多无人机自适应合作任务卸载决策

刘义[1],罗淳[1],钟伟锋[1],余意[2],欧智清[1]
(1.广东工业大学自动化学院,广东 广州 510000;2.长沙市电子工业学校,湖南 长沙 410016)

摘 要: 研究多无人机(UAV)协同移动边缘计算网络中自适应合作任务卸载与分配问题。为了提高时变环境下无人机之间的协同性,进而提升任务的执行效率,构建时变环境下无人机任务队列模型,并建立基于马尔可夫决策过程(MDP)的无人机任务卸载决策模型。提出一种基于合作的深度确定性策略梯度(CODDPG)算法,以解决多无人机卸载决策优化问题。CODDPG 算法结合神经网络 CommNet 与传统的多智能体深度确定性策略梯度(MADDPG)算法,实现了无人机的环境观测值共享,有效拓展了无人机的环境感知范围并增强了它们之间的协同决策能力,并且解决了 MADDPG 算法中智能体的训练仅依赖局部信息而陷入局部最优解问题,从而最小化总计算时延。经过实验证明,CODDPG 算法不仅有效降低了任务计算时延,而且与传统的 MADDPG 算法相比收敛速度更快。

关键词: 移动边缘计算;多无人机协同;深度强化学习;卸载优化;计算时延最小化
中图分类号: TP393 **文献标志码:** A **DOI:** 10.19678/j.issn.1000-3428.0069899

Adaptive Cooperative Task Offloading Decision for Multiple Unmanned Aerial Vehicles

LIU Yi[1], LUO Chun[1], ZHONG Weifeng[1], YU Yi[2], OU Zhiqing[1]
(1.School of Automation, Guangdong University of Technology, Guangzhou 510000, Guangdong, China;
2.Changsha Electronic Industry School, Changsha 410016, Hunan, China)

【Abstract】 This study investigates adaptive cooperative task offloading and allocation in a multiple Unmanned Aerial Vehicles (UAVs) collaborative mobile edge computing network. To enhance collaboration among UAVs in a time-varying environment and improve the efficiency of task execution, this study constructs a UAV task queuing model in a time-varying environment and establishes a UAVs task offloading decision model based on the Markov Decision Process (MDP). Moreover, this study proposes a Cooperative-based Deep Deterministic Policy Gradient (CODDPG) algorithm to address the optimization problem of multiple UAVs offloading. The CODDPG algorithm, which integrates CommNet with the traditional Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm, facilitates the sharing of environmental observations among all UAVs. This approach effectively extends the UAVs′ perception of the environment and enhances their collaborative decision capability. It also addresses the issue of local optima in the MADDPG algorithm caused by its sole dependence on local information during agent training, thereby minimizing total computation delay. Experimental results demonstrate that the CODDPG algorithm not only significantly reduces task computation delay effectively but also converges faster than the traditional MADDPG algorithm.

【Key words】 mobile edge computing; multiple Unmanned Aerial Vehicles (UAVs) collaboration; deep reinforcement learning; offloading optimization; computation delay minimization

0 引言

随着 5G 技术的广泛应用,计算任务数量呈现爆炸性增长,如交互式应用、虚拟现实、增强现实和自动驾驶等计算密集型应用的兴起,将产生巨大的计算需求[1-3],给云服务器带来严峻的挑战。因此,移动边缘计算应运而生。移动边缘计算的核心思想是将计算和存储资源从集中的云计算中心迁移到网络的边缘,旨在显著降低通信时延,提高数据处理速度,从而优化用户的服务质量[4-6]。在移动边缘计算框架中引入无人机(UAV)充当动态的边缘计算节点能够扩展系统服务的覆盖范围、增强计算处理能

基金项目: 国家重点研发计划(2020YFB1807805,2020YFB1807800);国家自然科学基金(62003099);广东省自然科学基金(2024A1515011795);广州市基础与应用基础研究项目(2023A04J1704)。

作者简介: 刘义,男,教授、博士、博士生导师,主研方向为智能电网、网络资源调度;罗淳,硕士研究生;钟伟锋(通信作者),副教授、博士;余意、欧智清,硕士研究生。

收稿日期: 2024-05-22 **修回日期:** 2024-08-12 **E-mail:** wfzhongs@gdut.edu.cn

计算机工程 www.ecice06.com

力,并提高整体网络的灵活性与效率[7-9]。多数研究致力于优化单个无人机的卸载决策、轨迹规划等问题。文献[10]描述无人机辅助边缘计算的场景,通过优化无人机与用户设备的任务卸载决策,从而减少无人机计算任务时的总能耗。文献[11]研究无人机计算任务卸载、资源分配和无人机轨迹的联合优化问题,同时考虑通信与计算时延的需求,旨在降低任务执行过程中用户设备和无人机的能量消耗。文献[12]站在用户的角度,通过对计算比特分配、时段调度和无人机轨迹等决策的联合优化,从而最小化系统的总能耗。文献[13]联合优化无人机位置、通信和计算资源分配以及任务拆分决策,旨在最小化用户的服务时延与无人机能耗的加权和。然而,在上述研究中,单无人机构成的边缘网络在计算资源和计算能力方面受到限制,难以有效应对大规模数据任务的处理需求。

为了提升无人机边缘网络在计算能力和覆盖范围上的性能,许多学者提出由多无人机构成的边缘网络架构。文献[14]在多无人机移动边缘计算系统中提出一种双层优化方法,该方法首先在上层使用动态规划算法解决无人机任务调度问题,然后在下层使用交替方向乘子法(ADMM)解决无人机计算资源分配和轨迹优化问题,从而最小化用户总能耗。文献[15]考虑由地面和无人机边缘服务器组成的异构多移动边缘计算系统,通过引入前景理论效用函数评估卸载决策对用户的影响,并使用非合作博弈纯纳什均衡算法实现用户满意度最大化。文献[16]在研究多无人机移动边缘计算系统时,进行无人机位置、波束宽度以及终端发射功率、卸载决策和服务器处理任务的周期数的联合优化,结合丁克尔巴赫方法和凸优化理论,提出一种四阶段交替迭代算法,以解决系统计算效率最大化的问题。然而,传统的优化算法在处理动态、非线性以及协作和竞争策略等复杂问题时存在局限性。

2017 年,OpenAI 针对多智能体环境中的混合合作与竞争问题,基于改进的 Actor-Critic[17] 框架提出多智能体深度确定性策略梯度(MADDPG)算法。文献[18]采用 MADDPG 算法独立地管理每架无人机的飞行轨迹,有效地解决多无人机辅助边缘计算中的轨迹优化问题。文献[19]联合优化设备关联、任务卸载和无人机轨迹问题,采用 MADDPG 算法对无人机轨迹进行优化。文献[20]在实时动态的多智能体移动边缘计算场景下,运用 MADDPG 算法解决了总能耗和总时延最小化问题。文献[21]构建多无人机与基站合作移动边缘计算系统模型,采用 MADDPG 算法解决资源分配和卸载决策的协同优化问题,实现任务卸载数量的最大化。

综上所述,在处理多无人机移动边缘计算系统相关问题时,现有算法仍然面临以下的不足和挑战:

1)传统优化算法在大规模无人机网络中可能面临高计算复杂度问题,且缺乏对动态环境的快速适应能力,难以实时响应。

2)MADDPG 算法采用分布式训练方法,使每架无人机根据其观测到的局部环境信息进行独立交互,并通过经验共享来促使智能体协同学习。然而,这种方式可能导致每架无人机在做出动作决策时仅依赖其局部视野,从而可能陷入局部最优。

考虑到现有算法在处理多无人机协同边缘计算网络时的局限性,本文提出一种基于合作的深度确定性策略梯度(CODDPG)算法。在 CODDPG 算法的实现中,采用 CommNet 来扩展无人机对环境的观测范围,从而实现所有无人机的状态观测值共享。

本文主要贡献如下:

1)构建一个多无人机移动边缘计算系统,建立基于离散马尔可夫决策过程(MDP)的无人机卸载决策模型,通过优化决策解决了由任务积压导致的无人机任务处理时延过高问题,从而最小化系统总计算时延。

2)将 CommNet 与 MADDPG 算法相结合,提出 CODDPG 算法,通过扩展无人机对环境的观测范围,使无人机在进行卸载决策时能综合考虑系统的全局状态,不局限于局部信息训练,提升了整个多无人机网络的协同效率。

1 系统模型与问题描述

1.1 系统决策模型

如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同),考虑一个多无人机协同移动边缘计算系统。在系统模型中,用户的位置是固定的,每个用户在时间上随机产生大小不一的计算任务,并将任务传输到无人机进行计算。同时,本文假设无人机与用户之间、无人机之间、无人机与基站之间的通信分别采用不同频段。无人机服务范围表示为无人机接收用户数据的范围,其通信半径设置为 1 km。此外,将无人机之间的通信半径设置为 2 km,并且将无人机与基站的通信半径设置为 5 km。地面用户将计算任务数据上传至无人机,进入无人机的数据队列,等待后续处理。考虑多时段的决策模型,为了更详细地描述系统决策模型,将每个时间段细分为时段的开端、中期和末端 3 个阶段。在时段

的开端，无人机数据队列长度继承上个时段末端的队列长度，同时地面用户随机生成计算任务。在时段的中期，无人机陆续收到地面用户上传的计算任务和其他无人机发送过来的计算任务，任务到达顺序由信道速率决定，同时无人机开始陆续处理队列中的计算任务。在时段的末端：如果无人机的队列中剩余待处理的计算任务数量小于 1，即无人机在单个时段内几乎把所有任务处理完毕，则该无人机无需做额外决策；如果无人机的队列中剩余的计算任务数量大于等于 1，则该无人机需要针对队列末尾的计算任务进行决策。决策类型有 3 种：1)无人机可以选择在本地执行计算任务；2)无人机可以选择将计算任务卸载至其他无人机；3)无人机可以选择将计算任务卸载至地面基站。如果选择决策 2)，被卸载的任务将在下一个时段的中期进入其他无人机的任务队列。通过优化多无人机的任务卸载决策，实现系统中的计算任务重新分配，降低任务处理与等待时延，从而提高用户的服务体验。

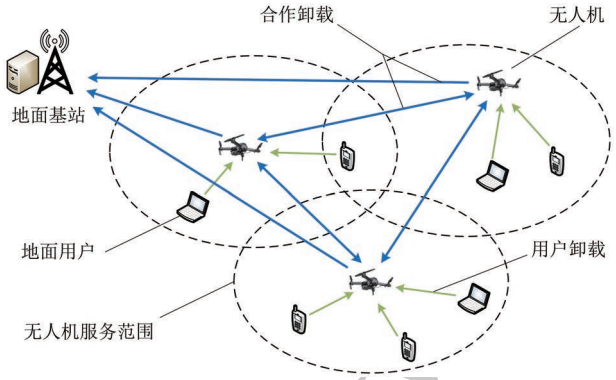

图 1 系统模型

Fig. 1 System model

使用 t 来标记某个时段，有 $t\in T$。用 u 来标记某架无人机，有 $u\in U$，其中 U 为系统中无人机的集合。无人机 u 在时段 t 末端针对队尾计算任务的决策可描述如下：

$$a_{u,t}\in\{0,1\}$$
$$a_{u,\mathrm{bs},t}\in\{0,1\}$$
$$a_{u,u',t}\in\{0,1\},\forall u'\in U\backslash\{u\} \tag{1}$$

式中：$a_{u,t}=1$ 表示无人机 u 在时段 t 执行本地计算；$a_{u,\mathrm{bs},t}=1$ 表示无人机 u 在时段 t 将计算任务卸载到基站；$a_{u,u',t}=1$ 表示无人机 u 在时段 t 将计算任务卸载到无人机 $u'\in U\backslash\{u\}$。

3 种决策不能同时发生，故无人机 u 在时段 t 的决策变量应满足以下等式：

$$a_{u,t}+a_{u,\mathrm{bs},t}+\sum_{u'\in U\backslash\{u\}}a_{u,u',t}=1 \tag{2}$$

1.2 队列模型

考虑到无人机的计算能力有限，设每架无人机只能同时处理一个计算任务，按先后顺序处理多个任务。因此，较晚到达的任务将在无人机的数据队列中等候。设 $Q_{\mathrm{Que},u,t}$ 为无人机 u 在时段 t 开端的队列长度，表示待处理任务所需的总 CPU 周期数，即计算量。在时段 $t+1$ 开端，无人机 u 的队列长度可以描述如下：

$$Q_{\mathrm{Que},u,t+1}=\max\{Q_{\mathrm{Que},u,t}+C_{u,\mathrm{all},t}+C_{u,\mathrm{other},t}-C_{u,\mathrm{out},t}-f_u\Delta t,0\} \tag{3}$$

式中：f_u 为无人机 u 的计算资源；$f_u\Delta t$ 表示无人机 u 在一个时段内可完成的计算量；$C_{u,\mathrm{all},t}$ 和 $C_{u,\mathrm{other},t}$ 分别表示无人机 u 在时段 t 中期收到来自地面用户的计算量和来自其他无人机的计算量；$C_{u,\mathrm{out},t}$ 表示无人机 u 在时段 t 末端向外卸载的计算量。无人机卸载决策与 $C_{u,\mathrm{other},t}$ 和 $C_{u,\mathrm{out},t}$ 有关，表示如下：

$$C_{u,\mathrm{other},t}=\sum_{u'\in U\backslash\{u\}}a_{u,u',t-1}C_{u',t-1} \tag{4}$$

$$C_{u,\mathrm{out},t}=\sum_{u'\in U\backslash\{u\}}(a_{u,u',t}+a_{u,\mathrm{bs},t})C_{u,t} \tag{5}$$

由于任务传输需要一定时间，且无人机在时段末端执行卸载决策，因此式(4)中在时段 t 收到的 $C_{u,\mathrm{other},t}$ 取决于在时段 $t-1$ 末端无人机 u' 传输到无人机 u 的计算量，$C_{u',t-1}$ 表示无人机 u' 在时段 $t-1$ 处于队列末尾的任务的计算量，简称为队尾计算量。式(5)表示无人机 u 在时段 t 末端将队尾计算量卸载至其他无人机 u' 或者基站。如果无人机 u 的卸载决策为本地计算，即 $a_{u,t}=1$，则 $C_{u,\mathrm{out},t}=0$，且队尾计算量 $C_{u,t}$ 已被包含在 $C_{u,\mathrm{all},t}$ 中。

1.3 传输与计算模型

用 m 来标记某个地面用户，设 $r_{m_t,u}$ 为用户 m 到无人机 u 的数据传输速率，有：

$$r_{m_t,u}=W_{m,u}\mathrm{lb}\left(1+\frac{p_m g_{m,u}}{N_0 W_{m,u}+\sum_{m'\in M\backslash\{m\}}p_{m'}g_{m',u}}\right) \tag{6}$$

式中：$W_{m,u}$ 表示用户 m 到无人机 u 的信道带宽；p_m 和 $g_{m,u}$ 分别表示用户 m 的发射功率及无人机 u 与用户 m 之间的信道功率增益；$\sum_{m'\in M\backslash\{m\}}p_{m'}g_{m',u}$ 表示其他用户对用户 m 的干扰功率，M 表示全部用户的集合；N_0 表示加性高斯白噪声功率谱密度。

针对无人机 u 的队尾计算量 $C_{u,t}$，如果 $C_{u,t}$ 在无人机 u 上进行本地处理，则其处理时延 $T_{u,t}$ 可表示如下：

$$T_{u,t}=\frac{D_{u,t}}{r_{m_t,u}}+\frac{C_{u,t}}{f_u}+T_{\mathrm{wait},u,t} \tag{7}$$

式中：$D_{u,t}$ 表示队尾计算量 $C_{u,t}$ 的数据大小，设计算任务 $D_{u,t}$ 来自用户 m_t，则 $D_{u,t}/r_{m_t,u}$ 表示该计算任务的数据传输时延；$C_{u,t}/f_u$ 表示无人机 u 处理该计算任务的计算时延；$T_{\text{wait},u,t}$ 表示计算任务 $D_{u,t}$ 在无人机 u 得到处理之前的等待时间，该时间可在计算任务 $D_{u,t}$ 刚到达队列时通过观察队列长度得到。

当无人机 u 将计算任务卸载至无人机 u' 进行协同计算时，它们之间的数据传输速率可表示如下：

$$r_{u,u'}=W_{u,u'}\text{lb}\left(1+\frac{p_u g_{u,u'}}{\delta^2}\right) \tag{8}$$

式中：$W_{u,u'}$ 表示无人机 u 到无人机 u' 的信道带宽；p_u 和 $g_{u,u'}$ 分别表示无人机 u 的发射功率和无人机 u 与无人机 u' 之间的信道功率增益；δ^2 为加性高斯白噪声功率。

如果无人机 u 把队尾任务卸载到无人机 u' 进行计算，则其总时延 $T_{u,u',t}$ 可表示如下：

$$T_{u,u',t}=\frac{D_{u,t}}{r_{m_t,u}}+\frac{D_{u,t}}{r_{u,u'}}+\frac{C_{u,t}}{f_{u'}}+T_{\text{wait},u',t} \tag{9}$$

式中：$f_{u'}$ 表示无人机 u' 的计算资源；$T_{\text{wait},u',t}$ 表示计算任务 $D_{u,t}$ 在无人机 u' 得到处理之前的等待时间。

当无人机 u 向远处基站传输计算任务时，它们之间的数据传输速率可表示如下：

$$r_{u,\text{bs}}=W_{u,\text{bs}}\text{lb}\left(1+\frac{p_u g_{u,\text{bs}}}{\delta^2}\right) \tag{10}$$

式中：$W_{u,\text{bs}}$ 表示无人机 u 到基站的信道带宽；$g_{u,\text{bs}}$ 表示无人机 u 与基站之间的信道功率增益。

如果计算任务卸载到基站进行辅助计算，则其总时延 $T_{u,\text{bs},t}$ 可表示如下：

$$T_{u,\text{bs},t}=\frac{D_{u,t}}{r_{m_t,u}}+\frac{D_{u,t}}{r_{u,\text{bs}}}+\frac{C_{u,t}}{f_{\text{bs}}} \tag{11}$$

式中：f_{bs} 表示基站服务器用于处理任务的计算资源。

这里假设基站的服务器拥有足够多的计算资源，能够快速地完成收到的计算任务，故忽略任务排队时间。

因此，在时段 t 末端无人机 u 队尾计算任务的总时延可以表示如下：

$$\overline{T}_{u,t}=a_{u,t}T_{u,t}+\sum_{u'\in U\setminus\{u\}}(a_{u,u',t}T_{u,u',t}+a_{u,\text{bs},t}T_{u,\text{bs},t}) \tag{12}$$

1.4 问题描述

本文通过优化多无人机的任务卸载决策，最小化多时段无人机队尾计算任务的总处理时延。因为队尾计算任务的处理速度能够反映整个队列的拥塞程度，所以这里仅考虑队尾任务的时延。因此，系统优化问题表示如下：

$$\begin{aligned}
&\text{P0}:\min_{a}\sum_{t\in T}\sum_{u\in U}\overline{T}_{u,t}\\
&\text{s.t. C1}:a_{u,t}\in\{0,1\},\forall u\in U,t\in T\\
&\quad\text{C2}:a_{u,u',t}\in\{0,1\},\forall u\in U,\\
&\qquad u'\in U\setminus\{u\},t\in T\\
&\quad\text{C3}:a_{u,\text{bs},t}\in\{0,1\},\forall u\in U,t\in T\\
&\quad\text{C4}:a_{u,t}+a_{u,\text{bs},t}+a_{u,u',t}=1,\\
&\qquad\forall u\in U,u'\in U\setminus\{u\},t\in T\\
&\quad\text{C5}:Q_{\text{Que},u,t}\leqslant Q_{\text{Que},\max},\forall u\in U,t\in T
\end{aligned} \tag{13}$$

式中：决策变量 $a=\{a_{u,t},a_{u,\text{bs},t},\{a_{u,u',t}\}_{u'\in U\setminus\{u\}}\}$，$u\in U$，$t\in T$，包含无人机本地计算、远处基站辅助计算、其他无人机辅助计算 3 种决策，且均为二进制变量；C4 表示无人机在时段 t 末端任务卸载决策的唯一性；C5 表示无人机 u 的队列长度不能超过其上限值 $Q_{\text{Que},\max}$；问题 P0 只含有 0-1 变量，但无法用传统的整数规划方法直接求解，因为系统中地面用户产生的计算任务具有随机性，决策者无法事先知道实际的计算任务量。

计算任务大小及卸载决策共同影响计算任务到达无人机队列的时间点，进而影响其等待时间。针对多无人机协同任务卸载场景中存在动态变化的环境因素，本文考虑使用启发式算法解决问题 P0。然而，传统的启发式算法在动态变化和不确定性较高的环境中，难以保证获得高质量的可行解，因此本文设计基于 MADDPG 和神经网络 CommNet 的 CODDPG 算法。下一节将描述如何使用 CODDPG 算法解决问题 P0。

2 基于深度强化学习的解决方案

2.1 MDP 模型

MDP 模型能够有效地处理决策过程中的不确定性和动态性[22]，适用于描述无人机在时变环境中的动态任务卸载决策过程。本文将每架无人机的卸载决策过程建模为离散的 MDP 模型[23]。MDP 模型包含状态、动作、奖励和转移概率等要素，用来描述在有限时间内智能体如何通过一系列动作来达到优化目标。接下来，对优化问题 P0 采用的 MDP 模型的状态、动作、奖励函数以及转移概率进行描述。

1）状态，系统状态定义为无人机队列信息，因此时段 t 的状态 S_t 可描述如下：

$$S_t=\{Q_{\text{Que},1,t},Q_{\text{Que},2,t},\cdots,Q_{\text{Que},l,t}\} \tag{14}$$

式中：S_t 为 l 维集合，$l=|U|$，描述当前时段无人机队列情况。

无人机 u 在时段 t 的状态 $s_{u,t}$ 可以被定义为

$s_{u,t}=Q_{\mathrm{Que},u,t}$，即 $s_{u,t}\in S_t=\{s_{1,t},s_{2,t},\cdots,s_{l,t}\}$。

2)动作，在时段 t，无人机可以根据不同的状态，作出不同的卸载决策。因此，将无人机集群的动作集合定义为 A_t，可描述如下：

$$A_t=\{(a_{1,t},a_{1,2,t},\cdots,a_{1,l,t},a_{1,\mathrm{bs},t}),\cdots,(a_{l,t},a_{l,1,t},\cdots,a_{l,l-1,t},a_{l,\mathrm{bs},t})\} \tag{15}$$

3)奖励函数，奖励值与目标函数值和无人机任务队列长度有关。将无人机 u 在时段 t 的奖励值定义如下：

$$rw_{u,t}=-\overline{T}_{u,t}-\rho\cdot\max(Q_{\mathrm{Que},\max}-Q_{\mathrm{Que},t},0) \tag{16}$$

式中：$\overline{T}_t^u$ 为无人机 u 在时段 t 队尾计算任务的时延；$\max(Q_{\mathrm{Que},\max}-Q_{\mathrm{Que},t},0)$表示当无人机队列长度大于队列上限值 $Q_{\mathrm{Que},\max}$ 时给系统带来的负奖励值；ρ 为权重系数。

用 $R_t=\sum_{u\in T}rw_{u,t}$ 表示在时段 t 无人机集群的奖励值总和。鉴于深度强化学习算法的目标是寻找最大化累积奖励的策略，在计算时延和队列长度惩罚项前引入负号，以将惩罚最小化问题转化为奖励最大化问题。

4)转移概率，在动态时变的环境中，转移概率对无人机在当前状态下选取卸载决策起着关键的作用，可将转移概率定义为 $\mathcal{P}(S_{t+1}|S_t,A_t)$。

2.2 基于 CommNet 与 MADDPG 的 CODDPG

针对大规模状态空间及动作空间的问题，本文采用深度强化学习算法对 MDP 模型进行求解。MADDPG 算法是为多智能体问题设计的深度强化学习算法，因此它被视为 DDPG 算法的扩展。DDPG 算法使用一个 Critic 网络来估计状态-动作对的 Q 值，并通过 Actor 网络学习确定性的策略，其目标是最大化累积奖励的预期值。在 MADDPG 中，每个智能体都有自己的 Actor 和 Critic 网络[24]，通过学习来实现性能的最大化。在 MADDPG 中，智能体的 Critic 网络首先通过接收来自其他智能体 Actor 网络输出的动作信息，将全局信息整合到估计的 Q 值中。然后通过时间差分算法最小化 Q 值的均方误差来更新 Critic 网络。Critic 网络的更新公式如下：

$$L(\omega_i)=\mathbb{E}_{s,a,r,s'}[(Q_{\mu,i}(s,a_1,a_2,\cdots,a_l)-\varphi)^2] \tag{17}$$

$$\varphi=R_i+\gamma Q_{\mu',i}(s',a'_1,a'_2,\cdots,a'_l)|_{a'_j=\mu'_j(o_j)} \tag{18}$$

接着每个智能体通过最大化其 Critic 网络估计的 Q 值来更新自己的 Actor 网络。Actor 网络的更新公式如下：

$$\nabla_{\omega_i}J(\theta_i)=\mathbb{E}_{s,a\sim F}[\nabla_{\omega_i}\theta_i(a_i|o_i)\nabla_{a_i}Q_{\theta,i}(s,a_1,a_2,\cdots,a_l)|a_i=\theta_i(o_i)] \tag{19}$$

MADDPG 算法中每个智能体使用自己的 Actor-Critic 网络进行训练，再通过经验共享去解决智能体协同问题[25-26]。然而，MADDPG 算法对于 Actor-Critic 网络的训练方式是建立在局部信息的基础上进行的，每个智能体在局部环境下作出决策。本文提出一种基于 CommNet 改进的 CODDPG 算法，如图 2 所示。该算法通过 CommNet 使所有无人机的环境观测值共享，从而扩大无人机的环境观测范围，以促进无人机的协同决策。

如图 3 所示，基于 CommNet 的多智能体训练神经网络框架是一个由 k 层隐藏层组成的多层神经网络。

由图 3 右侧隐藏层可知，每层隐藏层输出两个变量，分别用 $h_{k,u}$ 和 $v_{k,u}$ 表示隐藏状态和通信变量。每架无人机的通信变量通过 Mean 计算之后，得到的结果单独与各自无人机的隐藏状态进行 tanh 函数激活运算，随后得出下一层的隐藏状态 $h_{k+1,u}$。本文将下一层的隐藏状态 $h_{k+1,u}$ 和下一层的通信变量 $v_{k+1,u}$ 描述如下：

$$h_{k+1,u}=G_k(h_{k,u},v_{k,u}),\ k\in\{0,1,2\} \tag{20}$$

$$v_{k+1,u}=\frac{1}{U-1}\sum_{u'\in U\backslash\{u\}}h_{k,u'},\ k\in\{0,1,2\} \tag{21}$$

式中：G_k 表示一个线性层，其后紧跟一个非线性激活函数 σ。

因此，设 $\boldsymbol{H}_k$ 和 $\boldsymbol{V}_k$ 分别是 $h_{k,u}$ 和 $v_{k,u}$ 的系数矩阵，可以将式(20)转换描述如下：

$$h_{k+1,u}=\sigma(\boldsymbol{H}_kh_{k,u}+\boldsymbol{V}_kv_{k,u})=\tanh(\boldsymbol{H}_kh_{k,u}+\boldsymbol{V}_kv_{k,u}),k\in\{1,2,3\} \tag{22}$$

由式(21)可知，通信变量 $v_{k,u}$ 表示为除了第 u 架无人机以外其他无人机的隐藏状态之和，因此设 $\boldsymbol{h}_k$ 表示当前无人机群的状态向量集合，用矩阵描述如下：

$$\boldsymbol{h}_k=[h_{k,1},h_{k,2},\cdots,h_{k,l}]^{\mathrm{T}},\ k\in\{0,1,2\} \tag{23}$$

设 $\overline{\boldsymbol{V}}_k$ 表示第 u' 个无人机的系数矩阵，描述如下：

$$\overline{\boldsymbol{V}}_k=\frac{\boldsymbol{V}_k}{U-1} \tag{24}$$

因此，将 $\boldsymbol{H}_k$ 和 $\overline{\boldsymbol{V}}_k$ 组成矩阵 $\boldsymbol{B}_k$，表示如下：

$$\boldsymbol{B}_k=\begin{bmatrix}\boldsymbol{H}_k & \overline{\boldsymbol{V}}_k & \cdots & \overline{\boldsymbol{V}}_k\\ \overline{\boldsymbol{V}}_k & \boldsymbol{H}_k & \cdots & \overline{\boldsymbol{V}}_k\\ \vdots & \vdots & & \vdots\\ \overline{\boldsymbol{V}}_k & \overline{\boldsymbol{V}}_k & \cdots & \boldsymbol{H}_k\end{bmatrix} \tag{25}$$

www.ecicis.com 计算机工程

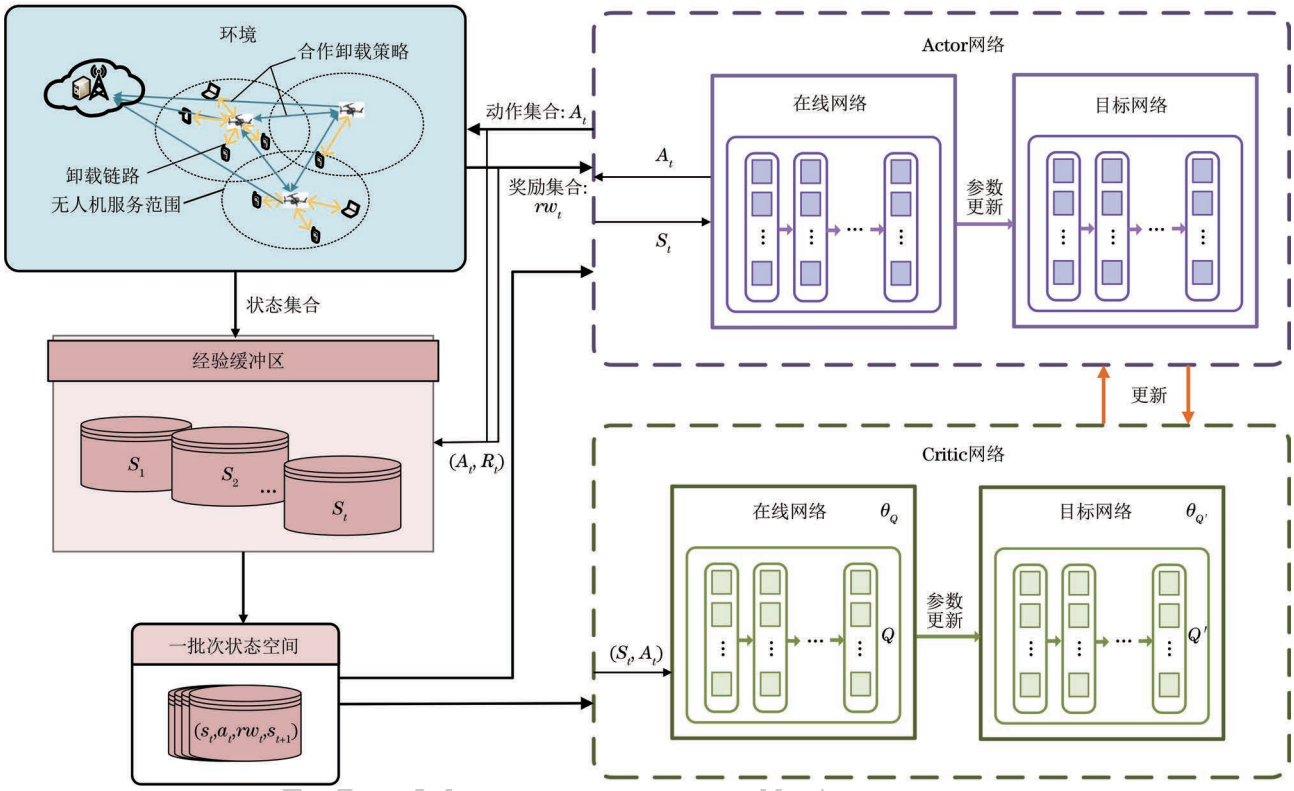

图 2 CODDPG 算法训练框架

Fig. 2 CODDPG algorithm training framework

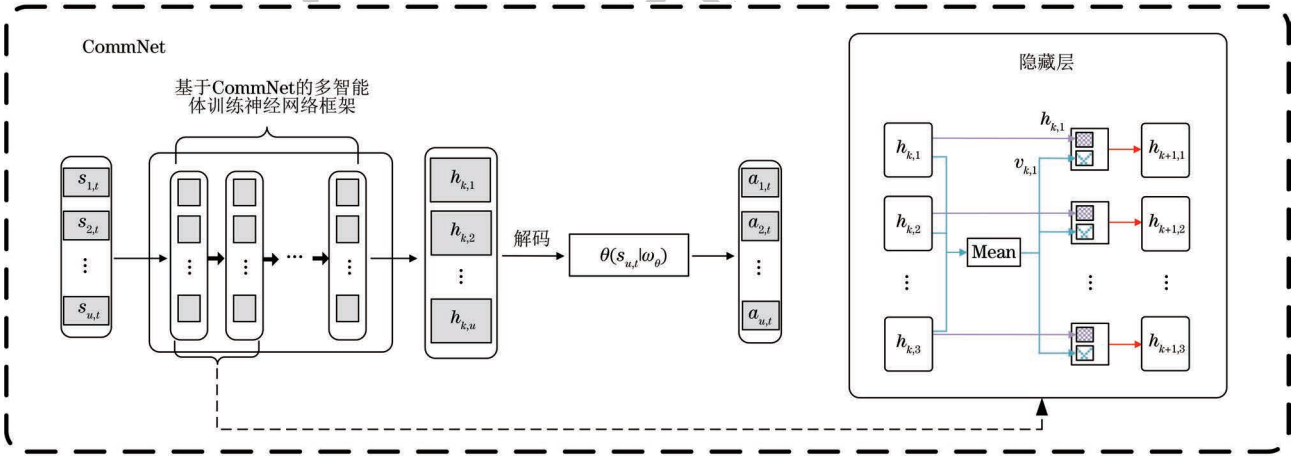

图 3 CommNet 训练框架

Fig. 3 CommNet training framework

式中：矩阵 $\boldsymbol{B}_k$ 的大小可以根据无人机集群的数量进行动态调整[27]。

当无人机因电量不足而需要与备用无人机进行更替时，或者在需要增派无人机进行任务部署时，无人机集群的数量可能会发生变化，矩阵 $\boldsymbol{B}_k$ 的大小也随之变化。

首先，初始化基于神经网络 CommNet 修改后的 Actor 网络和 Critic 网络的主网络以及目标网络的权重参数 ω_θ、ω_Q、$\omega_{\theta'}$ 和 $\omega_{Q'}$。其次，进入周期训练，通过每架无人机的实时状态，Actor 网络使用初始化的策略 $\theta(s_{u,t}|\omega_\theta)$ 进行第一轮的动作选择，在得到第一轮的转换对 (S_t, A_t, R_t, S_{t+1}) 后，将其存放于经验缓冲区 $\mathcal{D}$。再次，Critic 主网络通过 CommNet 模拟的贝尔曼方程，计算出当前策略 $\theta(s_{u,t}|\omega_\theta)$ 的 Q 值。在完成上述步骤后，在 $\mathcal{D}$ 中随机抽取一小批转换对 F，描述如下：

$$F=\{(S_t, A_t, R_t, S_{t+1})\}_{t=1}^{\mathrm{F}} \tag{26}$$

然后，通过 Actor 目标网络的策略 $\theta'(s_{u,t}|\omega_{\theta'})$ 选出训练后的动作 A_{t+1}，Critic 目标网络计算目标 Q 值，以训练 Critic 主网络。Critic 主网络的更新公

式可描述如下：

$$\varphi_t = r w_{u,t} + \gamma Q'(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_{Q'}) \tag{27}$$

$$L(\omega_Q) = \frac{1}{F}\sum_{t=1}^{F}[\varphi_t - Q(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_Q)]^2 \tag{28}$$

式中：F 表示转换对的样本数量。

根据 Critic 主网络的评估结果，更新 Actor 主网络的策略函数 $\theta(s_{u,t}|\omega_\theta)$，策略梯度更新公式可描述如下：

$$\nabla_{\omega_\theta} J(\theta) = \frac{1}{F}\sum_{t=1}^{F}[\nabla_{\omega_\theta}\theta(s_{u,t}|\omega_\theta)\nabla_{a_{u,t}} Q(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_Q)] \tag{29}$$

最后，更新 Actor 和 Critic 目标网络权重参数：

$$\omega_{Q'} = \eta\omega_Q + (1-\eta)\omega_{Q'} \tag{30}$$

$$\omega_{\theta'} = \eta\omega_\theta + (1-\eta)\omega_{\theta'} \tag{31}$$

采用策略参数化函数 Softmax，通过将 Actor 网络输出的连续动作值转换为动作选择的概率分布，以适应实时环境下的离散动作决策需求，满足约束条件 C_4。

算法 1 CODDPG 算法

输入 无人机群的状态集合 S_t

输出 无人机群的动作集合 A_t，奖励值 R_t，卸载延时 T_t

1. 随机初始化基于 CommNet 改进的 Actor 主网络的 $\theta(s_{u,t}|\omega_\theta)$ 和 Critic 主网络的 $Q(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_Q)$，以及它们的权重 ω_θ 和 ω_Q；

2. 通过 ω_θ 和 ω_Q，初始化 Actor 和 Critic 目标网络的 $\theta'(s_{u,t}|\omega_{\theta'})$ 和 $Q'(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_{Q'})$ 以及它们的权重 $\omega_{\theta'}$ 和 $\omega_{Q'}$；

3. for episode= 1, 2, … do

4. 初始化：用户位置和无人机位置等系统环境参数

5. for t= 1, 2, … do

6. 获取每架无人机的实时状态 $s_{u,t}$

7. 通过 CommNet，根据全局状态 θ，选取每架无人机的动作决策 $a_{u,t}$，得到动作集合 A_t

8. 在此时段的仿真环境中完成动作后，获得新状态集 S_{t+1}

9. 根据式(16)获得奖励值 R_t

10. 将转换对 (S_t, A_t, R_t, S_{t+1}) 存放于经验缓冲区 $\mathcal{D}$

11. if 时间步长达到设置的更新周期 do

12. 在 $\mathcal{D}$ 中随机抽取一小批转换对 F：

$F = \{(S_t, A_t, R_t, S_{t+1})\}_{t=1}^{F}$

13. 在目标网络中选择动作进行训练：

$A'_t \sim \theta'(s_{u,t}|\omega_{\theta'})$

14. 计算目标值：

$\varphi_t = rw_{u,t} + \gamma Q'(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_{Q'})$

15. 更新 Critic 主网络的 Q 值：

$L(\omega_Q) = \frac{1}{F}\sum_{t=1}^{F}[\varphi_t - Q(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_Q)]^2$

16. 更新 Actor 主网络的 $\theta(s_{u,t}|\omega_\theta)$：

$\nabla_{\omega_\theta} J(\theta) = \frac{1}{F}\sum_{t=1}^{F}[\nabla_{\omega_\theta}\theta(s_{u,t}|\omega_\theta)\nabla_{a_{u,t}} Q(s_{u,t}, \{a_{u,t}, a_{u,1,t}, a_{u,2,t}, \cdots, a_{u,l,t}, a_{u,\mathrm{bs},t}\} | \omega_Q)]$

17. 更新 Actor 和 Critic 目标网络权重参数：

$\omega_{Q'} = \eta\omega_Q + (1-\eta)\omega_{Q'}$

$\omega_{\theta'} = \eta\omega_\theta + (1-\eta)\omega_{\theta'}$

18. end if

19. end for

20. end for

3 仿真及结果分析

构建由 1 个基站和 3 架无人机组成的移动边缘计算网络。为了评估 CODDPG 算法在多无人机场景下任务卸载决策选择的有效性，选取 3 种固定的计算卸载方案以及 MADDPG 算法，与 CODDPG 算法进行对比分析，评估各自的性能和应用效果。3 种固定的计算卸载方案分别为：

1)本地计算策略(LOS)，即无人机在任何队列等待情况下均自行处理计算任务。

2)其他无人机辅助计算策略(OUAS)，即无人机将接收的计算任务随机转发给其他无人机处理。

3)基站边缘服务器辅助计算策略(EOS)，即无人机将所有接收到的计算任务卸载给边缘基站处理。

系统仿真参数设置如表 1 所示。

表 1 仿真参数设置

Table 1 Simulation parameter setting

参数名称	参数符号	参数值
无人机计算功率/W	$p_{u,\mathrm{com}}$	2
无人机发射功率/W	$p_u, p_{u'}$	[0.2,1.0]
无人机与边缘基站之间的信道增益/dB	$g_{u,\mathrm{bs}}$	−400
无人机与边缘基站之间的信道带宽/MHz	$W_{u,\mathrm{bs}}$	5
无人机处理器的计算资源/MHz	$f_u, f_{u'}$	100
用户发射功率/W	p_m	[0.1,0.5]
用户与无人机之间的信道增益/dB	$g_{m,u}$	−40
用户与无人机之间的信道带宽/MHz	$W_{m,u}$	1
基站设备处理器的计算资源/GHz	f_{bs}	2
无人机与无人机之间的信道增益/dB	$g_{u,u'}$	−20
无人机与无人机之间的信道带宽/MHz	$W_{u,u'}$	2
加性高斯白噪声功率/W	δ^2	10^{-9}
加性高斯白噪声功率谱密度/(dBm·Hz^{-1})	N_0	0.1
任务所需的 CPU 周期数/(10^9 cycle)	C	[1,5]
任务大小/MB	D	[100,500]
折扣因子	γ	0.99
学习率	μ	0.000 1

计算机工程 www.ecice0.com

3.1 收敛性对比

本文首先对 CODDPG 算法与 MADDPG 算法在计算卸载决策中的奖励值进行对比。如图 4 所示，可以观察到 CODDPG 算法在完成 200 次迭代之后开始收敛。相比之下，MADDPG 算法则需进行约 600 次迭代才开始明显收敛。在相同环境条件下，CODDPG 算法收敛后所获得的奖励显著高于 MADDPG 算法所达到的水平。

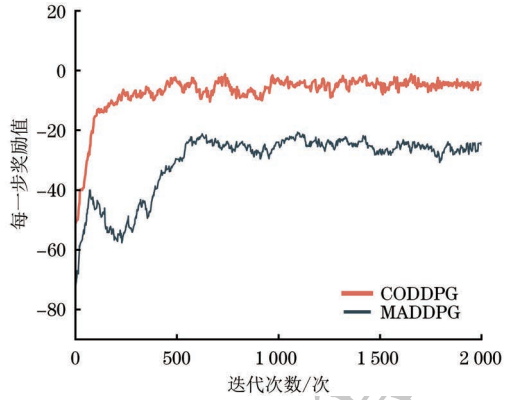

图 4 收敛性比较

Fig. 4 Convergence comparison

3.2 时延分析

在仿真中设定一个任务数量值，当系统中所有无人机完成的任务数量达到这个值时，则结束本轮训练。在图 5 中，针对不同的计算任务数量值进行多种算法和方案的计算时延对比。可以看出，随着任务数量的增加，CODDPG 算法的计算时延始终低于 MADDPG 算法及其他卸载方案，CODDPG 算法的增长率为 60%，MADDPG 算法的增长率为 140%，LOS 的增长率为 150%，OUAS 的增长率为 300%，EOS 的增长率为 350%。综合数据，随着任务数量的上升，CODDPG 算法在计算时延方面相较于上述算法及卸载方案分别降低了 133.33%、150.00%、400.00%和 483.33%。CODDPG 算法将无人机集群置于一个全局状态共享的环境下训练，通过扩展无人机的观测范围，加强无人机的协同性，使得在任务数量较大时，CODDPG 算法的卸载策略选择优于 MADDPG 算法以及其余 3 种卸载方案。

图 6 描绘了无人机数量对 CODDPG 算法、MADDPG 算法以及其余 3 种卸载方案计算时延的影响。如图 6 所示，随着无人机数量的增加，两种算法以及 3 种卸载方案计算所需时延随之降低。然而，本文提出的 CODDPG 算法在计算时延方面，始终优于其他算法及卸载方案。这归因于 CODDPG 算法利用 CommNet 增强无人机在动态环境中的状态观测能力，进而缩短任务的等待时间并降低计算任务所需的时延。

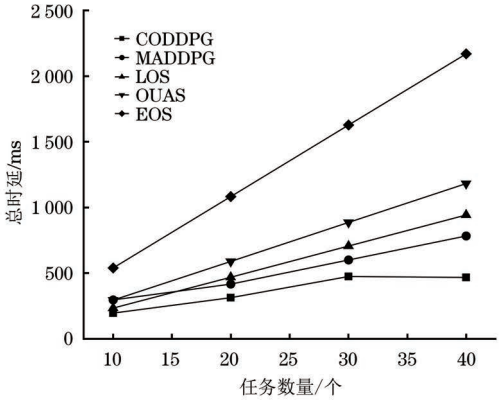

图 5 任务数量对计算时延的影响

Fig. 5 Impact of task quantity on computation delay

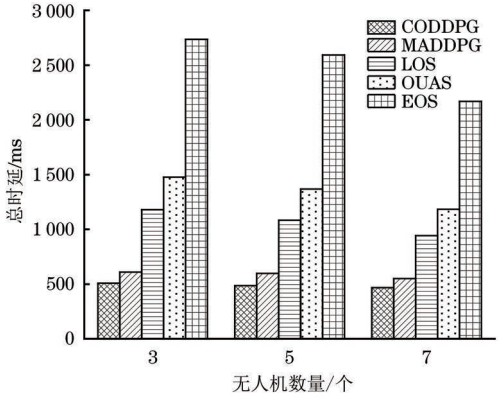

图 6 无人机数量对计算时延的影响

Fig. 6 Impact of UAV quantity on computation delay

3.3 能耗影响

本节通过改变无人机的发射功率，分析其对计算时延以及能耗的影响。无人机 u 的计算功率为 $p_{u,\mathrm{com}}$、发射功率为 p_u。无人机选择本地计算的主要能耗为计算能耗，选择传输到基站或者其他无人机辅助计算时的主要能耗为传输能耗。因此，将无人机的计算能耗定义为 E_{com}，可描述如下：

$$E_{\mathrm{com}}=p_{u,\mathrm{com}}\cdot\frac{C_{u,t}}{f_u} \tag{32}$$

式中：$\frac{C_{u,t}}{f_u}$表示无人机本地计算所需时间。

针对无人机的 3 个决策构建对应能耗模型，描述如下：

$$E_{\mathrm{local}}=E_{\mathrm{com}} \tag{33}$$

$$E_{u,u',t}=p_u\cdot\frac{D_{u,u',t}}{r_{u,u',t}} \tag{34}$$

$$E_{u,\mathrm{bs},t}=p_u\cdot\frac{D_{u,\mathrm{bs},t}}{r_{u,\mathrm{bs},t}} \tag{35}$$

式中：式(33)表示本地计算能耗；式(34)表示传输到

其他无人机计算所需能耗；式(35)表示传输到基站辅助计算所需能耗。

基于能耗模型，分别将无人机的发射功率设置为 0.2、0.4、0.6、0.8 以及 1.0 W 进行测试，探究无人机的发射功率对时延和能耗的影响。如图 7 所示，由于无人机发射功率与传输时延成反比，在无人机的发射功率从 0.2 W 逐渐增加到 1.0 W 的过程中，5 种算法与方案的计算总时延呈现下降的趋势。此外，可以明显观察到 CODDPG 算法相比其他算法与方案具有更低的计算时延。这进一步验证了 CODDPG 算法能加强无人机之间的协同工作效率，使得系统模型具有更低的计算时延。

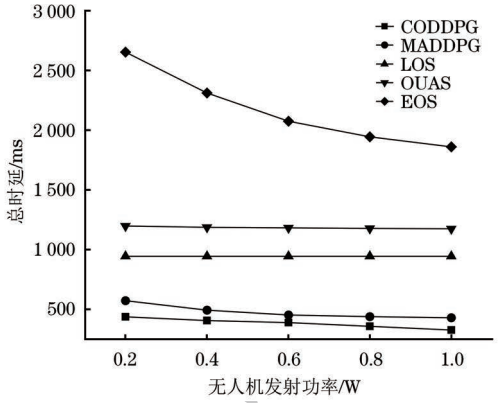

图 7 无人机发射功率对计算时延的影响

Fig. 7 Impact of UAV transmitting power on computation delay

尽管增加无人机的发射功率可以有效降低系统时延，但根据式(32)，发射功率与能耗成正比关系。因此，需要对不同发射功率下无人机的能耗进行数据分析，以评估其对系统整体性能的影响。如图 8 所示，在无人机发射功率增大的情况下，CODDPG 算法、MADDPG 算法、EOS 和 OUAS 方案的能耗随之上升。在图7和图8中，LOS方案仅在本机进行计算，因此无人机发射功率的变化对其计算时延和能耗基本没有影响。由图 8 可以看出，在无人机发射功率由 0.2 W 上升到 1.0 W 的过程中，CODDPG 算法、MADDPG 算法、OUAS 和 EOS 方案的平均能耗变化分别为 284、298、500 和 367 J。综合以上数据，在提高无人机发射功率后，CODDPG 算法具有更低的能耗，并加强了系统的可调节性。

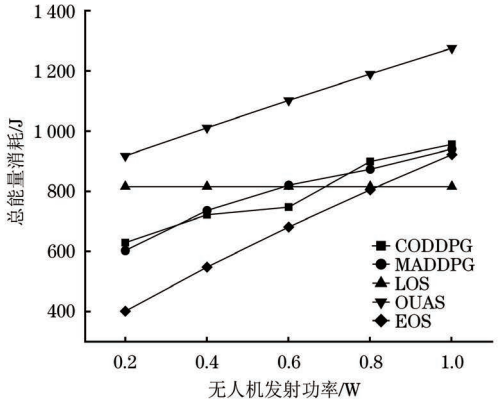

图 8 无人机发射功率对能耗的影响

Fig. 8 Impact of UAV transmitting power on energy consumption

4 结束语

本文关注多无人机协同边缘计算中的任务卸载决策优化问题。针对多无人机协同状态的时变性，首先采用离散的 MDP 对无人机卸载决策模型进行建模，然后提出一种结合神经网络 CommNet 与 MADDPG 的多智能体合作深度强化学习算法 CODDPG。该算法通过共享状态观测值对多无人机进行训练，扩展无人机的环境感知能力，从而实现无人机间自主协同卸载决策训练，并通过优化每架无人机在时变环境中的卸载决策，最小化总计算时延。经过仿真验证，本文提出的 CODDPG 算法在任务数量和无人机集群数量不同的情况下，与 MADDPG 算法及 LOS、OUAS 和 EOS 方案相比，CODDPG 算法在计算时延方面分别平均提高了 54%、150%、213%和 582%的性能，具有更快的收敛速度。

参考文献

[1] PAN W B, LIU Y, YANG C. Adaptive task offloading of rechargeable UAV edge computing network based on double decision value iteration [J]. Computer Communications, 2022, 193: 136-145.

[2] ZHANG L, ANSARI N. Optimizing the operation cost for UAV-aided mobile edge computing [J]. IEEE Transactions on Vehicular Technology, 2021, 70(6): 6085-6093.

[3] WANG J R, LIU K Y, PAN J P. Online UAV-mounted edge server dispatching for mobile-to-mobile edge computing [J]. IEEE Internet of Things Journal, 2020, 7(2): 1375-1386.

[4] 徐渊博，任静，王亮，等. 面向无人机辅助边缘计算的卸载任务准入控制机制[J]. 计算机工程，2025，51(2)：54-64.
XU Y B, REN J, WANG L, et al. Admission control mechanism for offloading tasks in UAV-assisted edge computing[J]. Computer Engineering, 2025, 51(2): 54-64. (in Chinese)

[5] 陈卓，操民涛，周致圆，等. 移动边缘计算中基于图到序列深度强化学习的复杂任务部署策略[J]. 通信学报，2024，45(3)：244-257.
CHEN Z, CAO M T, ZHOU Z Y, et al. Graph-to-sequence deep reinforcement learning based complex task deployment strategy in MEC [J]. Journal on Communications, 2024, 45(3): 244-257. (in Chinese)

[6] 彭世明，林士飚，贾硕，等. 基于负载预测的多目标优化任务卸载策略[J]. 计算机工程，2024，50(1)：206-215.
PENG S M, LIN S Y, JIA S, et al. Multi-objective optimization task offloading strategy based on load prediction [J]. Computer

Engineering, 2024, 50(1): 206-215. (in Chinese)
[7] HU Q Y, CAI Y L, YU G D, et al. Joint offloading and trajectory design for UAV-enabled mobile edge computing systems[J]. IEEE Internet of Things Journal, 2019, 6(2): 1879-1892.
[8] YANG Z H, PAN C H, WANG K Z, et al. Energy efficient resource allocation in UAV-enabled mobile edge computing networks [J]. IEEE Transactions on Wireless Communications, 2019, 18(9): 4576-4589.
[9] CHENG N, XU W C, SHI W S, et al. Air-ground integrated mobile edge networks: architecture, challenges, and opportunities [J]. IEEE Communications Magazine, 2018, 56(8): 26-32.
[10] XIONG J Y, GUO H Z, LIU J J. Task offloading in UAV-aided edge computing: bit allocation and trajectory optimization [J]. IEEE Communications Letters, 2019, 23(3): 538-541.
[11] TUN Y K, PARK Y M, TRAN N H, et al. Energy-efficient resource management in UAV-assisted mobile edge computing[J]. IEEE Communications Letters, 2021, 25(1): 249-253.
[12] ZHANG T K, XU Y, LOO J, et al. Joint computation and communication design for UAV-assisted mobile edge computing in IoT [J]. IEEE Transactions on Industrial Informatics, 2020, 16(8): 5505-5516.
[13] YU Z, GONG Y M, GONG S M, et al. Joint task offloading and resource allocation in UAV-enabled mobile edge computing[J]. IEEE Internet of Things Journal, 2020, 7(4): 3147-3159.
[14] LUO Y Z, DING W R, ZHANG B C. Optimization of task scheduling and dynamic service strategy for multi-UAV-enabled mobile-edge computing system [J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(3): 970-984.
[15] APOSTOLOPOULOS P A, FRAGKOS G, TSIROPOULOU E E, et al. Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty [J]. IEEE Transactions on Mobile Computing, 2023, 22(1): 175-190.
[16] DENG X H, ZHAO J H, KUANG Z F, et al. Computation efficiency maximization in multi-UAV-enabled mobile edge computing systems based on 3D deployment optimization[J]. IEEE Transactions on Emerging Topics in Computing, 2023, 11(3): 778-790.
[17] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [EB/OL]. [2024-04-11]. http://arxiv.org/abs/1706.02275.
[18] WANG L, WANG K Z, PAN C H, et al. Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(1): 73-84.
[19] CHEN J X, YANG P, REN S Q, et al. Enhancing AIoT device association with task offloading in aerial MEC networks [J]. IEEE Internet of Things Journal, 2024, 11(1): 174-187.
[20] GONG Y K, YAO H P, WANG J J, et al. Multi-agent driven resource allocation and interference management for deep edge networks [J]. IEEE Transactions on Vehicular Technology, 2022, 71(2): 2018-2030.
[21] PENG H X, SHEN X M. Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks [J]. IEEE Journal on Selected Areas in Communications, 2021, 39(1): 131-141.
[22] KIRAN B R, SOBH I, TALPAERT V, et al. Deep reinforcement learning for autonomous driving: a survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.
[23] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge, USA: MIT press, 2018.
[24] DEGRIS T, WHITE M, SUTTON R S. Off-policy actor-critic[EB/OL]. [2024-04-11]. https://arxiv.org/abs/1205.4839.
[25] MALEKI M R, MILI M R, JAVAN M R, et al. Multi-agent reinforcement learning trajectory design and two-stage resource management in CoMP UAV VLC networks [J]. IEEE Transactions on Communications, 2022, 70(11): 7464-7476.
[26] CHENG Z P, MIN M H, LIWANG M H, et al. Multiagent DDPG-based joint task partitioning and power control in fog computing networks [J]. IEEE Internet of Things Journal, 2022, 9(1): 104-116.
[27] SUKHBAATAR S, ARTHUR S, ROB F. Learning multiagent communication with backpropagation [EB/OL]. [2024-04-11]. https://arxiv.org/abs/1605.07736.

文字编辑 陆燕菲
栏目编辑 赖玉玲

计算机工程 www.ecice06.com