



基于多智能体协同强化学习的多目标追踪方法

王毅然¹, 经小川^{1,2}, 贾福凯², 孙宇健², 佟 轶²

(1. 中国航天系统科学与工程研究院, 北京 100048; 2. 航天宏康智能科技(北京)有限公司, 北京 100048)

摘 要: 针对现有多目标追踪方法通常存在学习速度慢、追踪效率低及协同追踪策略设计困难等问题, 提出一种改进的多目标追踪方法。基于追踪智能体和目标智能体数量及其环境信息建立任务分配模型, 运用匈牙利算法根据距离效益矩阵对其进行求解得到多个追踪智能体的任务分配情况, 并以缩短目标智能体的追踪路径为优化目标进行任务分工, 同时利用多智能体协同强化学习算法使多个智能体在相同环境中不断重复执行探索-积累-学习-决策过程, 最终根据经验数据更新策略完成多目标追踪任务。仿真结果表明, 与 DDPG 和 MADDPG 方法相比, 该方法能在避免碰撞和躲避障碍物的情况下, 使多个智能体通过相互协作形成针对多个运动目标的最短追踪路线。

关键词: 多智能体; 多目标追踪; 强化学习; 任务分配; 实时性

开放科学(资源服务)标志码(OSID):



中文引用格式: 王毅然, 经小川, 贾福凯, 等. 基于多智能体协同强化学习的多目标追踪方法[J]. 计算机工程, 2020, 46(11): 90-96.

英文引用格式: WANG Yiran, JING Xiaochuan, JIA Fukai, et al. Multi-target tracking method based on multi-agent collaborative reinforcement learning[J]. Computer Engineering, 2020, 46(11): 90-96.

Multi-Target Tracking Method Based on Multi-Agent Collaborative Reinforcement Learning

WANG Yiran¹, JING Xiaochuan^{1,2}, JIA Fukai², SUN Yujian², TONG Yi²

(1. China Aerospace Academy of Systems Science and Engineering, Beijing 100048, China;

2. Aerospace Hongkang Intelligent Technology(Beijing) Co., Ltd., Beijing 100048, China)

[Abstract] There are multiple problems with existing multi-target tracking methods, including low learning speed, inefficient tracking and high difficulty in collaborative tracking strategy design. To this end, this paper proposes an improved multi-target tracking method. The method builds a task assignment model based on the number of target agents and tracking agents and their environmental information. Then the model is solved by using Hungary algorithm according to the distance benefit matrix to acquire the task assignment information of multiple tracking agents, which is optimized to shorten the tracking paths of target agents. In addition, the multi-agent collaborative reinforcement learning algorithm is used to enable multiple agents to repeat the process of exploration-accumulation-learning-decision in the same environment and update the strategy based on empirical data to finally complete the multi-target tracking task. Simulation results show that compared with DDPG and MADDPG methods, the proposed method enables multiple agents to collaboratively form the shortest path for tracking multiple moving targets with collisions and obstacles avoided.

[Key words] multi-agent; multi-target tracking; reinforcement learning; task assignment; real-time

DOI: 10.19678/j.issn.1000-3428.0055904

0 概述

随着人工智能技术的不断发展, 多智能体协同控制在军事应用中取得了重大突破^[1-2], 以无人机、无人车^[3]和无人水面艇等为代表的无人智能体在执

行军事作战中的侦察、护航、打击等任务时^[4-6]通常以追踪问题为基础开展研究。而在现代战场环境下, 由于任务和环境的复杂性, 一般需要多个作战智能体协同完成对多个动态运动目标的追踪任务, 因此智能体面对动态变化的战场态势, 如何进行任务

基金项目: 广东省应用型科技研发基金(2016B010127005)。

作者简介: 王毅然(1994—), 男, 硕士研究生, 主研方向为目标跟踪、多智能体; 经小川, 研究员; 贾福凯、孙宇健、佟 轶, 工程师。

收稿日期: 2019-09-03 **修回日期:** 2019-11-11 **E-mail:** wangyr_caec@163.com

分工及采取何种行动策略将会影响智能体的作战质量和作战效率。

针对多目标追踪问题, 学者们进行了大量研究并取得一定的成果。文献[7]提出一种合作团队追踪单一运动目标的方法, 针对运动目标位置估计的不确定性, 最大限度地缩小目标可到达空间。仿真结果表明, 在位置不确定的情况下, 该方法能通过追踪智能体捕获目标。文献[8]提出基于轨迹集和随机有限集的多目标追踪问题求解方法, 通过多对象密度函数确定测量值的贝叶斯轨迹分布, 其中包含所有轨迹的信息。

强化学习主要解决智能决策问题, 其目前在单智能体决策领域取得了较大成功, 如 AlphaGo、AlphaGo Zero 等。针对团队最优决策问题, 学者们主要通过基于价值函数和概率这两种方法将单智能体强化学习扩展到多智能体强化学习。文献[9-11]基于价值函数的方法, 采用 Q-learning、DQN 和 IQL 算法并结合奖励函数, 仿真模拟了在完全协作、完全竞争及非完全协作/竞争环境下多个智能体的性能表现。但是, 当环境较复杂及智能体规模较大时, 上述算法的稳定性和可扩展性较差, 且难以应对较大的连续动作空间, 无法输出离散状态动作值。文献[12-14]基于概率的方法提出深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)算法, 在动作输出方面通过网络拟合策略函数并直接输出动作值, 可应对更大的动作空间以及连续动作的输出。针对上述方法存在的学习时间长、实时性差等问题, 本文提出一种基于多智能体协同强化学习的多目标追踪方法(Multi-Target Tracking method based on Multi-Agent Collaborative Reinforcement Learning, MTT-MACRL)。

1 多目标追踪问题

1.1 问题描述

多目标追踪问题涉及追踪和目标智能体两方面, 其主要的研究目标为多个自主追踪智能体的协同追踪策略^[15-17]。目前, 关于追踪问题的描述不一, 本文将追踪问题定义为: 假设在相同的有限二维空间内存在 n_p 个追踪智能体, 则追踪智能体集合 $P = \{P_1, P_2, \dots, P_{n_p}\}$, 假设在相同有限二维空间内存在 n_e 个目标智能体, 则目标智能体集合 $E = \{E_1, E_2, \dots, E_{n_e}\}$, 追踪智能体和目标智能体统称为智能体 $A, A = P \cup E$ 。 $O_{P_i} (i = 1, 2, \dots, n_p)$ 代表追踪智能体 P_i 的中心, $O_{E_j} (j = 1, 2, \dots, n_e)$ 代表目标智能体 E_j 的中心, $V_{P_i} (i = 1, 2, \dots, n_p)$ 代表追踪智能体 P_i 的运动速度, $V_{E_j} (j = 1, 2, \dots, n_e)$ 代表目标智能体 E_j 的

运动速度。

在时刻 $t \in T (T = 1, 2, \dots)$ 内所有智能体同时运动, 时刻 t 智能体在环境中的位置近似看作其中心位置, 即追踪智能体的位置 $X_p(t) = (x_p^1(t), x_p^2(t), \dots, x_p^{n_p}(t))$, 目标智能体的位置 $X_e(t) = (x_e^1(t), x_e^2(t), \dots, x_e^{n_e}(t))$, $r_{P_i} (i = 1, 2, \dots, n_p)$ 代表追踪智能体 P_i 的半径, $r_{E_j} (j = 1, 2, \dots, n_e)$ 代表目标智能体 E_j 的半径, $d_{P_i E_j} (i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_e)$ 代表追踪智能体 P_i 与目标智能体 E_j 的距离, $d_{E_j E_k} (j = 1, 2, \dots, n_e, k = 1, 2, \dots, n_e)$ 代表目标智能体 E_j 与 E_k 的距离, $d_{P_i P_k} (i = 1, 2, \dots, n_p, k = 1, 2, \dots, n_p)$ 代表追踪智能体 P_i 与 P_k 的距离。 $d_{P_i E_j} \leq r_{P_i} + r_{E_j}$ 表示追踪智能体 P_i 成功追到目标智能体 E_j , $d_{P_i P_k} \leq r_{P_i} + r_{P_k}$ 表示追踪智能体 P_i 与 P_k 相撞, $d_{E_j E_k} \leq r_{E_j} + r_{E_k}$ 表示目标智能体 E_j 与 E_k 相撞。在所有智能体中, 某一个智能体可以知道其他智能体的位置、相对距离及运动速度, 多智能体追踪问题的简易示意图如图 1 所示。

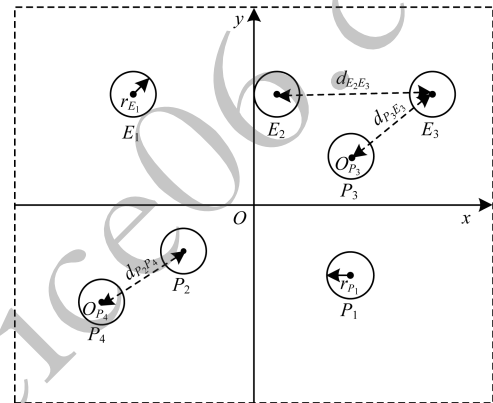


图 1 多智能体追踪示意图

Fig. 1 Schematic diagram of multi-agent tracking

1.2 多目标追踪方法框架

多目标追踪方法主要包括环境建模、任务分配和追踪策略学习 3 个方面, 具体框架如图 2 所示。先对追踪智能体、目标智能体、障碍数量及位置等环境信息进行建模, 将追踪智能体、目标智能体作为多智能体多目标任务分配算法的输入, 经过计算得到各个智能体的任务分配结果。根据各个智能体的任务分配结果和环境信息, 对其进行奖励函数设置。在每一个时间步长中, 各个智能体根据观察到的环境信息采取相应行动作用于环境, 使得环境的状态发生变化, 同时通过奖励函数从环境中获得奖励反馈并进行学习更新策略, 然后多个智能体根据观察新的环境状态采取行动从中获得奖励再进行学习。重复上述过程, 通过不断优化决策并更新策略库得到最优策略或较优策略。

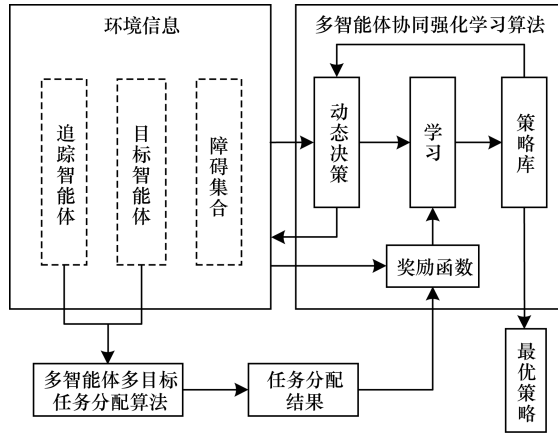


图2 多目标追踪方法框架

Fig.2 Framework of multi-target tracking method

2 基于多智能体协同强化学习的多目标追踪

2.1 多智能体多目标任务分配算法

在多智能体多目标追踪问题中,多个智能体需要通过协调与协作完成对多个运动目标的追踪任务。本文假设多个智能体之间能够进行交流与通信,同时可获取各个运动目标的位置,以缩短目标智能体的总追踪路径为优化目标,根据参与任务的智能体数目与运动目标数目建立以下任务分配模型:

1) 当追踪智能体数目与目标智能体数目相同时,即 $n_p = n_e$, 其数学模型为:

$$\begin{aligned} \min D &= \sum_{i=1}^{n_p} \sum_{j=1}^{n_e} d_{P_i E_j} x_{P_i E_j} \\ \text{s. t. } \sum_{j=1}^{n_e} x_{P_i E_j} &= 1, i = 1, 2, \dots, n_p \\ \sum_{i=1}^{n_p} x_{P_i E_j} &= 1, j = 1, 2, \dots, n_e \\ x_{P_i E_j} &= 0 \text{ 或 } 1, i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_e \end{aligned}$$

其中, D 代表追踪智能体对于目标智能体的总追踪路径; $x_{P_i E_j} = \begin{cases} 1, & \text{智能体 } P_i \text{ 追踪智能体 } E_j \\ 0, & \text{智能体 } P_i \text{ 不追踪智能体 } E_j \end{cases}$ 。

2) 当追踪智能体数目小于目标智能体数目时,即 $n_p < n_e$, 其数学模型为:

$$\begin{aligned} \min D &= \sum_{i=1}^{n_p} \sum_{j=1}^{n_e} d_{P_i E_j} x_{P_i E_j} \\ \text{s. t. } \sum_{j=1}^{n_e} x_{P_i E_j} &= 1, i = 1, 2, \dots, n_p \\ \sum_{i=1}^{n_p} x_{P_i E_j} &\leq 1, j = 1, 2, \dots, n_e \\ x_{P_i E_j} &= 0 \text{ 或 } 1, i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_e \end{aligned}$$

3) 当追踪智能体数目大于目标智能体数目时,

即 $n_p > n_e$, 其数学模型为:

$$\begin{aligned} \min D &= \sum_{i=1}^{n_p} \sum_{j=1}^{n_e} d_{P_i E_j} x_{P_i E_j} \\ \text{s. t. } \sum_{j=1}^{n_e} x_{P_i E_j} &= 1, i = 1, 2, \dots, n_p \\ \sum_{i=1}^{n_p} x_{P_i E_j} &\leq 1, j = 1, 2, \dots, n_e \\ x_{P_i E_j} &= 0 \text{ 或 } 1, i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_e \end{aligned}$$

多智能体多目标任务分配算法具体步骤如下:

步骤1 初始化追踪智能体数目 n_p 、各个追踪智能体的位置 $X_p(t_0)$ 、目标智能体数目 n_e 以及各个目标智能体的位置 $X_e(t_0)$ 。

步骤2 依次计算追踪智能体与每个目标智能体之间的距离 $d_{P_i E_j}$ 组成距离效益矩阵 D , 计算公式为:

$$D = \begin{bmatrix} d_{P_1 E_1} & d_{P_1 E_2} & \cdots & d_{P_1 E_{n_e}} \\ d_{P_2 E_1} & d_{P_2 E_2} & \cdots & d_{P_2 E_{n_e}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{P_{n_p} E_1} & d_{P_{n_p} E_2} & \cdots & d_{P_{n_p} E_{n_e}} \end{bmatrix}$$

步骤3 当追踪智能体数目 n_p 等于目标智能体数目 n_e 时转步骤4; 当追踪智能体数目 n_p 小于目标智能体数目 n_e 时转步骤5; 当追踪智能体数目 n_p 大于目标智能体数目 n_e 时转步骤6。

步骤4 运用匈牙利算法根据距离效益矩阵 D 对多个智能体的任务分配模型进行求解, 转步骤7。

步骤5 虚拟增加 $(n_e - n_p)$ 个追踪智能体, 采用加边补零法将该非标准指派问题转化为标准指派问题, 并利用匈牙利算法对多个智能体的任务分配模型进行求解, 转步骤7。

步骤6 虚拟增加 $(n_p - n_e)$ 个目标智能体, 采用加边补零法将该非标准指派问题转化为标准指派问题, 并利用匈牙利算法对多个智能体的任务分配模型求解, 转步骤7。

步骤7 输出多个追踪智能体的任务分配结果, 算法结束。

2.2 多智能体协同强化学习算法

2.2.1 状态和动作设置

在多智能体追踪问题中, 每个智能体的行为都会导致环境状态的改变进而影响其他智能体的行动。多个智能体之间存在合作关系或竞争关系, 每个智能体所获得的回报不仅与自身动作有关, 而且与其他智能体的动作有关^[18-20]。

在本文二维平面空间的多目标追踪问题中, 任意时刻的状态可以表示为 $s = \{X, V\}$, 其中, X 表示

各个智能体的初始位置及障碍等的位置信息, \mathbf{V} 表示各个智能体的运动速度。智能体的动作空间为智能体在二维平面空间 (x, y) 中任意方向移动的距离。

2.2.2 奖励函数设置

奖励函数的设置直接影响智能体的学习效果和效率。在本文多目标追踪问题中, 追踪智能体的奖励函数设置与其任务分配结果有关, 可以根据 $r_i = \omega^T m_i$ 确定, 即奖励值 r 是一组带权重的分解奖励之和。假设有 n_p 个追踪智能体追踪 n_e 个目标智能体, 得到实时的距离效益矩阵 \mathbf{D} 。根据多智能体多目标任务分配算法求解得出各个追踪智能体的任务分配结果, 假设追踪智能体 P_i 追踪目标 E_j , 则追踪智能体 P_i 的奖励函数设置为: $r_{P_i} = r_1 + r_2 + r_3 + r_4$, 其中, r_1 表示与其他追踪智能体 P_j 发生碰撞,

$$r_1 = \begin{cases} -5, & d_{P_i P_j} < r_{P_i} + r_{P_j} \\ 0, & \text{其他} \end{cases}, r_2 \text{ 表示碰到障碍 } Ob_k \text{ 受到}$$

$$\text{惩罚}, r_2 = \begin{cases} -1, & d_{P_i Ob_k} < r_{P_i} + r_{Ob_k} \\ 0, & \text{其他} \end{cases}, r_3 \text{ 表示成功追踪到}$$

$$\text{目标智能体 } E_j, r_3 = \begin{cases} 10, & d_{P_i E_j} < r_{P_i} + r_{E_j} \\ 0, & \text{其他} \end{cases}, r_4 \text{ 表示与目} \\ \text{标智能体 } E_j \text{ 的距离奖励}, r_4 = -d_{P_i E_j}。$$

对于目标智能体 E_i 而言, 其奖励函数 $r_{E_i} = r_a + r_b + r_c + r_d$, 其中: r_a 表示与其他目标智能体 E_j 发生碰撞, $r_a = \begin{cases} -1, & d_{E_i E_j} < r_{E_i} + r_{E_j} \\ 0, & \text{其他} \end{cases}$; r_b 表示碰到障碍受

到惩罚, 其计算方法同 r_2 ; r_c 表示目标智能体被任意追踪智能体 P_j 追踪到时受到惩罚, $r_c = \begin{cases} -10, & d_{P_j E_i} < r_{P_j} + r_{E_i} \\ 0, & \text{其他} \end{cases}$; r_d 表示超出环境边界受到惩

$$\text{罚}, r_d = \begin{cases} -10, & \text{智能体 } E_i \text{ 超出环境边界} \\ 0, & \text{其他} \end{cases}。$$

2.3 多目标追踪方法

本文采用多智能体多目标任务分配算法和多智能体协同强化学习算法进行多目标追踪, 其核心工作包括: 1) 根据环境中各个追踪智能体以及目标智能体的位置信息, 运用多智能体多目标任务分配算法确定多个智能体的任务分配结果; 2) 根据不同智能体的任务分配结果以及环境中的其他信息 (如障碍位置、环境边界等) 设计相应的学习模型, 多个智能体与仿真环境交互并将其经验数据存储在样本池中, 然后从样本池中随机取出一定数量的样本进行学习同时更新策略。在多个智能体的学习任务中, 所有智能体的策略由参数 $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ 确定,

其策略集合 $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, 则单智能体 i 的期望收益梯度为:

$$\nabla_{\theta_i} J(\theta_i) = E_{s \sim p^{\mu, a_i \sim \pi_i}} [\nabla_{\theta_i} \log_a \pi_i(a_i | o_i) \cdot Q_i^{\pi}(s, a_1, a_2, \dots, a_n)] \quad (1)$$

其中, $Q_i^{\pi}(s, a_1, a_2, \dots, a_n)$ 表示联合动作值函数, 状态 $s(s = (o_1, o_2, \dots, o_n))$ 包含所有智能体的观测值, 将其与所有智能体的联合动作 (a_1, a_2, \dots, a_n) 同时作为输入, 而输出为单智能体 i 的 Q 值。结合确定性策略梯度, 单智能体 i 的期望收益梯度为:

$$\nabla_{\theta_i} J(\mu_i) = E_{s, a \sim M} [\nabla_{\theta_i} \mu_i(a_i | o_i) \cdot \nabla_{a_i} Q_i^{\mu}(s, a_1, a_2, \dots, a_n) | a_i = \mu_i(o_i)] \quad (2)$$

其中, M 表示样本池, 其记录了所有智能体的经验数据。联合动作值函数 Q_i^{μ} 按式(3)进行更新:

$$\ell(\theta_i) = E_{s, a, r, s'} [(Q_i^{\mu}(s, a_1, a_2, \dots, a_n) - y)^2] \quad (3)$$

其中:

$$y = r_i + \gamma Q_i^{\mu'}(s', a'_1, a'_2, \dots, a'_n) | a'_j = \mu'_j(o_j) \quad (4)$$

多目标追踪的具体步骤如下:

1) 参数初始化, 设置环境的范围边界及追踪智能体、目标智能体及障碍的数量、位置、速度等信息以及样本池 M 的容量 K 和总训练回合数 N 。

2) 根据任务目标设置智能体的奖励函数和动作空间。

3) 设置初始训练回合数 $\text{Episode} = 0$ 、最小取样样本数 N_s 和已存储样本池数量 N_b 。

4) 判断训练回合数 Episode 是否小于 N , 如果是, 则执行下一步; 否则算法结束。

5) 根据当前状态 s_t , 每个智能体遵循当前策略选择动作 a_i 。

6) 执行动作 $a = (a_1, a_2, \dots)$, 各个智能体得到奖励值 r_i , 同时达到新的状态 s_{t+1} 。

7) 存储 (s_t, a, r, s_{t+1}) 至样本池 M , $N_b \leftarrow N_b + 1$ 。

8) 判断 N_s 是否小于 N_b , 若是, 则跳转至步骤 11; 否则执行步骤 9。

9) 对于每个智能体而言, 随机从样本池 M 中取 N_s 个样本, 根据式(4)计算 y^j , 并利用式(2)和式(3)分别更新 actor 网络和 critic 网络。

10) 更新目标网络。

11) 赋值更新 $s_t \leftarrow s_{t+1}$, 训练回合数 $\text{Episode} \leftarrow \text{Episode} + 1$, 返回步骤 4。

3 实验结果与分析

3.1 实验设置

为验证本文 MTT-MACRL 方法的可行性和有效性, 在同一实验环境下将本文 MTT-MACRL 方法与 DDPG 方法^[14] 和 MADDPG 方法^[21] 进行对比。

实验场景设置如图 3 所示,实验环境为连续的二维平面空间,其中存在 3 个追踪智能体、3 个目标智能体和 1 个障碍,在同一离散时间内 3 个追踪智能体与 3 个目标智能体同时运动,由于目标智能体被限制在该环境中,因此追踪智能体可以追踪到目标智能体。在上述实验场景中的参数设置如表 1 所示。

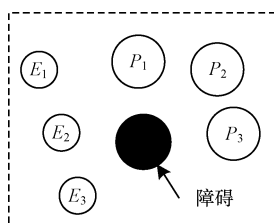


图 3 实验场景

Fig. 3 Experimental scene

表 1 参数设置

Table 1 Parameter setting

参数名	参数值
学习效率 α	0.01
衰减度 γ	0.95
样本池容量	10^6
最小取样样本数	1 024
随机种子数	3
每个训练回合的最大时间步长	25
总训练回合数	50 000

3.2 结果分析

3.2.1 学习速度和实时性对比

在多智能体追踪问题中,追踪智能体主要学习如何快速接近目标智能体以完成对多个目标智能体的追踪任务。图 4 为 10 000 个训练回合中,DDPG 方法^[14]、MADDPG 方法^[21]以及本文 MTT-MACRL 方法的目标智能体在平均每 100 个训练回合中被追踪到的总次数与训练回合数的关系。

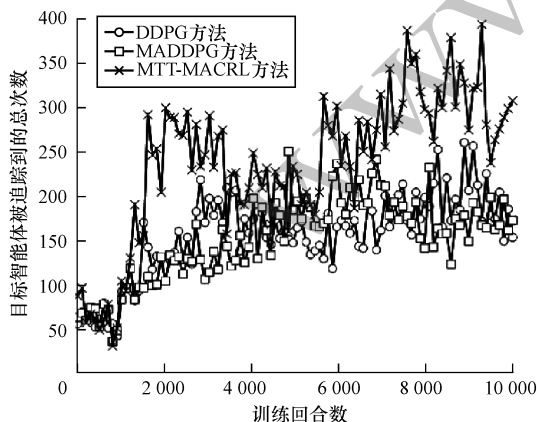


图 4 目标智能体被追踪到的总次数

Fig. 4 The total number of times the target agent has been tracked

可以看出,在 10 000 个训练回合中,本文 MTT-MACRL 方法平均每 100 个训练回合成功追踪到目标智能体的总次数为 239 次,DDPG 方法和 MADDPG 方法平均每 100 个训练回合成功追踪到目标智能体的总次数分别为 158 次、153 次。当平均每 100 个训练回合成功追踪到目标智能体的总次数达到 145 次时,运用本文 MTT-MACRL 方法、DDPG 方法和 MADDPG 方法至少分别需要进行 2 500 个、8 000 个和 7 600 个训练回合。综上所述,本文 MTT-MACRL 方法相比其他两种方法,学习速度更快,能够根据智能体的位置快速执行有效策略,且实时性更好。

3.2.2 有效性验证

为验证本文 MTT-MACRL 方法的有效性,将其与 DDPG 方法和 MADDPG 方法的学习策略分别在上场景中进行 3 次实验。在每次实验中,追踪智能体和目标智能体的位置为随机生成,每个训练回合的最大时间步长为 50 步,共进行 1 000 个回合的测试,并统计 3 次实验中每个目标智能体被追踪到的次数以及所有目标智能体被追踪到的总次数,具体情况如表 2 ~ 表 4 所示。

表 2 第 1 次实验中追踪到目标智能体的总次数

Table 2 The total number of times the target agent has been tracked in the first experiment

方法	追踪到 E_1 的次数	追踪到 E_2 的次数	追踪到 E_3 的次数
DDPG	877	1 403	274
MADDPG	162	2 280	101
MTT-MACRL	1 421	1 272	798

表 3 第 2 次实验中追踪到目标智能体的总次数

Table 3 The total number of times the target agent has been tracked in the second experiment

方法	追踪到 E_1 的次数	追踪到 E_2 的次数	追踪到 E_3 的次数
DDPG	921	1 450	270
MADDPG	133	2 504	94
MTT-MACRL	1 474	1 399	782

表 4 第 3 次实验中追踪到目标智能体的总次数

Table 4 The total number of times the target agent has been tracked in the third experiment

方法	追踪到 E_1 的次数	追踪到 E_2 的次数	追踪到 E_3 的次数
DDPG	969	1 365	286
MADDPG	152	2 428	96
MTT-MACRL	1 594	1 322	754

在 3 次实验中,采用 DDPG 方法、MADDPG 方法以及本文 MTT-MACRL 方法得到目标智能体被追踪到的总次数如图 5 所示。可以得出,利用

DDPG 方法、MADDPG 方法和本文 MTT-MACRL 方法平均每次实验追踪到目标智能体的总次数分别为 2 605 次、2 650 次和 3 605 次。本文 MTT-MACRL 方法对于目标智能体的成功追踪次数相比 DDPG 方法和 MADDPG 方法分别提高了 38.39% 和 36.04%。

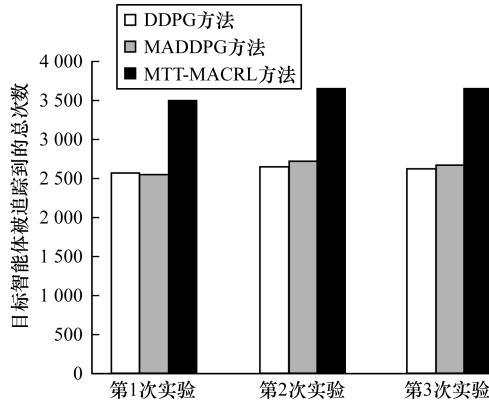


图5 3次实验中目标智能体被追踪到的总次数

Fig.5 The total number of times the target agent has been tracked in three experiments

3.2.3 协同情况对比

通过 DDPG 方法和 MADDPG 方法得到不同时刻追踪智能体及目标智能体的位置分布情况, 如图 6 和图 7 所示。可以看出, 多个追踪智能体未进行相互合作且出现了多个目标智能体同时追踪同一个目标智能体的情况, 因此造成某一个目标智能体无智能体追踪, 不能快速有效地完成追踪任务。

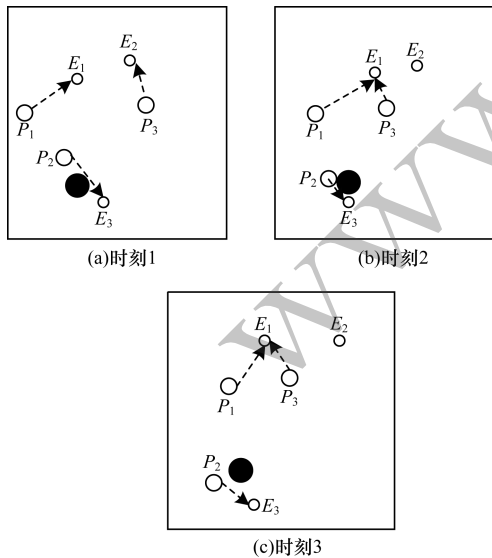


图6 DDPG 方法在不同时刻的智能体位置分布情况

Fig.6 The distribution of agent position of DDPG method at different moments

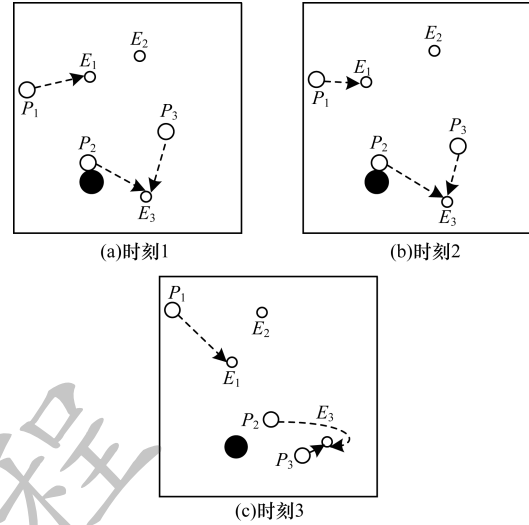


图7 MADDPG 方法在不同时刻的智能体位置分布情况

Fig.7 The distribution of agent position of MADDPG method at different moments

通过本文 MTT-MACRL 方法得到不同时刻追踪智能体及目标智能体的位置分布情况, 如图 8 所示。可以看出, 多个追踪智能体经过学习训练能够与其他智能体相互协作进行任务分配, 保证一个追踪智能体对应一个目标智能体。同时, 根据追踪智能体与目标智能体的位置信息能够实时更新任务分配情况, 保证参与追踪任务的智能体的总追踪路径最短。

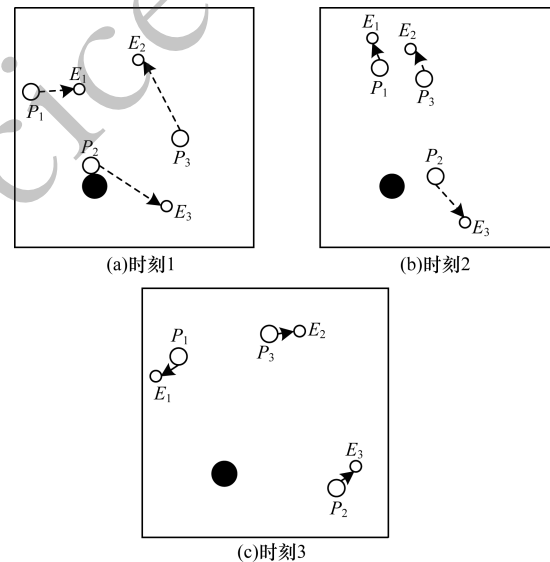


图8 MTT-MACRL 方法在不同时刻的智能体位置分布情况

Fig.8 The distribution of agent position of MTT-MACRL method at different moments

4 结束语

本文提出一种基于多智能体协同强化学习的多目标追踪方法。根据追踪和目标智能体数目及其位置信息建立任务分配模型, 运用匈牙利算法对其进

行求解得到多个追踪智能体的任务分配情况,并结合环境信息为多个追踪智能体设置奖励函数,同时通过多智能体协同强化学习算法使其在复杂环境中不断重复执行探索-积累-学习-决策过程,最终从经验数据中学习决策策略完成多目标追踪任务。实验结果表明,与 DDPG 方法和 MADDPG 方法相比,本文方法的学习速度更快,且多个智能体通过相互协作能更有效地追踪目标智能体。

参考文献

- [1] LAMINI C, FATHI Y, BENHLIMA S. Collaborative Q-learning path planning for autonomous robots based on holonic multi-agent system [C]//Proceedings of the 10th International Conference on Intelligent Systems: Theories and Applications. Washington D. C., USA: IEEE Press, 2015: 1-6.
- [2] HAJDUK M, SUKOP M, HAUN M. Agent approach to multi-agent systems [M]//HAJDUK M, SUKOP M, HAUN M. Studies in systems, decision and control. Berlin, Germany: Springer, 2018: 21-22.
- [3] HAN Xiangmin, BAO Hong, LIANG Jun, et al. An adaptive cruise control algorithm based on deep reinforcement learning [J]. Computing Engineering, 2018, 44(7): 32-35. (in Chinese)
韩向敏, 鲍泓, 梁军, 等. 一种基于深度强化学习的自适应巡航控制算法 [J]. 计算机工程, 2018, 44(7): 32-35.
- [4] YU H L, MEIER K, ARGYLE M, et al. Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles [J]. IEEE/ASME Transactions on Mechatronics, 2015, 20(2): 541-552.
- [5] YANG P, TANG K, LOZANO J A, et al. Path planning for single unmanned aerial vehicle by separately evolving waypoints [J]. IEEE Transactions on Robotics, 2015, 31(5): 1130-1146.
- [6] ZHOU Hailing, KONG Hui, WEI Lei, et al. Efficient road detection and tracking for unmanned aerial vehicle [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(1): 297-309.
- [7] SHAH K, SCHWAGER M. Multi-agent cooperative pursuit-evasion strategies under uncertainty [M]//CORRELL N, SCHWAGER M, OTTE M. Distributed autonomous robotic systems. Berlin, Germany: Springer, 2019: 451-468.
- [8] GARCIA-FERNANDEZ A F, SVENSSON L. Multiple target tracking based on sets of trajectories [J]. IEEE Transactions on Aerospace and Electronic Systems, 2020, 56(3): 1685-1707.
- [9] SOUIDI M E H S, SIAM A, PEI Z Y, et al. Multi-agent pursuit-evasion game based on organizational architecture [J]. Journal of Computing and Information Technology, 2019, 27(1): 1-11.
- [10] DUAN Yong, XU Xinhe. Research on multi-robot cooperation strategy based on multi-agent reinforcement learning [J]. Systems Engineering—Theory & Practice, 2014, 34(5): 1305-1310. (in Chinese)
段勇, 徐心和. 基于多智能体强化学习的多机器人协作策略研究 [J]. 系统工程理论与实践, 2014, 34(5): 1305-1310.
- [11] GUPTA J K, EGOROV M, KOCHENDERFER M. Cooperative multi-agent control using deep reinforcement learning [C]//Proceedings of International Conference on Autonomous Agents and Multiagent Systems. Berlin, Germany: Springer, 2017: 66-83.
- [12] WEI E, WICKE D, FREELAN D, et al. Multiagent soft Q-learning [C]//Proceedings of 2018 AAAI Spring Symposium Series. Palo Alto, USA: AAAI Press, 2018: 1-10.
- [13] FOERSTER J, FARQUHAR G, AFOURAS T, et al. Counterfactual multi-agent policy gradients [EB/OL]. [2019-08-01]. <https://arxiv.org/abs/1705.08926>.
- [14] LILLICRAP T, HUNT J, PRITZEL A, et al. Continuous control with deep reinforcement learning [EB/OL]. [2019-08-01]. <https://arxiv.org/abs/1509.02971>.
- [15] YAN Yalin. Research on multi-robot pursuit-evasion problem based on game theory [D]. Harbin: Harbin Engineering University, 2014. (in Chinese)
晏亚林. 基于博弈论的多机器人追捕问题的研究 [D]. 哈尔滨: 哈尔滨工程大学, 2014.
- [16] FANG Baofu, PAN Qishu, HONG Bingrong, et al. Constraint conditions of successful capture in multi-pursuers vs one-evader games [J]. Robot, 2012, 34(3): 282-291. (in Chinese)
方宝富, 潘启树, 洪炳榕, 等. 多追捕者-单一逃跑者追逃问题实现成功捕获的约束条件 [J]. 机器人, 2012, 34(3): 282-291.
- [17] ZHANG Xu, LI Ling, JIA Leilei. Research and simulation of multi-robot pursuit and escape strategy based on differential game [J]. Equipment Manufacturing Technology, 2015(9): 9-12. (in Chinese)
张旭, 李玲, 贾磊磊. 基于微分博弈的多机器人追逃策略研究及仿真 [J]. 装备制造技术, 2015(9): 9-12.
- [18] DU Wei, DING Shifei. Overview on multi-agent reinforcement learning [J]. Computer Science, 2019, 46(8): 1-8. (in Chinese)
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.
- [19] ZHANG Yue. Research on multi-agent deep reinforcement learning methods and applications [D]. Xi'an: Xidian University, 2018. (in Chinese)
张悦. 多智能体深度强化学习方法及应用研究 [D]. 西安: 西安电子科技大学, 2018.
- [20] WANG Weixun, HAO Jianye, WANG Yixi, et al. Towards cooperation in sequential prisoner's dilemmas: a deep multiagent reinforcement learning approach [EB/OL]. [2019-08-01]. <https://arxiv.org/abs/1803.00162>.
- [21] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [C]//Proceedings of Advances in Neural Information Processing Systems. Berlin, Germany: Springer, 2017: 6379-6390.