计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(11):203−211 [doi: 10.15888/j.cnki.csa.009277]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进 MADDPG 的 UAV 轨迹和计算卸载联合优化算法①

苏维亚[1], 徐　飞[1], 王　森[2]

[1](西安工业大学 计算机科学与工程学院, 西安 710021)
[2](西安工业大学 兵器科学与技术学院, 西安 710021)
通信作者: 徐　飞, E-mail: xufei@xatu.edu.cn

摘　要: 在地震、台风、洪水、泥石流等造成严重破坏的灾区, 无人机 (unmanned aerial vehicle, UAV) 可以作为空中边缘服务器为地面移动终端提供服务, 由于单无人机有限的计算和存储能力, 难以实时满足复杂的计算密集型任务. 本文首先研究了一个多无人机辅助移动边缘计算模型, 并构建了数学模型; 然后建立部分可观察马尔可夫决策过程, 提出了基于复合优先经验回放采样方法的 MADDPG 算法 (composite priority multi-agent deep deterministic policy gradient, CoP-MADDPG) 对无人机的时延能耗以及飞行轨迹进行联合优化; 最后, 仿真实验结果表明, 本文所提出算法的总奖励收敛速度和收敛值均优于其他基准算法, 且可为 90% 左右的地面移动终端提供服务, 证明了本文算法的有效性与实用性.

关键词: 移动边缘计算; 多智能体; 联合优化; 深度强化学习; 部分可观察马尔可夫决策过程; 计算卸载

引用格式: 苏维亚,徐飞,王森.基于改进 MADDPG 的 UAV 轨迹和计算卸载联合优化算法.计算机系统应用,2023,32(11):203–211. http://www.c-s-a.org.cn/1003-3254/9277.html

Joint Optimization Algorithm for UAV Trajectory and Computational Offloading Based on Improved MADDPG

SU Wei-Ya[1], XU Fei[1], WANG Sen[2]

[1](School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China)
[2](School of Ordnance Science and Technology, Xi'an Technological University, Xi'an 710021, China)

Abstract: Unmanned aerial vehicles (UAVs) can act as air edge servers to provide services for ground mobile terminals in disaster areas where earthquakes, typhoons, floods, and mudslides have caused severe damage. However, it is difficult to complete complex computationally intensive tasks in real time due to the limited computation and storage capacity of a single UAV. In this study, a multi-UAV-assisted mobile edge computing model is first investigated and a mathematical model is built. Then a partially observable Markov decision process is established and an improved multi-agent deep deterministic policy gradient (MADDPG) algorithm based on the composite priority experiential replay sampling method (CoP-MADDPG) is proposed to jointly optimize time delay, energy consumption, and flight trajectory of UAVs. Finally, the simulation experimental results show that the proposed algorithm outperforms other benchmark algorithms in terms of total reward convergence speed and convergence value, and can provide services for about 90% of ground mobile terminals, proving the effectiveness and practicality of the proposed algorithm.

Key words: mobile edge computing; multi-agent; joint optimization; deep reinforcement learning; partially observable Markov decision process; computational offloading

① 基金项目: 航天高可信嵌入式软件工程技术实验室基金; 西安市碑林区科技计划 (GX2137)
收稿时间: 2023-04-08; 修改时间: 2023-05-11; 采用时间: 2023-05-23; csa 在线出版时间: 2023-07-21
CNKI 网络首发时间: 2023-07-21

1 引言

智能体是环境中的实体, 可以执行对环境产生影响的行为. 多智能体系统是多个智能体组成的集合, 其目标是将大而复杂的系统建设成小而彼此互相通信协调的易于管理的系统[1]. 多智能体系统具有自主性、分布性、协调性, 并具有自组织能力、学习能力和推理能力, 较单智能体而言具有很强的鲁棒性和可靠性[2]. 多智能体系统广泛应用于自动驾驶[3]、智能电网优化调度[4,5]、网络数据传输路由优化[6]、多无人系统协同任务[7,8] 等领域中.

深度强化学习的兴起解决了由于计算设备性能的提高而产生的海量数据问题, 强化学习中的智能体通过与环境进行交互得到相应的奖励值, 进一步获得最优策略, 但在多个智能体环境中, 每个智能体为了自身的利益, 不断学习优化自身策略, 从而导致每个智能体在训练过程中的状态空间不断变化, 无法找出最优策略[9]. 在以上问题的基础上, 许多研究人员开始将深度强化学习 (deep reinforcement learning, DRL) 与多智能体系统结合起来, 多智能体深度强化学习 (multi-agent DRL) 方法应运而生, OpenAI Five、 AlphaStar、“绝悟”等游戏 AI 都能够达到甚至超越人类顶尖玩家的水平, 为 MADRL 在无人控制系统、智能决策等诸多领域的应用前景提供了广阔想象空间[10].

基于以上研究, 提出了基于 CoP-MADDPG 算法的多无人机辅助移动边缘计算系统, 本文主要贡献如下.

(1) 构建了一个由多架无人机组成的移动边缘计算系统模型. 采用三维动力学模型为在三维空间中随机移动的无人机和地面移动终端进行建模. 无人机与地面移动终端之间的信道模型为莱斯信道.

(2) 使用部分可观察马尔可夫决策过程对系统模型进行建模, 并提出了 CoP-MADDPG 算法实现无人机飞行轨迹和任务卸载的联合优化. 考虑到传统的数据抽取方式容易导致训练效率低, 收敛速度慢, 故在抽样过程中使用了复合优先级, 复合优先级由基于立即回报的优先级和基于 TD-error 的优先级组成.

(3) 构建了一个面向多无人机的飞行轨迹和任务卸载联合优化的仿真实验. 不同参数下的仿真实验表明, 当 Actor 网络和 Critic 网络的学习率均为 0.01, 折扣因子为 0.95 时总奖励可以在最短时间内收敛到最大值. 不同算法下的仿真实验表明, 本文算法的总奖励收敛速度和收敛值均优于其他基准算法; 在地理公平性方面, 本文算法可以较为均匀的覆盖整个区域, 且为 90% 左右的地面移动终端提供服务.

2 相关工作

Jiang 等人[11] 提出了一种基于 MADRL 的协同优化策略, 以解决 5G 超密集异构网络中的计算卸载和资源分配问题. Wu 等人[12] 提出了一个基于排队理论的时延和能耗联合约束优化模型, 并使用 MADRL 获得动态和随机多用户卸载环境中的最优卸载策略. Seid 等人[13] 提出了一种基于 MADRL 的方法, 以最大限度地降低整体网络计算成本, 同时确保物联网网络中物联网设备或 UE 的服务质量 (QoS) 要求. Zhou 等人[14] 提出了一个分层的多智能体深度强化学习 (H-MADRL) 框架来解决混合计算卸载问题, 高级代理驻留在 AP 中并优化波束成形策略, 低级用户代理学习并调整个人的卸载策略. Seid 等人[15] 提出了一个区块链和 MADRL 集成框架, 用于在多无人机辅助物联网网络中与 EH 进行计算卸载. Xue 等人[16] 考虑了 UE 卸载成本和 MEC 服务器的定价, 提出了一种 MADRL 算法, 通过联合优化功率控制、资源分配和 UE 关联来最小化系统能耗, 从而在保证系统性能的前提下, 有效提高无人机的整体收益. Li 等人[17] 研究了一种空间/空中辅助边缘计算网络架构, 提出了一种基于 MADRL 的方案, 以获得考虑动态计算请求和随机时变信道条件的最优任务卸载策略, 同时确保服务质量要求. Cheng 等人[18] 使用 MASAC 算法对无人机辅助和能量约束智能边缘网络下的联合任务和能量卸载问题进行了研究. Zhao 等人[19] 考虑无人机距离、碰撞和通信等因素, 构建了复杂环境下多无人机协同任务分配模型, 并提出了一种求解该模型的 MASAC 算法. Dai 等人[20] 在多智能体系统中引入了联邦学习框架, 通过无人机之间共享用户的非私有数据, 生成相应的全局模型, 从而生成无人机网络的全局最优决策.

3 网络模型

3.1 系统模型

如图 1 所示, 本文在环境中部署了 O 个障碍物、M 个地面移动终端和 K 架搭载 MEC 服务器的无人机, 无人机的整个飞行周期 T 被分为 t 个时隙, 设定无人机和地面移动终端的位置在每一时隙均不发生变化, 且无人机只在悬停时处理由地面移动终端卸载的部分任

务. $pos_t^k=\{x_t^k,y_t^k,z_t^k\}$表示$t\in\{1,2,3,\cdots,T\}$时隙第$k\in\{1,2,3,\cdots,K\}$架无人机的位置, $pos^m=\{x^m,y^m\}$表示地面移动终端$m\in\{1,2,3,\cdots,M\}$的位置, $pos^o=\{x^o,y^o,z^o\}$表示障碍物$o\in\{1,2,3,\cdots,O\}$的位置, 无人机k的 CPU 频率用f^k表示, 地面移动终端m的 CPU 频率用f^m表示, R_t^k表示t时隙无人机k的覆盖范围, 可以根据波束宽度进行计算.

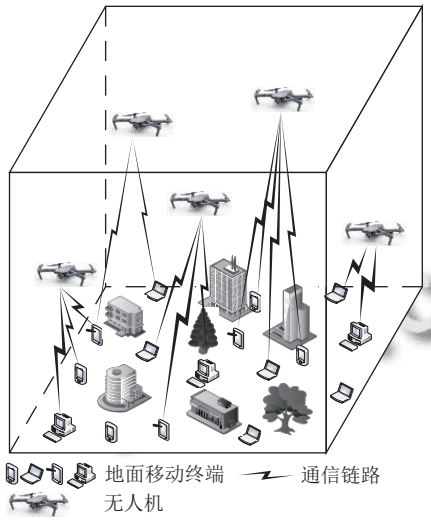


图 1 系统模型

传统的概率 Los 信道模型往往不能适应农村、城市及森林等复杂环境[21]. 故本文引入了莱斯衰弱信道模型[22–24]. 无人机与地面移动终端之间的信道增益可以表示为:

$$h_k=\sqrt{\beta_k}g_k \tag{1}$$

其中, $\beta_k=\beta_0(d_k)^{-\alpha_0}$是大尺度平均信道功率增益, β_0是参考距离为 1 m 时的信道增益, d_k是地面移动终端与无人机k之间的距离, α_0是路径损耗指数. g_k表示小尺度衰减系数, 可以定义为:

$$g_k=\sqrt{\frac{RF_k}{RF_k+1}}g+\sqrt{\frac{1}{RF_k+1}}\widetilde{g} \tag{2}$$

$$\theta_k=\arcsin\left(\frac{z_k}{d_k}\right) \tag{3}$$

其中, $RF_k=A_1\mathrm{e}^{A_2\theta_k}$, g对应可视距链路分量, $\widetilde{g}$对应散射分量, RF_k表示莱斯因子, A_1, A_2是由环境决定的常数, θ_k表示无人机k与地面移动终端之间的仰角.

无人机k最大传输速率可以定义为:

$$r_k=B\log_2\left(1+\frac{|h_k|^2p_k^{\mathrm{up}}}{\sigma^2\omega}\right) \tag{4}$$

其中, σ为无人机接收处的噪声功率, ω为信噪比差值, B表示信道带宽, p_k^{up}为无人机k的上行链路传输功率.

3.2 计算模型

由于地面移动终端有限的计算能力, 无法处理计算密集型任务, 故需要将部分任务卸载到无人机进行处理, $o(t)$表示任务的卸载比率, D_t^k表示t时隙地面移动终端m产生的总任务量, C表示处理单位字节所需的 CPU 周期数, 通过式 (5)–式 (7) 可以计算出t时隙的本地执行时延和卸载计算时延, 其中卸载计算时延为传输时延和计算时延的累加和, 即$T_t^{\mathrm{off}}+T_t^{\mathrm{up}}$.

$$T_t^{\mathrm{local}}=\frac{(1-o(t))D_t^kC}{f^m} \tag{5}$$

$$T_t^{\mathrm{up}}=\frac{o(t)D_t^k}{r_k} \tag{6}$$

$$T_t^{\mathrm{off}}=\frac{o(t)D_t^kC}{f^k} \tag{7}$$

能耗与无人机和地面移动终端的芯片结构有关, 可以通过式 (8)–式 (10) 进行计算, 其中卸载能耗为传输能耗和计算能耗的累加和, 即$E_t^{\mathrm{up}}+E_t^{\mathrm{off}}$.

$$E_t^{\mathrm{local}}=\rho_lk_l(f^m)^3 \tag{8}$$

$$E_t^{\mathrm{up}}=p_k^{\mathrm{up}}\cdot T_t^{\mathrm{up}} \tag{9}$$

$$E_t^{\mathrm{off}}=\rho_ok_o(f^k)^3 \tag{10}$$

其中, ρ_l, ρ_o分别是取决于地面移动终端芯片结构和无人机芯片结构的系数. k_l和k_o为有效开关系数, 取决于地面移动终端的芯片结构和无人机的芯片结构.

3.3 无人机运动模型

无人机需要在避开障碍物的同时为地面移动终端提供服务, 故本文使用了三维动力学模型为无人机进行建模. 无人机的位置由飞行速度、角度和时延决定, 角度包括与 XOY 面的夹角和与 Z 轴的夹角, 时延包括无人机的飞行时延和处理上传任务的时延. 由式 (11)–式 (13) 可以计算出无人机k在$t+1$时隙的位置:

$$x_{t+1}^k=x_t^k+distance_{\mathrm{fly}}\times\cos\theta\times\sin\beta \tag{11}$$

$$y_{t+1}^k=y_t^k+distance_{\mathrm{fly}}\times\sin\theta\times\sin\beta \tag{12}$$

$$z_{t+1}^k=z_t^k+distance_{\mathrm{fly}}\times\cos\beta \tag{13}$$

其中, θ是无人机飞行过程中与 XOY 面的夹角, β是无人机飞行过程中与Z轴的夹角, $distance_{\mathrm{fly}}=v_{\max}\times(t_{\mathrm{fly}}+t_{\mathrm{delay}})\times$

v_{weight}, $v_{\max}$ 表示无人机的最大飞行速度, v_{weight} 表示无人机的速度分量, t_{delay} 表示无人机处理上传任务的时延.

3.4 问题描述

本文采用了一种联合优化的策略, 将任务卸载比率、无人机飞行轨迹和无人机覆盖率 3 个方面作为优化目标, 旨在寻找一种平衡的解决方案, 使得整个系统的性能得到最优提升. 具体而言, 任务卸载比率的优化旨在保证整个系统任务处理的高效性和负载均衡性, 无人机飞行轨迹的优化则能够最小化整个时间段内的最大处理时延和能耗, 并确保无人机之间不会发生碰撞, 无人机覆盖率的优化则能够最大化地面移动终端的服务范围和服务质量, 使得整个网络系统的性能得到最大化提升. 本文的优化问题可以表示为:

$$\min_{\{\text{collision}_t^{ko}, r_t^k, cov\}} \sum_{t=1}^{T}\sum_{k=1}^{K} \max\left\{E_t^k + T_t^k\right\} \tag{14}$$

$$\text{s.t. } 0 \leqslant r_t^k \leqslant 1 \tag{14a}$$

$$cov \in \{0,1\} \tag{14b}$$

$$pos_t^k = \left\{(x_t^k, y_t^k, z_t^k) \middle| x_t^k \in [0,L], y_t^k \in [0,W], z_t^k \in [0,H]\right\} \tag{14c}$$

$$0 \leqslant Task(t) \leqslant sum\_Task \tag{14d}$$

$$0 \leqslant R_t^k \leqslant R_{\max} \tag{14e}$$

$$0 \leqslant v_k \leqslant v_{\max} \tag{14f}$$

其中, 式 (14) 中的 collision_t^{ko} 表示在 t 时隙无人机 k 和障碍物 o 之间是否存在障碍, 若存在则 $\text{collision}_t^{ko}=1$, 否则 $\text{collision}_t^{ko}=0$. 式 (14a) 表示计算任务卸载比率 r_t^k 的取值范围. 式 (14b) 表示无人机覆盖率 cov 的取值范围. 式 (14c) 表示无人机的移动范围. 式 (14d) 表示剩余任务量的范围, sum_Task 表示总任务量. 式 (14e) 表示 t 时隙无人机 k 的覆盖范围 R_t^k 应小于等于无人机的最大覆盖范围 $R_{\max}$. 式 (14f) 表示无人机 k 在飞行过程中的速度 v_k 不能超过无人机的最大飞行速度 $v_{\max}$.

4 CoP-MADDPG 算法

4.1 MADDPG 算法

DDPG 算法在单智能体领域得到了广泛的应用, 将其进行扩展便得到了多智能体领域的 MADDPG 算法[25], 如图 2 所示, 该算法引入了其他智能体的动作作为额外信息以获得 Q 值函数. MADDPG 算法在集中训练过程中不需要知道环境的动力学模型以及特殊的通信需求, 每个智能体根据其他智能体的行为评估当前动作的价值. 分散执行是指当每个智能体都训练充分后, 每个 Actor 网络就可以自己根据状态采取合适的动作, 此时是不需要其他智能体的状态或者动作的.

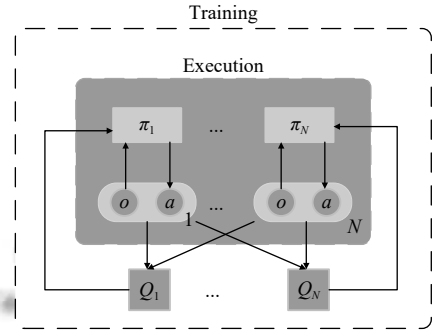


图 2 MADDPG 算法框架

传统强化学习算法在学习和应用时都必须使用相同的数据信息, 而 MADDPG 算法允许在学习时使用一些额外的信息, 即全局信息, 但是在应用决策的时候只使用局部信息. 因此, 该算法不仅可以用于合作环境, 也可以用于竞争环境. MADDPG 算法在保证精度的基础上, 能够解决因多智能体输出的动作维度太大而导致的算法收敛问题[26], 但传统的 MADDPG 算法从经验回放缓冲区中随机抽取数据, 而不考虑数据质量, 导致训练效果差, 收敛速度慢, 容易陷入局部最优. 因此, 本文使用复合优先级抽样方法对 MADDPG 算法进行了改进, 从而对任务卸载比率、无人机飞行轨迹和无人机覆盖率进行联合优化.

4.2 构建 POMDP

本文的联合状态空间可以表示为 $state=\{pos_t^k, pos^o, pos^m, task_t^m, block_t^k, ele_t^k\}$, $task_t^k$ 表示地面移动终端 m 在 t 时隙产生的任务量. $block_t^k$ 表示在 t 时隙无人机 k 与地面移动终端之间是否存在遮挡, 若存在遮挡, 则 $block_t^k=1$, 否则 $block_t^k=0$. ele_t^k 表示 t 时隙无人机 k 的电量.

本文的联合动作空间可以表示为 $a=[R_t^k, v_k, \varphi_k, \theta_k, offloading]$, φ_k 和 θ_k 表示无人机 k 的飞行角度, $offloading$ 表示卸载比率.

奖励函数会影响神经网络的收敛情况, 故它的设置是非常重要的, 本文的奖励函数设置为时延能耗、无人机覆盖率、碰撞因子和边界因子的加权和, 由于时延和能耗不是一个数量级, 故需要对其进行归一化, 无人机覆盖不同的区域应得到正向奖励鼓励对区域的

探索, 碰撞因子与无人机和障碍物之间的距离有关, 边界因子可以避免无人机飞出边界.

4.3 CoP-MADDPG 算法

经验回放机制随机重复地抽取过去的经验以提高神经网络的稳定性, 但未考虑到数据的质量, 导致训练效率低, 收敛速度慢, 故本文在抽样过程中使用了复合优先级, 其中, 复合优先级包括基于立即回报的优先级和基于 TD-error 的优先级. 复合优先经验回放采样方法的具体步骤如下所示.

(1) 使用 Q 值计算 TD-error.

(2) 利用式 (15) 分别定义基于立即汇报的优先级和基于 TD-error 的优先级, r_t表示经验的立即汇报, ε表示一个正常数, δ_t表示 TD-error:

$$Y_i = r_t + \varepsilon;\ Y_f = |\delta_t| + \varepsilon \tag{15}$$

(3) 把经验池中的经验分别按步骤 (2) 中得到的优先级从大到小进行排列得到序列$rank_i$和$rank_f$, 通过式 (16) 计算出每个经验的复合优先级, α表示算法使用优先级的程度, 当$\alpha = 0$时表示均匀采样.

$$u_k = \frac{rank_i + rank_f}{2};\ Y_k = \left(\frac{1}{u_k}\right)^{\alpha} \tag{16}$$

(4) 定义采样经验的概率$P_k = \left(\dfrac{Y_k}{\sum_n Y_n}\right)$, 其中$n$表示经验的数量.

算法 1. MADDPG 算法

输入: 使用参数随机初始化 Actor 和 Critic 网络; 初始化经验池D, 最小取样数量S, 回合数N.
输出: 奖励值.

for episode=1:N do
  初始化初始状态s
  for t=1:T do
    对于每个智能体i, 选择动作$a_i=\mu_{\theta_i}(o_i)+N_t$并执行动作$a=(a_1,\cdots,a_N)$得到奖励值和下一时刻状态$s'$并计算 TD-error;
    将经验(s,a,r,s',y_i,y_f)存储到经验池D, 首先将经验按优先级Y_i和Y_f从大到小进行排序, 得到$rank_i$和$rank_f$, 其次对经验做复合平均排序得到u_k并计算经验的复合优先级Y_k, 最后通过Y_k计算经验采样概率P_k;
    for $agenti$=1:M do
      从经验池D中根据经验采样概率抽取S个样本进行训练;
      设置:

$$y^j = r_i^j + \gamma Q_i^{\mu'}(s'^j, a'_1, \cdots, a'_N)\big|_{a'_k=\mu'_k(o_k^j)}$$

      使用 Loss 值更新 Critic 网络:

$$L(\theta_i) = \frac{1}{s}\sum_j (y^j - Q_i^{\mu}(s^j, a_1^j, \cdots, a_N^j))^2$$

      使用梯度下降方法更新 Actor 网络:

$$\nabla_{\theta_i} J \approx \frac{1}{s}\sum_j \nabla_{\theta_i}\mu_i(o_i^j)\nabla_{a_i} Q_i^{\mu}(s^j, a_1^j, \cdots, a_i^j, \cdots, a_N^j)\big|_{a_i=\mu_i(o_i^j)}$$

    end for
    更新目标网络:

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'};\ \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$$

  end for
end for

5 仿真实验

在进行仿真实验时, 通过比较不同学习率和折扣因子下的平均奖励值可以获得最佳学习率和折扣因子, 通过与基准算法进行对比, 验证了本文算法的可用性和先进性. 本文的基准算法为: MADDPG、MAA3C、MAPPO、MAAC.

5.1 参数设置

在模拟过程中, 使用的编程语言是 Python 3.8 和 TensorFlow 2.5. 使用了一台配备 Intel 酷睿 i7-1165G7 CPU 的 PC, 最高频率 2.80 GHz. 本文环境参数设置如表 1 所示.

表 1 参数设置

参数	意义	默认值
K	无人机数量	5个
O	障碍物数量	3个
M	地面移动终端数量	200个
R_{obs}	障碍物半径	{42, 32, 36}m
T	飞行周期	18 min
t_{fly}	无人机飞行时间	1 s
t_{delay}	无人机悬停时间	7 s
R_{max}	无人机最大覆盖范围	30 m
v_{max}	无人机最大飞行速度	21 m/s
B	带宽	1 MHz
σ_{los}	视距链路下的噪声功率	10^{-13} mW
σ_{Nlos}	非视距链路下的噪声功率	10^{-11} mW
f^k	无人机k的计算频率	1.2 GHz
r	影响因子	10^{-27}
C	CPU周期数	1 000
p_k^{up}	上行链路的传输功率	0.1 W
g	距离为1 m时的信道增益	−50 dB
ele_0^k	无人机k的初始电量	500 kJ
m	无人机质量	9.65 kg

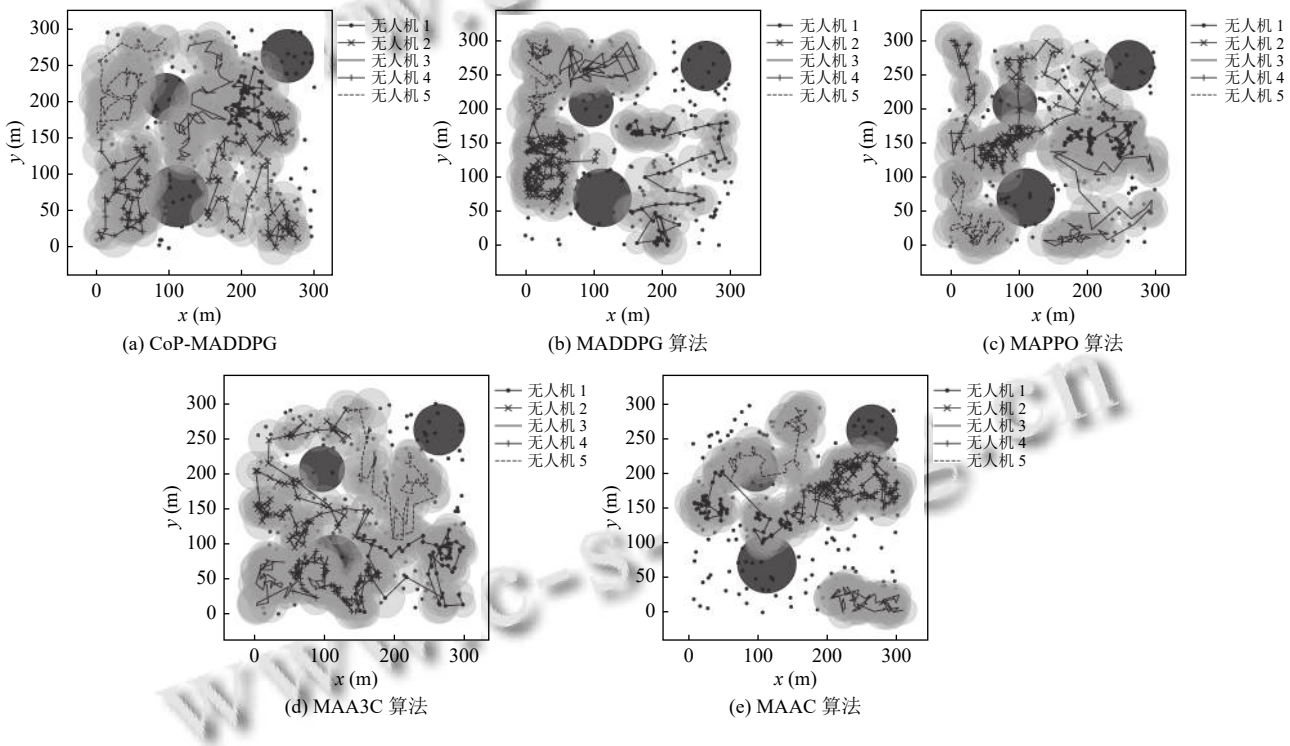
5.2 仿真结果

5.2.1 无人机分布示意图

二维坐标系中黑色实心小圆表示地面移动终端, 深灰色实心圆表示障碍物, 浅灰色实心圆表示无人机的覆盖范围, 5 种不同的线条分别表示 5 架无人机的飞行轨迹.

不同算法下的无人机飞行轨迹如图 3 所示. 图 3(a) 是基于本文算法的无人机飞行轨迹, 可以看出无人机

能够完全避开障碍物和避免无人机之间的碰撞, 5 架无人机在飞行过程中轨迹分布较为均匀, 基本实现了区域全覆盖, 通过深灰色实心圆中黑色实心圆的个数可以看出, 无人机在飞行过程中为 90% 左右的地面移动终端提供了服务, 实现了服务用户公平性, 提高了服务效率. 图 3(b) 是基于 MADDPG 算法的无人机飞行轨迹, 可以看出无人机能够完全避开障碍物和避免无人机之间的碰撞, 但为了避免碰撞, 大多数无人机陷入了局部最优, 只在一块较小的区域内移动, 导致该算法不能很好地覆盖整个区域, 从图中可以看出, 4 架无人机覆盖的总面积仅占整体区域的 1/4 左右, 且几乎一半的地面移动终端不能得到服务. 图 3(c) 是基于 MAPPO 算法的无人机飞行轨迹, 可以看出无人机可以较好地为地面移动终端提供服务且能够在飞行过程中避免无人机之间发生碰撞, 但无法完全避开障碍物, 该算法解决了 MADDPG 算法陷入局部最优的问题, 稳定性相较于 MADDPG 也得到了提升, 但最终训练效果较差. 图 3(d) 是基于 MAA3C 算法的无人机飞行轨迹, 可以看出无人机基本实现了地理公平性和服务用户公平性, 但由于 A3C 算法未使用经验回放机制, 导致采样速度变慢, 训练十分不稳定, 因此, 无人机在飞行过程中不能避免任何一种情况的碰撞. 图 3(e) 是基于 MAAC 算法的无人机飞行轨迹, 从图中可以看出除了无人机之间发生了碰撞之外, 还未能避免无人机与障碍物之间的碰撞, 在训练过程中无人机无法很好地对环境进行探索, 导致无人机分布较为集中, 且单架无人机覆盖区域较小, 因此, 无法全面覆盖整个区域且只为一半左右的地面移动终端提供了服务.



(a) CoP-MADDPG　(b) MADDPG 算法　(c) MAPPO 算法

(d) MAA3C 算法　(e) MAAC 算法

图 3　无人机飞行轨迹示意图

5.2.2　基于不同学习率的奖励曲线图

目标函数是否收敛以及何时收敛由学习率控制, 图 4 是不同学习率下的奖励曲线图, 其中 LA 表示 Actor 网络学习率, LC 表示 Critic 网络学习率. 从图中可以看出, 当 Actor 网络的学习率和 Critic 网络的学习率均为 0.000 1 时, 奖励值在−80 附近震荡, 无法收敛. 当 Actor 网络的学习率和 Critic 网络的学习率均为 0.001 时, 奖励值在 2 000 回合左右开始收敛, 最终收敛到−30 左右. 当 Actor 网络的学习率和 Critic 网络的学习率均为 0.01 时, 奖励值在 1 000 回合附近开始收敛, 最终收敛到−10 附近. 通过对图 4 的分析, 可以得出结论: 当 Actor 网络的学习率和 Critic 网络的学习率均为 0.01 时, 可以获得较好的收敛速度和收敛值.

5.2.3　基于不同折扣因子的平均奖励曲线图

折扣因子γ用来调节未来奖励对当前奖励值的影响, 它的选取原则是: 在算法能够收敛的前提下尽可能

的大. 图 5 是不同折扣因子下的奖励曲线图, 当$\gamma=0.3$时, 奖励值在−80 左右震荡不收敛, 当$\gamma=0.6$时, 奖励值在 2 000 回合左右收敛, 最终收敛到−50 左右, 当$\gamma=0.8$时, 奖励值在 1 800 回合左右开始收敛, 最终收敛到−40 左右, 当$\gamma=0.95$时, 奖励值在 1 000 回合左右开始收敛, 最终收敛到−10 左右. 通过分析不同折扣因子下奖励值的收敛速度和收敛值, 可以得出结论: 当$\gamma=0.95$时的收敛速度是最快的, 收敛值是最高的.

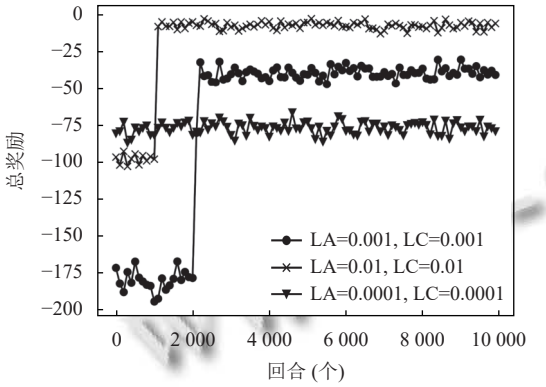


图 4 基于不同学习率的奖励值

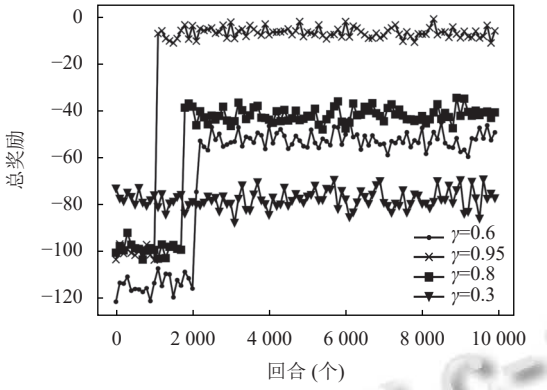


图 5 基于不同折扣因子的奖励值

5.2.4 本文算法与基准算法平均奖励和概率对比图

图 6 是本文算法与基准算法的奖励值对比, 从图中可以看出 5 种算法均有收敛趋势, MAA3C 和 MAAC 算法收敛速度慢, 收敛幅度小, MAPPO, MADDPG 算法和本文算法相比, 3 个算法都在 1 000 回合左右收敛, 但是本文算法的收敛值略高于其他两个算法且收敛幅度较大. 通过对图 6 中数据的分析比较, 验证了本文算法的可用性与有效性.

图 7 是本文算法与基准算法的无人机覆盖率对比图, 从图中可以看出 MAAC 算法在训练过程中无人机覆盖率从 55% 左右增加到了 60% 左右. MAA3C 算法在训练过程中无人机覆盖率从 50% 左右增加到了 65% 左右. MADDPG 算法在训练过程中无人机覆盖率从 58% 增加到 85% 左右. MAPPO 算法在训练过程中无人机覆盖率从 62% 增加到 85% 左右. 本文算法在训练过程中无人机覆盖率从 60% 左右增加到了 90% 左右, 并最终在 90% 左右震荡收敛. 通过对图 7 进行分析, 可以得出结论: 本文算法在训练过程中无人机可为 90% 左右的地面终端提供服务, 基本实现了地理公平性.

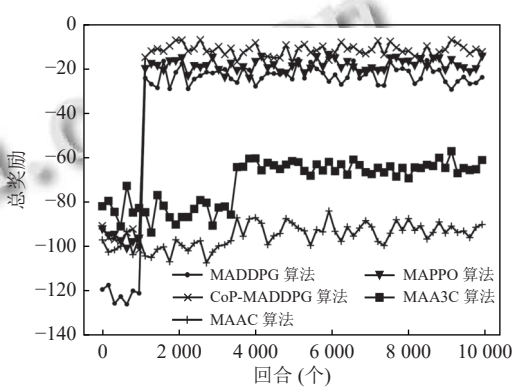


图 6 基于不同算法的奖励值

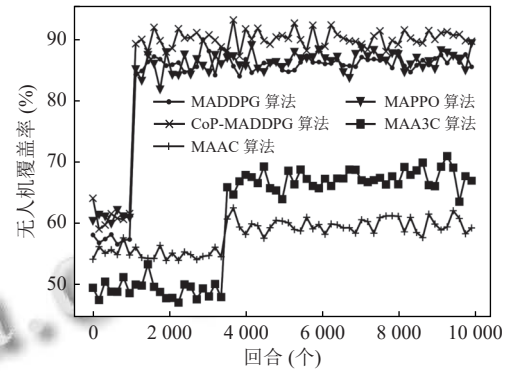
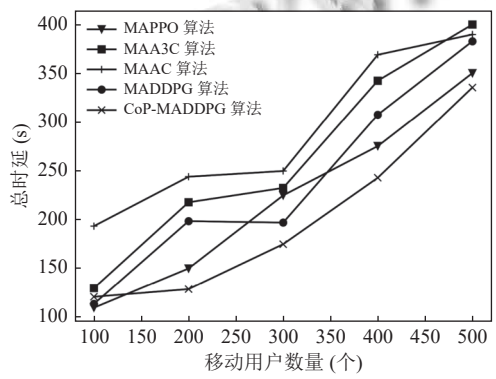


图 7 无人机覆盖率

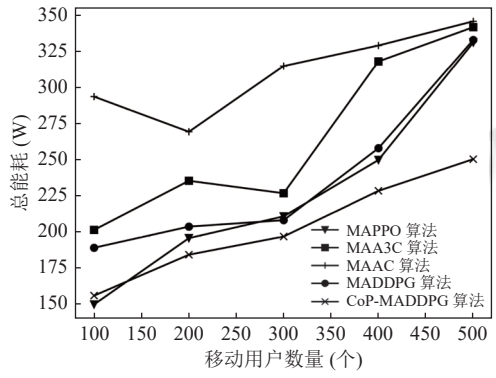
5.2.5 性能分析

地面移动终端和障碍物的数量在实际场景中可能并不固定. 因此, 本文算法必须适应真实场景的复杂性和可变性. 图 8 比较了 5 种算法在不同移动终端数量下的性能. 图 8 中显示, 延迟和能耗与移动终端数量成正比, 这是因为数据的传输速度和无人机的处理速度都会随着地面终端数量的增加而下降. 图 9 比较了 5 种算法在不同障碍物数量下的性能. 图 9 中折线呈现下降趋势, 因为随着障碍物的增加, 无人机为了更好地避开所有障碍物, 其机动性会随之下降, 因此, 能耗和时延也会减少. 通过图 9 中数据趋势可以看出: 本文所提出的算法使得无人机在飞行过程中使用了最少的时

延和能耗, 验证了本文算法的有效性. MAAC 算法具有最高的延迟和能耗, 因为 AC 算法存在空间复杂度高, 构建时间长的问题. 在延时和能耗方面; MAA3C 算法优于 MAAC 算法, A3C 算法中未使用较大存储空间存储历史经验, 大大加速了采样速度, 但训练十分不稳定, 网络难以收敛; MAPPO 算法和 MADDPG 算法性能相近且优于 MAA3C 算法, DDPG 算法探索环境的方式太过复杂, 且简单地从经验池中抽取数据的方法导致训练较慢且不稳定, PPO 算法在稳定性方面得到了提升, 但算法的时间复杂度较大, 因此, 这两种算法均会导致较高的延时和能耗. 本文算法在 MADDPG 算法的基础上使用复合优先级的方法从经验池中抽取数据, 以获得更加有用的经验, 使得训练速度和稳定性得到了一定的提升, 因此, 本文算法和所有基准算法相比, 具有最低的延迟和能耗.



(a) 总时延

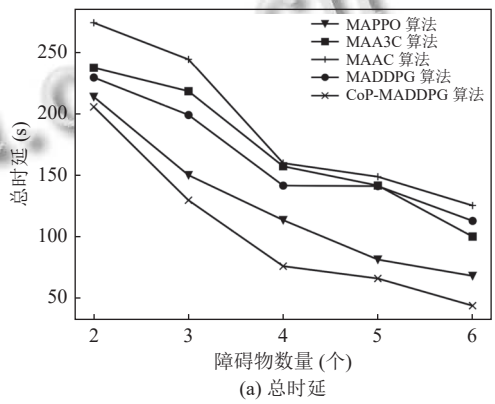


(b) 总能耗

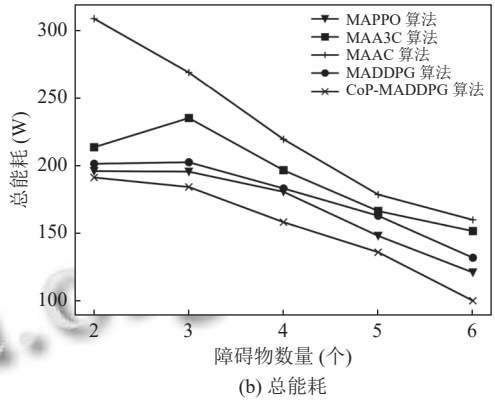
图 8 基于不同数量地面移动终端的性能分析

6 总结

本文针对灾区场景建立了多无人机辅助的移动边缘计算系统模型. 地面移动终端将部分任务卸载到对应的无人机进行处理, 配备 MEC 服务器的无人机在避免碰撞的情况下, 使用最小的时延和能耗为所有地面移动终端提供服务. 本文采用部分可观察马尔可夫决策过程对系统模型进行建模, 并采用 CoP-MADDPG 算法求解目标问题的最优解, 考虑到经验池的状态, 本文在抽样过程中使用了复合优先级, 其中, 复合优先级包括基于立即回报的优先级和基于 TD-error 的优先级. 仿真结果表明, 本文算法的奖励值和无人机覆盖率均优于其他基准算法, 在进行飞行轨迹和任务卸载联合优化时具有更好的性能.



(a) 总时延



(b) 总能耗

图 9 基于不同数量障碍物的性能分析

参考文献

1 王闯, 沈苏彬. 一种基于多智能体的分布式深度神经网络算法. 计算机技术与发展, 2021, 31(12): 45–49, 77.

2 林萌龙, 陈涛, 任棒棒, 等. 基于多智能体深度强化学习的体系任务分配方法. 指挥与控制学报, 2023, 9(1): 93–102.

3 Kiran BR, Sobh I, Talpaert V, *et al*. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909–4926. [doi: 10.1109/TITS.2021.3054625]

4 Ye YJ, Tang Y, Wang HY, *et al*. A scalable privacy-preserving multi-agent deep reinforcement learning approach for large-scale peer-to-peer transactive energy trading. IEEE

Transactions on Smart Grid, 2021, 12(6): 5185–5200. [doi: 10.1109/TSG.2021.3103917]

5 Zhang Y, Yang QY, An D, *et al*. Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid. IEEE Transactions on Cybernetics, 2023, 53(7): 4292–4305. [doi: 10.1109/TCYB.2022.3165074]

6 You XY, Li XJ, Xu YD, *et al*. Toward packet routing with fully distributed multiagent deep reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2): 855–868. [doi: 10.1109/TSMC.2020.3012832]

7 Sacco A, Esposito F, Marchetto G, *et al*. Sustainable task offloading in UAV networks via multi-agent reinforcement learning. IEEE Transactions on Vehicular Technology, 2021, 70(5): 5003–5015. [doi: 10.1109/TVT.2021.3074304]

8 Zhang JD, Yang QM, Shi GQ, *et al*. UAV cooperative air combat maneuver decision based on multi-agent reinforcement learning. Journal of Systems Engineering and Electronics, 2021, 32(6): 1421–1438. [doi: 10.23919/JSEE.2021.000121]

9 厉子凡. 基于多智能体值分解深度强化学习的多智能体协作算法研究 [硕士学位论文]. 合肥: 合肥工业大学, 2021.

10 李盛祥. 基于强化学习的多智能体协同关键技术及应用研究 [博士学位论文]. 郑州: 战略支援部队信息工程大学, 2021.

11 Jiang YY, Mao YX, Wu GX, *et al*. A collaborative optimization strategy for computing offloading and resource allocation based on multi-agent deep reinforcement learning. Computers and Electrical Engineering, 2022, 103: 108278. [doi: 10.1016/j.compeleceng.2022.108278]

12 Wu GW, Xu ZQ, Zhang H, *et al*. Multi-agent DRL for joint completion delay and energy consumption with queuing theory in MEC-based IIoT. Journal of Parallel and Distributed Computing, 2023, 176: 80–94. [doi: 10.1016/j.jpdc.2023.02.008]

13 Seid AM, Boateng GO, Mareri B, *et al*. Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network. IEEE Transactions on Network and Service Management, 2021, 18(4): 4531–4547. [doi: 10.1109/TNSM.2021.3096673]

14 Zhou H, Long YS, Gong SM, *et al*. Hierarchical multi-agent deep reinforcement learning for energy-efficient hybrid computation offloading. IEEE Transactions on Vehicular Technology, 2023, 72(1): 986–1001. [doi: 10.1109/TVT.2022.3202525]

15 Seid AM, Lu JF, Abishu HN, *et al*. Blockchain-enabled task offloading with energy harvesting in multi-UAV-assisted IoT networks: A multi-agent DRL approach. IEEE Journal on Selected Areas in Communications, 2022, 40(12): 3517–3532. [doi: 10.1109/JSAC.2022.3213352]

16 Xue JB, Wu QQ, Zhang HJ. Cost optimization of UAV-MEC network calculation offloading: A multi-agent reinforcement learning method. Ad Hoc Networks, 2022, 136: 102981. [doi: 10.1016/j.adhoc.2022.102981]

17 Li YL, Liang L, Fu JL, *et al*. Multiagent reinforcement learning for task offloading of space/aerial-assisted edge computing. Security and Communication Networks, 2022, 2022: 4193365.

18 Cheng ZP, Liwang MH, Chen N, *et al*. Deep reinforcement learning-based joint task and energy offloading in UAV-aided 6G intelligent edge networks. Computer Communications, 2022, 192: 234–244. [doi: 10.1016/j.comcom.2022.06.017]

19 Zhao XH, Jiang HL, An CY, *et al*. A method of multi-UAV cooperative task assignment based on reinforcement learning. Mobile Information Systems, 2022, 2022: 1147819.

20 Dai ZJ, Zhang Y, Zhang WC, *et al*. A multi-agent collaborative environment learning method for UAV deployment and resource allocation. IEEE Transactions on Signal and Information Processing over Networks, 2022, 8: 120–130. [doi: 10.1109/TSIPN.2022.3150911]

21 Wang GY, Yu XB, Xu FC, *et al*. Task offloading and resource allocation for UAV-assisted mobile edge computing with imperfect channel estimation over Rician fading channels. EURASIP Journal on Wireless Communications and Networking, 2020, 2020(1): 169. [doi: 10.1186/s13638-020-01780-8]

22 闫秋娜, 金思年, 岳殿武, 等. 莱斯信道下低精度 ADC 去蜂窝大规模 MIMO 系统的性能分析. 信号处理, 2022, 38(9): 1903–1911. [doi: 10.16798/j.issn.1003-0530.2022.09.013]

23 Del Prete S, Fuschini F, Barbiroli M. A study on secret key rate in wideband rice channel. Electronics, 2022, 11(17): 2772. [doi: 10.3390/electronics11172772]

24 Hajri N, Khedhiri R, Youssef N. On selection combining diversity in dual-hop relaying systems over double rice channels: Fade statistics and performance analysis. IEEE Access, 2020, 8: 72188–72203. [doi: 10.1109/ACCESS.2020.2986142]

25 Qie H, Shi DX, Shen TL, *et al*. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. IEEE Access, 2019, 7: 146264–146272. [doi: 10.1109/ACCESS.2019.2943253]

26 李波, 越凯强, 甘志刚, 等. 基于 MADDPG 的多无人机协同任务决策. 宇航学报, 2021, 42(6): 757–765.

(校对责编: 孙君艳)