doi:10.3772/j.issn.1002-0470.2024.03.004

结合先验知识的多智能体博弈对抗研究①

袁婷帅② 冯 宇 李永强

(浙江工业大学信息工程学院 杭州 310023)

摘 要 无实时奖励的复杂对抗环境是目前深度强化学习(DRL)领域的研究热点,面对此类环境,纯粹使用深度强化学习算法会导致智能体训练无法快速收敛以及对抗效果不佳等问题。基于此,本文提出了一种基于先验知识与深度强化学习相结合的智能博弈流程框架,设计了数据处理、增强机制以及动作决策3个模块,通过威胁评估、任务调度和损失比率3种增强机制来提升智能体在复杂对抗环境下的收敛速度和对抗效果。在数据堡垒(DC)平台上进行仿真,实验结果验证了本文所提出的智能博弈流程框架训练的智能体相较于单纯基于深度强化学习的智能体拥有更快的收敛速度以及更高的胜率。

关键词 智能博弈;先验知识;深度强化学习(DRL);威胁评估;任务调度

近年来,随着科学技术的进步,人工智能(artificial intelligence,AI)的发展迎来了新的机遇,其在智能博弈[1]、天气分类[2]、行为识别[3]、情感分析[4]等各个领域都有了显著的进步。其中智能博弈技术主要是解决人类不能及时有效处理的复杂度高且瞬息万变的对抗问题。对抗作为人类社会发展的主旋律,广泛存在于人与自然以及机器之间,它是人类智能的重要表现形式。智能博弈在对抗问题上的研究主要为游戏和推演平台,如何形成更为合理的决策是其重点[5]。

2016 年 3 月,AlphaGo[6]以 4∶1 的成绩战胜了当时的围棋世界冠军李世石,引起了全世界的关注。这场胜利展示了强化学习在博弈问题中的潜力,掀起了一场人工智能新的研究热潮。人工智能 AlphaStar[7]在游戏竞技中超越 99.8% 以上的人类玩家。世界冠军队伍在面对人工智能 OpenAI Five[8]时表现得不尽人意。Pluribus 在 6 人无限制德州扑克中击败人类职业选手[9]。“觉悟 AI”在 moba 手游中多次击败顶尖职业团队[10]。文献[11]通过自学习在五子棋领域取得远远超越普通人类玩家水准的成绩。

除了游戏领域外,智能博弈在推演平台方面的研究也颇为丰富。文献[12]以微分博弈为框架,研究了单对单的追逃博弈问题。针对智能决策模型问题,文献[13]对比分析了基于专家系统的博弈模型与智能博弈对抗模型。文献[14]提出了基于多智能体深度强化学习的博弈对抗模型。针对复杂对抗环境,文献[15]提出了基于监督学习和深度强化学习相结合的方法,提升了智能博弈的训练效果。针对无人机近距对抗问题,文献[16]提出了基于深度强化学习的近距智能博弈模型。针对多目标智能决策问题,文献[17]基于近端策略优化(proximal policy optimization,PPO)算法设计了包含增强机制的智能决策流程框架。尽管智能博弈在推演平台上的研究成果比较丰富,但由于推演平台本身的连续性和复杂性,大多基于智能博弈的方法难以满足高复杂度以及实时性的需求[18]。

从上述公开发表的成果来看,现阶段的智能对抗算法研究主要针对简单环境和典型动作进行验证,复杂对抗问题系统性研究较为鲜见。针对以上各方法的不足,本文将先验知识与深度强化学习算法相结合,提出了一种结合先验知识的智能博弈流

① 国家自然科学基金(61973276)资助项目。

② 男,1996 年生,硕士生;研究方向:深度强化学习;联系人,E-mail:582619138@qq.com。

(收稿日期:2022-09-24)

程框架,着重设计威胁评估、任务调度以及损失比率模块,进一步提高算法在高复杂环境下的收敛速度及仿真最终成绩。

1 深度强化学习

深度强化学习(deep reinforcement learning, DRL)将深度学习(deep learning, DL)与强化学习(reinforcement learning, RL)相结合,是一种更接近人类思维方式的人工智能方法[19]。深度强化学习不仅突破了传统强化学习方法关于感知问题的诸多限制,还解决了传统深度学习缺乏决策能力的问题,它将两者的优点完美结合,为复杂问题的解决提供了新的思路。

在深度强化学习的诸多算法中,AC(actor critic)[20]是最为常见的算法之一,它将基于值的方法与基于策略的方法结合了起来,具有样本利用率高、价值函数估计准确等优点。然而,在面对具有态势信息庞大,动作组合极其复杂等特点的复杂环境时,AC 算法存在收敛速度慢、高方差以及高维度时难以收敛等问题。为了解决这些问题,A2C(advantage actor critic)[21]算法使用优势函数代替 Critic 网络中的原始回报作为衡量选取动作值和所有动作平均值好坏的指标,这样做的好处是可以让梯度减小,从而使训练过程更稳定、更容易收敛。

如图 1 所示,A2C 算法分为 2 部分:Actor 与 Critic。Actor 基于策略梯度算法,可以根据当前的仿真环境在动作空间内选择动作;Critic会根据仿真环境改变所得到的回报对 Actor 本次选择的动作进行评价;Actor 根据 Critic 做出的评价修改所选动作的概率。

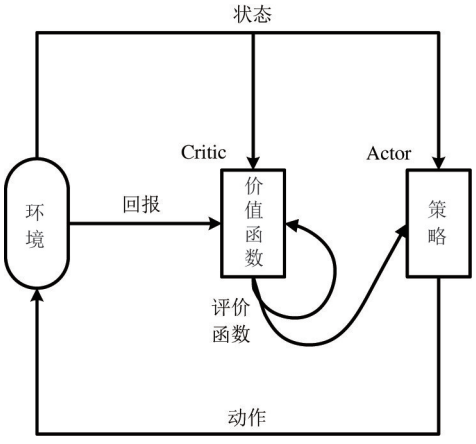

图 1 A2C 算法框架图

Actor 的梯度与优势函数 $Q(s_t, a_t) - V(s_t)$ 有关,其梯度为

$$\nabla J(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\nabla \log \pi_\theta(a_t^n \mid s_t^n)(Q^{\pi_\theta}(s_t^n, a_t^n) - V^{\pi_\theta}(s_t^n)) \tag{1}$$

其中,Q 表示动作值函数,V 表示状态值函数,π 表示策略函数,a 表示动作,s 表示状态,t 表示时间。

Critic 则是根据其估计出的 Q 值与实际的 Q 值之间的平方差进行更新,其损失函数(loss)表达式为

$$loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}(r_t^n + \max Q^{\pi_\theta}(s_{t+1}^n, a_{t+1}^n) - Q^{\pi_\theta}(s_t^n, a_t^n))^2 \tag{2}$$

其中 r 表示奖励函数。

这样的公式会有一个麻烦,那便是需要有 2 个网络分别计算动作值 Q 和状态值 V。以 $r_t + V(s_{t+1})$ 替代 $Q(s_t, a_t)$ 即可解决这个麻烦。于是,策略梯度公式可以进一步改写成:

$$\nabla J(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\nabla \log \pi_\theta(a_t^n \mid s_t^n)(r_t^n + V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n)) \tag{3}$$

相应地,损失函数的表达式变为

$$loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}(r_t^n + V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n))^2 \tag{4}$$

2 多智能体博弈对抗流程设计

传统博弈对抗实验的难点主要包含仿真环境庞大、单位信息复杂、环境本身的不确定性以及对抗的不稳定性等,这些问题都在一定程度上增加了强化学习的应用难度。本文以智能博弈对抗平台为例,设计了运用机器学习方法的数据处理模块、结合先验知识与深度强化学习算法的增强机制模块以及以人为经验为主的动作决策模块来重点解决以上问题。

2.1 总体框架设计

本文的仿真环境具有较高的复杂性,仅靠深度

强化学习算法很难解决此类高复杂度的训练问题。这是由于深度强化学习的训练过程类似黑盒子，在状态、动作以及奖励确定的情况下仿真结果的好坏主要依靠参数的设定，而调参的过程又极其复杂，很难找出最佳的参数配置。为了提升算法收敛速度以及仿真最终成绩，本文结合先验知识与深度强化学习算法，设计了数据处理模块、增强机制模块、动作决策模块，具体总体流程框架如图 2 所示。

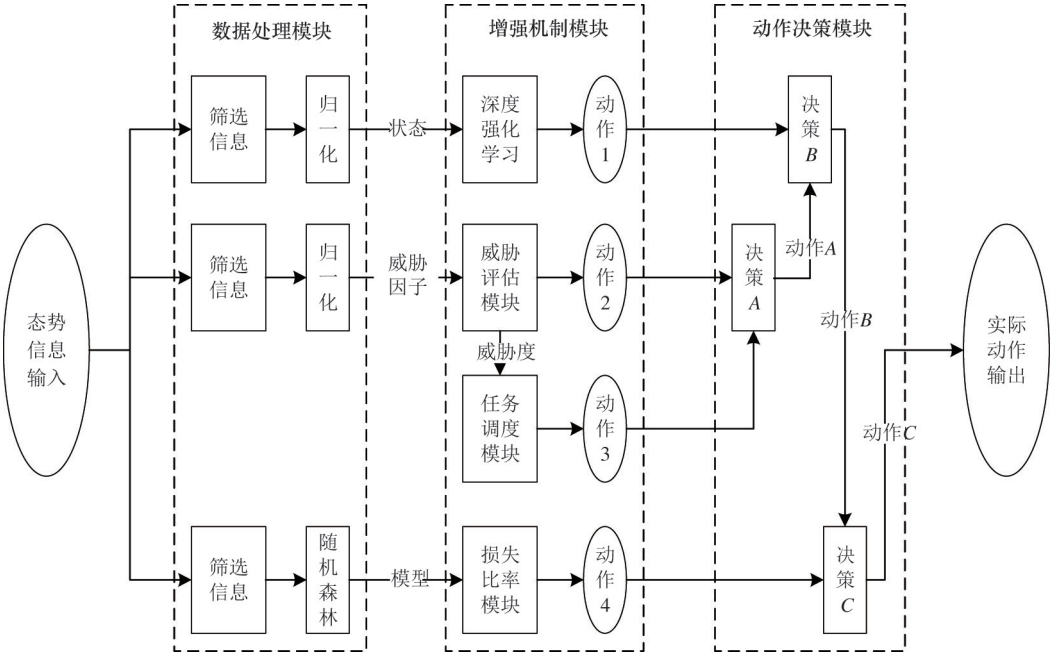

图 2　总体流程框架图

其中态势输入的信息是仿真环境中单方视角下所有单位的具体信息，包括各单位的位置、速度、类别、编号以及所属方等信息。

数据处理模块的作用是筛选出所需信息，并输出增强机制模块所需的参数或模型。由于各个支路的功能不同，因此所筛选的信息也不同；归一化是对筛选过后的态势信息进行处理得到 0 ~ 1 之间的数，从而向深度强化学习模块输入所需的状态信息以及向威胁评估模块输入威胁因子；随机森林是训练筛选数据的工具，该模块根据大量的筛选信息训练出可以评估战局的模型，并将此模型输入到损失比率模块。

增强机制模块是整个框架的核心，其目的是将先验知识与深度强化学习结合起来，进而提升深度强化学习算法的收敛速度以及仿真最终成绩。增强机制模块包括深度强化学习模块、威胁评估模块、任务调度模块以及损失比率模块。(1) 深度强化学习模块类似于黑盒子，它基于 A2C 算法根据输入的状态输出动作 1；(2) 为使算法更快的收敛，威胁评估模块将数据处理模块输出的威胁因子经处理后得到对应的威胁值，之后按照概率输出动作 2；(3) 为提高算法的最终成绩，任务调度模块按照规则输出动作 3；(4) 损失比率模块意在从整体战局上进行分析，进一步提升算法的性能，它可以根据随机森林训练出的评估战局模型输出动作 4。最终这些动作会输出到动作决策模块。具体内容在 2.2 ~ 2.4 节会详细介绍。

动作决策模块的功能在于对增强机制模块输出的 4 个动作进行决策，找出最佳动作，并将其输出至实际动作输出模块。首先对增强机制模块中输出的动作 2、3 通过决策 A 进行优先度判定，输出动作 A；然后根据决策 B 对动作 1 和动作 A 进行位置决策，输出动作 B；最后按照决策 C 对战局结果的评估对动作 4 和动作 B 进行决策，得到动作 C 并输出至动作输出模块。

实际动作输出模块根据动作决策模块输出的动作 C 发送仿真指令，输出相应的动作。

2.2 威胁评估模块

面对复杂多变的仿真环境,深度强化学习算法可能会出现收敛速度过慢的问题。本文采用深度强化学习与威胁评估相结合的方法来解决此问题。

威胁评估是基于当前态势推理目标单位对我方单位的威胁程度,它是进行情报收集信息处理后,作出行动安排的前提和核心[22]。威胁评估一般分为3个步骤,首先确定对己方构成威胁的威胁评估因子,然后通过威胁因子构建威胁评估模型,最后根据模型的值评判威胁度大小。

本文首先对所获取的己方与目标双方的作战能力、航向夹角、飞行速度以及相对距离进行数据转换,得到相应的能量威胁指数 ρ、角度威胁指数 θ、距离威胁指数 μ 以及速度威胁指数 v; 然后对这4项威胁指数进行归一化,将其转化为所需的数据格式;最后根据威胁评估公式即可得到相应的威胁度值。具体的威胁评估公式为

$$w = a\rho + b\theta + c\mu + dv \tag{5}$$

其中, w 表示威胁度大小; a、b、c、d 各威胁指数的权重系数,且 $a + b + c + d = 1$。

在上述威胁因子的基础上,本文引入目标单位所配备装备的状态作为评估威胁度的另一指标,将其归一化得到装备威胁指数 γ。结合 γ 威胁评估公式如式(6)所示。

$$w = \alpha(a\rho + b\theta + c\mu + dv) + \beta\gamma \tag{6}$$

其中 α、β 表示权重系数,且 $\alpha + \beta = 1$。

由式(6)可得到各个单位的威胁大小,然后以目标单位与己方单位的距离为指标,根据威胁大小以及距离指标设置不同的行动命令。

2.3 任务调度模块

加入威胁评估模块之后的仿真效果虽然比之前有了不小的提升,但是在复杂的仿真环境中难免会有威胁评估不适用的情况。如当某个目标对己方总体威胁不大,但是对指挥所威胁程度较高时,威胁评估模块可能不会对此目标重点打击。因此本文引入了任务调度模块。

在威胁评估的基础上,任务调度可根据目标的优先度对己方单位进行相应的行为调度,直接影响仿真的实时性能。该优先度可通过目标威胁指数 θ_p 与双方加权距离 d_θ 计算得到。由于仿真实验中红蓝双方单位种类、数量以及各自的任务目标不同,因此需对它们分别进行调度设计,具体信息见3.3.2节。

2.4 损失比率模块

在各种对战游戏中,有些可以直接获取对战分数,如超级马里奥、雷霆战机等,这类游戏每次击败敌方单位或者获得道具都会有分数奖励,拥有实时分数奖励的游戏对于强化学习来说比较适合训练;有些则无法直接获取对战分数,如刀塔(defemse pf amcoemts,DOTA)、星际争霸、英雄联盟(league of legends,LOL)等,这类游戏只能在对局结束时判定胜负,这类只靠稀疏奖励的游戏对于强化学习的训练比较困难。

本文选择的仿真平台与上述第2类游戏类似,即只有稀疏战局奖励而没有实时分数奖励。为此本文改进了奖励函数,具体见3.2.3节。

考虑到仿真平台无法直接根据当前态势信息预估胜负结果,因此本文设计了损失比率模块。具体内容为:首先利用数据处理模块筛选的数据(对战结束时的各单位的剩余数量、状态以及对战胜负信息),通过机器学习中的随机森林[23]算法进行训练;然后损失比率模块通过数据处理模块训练好的模型判断当前态势的优劣;最后根据判断结果做出相应的动作,并将其输出到动作决策模块。

3 仿真模型及具体设置

3.1 仿真介绍

本文的仿真平台为数据保垒(DataCastle,DC)竞赛推出的智能博弈对抗平台,其详细内容可查阅智能博弈挑战赛白皮书[24],这里主要介绍想定的目标、单位配置以及胜负条件。

想定目标:蓝方目标作为防守方,其目的是守卫己方岛屿2个指挥所重点目标;红方目标作为进攻方,其目的是摧毁蓝方2个指挥所重点目标。

单位配置:红方设置7种兵力共45个作战单位;蓝方设置7种兵力共30个作战单位。具体兵力设置如表1所示。

胜负条件:在比赛时间(150 min,该时间为仿真时间,非实际时间)内,若2个守卫目标均被摧毁,

表 1　想定单位配置表

作战单位	任务	红方数量	蓝方数量
轰炸机	空中突击	16 架	8 架
预警机	空海探测	1 架	1 架
无人侦察机	空中侦察	3 架	无
电子战飞机	干扰压制	1 架	无
歼击机	掩护护航	20 架	12 架
防空驱逐舰	舰艇防空	2 艘	1 艘
地面雷达	对空探测	1 部	2 部
地导营	地面防空	无	3 部
机场	支援保障	1 个	1 个
指挥所	重点目标	无	2 个

判红方胜;2 个守卫目标均未被摧毁,判定蓝方胜;达到作战时间时,若仅摧毁一个指挥所,则根据双方的损失比率进行判断。

3.2　模型构建

3.2.1　状态空间

此仿真平台的态势信息十分复杂,对战双方的态势信息包括机场信息、编队信息、单位信息、情报信息以及敌方发射导弹信息 5 大类。其中每类信息包含多个子信息,而每个子信息又分成各种具体信息。以单位信息为例,它包含全部的己方单位信息,每个单位信息又细分为类型、位置、速度、航向、弹药等信息。若将全部信息放入状态空间会导致维度灾难,因此本文提取了其中部分信息作为状态。

对此,本文对仿真平台的态势信息进行处理,选取单位信息和情报信息的部分信息构成状态空间。其中单位信息作为第 1 部分的状态信息,情报信息作为第 2 部分的状态信息,并分别为两者预留 50 个数据存储地址。如果状态空间某些信息不存在(如情报信息中不足 50 单位的信息),则对相应的信息进行置 0 处理,具体见表 2。

表 2　状态空间信息表

信息	空间	具体内容
单位	50	编号、坐标、速度、航向、军别等
情报	50	编号、坐标、速度、航向、军别等

3.2.2　动作空间

仿真平台的指令信息复杂,仅飞机指令就可细分为起飞、区域巡逻、航线巡逻、目标突击、返航等指令。以飞机的区域巡逻指令为例,该指令不仅要设置巡逻数量、巡逻方式,还要设置巡逻区域等信息。由于巡逻区域的无穷性,这样动作的数量理论上也是无穷的。

本文针对上诉情况,进行如下操作:首先对战场区域进行划分,将无限区域分割为 $(n \times m)$ 个子区域,从而降低学习的压力;然后对飞行单位进行编组,并对所有编组进行指令整合,使其由多个指令集合成为一个动作,大幅度精简了动作空间,从而提升智能体的训练速度。

在具体设计动作时双方会有一些区别。红方目标是攻占蓝岛,设置每个编队在决策时刻都有 2 个动作可以选择,即进攻蓝南岛和进攻蓝北岛,其动作空间设置为 2^{10} 组动作。蓝方目标是守护岛屿,设置每个编队在决策时刻有 $(n \times m)$ 个动作可以选择,即去往目标子区域进行防御,其动作空间应为 $(n \times m)^7$ 组动作,由于其数量级过大,因此这里随机选取其中的部分动作作为蓝方的动作空间。具体信息见表 3。

表 3　动作空间信息表

属方	编队数	每个编队的具体动作
红方	10	2 个:进攻南岛;进攻北岛
蓝方	7	$n \times m$ 个:选取防御区域进行防御

3.2.3　奖励函数

前面说过,本文所用的仿真平台无法直接获取对战分数,只能在对局结束时评定胜负。只靠对局结束的战局奖励对算法进行训练很难有较好的效果,因此本文对奖励函数进行了改进,设计了稀疏战局奖励和嵌入式奖励两部分内容。

(1)稀疏战局奖励主要分为对战结束奖励和实时击毁奖励,其中对战结束奖励是指在对战结束时根据胜负状态给出一个权重较大的奖励;实时击毁奖励是指对战过程中己方每击毁一个敌方单位给出一个权重较小的奖励。

(2)嵌入式奖励将奖励与仿真运行时间结合。由于仿真平台中红蓝双方的作战单位的数量、类型、

任务的不同,因此在设计嵌入式奖励函数时会有一些区别:红方奖励随着仿真运行时间的推移而减小,即越快的时间结束比赛获得的奖励越高;蓝方奖励则随着仿真运行时间的推移而增大,即坚持时间越久获得的奖励就越高。

3.3 增强机制具体方法

3.3.1 威胁评估

威胁评估模块的各个威胁因子[25]的计算方式如下所述。

能量威胁指数 ρ 根据敌方单位的类型直接给出,数值为 0~1 之间的浮点数。

角度威胁指数 θ 根据敌我双方的航向角得出,具体公式如式(7)所示。

$$\theta = \left(\left|\alpha_1 - \frac{y_1 - y_2}{x_1 - x_2}\right| + \left|\alpha_2 - \frac{y_1 - y_2}{x_1 - x_2}\right|\right)/360 \tag{7}$$

其中,α_1 表示己方航向角,α_2 表示敌方航向角,(x_1, y_1) 表示己方位置信息,(x_2, y_2) 表示敌方位置信息。

根据双方的相对距离 l 得出距离威胁指数 μ,具体公式为

$$\mu = \begin{cases} 1 & l < d_n \\ 0.8 & d_n < l \leqslant d_f \\ 0.5 & d_f < l \leqslant d_z \\ \dfrac{10000}{l(l+100)} & d_z < l \end{cases} \tag{8}$$

其中,l 表示双方的空间距离,d_n 表示自动攻击距离,d_f 表示最大攻击范围,d_z 表示单位探测范围。

速度威胁指数 υ 根据双方的速度得出,具体公式为

$$\upsilon = \begin{cases} 1 & 1.5v_1 < v_2 \\ -0.5 + \dfrac{v_2}{v_1} & 0.6v_1 < v_2 \leqslant 1.5v_1 \\ 0.1 & v_2 < 0.6v_1 \end{cases} \tag{9}$$

其中,v_1 表示己方单位的速度,v_2 表示敌方单位的速度。

装备威胁指数 γ 根据敌方装备的状态以及与己方的距离得出,具体公式如式(10)所示。

$$\gamma = \begin{cases} 1 & l \leqslant r_h \\ 0.8 & r_h < l \leqslant r_l \\ 0.5 & r_l < l \\ 0 & \text{无装备} \end{cases} \tag{10}$$

其中,r_h 表示高威距离,r_l 表示低威距离。

设置威胁评估式(10)中参数 α、β、a、b、c、d 分别为 0.60、0.40、0.45、0.27、0.20、0.08,然后将上述威胁指数代入此公式得出相应的威胁度值。得到敌方单位的威胁度大小后,进行如下操作:当敌方目标与己方单位的距离大于 $l_{\max}$ 的时候,仿真平台会基于 AC 训练的动作对己方单位进行任务派遣;当敌方目标与己方单位的距离小于 $l_{\min}$ 的时候,仿真平台会基于威胁评估的结果进行分析。根据权重选择对己方单位威胁前 3 的敌方目标发动攻击指令,其中威胁前 3 的权重分别为 0.5、0.3、0.2;当敌方目标与己方单位的距离介于 $l_{\min}$ 与 $l_{\max}$ 之间的时候,仿真平台会按照概率进行任务调度,即 80% 的概率按照 AC 训练的动作行动、20% 的概率基于威胁评估的结果行动。这里的 $l_{\max}$ 设置为 100,$l_{\min}$ 设置为 80。

3.3.2 任务调度

红蓝双方任务调度的具体内容如下所述。

对于红方:首先需计算目标单位对己方单位的优先度 y,并对其进行排序,找出最大的优先度 $y_{\max}$;然后对 $y_{\max}$ 进行判定,若该值大于 λ,则下令己方单位攻击该目标单位,若该值小于 λ,则直接进入动作决策模块,对威胁评估模块以及深度强化学习模块输出的动作 1 与动作 2 进行动作决策,输出动作 B。设置 λ 为 0.5。

对于蓝方:首先对目标单位与己方核心单位进行距离判定。若距离小于敌方最大攻击范围 l_b,则下令己方单位攻击该目标单位;若距离不小于 l_b,则直接进入动作决策模块,对威胁评估模块以及深度强化学习模块输出的动作 1 与动作 2 进行动作决策,输出动作 B。这里的 l_b 设置为 80。

3.3.3 损失比率

通过随机森林得到最终的模型预测结果,损失比率模块会根据预测结果指导智能体做出合适的动作。由于仿真平台中红蓝双方的参数不同,因此在

结合损失比率时双方会有一些区别,具体内容如下。

当模型的测试结果为胜利时输出动作 4,动作决策模块输出动作 C,红方最终动作输出为飞行单位全体返航,蓝方最终动作输出为全体轰炸机返航;当模型的测试结果为失败时,动作决策模块会根据前 3 个模块(深度强化学习模块、威胁评估模块、任务调度模块)输出的动作 1、2、3 进行决策,输出决策动作。由于红方在模型判定为优势时下达全员撤离命令,导致红方部分被攻击单位可能无法安全撤离,因此红方在此模块会有一定负面影响。

4　仿真结果

智能博弈对抗平台官方提供了红蓝双方的内置规则,可用于仿真效果的比较。为了验证仿真的效果,本文将仿真结果分成了 5 个模块:内置规则的仿真结果、基于 A2C 的仿真结果、结合威胁评估的仿真结果、结合任务调度的仿真结果以及结合损失比率的仿真结果。

首先,本文对红方内置规则与蓝方内置规则进行相互对抗,其目的是为了后续的仿真提供一个可以对比的标准。仿真轮次为 500 次,其对战结果为红方胜率为 67%、蓝方胜率为 33%。具体对抗情况如图 3 所示。

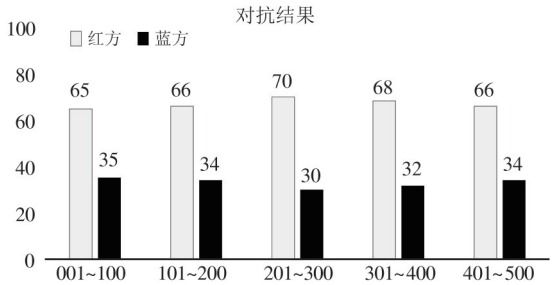

图 3　内置规则对战结果

然后,本文以内置规则的对战数据为基准,与增强机制模块训练的双方智能体训练数据进行比对,红方对比结果见图 4、蓝方对比结果见图 5。

由图 4 和 5 中的内置规则与 A2C 对比可以看出,随着训练场次的不断增加,红方智能体与蓝方智能体的胜率都在不断提升,且最终收敛下的胜率都超过了初始的内置规则胜率。由A2C与威胁评估

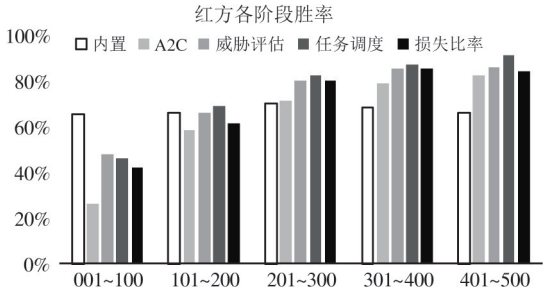

图 4　红方智能体训练效果图

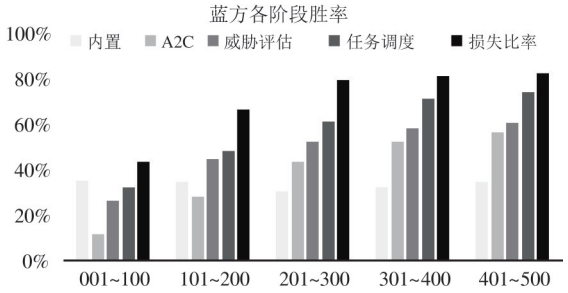

图 5　红方智能体训练效果图

对比可以看出,相比于纯粹用 A2C 算法进行训练,结合威胁评估后的算法在训练初始胜率以及收敛速度上有了明显的提升。任务调度的训练效果和前面的 3 个模块进行对比可以看出,相比于纯粹用 A2C 算法以及结合威胁评估的算法,结合任务调度后的算法在训练收敛后的胜率上有了较大提升。

对图 4 和图 5 整体分析,不难发现结合损失比率的智能体只需 300 轮次即可基本收敛,相比于前面需要训练 400 ~ 500 轮次收敛的智能体,其在收敛速度上有了较大提升。

对训练完成的各模块模型进行测试,测试方式为将保存的红(蓝)方模型与蓝(红)方的内置规则进行 100 轮次的对战,并统计红(蓝)方的胜场。测试结果如图 6 所示。

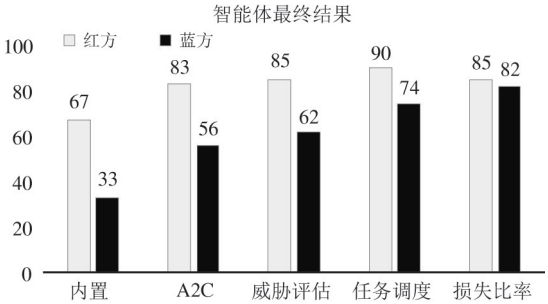

图 6　智能体最终效果图

图6显示了红蓝双方各个智能体在对战内置规则的胜场情况。由图6可以看出,红方智能体的最高胜率由67%提升到了90%,提高了23%;蓝方智能体的最高胜率由33%提升到了82%,提升了49%。从结果来看,本文新提出的结合先验知识的A2C智能体不仅在最终效果上远超基于规则的内置智能体,其在收敛速度上也优于普通的A2C智能体。

5 结 论

本文针对复杂环境下的智能博弈问题,提出了一种结合先验知识与深度强化学习的方法。该方法首先对仿真环境的态势信息进行筛选并进行归一化处理;然后设计了威胁评估、任务调度、损失比率等增强机制;最后根据动作决策模块输出最终动作,提升了算法的收敛速度以及仿真最终成绩,解决了仿真环境态势过于复杂、动作空间过大以及稀疏奖励等问题。由仿真结果可以看出,本文新提出的结合先验知识的A2C智能体不仅在最终效果上超越普通的A2C智能体,其在收敛速度上也优于普通的A2C智能体,红蓝双方相比内置的规则智能体胜率都有了显著的提高。仿真结果验证了本文所提方法的可行性,并展现出了此方法在处理复杂环境下的智能博弈问题的巨大潜力。

由于本文对态势提取的状态空间压缩较大,并且对动作空间做了诸多限制,未来会考虑通过升级硬件设备逐步放开对状态、动作空间的限制。此外,本文的重点在于如何处理复杂环境下的智能博弈问题,验证各种增强技术的可行性,后续的研究可以在本文的方法框架下结合不同算法,引入自我博弈机制使2个智能体同步提高,还可以尝试将本方法应用于更为复杂的环境。

参考文献

[1] 孙宇祥,彭益辉,李斌,等. 智能博弈综述:游戏AI对作战推演的启示[J]. 智能科学与技术学报,2022,4(2):157-173.

[2] 陈思玮,贾克斌,王聪聪,等. 深度学习在多天气分类算法中的研究与应用[J]. 高技术通讯,2020,30(10):1010-1017.

[3] 赵新秋,杨冬冬,贺海龙,等. 基于深度学习的人体行为识别研究[J]. 高技术通讯,2020,30(5):471-479.

[4] 刘文远,郭智存,郭丁丁. 旅游场景下的基于深度学习的文本方面级细粒度情感分类[J]. 高技术通讯,2022,32(1):22-32.

[5] 吴曦,孟祥林,杨镜宇. 下一代战略博弈推演系统研究[J]. 系统仿真学报,2021,33(9):2017-2024.

[6] 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.

[7] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019,575(7782):350-354.

[8] BERNER C, BROCKMAN G, CHAN B, et al. Dota 2 with large scale deep reinforcement learning[EB/OL]. (2019-12-13) [2022-09-30]. https://arxiv.org/pdf/1912.06680.pdf.

[9] BROWN N, SANDHOLM T. Superhuman AI for multiplayerpoker[J]. Science, 2019,365(6456):885-890.

[10] YE D H, CHEN G B, ZHAO P L, et al. Supervised learning achieves human-level performance in MOBA games: a case study of honor of kings[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022,33(3):908-918.

[11] 李大舟,沈雪雁,高巍,等. 一种自学习的智能五子棋算法的设计与实现[J]. 小型微型计算机系统,2020,41(6):1169-1175.

[12] WEINTRAUB I E, PACHTER M, GARCIA E. An introduction to pursuit-evasion differential games[EB/OL]. (2020-03-10) [2022-09-30]. https://arxiv.org/pdf/2003.05013v1.pdf.

[13] 陈希亮,李清伟,孙彧. 基于博弈对抗的空战智能决策关键技术[J]. 指挥信息系统与技术,2021,12(2):1-6.

[14] 孙彧,李清伟,徐志雄,等. 基于多智能体深度强化学习的空战博弈对抗策略训练模型[J]. 指挥信息系统与技术,2021,12(2):16-20.

[15] 张振,黄炎焱,张永亮,等. 基于近端策略优化的作战实体博弈对抗算法[J]. 南京理工大学学报,2021,45(1):77-83.

[16] 周攀,黄江涛,章胜,等. 基于深度强化学习的智能空战决策与仿真研究[J]. 航空学报,2022,43(12):1-

16.

[17] 施伟,冯旸赫,程光权,等. 基于深度强化学习的多机协同空战方法研究[J]. 自动化学报,2021,47(7):1610-1623.

[18] 孙智孝,杨晟琦,朴海音,等. 未来智能空战发展综述[J]. 航空学报,2021,42(8):35-49.

[19] 赵星宇,丁世飞. 深度强化学习研究综述[J]. 计算机科学,2018,45(7):1-6.

[20] KONDA V R, TSITSIKLIS J N. On Actor-Critic algorithms[J]. SIAM Journal on Control and Optimization, 2003,42(4):1143-1166.

[21] PENG B, LI X, GAO J, et al. Adversarial advantage Actor-Critic model for task-completion dialogue policy learning[C]// IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, Canada: IEEE, 2018:6149-6153.

[22] 王贝,杨世荣,钮小林. 基于不确定语言多属性决策的目标威胁度排序[J]. 火力与指挥控制,2008,33(s2):169-171.

[23] PETER H. 机器学习实战[M]. 李锐,李鹏,曲亚东,等译. 北京:人民邮电出版社,2013.

[24] DataCastle. 智能博弈挑战赛白皮书[EB/OL]. [2022-09-30]. https://js.dclab.run/v2/cmptDetail.html?id=377.

[25] 雷蕾,尚丽娜,张列航. 空战目标威胁排序与目标分配算法[J]. 电光与控制,2010,17(4):38-40,82.

Research on multi-agent game confrontation combined with prior knowledge

YUAN Tingshuai, FENG Yu, LI Yongqiang

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

The complex adversarial environment without real-time reward is the current research hot spot in the field of deep reinforcement learning(DRL). In such environment, the use of deep reinforcement learning algorithm alone in general leads to a lower convergence speed and unsatisfactory performance. In this regard, this paper proposes an intelligent game process framework based on the combination of prior knowledge and deep reinforcement learning, and designs three modules of data processing, enhancement mechanism and action decision-making to improve both the convergence speed and the countermeasure effect under complex confrontation environment through three enhancement mechanisms including threat assessment, task scheduling and loss ratio. The simulation results on the DataCastle (DC) platform show that the agent trained by the proposed intelligent game process framework has a fast convergence speed and higher winning rate than the agent only based on deep reinforcement learning.

Key words: intelligent game, prior knowledge, deep reinforcement learning(DRL), threat estimation, task dispatch