刘鑫, 赵莎莎. 基于 MARL 的无人机边缘计算任务卸载优化[J]. 智能计算机与应用,2024,14(9):170-178. DOI:10.20169/j.issn.2095-2163.240927

基于 MARL 的无人机边缘计算任务卸载优化

刘 鑫, 赵莎莎
(南京邮电大学 通信与信息工程学院, 南京 210003)

摘 要: 针对单个无人机覆盖范围有限、计算能力不足的问题,提出了联合边缘云的多无人机移动边缘计算系统;为了提高物联网设备的任务处理成功率和结果新鲜度,提出了基于多智能体强化学习(Multi-Agent Reinforcement Learning, MARL)的优化方案,旨在通过联合优化物联网设备的卸载选择、无人机飞行轨迹以及无人机任务卸载,来最大化物联网设备平均的任务处理成功率和累积结果新鲜度。首先将该优化问题建模为一个混合整数非线性问题并提出了一种任务卸载选择的方法;然后将问题构建为一个马尔可夫决策过程;最后通过多智能体柔性动作-评判(Multi-Agent Soft Actor-Critic, MASAC)算法进行求解。仿真结果表明,与其他基线方案相比,提出的方案能够有效地提高物联网设备平均的任务处理成功率和累积结果新鲜度。

关键词: 无人机; 移动边缘计算; 多智能体强化学习; 卸载选择
中图分类号: V279; TN929.5; TP181 **文献标志码**: A **文章编号**: 2095-2163(2024)09-0170-09

Multi-agent reinforcement learning approach for task offloading optimization in UAV assisted mobile edge computing

LIU Xin, ZHAO Shasha
(School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In order to deal with the problem of limited coverage and insufficient computing power of a single unmanned aerial vehicle (UAV), a multi-UAV mobile edge computing (MEC) system combined with edge clouds (ECs) was proposed. To improve the task processing success rate and result freshness of Internet of Things (IoT) devices, an optimization scheme based on multi-agent reinforcement learning (MARL) was proposed, aiming to maximize the average task processing success rate and cumulative results freshness of IoT devices by jointly optimizing the offloading choice of IoT devices, UAVs' flight trajectories and UAVs' tasks offloading. Specifically, the optimization problem is firstly modeled as a mixed integer nonlinear problem, and a method is proposed to make offloading choice for IoT devices. Then, the problem is remodeled as a Markov decision process. Finally, a multi-agent soft actor-critic (MASAC) algorithm was adopted to solve the problem. Simulation results show that compared with other baseline schemes, the proposed scheme can effectively improve the average success rate and cumulative results freshness for IoT devices.

Key words: unmanned aerial vehicle; mobile edge computing; multi-agent reinforcement learning; offloading choice

0 引 言

随着物联网、大数据、通信等技术和产业的快速发展,各种类型应用产生的数据量爆炸式地增长。物联网(Internet of Things, IoT)设备受计算能力和电池容量的限制,在处理海量数据时面临着巨大的挑战。移动边缘计算(Mobile Edge Computing, MEC)把计算资源部署在网络边缘侧,可以为网络边缘的 IoT 设备提供计算服务以满足这些设备的计算需求。与传统的云计算相比,MEC 除了能够降低 IoT 设备任务处理的时间和能量消耗,还能大幅减少任务传输的时间。但是,由于在 MEC 系统中通常把计算服务器固定地放置在基站处,导致在一些信号衰落严重的场景、网络设备缺失的场景或者因为自然灾害导致网络设备毁坏的场景下,IoT 设备与地面蜂窝基站之间无法建立可靠的通信链路,因此

作者简介: 刘 鑫(1998-),男,硕士研究生, 主要研究方向: 边缘计算及强化学习。
通讯作者: 赵莎莎(1983-),女,博士, 讲师, 主要研究方向: 认知无线网络,无线网络虚拟化,动态频谱共享等。Email:zhaoss@njupt.edu.cn
收稿日期: 2023-05-09

基于地面网络设施的 MEC 系统无法为这些 IoT 设备提供可靠的计算服务来帮助其完成计算需求[1]。

近年来，由于无人机通信展现出许多的优势，比如可靠的通信能力、动态的部署能力以及灵活的扩展能力，无人机在无线通信领域受到了广泛关注[2-3]。将无人机通信与 MEC 技术结合，提高了 MEC 系统的弹性，因此基于无人机的 MEC 系统逐渐成为研究热点。如：Liu 等[4]提出了一个基于无人机的 MEC 网络，以解决传统 MEC 网络中由于多径、信号阻塞或阴影效应而导致的信道质量差的问题。Jeong 等[5]在基于无人机的 MEC 系统中，通过优化无人机轨迹以及资源分配以最小化 IoT 设备的能量消耗。Yu 等[6]提出一种多址边缘计算方案，采用多址技术降低无人机 MEC 系统中的网络传输时延，以缩短任务的完成时间。Ji 等[7]将无线能量传输技术应用于无人机 MEC 系统中，无人机可以在给 IoT 设备提供计算服务的同时为 IoT 设备补充能量，延长 IoT 设备的运行时间。

由于无人机的尺寸较小，装载的处理器通常不具有充足的计算资源，因此为了满足 IoT 设备的服务质量要求，除了对无人机的资源分配策略进行优化外，还可以通过联合无人机、边缘云的计算卸载方案或者多无人机计算卸载方案来提高 MEC 系统的性能。在文献[8-9]中，无人机不仅充当计算节点为 IoT 设备提供计算服务，还充当了中继节点，将接收到的任务进一步传输到边缘云，利用边缘云更强大的计算能力，辅助无人机进行任务处理。在文献[10]中表明，多无人机之间彼此协作地为 IoT 设备提供计算服务，不仅能够共同分担系统中的计算压力，还可以提供更大的服务覆盖范围。

虽然基于多无人机的 MEC 方案具有更高的性能，但同时也有许多挑战需要面对。在基于多无人机的 MEC 方案中，每个 IoT 设备可能会有多个候选无人机可以进行任务卸载，并且由于 IoT 设备和无人机之间的卸载关系是动态变化，通过传统方法获取最优卸载选择是非常复杂的。文献[11]中将多 IoT 设备与多无人机之间的卸载选择问题建模为一个策略博弈问题，并通过势博弈的方法获得最优的卸载选择。文献[12]为了平衡无人机的负载，提出了基于差分进化的多无人机部署机制，并将 IoT 设备的卸载选择问题建模为广义分配问题，最终获得近似最优的卸载选择。文献[13]提出了一种基于 K-Means 的用户分组算法，按照距离的远近来制定用户的卸载选择。除了卸载选择问题外，在基于多无人机的 MEC 方案中，无人机之间以协作的方式专注于自己负责的区域具有重要意义，所以无人机的轨迹优化问题是另一个挑战。文献[14-16]中采用传统的优化方法将轨迹优化问题分解为多个子问题，通过交替优化和连续凸优化的技术进行问题求解。但在多无人机场景中，环境和无人机的状态是动态变化的，采用传统优化方法进行求解是很困难的。得益于深度强化学习在连续动作控制领域取得的出色成绩，以及多智能体强化学习近年来取得的重大进展，文献[17-18]创新地采用多智能体强化学习的方法来优化无人机的轨迹，并得到了显著效果。

在此背景下，本文研究了一个联合多无人机和多边缘云的 MEC 系统。以往的研究中，通常以时延和能耗作为衡量服务质量的指标，而本文把 IoT 设备平均的任务处理成功率以及累积的结果新鲜度作为衡量 MEC 系统服务质量的指标。因为任务结果往往具有时效性，所以任务越是及时地被成功处理，IoT 设备获得的结果新鲜度就越高。本文的目的是通过联合优化无人机的轨迹、任务卸载以及 IoT 设备的卸载选择来最大化 IoT 设备平均的任务处理成功率以及累积的结果新鲜度。为了解决这个问题，本文首先提出了一个 IoT 设备卸载选择的方法，然后采用多智能体强化学习（Multi-Agent Reinforcement Learning，MARL）方法，对无人机的轨迹以及任务卸载进行优化。不同于以往研究中采用的基于多智能体深度确定性策略梯度（Multi-Agent Deep Deterministic Policy Gradient，MADDPG）[17]算法的求解方案，本文提出的基于多智能体柔性动作-评价（Multi-Agent Soft Actor Critic，MASAC）算法的方案具有更好的性能表现。仿真结果表明，本文所提的方案优于其他基线方案。

1　系统模型及问题描述

如图 1 所示，本文考虑了一个基于多无人机的 MEC 系统，其中包括 U 个无人机、N 个 IoT 设备以及 K 个边缘云，每个无人机配备有小型基站和嵌入式计算模块，每个边缘云配备有充足的计算资源。在该系统中，IoT 设备需要执行传感任务，并且其产生的数据需要快速地得到处理，但是由于计算能力和电池容量的限制，假设 IoT 设备不能在本地进行任务处理，而且和边缘云之间无法建立起可靠的无线通信链路，需要无人机为这些 IoT 设备提供计算服务。此外，无人机不仅可以作为计算节点为 IoT 设备提供计算服务，还可以作为中继节点将 IoT 设

备的任务进一步传输到边缘云上处理,以弥补无人机计算能力不足的短板。

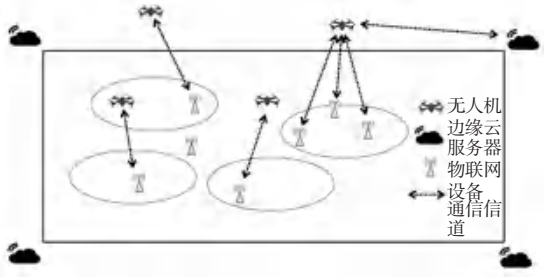

图 1 联合边缘云的多无人机移动边缘计算系统

Fig. 1 Multi-UAV mobile edge computing system combined with edge clouds

假设无人机需要服务的总时长为 T, 将服务时间均等地划分为 M 个时隙。每个时隙的持续时间为 $\Delta\tau = T/M$, 每个时隙由飞行时间 δ^{fly} 和悬停时间 δ^{hov} 两部分组成;无人机只能在悬停时间内接收 IoT 设备传输的任务并进行处理。本文假设在每个时隙所有 IoT 设备都会生成一个任务,将每个 IoT 设备在每个时隙生成的任务表示如下:

$$W_{n,t} = \{D_{n,t}, C_n\} \tag{1}$$

其中, $D_{n,t}$ 表示生成的任务数据量,单位是比特, C_n 表示处理每比特任务数据所需要的 CPU 周期数。

在每个时隙,IoT 设备需要将待处理的任务数据传输到无人机,将 IoT 设备的卸载选择记为 $l_{nu,t} = \{0,1\}$, $l_{nu,t}$ 等于 1 表示 IoT 设备 n 选择将任务传输到无人机 u ,否则 $l_{nu,t}$ 等于 0。本文规定 IoT 设备在每个时隙只能将任务完整地传输到一个无人机。

1.1 无人机移动模型

在每个时隙 t ,无人机以水平速度 $v_{u,t}$ 沿着水平方向 $\beta_{u,t}$ 飞行,并且无人机飞行的高度为 $h_{u,t}$, 因此无人机的水平坐标可以表示如下:

$$\begin{cases} x_{u,t} = x_{u,t-1} + v_{u,t}\delta^{\text{fly}}\cos(\beta_{u,t}) \\ y_{u,t} = y_{u,t-1} + v_{u,t}\delta^{\text{fly}}\sin(\beta_{u,t}) \end{cases} \tag{2}$$

其中, $x_{u,t}$ 为无人机的横坐标, $y_{u,t}$ 为无人机的纵坐标。

任意两个节点 n 和 u 之间的水平距离表示如下:

$$r_{nu,t} = \sqrt{(x_{n,t} - x_{u,t})^2 + (y_{n,t} - y_{u,t})^2} \tag{3}$$

其中, $x_{n,t}$ 为 IoT 设备横坐标; $y_{n,t}$ 为 IoT 设备纵坐标;IoT 设备的高 $h_{n,t}$ 为 0。

无人机和 IoT 设备之间仰角定义如下:

$$\varphi_{nu,t} = \arctan(h_{u,t}/r_{nu,t}) \tag{4}$$

将无人机最大仰角表示为 $\varphi_u^{\max}$, 因此无人机的覆盖半径可以表示为 $r_u^{\text{cov}} = h_{u,t}\tan(\varphi_u^{\max})$。IoT 设备只有在无人机的覆盖范围内,无人机才能够为其提供 MEC 服务。

在基于多无人机的 MEC 系统中,为了避免无人机之间发生碰撞,规定所有无人机之间保持最小碰撞距离 $d_{\min}$。

1.2 信道模型

假设任意两个节点之间的无线信道都经历小尺度衰落和大尺度路径损耗,地对空路径损耗模型兼具有视距链路 (Line of Sight, LoS) 和非视距链路 (Non Line of Sight, NLoS) 的特点,可以表示如下:

$$\begin{cases} \text{PL}_{\text{LoS}} = 20\log\left(\dfrac{4\pi f_c d_{nu,t}}{c}\right) + \eta_{\text{LoS}} \\ \text{PL}_{\text{NLoS}} = 20\log\left(\dfrac{4\pi f_c d_{nu,t}}{c}\right) + \eta_{\text{NLoS}} \end{cases} \tag{5}$$

其中, f_c 表示载波频率, c 表示光速。

无人机与 IoT 设备之间的距离表示为 $d_{nu,t} = \sqrt{(r_{nu,t})^2 + (h_{u,t})^2}$, 式中 η_{LoS} 和 η_{NLoS} 分别是 LoS 和 NLoS 所对应的损耗。根据文献[18],视距链路的概率可以表示如下:

$$P_{nu,t}^{LoS}(\varphi_{nu,t}) = \frac{1}{1 + a\exp(-b(\varphi_{nu,t} - a))} \tag{6}$$

其中, a 和 b 都是和环境有关的常量。因此平均路径损耗可以表示如下:

$$L(\varphi_{nu,t}, d_{nu,t}) = 20\log\left(\frac{4\pi f_c d_{nu,t}}{c}\right) + P_{nu,t}^{LoS}(\varphi_{nu,t})\eta_{\text{LoS}} + (1 - P_{nu,t}^{LoS}(\varphi_{nu,t}))\eta_{\text{NLoS}} \tag{7}$$

由于在接收端可能会经历 LoS 和多径散射,因此本文选择用莱斯分布模拟无人机与地面 IoT 设备之间的小尺度衰落。将无人机与 IoT 设备之间的信道增益记为 $g_{nu,t}$,根据文献[19]将 $g_{nu,t}$ 的概率函数分布表示如下:

$$f_{g_{nu,t}}(\omega) = \frac{(K_{nu,t} + 1)e^{-K_{nu,t}}}{\Omega}\exp\left(\frac{-(K_{nu,t} + 1)\omega}{\Omega}\right) \times I_0\left(2\sqrt{\frac{K_{nu,t}(K_{nu,t} + 1)\omega}{\Omega}}\right), \ \omega \geqslant 0 \tag{8}$$

其中, $I_0(\cdot)$ 是第一类零阶修正贝塞尔函数, $K_{nu,t}$ 是莱斯因子,定义为 LoS 分量中的功率与非 LoS 多径散射中的功率之比。Ω 是平均衰落功率, $\Omega = 1$。研究表明,无人机相对于地面节点的仰角在决定莱斯因子中起主导作用。通过引入非递减函数,将莱斯因子建模为仰角的函数表示如下:

$$K(\varphi_{nu,t}) = \kappa_0 \cdot \exp[\frac{2}{\pi}\ln\left(\frac{\kappa_{\pi/2}}{\kappa_0}\right)\varphi_{nu,t}] \quad (9)$$

其中，$\kappa_0 = K(0)$，$\kappa_{\pi/2} = K(\pi/2)$。

假设 IoT 设备的传输功率为 P_T，IoT 设备可以通过正交频分多址技术与无人机通信，因此无人机与 IoT 设备之间的信噪比可以表示如下：

$$\gamma_{nu,t}^{SNR} = \frac{10^{\log_{10}P_T - L(\varphi_{nu,t},d_{nu,t})/10}}{\sigma^2} g_{nu,t} \quad (10)$$

其中，σ^2 为噪声功率。最终根据文献[20]，无人机与 IoT 设备之间的平均数据传输速率可以表示如下：

$$R_{nu,t} \approx \frac{B_{un,t}}{\ln 2}[\ln(1+\bar{\gamma}_{nu,t}) - \frac{\bar{\gamma}_{nu,t}^2(2e^{-K(\varphi_{nu,t})}(3K(\varphi_{nu,t})^2 + 3K(\varphi_{nu,t}) + 1) - (1+K(\varphi_{nu,t}))^2)}{2(1+\bar{\gamma}_{\varphi_{nu,t}})^2(1+K(\varphi_{nu,t}))^2} \quad (11)$$

其中，$\bar{\gamma}_{nu,t} = \frac{10^{\log_{10}P_T - L(\varphi_{nu,t},d_{nu,t})/10}}{\sigma^2}$，$B_{un,t}$ 为无人机分配给 IoT 设备的带宽资源[10]，本文采用均匀分配带宽资源的策略。

1.3　计算模型

在时隙 t，如果 IoT 设备 n 将第 m 个任务传输到无人机 u，则 IoT 设备的传输时间可以表示如下：

$$T_{nu,t}^{tr,m} = \frac{D_{n,m}}{R_{nu,t}} \quad (12)$$

无人机 u 接收到 IoT 设备 n 传输过来的任务数据后，为该任务处理分配相应的计算资源。因此，无人机为处理任务花费的计算时间可以表示如下：

$$T_{nu,t}^{com,m} = \frac{(1-\chi_{u,t})D_{n,m}C_n}{f_{nu,t}} \quad (13)$$

其中，$f_{nu,t}$ 为无人机 u 为处理 IoT 设备 n 的任务数据分配的计算资源。

为了辅助无人机进行任务处理，本文在多无人机 MEC 系统中加入了 K 个边缘云，无人机可以进一步将接收到的任务传输到边缘云上进行处理。无人机采用部分卸载模式，将任务传输比例记为 $\chi_{u,t}$，因此 IoT 设备 n 的部分任务数据通过无人机 u 传输到边缘云 k 的时间可以表示如下：

$$T_{nuk}^{tr,m} = \frac{D_{n,m}\chi_{u,t}}{R_{uk,t}} \quad (14)$$

边缘云 k 在接收到无人机传输的任务数据后进行处理，处理的时间可以表示如下：

$$T_{nuk,t}^{com,m} = \frac{\chi_{u,t}D_{n,m}C_n}{f_{nk,t}} \quad (15)$$

其中，$f_{nk,t}$ 表示边缘云 k 分配给 IoT 设备 n 用来处理任务数据的计算资源。由于任务处理结果的数据量远远小于任务本身的数据量，所以参考文献[21-23]，本文忽略计算结果的传输时间。因此 IoT 设备 n 的第 m 个任务的处理时间可以分为两部分，其中包括 IoT 设备的传输时间和后续的任务处理时间，因此 IoT 设备 n 的第 m 个任务的数据处理时间可以表示如下：

$$T_{nu,t}^{m} = T_{nu,t}^{tr,m} + \max(T_{nu,t}^{com,m}, T_{nuc,t}^{tr,m} + T_{nue,t}^{com,m}) \quad (16)$$

由于无人机每个时隙中悬停时间为 δ^{hov}，如果任务在悬停时间内 δ^{hov} 处理完成了，表示任务处理成功，否则表示任务处理失败。假设 IoT 设备的每个任务之间都具有关联性，必须要等到前面的任务都成功处理了才可以处理后面的任务。处理失败的任务会存储在 IoT 设备的本地任务队列 L_n 中，等待下一个时隙重新请求处理。在时隙 $[t, t+1)$，每个 IoT 设备会尝试将本地剩余的任务尽可能多的传输到无人机进行处理。假设在第 t 时隙内，IoT 设备 n 中有 $\zeta_{n,t}$ 个任务被成功处理，则在 t 时隙结束时，IoT 设备 n 的任务队列 L_n 中的剩余任务数可以表示如下：

$$\ell_{n,t} = \begin{cases} \ell_{n,t-1} + 1, & \text{if } \zeta_{n,t} = 0 \\ \ell_{n,t-1} + 1 - \zeta_{n,t}, & \text{if } \zeta_{n,t} > 0 \end{cases} \quad (17)$$

将每个 IoT 设备任务处理的成功率定义为当前已经完成的任务数与其产生的总任务数之比，因此成功率可以表示如下：

$$\mathcal{P}_{n,t} = (t - \ell_{n,t})/t \quad (18)$$

本文结合信息年龄（Age of Information，AoI）[21]，将 IoT 设备每个任务 $W_{n,t}$ 结果的新鲜度表示如下：

$$TL_{n,t} = \begin{cases} TL_{n,t}, & \text{在当前时隙完成} \\ TL_{n,t} + 1, & \text{其它} \end{cases} \quad (19)$$

当任务 $W_{n,t}$ 产生时，将 $TL_{n,t}$ 的初始值设置为 1。如果 $W_{n,t}$ 没能在当前时隙被成功处理，则 $TL_{n,t}$ 加 1，$TL_{n,t}$ 越小，任务结果的新鲜度越高，代表任务处理越及时。

1.4　问题描述

本文的目的是通过联合优化无人机的轨迹、无人机的任务卸载以及 IoT 设备的卸载选择来最大化 IoT 设备平均的任务处理成功率和累积结果新鲜度。因此，本文的优化问题可以定义如下：

$$\max_{x_{u,t},y_{u,t},\beta_{u,t},v_{u,t},\chi_{u,t},l_{nu,t}} -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{M}(2-\mathcal{P}_{n,t})\cdot TL_{n,t} \quad (20)$$

$$\text{s.t.}\quad C1:\sum_{u=1}^{U} l_{nu,t} \leqslant 1,$$
$$C2:0 \leqslant x_{u,t} \leqslant X_{\max},$$
$$C3:0 \leqslant y_{u,t} \leqslant Y_{\max},$$
$$C4:0 \leqslant v_{u,t} \leqslant v^{\max},$$
$$C5:0 \leqslant \beta_{u,t} < 2\pi,$$
$$C6:0 \leqslant \chi_{u,t} \leqslant 1,$$
$$C7:r_{nu,t} \leqslant r_u^{\text{cov}},$$
$$C8:\sqrt{(r_{uu',t})^2 + (h_{u,t} - h_{u',t})^2} \geqslant d_{\min},\ u' \neq u$$

其中，$C1$ 表示 IoT 设备每个时刻产生的数据只能传输给其中一个无人机处理，$C2$、$C3$ 表示无人机需要在指定的区域内飞行，$C4$、$C5$ 表示每时刻无人机的飞行速度和飞行方向，$C6$ 表示无人机的任务卸载比例，$C7$ 表示无人机的覆盖范围，$C8$ 表示无人机之间避免碰撞的最小间距。

在以上优化问题中，因为既包含离散的变量 $l_{nu,t}$，又包含连续变量 $\{\beta_{u,t}, v_{u,t}, \chi_{u,t}\}$，所以使用传统的方法求解该问题复杂度较高。因此本文先针对变量 $l_{nu,t}$ 提出一种 IoT 设备的卸载选择方法，然后提出基于多智能体强化学习算法的方案对无人机的轨迹以及任务卸载进行优化。

2 算法设计

2.1 卸载决策

根据所有无人机和 IoT 设备的坐标，用 $I_{u,t}^{lb}$ 表示只能与无人机 u 通信的 IoT 设备数量。由于本文假设无人机的资源均等地分配给每个被服务的 IoT 设备，所以在第 t 时隙，当 IoT 设备 n 选择无人机 u 进行任务卸载时，可以根据公式(16)将 IoT 设备 n 的第 m 个任务花费的处理时间下限表示为和无人机 u 服务的 IoT 设备数量 $I_{u,t}^{lb}$ 以及无人机 u 与 IoT 设备 n 之间的距离 $d_{nu,t}$ 有关的表达式 $T_{nu,t}^{lb,m}(I_{u,t}^{lb}, d_{nu,t})$。类似地，用 $I_{u,t}^{ub}$ 表示处于无人机 u 覆盖范围内的所有 IoT 设备数量，因此任务处理花费的时间上限可以表示为 $T_{nu,t}^{ub,m}(I_{u,t}^{ub}, d_{nu,t})$。因此 IoT 设备 n 选择将任务卸载到无人机 u 处理所要花费的时间可以表示如下：

$$\tilde{T}_{nu,t}^{m}(I_{u,t}, d_{nu,t}) = \lambda_{u,t} T_{nu,t}^{ub}(I_{u,t}^{ub}, d_{nu,t}) + (1 - \lambda_{u,t}) T_{nu,t}^{lb}(I_{nu,t}^{lb}, d_{nu,t}) \quad (21)$$

其中，每个无人机的卸载选择因子 $\lambda_{u,t} \in [0, 1]$。每个 IoT 设备可以根据式(21)的任务处理时间，选择最优的无人机进行任务卸载，所以卸载选择策略可以表示为：

$$l_{nu,t} = \begin{cases} 1, & u = \min\limits_{u \in U} \{\tilde{T}_{nu,t}^{m}\} \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

式中：任务处理的时间与因子 $\lambda_{u,t}$ 有关，$\lambda_{u,t}$ 的取值影响着估计处理时间 $\tilde{T}_{nu,t}^{m}$ 与真实处理时间 $T_{nu,t}^{m}$ 之间的差别，进而影响 IoT 设备卸载选择策略。下面将介绍使用多智能体强化学习算法对 $\lambda_{u,t}$ 进行求解，通过智能体的策略网络对 $\lambda_{u,t}$ 进行训练和学习，获得最优的 $\lambda_{u,t}^{*}$。

2.2 马尔可夫决策过程构建

在强化学习中，通常把智能体与环境交互的过程建模为马尔可夫决策过程。马尔可夫决策过程可以用一个元组 (S, A, R, P) 来定义，其中 S 为环境产生的状态空间；A 是智能体的动作空间；R 是奖励函数；P 是转移概率，因为本文采用无模型的强化学习算法，因此转移概率 P 不做讨论。关于马尔可夫决策过程的各个元素的详细定义如下。

1）状态空间 S_u

IoT 设备在每个时隙都会把其产生的计算任务按照卸载选择的结果传输到一个无人机上，且无人机缓存 IoT 设备的位置和任务信息。因此无人机 u 的状态空间可以定义如下：

$$s_{u,t} = \{q_{u,t}, q_{1,t}, \cdots, q_{N,t}, D_{1,t}, \cdots, D_{N,t}\}, s_{u,t} \in S_u \quad (23)$$

其中，$q_{u,t}$ 表示无人机的位置；$q_{n,t}$ 表示缓存的 IoT 设备的位置信息；$D_{n,t}$ 表示 IoT 设备卸载的任务。

2）动作空间 A_u

为了最大化 IoT 设备平均的任务处理成功率以及累积结果新鲜度，每个无人机需要优化其飞行轨迹、任务卸载和卸载选择因子，从而降低任务的传输延时以及处理延时。因此无人机的动作空间可以定义如下：

$$a_{u,t} = \{\beta_{u,t}, v_{u,t}, \chi_{u,t}, \lambda_{u,t}\}, a_{u,t} \in A_u \quad (24)$$

其中，$\beta_{u,t}$ 表示无人机飞行的角度；$v_{u,t}$ 表示无人机飞行的速度；$\chi_{u,t}$ 表示无人机卸载任务的比例；$\lambda_{u,t}$ 为无人机 u 卸载选择因子。

3）奖励函数 R

无人机 u 在环境状态 $s_{u,t}$ 执行动作 $a_{u,t}$ 后会获得环境奖励，帮助无人机改进策略，强化学习的目标是最大化累积奖励。因此为了与优化问题的目标保持一致，奖励函数可以定义如下：

$$r_{u,t} = -\frac{1}{N}\sum_{n \in N}(2 - \mathcal{P}_{n,t}) \cdot \ell_{n,t} + \sum_{i=1,2}\eta_i \quad (25)$$

其中，$\ell_{n,t}$ 为第 t 时隙 IoT 设备 n 的任务队列中剩余未处理完成的任务数，IoT 设备每个时隙的 $\ell_{n,t}$ 累加起来就等效于其所有任务结果的新鲜度累加；

$P_{n,t}$ 为任务处理的成功率；η_1 为无人机发生碰撞时的惩罚；η_2 为每个时隙成功服务一定数量的用户获得的额外奖励。本文中多个无人机共享同一奖励以获得最佳的协作策略。

2.3　基于 MASAC 算法的解决方案

本文提出的 MASAC 算法是利用集中式训练分布式执行框架，对 SAC[22] 算法在多智能体领域进行的扩展。利用集中式训练分布式执行框架，可以解决多个智能体之间由于策略变化和观测信息不完全导致的学习不稳定问题。在集中式训练阶段，每个智能体价值网络的输入不仅包含本地的状态信息 $s_{u,t}$ 和动作信息 $a_{u,t}$，还包含其它智能体的状态信息 $s_{-u,t}$ 和动作信息 $a_{-u,t}$，而策略网络只需要输入智能体本地的状态信息。在执行阶段，每个智能体只需要使用策略网络各自执行。本文将全局的状态信息定义为 $s_t=\{s_{u,t},s_{-u,t}\}$，全局的动作信息定义为 $a_t=\{a_{u,t},a_{-u,t}\}$。

SAC 算法是一种基于最大熵框架的强化学习算法，通过引入策略熵，提高了算法的探索性和稳定性。SAC 的目标是最大化累积奖励和策略熵，目标函数表示为

$$\pi^*=\arg\max_{\pi}\sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t)+\alpha H(\pi(\cdot\mid s_t))] \tag{26}$$

其中，$H(\pi(\cdot\mid s_{u,t}))=-\log(\pi(\cdot\mid s_{u,t}))$ 表示策略熵，$\alpha_u\in(0,1)$ 表示温度系数。α_u 决定了策略熵项对奖励的相对重要性，从而控制最优策略的随机性。

由于在集中式训练分布式执行框架中，价值网络的输入需要全体智能体的状态信息以及动作信息，因此集中式的 Q 值可以定义如下：

$$Q(s_t,a_t)=r(s_{u,t},a_{u,t})+\gamma\mathbb{E}_{s_{t+1}\sim\rho_\pi}[V(s_{t+1})] \tag{27}$$

其中 $\gamma\in(0,1)$ 是折扣因子，$V(s_t)$ 定义如下：

$$V(s_t)=E_{a_t\sim\pi}[Q(s_t,a_t)-\alpha_u\log\pi(a_{u,t}\mid s_{u,t})] \tag{28}$$

在 MASAC 算法中，每个智能体使用两个价值网络 $Q_{\phi_{u,1}}$ 和 $Q_{\phi_{u,2}}$，可以减少 Q 值过高估计导致的偏差，这两个价值网络具有相同的网络结构，不共享参数，价值网络的损失函数 $J_Q(\phi_{u,i})$ 如下：

$$J_Q(\phi_{u,i})=\mathbb{E}_{(s_t,a_t,r_{u,t},s_{t+1})\sim B_u}[\frac{1}{2}(Q_{\phi_{u,i}}(s_t,a_{u,t},a_{-u,t})-(r_{u,t}+\gamma(\min_{j=1,2}Q_{\bar{\phi}_{u,j}}(s_{t+1},a_{u,t+1},a_{-u,t+1})-\alpha_u\log\pi_{\theta_u}(a_{u,t+1}\mid s_{u,t+1}))))^2] \tag{29}$$

其中，B_u 表示重放缓冲区，$a_{u,t+1}=\pi_{\theta_u}(s_{u,t+1})$，$\phi_{u,i}$ 是价值网络的参数，其可以通过最小化损失函数 $J_Q(\phi_{u,i})$ 来更新，$\bar{\phi}_{u,i}$ 是目标价值网络的参数。

策略网络的参数 θ_u 可以通过最小化损失函数 $J_\pi(\theta_u)$ 来更新。

$$J_\pi(\theta_u)=\mathbb{E}_{s_t\sim B_u}[\mathbb{E}_{a_t\sim\pi_{\theta_u}}[\alpha_u\log\pi_{\theta_u}(a_{u,t}\mid s_{u,t})-\min_{j=1,2}Q_{\phi_{u,j}}(s_t,a_{u,t},a_{-u,t})]] \tag{30}$$

在随机策略中，一般用高斯分布来表示策略的分布，即通过参数 θ_u 将状态映射为高斯分布的均值和方差，并通过高斯分布中采样得到动作，所以在最小化 $J_\pi(\theta_u)$ 时不能直接对 $J_\pi(\theta_u)$ 关于 θ_u 求梯度。因此，为了方便求梯度，可以使用重参数化技巧，使其产生一个较低的方差估计。重参数化可以表示为 $a_{u,t}=f_{\theta_u}(\epsilon_{u,t};s_{u,t})$，$\epsilon_{u,t}$ 是一个输入噪声向量，可以从一些固定分布抽样得到（如高斯分布）。因此可以将公式(30)重写：

$$J_\pi(\theta_u)=\mathbb{E}_{s_t\sim D_u,\epsilon_{u,t}\sim N}[\alpha_u\log\pi_{\theta_u}(f_{\theta_u}(\epsilon_{u,t};s_{u,t})\mid s_{u,t})-\min_{j=1,2}Q_{\phi_{u,j}}(s_t,a_{u,t},a_{-u,t})\mid a_{u,t}=f_{\theta_u}(\epsilon_{u,t};s_{u,t})] \tag{31}$$

在 SAC 算法中，温度系数 α_u 控制了最优策略的随机性，当智能体处于未知环境时，增加 α_u 有助于智能体充分地探索未知环境。当智能体已经熟悉环境时，应该减少 α_u 以便智能体能够选择最佳策略。因此，为了保持探索和利用之间的平衡，文献[23]提出了一种自动调整温度系数 α_u 的方法，可以通过最小化损失函数 $J(\alpha_u)$ 来更新 α_u。

$$J(\alpha_u)=\mathbb{E}_{a_{u,t}\sim\pi_{\theta_{u,t}}}[-\alpha_u\log\pi_{\theta_{u,t}}(a_{u,t}\mid s_{u,t})-\alpha_u\bar{H}] \tag{32}$$

其中，$\bar{H}$ 表示目标策略熵。

基于 MASAC 算法的无人机轨迹控制、无人机任务卸载以及 IoT 设备卸载选择如算法 1 所示。

算法 1　初始化每个无人机的策略网络参数 θ_u、价值网络参数 $\phi_{u,1}$ 和 $\phi_{u,2}$，并将价值网络的参数拷贝到对应的目标网络 $\bar{\phi}_{u,1}\leftarrow\phi_{u,1}$，$\bar{\phi}_{u,2}\leftarrow\phi_{u,2}$，初始化经验缓冲区 B_u。

1for episode e=1：e^{max} do

2　初始化每个无人机的本地状态信息 $s_{u,1}$；

3　for time slot $t=1$:M do

4　　获取无人机全局状态信息 s_t；

5　　每个无人机获取动作 $a_{u,t}\sim\pi_{\theta_u}(\cdot\mid s_{u,t})$；

6　每个无人机沿着 $\beta_{u,t}$ 方向以速度 $v_{u,t}$ 飞行 δ^{fly} 时间，如果无人机执行飞行动作后将飞出给定区域，则当前时隙无人机保持在原地；

7　每个 IoT 设备根据每个无人机的卸载选择因子 $\lambda_{u,t}$ 制定最优的卸载选择策略并卸载任务；

8 每个无人机将接收到的任务按照任务卸载比例$\chi_{u,t}$卸载到一个最近的边缘云并处理剩余任务。最终将结果响应给 IoT 设备,并获得环境奖励$r_{u,t}$,以及下一状态信息$s_{u,t+1}$;

9 获取无人机全局动作信息a_t;

10 获取无人机全局状态信息s_{t+1};

11 for UAV $u = 1:U$ do

12 将元组$(s_t, a_t, r_{u,t}, s_{t+1})$存储到$B_u$;

13 从B_u中随机抽取小批量元组$\{(s_j, a_j, r_{u,j}, s_{j+1})\}$作为训练数据;

14 根据公式(29)更新价值网络

$\phi_{u,i} \leftarrow \phi_{u,i} - \lambda_{Q_{u,i}} \nabla_{\phi_{u,i}} J_Q(\phi_{u,i})$ for $i \in \{1,2\}$

根据公式(31)更新策略网络

$\theta_u \leftarrow \theta_u - \lambda_{\pi_u} \nabla_{\theta_u} J_\pi(\theta_u)$;

根据公式(32)更新温度系数

$\alpha_u \leftarrow \alpha_u - \lambda_{\alpha_u} \nabla_{\alpha_u} J(\alpha_u)$;

按软更新方式更新目标价值网络

$\bar{\phi}_{u,i} \leftarrow \tau\phi_{u,i} + (1-\tau)\bar{\phi}_{u,i}$ for $i \in \{1,2\}$;

15 end for

16 end for

17 end for

3 仿真结果分析

本节将从多个方面来评估所提方案的性能,通过仿真结果验证该方案的有效性。

3.1 仿真参数设置

本文假设有 50 个 IoT 设备随机分布在一个 200 m×200 m 的给定区域内,一个基于多无人机的 MEC 系统需要为给定区域内的 IoT 设备提供 MEC 服务。MEC 系统中包含 4 个边缘云和 4 个无人机。边缘云的坐标为[0,0]、[0,200]、[200,0]、[200,200],无人机的初始水平坐标分别为[50,50]、[50,150]、[150,50]、[150,150]。总时隙数 M 为 40,每个时隙的飞行时间δ^{fly}为 2.5 s,悬停时间δ^{hov}为 10 s。每个无人机的最大飞行速度为 20 m/s,无人机的飞行高度固定,无人机最大的仰角φ_{max}为 42.44°。每个无人机需要保持的最小碰撞距离d_{min}为 10 m,每个无人机的 CPU 频率为 1 GHz,每个边缘云的 CPU 频率是 5 GHz。每个设备在每个时隙都会产生一个计算任务$W_{n,t}$,每个计算任务$W_{n,t}$的数据量固定,处理每比特计算任务需要的 CPU 周期数C_n为 1 000 cycles/bit。每个无人机的带宽B_u为 1 MHz,边缘云给无人机分配的带宽$B_{k,u}$为 0.5 MHz。无人机的传输功率为 5 W,IoT 的传输功率为 0.1 W。η_1和η_2分别为-5 和 10,其他参数见表 1。

表 1 系统仿真参数设置

Table 1 System simulation parameters setting

参数	值
环境参数 a	12.08
环境参数 b	0.11
噪声功率 σ^2 /dBm	-100
传输损耗 η_{LoS} /dB	1.6
传输损耗 η_{NLoS} /dB	23
莱斯因子 κ_0 /dB	5
莱斯因子 $\kappa_{\pi/2}$ /dB	15

此外,在本文提出的算法中,所有策略网络和价值网络都有 3 个完全连接的隐藏层,分别由 400、300 和 300 个神经元组成,训练 1 000 回合,使用 python 3.6 和 TensorFlow 1.14.0 进行仿真。其它与算法有关的超参数见表 2。

表 2 算法超参数设置

Table 2 Algorithm hyperparameters setting

参数	值
激活函数	ReLU
更新间隔	1
更新率 τ	0.01
批量数据大小	256
重放缓冲区大小	100 000
折扣因子 γ	0.95
优化器	Adam
学习率	0.000 5
目标熵	-dim(A)

3.2 仿真性能分析

为了评估本文算法的性能,将其方案与以下两种基线方案进行比较:

1)基线方案 1:基于 MADDPG 算法的 IoT 设备卸载选择以及无人机轨迹控制、任务卸载的方案。

2)基线方案 2:方案采用 MASAC 算法进行无人机轨迹控制以及任务卸载,并根据距离决定 IoT 设备的卸载选择。

图 2 展示了无人机飞行高度为 80 m 并且每个任务数据量为 3 Mb 时,累积奖励随着训练回合增加而变化的曲线。可以看出,本文所提方案最终可以收敛到更高的累积奖励值,比其他两个基线方案更有优势。本文所提方案在训练的前 300 回合中,由于需要学习卸载选择因子$\lambda_{u,t}$帮助 IoT 设备制定卸载选择,因此收敛速度在开始的时候会慢于基线方案 2,但最终会收敛到更高的累计奖励值。本文所提的方案和基线方案 2 最终的累积奖励值都大于基线方案 1,是因为 SAC 在标准的强化学习目标中引入了策略熵,SAC 的目标是最大化累积奖励并且最大化策略熵,所以 SAC 具有更好的探索性和稳定性,能够充分地探

索环境寻找最优策略。而 MADDPG 是确定性策略梯度方法，其动作的探索性不如 MASAC，因此基线方案 1 的最终表现不如基于 MASAC 算法的方案，所以最终累积奖励收敛于一个较低的值。本文所提方案优于基线方案 2，因为在基线方案 2 中每个 IoT 设备根据与无人机之间的距离选择最近的无人机进行任务卸载，这就意味着 IoT 设备实际上并不一定能真正地获取最优的卸载选择策略，并且由于 IoT 设备的位置是随机的，可能导致大量的 IoT 设备都选择将任务卸载到同一个无人机上，导致该无人机过载，造成任务处理失败。所以基线方案 2 也难以获得最优策略。

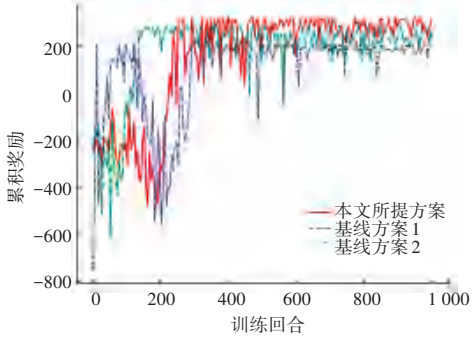

图 2　累积奖励与训练回合的关系

Fig. 2　Relationship between cumulative reward and the number of training episodes

图 3、图 4 分别展示了每个任务为 3 Mb 时，无人机在不同飞行高度下 3 种方案的平均任务处理成功率和累积结果新鲜度。可以看到，随着飞行高度的增加，IoT 设备平均任务处理成功率和累积结果新鲜度越大，这是因为无人机的飞行高度越高，其与地面 IoT 设备之间的通信链路为 LoS 链路的概率越大，同时无人机的覆盖范围也会增加。其中，本文所提方案有着更高的平均任务处理成功率和累积结果新鲜度，是因为无人机的高度增加导致覆盖范围变大，所以 IoT 设备可以选择进行任务卸载的无人机可能会不止一个，按照距离制定的卸载选择不一定是最优的策略，而本文所提的卸载选择方法中，$\lambda_{u,t}$ 经过策略网络不断地训练和学习之后，会变得越来越恰当，因此 IoT 设备可以基于 $\lambda_{u,t}$ 做出更好的卸载选择策略。本文所提方案在不同飞行高度的成功率平均相比基线方案 1 和基线方案 2 分别高 10.2%和 4.4%。

图 5 和图 6 分别展示了无人机飞行高度为 80 m 时，不同任务数据量下的平均任务处理成功率和累积结果新鲜度。由于本文假设无人机和边缘云的计算资源均等地分配给每个 IoT 设备，所以在同时处理大量设备传输的任务时，会出现随着处理的任务数据量增加失败率变高的问题。由于无人机和边缘云计算能力的限制，当任务量达到 7 Mb 时，3 种方案的性能基本接近。本文所提方案在不同任务数据量的成功率平均比基线方案 1 和基线方案 2 方案分别高 7.1%和 2.9%。

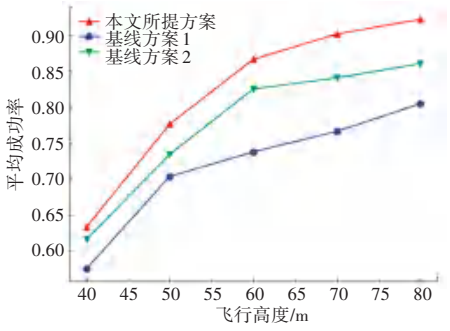

图 3　飞行高度与平均成功率关系

Fig. 3　Relationship between flight altitude and success rate

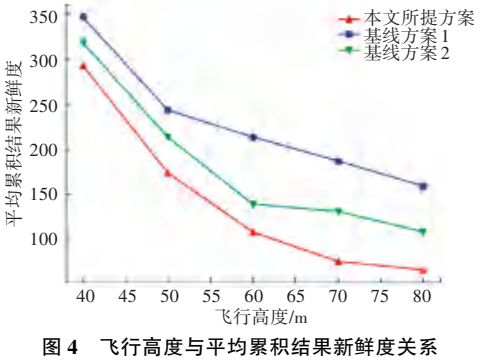

图 4　飞行高度与平均累积结果新鲜度关系

Fig. 4　Relationship between flight altitude and average cumulative result freshness

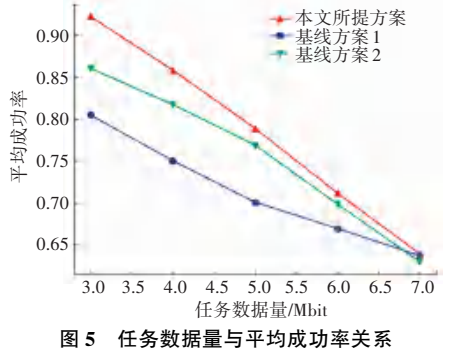

图 5　任务数据量与平均成功率关系

Fig. 5　Relationship between data size and success rate

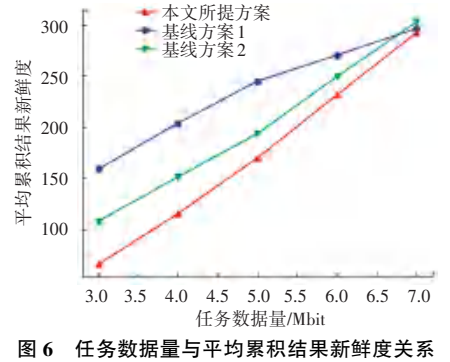

图 6　任务数据量与平均累积结果新鲜度关系

Fig. 6　Relationship between data size and average cumulative result freshness

综上所述,本文提出的方案相比于其它两种比较方案有着更高的性能优势。但是,由于 IoT 设备需要根据 $\lambda_{u,t}$ 制定卸载选择,因此策略网络在训练中需要不断地训练和学习 $\lambda_{u,t}$,造成算法在训练的前期阶段收敛速度不如基于距离选择的基线方案 2,但是最终获得的卸载策略优于基线方案 2;基线方案 1 探索能力不如其它两种方案,最终性能不如基于 MASAC 的方案。

4 结束语

本文研究了一个联合多无人机、多边缘云的移动边缘计算系统,为了最大化 IoT 设备平均的任务处理成功率和累积结果新鲜度,考虑了一个联合多 IoT 设备卸载选择和多无人机轨迹控制、任务卸载的优化问题,并提出了一种基于 MASAC 算法的解决方案。实验结果表明本文提出方案优于其它两种基线方案,可以提高 IoT 设备平均的任务处理成功率和累积结果新鲜度。

参考文献

[1] HU Q, CAI Y, YU G, et al. Joint offloading andtrajectory design for UAV-enabled mobile edge computing systems[J]. IEEE Internet of Things Journal,2019, 6(2): 1879-1892.

[2] SHI W, ZHOU H, LI J, et al. Drone assisted vehicular networks: architecture, challenges and opportunities [J]. IEEE Network, 2018, 32(3): 130-137.

[3] LI B, FEI Z, ZHANG Y. UAV communications for 5G and beyond: Recent advances and future trends[J]. IEEE Internet of Things Journal, 2019, 6(2): 2241 -2263.

[4] LIU B, WAN Y, ZHOU F, et al. Resource allocation and trajectory design for MISO UAV-assisted MEC networks[J]. IEEE Transactions on Vehicular Technology, 2022, 71(5): 4933-4948.

[5] JEONG S, SIMEONE O, KANG J. Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning[J]. IEEE Transactions on Vehicular Technology, 2018, 67(3):2049-2063.

[6] YU Y, BU X, YANG K, et al. UAV-aided low latency multi-access edge computing [J]. IEEE Transactions on Vehicular Technology, 2021,70(5): 4955-4967.

[7] JI L, GUO S. Energy-efficient cooperative resource allocation in wireless powered mobile edge computing[J]. IEEE Internet of Things Journal, 2019, 6(3): 4744-4754.

[8] YU Z, GONG Y, GONG S, et al. Joint taskoffloading and resource allocation in UAV-enabled mobile edge computing[J]. IEEE Internet of Things Journal, 2020, 7(4): 3147-3159.

[9] ZHAO N, YE Z, PEI Y, et al. Multi-agent deep reinforcement learning for task offloading in UAV - assisted mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2022, 17(9): 6949-6960.

[10] WANG L, WANG K, PAN C, 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.

[11] GAO A, WANG Q, LIANG W, et al. Game combined multi-agent reinforcement learning approach for UAV assisted offloading [J]. IEEE Transactions on Vehicular Technology, 2021, 70 (12): 12888-12901.

[12] YANG L, YAO H, WANG J, et al. Multi-UAV-enabled load-balance mobile - edge computing for IoT networks [J]. IEEE Internet of Things Journal, 2020, 7(8): 6898-6908.

[13] ZHONG R, LIU X, LIU Y, et al. Multi - agent reinforcement learning in NOMA - aided UAV networks for cellular offloading [J]. IEEE Transactions on Wireless Communications, 2022, 21 (3): 1498-1512.

[14] ZHAN C, HU H, LIU Z, et al. Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications[J]. IEEE Internet of Things Journal, 2021, 8(20): 15553-15567.

[15] QIN X, SONG Z, HAO Y, et al. Joint resource allocation and trajectory optimization for multi - UAV - assisted multi - access mobile edge computing [J]. IEEE Wireless Communications Letters, 2021, 10(7): 1400-1404.

[16] LUO Y, DING W, ZHANG B, et al. 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.

[17] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach:IEEE, 2017:6382-6393.

[18] Al - HOURANI A, KANDEEPAN S, LARDNER S. Optimal LAP altitude for maximum coverage [J]. IEEE Wireless Communications Letters, 2014, 3(6): 539-572.

[19] AZARI M, MAHDI R F, CHEN K C, et al. Ultra reliable UAV communication using altitude and cooperation diversity[J]. IEEE Transactions on Communications, 2018, 66(1): 330-344.

[20] QI H, HU Z, HUANG H, et al. Energy efficient 3-D UAV control for persistent communication service and fairness: A deep reinforcement learning approach [J]. IEEE Access, 2020, 8: 53172-53184.

[21] ZHANG X, WANG J, POOR H V. AoI-driven statistical delay and error-rate bounded QoS provisioning for URLLC in the finite block- length regime [C]//Proceedings of 2021 55th Annual Conference on Information Sciences and Systems (CISS). USA: IEEE, 2021: 1-6.

[22] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proceedings of International Conference on Machine Learning. PMLR, 2018: 1861-1870.

[23] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications[J]. arXiv preprint arXiv:1812.05905, 2018.