享的注意机制提高算法的可扩展性. ALC框架[138]通过分析个体间的相关性,加强相关性强的智能体之间的策略协同,构建MARL策略协调训练框架. FOP[139]和VDAC[140]分别提出了熵最大RL和AC基础上的值分解算法. ROMA[141]和RODE[142]提出了面向角色的MARL框架,角色相似的智能体通过分享学习实现策略的相似性.

表5对现有主流的集中式训练-分散式执行类算法进行总结和对比.

**表5 集中式训练-分散式执行类算法的总结和对比**

| 算法名称 | 算法类型 | 适用任务 | 算法结果 |
|---|---|---|---|
| VDN[130] | 无模型,值函数,TD类 | 合作型 | 对联合价值函数进行线性加和分解 |
| QMIX[131] | 无模型,值函数,TD类 | 合作型 | 用神经网络拟合联合价值函数与个体价值函数之间的单调非线性关系 |
| QTRAN[132] | 无模型,值函数,TD类 | 合作型 | 对QMIX的单调假设进行放松,将联合$Q$函数转化为易于分解的函数 |
| Qatten[133] | 无模型,值函数,TD类 | 合作型 | 利用注意力机制度量个体对全局的重要性,然后分解联合价值函数 |
| MADDPG[135] | 无模型,AC类,TD类,离轨策略 | 任意类型 | DDPG的多智能体扩展,每个智能体对其他智能体的策略进行建模 |
| MAAC[137] | 无模型,AC类,TD类,同轨策略 | 合作型 | 利用反事实基线来解决多智能体信度分配问题 |

### 2.2.4 集中式训练分布式执行

分布式通信是介于集中式与分散式之间的通信模式. 在分布式通信模式中,每个智能体通过局部通信与自己的邻居节点进行交互,有选择地传递与对环境的观测值、策略等有关的信息,并基于自己和邻居的信息进行决策.

集中式训练分布式执行,是在CTDE类算法的分散式执行上进行了通信增强. 一方面,集中式学习假设每个智能体都能获取其余智能体以及环境的全部信息,在产生高昂通信成本的同时也造成了干扰和冗余通信;另一方面,当没有智能体可以访问环境的完整状态时,智能体间的通信和信息交互有助于更好地进行决策. 分布式执行分为预定义通信和自主通信两种通信模式. 预定义通信模式在预定义的通信拓扑上进行通信,限制了智能体之间的通信方式,从而限制了潜在的合作. 在自主通信模式中,智能体需要学习何时与哪些智能体进行通信.

基于通信的多智能体算法主要处理以下问题:1) 判定何时通信;2) 选择通信对象;3) 对接收的信息进行融合;4) 学习信息编码方式和策略,以实现算法可扩展性提升、通信成本降低、重要信息提取等.

集中式训练-分布式执行算法可以为不同的CTDE类算法设计通信拓扑和通信方案,解决问题的类型取决于原有的CTDE算法. 此外,通信增强通常

发生在合作型任务中的智能体之间,因此这类算法基本也都是针对合作型任务的. TarMAC[143] 改进了CommNet,通过基于签名的注意机制差异化地处理接收到的信息,每个智能体利用注意力机制对收到不同智能体的信息进行加权平均,并利用门控循环神经网络(gated recurrent neural networks on sequence modeling, GRU)[144] 更新隐层状态信息,进一步学习局部策略.

尽管通信是必要的,但冗余的通信信息会对决策造成干扰,因此允许每个智能体决定是否向其他智能体发送消息是一种明智的策略. 在BiCNet、ATOC、IC3Net等算法中,智能体通过选择是否在一定范围内广播个体的信息来减少冗余通信. BiCNet[145] 针对集群对抗任务提出了一种多智能体双向协调网络,只考虑部分其他智能体发送的信息,并利用双向循环神经网络整合来自其他智能体的消息. ATOC[146] 提出了注意交流模型来学习何时需要通信以及如何整合共享信息,该算法根据通信和非通信的平均$Q$值差异判定何时与邻居智能体进行通信,然后利用双向长短期记忆模型(long short-term memory, LSTM)[147] 对收到的信息进行整合. IC3Net[148] 使用门控机制实现个性化的受控连续通信模型,适用于合作、竞争和混合型任务. IC3Net通过引入基于神经网络的门控机制指导智能体学习何时进行广播通信,选择通信的智能体将在该时段广播其信息,然后每个智能体利用LSTM根据局部观测和接收的信息预测隐层状态信息,进一步学习局部策略. 设$t$时刻智能体$i$的隐层信息为$h_i^t$,具体的通信和信息整合过程如下:

$$h_i^{t+1}, s_i^{t+1} = F(e(o_i^t) + c_i^t, h_i^t, s_i^t), \tag{23}$$

$$g_i^{t+1} = f^g(h_i^t), \tag{24}$$

$$c_i^{k+1} = \frac{1}{N-1}\sum_{j\neq i} h_j^{k+1}\cdot g_i^{t+1}. \tag{25}$$

其中: $F(\cdot)$为LSTM模型, $h_i^t$和$s_i^{t+1}$为LSTM的隐层信息和长期记忆信息, $e(\cdot)$为全连接神经网络构成的参数化编码函数, $f^g(\cdot)$为单层神经网络函数, $g_i^t$为智能体$i$决定是否进行广播通信的门控机制,智能体$i$根据隐层信息$h_i^t$进行决策. 针对广播通信造成的较大通信开销和冗余信息干扰, I2C[149] 进一步提出了独立推断通信和请求应答通信机制,利用因果推断和前馈网络在通信范围内依概率选择通信对象并进行信息交互.

表6对现有主流的集中式训练-分布式执行类算法进行总结和对比.

**表 6　集中式训练-分布式执行类算法的总结和对比**

| 算法名称 | 算法类型 | 适用任务 | 算法结果 |
|---|---|---|---|
| TarMAC[143] | 无模型, AC类, TD类, 通用通信协议 | 合作型 | 选择部分通信对象并基于签名的注意机制对收到的信息加权 |
| BiCNet[145] | 无模型, AC类, TD类, 通用通信协议 | 团体对抗型 | 针对集群对抗任务提出了多智能体双向协调网络 |
| ATOC[146] | 无模型, AC类, TD类, 通用通信协议 | 合作型 | 提出注意交流模型来学习何时需要通信,以及如何整合共享信息 |
| IC3Net[148] | 无模型, AC类, TD类, 通用通信协议 | 任意类型 | 使用门控机制来实现个性化受控连续通,广播部分个体的信息 |
| I2C[149] | 无模型, AC类, TD类, 通用通信协议 | 合作型 | 提出独立推断通信,选择通信部分通信对象进行信息交互 |

### 2.2.5　分布式训练分布式/分散式执行

分布式学习是介于集中式学习与分散式学习之间的学习方式,在基于通信的分布式学习中,每个智能体可以进行局部通信,包括分布式训练分布式执行、分散式执行两类. 智能体可以通过与自己的邻居进行通信,有选择地传递对环境的观测值、动作、局部策略等信息. 在分布式算法中,通常假设对每个智能体状态$s_t$是全局可观测的,或可以通过分布式方法获得的.

现有的完全分布式算法大多基于一致性方法. Yan等[150] 提出了一种分布式离轨AC算法解决同质智能体RL问题,通过引入一致性步骤保证所有智能体获得相同的最优策略. Zhang等[151] 提出了一种时变无向通信拓扑上的完全分布式的随机策略AC算法,通过引入一致性步骤实现价值函数估计的一致性,并证明了所提出算法的收敛性. Lin等[152-153] 在此基础上进一步研究了时变和切换有向通信拓扑上的分布式AC算法. 平均场从局部角度描述了一种分布式通信结构,平均场$Q$学习算法和平均场AC算法[154-156] 用均值替代周围智能体的影响. 智能体$i$在估计值函数时只考虑邻居节点的动作,有

$$Q_i(s,a) \approx \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} Q_i(s,a_i,a_j), \tag{26}$$

其中$\mathcal{N}_i$为智能体$i$的邻居节点的集合. 用$\bar{a}_i$代表智能体$i$对周围节点的动作均值的估计,有

$$Q_i(s,a) \approx Q_i(s,a_i,\bar{a}_i) \text{ 且 } \bar{a}_i = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} a_j. \tag{27}$$

于是更新规则如下:

$$Q_i(s_t,a_i,\bar{a}_i) \leftarrow$$
$$(1-\alpha)Q_i(s_t,a_i,\bar{a}_i) + \alpha[r_i + \gamma V_i(s_{t+1})]. \tag{28}$$

为了解决有限通信和部分可观测性问题,

FMA2C[157]提出了一种结合了联邦学习思想的分层MARL算法,在交通信号控制中取得了较好的效果. NDQ[158]提出通过通信最小化学习近似分解$Q$函数的新框架,智能体之间通过低频的通信实现有效的协调. 为了提高多智能体决策的协调性,Liang等[159]提出了一种称为异步可扩展多智能体近端策略优化(asynchronous and scalable multi-agent PPO, ASM-PPO)的算法,该算法在部分可观测MDP中进行异步学习和决策. GCS等[160]提出了基于图形协调策略的MARL算法,将联合团队策略分解为图形生成器和基于图形的协调策略,以实现智能体之间的协调行为. Cohen等[161]提出了基于层次图概率递归推理的多智能体协调优化算法,利用概率递归推理(graph probabilistic recursive reasoning, GrPR2)在局部交互域中进行对手建模,每个智能体都以迭代方式对其他智能体的策略做出最佳响应.

很多现实问题常常具有复杂的耦合约束,例如智能电网中多个发电机总发电量上的供需平衡约束,然而绝大部分MARL的算法设计都没有考虑多智能体的联合动作上带耦合约束的情况. 设计分布式RL算法解决带耦合约束的问题是困难的,智能体需要分布式地协调各自的动作以满足耦合约束条件. Wen等[162]提出了分布式训练分散式执行(distributed training with decentralized execution, DTDE)的MARL框架,该框架是一个完全分布式的学习框架,并且考虑了耦合约束条件的处理. 在该框架中,无论是训练阶段还是执行阶段,智能体都只与自己的邻居节点通信,并根据局部信息进行学习和决策. 此外,还有一些智能电网背景下的分布式RL算法[163-166]也给出了带耦合约束条件问题的分布式解决方案. Dai等[166]提出了一种新的分布式RL优化算法解决智能电网的动态经济调度问题,Hu等[167]在此基础上进一步利用分布式投影算法和动作网络提高算法处理耦合约束的效率和对波动电力需求的鲁棒性.

表7对现有主流的分布式训练-分布式/分散式执行类算法进行了总结和对比.

**表7 分布式训练-分布式/分散式执行类算法的总结和对比**

| 算法名称 | 算法类型 | 适用任务 | 算法结果 |
|---|---|---|---|
| 分布式离轨AC算法[150] | 无模型, AC类, TD类, 离轨策略 | 合作型 | 利用一致性方法分布式地获得相同的最优策略 |
| 无向拓扑上的AC算[151] | 无模型, AC类, TD类, 同轨策略 | 合作型 | 提出无向通信拓扑上的完全分布式AC算法 |
| 无向拓扑上的AC算法[152-153] | 无模型, AC类, TD类, 同轨策略 | 合作型 | 提出切换有向通信拓扑上的完全分布式AC算法 |
| 平均场$Q$学习/AC算法[154-156] | 无模型, 表格型/AC类, TD类 | 合作型 | 用均值刻画周围智能体的影响 |
| 分布式RL优化算法[166-167] | 无模型, 值函数, TD类, 离轨策略 | 合作型 | 针对智能电网经济调度问题提出完全分布式算法 |

近年来,MARL已广泛应用于无人系统协同控制、智能电网优化调度、智能通信等领域. 针对充电站能量调度问题,Zhang等[25]结合MADDPG和LSTM提出了一种MARL方法以学习最优能源购买策略,并提出了一种在线启发式调度方案制定能源分配策略,该方法在经济利润和用户满意度方面均优于充电市场中的其他能量调度方法. 针对数据包路由问题,Ding等[26]提出了改进的分布式DQN算法评估邻居路由器,其中每个路由器智能地自行选择下一跳路由器,降低了计算复杂度和网络拥塞概率. Dai等[166-167]将MARL算法和分布式优化算法相结合解决了智能电网中的动态经济调度问题. Na等[168]将噪声添加到人工神经网络的权重和偏差中,提出了一种基于DDPG的新算法解决自动驾驶中的避碰任务. Chen等[169]将深度MARL与注意力机制相结合来学习和预测周围人群的交互和运动,实现了机器人集群在人群中导航和高效运动的目标. 针对机器人的运动规划,Everett等[170]开发了一种算法学习多个动态机器人之间的避碰,结合LSTM算法适应多个智能体数目的变化. Zhao等[171]研究了包含未知非线性动力学和扰动的协作欠驱动四旋翼的无模型鲁棒编队控制问题,基于分层控制方案和强化学习理论提出了一种四旋翼动力学未知的鲁棒控制器. 针对复杂海洋环境下的无人艇轨迹跟踪问题,Wang等[172]提出了一种仅基于无人艇输入输出信号的自学习无模型解决方案,在规定的跟踪精度下实现了无人艇的最优控制.

DP是解决多阶段决策过程中最优化问题的传统方法,其基本思想也是将待求解问题分解成若干个子问题,先求解子问题,再用这些子问题的解得到原问题的解. RL方法发展了传统的DP方法,将环境建模为概率转移函数,解决DP方法在状态和动作空间较大时遭受的维数灾难问题,放松DP方法需要完全环境模型的缺陷. 控制论研究人员引入RL这两方面的思想形成ADP方法,用于解决最优控制问题. 为了全面展现RL理论与算法,探索RL在系统控制领域的最新理论成果,进一步探讨RL和ADP的区别与联系,下面将介绍ADP的理论发展和应用.

## 3　自适应动态规划

本章从3个部分介绍ADP理论的发展与应用: ADP结构发展、ADP算法发展和ADP在MAS最优协调控制中的应用.

### 3.1　ADP结构发展

ADP方法在具有未知动力学的系统最优控制研究中得到了广泛关注. ADP依结构划分包括3种基本类型[35,173]: 启发式动态规划(heuristic dynamic programming, HDP)、二次启发式规划(dual heuristic programming, DHP)和全局二次启发式规划(globalized DHP, GDHP). 这3种类型都包含3层神经网络, 即评价网络(critic network)、模型网络(model network)和执行网络(action network), 且评价网络与执行网络通过模型网络间接连接. 如果执行网络与评价网络直接连接, 则这样的算法称为上述算法的动作依赖(action-dependent)形式, 分别为控制依赖启发式动态规划(action dependent HDP, ADHDP)[35]、控制依赖二次启发式规划(action dependent DHP, ADDHP)[35]和控制依赖全局二次启发式规划(action dependent globalized DHP, ADGDHP)[173]. 进一步, 上述算法中去掉执行网络便形成了单网络自适应评价(single network adaptive critic, SNAC)方法[174]. 对于以上几种ADP结构, 更详细的介绍和区别可参见文献[84,175-177].

### 3.2　ADP算法发展

由基于模型的离线规划到无模型的在线学习的过程, 就是由DP方法发展到ADP方法的过程. 为了方便内容的展开介绍, 给出以下定义.

**定义1**　在某一时间区间上的所有控制输入的值, 称作控制历史(简称为控制), 统一记作$u(\cdot)$.

**定义2**　在某一时间区间上的所有状态的值, 称作状态轨迹(简称为状态), 统一记作$x(\cdot)$.

#### 3.2.1　基于模型的算法

下面分别就离散系统和连续系统两种情况介绍基于模型的离线迭代DP方法. Zhao等[178]考虑了如下离散系统:

$$x(k+1)=f(x(k))+g(x(k))u(k). \tag{29}$$

其中: $k$为离散时间指标, $x(k)\in \boldsymbol{R}^n$为系统状态, $u(k)\in \boldsymbol{R}^m$为系统输入, $f(x(k))$和$g(x(k))$为具有适当维数的(非)线性函数. 系统的性能指标定义如下:

$$J(x(0),u(\cdot))=\sum_{i=0}^{+\infty}\gamma^i r(x(i),u(i)). \tag{30}$$

其中: $\gamma\in(0,1]$为折扣因子, $r(x(i),u(i))$为成本函数. 对于任意固定策略$u(\cdot)$, 给出相应的值函数

$$V(x(k))=\sum_{i=k}^{+\infty}\gamma^{i-k} r(x(i),u(i)). \tag{31}$$

**定义3**　如果策略$u(\cdot)=u(x(\cdot))$能够使得系统(29)在一个紧集$\Omega$上稳定, $u(0)=0$, 且使得$V(x(\cdot))$有限, 则此控制策略称为容许控制策略.

离散系统最优控制问题的目标是设计最小化性能函数(30)的容许反馈控制策略$u(\cdot)$. 将式(31)改写为如下递推的形式:

$$V(x(k))=r(x(k),u(k))+\gamma V(x(k+1)). \tag{32}$$

式(32)称作贝尔曼方程. 最优值函数定义如下:

$$V^*(x(k))=\min_{u(\cdot)}\{r(x(k),u(k))+\gamma V(x(k+1))\}. \tag{33}$$

根据贝尔曼最优性原理, 得到如下离散时间(discrete-time, DT) HJB方程:

$$V^*(x(k))=\min_{u(k)}\{r(x(k),u(k))+\gamma V^*(x(k+1))\}.$$

因此最优策略表示为

$$u^*(k)=\arg\min_{u(k)}\{r(x(k),u(k))+\gamma V^*(x(k+1))\}. \tag{34}$$

考虑如下成本函数:

$$r(x(k),u(k))=Q(x(k))+u^{\mathrm{T}}(k)Ru(k), \tag{35}$$

其中$R$为正定矩阵, 并且对于任意非零向量$x(k)$, $Q(x(k))$是正定矩阵. 特别地, 可选择$Q(x(k))=x^{\mathrm{T}}(k)Qx(k)$, 其中$Q$为正定矩阵. 最优策略(34)变为

$$u^*(k)=-\frac{\gamma}{2}R^{-1}g^{\mathrm{T}}(x(k))\nabla V^*(x(k+1)), \tag{36}$$

其中$\nabla V(x)=\dfrac{\partial V(x)}{\partial x}$.

下面给出迭代求解此最优控制问题的方法[1]: PI算法(算法1)、VI算法(算法2)、GPI算法(算法3).

**算法1**　PI算法.

输入: 容许的初始控制策略$u_0(k)$, 迭代次数$\iota=0,1,\ldots$;

输出: $V^*(x(k))$, $u^*(k)$.

策略评估: 在每一步$\iota$, 求解如下贝尔曼方程得到当前策略$u_\iota(k)$下的值函数:

$$V_\iota(x(k))=r(x(k),u_\iota(k))+\gamma V_\iota(x(k+1)). \tag{37}$$

策略改进: 由下式改进策略:

$$u_{\iota+1}(k)=\arg\min_{u(k)}\{r(x(k),u(k))+\gamma V_\iota(x(k+1))\}. \tag{38}$$

若成本函数选取为式(35), 则依下式进行策略更新:

$$u_{\iota+1}(k)=-\frac{\gamma}{2}R^{-1}g^{\mathrm{T}}(x(k))\nabla V_\iota(x(k+1)).$$

**算法2**　VI算法.

输入:任意初始控制策略$u_0(k)$、$V_0(x(k))$,迭代次数$\iota=0,1,\ldots$;

输出:$V^*(x(k))$, $u^*(k)$.

值更新:在每一步$\iota$,根据下式更新值函数:

$$V_{\iota+1}(x(k))=r(x(k),u_\iota(k))+\gamma V_\iota(x(k+1)). \quad (39)$$

策略改进:由下式改进策略:

$$u_{\iota+1}(k)=\arg\min_{u(k)}\{r(x(k),u(k))+\gamma V_{\iota+1}(x(k+1))\}. \quad (40)$$

若成本函数选取为式(35),则依下式进行策略更新:

$$u_{\iota+1}(k)=-\frac{\gamma}{2}R^{-1}g^{\mathrm{T}}(x(k))\nabla V_\iota(x(k+1)).$$

**算法3** GPI算法.

输入:任意初始控制策略$u_0(k)$、$V_0(x(k))$,迭代次数$\iota=0,1,\ldots$,常数$T_\iota$;

输出:$V^*(x(k))$, $u^*(k)$.

值更新:在每一步$\iota$,由下式更新值函数:

$$V_\iota^{\kappa+1}(x(k))=r(x(k),u_\iota(k))+\gamma V_\iota^{\kappa}(x(k+1)),\quad \kappa=1,2,\ldots,T_\iota. \quad (41)$$

其中

$$V_\iota(x(k))=V_\iota^0(x(k)),\ V_{\iota+1}(x(k))=V_\iota^{T_\iota+1}(x(k)).$$

策略改进:由下式改进策略:

$$u_{\iota+1}(k)=\arg\min_{u(k)}\{r(x(k),u(k))+\gamma V_{\iota+1}(x(k+1))\}. \quad (42)$$

若成本函数选取为式(35),则依下式进行策略更新:

$$u_{\iota+1}(k)=-\frac{\gamma}{2}R^{-1}g^{\mathrm{T}}(x(k))\nabla V_\iota(x(k+1)).$$

Leake等[179]证明了初始策略容许的情况下,PI算法收敛到最优值(33)和最优策略(34)(或(36)). 但是观察式(37)可以发现,需要先知道$k+1$时刻值函数的值$V_\iota(x(k+1))$用于确定$k$时刻值函数的值$V_\iota(x(k))$,此过程是时间后推的,计算量较大. 基于贝尔曼方程是固定点方程这一事实,给出式(37)的迭代求解方法为

$$V_\iota^{\kappa+1}(x(k))=r(x(k),u_\iota(k))+\gamma V_\iota^{\kappa}(x(k+1)). \quad (43)$$

其中:$\kappa=0,1,\ldots,T$, $V_\iota^0(x(k))=V_{\iota-1}(x(k))$. 根据上述递推式,当$T=+\infty$,即$\kappa\to+\infty$时,$V_\iota^\kappa(x(k))\to V_\iota(x(k))$. 这种迭代求解贝尔曼方程的方法称作策略迭代(iterative PI)方法[5]. 当$T$取任意有限值时,可得到GPI算法. 特别地,当$T=1$时产生VI算法. 相比于PI算法,VI算法的收敛性不需要容许的初始策略[175].

至此,给出一般离散非线性系统最优控制问题基于模型的控制器设计方法. 特别地,Shaiju等[180]考虑了如下一般线性离散系统:

$$x(k+1)=Ax(k)+Bu(k), \quad (44)$$

其中$A$、$B$为适当维数的系统矩阵,且满足$(A,B)$可镇定. 对于给定的策略,得出如下值函数:

$$V(x(k))=\sum_{i=k}^{+\infty}x^{\mathrm{T}}(i)Qx(i)+u^{\mathrm{T}}(i)Ru(i). \quad (45)$$

离散线性系统最优控制问题的控制目标是,设计容许控制器最小化性能函数(45),称为线性二次调节(linear quadratic regulator, LQR)问题.

给定一个控制器,假设值函数的形式为$V(k)=x^{\mathrm{T}}(k)Px(k)$. 类似于式(32),可以写出如下贝尔曼方程:

$$x^{\mathrm{T}}(k)Px(k)=x^{\mathrm{T}}(k)Qx(k)+u^{\mathrm{T}}(k)Ru(k)+x^{\mathrm{T}}(k+1)Px(k+1). \quad (46)$$

定义Hamiltonian函数

$$H(x(k),u(k),\Delta V(k))=x^{\mathrm{T}}(k)Qx(k)+u^{\mathrm{T}}(k)Ru(k)-x^{\mathrm{T}}(k)Px(k)+x^{\mathrm{T}}(k+1)Px(k+1).$$

由平稳性条件$\dfrac{\partial H(x(k),u(k),\Delta V(k))}{\partial u(k)}=0$和最优值函数$V^*(k)=x^{\mathrm{T}}(k)P^*x(k)$可以得到如下最优控制策略和HJB方程(即代数黎卡提方程(algebraic Riccati equation, ARE)):

$$u^*(k)=-K^*x(k)=-(R+B^{\mathrm{T}}P^*B)^{-1}B^{\mathrm{T}}P^*Ax(k), \quad (47)$$

$$A^{\mathrm{T}}P^*A-P^*+Q-A^{\mathrm{T}}P^*B(R+B^{\mathrm{T}}P^*B)^{-1}B^{\mathrm{T}}P^*A=0. \quad (48)$$

Lewis等[181-182]总结了线性离散系统LQR问题的PI算法(算法4)、VI算法(算法5)、GPI算法(算法6), Hewer等[183-184]分析了这些算法的收敛性.

**算法4** PI算法解决DT LQR问题.

输入:任意容许的初始控制增益$K_0$,迭代次数$\iota=0,1,\ldots$;

输出:$K^*$, $P^*$.

策略评估:在每一步$\iota$,由下式评估值函数:

$$P_\iota=(A-BK_\iota)^{\mathrm{T}}P_\iota(A-BK_\iota)+Q+K_\iota^{\mathrm{T}}RK_\iota. \quad (49)$$

策略更新:利用下式确定更新策略:

$$K_{\iota+1}=(R+B^{\mathrm{T}}P_\iota B)^{-1}B^{\mathrm{T}}P_\iota A. \quad (50)$$

**算法5** VI算法解决DT LQR问题.

输入: 任意初始控制增益$K_0$、$P_0$, 迭代次数$\iota = 0, 1, \ldots$, 常数$T_\iota$;

输出: $K^*$, $P^*$.

值更新: 在每一步$\iota$, 由下式更新值函数:

$$P_{\iota+1} = (A - BK_\iota)^{\mathrm{T}} P_\iota (A - BK_\iota) + Q + K_\iota^{\mathrm{T}} R K_\iota. \tag{51}$$

策略更新: 利用下式确定更新策略:

$$K_{\iota+1} = (R + B^{\mathrm{T}} P_{\iota+1} B)^{-1} B^{\mathrm{T}} P_{\iota+1} A. \tag{52}$$

**算法6**　GPI算法解决DT LQR问题.

输入: 任意初始控制增益$K_0$, 迭代次数$\iota = 0, 1, \ldots$, 常数$T_\iota$;

输出: $K^*$, $P^*$.

值更新: 在每一步$\iota$, 由下式更新值函数:

$$P_\iota^{\kappa+1} = (A - BK_\iota)^{\mathrm{T}} P_\iota^{\kappa} (A - BK_\iota) + Q + K_\iota^{\mathrm{T}} R K_\iota. \tag{53}$$

其中: $\kappa = 1, 2, \ldots, T_\iota$, $P_j^0 = P_j$, $P_{j+1} = P_j^K$.

策略更新: 利用下式确定更新策略:

$$K_{\iota+1} = (R + B^{\mathrm{T}} P_{\iota+1} B)^{-1} B^{\mathrm{T}} P_{\iota+1} A. \tag{54}$$

需要特别说明的是, 对于非线性系统, 算法1 ~ 算法3, 仅能解决有限状态空间和有限动作空间的问题. 对于最优控制问题而言, 这显然是不合适的. 后面章节将介绍利用函数逼近的方法执行这3种算法.

对于连续控制系统情况, Vamvoudakis 等[185] 考虑了如下非线性系统的最优控制问题:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t),\ t \geqslant t_0. \tag{55}$$

其中: $x(t) \in \boldsymbol{R}^n$ 为系统状态, $u(t) \in \boldsymbol{R}^m$ 为系统输入, $f(x(k))$ 和 $g(x(k))$ 为具有适当维数的(非)线性函数. 考虑如下性能函数:

$$J(x(t_0), u(\cdot)) = \int_{t_0}^{+\infty} r(x(t), u(t)) \mathrm{d}t. \tag{56}$$

相比于离散系统, 此处考虑的是非折扣型性能函数. 对于任一给定策略, 定义如下值函数:

$$V(x(t)) = \int_t^{+\infty} r(x(\tau), u(\tau)) \mathrm{d}\tau. \tag{57}$$

贝尔曼方程为

$$0 = r(x(t), u(t)) + \nabla V^{\mathrm{T}}(x(t))(f(x(t)) + g(x(t))u(t)). \tag{58}$$

由贝尔曼最优性原理, 得到如下HJB方程:

$$0 = \nabla V^{*\mathrm{T}}(x(t))(f(x(t)) + g(x(t))u^*(t)) + r(x(t), u^*(t)). \tag{59}$$

观察算法1 ~ 算法3, 如果开发连续系统的PI算法、VI算法、GPI算法, 则根据式(58)可知, 策略评估步或者值更新步需要$f(x(t))$、$g(x(t))$的信息, 这不利于推广DP离线算法到ADP在线算法. Vamvoudakis 等[185-186] 针对连续系统最优控制问题提出了积分强化学习(integral RL, IRL)算法.

类比于离散系统, 将贝尔曼方程(58)改为如下递推形式:

$$V(x(t)) = \int_t^{t+\Delta t} r(x(\tau), u(\tau)) \mathrm{d}\tau + V(x(t + \Delta t)). \tag{60}$$

式(60)与(58)等价[10], 因此得到如下最优控制策略:

$$u^*(x(t)) = \arg \min_{u(t:t+\Delta t)} \Big\{ \int_t^{t+\Delta t} r(x(\tau), u(\tau)) \mathrm{d}\tau + V^*(x(t + \Delta t)) \Big\}. \tag{61}$$

考虑成本函数$r(x(t), u(t)) = Q(x(t)) + u^{\mathrm{T}}(t) R u(t)$, 其中$R$为正定矩阵, 且对于任意非零向量$x(t)$有$Q(x(t))$是正矩阵. 最优策略(61)变为

$$u^*(t) = -\frac{1}{2} R^{-1} g^{\mathrm{T}}(x(t)) \nabla V^*(x(t)),$$

类比于算法1 ~ 算法3, 可得到连续系统PI算法、VI算法、GPI算法[1,187].

Jiang 等[34] 考虑了线性连续系统

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{62}$$

的LQR问题. 定义性能函数

$$J(x(t_0), u(\cdot)) = \int_{t_0}^{+\infty} (x^{\mathrm{T}}(t) Q x(t) + u^{\mathrm{T}}(t) R u(t)) \mathrm{d}t,$$

根据最优控制理论可得到最优控制策略

$$u_i^*(t) = K^* x(t) = R^{-1} B^{\mathrm{T}} P^* x(t),$$

其中$P^*$为ARE $(A - BK^*)^{\mathrm{T}} P^* + P^*(A - BK^*) + Q + K^{*\mathrm{T}} R K^* = 0$的解. 类似于离散系统LQR问题, 可以分别给出线性连续系统LQR问题的PI算法、VI算法、GPI算法, 并证明算法的收敛性[188-189].

### 3.2.2　无模型的算法

对于有限状态空间和动作空间的最优控制问题, 算法1 ~ 算法3可适用, 但是对于状态空间或动作空间较大或者连续的情况, 上述3种算法均难以实施. 为了解决该问题, 提出值函数近似的方法. 以离散系统、成本函数为(35)情形为例, 对ADP方法原理进行说明. (非)线性连续系统可以类似给出[34,190].

首先考虑ADP算法用于非线性离散系统最优控制问题的研究进展. 基于Weierstrass高阶逼近定理, 值函数可以表示为$V(x(k)) = W^{\mathrm{T}} \phi(x(k)) + \epsilon(x(k))$. 其中: $W \in \boldsymbol{R}^l$ 为未知的理想权重, $\phi(x(k)) \in \boldsymbol{R}^l$、$\epsilon(x(k)) \in \boldsymbol{R}$分别为激活函数和逼近误差. 进一步, 由于$W$是未知的, $V(x(k))$可以近似表示为

$$\hat{V}(x(k)) = \hat{W}_1^{\mathrm{T}}(k) \phi(x(k)), \tag{63}$$

其中$\hat{W}_1(k)$为$W$的估计值. 同样地,控制输入近似表示为

$$\hat{u}(k)=-\frac{1}{2}R^{-1}g^{\mathrm{T}}(x(k))\hat{W}_2^{\mathrm{T}}(k)\nabla\phi(x(k)),\quad(64)$$

其中$\hat{W}_2(k)$为$W$的估计值. 控制输入也可以近似表示为

$$\hat{u}(k)=\hat{W}_2^{\mathrm{T}}(k)\phi_a(x(k)),\quad(65)$$

其中$\phi_a(x(k))$为激活函数. 注意到, 式(65)中的$\hat{W}_2(k)$与(64)是不同的,但为了下述参数自适应律设计的统一,采取相同的符号.

将式(63)和(64)(或者(65))代入(32),得到误差

$$e_1(k)=Q(x(k))+\hat{u}^{\mathrm{T}}(k)R\hat{u}(k)+\gamma\hat{V}(x(k+1))-\hat{V}(x(k)).$$

最小化损失函数$E_1(k)=\frac{1}{2}e_1^{\mathrm{T}}(k)e_1(k)$以设计$\hat{W}_1(k)$的自适应律

$$\hat{W}_1(k+1)=\hat{W}_1(k)-\frac{\partial E_1(k)}{\partial\hat{W}_1(k)}.$$

类似地,最小化损失函数$E_2(k)=\frac{1}{2}e_2^{\mathrm{T}}(k)e_2(k)$,其中

$$e_2(k)=\frac{1}{2}R^{-1}g^{\mathrm{T}}(x(k))\hat{W}_1^{\mathrm{T}}(k)\nabla\phi(x(k))+\hat{u}(x(k)),$$

以设计$\hat{W}_2(k)$的自适应律

$$\hat{W}_2(k+1)=\hat{W}_2(k)-\frac{\partial E_2(k)}{\partial\hat{W}_2(k)}.$$

可以看到,为了设计$\hat{W}_2(k)$的自适应律,$g(x(k))$的信息是需要的.

算法7提供了基于PI的在线RL算法,其中下标$\iota$表示第$\iota$次迭代对应的参数. 类似地,可以给出基于VI的在线强化学习算法.

**算法7** 在线PI算法.

输入: 容许的初始控制策略$u_0(k)$,迭代次数$\iota=0,1,\ldots$;

输出: $V^*(x(k))$, $u^*(k)$.

策略评估: 由下式更新值函数的权值:

$$\hat{W}_{1\iota}(k+1)=\hat{W}_{1\iota}(k)-\frac{\partial E_{1\iota}(k)}{\partial\hat{W}_{1\iota}(k)}.\quad(66)$$

也可以按下式更新参数: 确定方程$\hat{W}_{1\iota}^{\mathrm{T}}\phi(x(k))=r(x(k),u_\iota(x(k)))$的最小二乘解$\hat{W}_{1\iota}$,其中$\phi(k)=\phi(x_k)-\gamma\phi(x_{k+1})$.

策略改进: 由下式改进策略的权值:

$$\hat{W}_{2\iota}(k+1)=\hat{W}_{2\iota}(k)-\frac{\partial E_{2\iota}(k)}{\partial\hat{W}_{2\iota}(k)}.\quad(67)$$

为了得到参数$\hat{W}_1(k)$、$\hat{W}_2(k)$的收敛性,需要$\phi(x(k))$满足持续激励条件(PE条件),这便需要在输入中注入噪声,Kiumarsi等[191]已证明得到的最优解是有偏的,该方法称作同策略(on-policy)算法. 随后,Kiumarsi等[191]和Jiang等[34]分别对离散系统和连续系统提出了异策略(off-policy)算法.

进一步地,为了放松算法对$g(x(k))$的需求,引入模型网络$\hat{x}(k+1)=\hat{W}_3^{\mathrm{T}}(k)\phi_1(x(k),u(k))$. 其中: $\phi_1(x(k),u(k))$为激活函数,$u(k)$为行为策略. 最小化损失函数$E_3(k)=\frac{1}{2}e_3^{\mathrm{T}}(k)e_3(k)$以设计$\hat{W}_3(k)$的自适应律,其中$e_3(k)=\hat{x}(k+1)-x(k+1)$. 因此,$g(x(k))$可以被$\partial\hat{x}(k+1)/\partial u(k)$替代. 至此,给出解决非线性离散系统最优控制问题的无模型方法,其中$f(x(k))$和$g(x(k))$的信息均不需要.

文献[191]详细介绍了同策略算法和异策略算法用于处理线性离散系统最优控制问题的情形. 考虑离散线性系统(44)及性能函数(45),首先介绍同策略算法.

**算法8** 同策略在线PI算法.

输入: 容许的初始控制策略$u_0(k)$,迭代次数$\iota=0,1,\ldots$;

输出: $V^*(x(k))$, $u^*(k)$.

策略评估: 在每一步$\iota$,通过求解如下贝尔曼方程获得$P_\iota$:

$$x(k)^{\mathrm{T}}P_\iota x(k)=x(k)^{\mathrm{T}}Qx(k)+u_\iota^{\mathrm{T}}(k)Ru_\iota(k)+x^{\mathrm{T}}(k+1)P_\iota x(k+1).\quad(68)$$

策略更新: 通过下式更新策略:

$$u_{\iota+1}(k)=-K_{\iota+1}x(k)=-(R+B^{\mathrm{T}}P_\iota B)^{-1}B^{\mathrm{T}}P_\iota Ax(k).\quad(69)$$

根据以上算法可以发现[191],在策略评估步为了获得$P_\iota$,需要在控制输入中注入噪声,这导致根据同策略算法得到的最优解是有偏的. 因此考虑异策略算法,此时系统改写为

$$x(k+1)=A_kx(k)+B(K_\iota x(k)+u(k)),\quad(70)$$

其中$A_k=A-BK_\iota$. 经过一些变换,贝尔曼方程可以改写为

$$\begin{aligned}&x^{\mathrm{T}}(k)P_\iota x(k)-x^{\mathrm{T}}(k+1)P_\iota x(k+1)=\\&x^{\mathrm{T}}(k)Qx(k)+x^{\mathrm{T}}(k)K_\iota^{\mathrm{T}}RK_\iota x(k)-\\&(u(k)+K_\iota x(k))^{\mathrm{T}}B^{\mathrm{T}}P_\iota x(k+1)-\\&(u(k)+K_\iota x(k))^{\mathrm{T}}B^{\mathrm{T}}P_\iota A_kx(k).\end{aligned}\quad(71)$$

**算法9** DT LQR异策略在线PI算法.

输入: 容许的初始控制策略$u_0(k)$,迭代次数$\iota=0,1,\ldots$;

输出: $V^*(x(k))$, $u^*(k)$.

策略评估与更新: 在每一步$\iota$,通过求解异策略贝

尔曼方程(71)获得$P_\iota$和$K_{\iota+1}$.

**注 1**　式(71)里面不显含$K_{\iota+1}$,但是未知量$B^{\mathrm{T}}P_\iota B$和$B^{\mathrm{T}}P_\iota A$显含在其中. 根据算法4进一步可以得到$K_{\iota+1}$,在每一步$\iota$,通过收集状态与输入数据求解出$P_\iota$和$K_{\iota+1}$.

一般的VI算法收敛速度慢于PI算法[175,192-193],但是PI算法要求初始策略为容许策略,这一要求的满足通常需要知道关于系统动力学的先验知识. 为了放松这一限制, Yang等[192]和Jiang等[193]分别关于离散系统和连续系统LQR问题提出了$\lambda$-PI算法和bias-PI算法. Chen等[194]提出同伦PI用于未知线性连续系统控制,该算法也不需要容许的初始策略. 上述3篇文献同时给出了3种算法的在线学习版本,更详细地, Liu等[195]综述了ADP方法在单个系统最优状态调节、最优输出调节、最优追踪控制等方面的研究进展.

Zhang等[196-200]基于ADP方法研究了有输入约束或状态约束的非线性系统最优控制问题. Wang等[201]利用ADP方法研究了具有外部扰动的不确定非线性系统的最优控制,引入补偿扰动的控制器和最优控制器,并使用事件驱动策略更新控制器和网络权重. Liu等[202]从攻击者的角度,利用ADP方法考虑了DoS攻击下远程状态估计优化调度问题,其中攻击者的目标是设计最优的攻击策略,以最少的能量消耗降低网络物理系统的控制性能. Moghadam等[203-204]利用ADP方法研究了时滞系统的最优控制问题. Gao等[205]考虑了DOS攻击下的鲁棒输出调节问题,与现有的强化学习工作不同,该方法严格分析了闭环系统的抗攻击能力和对动态不确定性的鲁棒性.

随着理论研究的深入,ADP在单个系统中的应用也日益增多[206-221]. Tang等[209]针对基于双馈感应发电机的风电场提出了一种基于目标表达的启发式动态规划控制器,以提高故障条件下系统的暂态稳定性. Zhu等[210]研究了控制受限的多电池储能系统最小电力成本问题. Liu等[211]研究了复杂天气环境下太阳能住宅能源调度问题. 基于ADP技术, Wei等[212]提出了一种适用于智能家居能源系统管理的新型自学习电池顺序控制方案. Mu等[213]提出了一种基于动作依赖启发式动态规划的具有自适应学习能力的数据驱动辅助控制方法,用于解决吸气式高超声速飞行器跟踪控制. 针对车辆系统, Sun等[214]提出了一种无模型的控制器设计方案,以提高横向稳定性. Li等[215-217]研究了复杂海洋环境下无人驾驶水面无人艇(USV)最优追踪控制. Kong等[218]在未知非线性扰动的影响下最优化机器人的跟踪控制以提高鲁棒性. 此外,ADP在医疗健康领域也有诸多应用[219-220].

从以上分析可以看出,ADP技术呈现出从有模型算法到无模型算法的演进趋势. 目前,针对ADP应用于单个智能体(控制对象)序贯决策(最优控制)问题的研究已趋于完善,近期关于此类问题的研究主要集中在完善与改进已有算法,如加快算法收敛速度、放松算法对系统先验知识的要求、结合事件驱动技术减轻通信负担等. 此外,ADP算法在航空航天、医疗健康、电力系统等实际系统的应用也是一个重要的研究方向. 特别地,由函数近似、状态估计、测量噪声、外部/对抗性干扰、攻击等引起的各种误差,使得PI和VI算法难以准确实施. 尽管ADP技术在理论和应用两方面有很多研究,但由于迭代过程固有的非线性以及误差导致的值函数序列的非单调性,算法鲁棒性问题尚未得到充分解决. 当状态和控制输入空间无界且连续时,问题变得更加复杂. 总体而言,ADP算法对迭代误差的鲁棒性分析是目前较为突出的难点.

### 3.3　ADP在多智能体系统最优协同控制中的应用

本节将介绍ADP方法在MAS协同控制中的应用. 最优协同控制问题包括同构MAS最优一致性控制(此问题一般考虑状态达到一致)、异构MAS最优输出同步和最优输出调节问题、MAS $H_\infty$控制问题以及ADP在无人集群系统最优控制研究中的应用等. 在展开介绍ADP在MAS最优协同控制中的应用前,首先介绍图论的相关知识.

考虑如下无向图$\mathcal{G}=(\mathcal{I}_N,\mathcal{E})$,包含$N$个智能体,智能体指标集为$\mathcal{I}_N=\{1,2,\ldots,N\}$,边集为$\mathcal{E}=\{(i,j):.\, i,j\in\mathcal{I}_N\}\subseteq\mathcal{I}_N\times\mathcal{I}_N$. $(i,j)\in\mathcal{E}$表示智能体$i$与$j$可以相互传递信息. $\mathcal{A}=[a_{ij}]\in\boldsymbol{R}^{N\times N}$表示邻接矩阵,如果$(i,j)\in\mathcal{E}$且$i\neq j$,则$a_{ij}=1$,否则$a_{ij}=0$. 智能体$i$的邻居集定义为$\mathcal{N}_i=\{j\in\mathcal{I}_N:(i,j)\in\mathcal{E}\}$. 度矩阵为$\mathcal{D}=\mathrm{diag}(d_i)$,其中$d_i=\sum\limits_{j\in\mathcal{N}_i}a_{ij}$,则拉普拉斯矩阵定义为$\mathcal{L}=\mathcal{D}-\mathcal{A}$.

#### 3.3.1　ADP与多智能体系统最优一致性问题

一致性问题旨在通过设计一组分布式控制协议,使得所有智能体的状态趋于相同. 一般而言,一致性问题可以从多角度划分为:无(有)领导者的一致性控制、分组一致性(group consensus)、有限时间一致性(finite-time consensus)和均方一致性(mean-square consensus)等. 最优一致性问题是研究通过设计一组分布式控制协议,使得所有(对应组)智能体的状态趋于相同且每个智能体优化一个局部性能函数. 下面以有领导者的MAS最优一致性控制问题为例进行详

细介绍.

Vamvoudakis等[45]考虑了包含$N$个智能体的线性MAS,每个智能体的动力学为

$$\dot{x}_i(t)=Ax_i(t)+B_iu_i(t),\ i\in\mathcal{I}_N. \tag{72}$$

其中:$x_i(t)\in\boldsymbol{R}^n,u_i(t)\in\boldsymbol{R}^m,A\in\boldsymbol{R}^{n\times n},B_i\in\boldsymbol{R}^{n\times m}$,分别为第$i$个子系统的状态、控制输入和系统矩阵. 智能体之间通过无向连通图$\mathcal{G}$连接. 领导者系统的动力学为

$$\dot{x}_0(t)=Ax_0(t), \tag{73}$$

其中$x_0(t)$为领导者状态.

定义智能体$i$的局部一致性误差

$$\varepsilon_i(t)=\sum_{j\in\mathcal{N}_i}a_{ij}(x_i(t)-x_j(t))+g_i(x_i(t)-x_0(t)). \tag{74}$$

若考虑无领导者的最优一致问题,则有$g_i=0,\forall i$. 根据式(72)和(73)得到第$i$个局部一致性误差动力学为

$$\dot{\varepsilon}_i(t)=A\varepsilon_i(t)+(d_i+g_i)B_iu_i(t)-\sum_{j\in\mathcal{N}_i}a_{ij}B_ju_j(t). \tag{75}$$

对于智能体$i$,给出如下合作型性能指标:

$$J_i(\varepsilon_i(0),\varepsilon_{-i}(0),u_i(\cdot),u_{-i}(\cdot))=\frac{1}{2}\int_0^{+\infty}\Big(\varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_iu_i(t)+\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(t)R_{ij}u_j(t)\Big)\mathrm{d}t. \tag{76}$$

其中:$Q_i$、$R_i$为正定矩阵,$R_{ij}$为半正定矩阵,$\varepsilon_{-i}(t)=\{\varepsilon_j(t):j\in\mathcal{N}_i\},u_{-i}(t)=\{u_j(t):j\in\mathcal{N}_i\}$,$\mathcal{N}_i$为智能体$i$所有邻居智能体组成的集合. 固定智能体$i$和其邻居们的策略,相应的智能体$i$的值函数记为

$$V_i(\varepsilon_i(t),\varepsilon_{-i}(t))=\frac{1}{2}\int_t^{\infty}\Big(\varepsilon_i^{\mathrm{T}}(\tau)Q_i\varepsilon_i(\tau)+u_i^{\mathrm{T}}(\tau)R_iu_i(\tau)+\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(\tau)R_{ij}u_j(\tau)\Big)\mathrm{d}\tau.$$

**假设1** $V_i(\varepsilon_i(t),\varepsilon_{-i}(t))$是分布式的,即$V_i(\varepsilon_i(t),\varepsilon_{-i}(t))=V_i(\varepsilon_i(t))$.

**定义4** 如果策略$(u_i^*(t),u_{-i}^*(t))$满足$J_i(u_i^*(t),u_{-i}^*(t))\leqslant J_i(u_i(t),u_{-i}^*(t))$,则该策略称作一个Nash均衡解.

**问题1** 考虑线性MAS (72),最优一致性追踪问题的控制目标是设计一组分布式控制策略$u_i(\cdot)$,使得MAS的状态达到一致且最小化一组局部合作性能函数(76).

基于假设1,最优值函数定义为$V_i^*(\varepsilon_i(t))=\min\limits_{u_i(\cdot)}V_i(\varepsilon_i(t))$. Hamiltonian函数定义如下:

$$H_i(\varepsilon_i(t),u_i(t),u_{-i}(t),\nabla V_i(\varepsilon_i(t)))=\nabla V_i^{\mathrm{T}}(\varepsilon_i(t))\dot{\varepsilon}_i(t)+\frac{1}{2}\Big(\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(t)R_{ij}u_j(t)+\varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_iu_i(t)\Big), \tag{77}$$

其中$\dot{\varepsilon}_i(t)$在式(75)中给出. 然后,通过求解

$$\frac{\partial H_i(\varepsilon_i(t),u_i(t),u_{-i}(t),\nabla V_i(\varepsilon_i(t)))}{\partial u_i(t)}=0,$$

得到最优控制策略

$$u_i^*(t)=-(d_i+g_i)R_i^{-1}B_i^{\mathrm{T}}\nabla V_i^*(\varepsilon_i(t)), \tag{78}$$

$V_i^*(\varepsilon_i(t))$满足如下耦合HJB方程:

$$\frac{1}{2}(d_i+g_i)^2\nabla V_i^{*\mathrm{T}}(\varepsilon_i(t))B_iR_i^{-1}B_i^{\mathrm{T}}\nabla V_i^*(\varepsilon_i(t))+\frac{1}{2}\sum_{j\in\mathcal{N}_i}(d_j+g_j)^2\nabla V_j^{*\mathrm{T}}(\varepsilon_j(t))B_j\tilde{R}_{ij}^{-1}B_j^{\mathrm{T}}\nabla V_j^*(\varepsilon_j(t))+\nabla V_i^{*\mathrm{T}}(\varepsilon_i(t))A_i^c(t)+\frac{1}{2}\varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)=0. \tag{79}$$

其中

$$\tilde{R}_{ij}^{-1}=R_j^{-1}R_{ij}R_j^{-1},$$

$$A_i^c(t)=A\varepsilon_i(t)-(d_i+g_i)^2B_iR_i^{-1}B_i^{\mathrm{T}}\nabla V_i^*(\varepsilon_i(t))+\sum_{j\in\mathcal{N}_i}a_{ij}(d_j+g_j)B_jR_j^{-1}B_j^{\mathrm{T}}\nabla V_j^*(\varepsilon_j(t)).$$

**算法10** $N$-智能体合作博弈PI算法.

输入:每个智能体选择一个初始的容许策略$u_i^0(t),\forall i\in\mathcal{I}_N$,迭代次数$\iota=0,1,\ldots$;

输出:$V_i^*(x(t)),\ u_i^*(t)$.

策略评估:根据式(77)求解$V_i^\iota(\varepsilon_i(t)),\forall i\in\mathcal{I}_N$,有

$$H_i(\varepsilon_i(t),u_i^\iota(t),u_{-i}^\iota(t),\nabla V_i^\iota(\varepsilon_i(t)))=0. \tag{80}$$

策略改进:利用式(78)更新$N$个智能体的策略

$$u_i^{\iota+1}(t)=-(d_i+g_i)R_i^{-1}B_i^{\mathrm{T}}\nabla V_i^\iota(\varepsilon_i(t)). \tag{81}$$

算法10给出了利用图博弈问题版本的PI算法求解耦合的HJB方程(79). Vamvoudakis等[45]对算法的收敛性证明进行了详细阐述. 此外,基于值函数逼近技术,给出了算法10在线执行方案. 但此在线学习算法没有采用IRL技术,导致实施时需要知道系统动力学的完备信息.

**注2** 需要强调,耦合的HJB方程(79)不一定有一组解$\{V_i^*(\varepsilon_i(t))\}$,即使有解也不一定满足假设1. 但是在MAS 工程应用中,人们总是希望能够找到一组分布式策略,这促使研究者提出Minmax策略[39].

**定义5** 在微分图博弈中,如果策略$u_i^*(\cdot)$满足

$$u_i^*(\cdot)=\arg\min_{u_i(\cdot)}\max_{u_{-i}(\cdot)}J_i(\varepsilon_i(t),u_i(\cdot),u_{-i}(\cdot)), \tag{82}$$

则将其称作Minmax策略.

根据上述定义,要求修改性能函数(76),使其能够描述智能体与其邻居之间的零和博弈. 因此,定义如下新的智能体$i$的性能函数:

$$J_i(\varepsilon_i(0),u_i(t),u_{-i}(t))=\int_0^\infty \varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_iu_i(t)-\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(t)R_{ij}u_j(t)\mathrm{d}t, \tag{83}$$

其中智能体$i$期望最小化性能函数(83),且假设其邻居期望最大化式(83). 相应的智能体$i$的值函数记为

$$V_i(\varepsilon_i(t))=\int_t^\infty \varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_iu_i(t)-\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(t)R_{ij}u_j(t)\mathrm{d}t. \tag{84}$$

与式(84)相关的Hamiltonian函数定义如下:

$$H_i(\varepsilon_i(t),u_i(t),u_{-i}(t),\nabla V_i(\varepsilon_i(t)))=\nabla V_i^{\mathrm{T}}(\varepsilon_i(t))\dot{\varepsilon}_i(t)+\Big(-\sum_{j\in\mathcal{N}_i}a_{ij}u_j^{\mathrm{T}}(t)R_{ij}u_j(t)+\varepsilon_i^{\mathrm{T}}(t)Q_i\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_iu_i(t)\Big), \tag{85}$$

其中$\dot{\varepsilon}_i(t)$在式(75)中给出. 假设$V_i(\varepsilon_i(t))$具有二次型形式,即$V_i(\varepsilon_i(t))=\varepsilon_i^{\mathrm{T}}(t)P_i\varepsilon_i(t)$,则可以得到最优控制策略$u_i^*(t)=-(d_i+g_i)R_i^{-1}B_i^{\mathrm{T}}P_i^*\varepsilon_i(t)$,其中$P_i^*$满足如下解耦的HJB方程(即博弈代数黎卡提方程(game ARE, GARE)):

$$Q_i+P_iA+A^{\mathrm{T}}P_i-(d_i+g_i)P_iBR_i^{-1}B^{\mathrm{T}}P_i+\sum_{j=1}^{N}a_{ij}P_iBR_{ij}^{-1}B^{\mathrm{T}}P_i=0.$$

详细阐述可参见文献[39],但此文献并没有给出ADP方法求解GARE. 由于该方程是解耦的,可以利用单个系统求解$H_\infty$问题的PI算法或者VI算法进行求解[191],或者将GARE看作ARE利用单个系统求解LQR问题的PI算法或者VI算法进行求解[34].

可以看到,问题1考虑的是同构的连续线性MAS最优一致问题. 对于离散MAS[41-42,46]和非线性MAS有类似的结果[40,47-48]. 基于RL算法, Wang等[49]考虑了多Euler-Lagrange系统无模型事件触发最优一致性控制问题. 首先,通过定义预补偿器构造增广系统,以避免对系统动力学的依赖;其次,应用HJB方程推导了无模型事件触发最优控制器. 基于RL算法, Ji等[50-51]考虑了具有未知动力学的MAS最优二部一致性控制问题. 具有输入约束或者状态约束的MAS最优一致性控制问题也得到了大量研究[52-54]. 对于异构MAS,一般考虑输出调节问题.

除以上考虑的有(无)领导MAS一致性问题外,在实际问题中,随着MAS的规模和复杂性的增加,系统可能需要被划分为不同的子组,智能体的一致协议也会随着不同子组中环境或任务的变化而不同. 近年来,MAS分组一致控制问题开始引起关注[222]. 分组一致作为一致控制问题的扩展问题是指多智能系统中智能体分组地实现状态一致性. MAS最优分组一致是指智能体实现分组一致的同时能够最优化某些性能指标. Li等[223]利用ADP和事件驱动机制解决了二阶MAS最优分组一致问题. Peng等[224-225]研究了具有合作竞争关系的未知MAS最优二分一致追踪控制问题,并给出了无模型强化学习算法. 现有的基于ADP方法的研究成果绝大多数针对无限域近似最优一致控制问题,然而在某些实际情况下,网络中的智能体需要在有限时间内达成一致. 基于此,有限时间一致控制问题近年来受到广泛关注. 有限时间MAS最优一致控制问题是指设计控制协议使得智能体在有限时间实现状态一致的同时最小化某些性能指标. Zhang等[226]研究了具有状态时滞的未知MAS有限时间最优一致控制问题,设计了离策略强化学习算法求解最优策略. 基于强化学习算法, Liu等[227]研究了具有状态时滞的非线性MAS有限时间鲁棒事件触发控制. 此外,设计合适的一致协议以减少噪声对系统性能的影响,并考虑均方一致问题在MAS一致性研究中也引起了广泛关注[228-229],而基于ADP方法的最优均方一致性控制问题的研究成果较为少见.

总体而言,利用ADP技术求解MAS一致性控制问题的研究呈现出从解决一些特殊系统的最优一致性控制问题到解决一般系统的最优(分组)一致性控制、有限时间MAS最优(分组)一致性控制以及平均一致性问题的研究历程.

### 3.3.2　ADP与多智能体系统最优输出调节问题

Gao等[60]考虑了如下异构MAS:

$$\dot{x}_i(t)=A_ix_i(t)+B_iu_i(t)+D_ix_0(t), \tag{86}$$

$$y_i(t)=C_ix_i(t), \tag{87}$$

$$\dot{x}_0(t)=Sx_0(t), \tag{88}$$

$$y_0(t)=Rx_0(t), \tag{89}$$

$$e_i(t)=y_i(t)-y_0(t). \tag{90}$$

其中:$x_i(t)\in\boldsymbol{R}^{n_i}$、$u_i(t)\in\boldsymbol{R}^{m_i}$、$y_i(t)\in\boldsymbol{R}^r$分别为智能体$i$的状态、输入、输出;$x_0(t)\in\boldsymbol{R}^{n_0}$、$y_0(t)\in\boldsymbol{R}^r$分别为领导者系统的状态、输出;$A_i$、$B_i$、$D_i$、$C_i$、$S$、$R$为合适维数的系统矩阵;$e_i(t)$为追踪误差.

**问题2**　考虑异构MAS(86)～(90),最优输出同步问题的控制目标是设计一组分布式控制策略

$u_i(t)$,使得MAS的输出达到同步,即$e_i(t)\to 0, t\to +\infty, \forall i\in\mathcal{I}_N$.

首先考虑如下静态控制器:

$$u_i(t)=-K_ix_i(t)+L_ix_0(t), \tag{91}$$

其中$K_i$、$L_i$分别为反馈增益矩阵和前馈增益矩阵. 定义两个辅助变量$\bar{x}_i(t)=x_i(t)-\Pi_ix_0(t)$, $\bar{u}_i(t)=u_i(t)-\Gamma_ix_0(t)$,其中$\Pi_i$、$\Gamma_i$为如下调节方程的解:

$$A_i\Pi_i+B_i\Gamma_i+D_i=\Pi_iS, \tag{92}$$

$$C_i\Pi_i=R. \tag{93}$$

因此,系统(86)~(90)变为

$$\dot{\bar{x}}_i(t)=A_i\bar{x}_i(t)+B_i\bar{u}_i(t), \tag{94}$$

$$e_i(t)=C_i\bar{x}_i(t). \tag{95}$$

设计$\bar{u}_i^*(t)=-K_i^*\bar{x}_i(t)$为连续线性系统(94) LQR问题的最优控制器,根据前文介绍的LQR问题可以由ADP方法解决,给出式(91)中的控制增益$K_i=K_i^*$, $L_i=\Gamma_i+K_i^*\Pi_i$.

关于输出调节方程(92)和(93), Chen等[207]给出了基于系统辨识的求解方案, Odekunle等[208]给出了部分未知系统利用数据驱动技术的求解方案, Cai等[230]给出了自适应的求解方案. 此外,有些文献不求解输出调节方程. 如Ali等[206]通过为前馈增益设计自适应律实现追踪; Modares等[65]将MAS输出同步问题转化为单个系统最优输出调节问题,最终归结到求解ARE. 通过前文介绍, ARE可以利用ADP方法解决. 注意到,在控制器(91)中用到了$x_0(t)$,但是在分布式控制架构中并不是每个智能体都能够获得这一信息. 因此一般的处理方案是为每个智能体设计观测器用于估计$x_0(t)$,这样控制器(91)变为$u_i(t)=-K_ix_i(t)+L_i\hat{x}_0^i(t)$,其中$\hat{x}_0^i(t)$为智能体$i$的观测器状态. 针对MAS输出调节问题设计静态控制器,更详细的描述参见文献[58,65]. 基于上述思路,研究者们对各种具体的系统做了大量研究. Mu等[59]考虑切换拓扑下异构MAS的最优无模型输出同步问题,设计了完全不需要领导者动力学信息的观测器,将多智能体系统输出同步问题转化为一对一的追踪问题,并基于此设计了$Q$学习算法求解此类最优控制问题. Wen等[62,64]分别研究了非线性连续和离散MAS输出同步问题,其中系统具有严格反馈形式. 基于ADP方法, Wang等[61]解决了具有输入饱和的异构MAS半全局次优输出调节问题.

关于系统(86)~(90),基于内模原理,考虑如下动态控制器:

$$u_i(t)=-K_ix_i(t)-H_iz_i(t), \tag{96}$$

$$z_i(t)=F_iz_i(t)+G_ie_y^i(t). \tag{97}$$

其中: $K_i$、$H_i$为待设计的反馈增益矩阵; $F_i$、$G_i$为待设计内模矩阵, $e_y^i(t)=\sum\limits_{j\in\mathcal{N}_i}a_{ij}(y_i(t)-y_j(t))+g_i(y_i(t)-y_0(t))$. 反馈增益矩阵$K_i$和$H_i$的设计最后归结为求解一类GARE,此方程可以由ADP方法解决,详见文献[63].

**注3** 如果$D_i=0, \forall\, i\in\mathcal{I}_N$,则问题2转化为输出同步问题.

Li等[231]针对符号有向图上存在对抗性输入的异构MAS,研究了领导者-跟随者最优二部输出同步问题. 近年来研究人员开始关注最优分组输出一致性、有限时间最优输出一致性以及最优均方输出一致性问题. 此外, ADP算法的鲁棒性以及动态拓扑和事件驱动机制下MAS最优输出调节问题也值得进一步研究.

### 3.3.3 ADP与多智能体系统$H_\infty$控制问题

Jiao等[66]考虑了包含$N$个跟随者和一个领导者的MAS,给出智能体$i$的动力学和领导者动力学如下:

$$\dot{x}_i(t)=Ax_i(t)+B_iu_i(t)+D_iv_i(t), \tag{98}$$

$$\dot{x}_0(t)=Ax_0(t). \tag{99}$$

其中: $x_i(t)\in\boldsymbol{R}^n$、$u_i(t)\in\boldsymbol{R}^{m_i}$、$v_i(t)\in\boldsymbol{R}^{q_i}$、$x_0(t)\in\boldsymbol{R}^n$、$A$、$B_i$、$D_i$分别为智能体$i$的状态、控制输入、额外扰动、领导者状态、系统矩阵.

智能体$i$的局部一致性误差如式(74)所示,其动力学可以计算为

$$\begin{aligned}&\dot{\varepsilon}_i(t)=\\&A\varepsilon_i(t)+(d_i+g_i)B_iu_i(t)-\sum_{j\in\mathcal{N}_i}a_{ij}B_ju_j(t)+\\&(d_i+g_i)D_iv_i(t)-\sum_{j\in\mathcal{N}_i}a_{ij}D_jv_j(t).\end{aligned} \tag{100}$$

**问题3** 考虑MAS(98), $H_\infty$控制问题的控制目标是设计一组分布式控制策略$u_i(t)$,使得:

1) 当$v_i(t)=0, \forall i\in\mathcal{I}_N$时,MAS达到一致;

2) 当$v_i(t)\neq 0, \exists i\in\mathcal{I}_N$时,每个智能体满足如下扰动衰减条件:

$$\int_0^{\mathrm{T}}\|z_i(t)\|^2\mathrm{d}t\leqslant\gamma^2\int_0^{\mathrm{T}}\|\omega_i(t)\|^2\mathrm{d}t+\beta(\delta_i(0)). \tag{101}$$

其中: $\gamma$为给定的参数,且有

$$\|z_i(t)\|^2=\varepsilon_i^{\mathrm{T}}(t)Q_{ii}\varepsilon_i(t)+u_i^{\mathrm{T}}(t)R_{ii}u_i(t)+$$

$$\sum_{j\in\mathcal{N}_i} u_j^{\mathrm{T}}(t)R_{ij}u_j(t),$$

$$\|\omega_i(t)\|^2 = v_i^{\mathrm{T}}(t)T_{ii}v_i(t) + \sum_{j\in\mathcal{N}_i} v_j^{\mathrm{T}}(t)T_{ij}v_j(t).$$

对每个智能体定义如下性能函数:

$$J_i(\varepsilon_i(0), u_i(t), u_{-i}(t), v_i(t), v_{-i}(t)) = \frac{1}{2}\int_0^\infty \varepsilon_i^{\mathrm{T}}(t)Q_{ii}\varepsilon_i(t) + u_i^{\mathrm{T}}(t)R_{ii}u_i(t) + \sum_{j\in\mathcal{N}_i} u_j^{\mathrm{T}}(t)R_{ij}u_j(t) - \gamma^2 v_i^{\mathrm{T}}(t)T_{ii}v_i(t) - \gamma^2\sum_{j\in\mathcal{N}_i} v_j^{\mathrm{T}}(t)T_{ij}v_j(t)\mathrm{d}t. \tag{102}$$

相应的值函数定义为

$$V_i(\varepsilon_i(t)) = \frac{1}{2}\int_t^\infty \varepsilon_i^{\mathrm{T}}(\tau)Q_{ii}\varepsilon_i(\tau) + u_i^{\mathrm{T}}(t)R_{ii}u_i(\tau) + \sum_{j\in\mathcal{N}_i} u_j^{\mathrm{T}}(\tau)R_{ij}u_j(\tau) - \gamma^2 v_i^{\mathrm{T}}(\tau)T_{ii}v_i(\tau) - - \gamma^2\sum_{j\in\mathcal{N}_i} v_j^{\mathrm{T}}(\tau)T_{ij}v_j(\tau)\mathrm{d}\tau. \tag{103}$$

问题3等价于求解如下零和博弈问题:

$$V_i(\varepsilon_i(0)) = \min_{u_i(t)}\max_{v_i(t)} J_i(\varepsilon_i(0), u_i(t), u_{-i}(t), v_i(t), v_{-i}(t)).$$

求解上述博弈问题等价于求解一组耦合的HJB方程, 详细过程见文献[66]. 该文献利用同策略技术给出了多智能体PI算法, 其缺点除了前面提到的有偏之外还有假设扰动策略是迭代更新的(这是不符合实际的). Zhao等[67-69]基于事件触发技术研究了非线性连续MAS $H_\infty$控制问题.

### 3.3.4　ADP在无人集群系统最优控制研究中的应用

Zhao等[71]考虑如下遭受执行器故障的四旋翼无人机系统:

$$\ddot{\Theta}_i(t) = -J_{\Theta i}^{-1}(C(\Theta_i(t), \dot{\Theta}_i(t))\dot{\Theta}_i(t)) + b_{\tau i}(u_{\tau i}(t) - \rho(t - t_{si})u_{\Delta i}(t)). \tag{104}$$

其中: $i \in \mathcal{I}_N$; $\Theta_i(t) \in \boldsymbol{R}^3$为系统状态, 具体表示为四旋翼的欧拉角; $u_{\tau i}(t)$为系统输入; $\rho(t-t_{si})$为阶跃函数; $t_{si}$为故障开始时间; $u_{\Delta i}(t)$为执行器故障; $J_{\Theta i}$、$b_{\tau i}$为系统参数; $C(\Theta_i(t), \dot{\Theta}_i(t))$为一个非线性函数. 令$x_i(t) = (\Theta_i^{\mathrm{T}}(t), \dot{\Theta}_i^{\mathrm{T}}(t))$, 则式(98)可以改写为

$$\dot{x}_i(t) = f_{\Theta i}(x_i(t)) + B_{\tau i}(u_{\tau i}(t) - \rho(t - t_{si})u_{\Delta i}(t)),$$

其中$f_{\Theta i}(x_i(t))$和$B_{\tau i}$分别为适当的非线性函数和常矩阵. 领导者系统为$\dot{x}_0(t) = A_{\Theta 0}x_0(t)$, 其中$A_{\Theta 0}$和$x_0(t)$分别为领导者系统矩阵和状态.

通过设计观测器估计领导者状态, 进一步, 建立如下增广系统:

$$\dot{X}_i(t) = F_{\Theta i}(X_i(t)) + \bar{B}_i(u_{\tau i}(t) + \rho(t - t_{si})u_{\Delta i}(t)) + M_i\varepsilon_i(t). \tag{105}$$

其中: $\varsigma_i(t)$为观测器状态, $X_i(t) = (x_i^{\mathrm{T}}(t), \varsigma_i^{\mathrm{T}}(t))^{\mathrm{T}}$, $F_{\Theta i}(X_i(t)) = (f_{\Theta i}^{\mathrm{T}}(x_i(t)), \varsigma_i^{\mathrm{T}}(t)A_{\Theta 0}^{\mathrm{T}})^{\mathrm{T}}$, $\bar{B}_i = (B_{\tau i}^{\mathrm{T}}, 0)^{\mathrm{T}}$, $M_i = (0, -v)^{\mathrm{T}}$, $v$为一个参数, $\varepsilon_i(t) = \sum_{j\in N_i} a_{ij}(\varsigma_i(t) - \varsigma_j(t)) + g_i(\varsigma_i(t) - x_0(t))$. 对于正常的没有遭受故障的系统(105), 建立如下最优控制问题.

寻找正常的控制律最小化如下性能函数:

$$V_i(X_i(0)) = \int_0^\infty e_{\Theta i}^{\mathrm{T}}(t)Q_{\Theta i}e_{\Theta i}(t) + u_{\tau i}^{\mathrm{T}}(t)R_{\Theta i}u_{\tau i}(t)\mathrm{d}t, \tag{106}$$

其中$e_{\Theta i}(t) = [I_3, 0_{3\times 3}, -I_3, 0_{3\times 3}]X_i(t)$. 随后建立鲁棒控制器. 最优控制问题的解可以根据单个系统ADP方法得到, 详细介绍参见文献[71].

基于积分强化学习算法, He等[72]考虑了存在未知环境干扰情况下的多机器人最小时间路径规划和避碰问题. 基于多智能体ADP技术, Liu等[232]研究了人-多机器人协作优化控制问题. Tan[233]针对存在不确定性的非完整约束移动多机器人系统, 研究了在运动学和动力学方面都具有干扰抑制的分布式跟踪控制器设计问题. 基于数据驱动强化学习算法, Zhao等[73]讨论了具有虚拟领导者的异构四旋翼无人机系统最优编队控制问题. Zhao等[74]重点研究了通信链路故障和执行器故障下空地协调多异构车辆的编队控制问题. Lin等[75-77]研究了遭受扰动的具有部分未知动力学的异构MAS 鲁棒编队控制问题. 基于强化学习的离策略算法, Zhao等[78]研究了由无人机和无人车组成的异构MAS鲁棒最优编队控制, 其中四旋翼无人机和无人驾驶地面车辆的位置以及航向的参考信号仅通过它们自己与邻居的本地信息估计生成. 基于事件触发技术, Dou等[79]研究了多四旋翼无人机分布式编队控制问题.

可以看出, ADP技术在UAV编队控制和姿态控制、UGV编队控制以及USV最优路径规划等方面已取得较多的研究成果. 考虑到一些实际任务需要USV、UAV 和UGV 跨域联合完成, 将ADP技术应用于求解跨域异构无人系统的最优协同控制问题将会是一个研究热点.

### 3.3.5　IRL与多智能体系统逆最优控制问题

逆强化学习(inverse reinforcement learning, IRL)算法是由RL算法衍生出来的一种用于解决逆最优控制(inverse optimal control, IOC)问题(也称作IRL

问题)的学习算法. 最优控制问题的目标是寻找一个针对已知的性能函数的最优控制律. 然而,智能体的性能目标在一些实际场景中是未知的. 在这一背景下,IOC问题被人们提出并受到广泛关注. IOC问题是最优控制的逆问题,目标在于寻找一个稳定的和有一定意义的性能函数,且针对这一性能函数智能体的控制律是最优的. 粗略来讲,IRL技术是一种试图通过观察智能体的状态演化行为来重建其未知的性能目标的技术.

近年来,一些学者围绕利用IRL方法解决IOC及其相关问题做了一些探索. 对于单个线(非)性智能体系统,一些研究人员基于IRL方法分别考虑了追踪控制、对抗性学徒博弈、非零和多人博弈等问题[234-240]. 对于MAS,研究人员也做了一些研究与探索. 具体地,Donge等[241]考虑了MAS 图学徒博弈问题. Neumeyer等[242]设计了一种新的IRL算法来有效地推断具有有界理性(次优的)智能体的随机博弈中的性能函数. Belfo等[243]针对离散时间非线性多智能体系统,提出了一种鲁棒分布式IOC框架,其中每个智能体的动力学直接受到相邻智能体的状态和输入以及其他干扰信号项的影响.

## 4 总结与展望

本文综述了RL和ADP算法在人工智能和系统与控制领域的发展历程,着重介绍了其应用于从单个智能体(控制对象)序贯决策(最优控制)问题到MAS序贯决策(最优协调控制)问题的研究进展. 从控制中心与智能体、以及智能体之间信息交互模式的角度对MARL算法进行了更为细致的分类与讨论. 进一步地,在简要介绍ADP算法的结构变化历程和其从基于模型的离线规划到无模型的在线学习发展演进的基础上,综述了ADP算法在MAS最优协调控制问题中的研究进展. 目前,RL和ADP算法的理论及应用研究均呈现出蓬勃发展的趋势,一些新颖且有效的算法如雨后春笋般涌现. 需要特别指出的是,囿于作者水平有限且论文篇幅限制,一些关于RL和ADP算法理论及其应用研究方面的优秀研究工作未能涵盖在本综述中. 最后,根据对当前研究现状的分析以及存在问题的思考,给出关于MARL算法研究和利用ADP算法解决MAS协调控制问题研究中的一些未来研究方向.

### 4.1 MARL算法研究中的若干未来研究方向

1) 集中式学习算法的“维数灾难”问题. 集中式学习算法利用所有智能体的观测、动作以及收益信息,训练智能体最优的动作策略,具有收敛速度快、稳定性高等优点. 然而,随着智能体个数增加,环境状态空间和动作空间呈指数级增加,强化学习算法计算量以及收敛时间均将快速上升,因此难以应用于大规模多智能体强化学习任务. 在完全信息场景下,如何高效地整合共享信息并从中提取有效信息,降低通讯以及计算要求仍有待进一步研究.

2) 分散式算法的非平稳性挑战. 分散式学习算法可以控制单个智能体强化学习问题规模,只需根据局部观测信息及奖励信息即可更新本地值函数或策略网络参数,应用较为简便且算法实现具有较大灵活性. 然而,由于忽略了环境中同时进行策略学习的其他智能体,本地智能体强化学习过程中环境平稳性假设遭到破坏,学习过程的稳定性和强化学习算法的收敛性难以分析和保证. 因此,在无法进行通信的场景下,如何改进分散式算法以更好地克服非平稳性挑战仍有待进一步研究.

3) 分布式算法的设计与应用问题. 分布式强化学习算法综合了集中式算法和分散式算法的优势,利用智能体间的通讯获得更多全局状态和动作信息,进而降低了环境非平稳性影响,提高学习过程的稳定性和收敛性. 然而,不同的通讯策略以及通讯模式对强化学习算法的稳定性和收敛性具有较大影响. 同时,通讯过程中的干扰、噪声、时滞等不确定性也会影响学习算法的收敛性和最优策略网络性能. 针对完全分布式算法,如何设计合理高效的通讯策略,以及在通信受限场景下分析分布式学习算法性能仍有待进一步研究.

4) 博弈强化学习算法的可扩展性问题. 目前大多数多智能体强化学习算法主要关注完全合作式应用场景. 然而许多实际应用领域中,如智能电网、智能交通网络等,多智能体间具有合作、竞争、混合博弈等多种交互影响关系,此时现有的集中式、分散式或分布式多智能体强化学习算法普遍具有计算复杂度高、可扩展性差、算法设计复杂等缺陷. 如何基于分布式学习思想,设计更加高效、稳定以及具有普适性的博弈强化学习算法是需要进一步研究的方向.

5) MARL算法应用于实际场景中的一些挑战问题. 虽然近年来多智能体强化学习算法及相关理论得到快速发展,大量经典算法、框架不断涌现,但大多数结果是通过仿真实验对算法有效性进行验证的. 已有不少研究表明,将强化学习算法从仿真环境应用到实际环境过程中,存在仿真到实际隔阂(Sim-to-real gap). 因此,算法设计过程中尽可能考虑实际应用场景中面临的多种非理想因素并融合一些在线

学习技术将是未来值得研究的课题. 此外,如何有效地解决实际场景中的强耦合约束、部分可观测问题、通信不稳定、稀疏奖励、恶意攻击等问题亦是值得进一步研究的课题.

### 4.2 ADP算法在解决MAS最优协调控制问题研究中的一些未来研究方向

1) 有限时间ADP算法的设计与分析及其在MAS最优协调控制问题中的应用,如有限时间最优一致追踪、平均一致问题以及有限时间分组一致问题等. 大部分已有的ADP算法基于传统最优控制理论设计满足闭环系统稳定性和性能指标最优性的反馈控制律,基于学习出的控制律,闭环系统一般只能实现渐近稳定. 然而,许多实际应用任务中,如卫星姿态调节、火控系统瞄准等,闭环系统需在有限时间内达到指定性能指标. 因此,有限时间ADP算法设计与分析是值得进一步研究及发展的方向.

2) 基于输出反馈的ADP算法的设计与分析及其在MAS最优协调控制问题中的应用,如状态不可测量情况下的最优控制问题等. 目前大多数基于ADP算法的MAS最优协调控制器均依赖于状态反馈. 然而,由于智能体感知能力受限或者成本约束,全状态信息一般难以实时获得. 基于输入信息的ADP算法可以降低智能体对于感知能力的要求,进一步可降低传感器成本,扩大智能体应用范围. 因此,研究基于输入反馈的ADP算法是拓展当前状态反馈成果,完善ADP相关理论的另一重要研究方向.

3) 利用ADP算法解决存在输入或状态约束的MAS最优协调控制问题,如有约束的MAS最优一致性控制问题、输出调节问题等. 现有ADP算法大多只考虑系统动力学约束下的最优性能指标控制问题,对于存在系统执行器饱和约束以及系统状态约束情形还没有形成系统性的算法设计及分析理论. 考虑到实际控制对象执行机构一般具有饱和约束特性,同时系统运动过程中由于安全性等因素可能存在状态约束,设计ADP算法解决存在执行器机构饱和约束以及状态约束的MAS最优协同控制问题是一个值得研究的问题.

4) 设计具有一定安全性能保证的ADP算法,如恶意攻击下ADP算法的收敛性等. 随着网络化控制的发展,传统ADP算法可能基于无线通讯网络进行部署实施. 通讯网络的引入一方面增加了控制系统的灵活性和开放性,另一方面也使得闭环网络化多智能体系统存在被攻击的可能[244],增加了控制算法执行过程中对于网络攻击的脆弱性. 网络攻击环境下,可能存在的恶意破坏包括攻击者恶意篡改甚至操纵系统信号、发动传感器/执行器网络协同攻击等. 因此,分析网络攻击对ADP算法的定性和定量影响,并在此基础上设计具有安全性能保证的ADP算法是非常必要的.

5) 利用ADP算法解决工程领域MAS系统的最优调控控制问题,如无人机-无人艇跨域集群系统最优协调控制、无人机-无人车跨域集群系统最优协调控制、无人机最优编队控制、多机器人系统最优协调控制等. 近年来,多类ADP算法已被广泛提出和采纳,并在仿真实验中展现出较好的学习控制效果. 然而,与多智能体强化学习算法面临的挑战一致,ADP算法在实际控制系统中的应用仍需要进一步研究和验证.

## 参考文献(References)


[1] Lewis F L, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control[J]. IEEE Circuits and Systems Magazine, 2009, 9(3): 32-50.

[2] 徐昕. 增强学习与近似动态规划[M]. 北京: 科学出版社, 2010: 1-31.
(Xu X. Reinforcement learning and approximate dynamic programming[M]. Beijing: Science Press, 2010: 1-31.)

[3] Lewis F L, Liu D R. Reinforcement learning and approximate dynamic programming for feedback control[M]. Hoboken: Wiley, 2013: 1-30.

[4] Meyn S P. Control systems and reinforcement learning[M]. Cambridge: Cambridge University Press, 2022: 1-50.

[5] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. The 2nd edition. Cambridge: MIT Press, 2018.

[6] Pavlov I P. Conditioned reexes[M]. London: Oxford University Press, 1927.

[7] Minsky B M L. Theory of neural-analog reinforcement systems and its application to the brain model problem[M]. Princeton: Princeton University, 1954: 1-7.

[8] Waltz M, Fu K. A heuristic approach to reinforcement learning control systems[J]. IEEE Transactions on Automatic Control, 1965, 10(4): 390-398.

[9] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep reinforcement learning: A brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[10] Kostrikov I, Yarats D, Fergus R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels[J/OL]. 2020, arXiv: 2004.13649.

[11] Wang F Y, Zhang J J, Zheng X H, et al. Where does AlphaGo go: From church-Turing thesis to AlphaGo thesis and beyond[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(2): 113-120.

[12] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J].

Nature, 2016, 529(7587): 484-489.

[13] Berner C, Brockman G, Chan B, et al. Dota 2 with large scale deep reinforcement learning[J/OL]. 2019, arXiv: 1912.06680.

[14] Ibarz J, Tan J, Finn C, et al. How to train your robot with deep reinforcement learning: Lessons we have learned[J]. International Journal of Robotics Research, 2021, 40(4/5): 698-721.

[15] Zhang Y, Zhu H H, Tang D B, et al. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems[J]. Robotics and Computer-Integrated Manufacturing, 2022, 78: 102412.

[16] Luo S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning[J]. Applied Soft Computing, 2020, 91: 106208.

[17] Liu C L, Chang C C, Tseng C J. Actor-critic deep reinforcement learning for solving job shop scheduling problems[J]. IEEE Access, 2020, 8: 71752-71762.

[18] Yu L, Sun Y, Xu Z B, et al. Multi-agent deep reinforcement learning for HVAC control in commercial buildings[J]. IEEE Transactions on Smart Grid, 2021, 12(1): 407-419.

[19] 张韵悦, 孙志毅, 孙前来, 等. 基于强化学习的挖掘机时间最优轨迹规划[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2022.0811.
(Zhang Y Y, Sun Z Y, Sun Q L, et al. Time optimal trajectory planning of excavator based on deep reinforcement learning[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.0811.)

[20] 闫敬, 徐龙, 曹文强, 等. 基于深度强化学习的多潜器编队控制算法设计[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2022.1424.
(Yan J, Xu L, Cao W Q, et al. Design of formation control algorithm for multiple autonomous underwater vehicles based on deep reinforcement learning[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.1424.)

[21] 隋丽蓉, 高曙, 何伟. 基于多智能体深度强化学习的船舶协同避碰策略研究[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2022.1159.
(Sui L R, Gao S, He W. Research on ship cooperative collision avoidance strategy based on multi-agent deep reinforcement learning[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.1159.)

[22] Yu C, Wang X, Xu X, et al. Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(2): 735-748.

[23] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.

[24] Ye Y J, Tang Y, Wang H Y, et al. A scalable privacy-preserving multi-agent deep reinforcement learning approach for large-scale peer-to-peer transactive energy trading[J]. IEEE Transactions on Smart Grid, 2021, 12(6): 5185-5200.

[25] Zhang Y, Yang Q Y, An D, et al. Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid[J]. IEEE Transactions on Cybernetics, DOI: 10.1109/TCYB.2022.3165074.

[26] Ding R J, Yang Y W, Liu J, et al. Packet routing against network congestion: A deep multi-agent reinforcement learning approach[C]. International Conference on Computing, Networking and Communications. Big Island, 2020: 932-937.

[27] You X Y, Li X J, Xu Y D, et al. Toward packet routing with fully distributed multiagent deep reinforcement learning[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2): 855-868.

[28] Chen Y J, Chang D K, Zhang C. Autonomous tracking using a swarm of UAVs: A constrained multi-agent reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13702-13717.

[29] Sacco A, Esposito F, Marchetto G, et al. Sustainable task offloading in UAV networks via multi-agent reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2021, 70(5): 5003-5015.

[30] Zhang J D, Yang Q M, Shi G Q, et al. UAV cooperative air combat maneuver decision based on multi-agent reinforcement learning[J]. Journal of Systems Engineering and Electronics, 2021, 32(6): 1421-1438.

[31] Matignon L, Laurent G J, Le F N. Review: Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems[J]. The Knowledge Engineering Review, 2012, 27(1): 1-31.

[32] Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents[C]. Machine Learning Proceedings 1993. Amsterdam: Elsevier, 1993: 330-337.

[33] Bellman R E. Dynamic programming[M]. Princeton: Princeton University Press, 1957.

[34] Jiang Y, Jiang Z P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics[J]. Automatica, 2012, 48(10): 2699-2704.

[35] Werbos P J. Advanced forecasting methods for global crisis warning and models of intelligence[J]. General Systems Yearbook, 1977, 22: 2538.

[36] Werbos P J. Beyond regression: New tools for prediction and analysis in the behavioral sciences[D]. Cambridge: Harvard University, 1975.

[37] Werbos P J. Approximate dynamic programming for realtime control and neural modelling[S]. Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, 1992: 493-525.

[38] Werbos P J. Neural networks for control and system identification[C]. Proceedings of the 28th IEEE Conference on Decision and Control. Tampa, 2002: 260-265.

[39] Lopez V G, Lewis F L, Wan Y, et al. Stability and robustness analysis of minmax solutions for differential graphical games[J]. Automatica, 2020, 121: 109177.

[40] Zhang J L, Zhang H G, Feng T. Distributed optimal consensus control for nonlinear multiagent system with unknown dynamic[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(8): 3339-3348.

[41] Abouheaf M I, Lewis F L, Vamvoudakis K G, et al. Multi-agent discrete-time graphical games and reinforcement learning solutions[J]. Automatica, 2014, 50(12): 3038-3053.

[42] Zhang H G, Jiang H, Luo Y H, et al. Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method[J]. IEEE Transactions on Industrial Electronics, 2017, 64(5): 4091-4100.

[43] Lopez V G, Lewis F L, Wan Y, et al. Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors[J]. IEEE Transactions on Automatic Control, 2020, 65(5): 1911-1923.

[44] Lopez V G, Wan Y, Lewis F L. Bayesian graphical games for synchronization in networks of dynamical systems[J]. IEEE Transactions on Control of Network Systems, 2019, 7(2): 1028-1039.

[45] Vamvoudakis K G, Lewis F L, Hudas G R. Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality[J]. Automatica, 2012, 48(8): 1598-1611.

[46] Yang X D, Zhang H, Wang Z P. Data-based optimal consensus control for multiagent systems with policy gradient reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(8): 3872-3883.

[47] Zhang H G, Zhang J L, Yang G H, et al. Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming[J]. IEEE Transactions on Fuzzy Systems, 2015, 23(1): 152-163.

[48] Wang H, Li M. Model-free reinforcement learning for fully cooperative consensus problem of nonlinear multiagent systems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(4): 1482-1491.

[49] Wang S W, Jin X, Mao S, et al. Model-free event-triggered optimal consensus control of multiple Euler-Lagrange systems via reinforcement learning[J]. IEEE Transactions on Network Science and Engineering, 2021, 8(1): 246-258.

[50] Ji L H, Li X, Zhang C J, et al. Optimal group consensus control for the second-order agents in the coopetition networks via adaptive dynamic programming and event-triggered methods[J]. Optimal Control Applications and Methods, 2022, 43(5): 1546-1567.

[51] Zhang J, Chen Y, Hu J, et al. Optimal bipartite consensus control for unknown coopetition multi-agent systems with time-delay via reinforcement learning method[C]. Proceedings of IECON 2022 48th Annual Conference of the IEEE Industrial Electronics Society. Brussels: IEEE, 2022: 1-8.

[52] Shi Z Q, Zhou C. Distributed optimal consensus control for nonlinear multi-agent systems with input saturation based on event-triggered adaptive dynamic programming method[J]. International Journal of Control, 2022, 95(2): 282-294.

[53] Xu J H, Wang L J, Liu Y, et al. Finite-time adaptive optimal consensus control for multi-agent systems subject to time-varying output constraints[J]. Applied Mathematics and Computation, 2022, 427: 127176.

[54] Li K W, Li Y M. Fuzzy adaptive optimal consensus fault-tolerant control for stochastic nonlinear multiagent systems[J]. IEEE Transactions on Fuzzy Systems, 2022, 30(8): 2870-2885.

[55] Qasem O, Jebari K, Gao W N. Adaptive dynamic programming and data-driven cooperative optimal output regulation with adaptive observers[J/OL]. 2022, arXiv: 2209.12225.

[56] Zhang H G, Liang H J, Wang Z S, et al. Optimal output regulation for heterogeneous multiagent systems via adaptive dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(1): 18-29.

[57] Gao W N, Liu Y Y, Odekunle A, et al. Adaptive dynamic programming and cooperative output regulation of discrete-time multi-agent systems[J]. International Journal of Control, Automation and Systems, 2018, 16(5): 2273-2281.

[58] Gao W N, Jiang Y, Davari M. Data-driven cooperative output regulation of multi-agent systems via robust adaptive dynamic programming[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2019, 66(3): 447-451.

[59] Mu C X, Zhao Q, Sun C Y. Optimal model-free output synchronization of heterogeneous multiagent systems under switching topologies[J]. IEEE Transactions on Industrial Electronics, 2020, 67(12): 10951-10964.

[60] Gao W N, Jiang Z P, Lewis F L, et al. Cooperative optimal output regulation of multi-agent systems using adaptive dynamic programming[C]. American Control Conference. Seattle, 2017: 2674-2679.

[61] Wang B J, Xu L, Yi X L, et al. Semiglobal suboptimal output regulation for heterogeneous multi-agent systems with input saturation via adaptive dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2022.3191673.

[62] Wen G, Chen C L P. Optimized backstepping consensus control using reinforcement learning

for a class of nonlinear strict-feedback-dynamic multi-agent systems[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3105548.

[63] Chen C, Lewis F L, Xie K, et al. Off-policy learning for adaptive optimal output synchronization of heterogeneous multi-agent systems[J]. Automatica, 2020, 119: 109081.

[64] Jiang Y, Fan J L, Gao W N, et al. Cooperative adaptive optimal output regulation of nonlinear discrete-time multi-agent systems[J]. Automatica, 2020, 121: 109149.

[65] Modares H, Nageshrao S P, Lopes G A D, et al. Optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning[J]. Automatica, 2016, 71: 334-341.

[66] Jiao Q, Modares H, Xu S Y, et al. Multi-agent zero-sum differential graphical games for disturbance rejection in distributed control[J]. Automatica, 2016, 69: 24-34.

[67] Zhao W, Yu W W, Zhang H P. Event-triggered optimal consensus tracking control for multi-agent systems with unknown internal states and disturbances[J]. Nonlinear Analysis: Hybrid Systems, 2019, 33: 227-248.

[68] Sun J L, Long T. Event-triggered distributed zero-sum differential game for nonlinear multi-agent systems using adaptive dynamic programming[J]. ISA Transactions, 2021, 110: 39-52.

[69] Chen Z T, Chen K R, Chen S Z, et al. Event-triggered $H_1$ consensus for uncertain nonlinear systems using integral sliding mode based adaptive dynamic programming[J]. Neural Networks, 2022, 156: 258-270.

[70] Luy N T. Distributed cooperative $H_1$ optimal tracking control of MIMO nonlinear multi-agent systems in strict-feedback form via adaptive dynamic programming[J]. International Journal of Control, 2018, 91(4): 952-968.

[71] Zhao W B, Liu H, Lewis F L. Data-driven fault-tolerant control for attitude synchronization of nonlinear quadrotors[J]. IEEE Transactions on Automatic Control, 2021, 66(11): 5584-5591.

[72] He C Y, Wan Y, Gu Y X, et al. Integral reinforcement learning-based multi-robot minimum time-energy path planning subject to collision avoidance and unknown environmental disturbances[J]. IEEE Control Systems Letters, 2021, 5(3): 983-988.

[73] Zhao W B, Liu H, Lewis F L, et al. Data-driven optimal formation control for quadrotor team with unknown dynamics[J]. IEEE Transactions on Cybernetics, 2022, 52(8): 7889-7898.

[74] Zhao W B, Liu H, Valavanis K P, et al. Fault-tolerant formation control for heterogeneous vehicles via reinforcement learning[J]. IEEE Transactions on Aerospace and Electronic Systems, 2022, 58(4): 2796-2806.

[75] Lin W, Zhao W B, Liu H. Robust optimal formation control of heterogeneous multi-agent system via reinforcement learning[J]. IEEE Access, 2020, 8: 218424-218432.

[76] Chen L, Dong C, He S, et al. Adaptive optimal formation control for unmanned surface vehicles with guaranteed performance using actorcritic learning architecture[J]. International Journal of Robust and Nonlinear Control, DOI: 10.1002/rnc.

[77] Yan B, Shi P, Lim C C, et al. Optimal robust formation control for heterogeneous multi-agent systems based on reinforcement learning[J]. International Journal of Robust and Nonlinear Control, 2022, 32(5): 2683-2704.

[78] Zhao W B, Liu H, Wan Y, et al. Data-driven formation control for multiple heterogeneous vehicles in air-ground coordination[J]. IEEE Transactions on Control of Network Systems, 2022, 9(4): 1851-1862.

[79] Dou L Q, Cai S Y, Zhang X Y, et al. Event-triggered-based adaptive dynamic programming for distributed formation control of multi-UAV[J]. Journal of the Franklin Institute, 2022, 359(8): 3671-3691.

[80] 杜威, 丁世飞. 多智能体强化学习综述[J]. 计算机科学, 2019, 46(8): 1-8.
(Du W, Ding S F. Overview on multi-agent reinforcement learning[J]. Computer Science, 2019, 46(8): 1-8.)

[81] 孙长银, 穆朝絮. 多智能体深度强化学习的若干关键科学问题[J]. 自动化学报, 2020, 46(7): 1301-1312.
(Sun C Y, Mu C X. Important scientific problems of multi-agent deep reinforcement learning[J]. Acta Automatica Sinica, 2020, 46(7): 1301-1312.)

[82] 梁星星, 冯旸赫, 马扬, 等. 多Agent深度强化学习综述[J]. 自动化学报, 2020, 46(12): 2537-2557.
(Liang X X, Feng Y H, Ma Y, et al. Deep multi-agent reinforcement learning: A survey[J]. Acta Automatica Sinica, 2020, 46(12): 2537-2557.)

[83] Zhang K, Yang Z, Basar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms[S]. Handbook of Reinforcement Learning and Control, Cham: Springer, 2021: 321-384.

[84] 张化光, 张欣, 罗艳红, 等. 自适应动态规划综述[J]. 自动化学报, 2013, 39(4): 303-311.
(Zhang H G, Zhang X, Luo Y H, et al. An overview of research on adaptive dynamic programming[J]. Acta Automatica Sinica, 2013, 39(4): 303-311.)

[85] Sutton R S. Learning to predict by the methods of temporal differences[J]. Machine Learning, 1988, 3(1): 9-44.

[86] Rummery G A, Niranjan M. On-line $Q$-learning using connectionist systems[M]. Cambridge: University of Cambridge, 1994: 5-7.

[87] Watkins C J C H, Dayan P. $Q$-learning[J]. Machine Learning, 1992, 8(3): 279-292.

[88] Nian R, Liu J F, Huang B. A review on reinforcement learning: Introduction and applications in industrial process control[J]. Computers & Chemical Engineering, 2020, 139: 106886.

[89] Konidaris G, Osentoski S, Thomas P. Value function

approximation in reinforcement learning using the fourier basis[C]. Proceedings of the 25th AAAI Conference on Articial Intelligence. San Francisco: AAAI Press, 2011: 380-385.

[90] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J/OL]. 2013, arXiv: 1312.5602.

[91] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double *Q*-learning[C]. Proceedings of the 30th AAAI Conference on Articial Intelligence. Phoenix: AAAI Press, 2016: 1-8.

[92] 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: ACM Press, 2016: 1995-2003.

[93] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]. Proceedings of the 32nd AAAI Conference on Articial Intelligence. Louisiana: AAAI Press, 2018: 352-366.

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

[95] Bellemare M G, Dabney W, Munos R. A distributional perspective on reinforcement learning[C]. Proceedings of the 34th International Conference on Machine Learning. Sydney: ACM Press, 2017: 449-458.

[96] Fortunato M, Azar M G, Piot B, et al. Noisy networks for exploration[J/OL]. 2017, arXiv: 1706.10295.

[97] Sutton R S, Mcallester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation[C]. Proceedings of the 12th International Conference on Neural Information Processing Systems. Denver, 1999: 1057-1063.

[98] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3/4): 229-256.

[99] Konda V, Tsitsiklis J. Actor-critic algorithms[C]. Proceedings of the 12th Conference on Neural Information Processing Systems. Denver: NIPS, 1999: 1-11.

[100] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]. Proceedings of the 32nd International Conference on Machine Learning. Lille: ACM Press, 2015: 1889-1897.

[101] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J/OL]. 2017, arXiv: 1707.06347.

[102] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ACM Press, 2018: 1861-1870.

[103] Peters J, Schaal S. Natural actor-critic[J]. Neurocomputing, 2008, 71(7/8/9): 1180-1190.

[104] Schulman J, Moritz P, Levine S, et al. High-dimensional continuous control using generalized advantage estimation[J/OL]. 2015, arXiv: 1506.02438.

[105] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. Proceedings of the 31st International Conference on Machine Learning. Beijing: ACM Press, 2014: 387-395.

[106] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J/OL]. 2015, arXiv: 1509.02971.

[107] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ACM Press, 2018: 1587-1596.

[108] 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: ACM Press, 2016: 1928-1937.

[109] Zimmer M, Glanois C, Siddique U, et al. Learning fair policies in decentralized cooperative multi-agent reinforcement learning[C]. Proceedings of the 38th International Conference on Machine Learning. Online: ACM Press, 2021: 12967-12978.

[110] Yu C, Velu A, Vinitsky E, et al. The surprising effectiveness of PPO in cooperative, multi-agent games[J/OL]. 2021, arXiv: 2103.01955.

[111] Foerster J N, Assael Y M, de Freitas N, et al. Learning to communicate with deep multi-agent reinforcement learning[C]. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, 2016: 2145-2153.

[112] Sukhbaatar S, Fergus R. Learning multiagent communication with backpropagation[C]. Proceedngs of the 29th Conference on Neural Information Processing Systems. Barcelona: NIPS, 2016: 2252-2260.

[113] Wang L, Zhang Y , Hu Y , et al. Individual reward assisted multi-agent reinforcement learning[C]. Proceedings of the 39th International Conference on Machine Learning. Baltimore: ACM Press, 2022: 23417-23432.

[114] Yuan L, Wang J, Zhang F, et al. Multi-agent incentive communication via decentralized teammate modeling[C]. Proceedings of the 36th AAAI Conference on Articial Intelligence. Ottawa: AAAI Press, 2022: 9466-9474.

[115] de Witt C S, Gupta T, Makoviichuk D, et al. Is independent learning all You need in the StarCraft multi-agent challenge? [J/OL]. 2020, arXiv: 2011.09533.

[116] Lauer M, Riedmiller M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems[C]. Proceedings of the 17th International Conference on Machine Learning. Stanford: ACM Press, 2000: 11.

[117] Matignon L, Laurent G J, Le Fort-Piat N. Hysteretic *Q*-learning: An algorithm for decentralized reinforcement learning in cooperative multi-agent

teams[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. San Diego, 2007: 64-69.

[118] Omidshaei S, Pazis J, Amato C, et al. Deep decentralized multi-task multi-agent reinforcement learning under partial observability[C]. Proceedings of the 15th International Conference on Machine Learning. Madison: ACM Press, 2017: 2681-2690.

[119] Panait L, Sullivan K, Luke S A. Lenient learners in cooperative multiagent systems[C]. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems. Hakodate, 2006: 801-803.

[120] Wei E, Luke S. Lenient learning in independent-learner stochastic cooperative games[J]. The Journal of Machine Learning Research, 2016, 17(1): 2914-2955.

[121] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Stockholm, 2018: 443-451.

[122] Littman M L. Markov games as a framework for multi-agent reinforcement learning[C]. Machine Learning Proceedings 1994. Amsterdam: Elsevier, 1994: 157-163.

[123] Fan J, Wang Z, Xie Y, et al. A theoretical analysis of deep $Q$-learning[C]. Proceedings of the 2rd Learning for Dynamics and Control Conference. Berkeley: PMLR, 2019: 4213-4220.

[124] Li S, Wu Y , Cui X, et al. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient[C]. Proceedings of the 36th AAAI Conference on Articial Intelligence. Hawaii: AAAI Press, 2019: 4213-4220.

[125] Fudenberg D, Tirole J. Game theory[M]. Cambridge: MIT Press, 1991.

[126] Hu J, Wellman M P. Multiagent reinforcement learning: Theoretical framework and an algorithm[C]. Proceedings of the 15th International Conference on Machine Learning. Madison: ACM Press, 2017: 449-458.

[127] Hu J, Wellman M P. Experimental results on $Q$-learning for general-sum stochastic games[C]. Procedings of the 17th International Conference on Machine Learning. Stanford: ACM Press, 2000: 407-414.

[128] Hu J, Wellman M P. Nash $Q$-learning for general-sum stochastic games[J]. Journal of Machine Learning Research, 2003, 4(4): 1039-1069.

[129] Greenwald A, Hall K, Serrano R. Correlated $Q$-learning[C]. Proceedings of the 20th International Conference on Machine Learning. Washington: ACM Press, 2003: 1-31.

[130] Sunehag P, Lever G, Audrunas G, 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 Multi-Agent Systems. Stockholm: Springer, 2018: 2085-2087.

[131] Rashid T, Samvelyan M, Schroeder C, et al. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ACM Press, 2018: 4295-4304.

[132] 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 35th International Conference on Machine Learning. Long Beach: ACM Press, 2019: 5887-5896.

[133] Yang Y D, Hao J Y, Liao B, et al. Qatten: A general framework for cooperative multiagent reinforcement learning[J/OL]. 2020, arXiv: 2002.03939.

[134] Wang J H, Ren Z Z, Liu T, et al. QPLEX: Duplex dueling multi-agent $Q$-learning[J/OL]. 2020, arXiv: 2008.01062.

[135] 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. Long Beach, 2017: 6382-6393.

[136] Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients[J/OL]. 2018, arXiv: 1705.08926.

[137] Iqbal S, Sha F. Actor-attention-critic for multi-agent reinforcement learning[C]. Proceedings of the 35th International Conference on Machine Learning. Long Beach: ACM Press, 2019: 2961-2970.

[138] Zhang Y, Yang Q Y, An D, et al. Coordination between individual agents in multi-agent reinforcement learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(13): 11387-11394.

[139] Zhang T, Li Y, Wang C, et al. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning[C]. Proceedings of the 38th International Conference on Machine Learning. Online: ACM Press, 2021: 12491-12500.

[140] Su J, Adams S, Beling P. Value-decomposition multi-agent actor-critics[C]. Proceedings of the 38th AAAI Conference on Articial Intelligence. Online: AAAI Press, 2021, 35(13): 11352-11360.

[141] Wang T H, Dong H, Lesser V, et al. ROMA: Multi-agent reinforcement learning with emergent roles[J/OL]. 2020, arXiv: 2003.08039.

[142] Wang T H, Gupta T, Mahajan A, et al. RODE: Learning roles to decompose multi-agent tasks[J/OL]. 2020, arXiv: 2010.01523.

[143] Das A, Gervet T, Romoff J, et al. Tarmac: Targeted multi-agent communication[C]. Proceedings of the 35th International Conference on Machine Learning. Long Beach: ACM Press, 2019: 1538-1546.

[144] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J/OL]. 2014, arXiv: 1412.3555.

[145] Peng P, Wen Y, Yang Y D, et al. Multiagent

bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games[J/OL]. 2017, arXiv: 1703.10069.

[146] Jiang J C, Lu Z Q. Learning attentional communication for multi-agent cooperation[C]. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, 2018: 7265-7275.

[147] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[148] Singh A, Jain T, Sukhbaatar S. Individualized controlled continuous communication model for multi-agent cooperative and competitive tasks[J/OL]. 2019, arXiv: 1812.09755.

[149] Ding Z L, Huang T J, Lu Z Q. Learning individually inferred communication for multi-agent cooperation[C]. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver, 2020: 22069-22079.

[150] Yan Z, Zavlanos M M. Distributed off-policy actor-critic reinforcement learning with policy consensus[C]. Proceedings of the 58th IEEE Conference on Decision and Control. Nice: IEEE, 2019: 4674-4679.

[151] Zhang K, Yang Z, Liu H, et al. Fully decentralized multi-agent reinforcement learning with networked agents[C]. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ACM Press, 2018: 5872-5881.

[152] Lin Y X, Zhang K Q, Yang Z R, et al. A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning[C]. IEEE 58th Conference on Decision and Control. Nice, 2020: 5562-5567.

[153] Dai P, Yu W, Wang H, et al. Distributed actor-critic algorithms for multiagent reinforcement learning over directed graphs[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3139138.

[154] Yang Y, Luo R, Li M, et al. Mean eld multi-agent reinforcement learning[C]. Proceedings of the 35th International Conference on Machine Learning. Stockholm: ACM Press, 2018: 5571-5580.

[155] Anahtarci B, Kariksiz C D, Saldi N. $Q$-learning in regularized mean-field games[J/OL]. 2020, arXiv: 2003.12151.

[156] Elie R, Perolat J, Lauriere M, et al. On the convergence of model free learning in mean eld games[C]. Proceedings of the 34th AAAI Conference on Articial Intelligence. New York: AAAI Press, 2020: 7143-7150.

[157] Ma J M, Wu F. Feudal multi-agent deep reinforcement learning for traffic signal control[C]. Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. New York: ACM, 2020: 816-824.

[158] Wang T H, Wang J H, Zheng C Y, et al. Learning nearly decomposable value functions via communication minimization[J/OL]. 2019, arXiv: 1910.05366.

[159] Liang Y H, Wu H J, Wang H T. ASM-PPO: Asynchronous and scalable multi-agent PPO for cooperative charging[C]. Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. Virtual Event, 2022: 798-806.

[160] Ruan J Q, Du Y L, Xiong X T, et al. GCS: Graph-based coordination strategy for multi-agent reinforcement learning[C]. Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. Virtual Event, 2022: 1128-1136.

[161] Cohen S, Agmon N. Optimizing multi-agent coordination via hierarchical graph probabilistic recursive reasoning[C]. Proceedings of the 21th International Conference on Autonomous Agents and Multi-Agent Systems. Auckland: Springer, 2022: 290-299.

[162] Wen G H, Fu J J, Dai P C, et al. DTDE: A new cooperative multi-agent reinforcement learning framework[J]. Innovation: Cambridge Mass, 2021, 2(4): 100162.

[163] Li F Y, Qin J H, Zheng W X. Distributed $Q$-learning-based online optimization algorithm for unit commitment and dispatch in smart grid[J]. IEEE Transactions on Cybernetics, 2020, 50(9): 4146-4156.

[164] Li D, Yu L, Li N, et al. Virtual-action-based coordinated reinforcement learning for distributed economic dispatch[J]. IEEE Transactions on Power Systems, 2021, 36(6): 5143-5152.

[165] Ding L, Lin Z, Yan G. Multi-agent deep reinforcement learning algorithm for distributed economic dispatch in smart grid[C]. Proceedings of the 46th Annual Conference of the IEEE Industrial Electronics Society. Singapore: IEEE, 2020: 3529-3534.

[166] Dai P, Yu W, Wen G, et al. Distributed reinforcement learning algorithm for dynamic economic dispatch with unknown generation cost functions[J]. IEEE Transactions on Industrial Informatics, 2019, 16(4): 2258-2267.

[167] Hu C, Wen G, Wang S, Fu J, et al. Distributed multi-agent reinforcement learning with action networks for dynamic economic dispatch[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2023.3234049.

[168] Na S, Niu H, Lennox B, et al. Bio-inspired collision avoidance in swarm systems via deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2022, 71(3): 2511-2526.

[169] Chen C, Liu Y, Kreiss S, et al. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning[C]. International Conference on Robotics and Automation. Piscataway: IEEE, 2019: 6015-6022.

[170] Everett M, Chen Y F, How J P. Motion planning among dynamic, decision-making agents with deep reinforcement learning[C]. IEEE/RSJ International

Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2018: 3052-3059.

[171] Zhao W, Liu H, Lewis F L. Robust formation control for cooperative underactuated quadrotors via reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(10): 4577-4587.

[172] Wang N, Gao Y, Zhang X. Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(12): 5456-5467.

[173] Prokhorov D V, Wunsch D C. Adaptive critic designs[J]. IEEE Transactions on Neural Networks, 1997, 8(5): 997-1007.

[174] Padhi R, Unnikrishnan N, Wang X H, et al. A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems[J]. Neural Networks, 2006, 19(10): 1648-1660.

[175] Liu D R, Wei Q L, Wang D, et al. Adaptive dynamic programming with applications in optimal control[M]. Berlin: Springer International Publishing, 2017: 37-90.

[176] Ni Z, He H B, Zhong X N, et al. Model-free dual heuristic dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(8): 1834-1839.

[177] Venayagamoorthy G K, Harley R G, Wunsch D C. Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator[J]. IEEE Transactions on Neural Networks, 2002, 13(3): 764-773.

[178] Zhao D B, Xia Z P, Wang D. Model-free optimal control for affine nonlinear systems with convergence analysis[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(4): 1461-1468.

[179] Leake R J, Liu R W. Construction of suboptimal control sequences[J]. SIAM Journal on Control, 1967, 5(1): 54-63.

[180] Shaiju A J, Petersen I R. Formulas for discrete time LQR, LQG, LEQG and minimax LQG optimal control problems[J]. IFAC Proceedings Volumes, 2008, 41(2): 8773-8778.

[181] Lewis F L, Syrmos V L. Optimal control[M]. The 2nd edition. New York: Wiley, 1995: 461-517.

[182] Lewis F L, Vrabie D, Vamvoudakis K G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers[J]. IEEE Control Systems Magazine, 2012, 32(6): 76-105.

[183] Hewer G. An iterative technique for the computation of the steady state gains for the discrete optimal regulator[J]. IEEE Transactions on Automatic Control, 1971, 16(4): 382-384.

[184] Lancaster P, Rodman L. Algebraic Riccati aligns[M]. Oxford: Clarendon Press, 1995: 147-346.

[185] Vamvoudakis K G, Vrabie D, Lewis F L. Online adaptive algorithm for optimal control with integral reinforcement learning[J]. International Journal of Robust and Nonlinear Control, 2014, 24(17): 2686-2710.

[186] Vrabie D, Vamvoudakis K, Lewis F. Adaptive optimal controllers based on generalized policy iteration in a continuous-time framework[C]. The 17th Mediterranean Conference on Control and Automation. Thessaloniki, 2009: 1402-1409.

[187] Vrabie D, Pastravanu O, Abu-Khalaf M, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration[J]. Automatica, 2009, 45(2): 477-484.

[188] Kleinman D. On an iterative technique for Riccati align computations[J]. IEEE Transactions on Automatic Control, 1968, 13(1): 114-115.

[189] Lee J Y, Park J B, Choi Y H. On integral value iteration for continuous-time linear systems[C]. 2013 American Control Conference. Washington, 2013: 4215-4220.

[190] Vamvoudakis K G, Lewis F L. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem[J]. Automatica, 2010, 46(5): 878-888.

[191] Kiumarsi B, Lewis F L, Jiang Z P. $H_1$ control of linear discrete-time systems: Off-policy reinforcement learning[J]. Automatica, 2017, 78: 144-152.

[192] Yang Y L, Kiumarsi B, Modares H, et al. Model-free $\lambda$-policy iteration for discrete-time linear quadratic regulation[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3098985.

[193] Jiang H Y, Zhou B. Bias-policy iteration based adaptive dynamic programming for unknown continuous-time linear systems[J]. Automatica, 2022, 136: 110058.

[194] Chen C, Lewis F L, Li B. Homotopic policy iteration-based learning design for unknown linear continuous-time systems[J]. Automatica, 2022, 138: 110153.

[195] Liu D R, Xue S, Zhao B, et al. Adaptive dynamic programming for control: A survey and recent advances[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(1): 142-160.

[196] Zhang H G, Luo Y H, Liu D R. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints[J]. IEEE Transactions on Neural Networks, 2009, 20(9): 1490-1503.

[197] Greene M L, Deptula P, Nivison S, et al. Sparse learning-based approximate dynamic programming with barrier constraints[J]. IEEE Control Systems Letters, 2020, 4(3): 743-748.

[198] Cohen M H, Belta C. Approximate optimal control for safety-critical systems with control barrier functions[C]. The 59th IEEE Conference on Decision and Control. Jeju, 2021: 2062-2067.

[199] Yang Y L, Yin Y X, He W, et al. Safety-aware reinforcement learning framework with an actor-critic-barrier structure[C]. 2019 American

Control Conference. Philadelphia, 2019: 2352-2358.

[200] Xue S, Luo B, Liu D R, et al. Event-triggered integral reinforcement learning for nonzero-sum games with asymmetric input saturation[J]. Neural Networks, 2022, 152: 212-223.

[201] Wang A J, Liao X F, Dong T. Event-driven optimal control for uncertain nonlinear systems with external disturbance via adaptive dynamic programming[J]. Neurocomputing, 2018, 281: 188-195.

[202] Liu R R, Hao F, Yu H. Optimal SINR-based DoS attack scheduling for remote state estimation via adaptive dynamic programming approach[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(12): 7622-7632.

[203] Moghadam R, Jagannathan S. Approximate optimal adaptive control of partially unknown linear continuous-time systems with state delay[C]. IEEE 58th Conference on Decision and Control. Nice, 2020: 1985-1990.

[204] Ali Asad Rizvi S, Wei Y S, Lin Z L. Model-free optimal stabilization of unknown time delay systems using adaptive dynamic programming[C]. IEEE 58th Conference on Decision and Control. Nice, 2020: 6536-6541.

[205] Gao W N, Deng C, Jiang Y, et al. Resilient reinforcement learning and robust output regulation under denial-of-service attacks[J]. Automatica, 2022, 142: 110366.

[206] Ali Asad Rizvi S, Lin Z L. Output feedback optimal tracking control using reinforcement $Q$-learning[C]. Annual American Control Conference. Milwaukee, 2018: 3423-3428.

[207] Chen C, Modares H, Xie K, et al. Reinforcement learning-based adaptive optimal exponential tracking control of linear systems with unknown dynamics[J]. IEEE Transactions on Automatic Control, 2019, 64(11): 4423-4438.

[208] Odekunle A, Gao W N, Davari M, et al. Reinforcement learning and non-zero-sum game output regulation for multi-player linear uncertain systems[J]. Automatica, 2020, 112: 108672.

[209] Tang Y F, He H B, Wen J Y, et al. Power system stability control for a wind farm based on adaptive dynamic programming[J]. IEEE Transactions on Smart Grid, 2015, 6(1): 166-177.

[210] Zhu Y H, Zhao D B, Li X J, et al. Control-limited adaptive dynamic programming for multi-battery energy storage systems[J]. IEEE Transactions on Smart Grid, 2019, 10(4): 4235-4244.

[211] Liu D R, Xu Y C, Wei Q L, et al. Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 5(1): 36-46.

[212] Wei Q L, Liu D R, Liu Y, et al. Optimal constrained self-learning battery sequential management in microgrid via adaptive dynamic programming[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 4(2): 168-176.

[213] Mu C X, Ni Z, Sun C Y, et al. Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(3): 584-598.

[214] Sun W C, Wang X, Zhang C Z. A model-free control strategy for vehicle lateral stability with adaptive dynamic programming[J]. IEEE Transactions on Industrial Electronics, 2020, 67(12): 10693-10701.

[215] Li K, Wang N, Weng Y. Online reinforcement learning-based adaptive tracking control of an unknown unmanned surface vehicle with input saturations[C]. Proceedings of 2020 International Conference on System Science and Engineering. Kagawa: IEEE, 2020: 1-6.

[216] Wang N, Gao Y, Zhao H, et al. Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(7): 3034-3045.

[217] Wang N, Gao Y, Yang C, et al. Reinforcement learning-based finite-time tracking control of an unknown unmanned surface vehicle with input constraints[J]. Neurocomputing, 2022, 484: 26-37.

[218] Kong L H, He W, Yang C G, et al. Robust neurooptimal control for a robot via adaptive dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(6): 2584-2594.

[219] Wen Y, Si J, Gao X, et al. A new powered lower limb prosthesis control framework based on adaptive dynamic programming[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(9): 2215-2220.

[220] Sun J Y, Dai J, Zhang H G, et al. Neural-network-based immune optimization regulation using adaptive dynamic programming[J]. IEEE Transactions on Cybernetics, DOI: 10.1109/TCYB.2022.3179302.

[221] 赵振根, 程磊. 基于增量式 $Q$ 学习的固定翼无人机跟踪控制性能优化[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2022.0708.
(Zhao Z G, Cheng L. Performance optimization for tracking control of xed-wing UAV with incremental $Q$-learning [J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.0708.)

[222] Yu J, Wang L. Group consensus of multi-agent systems with undirected communication graphs[C]. Proceedings of the 2009 7th Asian Control Conference. Piscataway: IEEE, 2009: 105-110.

[223] Li X, Ji L, Yang S, et al. Optimal group consensus control for multi-agent systems in coopetition networks via dynamic event-triggered methods[C]. Proceedings of 5th Chinese Conference on Swarm Intelligence and Cooperative Control. Singapore: Springer Nature Singapore, 2022: 134-145.

[224] Peng Z N, Hu J P, Shi K B, et al. A novel optimal bipartite consensus control scheme for unknown multi-agent

systems via model-free reinforcement learning[J]. Applied Mathematics and Computation, 2020, 369: 124821.

[225] Li J, Ji L H, Zhang C J, et al. Optimal couple-group tracking control for the heterogeneous multi-agent systems with cooperative-competitive interactions via reinforcement learning method[J]. Information Sciences, 2022, 610: 401-424.

[226] Zhang H P, Park J H, Yue D, et al. Finite-horizon optimal consensus control for unknown multiagent state-delay systems[J]. IEEE Transactions on Cybernetics, 2020, 50(2): 402-413.

[227] Liu C, Liu L. Finite-horizon robust event-triggered control for nonlinear multi-agent systems with state delay[J]. Neural Processing Letters, 2022: 1-25.

[228] Cheng L, Hou Z G, Tan M. A mean square consensus protocol for linear multi-agent systems with communication noises and fixed topologies[J]. IEEE Transactions on Automatic Control, 2014, 59(1): 261-267.

[229] Li T, Zhang J F. Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions[J]. Automatica, 2009, 45(8): 1929-1936.

[230] Cai H, Lewis F L, Hu G Q, et al. The adaptive distributed observer approach to the cooperative output regulation of linear multi-agent systems[J]. Automatica, 2017, 75: 299-305.

[231] Li Q, Xia L N, Song R Z, et al. Leader-follower bipartite output synchronization on signed digraphs under adversarial factors via data-based reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(10): 4185-4195.

[232] Liu X, Ge S S. Optimized control for human-multi-robot collaboration via multi-agent adaptive dynamic programming[J]. IFAC-Papers Online, 2020, 53(2): 9207-9212.

[233] Tan L N. Omnidirectional-vision-based distributed optimal tracking control for mobile multirobot systems with kinematic and dynamic disturbance rejection[J]. IEEE Transactions on Industrial Electronics, 2018, 65(7): 5693-5703.

[234] Xue W Q, Kolaric P, Fan J L, et al. Inverse reinforcement learning in tracking control based on inverse optimal control[J]. IEEE Transactions on Cybernetics, 2022, 52(10): 10570-10581.

[235] Lian B S, Xue W Q, Lewis F L, et al. Online inverse reinforcement learning for nonlinear systems with adversarial attacks[J]. International Journal of Robust and Nonlinear Control, 2021, 31(14): 6646-6667.

[236] Lian B S, Xue W Q, Lewis F L, et al. Inverse reinforcement learning for multi-player noncooperative apprentice games[J]. Automatica, 2022, 145: 110524.

[237] Lian B S, Xue W Q, Lewis F L, et al. Inverse reinforcement learning for adversarial apprentice games[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3114612.

[238] Lian B S, Donge V S, Lewis F L, et al. Data-driven inverse reinforcement learning control for linear multiplayer games[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2022.3186229.

[239] Xue W Q, Lian B S, Fan J L, et al. Inverse reinforcement $Q$-learning through expert imitation for discrete-time systems[J]. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3106635.

[240] Self R, Abudia M, Mahmud S M N, et al. Model-based inverse reinforcement learning for deterministic systems[J]. Automatica, 2022, 140: 110242.

[241] Donge V S, Lian B S, Lewis F L, et al. Multi-agent graphical games with inverse reinforcement learning[J]. IEEE Transactions on Control of Network Systems, DOI: 10.1109/TCNS.2022.3210856.

[242] Neumeyer C, Oliehoek F A, Gavrila D M. General-sum multi-agent continuous inverse optimal control[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 3429-3436.

[243] Belfo J P, Aguiar A P, Lemos J M. Distributed inverse optimal control for discrete-time nonlinear multi-agent systems[J]. IEEE Control Systems Letters, 2021, 5(6): 2096-2101.

[244] Wen G H, Yu W W, Yu X H, et al. Complex cyber-physical networks: From cybersecurity to security control[J]. Journal of Systems Science and Complexity, 2017, 30(1): 46-67.

## 作者简介

温广辉 (1983−), 男, 教授, 博士生导师, 从事网络群体智能理论与技术、分布式控制理论与控制工程等研究, E-mail: ghwen@seu.edu.cn;

杨涛 (1981−), 男, 教授, 博士生导师, 从事复杂工业过程协同控制与优化等研究, E-mail: yangtao@mail.neu.edu.cn;

周佳玲 (1991−), 女, 副教授, 博士生导师, 从事群智协同控制与优化、集群博弈等研究, E-mail: jlzhou@bit.edu.cn;

付俊杰 (1989−), 男, 副研究员, 博士生导师, 从事受限多智能体系统分布式协同控制、多智能体安全强化学习等研究, E-mail: fujunjie@seu.edu.cn;

徐磊 (1994−), 男, 博士生, 从事分布式协同控制和优化、网络控制系统等研究, E-mail: 2010345@stu.neu.edu.cn.

(责任编辑：郑晓蕾)