软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2024,35(2):739−757 [doi: 10.13328/j.cnki.jos.006832]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 合作-竞争混合型多智能体系统的虚拟遗憾优势自博弈方法*

张明悦[1,2,3], 金 芝[2,3], 刘 坤[2,3]

[1](西南大学 计算机信息科学学院 & 软件学院, 重庆 400715)

[2](北京大学 计算机学院, 北京 100871)

[3](高可信软件技术教育部重点实验室 (北京大学), 北京 100871)

通信作者: 金芝, E-mail: zhijin@pku.edu.cn

**摘 要**: 合作-竞争混合型多智能体系统由受控的目标智能体和不受控的外部智能体组成. 目标智能体之间互相合作, 同外部智能体展开竞争, 应对环境和外部智能体的动态变化, 最终完成指定的任务. 针对如何训练目标智能体使他们获得完成任务的最优策略的问题, 现有工作从两个方面展开: (1) 仅关注目标智能体间的合作, 将外部智能体视为环境的一部分, 利用多智能体强化学习来训练目标智能体. 这种方法难以应对外部智能体策略未知或者动态改变的情况; (2) 仅关注目标智能体和外部智能体间的竞争, 将竞争建模为双人博弈, 采用自博弈的方法训练目标智能体. 这种方法主要针对单个目标智能体和单个外部智能体的情况, 难以扩展到由多个目标智能体和多个外部智能体组成的系统中. 结合这两类研究, 提出一种基于虚拟遗憾优势的自博弈方法. 具体地, 首先以虚拟遗憾最小化和虚拟多智能体策略梯度为基础, 设计虚拟遗憾优势策略梯度方法, 使目标智能体能更准确地更新策略; 然后, 引入模仿学习, 以外部智能体的历史决策轨迹作为示教数据, 模仿外部智能体的策略, 显式地建模外部智能体的行为, 来应对自博弈过程中外部智能体策略的动态变化; 最后, 以虚拟遗憾优势策略梯度和外部智能体行为建模为基础, 设计一种自博弈训练方法, 该方法能够在外部智能体策略未知或者动态变化的情况下, 为多个目标智能体训练出最优的联合策略. 以协同电磁对抗为研究案例, 设计具有合作-竞争混合特征的 3 个典型任务. 实验结果表明, 同其他方法相比, 所提方法在自博弈效果方面有至少 78% 的提升.

**关键词**: 多智能体强化学习; 虚拟遗憾最小化; 自博弈; 动态决策

**中图法分类号**: TP18

中文引用格式: 张明悦, 金芝, 刘坤. 合作-竞争混合型多智能体系统的虚拟遗憾优势自博弈方法. 软件学报, 2024, 35(2): 739–757. http://www.jos.org.cn/1000-9825/6832.htm

英文引用格式: Zhang MY, Jin Z, Liu K. Counterfactual Regret Advantage-based Self-play Approach for Mixed Cooperative-competitive Multi-agent Systems. Ruan Jian Xue Bao/Journal of Software, 2024, 35(2): 739–757 (in Chinese). http://www.jos.org.cn/1000-9825/6832.htm

## Counterfactual Regret Advantage-based Self-play Approach for Mixed Cooperative-competitive Multi-agent Systems

ZHANG Ming-Yue[1,2,3], JIN Zhi[2,3], LIU Kun[2,3]

[1](College of Computer and Information Science & School of Software, Southwest University, Chongqing 400715, China)

[2](School of Computer Science, Peking University, Beijing 100871, China)

[3](Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

**Abstract**: The mixed cooperative-competitive multi-agent system consists of controlled target agents and uncontrolled external agents. The target agents cooperate with each other and compete with external agents, so as to deal with the dynamic changes in the environment and

* 基金项目: 国家自然科学基金 (62192731)
收稿时间: 2022-06-19; 修改时间: 2022-09-01; 采用时间: 2022-11-14; jos 在线出版时间: 2023-07-19
CNKI 网络首发时间: 2023-07-20

the external agents and complete tasks. In order to train the target agents and make them learn the optimal policy for completing the tasks, the existing work proposes two kinds of solutions: (1) focusing on the cooperation between target agents, viewing the external agents as a part of the environment, and leveraging the multi-agent-reinforcement learning to train the target agents; but these approaches cannot handle the uncertainty of or dynamic changes in the external agents' policy; (2) focusing on the competition between target agents and external agents, modeling the competition as two-player games, and using a self-play approach to train the target agents; these approaches are only suitable for cases where there is one target agent and external agent, and they are difficult to be extended to a system consisting of multiple target agents and external agents. This study combines the two kinds of solutions and proposes a counterfactual regret advantage-based self-play approach. Specifically, first, based on the counterfactual regret minimization and counterfactual multi-agent policy gradient, the study designs a counterfactual regret advantage-based policy gradient approach for making the target agent update the policy more accurately. Second, in order to deal with the dynamic changes in the external agents' policy during the self-play process, the study leverages imitation learning, which takes the external agents' historical decision-making trajectories as training data and imitates the external agents' policy, so as to explicitly model the external agents' behaviors. Third, based on the counterfactual regret advantage-based policy gradient and the modeling of external agents' behaviors, this study designs a self-play training approach. This approach can obtain the optimal joint policy for training multiple target agents when the external agents' policy is uncertain or dynamically changing. The study also conducts a set of experiments on the cooperative electromagnetic countermeasure, including three typical mixed cooperative-competitive tasks. The experimental results demonstrate that compared with other approaches, the proposed approach has an improvement of at least 78% in the self-game effect.

**Key words**: multi-agent reinforcement learning; counterfactual regret minimization; self-play; dynamic decision-making

合作-竞争混合型多智能体系统由受控的目标智能体和不受控的外部智能体组成, 在诸如智能交通控制、协同电磁对抗、多人即时战略游戏中有着重要的应用[1−3]. 在这类系统中, 多个目标智能体在策略的控制下相互合作, 同外部智能体竞争, 并调整自身行为以应对外部智能体或环境的动态变化, 最终完成指定的任务. 任务完成的情况受到运行环境、目标智能体和外部智能体的共同影响; 同时, 由于外部智能体不受系统控制, 一方面很难为外部智能体预先设计出合适的联合策略, 另一方面它们的策略可能不断发生改变, 偏离预先的设定[1]. 这种外部智能体的不确定性和动态性给目标智能体策略的构造带来了很大的挑战.

为了学到目标智能体完成任务的最优策略, 现有工作从合作型多智能体强化学习和竞争型自博弈两个方面展开. 合作型多智能体强化学习以目标智能体间的合作为核心关注点, 将外部智能体考虑为环境的一部分, 并假定环境是稳态的 (即环境的变化规律不会发生改变)[1,2,4], 将目标智能体和外部智能体之间的竞争问题转换为目标智能体之间的合作问题, 最终利用多智能体强化学习算法来训练出目标智能体的最优联合策略. 典型的工作包括 VDN[3]、QMIX[4]、COMA[2]等. QMIX 在各种即时战略类游戏中表现最优, 它将全体目标智能体的联合价值函数通过 MIX 神经网络分解为单个目标智能体的个体价值函数, 每个目标智能体根据个体价值函数采取行动; 这种基于价值函数分解的方法能以较低的训练成本学习到高质量的合作策略, 适合于完全合作的场景. 但是, 在合作-竞争混合型多智能体系统中, 采用这类方法需要首先构造外部智能体的策略, 再训练目标智能体. 外部智能体的策略水平很大程度上决定了目标智能体的策略水平, 当前者采用随机策略或者简单规则时, 后者也只能学习到低水平策略. 如果目标智能体和外部智能体均采用强化学习算法, 双方都需要把对方当成稳态环境的一部分, 但事实上对方的策略会随学习过程不断改变. 这导致在双方视角下环境都是非稳态的, 而非稳态的环境使学习过程丧失收敛性的保证, 并可能使双方都陷入到低水平的策略中.

竞争型自博弈的研究主要关注目标智能体和外部智能体的竞争, 将二者的竞争建模为双人博弈, 利用二者在博弈中的交互数据, 采用自博弈的方法训练出目标智能体的最优策略. 典型的工作包括 FSP[5]、NFSP[6]、PSRP[7]. 这类方法的训练流程为: 先固定外部智能体的策略, 目标智能体利用强化学习或者其他优化方法, 得到针对当前外部智能体策略的最优反应策略; 然后固定目标智能体的策略, 外部智能体利用同样的方法得到针对当前目标智能体策略的最优反应策略; 如此交替, 最终使两者的策略收敛到纳什均衡. 特别地, FSP 在构建外部智能体的最优反应策略时, 引入了监督学习, 利用外部智能体在训练过程中的历史数据学习出一个外部智能体的历史平均最优反应策略. PSRP 引入了元博弈, 将原有博弈中的策略建模为元博弈中的动作, 通过求解元博弈中的最优动作来得到目标智能体的最优策略. 但是, 这类方法目前主要关注于双人博弈, 难以扩展到由多个目标智能体和多个外部智能体组成的系统.

为了解决合作-竞争混合型多智能体系统中目标智能体的联合策略的生成问题, 本文将合作型多智能体强化

学习方法和竞争型自博弈方法相结合, 提出了一种基于虚拟遗憾优势的自博弈方法. 首先, 以虚拟遗憾最小化 (counterfactual regret minimization, CFR)[8]和虚拟多智能体策略梯度 (counterfactual multi-agent policy gradient, COMA)[2]为基础, 设计了虚拟遗憾优势 (counterfactual regret advantage, CRA); CRA 根据全部智能体的联合策略和目标智能体的期望奖励, 计算出每个目标智能体的虚拟遗憾, 评估每个目标智能体当前策略的优劣, 为目标智能体的策略更新提供更为准确的指导信号. 其次, 不再将外部智能体考虑为环境的一部分, 而是将其建模为一种受策略控制、根据观测采取动作、且会自我改变策略的个体; 为建模这样的外部智能体, 本文引入了模仿学习 (imitation learning), 具体地, 本文采用模仿学习中行为克隆技术, 以外部智能体历史上的观测和动作序列作为示教数据, 学习一个神经网络来拟合外部智能体的策略, 从而建立起外部智能体行为的预测模型. 最后, 结合 CRA 和外部智能体行为模型, 设计了一种模仿虚拟遗憾优势算法 (imitation counterfactual regret advantage, ICRA), 进而以 ICRA 为基础, 设计了一种自博弈方法. 在自博弈训练阶段, 本文假定算法可以控制所有的智能体, 且能够观察到所需要的环境数据, 目标智能体和外部智能体均采用 ICRA 算法进行学习, 双方交替地进行策略更新, 当双方策略不再提升时, 自博弈训练终止. 为了验证方法的有效性, 本文以协同电磁对抗为研究案例, 分别设计了航空兵突防、空战对抗、空地协同作战等 3 个想定. 实验结果表明, 本文所提出的基于虚拟遗憾优势的自博弈方法能够在外部智能体策略未知的情况下, 学习到高水平的目标智能体的联合策略; 同其他方法相比, 本文的方法在自博弈效果方面有至少 78% 的提升, 而在外部智能体策略已知的情况下, 本文的方法也能够获得和目前最先进的合作型多智能体强化学习方法相当的效果.

本文的核心贡献如下.

(1) 提出了一种虚拟遗憾优势, 它以虚拟遗憾最小化为理论基础, 是虚拟多智能体策略梯度的一种泛化形式; 同虚拟多智能体策略梯度以及其他策略梯度相比, 它能够更加精确地评估每个目标智能体所采取行动的优劣程度, 进而更加准确地指导每个目标智能体进行策略更新.

(2) 提出了一种基于模仿学习的智能体建模方法, 该方法在外部智能体的历史决策轨迹上, 以行为克隆的方式模仿外部智能体的策略, 显式建模它们的行为, 以解决自博弈训练过程中外部智能体策略动态变化的问题.

(3) 结合虚拟遗憾优势和外部智能体行为模型, 设计了一种模仿虚拟遗憾优势算法, 以及基于该算法的自博弈方法, 该自博弈方法没有专家规则引导的情况下, 能够学习到高质量的目标智能体联合策略.

本文第 1 节介绍相关工作. 第 2 节介绍本文相关的基础知识. 第 3 节介绍本文所提出 ICRA 算法, 以及基于 ICRA 的自博弈训练方法. 第 4 节进行实验并分析实验结果. 第 5 节给出结论, 并简述未来的研究方向.

## 1 相关工作

近年来, 多智能体强化学习发展迅速, 研究者针对各种不同场景相继提出了很多算法, 被广泛应用于电子游戏、交通控制、机器人编队等领域. 和本文研究高度相关的多智能体强化学习算法包括: MADDPG[9]、VDN[3]、QMIX[4]、COMA[2]等. 这些算法关注在完全合作的情况下, 如何控制系统中的智能体获得最大的累计奖励. MADDPG 将 DDPG 算法[10]扩展到了多智能体系统中, 首次提出面向竞争-合作混合的多智能体系统的“集中式训练-分布式执行”的框架, 被后续算法所沿用[1−4].

基于集中式训练-分布式执行的算法大致可分为两类: 值函数分解方法和置信度分配方法. 值函数分解方法考虑的问题是: 在集中式训练过程中, 智能体的联合策略会依赖于一个联合 Q 神经网络, 该神经网络评估了给定状态下, 某个联合动作的价值; 然而, 智能体群体的联合动作空间随着智能体的数量呈指数增加, 这导致联合 Q 神经网络的输入维数巨大, 直接训练这个 Q 网络会比较困难. 值函数分解方法将联合 Q 神经网络分解为智能体的个体 Q 神经网络. 目前典型的值函数分解类算法有 VDN、QMIX、QTRAN[11]等, 这类算法需要满足: 取得联合 Q 网络最大值的联合动作, 也能使得智能体的个体 Q 网络的值最大, 即个体全局最大条件 (individual-global-max, IGM), 而不同算法的主要差别在于值函数分解的方式不同. VDN 将联合 Q 神经网络线性分解为个体 Q 神经网络 (即, $Q(s, \boldsymbol{a})=\Sigma\alpha_i Q_i(s, a_i)$, 其中 $Q$ 为联合 Q 网络, $Q_i$ 为智能体 $i$ 的个体 Q 网络, $\alpha_i$ 为权值, $\boldsymbol{a}=[a_1,\ldots, a_i,\ldots, a_n]$ 是联合动作, $a_i$ 是智能体 $i$ 的动作, $s$ 是状态); VDN 通过训练个体 Q 网络, 来得到群体的联合 Q 网络, 从而缩小了联合 Q 网络在训练时所要应对的状态-动作组合空间. QMIX 在 VDN 的基础上进行了扩展, 将线性分解的方法改进为基

于 MIX 网络的分解方法, 可以应对个体 Q 网络非线性组合为联合 Q 网络的情况. QTRAN[11]则进一步在 VDN 和 QMIX 的基础上, 在分解过程中增加了一个偏置量来修正原本的联合 Q 网络和分解后的联合 Q 网络之间的误差, 从而提升分解的准确性与通用性. 置信度分配方法关注的问题是: 如何根据全局奖励信号, 指导每个智能体独立地更新自身的策略. 比如, COMA[2]设计一个集中式的价值网络和对应于每个智能体的策略网络; 集中式的价值网络计算反事实基线 (counterfactual baseline), 该反事实基线用于评估智能体的个体策略在当前情况下的期望收益; 每个策略网络根据反事实基线来更新网络参数. LIIR[12]为每个智能体设计一个可学习的个体内在奖励函数, 该奖励函数能够将全局奖励信号分解给每个个体, 用以评估该个体当前策略的好坏, 并指导其更新策略.

另一类相关工作是自博弈方法. 这类方法关注在完全竞争的情况下, 两个智能体如何通过与对手的交互, 学习到纳什均衡策略. 较为典型的自博弈方法有, PHC[13]关注双人零和博弈, 引入强化学习的 Q 学习; 博弈的双方均按照 Q 学习独立并同时地在环境中采取动作, 根据环境的反馈来更新自身的 Q 函数. WoLF-PHC[13]以 PHC 为基础, 引入“变学习率”的方法以提升学习性能: 在每一轮学习过程中, 博弈的每个智能体计算当前奖励和基线奖励的差值, 若差值大于零, 则用小的学习率更新策略, 反之则用大的学习率更新策略. FSP[5]和 NFSP[5]将深度学习和深度强化学习引入到了自博弈中, 过程如下: 在第 $n$ 轮训练时, 固定外部智能体策略 $\pi_e$, 目标智能体利用强化学习训练出最优反应策略 $\pi_t$, 将 $\pi_t$ 加入到策略库中; 利用监督学习, 根据策略库中的策略训练出模仿策略 $\pi_i$, 将 $\pi_e$ 设置为 $\pi_i$; 然后再进行第 $n$+1 轮训练, 如此交替进行, 最终使得双方的策略收敛到纳什均衡. PSRP[7]引入元博弈, 即分别将原来博弈中的策略、和两个策略交互后的平均奖励值建模为元博弈中的动作和奖励函数, 进而将双方的策略选择建模为元博弈中的动作选择, 由此设计出矩阵博弈, 并通过求解矩阵博弈的均衡解来得到目标智能体的最优策略.

在表 1 中, 我们对比了和本文所提出的方法紧密相关的工作. 特别地, 我们从算法设计和应用场景两个方面进行了比较. 从算法设计角度来看, 本文的工作扩展了 MICRA, 设计了一种自博弈方法以及改进了策略梯度的计算方式, 从而提升了 MICRA 的策略优化的能力, 而 MICRA 则是在 COMA 的基础上增加了模仿学习的模块. 从应用场景的角度来看, 本文的方法适合应对智能体规模适中、目标智能体和外部智能体策略均发生改变, 同时系统中的智能体还呈现出组内合作、组间竞争特点的场景, 同 COMA 和 QMIX 相比, 本文的方法更适合于兵棋推演、武器装备仿真等场景. 现有的多智能体强化学习算法 (即表 1 中 1–6 行所示的工作) 一般将外部智能体考虑为环境的一部分, 在扩展到自博弈训练时, 很容易出现不收敛或者收敛到低水平策略的情况; 而自博弈的方法尽管可以控制多个目标智能体, 但是它们是将所有目标的智能体和外部智能体分别建模为两个竞争的个体, 从而使得算法收敛于双人纳什均衡.

表 1　本文所提方案在多智能体强化学习领域中的大概位置

| 方法 | 算法设计 | | | | | | 应用场景 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 有无深度学习 | 训练方式 | 执行方式 | 执行时的通信 | 策略优化方法 | 对手模型 | 智能体规模 $N$ | 关注的非稳态问题 | 多智能体环境 |
| WoLF-PHC[13] | 无 | — | — | 无 | 梯度下降 | 无 | $N$≤2 | 外部智能体的变化 | 完全竞争 |
| IDQN[14] | 有 | 分布式训练 | 分布式执行 | 无 | 时间差分 | 无 | $N$≤5 | 无 | 无限制 |
| COMA[2] | 有 | 集中式训练 | 分布式执行 | 无 | Actor-Critic | 无 | $N$≤5 | 目标智能体的变化 | 合作型 |
| MADDPG[9] | 有 | 集中式训练 | 分布式执行 | 无 | Actor-Critic | 无 | $N$≤10 | 目标/外部智的能体 | 竞争合作混合 |
| QMIX[4] | 有 | 集中式训练 | 分布式执行 | 无 | 值函数分解 | 无 | $N$≤10 | 目标智能体的变化 | 合作型 |
| MICRA[1] | 有 | 集中式训练 | 分布式执行 | 广播观测信息 | Actor-Critic | 模仿学习 | $N$≤10 | 目标/外部智能体 | 组内合作, 组间竞争 |
| ICRA (ours) | 有 | 集中式训练 | 分布式执行 | 广播观测信息 | Actor-Critic 自博弈优化 | 模仿学习 | $N$≤10 | 目标/外部智能体的变化 | 组内合作, 组间竞争 |
| NFSP[5] | 有 | 集中式训练 | 集中执行 | 无 | 自博弈优化 | 虚拟对手 | $N$≤10 | 外部智能体的变化 | 双人博弈或者可以建模为双人博弈的环境 |

## 2 背景知识

本文所提出的方法以随机博弈和虚拟遗憾最小化为基础, 以下将介绍这两方面的相关背景知识.

### 2.1 随机博弈

随机博弈 (stochastic game) 也称为马尔可夫博弈 (Markov game), 它是马尔可夫决策过程在多智能体场景中的扩展. 在随机博弈中, 多个智能体同时采取动作, 共享的环境根据全部智能体采取的联合动作来改变自身状态并将新的状态和奖励信号反馈给每个智能体[2,3].

**定义 1**. 随机博弈是一个七元组 $G=<S, N, A, T, R, O, \Omega>$, 其中,

• $S$ 是环境状态的有限集合. $s^t \in S$ 表示 $t$ 时刻的环境状态.

• $N$ 是由 $n$ 个智能体组成的集合, 每个智能体分别用 1, 2,…, $n$ 来标记.

• $A=A_1\times A_2\times\ldots\times A_n$ 是联合动作集合, $A_i$ 表示智能体 $i$ 能采取的动作集合. $a_i^t \in A_i$ 表示智能体 $i$ 在 $t$ 时刻采取的动作, $\vec{a}^t=[a_1^t, a_2^t, \ldots, a_n^t]$ 表示在 $t$ 时刻全部智能体采取的联合动作.

• $T$: $S\times A\times S\to[0, 1]$ 是状态转移概率函数. $T(s'|s,\vec{a})$ 返回在状态 $s$ 下采取联合动作 $\vec{a}$ 后环境转移到状态 $s'$的概率.

• $O=O_1\times O_2\times\ldots\times O_n$ 是联合观测的有限集合, 其中 $O_i$ 是智能体 $i$ 的观测集合. $o_i^t \in O_i$ 表示智能体 $i$ 在 $t$ 时刻的观测, $\vec{o}^t=[o_1^t, o_2^t, \ldots, o_n^t]$ 表示 $t$ 时刻全部智能体的联合观测.

• $\Omega$: $S\times A\to O$ 是观测函数.

• $R=\{R_1, R_2,\ldots, R_n\}$是奖励函数的集合, 其中 $R_i$: $S\times A\to\mathbb{R}$ 表示智能体 $i$ 的奖励函数.

在随机博弈中, 智能体 $i$ 的个体策略定义为 $\pi_i$: $O_i\times A_i\to[0, 1]$; 全部智能体的联合策略定义为个体策略的乘积形式: $\pi(\vec{a}|\vec{o})=\prod_{i\in N}\pi_i(a_i|o_i)$. 在联合策略 $\pi$ 下, 智能体 $i$ 的状态价值函数定义为 $v_{\pi,i}(s)=\sum_{t=0}^{H}\gamma^t\mathbb{E}[r^t|s^0=s,\pi]$, 其中 $H$ 为终止步长, $\gamma\in[0, 1]$ 为折扣系数, $r^t=R_i(s^t,\vec{a}^t)$; 智能体 $i$ 的状态-动作价值函数 (即 Q 函数) 定义为 $Q_{\pi,i}(s,\vec{a})=\mathbb{E}[R_i(s,\vec{a})+\gamma v_{\pi,i}(s')]$, 其中, $s'$表示执行联合动作 $\vec{a}$ 后环境跳转到的下一个状态. 为了符号简便, 后文将使用 $v_i(s)$ 和 $Q_i(s, \vec{a})$ 来分别表示在当前策略下智能体 $i$ 的状态价值函数和 Q 函数.

### 2.2 虚拟遗憾最小化

虚拟遗憾最小化 (counterfactual regret minimization, CFR)[8] 是一种利用对手在博弈中采取的实际策略, 在线地最小化己方的虚拟遗憾, 从而计算出己方最优策略的方法. 该方法在德州扑克、国际象棋、围棋等应用中取得了很好的效果[7,15,16].

博弈中的遗憾被定义为智能体采取的实际行动的代价和它可能获得的最小代价的差值.

**定义 2**. 在一个博弈中, 智能体 $i$ 的动作序列$<a^1, a^2,\ldots, a^H>$ 产生的遗憾为:

$$\frac{1}{H}\left[\sum_{t=1}^{H}c^t(a^t)-\min_{a\in A}\sum_{t=1}^{H}c^t(a)\right] \tag{1}$$

其中, $\{c^1, c^2,\ldots, c^H\}$为一组代价函数, $c^t(a)$ 返回 $t$ 时刻智能体采取动作 $a\in A_i$ 的代价, $H$ 是终止步长.

在 CFR 中, $H$ 时刻的虚拟遗憾计算方式如下:

$$R_{i,H}^{CF}(o_i,a)=\sum_{t=1}^{H}v_{\pi_i^t|o_i\mapsto a,i}(o_i)-v_{\pi_i^t,i}(o_i) \tag{2}$$

其中, $o_i$ 是智能体 $i$ 的观测值; 将策略 $\pi_i^t$ 在 $o_i$ 观测时采取的动作修改为 $a$, 其他部分保持不变, 由此得到的新策略记作 $\pi_i^t|o_i\mapsto a$; $v_{\pi,i}(o_i)$ 评估了在策略 $\pi$ 下, 智能体 $i$ 观察到 $o_i$ 时的期望收益. $v_{\pi,i}(o)$ 可以通过采样或者学习的方法估计出来; 一种可行的方法是按照第 2.1 节的定义式 $v_{\pi,i}(s)=\sum_{t=0}^{H}\gamma^t\mathbb{E}[r^t|s^0=s,\pi]$ 进行计算, 此时, 智能体 $i$ 的观测是环境的状态, 即 $o_i=s$.

$R_{i,H}^{CF}(o_i,a)$ 的直观含义为: 智能体 $i$ 在观测 $o_i$ 下采取动作 $a$ 的实际收益 (公式 (2) 第 1 项) 和按照现有策略能够

得到的期望收益 (公式 (2) 第 2 项) 之间的差值; 该差值表现了如果没执行 $a$ 动作, 智能体 $i$ 的后悔程度.

在 $t$ 时刻, 智能体 $i$ 首先分别计算出每个可能动作 $a\in A_i$ 的虚拟遗憾; 接着更新策略, 增加虚拟遗憾更大的动作的执行概率; 然后采取行动, 获得奖励, 根据奖励更新期望收益函数 $v(o)$; 随后在 $t$+1 时刻重复上述操作. 在这样的迭代中, 智能体的虚拟遗憾会逐渐变小, 从而实现策略最优化.

## 3 问题描述

本文所关注的多智能体系统形式化建模为定义 1 的形式. 特别地, 该系统的智能体被划分为两个群体, 即目标智能体 $\tau_1$ 和外部智能体 $\tau_2$, $\tau_1\cap\tau_2=\varnothing$ 且 $\tau_1\cup\tau_2=N$. 同时, 该系统满足如下 4 个基本假设.

假设 1. 环境是联合可观测的, 即任意时刻下, $s^t=\vec{o}^t$.

假设 2. 目标智能体能够直接观察到局部信息, 又能够接收到其他智能体传输的共享信息.

假设 3. 在训练阶段, 系统可以控制目标智能体和外部智能体, 并获得每个时间步下的 $\vec{o}^t$ 和 $\vec{a}^t$; 在执行阶段, 系统只能控制目标智能体, 每个目标智能体按照系统学习到的策略, 仅根据观测 $o_i^t$ 采取行动.

假设 4. 目标智能体之间完全合作, 它们的奖励函数相同, 记作 $R_0$. 外部智能体之间也完全合作, 并和目标智能体展开竞争, 它们的奖励函数也均相同且和目标智能体的奖励函数相反, 记作-$R_0$.

假设 1 和很多实际应用相符, 例如, 在协同电磁对抗中, 当完全掌握了敌我双方每个作战单位的观测值后, 自然就组成了全局状态. 假设 2 允许目标智能体之间进行必要的通信以共享信息. 假设 3 限定了各目标智能体能够独立地采取行动, 而不是依赖于某个决策中心, 同时由于假设 2 的共享信息, 每个目标智能体也具有一定的全局视野; 假设 3 和多智能体强化学习中经典的“集中式训练-分布式执行”框架[2,4,9]相一致. 假设 4 限定了本文所关注的合作-竞争混合型多智能体系统, 即群体内部完全合作, 群体之间相互竞争.

特别地, 目标智能体的联合策略记作 $\pi_{\tau_1}(\vec{a}_{\tau_1}|\vec{o}_{\tau_1})=\prod_{i\in\tau_1}\pi_i(a_i|o_i)$; 外部智能体的联合策略记作 $\pi_{\tau_2}(\vec{a}_{\tau_2}|\vec{o}_{\tau_2})=\prod_{j\in\tau_2}\pi_j(a_j|o_j)$. $\vec{o}_{\tau_1}$, $\vec{a}_{\tau_1}$, $\vec{o}_{\tau_2}$ 和 $\vec{a}_{\tau_2}$ 分别表示目标智能体的联合观测和联合动作, 以及外部智能体的联合观测和联合动作.

本文的目标是, 在合作-竞争混合型多智能体系统中, 当外部智能体策略未知时, 为目标智能体训练出最优联合策略. 在该最优联合策略下, 目标智能体面对任意的外部智能体策略, 都能够获得最大累计奖励.

## 4 虚拟遗憾优势自博弈方法

本节首先概述本文所提出的方法, 然后具体介绍其中的两个关键技术——虚拟遗憾优势和基于模仿学习的智能体建模, 最后给出模仿虚拟遗憾优势算法以及基于该算法的自博弈训练方法.

### 4.1 方法概览

模仿虚拟遗憾优势 (imitation counterfactual regret advantage, ICRA) 算法, 是在 COMA[2]的基础上提出的, 通过集成虚拟遗憾优势和智能体建模两个关键技术, 改进 COMA 中优势函数的计算方式, 使该优势函数能更准确地指导每个目标智能体更新策略. 基于 ICRA 算法, 本文提出了一种自博弈训练方法, 其中, 目标智能体和外部智能体同时采用 ICRA 算法, 交替进行策略更新, 同时生成一个外部智能体策略库; 当策略交替更新过程满足终止条件后, 根据外部智能体策略库再次训练目标智能体的联合策略, 最后得到目标智能体的最优策略.

图 1 给出了 ICRA 算法的基本框架. 它包括一个集中式的评价器 (critic), 多个模仿器 (imitator) 和多个行动器 (actor). 评价器根据每个行动器学到的目标智能体的个体策略、模仿器学习到的外部智能体的个体策略、以及环境反馈的奖励值和观测值, 为每个行动器计算对应的虚拟遗憾优势 (counterfactual regret advantage, CRA); 同时评价器还通过时序差分学习 (temporal difference learning, TD) 来训练目标智能体的联合 Q 网络, 其损失函数如下:

$$L(\theta^c)=(r+\gamma\max_{a'}Q(\vec{o}'^{+1},\vec{a}';\hat{\theta}^c)-Q(\vec{o}^t,\vec{a}^t;\theta^c))^2 \tag{3}$$

其中, $r=R_0(\vec{o}^t,\vec{a}^t)$, $\theta^c$ 是 Q 网络的参数, $\hat{\theta}^c$ 是目标 Q 网络的参数, 目标 Q 网络是 Q 网络历史时刻的副本. 以 Q 网络和目标网络的差值为损失函数, 可以保证学习过程的稳定性[17].

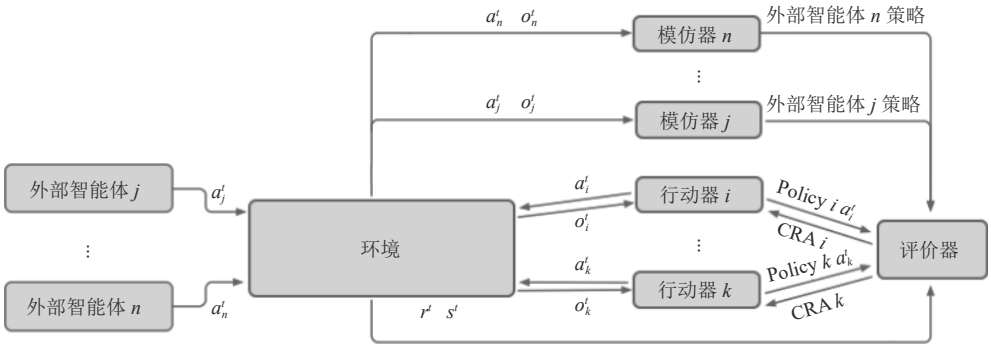



图 1 ICRA 算法基本框架

每个模仿器对应一个外部智能体, 模仿器 $i$ 以外部智能体 $i$ 的观测为输入, 输出该智能体可以采取的各种可能动作的概率. 为建立从观测到动作概率的映射, 本文采用模仿学习, 将外部智能体的历史决策轨迹作为示教数据, 并以示教数据来模仿外部智能体的策略.

每个行动器对应一个目标智能体, 行动器利用虚拟遗憾优势, 按照策略梯度方法[18,19]来更新其对应的智能体的策略神经网络 $\pi_i(a^t|o^t;\theta_i^a)$, 其中, $\theta_i^a$ 是行动器 $i$ 的策略神经网络的参数. 策略梯度的计算方式如下:

$$g=\mathbb{E}\left[\sum_{t=0}^{H}\Psi_t\nabla_\theta\log\pi_i(a^t|o^t;\theta_i^a)\right] \tag{4}$$

其中, $H$ 表示终止时间步, $\Psi_t$ 表示评价函数. 评价函数可以有多种形式, 例如 Q 函数 $Q_\pi(s^t,a^t)$、轨迹奖励总和函数 $\sum_{t=0}^{H}r^t$, 或者基线优势函数 $A_\pi(s^t,a^t)=Q_\pi(s^t,a^t)-v_\pi(s^t)$. 以 $Q_\pi(s^t,a^t)$ 或 $A_\pi(s^t,a^t)$ 作为评价函数产生的方差要明显低于以 $\sum_{t=0}^{H}r^t$ 为评价函数的方差[19]. 然后, 行动器根据公式 (4) 计算出的梯度 $g$, 采用随机梯度下降法更新策略神经网络 $\pi_i$ 的参数.

### 4.2 基于虚拟遗憾优势的策略梯度方法

本节设计虚拟遗憾优势作为评价函数 $\Psi_t$, 以指导行动器按照公式 (4) 进行策略更新. 首先, 参考公式 (2) 的虚拟遗憾, 设计虚拟遗憾优势如下. 在第 $H$ 轮训练中, 目标智能体 $i$ 的虚拟遗憾优势为:

$$\begin{aligned}\mathcal{A}_{H,i,\pi^H}(\vec{o},\vec{a})&=v_{\pi^H|\vec{o}\mapsto a_i}(\vec{o})-v_{\pi^H}(\vec{o})\\&=\mathbb{E}_{\vec{a}_{-i}\sim\pi^H_{-i}}Q(\vec{o},[a_i,\vec{a}_{-i}])-\mathbb{E}_{\vec{a}_{\tau_1}\sim\pi^H_{\tau_1},\vec{a}_{\tau_2}\sim\pi^H_{\tau_2}}[Q(\vec{o},[\vec{a}_{\tau_1},\vec{a}_{\tau_2}])]\\&=\sum_{\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}}\pi^H_{\tau_1-i}(\cdot)\pi^H_{\tau_2}(\cdot)Q(\vec{o},[a_i,\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}])-\sum_{\vec{a}_{\tau_1},\vec{a}_{\tau_2}}\pi^H_{\tau_1}(\cdot)\pi^H_{\tau_2}(\cdot)Q(\vec{o},[\vec{a}_{\tau_1},\vec{a}_{\tau_2}])\end{aligned} \tag{5}$$

其中, $\pi^H$ 表示第 $H$ 轮训练时的联合策略 (为了简化符号, 在不引起歧义的情况下, 训练轮数符号 $T$ 可以省略, 即 $\pi^H=\pi$, $\mathcal{A}_{H,i,\pi^H}=\mathcal{A}_{i,\pi}$), $\pi^H|\vec{o}\mapsto a_i$ 表示在联合观测 $\vec{o}$ 下智能体 $i$ 始终执行动作 $a_i$, 而其他智能体遵循原策略 $\pi$ 采取行动; $\vec{a}_{\tau_1}$ 表示目标智能体的联合动作; $\vec{a}_{\tau_2}$ 表示外部智能体的联合动作; $\pi_{-i}$ 表示除智能体 $i$ 以外其他所有智能体的联合策略; $\pi_{\tau 1-i}$ 表示除智能体 $i$ 以外其他目标智能体的联合策略, $\vec{a}_{\tau_1-i}$ 表示除智能体 $i$ 外其他目标智能体的联合动作. 直观来看, 虚拟遗憾优势表达了“在当前的联合策略 $\pi$ 下如果目标智能体 $i$ 在联合观测 $\vec{o}$ 下不采取动作 $a_i$, 它会有多少遗憾?”

相比于在其他策略梯度方法使用的评价函数 $\Psi_t$, 公式 (5) 对于策略的评价更为准确, 其他形式的评价函数可以视为公式 (5) 在一定条件下的特例. 下面根据虚拟遗憾优势推导出两种典型的优势函数.

(1) 单智能体. 在单智能体场景中, $s$=$o$, 有 $v_{pi|s\mapsto a}(s)=R(s,a)+\gamma v_{\pi|s\mapsto a}(s')=R(s,a)+\gamma v_\pi(s')$. 因为 $\pi|s\mapsto a$ 只会改变状态 $s$ 下的动作, 即 $v_{\pi|s\mapsto a}(s')=v_\pi(s')$, 所以根据公式 (5) 有:

$$\mathcal{A}_\pi(s,a)=R(s,a)+\gamma v_\pi(s')-v_\pi(s) \tag{6}$$

这种优势函数 $A_\pi$ 在文献 [19] 中被称作 GAE($\gamma$, 0) 优势函数. 如果我们假设 $v_{\pi|s\mapsto a}(s)=Q_\pi(s,a)$, 则:

$$\mathcal{A}_\pi(s,a)=Q_\pi(s,a)-v_\pi(s) \tag{7}$$

这种优势函数 $A_\pi$ 就是文献 [18] 中给出的基本优势函数.

(2) 多智能体. 在多智能体场景中, $s=\vec{o}$, 如果假定在状态 $s$ 下, 除 $i$ 以外的其他智能体的联合动作 $\vec{a}_{-i}$ 保持不变, 即 $\pi=\pi|s\mapsto\vec{a}_{-i}$, 那么就有:

$$\mathcal{A}_{i,\pi}(s,\vec{a})=v_{\pi|s\mapsto a_i}(s)-v_\pi(s)=v_{\pi|s\mapsto\vec{a}}(s)-v_{\pi|s\mapsto\vec{a}_{-i}}(s)=Q(s,\vec{a})-\sum_{a_i\in A_i}(\pi_i(a_i|o_i)Q(s,a_i,\vec{a}_{-i})) \tag{8}$$

公式 (8) 就是 COMA 算法[2]中的反事实基线优势. 显然, COMA 中的反事实基线优势是本文的虚拟遗憾优势在 $\pi=\pi|s\mapsto\vec{a}_{-i}$ 条件下的特殊情况.

在多智能体场景中, $\pi=\pi|s\mapsto\vec{a}_{-i}$ 只有在算法收敛到固定策略时才有可能成立, 此时, 除了 $i$ 之外的其他智能体在 $s$ 状态下确定地采取 $\vec{a}_{-i}$ 动作; 然而, 实际上, 在学习过程中, $\pi\neq\pi|s\mapsto\vec{a}_{-i}$, $\vec{a}_{-i}$ 随着学习过程在不断变化. 公式 (5) 相当于 COMA 中的反事实基线优势在 $\pi$ 不一定等于 $\pi|s\mapsto\vec{a}_{-i}$ 的条件下的扩展. 因此, 相较于 COMA 的反事实基线优势, 公式 (5) 所计算的虚拟遗憾优势能够更准确地评价策略 $\pi$ 的价值.

然而公式 (5) 的计算量随着智能体数量的增加而呈指数增长. 当智能体规模较大时, 直接利用公式 (5) 计算虚拟遗憾优势不太可行. 因此, 本文设计了一种剪枝方法, 忽略公式 (5) 中的低概率项以降低计算量, 方法如下.

前 $K$ 项剪枝. 给定智能体 $i$, 状态 $s$=[$o_1$,…, $o_n$], 联合动作 $\vec{a}$=[ $a_1$,…, $a_n$], 和个体策略 $\pi_i$, 则智能体 $i$ 在观测 $o_i$ 下, 选出其概率 $\pi_i(a|o_i)$ 排名前 $K$ 的动作组成的集合为 $\{a_{i,1}, a_{i,2},\ldots, a_{i,K}\}$. 这些动作和智能体当前执行的动作 $a_i$ 构成了动作空间 $A_i^{\text{topK}}=\{a_{i,1}, a_{i,2},\ldots, a_{i,K}\}\cup\{a_i\}$. 根据智能体 $i$ 的原策略 $\pi_i$, 按照公式 (9) 计算出剪枝后的所有可能的动作的执行概率:

$$\pi_i^{\text{topK}}(a|o_i)=\begin{cases}\pi_i(a|o_i), & a\in A_i^{\text{topK}}\\ 0, & a\notin A_i^{\text{topK}}\end{cases} \tag{9}$$

概率函数 $\pi_i^{\text{topK}}$ 的其他部分和 $\pi_i$ 相同. 在使用公式 (5) 计算虚拟遗憾优势时, 将 $\pi_i$ 替换为 $\pi_i^{\text{topK}}$. 注意, 此处 $\pi_i^{\text{topK}}$ 相当于权重调整, 而不是一个严格的策略, 有 $\sum_{a\in A_i}\pi_i^{\text{topK}}(a|o_i)\leqslant 1$.

下面用一个简单的例子来展示如何利用前 $K$ 项剪枝来减少公式 (5) 的计算量. 假设有两个智能体 1 和 2, 其动作空间均为 $\{l, r, g\}$, 当前的状态为 $s$=[$o_1$, $o_2$], 需要评价的联合动作为 $\vec{a}=[l,g]$. 已知这两个智能体的个体策略为: $\pi_1(l|o_1)$=0.7, $\pi_1(r|o_1)$=0.2, $\pi_1(g|o_1)$=0.1, $\pi_2(l|o_2)$=0.1, $\pi_2(r|o_2)$=0.8, $\pi_2(g|o_2)$=0.1. 假如取 $K$=1, 则有 $A_1^{\text{topK}}=\{l\}$, $A_2^{\text{topK}}=\{r,g\}$, 那么, $\pi_1^{\text{topK}}(l|o_1)=0.7$, $\pi_1^{\text{topK}}(r|o_1)=0$, $\pi_1^{\text{topK}}(g|o_1)=0$, $\pi_2^{\text{topK}}(l|o_2)=0$, $\pi_2^{\text{topK}}(r|o_2)=0.8$, $\pi_2^{\text{topK}}(g|o_2)=0.1$. 则智能体 1 的虚拟遗憾优势为:

$$\begin{aligned}\mathcal{A}_{1,\pi}(s,[l,g])\approx&\mathcal{A}_{1,\pi^{\text{topK}}}(s,[l,g])\\=&(\pi_2^{\text{topK}}(r|o_2)Q(s,[l,r])+\pi_2^{\text{topK}}(g|o_2)Q(s,[l,g]))-(\pi_1^{\text{topK}}(l|o_1)\pi_2^{\text{topK}}(r|o_2)Q(s,[l,r])\\&+\pi_1^{\text{topK}}(l|o_1)\pi_2^{\text{topK}}(g|o_2)Q(s,[l,g]))=0.24Q(s,[l,r])+0.03Q(s,[l,g]).\end{aligned}$$

如果不进行剪枝, 公式 (5) 需要计算 $2\times|A_1|\times\ldots\times|A_n|$项; 而在使用前 $K$ 项剪枝后, 公式 (5) 最多只需计算 $2\times(K+1)^n$ 项 (小于 $2\times|A_1|\times\ldots\times|A_n|$). 直观地看, $K$ 越大 ($K\leqslant|A_i|$), 估计误差 $|\mathcal{A}_{i,\pi}(s,\vec{a})-\mathcal{A}_{i,\pi^{\text{topK}}}(s,\vec{a})|$ 就越小, 但计算量也越大. 在实际问题中, 如果智能体的数量不超过 5 个, 那么将 $K$ 设定为$|A_i|$; 否则, 将 $K$ 设置为一个较小的正整数 (在本文的实验中, $K$ 设定为 2).

接下来, 介绍利用强化学习的 Q 网络来估计虚拟遗憾优势 CRA 的方法.

在第 $H$ 轮训练中, 智能体 $i$ 的全局平均 CRA 按公式 (10) 计算:

$$\mathcal{A}_{H,i,\pi^H}^{\text{all}}=\frac{1}{H}\sum_{t=1}^{H}(v_{\pi^t|s\mapsto a_i}(s)-v_{\pi^t}(s))=\frac{1}{H}((H-1)\mathcal{A}_{H-1,i,\pi^{H-1}}^{\text{all}}+\mathcal{A}_{H,i,\pi^H}(s,\vec{a})) \tag{10}$$

训练过程中, 由于强化学习算法对 $v(s)$ 的最新估计要比之前的估计更为准确, 因此在决策时要给予新的 CRA 更高的权重. 于是, 本文设计了累计折扣 CRA:

$$\mathcal{A}^{\gamma}_{H,i,\pi^H}=\gamma_c\mathcal{A}^{\gamma}_{H-1,i,\pi^{H-1}}+\mathcal{A}_{H,i,\pi^H} \tag{11}$$

其中, $\gamma_c\in[0,1]$ 是折扣因子, $\gamma_c$ 通常的取值范围为 $1\times10^{-3}$–$1\times10^{-1}$. 在状态-动作空间较大时, 想要维系一个表格来存储所有的 $\mathcal{A}^{\gamma}_{H,i,\pi^H}(s,\vec{a})$ 是不现实的. 在基于优势的遗憾最小化 (advantage-based regret minimization, ARM)[20] 中, 设计了一种特殊的 Q 网络来估计 $\mathcal{A}^{\text{all}}_{H,i,\pi^H}$, 受此启发, 本文使用目标 Q 网络来估计累计折扣 CRA:

$$\begin{aligned}\mathcal{A}^{\gamma}_{H,i,\pi^H}(s,\vec{a})\approx&\gamma_c(Q(s,\vec{a};\theta^c)-\sum_{a\in A_i}(\pi_i(a|o_i)Q(s,a,\vec{a}_{-i};\hat{\theta}^c)))+\mathcal{A}_{H,i,\pi^H}(s,\vec{a})\\=&\gamma_c(Q(s,\vec{a};\theta^c)-\sum_{a\in A_i}(\pi_i(a|o_i)Q(s,a,\vec{a}_{-i};\hat{\theta}^c)))+v_{\pi^H|s\mapsto a_i}(s)-v_{\pi^H}(s)\\=&\gamma_c(Q(s,\vec{a};\theta^c)-\sum_{a\in A_i}(\pi_i(a|o_i)Q(s,a,\vec{a}_{-i};\hat{\theta}^c)))+\Bigg(\sum_{\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}}\pi^H_{\tau_1-i}(\cdot)\pi^H_{\tau_2}(\cdot)Q(s,[a_i,\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}];\theta^c)\\&-\sum_{\vec{a}_{\tau_1},\vec{a}_{\tau_2}}\pi^H_{\tau_1}(\cdot)\pi^H_{\tau_2}(\cdot)Q(s,[\vec{a}_{\tau_1},\vec{a}_{\tau_2}];\theta^c)\Bigg)\\\approx&\gamma_c(Q(s,\vec{a};\theta^c)-\sum_{a\in A_i}(\pi_i(a|o_i)Q(s,a,\vec{a}_{-i};\hat{\theta}^c)))+\Bigg(\sum_{\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}}\pi^{\text{topK}}_{\tau_1-i}(\cdot)\pi^{\text{topK}}_{\tau_2}(\cdot)Q(s,[a_i,\vec{a}_{\tau_1-i},\vec{a}_{\tau_2}];\theta^c)\\&-\sum_{\vec{a}_{\tau_1},\vec{a}_{\tau_2}}\pi^{\text{topK}}_{\tau_1}(\cdot)\pi^{\text{topK}}_{\tau_2}(\cdot)Q(s,[\vec{a}_{\tau_1},\vec{a}_{\tau_2}];\theta^c)\Bigg)\end{aligned} \tag{12}$$

其中, $\pi^{\text{topK}}_{\tau_1}$ 和 $\pi^{\text{topK}}_{\tau_2}$ 分别表示经过前 $K$ 项剪枝后当前目标智能体和外部智能体的策略. $Q(\cdot;\theta^c)$ 表示当前的 Q 网络, $Q(\cdot;\hat{\theta}^c)$ 表示目标 Q 网络, 目标网络的设计可以进一步参考文献 [17]. 值得注意的是, 公式 (12) 最后的计算由两部分组成, 前一部分 (即项 $Q(s,\vec{a};\theta^c)-\sum_{a\in A_i}(\pi_i(a|o_i)Q(s,a,\vec{a}_{-i};\hat{\theta}^c))$) 实际上是 COMA 中的反事实基线优势 (参考公式 (8)), 而后一项则是经前 $K$ 项剪枝后的 CRA. 考虑到 CRA 会随着当前策略的变化而变化, 因此, 公式 (12) 采用了当前 Q 网络来进行估计, 而没有使用相对稳定的目标 Q 网络.

最后, 智能体 $i$ 的基于 CRA 的策略梯度按照公式 (13) 计算:

$$g_{cr,i}=\mathbb{E}_{s^t\sim D,\vec{a}^t\sim\pi}\left[\sum_{t=0}^{H}\nabla_{\theta^a_i}\log(\pi_i(a^t_i|o^t_i;\theta^a_i))\mathcal{A}^{\gamma}_{H,i,\pi}(s^t,\vec{a}^t)\right] \tag{13}$$

其中, $D$ 是交互轨迹数据, 即在当前策略的控制下产生的状态、动作、奖励序列. 每个行动器按公式 (14) 训练其对应的智能体的个体策略神经网络 $\pi_i(\cdot;\theta^a_i)$:

$$\theta^a_i=\theta^a_i+\alpha g_{cr,i} \tag{14}$$

其中, $\alpha\in(0,1]$ 表示学习率.

基于虚拟遗憾优势的策略梯度方法的全过程简单总结如下: 首先根据虚拟遗憾设计出虚拟遗憾优势的定义式, 即公式 (5); 并利用前 $K$ 项剪枝, 简化公式 (5) 的计算量; 然后用目标 Q 网络, 按照公式 (12) 来估计虚拟遗憾优势; 最后根据公式 (13), 公式 (14) 来更新目标智能体的个体策略.

### 4.3 基于模仿学习的智能体建模

本节介绍用于模仿外部智能体策略的模仿器的设计方法. 智能体建模是指通过构造预测模型, 来推断所关注的智能体的行为及其采取该行为的原因[1,21].

公式 (12) 在估计虚拟遗憾优势时, 采用了全部智能体的联合策略 $\pi$, 目标智能体可以直接从行动器获得这个联合策略 $\pi_{\tau 1}$; 而对外部智能体而言, 其联合策略 $\pi_{\tau 2}$ 无法直接得到, 本文基于模仿学习设计外部智能体的联合策略

获取方法.

本文采用行为克隆 (behavior cloning) 方法[22]来模仿外部智能体的策略, 其方法为利用监督学习从专家的示教轨迹数据中学习行为策略. 在本文中, 对于外部智能体 $i$, 根据假设 3, 在训练阶段, 可以获得它所有的观测和行动数据. 外部智能体 $i$ 的每一条行动数据形如 $(o_i, a_i)$, 其中 $o_i$ 作为监督学习中的属性/特征, $a_i$ 作为作监督学习中的标签, 这些数据组成了智能体 $i$ 的历史行为数据集 $D_i$. 用于拟合外部智能体 $i$ 策略的神经网络记作 $\pi_{i,\tau_2}(\cdot;\theta_i^a)$, 用一个多层感知机来实现. $\pi_{i,\tau_2}(\cdot;\theta_i^a)$ 的损失函数为 KL 散度:

$$L^{KL}(\pi_{i,\tau_2},\delta_{\tau_2})=-\frac{1}{N}\sum_{j=0}^{N-1}\sum_{k=0}^{K-1}a_{j,k}\ln\pi_{i,\tau_2}(a_{j,k}|o_j;\theta_i^a) \tag{15}$$

其中, $\delta_{\tau_2}$ 表示数据集 $D_i$ 上 $o_i$ 和 $a_i$ 的真实分布, $a_{j,k}$ 表示在数据集 $D_i$ 中的第 $j$ 个样本的真实动作为第 $k$ 号动作, $o_j$ 为数据集 $D_i$ 中的第 $j$ 个样本的真实观测值, $K$=|$A_i$|, $N$ 为数据集 $D_i$ 的样本数. 动作空间中的所有动作编码为独热码 (one-hot) 向量形式.

对于环境中的每个外部智能体, 均采用上述方法建立起一个模仿器用于拟合其策略. 得到每一个外部智能体的模仿策略后, 根据 $\pi_{\tau_2}(\vec{a}_{\tau_2}|\vec{o}_{\tau_2})=\prod_{j\in\tau_2}\pi_{j,\tau_2}(a_j|o_j)$ 计算出外部智能体的联合策略 $\pi_{\tau 2}$.

### 4.4 ICRA 算法和自博弈训练方法

本节在第 4.2 和 4.3 节的基础上, 提出模仿虚拟优势 (ICRA) 的神经网络架构以及 ICRA 算法, 并设计基于 ICRA 的自博弈训练方法. 图 2 展示 ICAR 的神经网络架构的示意图, 为简洁起见, 该网络只展示了两个目标智能体 ($k$ 和 $n$) 和两个外部智能体 ($i$ 和 $j$) 的情况.

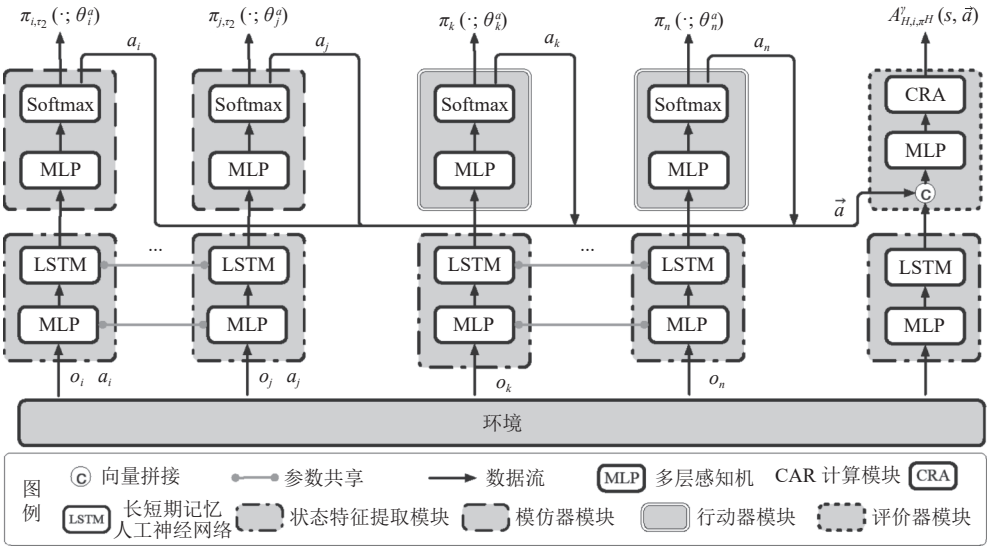



图 2 ICRA 算法的神经网络架构示意图

该神经网络中有如下 4 类模块.

• 状态特征提取器: 它从环境反馈的原始观测中提取高层特征, 供后续模块处理. 本文利用循环神经网络来处理每个智能体的局部观测. 因为所有目标智能体/外部智能体的观测都有相似的结构, 本文对目标智能体和外部智能体的状态特征提取器进行参数共享.

• 模仿器: 它利用行为克隆来模仿外部智能体的策略, 其输出层是 Softmax 层, 能够生成对应外部智能体在特定状态下采取各种可能动作的概率.

• 行动器: 它的目标是学习每个目标智能体的个体策略, 学习方法是基于 CRA 的策略梯度法.

• 评价器: 它的目标是学习一个联合 Q 函数 $Q(\vec{o},\vec{a};\theta^c)$, 并为每个行动器计算出相应的 CRA 值, 用于指导行动器进行策略更新.

算法 1 给出了 ICAR 算法的伪代码. 在算法的一个训练轮 $e$ 中, 包含如下 4 个步骤. (1) 收集数据: 算法第 2–4 行, 目标智能体和外部智能体同时在环境中采取行动, 环境给出反馈; 一条状态-动作-奖励轨迹的形式为 $(\vec{o}^1,\vec{a}^1,r^1,\ldots,\vec{o}^H,\vec{a}^H,r^H)$, $H$ 为终止时间步, $r^t$ 是 $t$ 时刻环境反馈给目标智能体的即时奖励. (2) 训练评价器: 算法第 5–9 行采用了时间差分学习[17]来训练目标智能体的联合 Q 网络, 采用硬更新的方式每隔 $C$ 步同步目标 Q 网络和 Q 网络的参数, 实验中 $C$ 设置为 150. (3) 训练模仿器: 算法第 10–12 行以轨迹缓存 $D$ 为训练数据, 对于每个外部智能体, 训练一个用于模仿其策略的个体策略网络; 特别地, 对于外部智能体 $i$, 从 $D$ 中的每个轨迹 $tr=(\vec{o}^1,\vec{a}^1,r^1,\ldots,\vec{o}^H,\vec{a}^H,r^H)$ 中抽取出外部智能体 $i$ 的轨迹 $tr_i=(o_i^1,a_i^1,\ldots,o_i^H,a_i^H)$, 并将所有轨迹 $tr_i$ 拆解成 $(o_i, a_i)$ 的形式, 以公式 (15) 为损失函数训练网络 $\pi_{i,\tau_2}(\cdot;\theta_i^a)$. (4) 训练行动器: 在算法第 13–20 行, 基于 CRA 策略梯度, 为每个目标智能体训练其个体策略网络 $\pi_i(\cdot;\theta_i^a)$. 在训练 $Q(.;\theta^c)$, $\pi_i(\cdot;\theta_i^a)$ 和 $\pi_{j,\tau_2}(\cdot;\theta_j^a)$ 时, 均要调整其所属模块下的状态特征提取模块的网络参数, 并和其他智能体同步对应网络的参数 (如图 2 中带圆端点的直线所示).

**算法 1**. ICRA 算法训练过程.

输入: 随机初始化的联合 Q 网络 $Q(\cdot;\theta^c)$ 和目标联合 Q 网络 $Q(\cdot;\hat{\theta}^c)$; 为每个目标智能体随机初始化个体策略网络 $\pi_i(\cdot;\theta_i^a)$; 为每个外部智能体随机初始化的个体策略网络 $\pi_j^{\tau_2}(\cdot;\theta_j^a)$;

输出: 训练后的每个目标智能体个体策略的网络参数 $\theta_i^a$.

1. **for** 每个训练轮 $e$ do
2. 清空轨迹缓存 $D$;
3. 每个目标智能体 $i$ 同时采用 $\pi_i(a|o_i;\theta_i^a)$ 和环境交互, 得到 $n$ 条状态-动作-奖励轨迹, 并将它们存到 $D$ 中;
4. 从 $D$ 中收集 $m$ 个回合数据;
5. **for** $t$=1 to $H$ **do** /*训练评价器模块的联合 Q 网络*/
6. 求公式 (3) 关于 $\theta^c$ 的梯度$\Delta\theta^c$;
7. $\theta^c \leftarrow \theta^c + \alpha\Delta\theta^c$;
8. 每隔 $C$ 步设置 $\hat{\theta}^c \leftarrow \theta^c$;
9. **end for**
10. **for** $i \in \tau_2$ **do** /*采用基于模仿学习的方法训练每个模仿器模块的个体策略网络*/
11. 在缓存 $D$ 上, 用公式 (15) 优化神经网络参数 $\theta_i^a$;
12. **end for**
13. **for** $i \in \tau_1$ **do** /*采用基于 CRA 策略梯度的方法训练每个行动器模块的个体策略网络*/
14. **for** $t$=$H$ **down to** 1 **do**
15. 根据公式 (12) 计算累计折扣 CRA $\mathcal{A}^{\gamma}_{H,i,\pi^H}$;
16. $\Delta g \leftarrow \nabla_{\theta_i^a}\log(\pi_i(a_i^t|o_i^t;\theta_i^a))\mathcal{A}^{\gamma}_{H,i,\pi^H}(s^t,\vec{a}^t)$;
17. $g_{cr,i} \leftarrow g_{cr,i} + \Delta g$;
18. **end for**
19. $\theta_i^a \leftarrow \theta_i^a + \alpha g_{cr,i}$;
20. **end for**
21. **end for**

在 ICRA 基础上, 本文设计了自博弈训练方法. 为了方便叙述, 将在线执行阶段环境中的博弈双方分别称为红方和蓝方. 在自博弈训练过程中, 红蓝双方同时采用 ICRA 算法进行训练, 对于红方而言, 红方的智能体为目标智能体, 蓝方的智能体为外部智能体; 对于蓝方而言, 蓝方的智能体为目标智能体, 红的智能体为外部智能体. 控制红、

蓝双方的 ICRA 算法及其神经网络分别记作 ICRA1 和 ICRA2. 算法 2 给出了基于 ICRA 的自博弈训练过程的伪代码.

**算法 2**. 基于 ICRA 的自博弈训练.

输入: 分别输入两组 ICRA 的参数, 每组参数和算法 1 的输入相同;
输出: 训练后的每个目标智能体 (红方) 个体策略的网络参数 $\theta_i^a$.

1. 初始化 ICRA1 和 ICRA2 中的所有参数;
2. 初始化红方策略库 K1 和蓝方策略库 K2;
3. **for** $e$=1 to $H$ **do**
4. ICRA1 和 ICRA2 同时在环境中运行, 固定 ICRA2 所有网络参数, 只训练 ICRA1, 训练 $c$ 轮;
5. 将 ICRA1 训练好的目标智能体的联合策略 $\pi_e$ 加入到红方策略库中;
6. ICRA1 和 ICRA2 同时在环境中运行, 固定 ICRA1 所有网络参数, 只训练 ICRA2, 训练 $c$ 轮;
7. 将 ICRA2 训练好的目标智能体的联合策略 $\pi_e$ 加入到蓝方策略库中;
8. **end for**
9. *flag*←TRUE;
10. **while** *flag* **do**
11. *flag*←FALSE;
12. **for** 红方策略库 K1 中的每个联合策略 $\pi_{e1}$ **do**
13. **if** $\forall\pi_{e2} \in$ K2, $run1(\pi_{e1}, \pi_{e2})<\max_{\pi\in K1} run1(\pi, \pi_{e2})$ **then**
14. 从 K1 中剔除联合策略 $\pi_{e1}$;
15. *flag*←TRUE;
16. **end if**
17. **end for**
18. **for** 蓝方策略库 K2 中的每个联合策略 $\pi_{e2}$ **do**
19. **if** $\forall\pi_{e1} \in$ K1, $run2(\pi_{e1}, \pi_{e2})<\max_{\pi\in K2} run2(\pi_{e1}, \pi)$ **then**
20. 从 K2 中剔除联合策略 $\pi_{e2}$;
21. *flag*←TRUE;
22. **end if**
23. **end for**
24. **end while**
25. **for** $e$=1 to $H'$ **do**
26. 从 K2 中随机选择一个联合策略 $\pi_{e2}$ 作为蓝方策略, 蓝方按照 $\pi_{e2}$ 策略在环境中运行, 训练 ICRA1, 训练 $c$ 轮;
27. **end for**

算法 2 可以分为 3 个步骤. (1) 生成策略库: 在第 3 行到第 7 行, 红蓝双方利用 ICRA 交替地生成策略, 并将策略分别添加到策略库 K1 和 K2 中. (2) 重复剔除严格劣策略: 在第 10–24 行, 分别从 K1 和 K2 中选出两个策略进行对抗, 通过重复剔除严格劣策略的方法, 将 K1 和 K2 中比其他策略的水平更低的策略剔除掉; 第 13 行、19 行的 $run1(\pi_1, \pi_2)$/$run2(\pi_1, \pi_2)$ 函数是指, 红方采用联合策略 $\pi_1$, 蓝方采用联合策略 $\pi_2$ 后, 红方/蓝方获得的累计奖励的期望值. (3) 在精化后的策略库中训练红方策略 (算法 2 的第 25–27 行).

## 5 实验分析

本文以协同电磁对抗作为研究案例, 设计了 3 种具有合作-竞争混合特点的任务场景, 并分别评估了基于 ICRA

的自博弈方法的性能. 主要关注如下 3 个研究问题.

• 研究问题 1 (学习性能). 同其他方法相比, ICRA 算法在对抗基于规则的外部智能体时表现如何?

• 研究问题 2 (自博弈效果). 同其他方法相比, 本文所提出的自博弈方法在没有基于规则的外部智能体的引导下表现如何?

• 研究问题 3 (消融实验). 本文方法中的各个部分对该方法的效果分别有怎样的影响?

### 5.1 实验设置

协同电磁对抗在现代战场上扮演着重要的角色. 在协同电磁对抗中, 具有不同电磁作战能力的空、地单位互相配合、采取各种电磁措施, 来提升己方优势, 削弱对方力量. 同时, 由于战场环境复杂、多变、不确定, 以及对方会采取反制措施, 事先为电磁作战单位设定静态的协同作战条令/策略往往难以获得理想的效果. 因此, 非常需要各个作战单位在交战过程中, 根据感知到的战场信息, 利用策略, 动态自主地进行决策[23,24].

本文设计的协同电磁对抗实验中, 有红、蓝两组作战单位, 学习算法控制红方作战单位. 在红方视角下, 红方所属的全部作战单位组成目标智能体 $\tau_1$, 蓝方所属的全部作战单位组成外部智能体 $\tau_2$.

不同类型的智能体 (作战单位) 具有不同的观测空间. 例如, 地面远程警戒雷达的探测半径在 400 公里以上, 能够确定多个目标的相对位置、高度等信息; 电子战攻击机可以观察到 50 公里以内的目标, 能够确定多个目标的相对位置、高度、速度、型号、武器挂载等信息. 尽管如何聚合复杂电磁环境下多源异构信息也是一个有意义的研究问题, 但本文聚焦于决策问题, 将聚合每个智能体的观测信息进行了简化. 每个智能体具有 3 种类型的观测, 分别是智能体本身的位置、高度、航向、武器挂载等信息, 记作 $o_s$; 智能体通过自身传感器观察到的其他智能体的位置、高度、航向、武器挂载等信息, 记作 $o_e$; 智能体通过通信网络获得的战场信息, 包括己方感知到战场环境中所有智能体的位置、高度、航向、武器挂载等信息, 记作 $o_g$. 这 3 种类型的观测信息连接起来得到单个智能体的观测信息 $o_i$=[$o_s$, $o_e$, $o_g$]. 特别地, 当智能体的通信遭受干扰后, $o_g$ 会缺失, 当智能体关闭主动雷达后, $o_e$ 会部分缺失, 缺失的部分, 在 $o_i$ 中用特定的占位符表示.

不同类型的智能体 (作战单位) 具有不同的动作空间. 本文将智能体的动作作如下简化. 每个智能体具有两种类型的动作, 一种是改变智能体的航向、航速以及高度的机动动作, 这类动作空间记作 $A_m$; 另一种是控制电磁设备的开关和工作模式, 武器发射等, 这类动作记作 $A_e$. 智能体不采取任何动作, 记作 null, 这也是一种特殊的动作. 智能体 $i$ 的动作空间为 $A_i$=$A_m$∪$A_e$∪{null}.

本文的实验环境为墨子联合作战智能体开发平台[25], 该平台能够模拟各种武器装备, 并提供了面向强化学习的应用程序编程接口. 关于各智能体观测空间、动作空间、模拟作战环境等其他更为具体的内容可参考墨子系统的开发者指南.

本文在墨子作战平台的环境上设计了如下 3 个有代表性的对抗想定.

想定 1. 航空兵突防 (双方力量非对称的协同作战任务). 红方的作战单位有: 两架空空战斗机, 一架空地攻击机, 一架电子支援机, 以及一架预警机. 蓝方的作战单位和军用设施有: 4 辆雷达防空导弹车, 两架空空战斗机, 一个油库. 其中, 每架空空战斗机携带 10 枚对空导弹, 可以打击空中目标; 每架空地攻击机携带 10 枚对地反辐射导弹, 这种导弹可以打击地面雷达设备处于开机状态的作战单位, 携带 4 枚凝固汽油弹, 可以轰炸地面单位或设施; 电子支援机可以对被保护目标实施电子掩护, 降低其被雷达发现、以及被导弹击中的概率; 预警机能够提供远程侦察; 每辆雷达防空导弹车可以移动, 带有 10 枚雷达制导防空导弹, 当导弹车的雷达处于工作模式时, 可以锁定空中目标实施打击, 当其雷达关闭时, 无法打击空中目标.

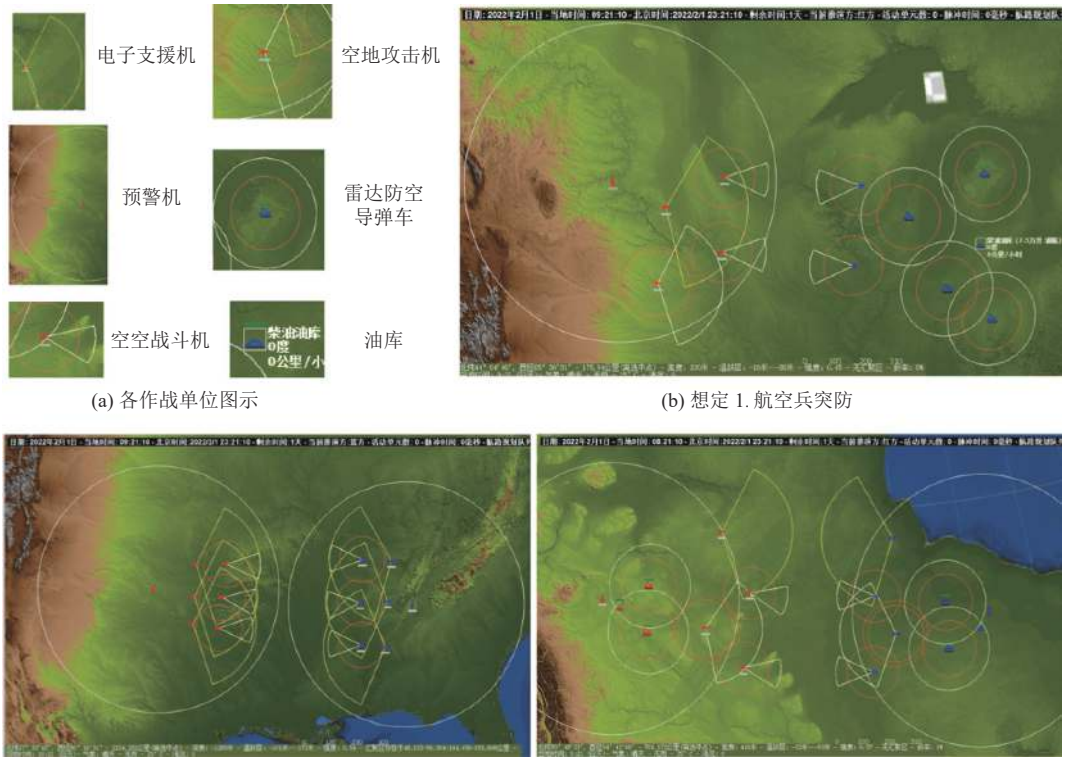
作战单位以及军用设施的价值分别为: 一架空空战斗机价值 10 分; 一架空地攻击机价值 20 分; 一架电子支援机价值 20 分; 一架预警机价值 50 分; 一辆雷达防空导弹车价值 10 分; 油库价值 50 分.

想定 2. 空战对抗 (双方力量对称的协同作战任务). 红蓝双方兵力相同, 分别拥有: 3 架空空战斗机, 3 架电子支援机, 以及 1 架预警机. 各作战单位能力和价值与想定 1 的设定相同.

想定 3. 空地协同作战 (需控制空地、攻防多种类型的作战单位). 红蓝双方兵力相同, 分别拥有: 两架空空战

斗机, 一架空地攻击机, 一架电子支援机, 一架预警机, 两辆雷达防空导弹车, 一个油库. 各作战单位能力与价值和想定 1 的设定相同.

3 个想定的初始兵力分布如图 3 所示. 图中白色圆圈和白色扇形表示该战斗单位的雷达探测范围, 黄色扇形表示电子支援机的掩护范围, 红色圆圈表示该战斗单位的攻击范围.




(a) 各作战单位图示 (b) 想定 1. 航空兵突防

(c) 想定 2. 空战对抗 (d) 想定 3. 空地协同作战

图 3 本文实验中的 3 个协同电磁对抗想定

本文所比较的基线算法包括: IDQN[25]、MADDPG[9]、COMA[2]和 QMIX[4].

### 5.2 实验结果

● 研究问题 1: 学习性能

在评估学习性能的实验中, 蓝方作战单位按照专家设计的行为树进行战斗, 红方作战单位由各学习算法来控制. 实验的奖励信号设置为: 当击毁对方作战单位或设施时, 己方获得该单位或设施对应的价值的分数. 胜负判定规则为: 推演结束时, 得分更高的一方获胜, 双方得分相同则为平局.

图 4(a)–图 4(c) 这 3 个子图, 我们设置了 20 个随机种子, 进行了 50 轮实验, 这 3 幅图展示了这 50 轮实验的平均训练曲线, 纵轴为算法获得的奖励值, 横轴为训练的片段数. 实验结果表明, 在对抗基于专家规则的蓝方智能体时, 本文提出的 ICRA 算法表现出了和 QMIX 相当的效果, 学习速度略慢于 QMIX, 收敛后学习到的红方联合策略所获得的平均奖励和 QMIX 策略的平均奖励值基本相同. 而在想定 3 中, ICRA 算法的学习速度和学习到的策略的平均奖励均优于 QMIX.

在 3 个想定中, ICRA 收敛后的红方联合策略的平均奖励值比 COMA 的策略的平均奖励值分别高出 27%、31% 和 82%; ICRA 能够比 COMA 学习到质量更好的联合策略. 在图 4 (d) 中, 横轴的 1、2 和 3 分别表示想定 1、想定 2 和想定 3. 图 4(d) 展示了 5 种算法所学到的红方策略同蓝方对抗 100 次的胜率. 胜率计算的方式为: $(w+d/2)/n$, 其中 $w$ 为获胜的次数, $d$ 为平局的次数, $n$ 为总次数. 结果表明, ICRA 在胜率方面的表现和 QMIX 相当,

在想定 1 和想定 2 中的胜率略高于 QMIX, 且两者的胜率都远远高于其他算法. 特别地, 在 3 个想定中, ICRA 的胜率比 COMA 的胜率分别高出了 0.41、0.22、0.40.

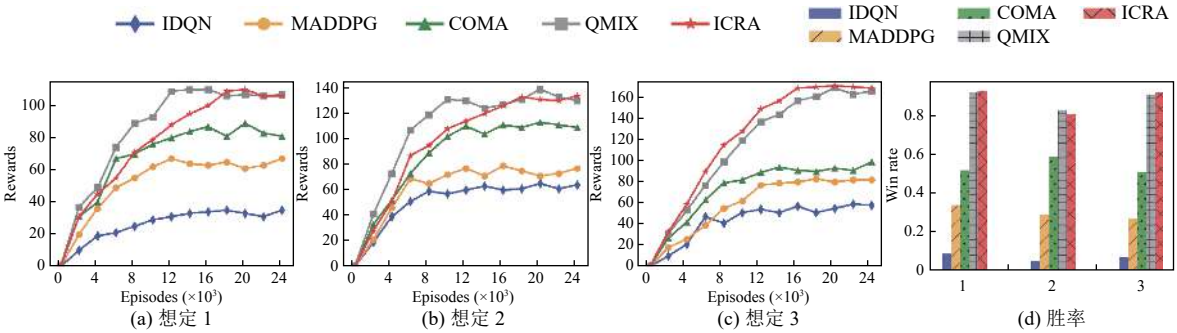



图 4　不同算法在 3 种想定下的训练曲线以及训练后策略的胜率

ICRA 是在 COMA 基础上, 设计了基于 CRA 的策略梯度和基于模仿学习的智能体建模, 以改进 COMA 中优势函数的计算. 而同 COMA 相比, ICRA 有着明显的学习效果的提升, 这表明, 本文所设计的这两个技术有助于提升多智能体强化学习算法的学习性能.

● 研究问题 2: 自博弈效果

为了实验的公平性, 首先将待比较的基线算法扩展为自博弈训练. 扩展的方法是将算法 2 中的 ICRA 算法分别替换为待比较的基线算法. 特别地, 对于算法 2, 其中第 4 行和第 6 行的参数 $c$ 设置为 3 000, 第 3 行的 $H$ 设置为 22, 第 25 行的 $H'$设置为 22. 在实验的 3 个想定中, $H$ 和 $H'$的设置对自博弈方法的效果影响不大, 设置为 20 以上均可.

由于在自博弈过程中, 红蓝双方的策略水平都在不断提升, 训练过程中红方获得的奖励变化并不能表现出其策略水平的变化. 因此, 本节实验中, 在每一轮自博弈训练结束后, 我们将该轮得到的红方策略 $\pi_1$ 同基于专家规则的蓝方策略 $\pi_2$ 对抗 50 次, 得到红方的得分 (即算法 2 中的 $run1(\pi_1, \pi_2)$ 函数). 我们用这个得分来衡量红方策略的水平.

图 5(a)–图 5(c) 这 3 个子图分别展示了在 3 个想定下的自博弈训练过程训练曲线, 我们设置了 20 个随机种子, 进行了 50 次实验, 横轴表示自博弈的轮数 (算法 2 中的 $e$), 纵轴表示红方得分 $run1(\pi_1, \pi_2)$, 图中的曲线是 50 次自博弈的平均结果. 从图 5(a)–图 5(c) 中可以看出, 在 3 个想定中, 基于 ICRA 的自博弈方法收敛后获得的平均奖励值比次优方法得到的平均奖励值分别高出 78%、180% 和 191%.

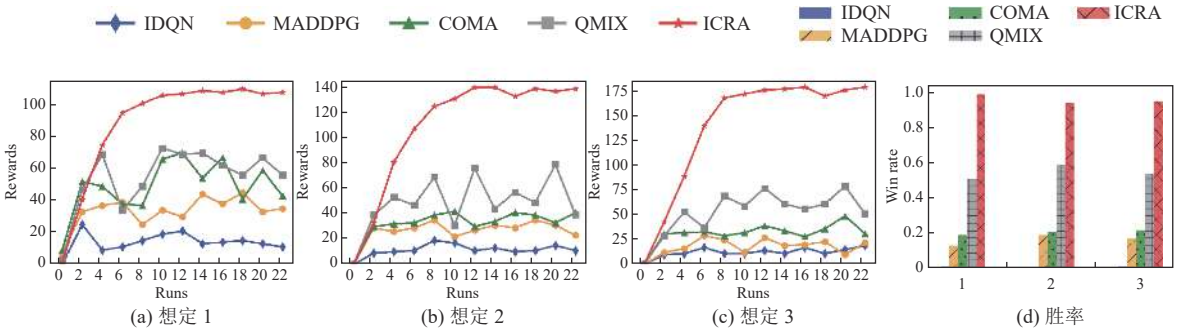



图 5　不同算法在 3 种想定下的自博弈的训练曲线以及训练后策略的胜率

图 5(d) 展示了 5 种方法自博弈训练过程结束后, 同基于规则的蓝方对手对抗 100 次的平均胜率. 图 5 中纵轴表示胜率, 横轴的 1、2 和 3 分别表示想定 1、想定 2 和想定 3. 从图 5(d) 可以看出, 在这 3 个想定中, 基于 ICRA 的自博弈方法比次优方法分别高出了 0.48、0.35、0.41.

上述实验结果表明, 基于 ICRA 的自博弈方法可以在没有任何专家知识的引导下, 学习到高水平的联合策略, 这种方法要远远优于其他方法.

图 6 比较了在想定 2 中, 5 种方法自博弈训练完成后, 交叉战斗 (即任意两个算法训练出的策略分别控制红蓝双方进行对抗) 的胜率. 从图 6 中可以看到, 基于 ICRA 的自博弈方法对抗其他方法的胜率均在 0.7 以上; 基于 QMIX 的自博弈方法在对抗除了基于 ICRA 的自博弈方法外的其他方法也有较高的胜率; 在对抗同样的对手策略时, 基于 ICRA 的自博弈方法的胜率要高于基于 QMIX 的自博弈方法 (例如, QMIX 对抗 IDQN 的胜率约为 0.93, ICRA 对抗 IDQN 的胜率约为 0.98; QMIX 对抗 QMIX 的胜率约为 0.5, ICRA 对抗 QMIX 的胜率约为 0.78). 从交叉战斗胜率的高低来看, 不存在循环克制 (即 A 能胜过 B, B 能胜过 C, C 能胜过 A) 的情况, 5 种方法自博弈训练后的策略水平从高到低分别为: ICRA、QMIX、COMA、MADDPG、IDQN.

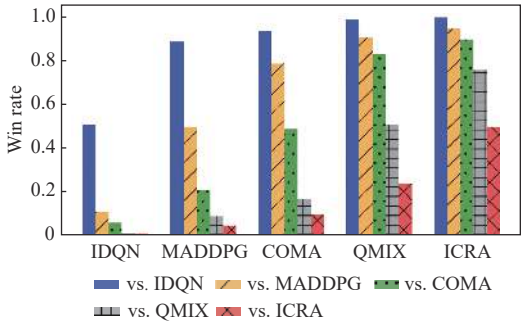
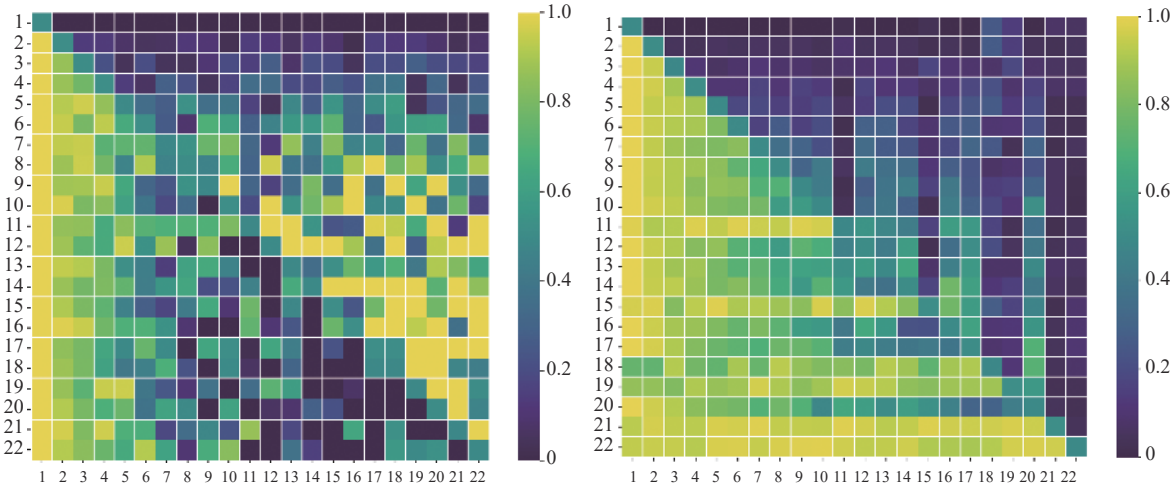



图 6 自博弈训练后交叉战斗的结果

图 7 分别展示了在想定 2 中, 基于 QMIX 和 ICRA 的自博弈方法中红方策略库 K1 的变化情况 (由于想定 2 中, 红蓝双方的兵力完全一样, K2 的变化和 K1 的变化类似). 在这两个交叉战斗热力图中, $i$ 行 $j$ 列的值表示策略 $\pi_i$ 和 $\pi_j$ 对抗 50 轮, $\pi_i$ 的胜率 (胜率计算方式和研究问题 1 中的计算方式相同), 其中 $\pi_i$ 表示自博弈方法在自博弈训练的第 $i$ 轮产生的策略. 在 ICRA 的交叉战斗热力图中, 以矩阵的对角线为分割线, 左下角的胜率普遍高于右上角, 对角线上的胜率约为 0.5. 这表明, 随着自博弈轮数的增加, 其新学习到的策略同之前自博弈轮数学到的策略相比, 水平更高. 而 QMIX 的结果并没有出现这种情况, 靠后自博弈轮数学到的策略的水平经常低于靠前博弈轮数得到的策略的水平; 例如, 在 QMIX 的交叉战斗热力图中的第 14 行第 15 列的值和第 14 行第 22 列的值, 就出现了先学习到的策略 $\pi_{14}$ 在对抗 $\pi_{15}$、$\pi_{22}$ 这两个后学习到的策略时, $\pi_{14}$ 的水平优于后两者 (即 $\pi_{14}$ 的胜率远远高于 0.5). 这一结果再次表明, ICRA 算法能够很好地进行自博弈训练, 并能够不断提升其策略水平; 而缺乏智能体建模的 QMIX 方法在自博弈训练方面表现不佳.




(a) QMIX (b) ICRA

图 7 红方策略库 K1 中策略的交叉战斗结果

● 研究问题 3: 消融实验

本文所提出的方法涉及到如下 3 个重要的设计: (1) 基于 CRA 的策略梯度 (简记为 CRA); (2) 基于模仿学习的智能体建模 (简记为 IM); (3) 自博弈中通过重复剔除严格劣策略得到外部智能体策略库, 并以此训练目标智能体策略 (即算法 2 第 9–27 行, 简记为 Del). 本文所设计的完整的基于 ICRA 的自博弈方法记为 CRA+IM+Del; 在 COMA 的基础上采用了 CRA 、并使用 IM 建模外部智能体的方法记作 CAR+IM; 只是将 COMA 中的反事实基线优势替换为 CRA, 并将外部智能体考虑为环境的方法记作 CRA; 若均不采用 CRA、IM 和 Del 的算法, 就是原始的 COMA 算法.

表 2 给出了基于 ICRA 的自博弈方法及其 3 种变种在 3 个想定中的实验结果. 这 4 种方法按照自博弈进行训练, 自博弈训练结束后, 由学习方法控制红方, 专家规则控制蓝方, 统计红蓝双方 50 次的对抗结果, 得到红方在对抗中的平均奖励值.

表 2　消融实验结果

| 本文的方法及其变体 | 想定1 | 想定2 | 想定3 |
|---|---|---|---|
| COMA | 51.2 | 39.4 | 43.5 |
| CRA | 71.7 | 60.8 | 64.3 |
| CRA+IM | 101.5 | 127.6 | 149.1 |
| CRA+IM+Del | 107.3 | 134.1 | 175.2 |

从消融实验的结果可以看到: (1) 如果把外部智能体考虑为环境, 基于 CRA 的策略梯度 (即表 2 中的 CRA) 能比 COMA 中使用反事实基线优势的策略梯度 (即表 2 中的 COMA), 在 3 个想定中分别有 39.7%, 54.3% 和 32.6% 的效果提升, 这表明虚拟遗憾优势要比反事实基线优势在合作-竞争混合的多智能体环境下, 能够更加准确地估计策略的价值, 更好地指导策略更新; (2) 如果在 CRA 的基础上, 再采用基于模仿学习的方法来应对外部智能体 (即表 2 中的 CRA+IM), 同 COMA 相比, 效果有很大的提升, 平均奖励在 3 个想定中分别提升 98.2%, 223.9% 和 207.4%; (3) 利用剔除严格劣策略的外部智能体策略库来进行训练目标智能体的策略, 该技术也会带来一定的效果增益, 同 CRA+IM 方法相比, CRA+IM+Del 在 3 个想定中分别有 5.7%, 5.1% 和 17.5% 的平均奖励的提升. 因此, 本文所设计的 3 个关键技术均促进了方法的效果提升. 特别地, 将外部智能体和环境显式区分开, 并用智能体建模的方法刻画外部智能体会给自博弈方法带来显著的效果提升.

### 5.3 讨　论

当外部智能体为规则控制的情况下 (这时可以将外部智能体看作环境的一部分), 本文提出的 ICRA 算法较 COMA 的提升很大, 但是同 QMIX 相比, 提升并不明显, 在想定 1 中, QMIX 的表现还略优于 ICRA. 这表明, 在面向完全合作的任务下, 利用基于虚拟遗憾优势的策略梯度比 COMA 的基于反事实基线优势的策略梯度在学习性能方面有提升, 但没有超过值函数分解的方法. 特别地, 想定 1 和想定 2 中的单位类型不多, QMIX 表现的效果较好, 而在想定 3 中, 单位类型更加丰富, ICRA 表现效果较好.

在外部智能体策略会变化的情况下, 由于基于值函数分解的方法很难将对手策略建模到值函数中, 于是 QMIX 在自博弈的场景下很容易陷于次优策略. 而基于虚拟遗憾优势的策略梯度可以比较容易地将对手策略建模到优势函数中, 利用模仿学习对对手进行实时建模, 然后再利用集中式的评价器估计全局状态动作价值函数, 因此 ICRA 能够适应外部智能体策略改变的场景. 特别地, 在表 2 中, 消融实验的结果 (即 CRA 和 CRA+IM) 也清楚地表明了, 显式地建模外部智能体的策略是 ICRA 算法在自博弈场景下的效果提升的关键因素.

## 6 总　结

本文针对合作-竞争混合型多智能体系统, 关注如何为目标智能体生成联合策略. 本文提出了一种基于虚拟遗憾优势的自博弈方法, 该方法结合了合作型多智能体强化学习算法和竞争型自博弈方法, 能够在外部智能体策略未知的情况下, 为目标智能体群体训练出高质量的合作策略. 该方法改进了多智能体强化学习中的优势函数, 提出

了一种更为准确的优势函数——虚拟遗憾优势, 以指导目标智能体进行策略更新; 引入了模仿学习, 以外部智能体的历史决策轨迹作为示教数据, 模仿外部智能体的策略, 由此来显式建模外部智能体; 进而, 本文设计了自博弈方法, 该方法能够克服自博弈训练过程中的非稳态问题, 使得训练过程能持续提升目标智能体的策略水平. 本文在协同电磁对抗的 3 个典型场景下进行实验, 同基线方法相比, 本文所提出的方法在学习性能方面同目前最先进的方法相当, 且在自博弈效果方面显著优于目前最先进的方法.

未来工作包括以下几个方向. 首先, 在很多实际应用中, 需要控制的目标智能体的规模可能非常大, 未来工作将引入诸如图神经网络以及平均场等技术, 将本文的方法进一步扩展到更大规模场景中. 其次, 大规模场景中的决策问题可能会出现明显的分层, 例如, 在协同电磁对抗中, 宏观决策主要考虑以什么顺序打击敌方目标, 微观决策主要考虑如何打击所选定的目标; 本文的方法目前主要考虑电磁对抗中的微观决策, 将分层强化学习同本文的方法结合起来以实现从微观到宏观的全过程决策, 这也是一个值得探索的方向. 最后, 从应用场景来看, 本文目前对于协同电磁对抗的决策问题进行了一些简化, 主要考虑在时域、空域中进行决策; 我们将进一步完善电磁对抗模型, 把决策扩展到时、空、域、能 4 域.

**References:**


[1] Zhang MY, Jin Z, Xu Y, Shen ZH, Liu K, Pan KY. Fast adaptation to external agents via meta imitation counterfactual regret advantage. In: Proc. of the 20th Int'l Conf. on Autonomous Agents and Multiagent Systems, Int'l Foundation for Autonomous Agents and Multiagent Systems, 2021. 1709–1711.

[2] Foerster JN, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conf. and 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI Press, 2018. 363.

[3] Sunehag P, Lever G, Gruslys A, Czarnecki WM, Zambaldi V, Jaderberg M, Lanctot M, Sonnerat N, Leibo JZ, Tuyls K, Graepel T. Value-decomposition networks for cooperative multi-agent learning. arXiv:1706.05296, 2017.

[4] Rashid T, Samvelyan M, de Witt CS, Farquhar G, Foerster JN, Whiteson S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholmsmässan: PMLR, 2018. 4292–4301.

[5] Heinrich J, Lanctot M, Silver D. Fictitious self-play in extensive-form games. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 805–813.

[6] Heinrich J, Silver D. Deep reinforcement learning from self-play in imperfect-information games. arXiv:1603.01121, 2016.

[7] Lanctot M, Zambaldi V, Gruslys A, Lazaridou A, Tuyls K, Pérolat J, Silver D, Graepel T. A unified game-theoretic approach to multiagent reinforcement learning. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 4190–4203.

[8] Zinkevich M, Johanson M, Bowling M, Piccione C. Regret minimization in games with incomplete information. In: Proc. of the 20th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2007. 1729–1736.

[9] Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent Actor-Critic for mixed cooperative-competitive environments. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6382-6393.

[10] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In: Proc. of the 4th Int'l Conf. on Learning Representations. San Juan: ICLR, 2016.

[11] Son K, Kim D, Kang WJ, Hostallero D, Yi Y. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 5887–5896.

[12] Du YL, Han L, Fang M, Dai TH, Liu J, Tao DC. LIIR: Learning individual intrinsic reward in multi-agent reinforcement learning. In: Proc. of the 33rd Inte'l Conf. on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2019. 396.

[13] Bowling M, Veloso M. Multiagent learning using a variable learning rate. Artificial Intelligence, 2002, 136(2): 215–250. [doi: 10.1016/S0004-3702(02)00121-2]

[14] Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, Aru J, Vicente R. Multiagent cooperation and competition with deep reinforcement learning. PLoS One, 2017, 12(4): e0172395. [doi: 10.1371/journal.pone.0172395]

[15] Li X, Jiang XH, Chen YZ, Bao YJ. Game in multiplayer no-limit texas Hold'em based on hands prediction. Chinese Journal of Computers, 2018, 41(1): 47–64 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.00047]

[16] Hu YJ, Gao Y, An B. Online counterfactual regret minimization in repeated imperfect information extensive games. Journal of Computer Research and Development, 2014, 51(10): 2160–2170 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2014.20130823]

[17] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533. [doi: 10.1038/nature14236]

[18] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018.

[19] Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P. High-dimensional continuous control using generalized advantage estimation. In: Proc. of the 4th Int'l Conf. on Learning Representations. San Juan: ICLR, 2016.

[20] Jin PH, Keutzer K, Levine S. Regret minimization for partially observable deep reinforcement learning. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 2342–2351.

[21] Albrecht SV, Stone P. Autonomous agents modelling other agents: A comprehensive survey and open problems. Artificial Intelligence, 2018, 258: 66–95. [doi: 10.1016/j.artint.2018.01.002]

[22] Pomerleau D A. Efficient training of artificial neural networks for autonomous navigation. Neural computation, 1991, 3(1): 88–97. [doi: 10.1162/neco.1991.3.1.88]

[23] Liu SH, Sun GZ. Analysis of the concept and effects of complex electromagnetic environment. Journal of Equipment Academy, 2008, 19(1): 1–5 (in Chinese with English abstract). [doi: 10.3783/j.issn.1673-0127.2008.01.001]

[24] Moreno G, Kinneer C, Pandey A, Garlan D. DARTSim: An exemplar for evaluation and comparison of self-adaptation approaches for smart cyber-physical systems. In: Proc. of the 14th IEEE/ACM Int'l Symp. on Software Engineering for Adaptive and Self-managing Systems (SEAMS). Montreal: IEEE, 2019. 181–187. [doi: 10.1109/SEAMS.2019.00031]

[25] Huashu Defence. Mozi joint intelligence development platform. 2022 (in Chinese). http://www.hs-defense.com/col.jsp?id=124

**附中文参考文献**:

[15] 李翔, 姜晓红, 陈英芝, 包友军. 基于手牌预测的多人无限注德州扑克博弈方法. 计算机学报, 2018, 41(1): 47–64. [doi: 10.11897/SP.J.1016.2018.00047]

[16] 胡裕靖, 高阳, 安波. 不完美信息扩展式博弈中在线虚拟遗憾最小化. 计算机研究与发展, 2014, 51(10): 2160–2170. [doi: 10.7544/issn1000-1239.2014.20130823]

[23] 刘尚合, 孙国至. 复杂电磁环境内涵及效应分析. 装备指挥技术学院学报, 2008, 19(1): 1–5. [doi: 10.3783/j.issn.1673-0127.2008.01.001]

[25] 华成防务. 墨子联合作战智能体开发平台. 2022. http://www.hs-defense.com/col.jsp?id=124

**张明悦**(1994－), 男, 博士, 主要研究领域为软件自适应, 强化学习.

**刘坤**(1996－), 男, 硕士生, CCF 学生会员, 主要研究领域为软件自适应, 因果推断.

**金芝**(1962－), 女, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为需求工程, 知识工程.