

# 基于数据压缩的无人机边缘计算卸载优化

李 斌<sup>1</sup>, 朱 潇<sup>1</sup>, 王俊义<sup>2</sup>

(1. 南京信息工程大学计算机学院, 南京 210044; 2. 桂林电子科技大学信息与通信学院, 桂林 541004)

**摘要:** 数据压缩技术通过压缩计算任务可以降低移动边缘计算(Mobile edge computing, MEC)网络中终端用户的卸载能耗。针对终端用户与基站之间的通信链路被障碍物阻挡对通信质量有影响问题, 为满足应急通信和节能卸载需求, 提出了一种无人机搭载中继设备和边缘服务器辅助MEC中基于数据压缩的任务卸载方案。考虑任务压缩比例、系统资源和无人机机载能量等约束条件, 建立用户总能耗最小化问题。将该非凸优化问题建模成一个马尔可夫决策过程, 使用深度强化学习中柔性演员-评论家算法求解。仿真结果表明, 所提方案具有良好的收敛性, 与基准算法相比, 能耗降低了24.7%~42.2%。

**关键词:** 移动边缘计算; 数据压缩; 无人机; 深度强化学习; 任务卸载

中图分类号: TN92

文献标志码: A

## Offloading Optimization Based on Data Compression in UAV-Assisted Edge Computing

LI Bin<sup>1</sup>, ZHU Xiao<sup>1</sup>, WANG Junyi<sup>2</sup>

(1. School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China; 2. School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** Data compression technology can reduce the offloading energy consumption of users in mobile edge computing (MEC) by compressing computing tasks. Aiming at the problem that the communication link between the mobile users and the base station is blocked, which has an impact on communication quality, this paper proposes a task offloading scheme based on data compression to meet the requirements of emergency communication and energy-saving offloading in MEC assisted by the unmanned aerial vehicle (UAV) equipped with relay devices and edge servers. Considering constraints such as task compression ratios, system resource and the onboard energy of UAV, we formulate a problem to minimize the sum energy consumption of users. The non-convex optimization problem is modeled as a Markov decision process and the soft actor-critic algorithm based deep reinforcement learning is used to tackle the problem. The simulation results reveal that the proposed scheme achieves better convergence performance and the total energy consumption of users can be reduced by 24.7%—42.2%, compared with the benchmark algorithms.

**Key words:** mobile edge computing(MEC); data compression; unmanned aerial vehicle(UAV); deep reinforcement learning; task offloading

基金项目: 国家自然科学基金(62101277, 62371149)。

收稿日期: 2023-12-17; 修订日期: 2024-02-07

## 引言

移动边缘计算(Mobile edge computing, MEC)利用网络边缘节点为用户提供了充足的计算资源,克服了用户与远程云之间距离较长的问题,并摆脱了回程容量拥塞的困境<sup>[1]</sup>。但是,在某些紧急情况下,MEC系统可能无法正常工作,从而导致计算服务中断<sup>[2]</sup>。例如,当终端与基站(Base station, BS)之间的链路被障碍物阻碍时,可能会导致通信质量变差,从而影响计算卸载过程。无人机(Unmanned aerial vehicle, UAV)由于其灵活性高、机动性强、可靠性高等优点被广泛应用于MEC中,在人口密集的城市或灾区可快速部署,为终端用户提供应急通信与计算服务<sup>[3-4]</sup>。因此,研究UAV辅助MEC具有重要的现实意义。

目前,已有许多关于UAV辅助MEC的探索研究。如文献[5]研究了UAV辅助MEC系统中总能耗最小化问题并且优化了UAV的路线规划和任务分配。文献[6]研究了在物联网设备任务和能源预算限制下,通过联合优化任务卸载、资源分配以及UAV轨迹最小化UAV能耗和完成时间。UAV由于其能量、尺寸、质量等限制携带的计算资源是有限的,当大量用户需要计算服务时,UAV计算资源可能会耗尽<sup>[7]</sup>。为了有效提高MEC网络的任务卸载能力,UAV中继辅助MEC模型被提出。文献[8]部署UAV用于改善用户和BS之间的连接,并且服务时延和UAV能耗的加权和被最小化。文献[9]研究了在任务、信息因果关系、带宽分配和UAV轨迹约束下最小化UAV和用户设备能耗的加权和。文献[10]优化UAV部署和资源分配来降低平均用户延迟,以达到保障服务质量的目的。上述研究工作假设UAV飞行在固定高度,不能很好地发挥UAV的优势。

近年来,数据压缩技术被广泛应用于各种类型的文件,如文本、音频、视频等。采用数据压缩技术,可以减小数据大小,从而降低任务卸载的时间和能耗。文献[11]采用无损压缩技术,在满足时延约束的条件下,通过联合优化计算卸载、数据压缩和资源分配,提出了多用户MEC系统中的能耗最小化问题。文献[12]考虑有损压缩技术,研究了3层MEC中基于深度学习服务的服务请求延迟和精度的权衡问题。文献[13]提出了一种基于MEC的有损压缩和无损压缩混合的数据压缩方案,以提高传感器的数据处理能力。文献[14]研究了本地压缩、边缘云压缩和部分压缩卸载3种压缩模型以最小化所有设备的时延总和。文献[15]研究了一种采用数据压缩技术的多用户MEC系统,通过联合优化传输带宽分配和数据压缩比例来最小化系统总能耗。上述工作采用数据压缩技术有效降低了MEC系统成本,但仅考虑了边缘服务器部署在地面BS的情况,这种模式并不适用于地形复杂的区域。

综上所述,UAV在MEC的应用中仍存在一些不足:(1)大多数研究只考虑了用户与UAV能耗最小化问题,而忽略了UAV机载能量有限的事实<sup>[9,16]</sup>,这可能导致UAV无法完成飞行任务;(2)主要集中于通过优化资源分配、传输功率、UAV轨迹等来最小化能耗,并未考虑采用数据压缩技术以进一步降低成本。针对以上问题,本文研究UAV中继MEC环境下用户总能耗最小化问题,主要贡献如下:

(1)提出了基于数据压缩的UAV中继MEC系统模型。考虑到UAV的机载能耗和飞行高度等实际因素,构建了飞行轨迹、资源分配、计算卸载策略与数据压缩的联合优化问题,旨在实现终端用户总能耗最小化。

(2)将问题建模为一个无转移概率的马尔可夫决策过程(Markov decision process, MDP),并根据场景与需求定义该过程的状态空间、动作空间和奖励函数,最后采用深度强化学习(Deep reinforcement learning, DRL)中的柔性演员-评论家(Soft actor-critic, SAC)算法求解,实现了节能卸载决策。

## 1 系统模型

如图1所示,本文考虑的MEC系统由 $K$ 个用户、单个UAV和单个BS组成。其中, $K$ 个用户随机分散在地面上的一个区域内,第 $k(k \in \mathcal{K} \triangleq \{1, 2, \dots, K\})$ 个用户的坐标表示为 $\boldsymbol{w}_k = [x_k, y_k, 0]$ ;BS配备有MEC服务器,可以处理用户卸载的任务,其位置表示为 $\boldsymbol{w}_{\text{BS}} = [x_{\text{BS}}, y_{\text{BS}}, 0]$ ,用户与BS之间无直接通信链路;UAV既可辅助终端计算,又可充当数据传输的媒介。UAV的飞行周期表示为 $T$ ,为保证在飞行过程中UAV与用户的相对位置近乎不变,将飞行周期 $T$ 划分为 $\delta = T/N$ 足够小的时隙,且仅在不同时隙下UAV的位置发生变化,时隙集合表示为 $\mathcal{N} \triangleq \{1, 2, \dots, N\}$ 。在每个时隙中,地面用户产生一个计算任务 $I_k[n] = \{D_k[n], C_k[n], t_k^{\max}\}$ ,其中, $D_k[n]$ 表示计算任务的数据量大小, $C_k[n]$ 表示计算密度, $t_k^{\max}$ 为任务最大容忍时间。

本文优化UAV三维飞行轨迹。在时隙 $n$ 中,UAV所处位置可以表示为 $\boldsymbol{q}[n] = [L[n], H[n]]$ ,其中 $L[n] = [x[n], y[n]]$ 为UAV的水平位置, $H[n]$ 为UAV的飞行高度。相邻时隙内UAV的位移变化与速度 $\boldsymbol{v}[n]$ 和加速度 $\boldsymbol{a}[n]$ 有关。因此,UAV位移需满足如下约束

$$\boldsymbol{q}[n+1] = \boldsymbol{q}[n] + \boldsymbol{v}[n]\delta + \frac{1}{2}\boldsymbol{a}[n]\delta^2 \quad (1)$$

$$\|\boldsymbol{q}[n+1] - \boldsymbol{q}[n]\| \leq v_{\max}\delta \quad \forall n \in \mathcal{N} \quad (2)$$

$$H_{\min} \leq H[n] \leq H_{\max} \quad \forall n \in \mathcal{N} \quad (3)$$

式中: $H_{\min}$ 和 $H_{\max}$ 分别表示为UAV的最小和最大飞行高度, $v_{\max}$ 为UAV最大飞行速度。

### 1.1 通信模型

为避免干扰,基于正交频分多址技术,每个用户被分配带宽资源 $B_k[n]$ 用于任务卸载。为了模拟真实无线传输环境,UAV与用户之间和UAV与BS的通信链路均采用莱斯信道模型<sup>[17]</sup>,信道增益分别表示为

$$h_{k,u}[n] = \sqrt{\beta_0 d_{k,u}^{-\alpha}[n]} g_{k,u} \quad \forall n \in \mathcal{N} \quad (4)$$

$$h_{u,\text{BS}}[n] = \sqrt{\beta_0 d_{u,\text{BS}}^{-\alpha}[n]} g_{u,\text{BS}} \quad \forall n \in \mathcal{N} \quad (5)$$

式中: $d_{k,u}[n] = \sqrt{\|\boldsymbol{q}[n] - \boldsymbol{w}_k\|^2}$ 为用户到UAV的距离, $d_{u,\text{BS}}[n] = \sqrt{\|\boldsymbol{q}[n] - \boldsymbol{w}_{\text{BS}}\|^2}$ 为UAV到BS之间的距离; $\beta_0$ 为参考距离为1 m时的信道增益, $\alpha \geq 2$ 表示路径损耗指数, $g_{k,u}$ 和 $g_{u,\text{BS}}$ 为考虑小尺度衰落的复值随机变量。 $E[|g_{k,u}|^2] = 1$ , $E[|g_{u,\text{BS}}|^2] = 1$ ,分别表示为

$$g_{k,u} = \sqrt{\frac{K_{k,u}}{K_{k,u} + 1}} g + \sqrt{\frac{1}{K_{k,u} + 1}} \tilde{g} \quad (6)$$

$$g_{u,\text{BS}} = \sqrt{\frac{K_{u,\text{BS}}}{K_{u,\text{BS}} + 1}} g + \sqrt{\frac{1}{K_{u,\text{BS}} + 1}} \tilde{g} \quad (7)$$

式中: $g$ 表示视距信道部分; $\tilde{g}$ 为一个零均值、单位方差的、循环对称复高斯变量,表示随机散射成分; $K_{k,u}$ 和 $K_{u,\text{BS}}$ 为莱斯因子。用户传输功率为 $P_k$ , $P_{u,k}$ 为UAV转发用户卸载任务的功率,则用户到UAV和UAV至BS最大可实现传输速率分别表示为

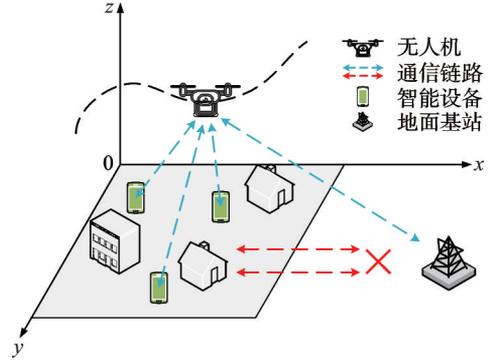


图1 系统模型

Fig.1 System model

$$r_{k,u}[n] = B_k[n] \log_2 \left( 1 + \frac{P_k |h_{k,u}[n]|^2}{B_k[n] N_0} \right) \quad (8)$$

$$r_{u,BS}[n] = B_k[n] \log_2 \left( 1 + \frac{P_{u,k} |h_{u,BS}[n]|^2}{B_k[n] N_0} \right) \quad (9)$$

式中  $N_0$  为噪声功率谱密度。由于随机变量  $|h_{k,u}[n]|^2$  和  $|h_{u,BS}[n]|^2$  的存在,在 UAV 飞行之前,  $r_{k,u}[n]$  和  $r_{u,BS}[n]$  并不完全可知,为了避免推导复杂的累计分布函数,本文采用逻辑斯蒂函数近似替换  $|h_{k,u}[n]|^2$  和  $|h_{u,BS}[n]|^2$ <sup>[17]</sup>,重写  $r_{k,u}[n]$  和  $r_{u,BS}[n]$  得

$$r_{k,u}[n] = B_k[n] \log_2 \left( 1 + \frac{P_k \beta_0 v_{k,u}}{B_k[n] N_0 (\|\mathbf{q}[n] - \mathbf{w}_k\|^2)^{\alpha/2}} \right) \quad (10)$$

$$r_{u,BS}[n] = B_k[n] \log_2 \left( 1 + \frac{P_{u,k} \beta_0 v_{u,BS}}{B_k[n] N_0 (\|\mathbf{q}[n] - \mathbf{w}_{BS}\|^2)^{\alpha/2}} \right) \quad (11)$$

式中  $v_{k,u}$  和  $v_{u,BS}$  为近似功率衰落函数<sup>[18]</sup>,其定义分别为

$$v_{k,u} = C_1 + \frac{C_2}{1 + e^{-(B_1 + B_2 u_{k,u})}} \quad (12)$$

$$v_{u,BS} = C_1 + \frac{C_2}{1 + e^{-(B_1 + B_2 u_{u,BS})}} \quad (13)$$

式中:  $B_1$ 、 $B_2$ 、 $C_1$  和  $C_2$  为由环境决定的常数,  $B_1 < 0$  表示正向逻辑斯蒂函数中点,  $B_2 > 0$  为逻辑斯蒂增长率,  $C_1 > 0$  且  $C_2 > 0$  并满足  $C_1 + C_2 = 1$ <sup>[19]</sup>。此外,  $u_{k,u}$  和  $u_{u,BS}$  分别表示用户到 UAV 和 UAV 至 BS 仰角的正弦值,定义为

$$u_{k,u} = \frac{H[n]}{\sqrt{\|\mathbf{q}[n] - \mathbf{w}_k\|^2}} \quad (14)$$

$$u_{u,BS} = \frac{H[n]}{\sqrt{\|\mathbf{q}[n] - \mathbf{w}_{BS}\|^2}} \quad (15)$$

## 1.2 计算模型

本文假设终端不执行任务计算,仅做压缩处理。任务首先在终端被压缩,然后卸载至 UAV, UAV 处理部分计算任务并将剩余部分进一步卸载到 BS 处执行。

(1) 数据压缩。采用无损压缩技术并且假设所有用户采用相同的压缩算法,  $\gamma_k[n] \in [\gamma_{\min}, 1]$  表示任务压缩比例,其中  $\gamma_{\min}$  为最小压缩比例。在当前的文献中,关于数据压缩的计算复杂度仍缺乏公认的模型。文献[20]采用了一个易处理的模型,将  $J_k[n]$  (压缩 1 bit 数据需要的 CPU 转数) 近似为压缩比例  $\gamma_k[n]$  的指数函数,具体表示为

$$J_k[n] = e^{\epsilon/\gamma_k[n]} - e^\epsilon \quad (16)$$

式中:  $\epsilon$  为取决于压缩方法的正常数,当  $\gamma_k[n] = 1$  时,  $J_k[n] = 0$ , 表示压缩算法不执行。本地压缩的时延  $t_{k,tr}[n]$  和能耗  $e_{k,tr}[n]$  分别表示为

$$t_{k,tr}[n] = \frac{D_k[n] J_k[n]}{F_k} \quad (17)$$

$$e_{k,tr}[n] = D_k[n] J_k[n] F_k^2 \tau_k \quad (18)$$

式中:  $F_k$  表示用户  $k$  的 CPU 频率,  $\tau_k$  为 CPU 的有效电容系数,取决于处理器芯片的结构。本文假设终端有足够的能源用于压缩和卸载操作。此外,解压成本相较于压缩小得多,因此忽略不计。

(2) 任务卸载。为了保证卸载任务的数据完整性,假设当压缩操作完成之后才会执行任务卸载。

用户  $k$  卸载花费的时间  $t_{k, \text{off}}[n]$  和能量  $e_{k, \text{off}}[n]$  分别表示为

$$t_{k, \text{off}}[n] = \frac{\gamma_k[n] D_k[n]}{r_{k, u}[n]} \quad (19)$$

$$e_{k, \text{off}}[n] = t_{k, \text{off}} P_k \quad (20)$$

UAV 的中继设备以半双工模式运行, 其卸载用户  $k$  任务的时延  $t_{u, kr}[n]$  和能耗  $e_{u, kr}[n]$  可表示为

$$t_{u, kr}[n] = \frac{\alpha_k[n] \gamma_k[n] D_k[n]}{r_{u, \text{BS}}[n]} \quad (21)$$

$$e_{u, kr}[n] = t_{u, kr} P_{u, k} \quad (22)$$

式中  $\alpha_k[n] \in [0, 1]$  为 UAV 卸载比例。此外, 旋翼 UAV 在时隙  $n$  中的飞行能耗<sup>[21]</sup>可表示为

$$e_{\text{fly}}[n] = \delta \left( P_0 \left( 1 + \frac{3(v^h[n])^2}{U_{\text{tip}}^2} \right) + \frac{1}{2} d_0 \rho s G (v^h[n])^3 + P_1 \left( \sqrt{1 + \frac{(v^h[n])^4}{4v_0^4}} - \frac{(v^h[n])^2}{2v_0^2} \right)^{\frac{1}{2}} + P_2 v^v[n] \right) \quad (23)$$

式中  $v^h[n]$  表示 UAV 在时隙  $n$  中水平方向飞行速度, 其定义为

$$v^h[n] = \frac{\|L[n+1] - L[n]\|}{\delta} \quad (24)$$

若  $v^h[n] = 0$ , 则表示 UAV 悬停; 类似地,  $v^v[n]$  表示 UAV 在垂直方向上的速度, 定义为

$$v^v[n] = \frac{\|H[n+1] - H[n]\|}{\delta} \quad (25)$$

式中:  $P_0$  和  $P_1$  分别表示悬停状态下型阻功率和诱导功率,  $P_2$  表示恒定上升或下降功率,  $U_{\text{tip}}$  为旋翼的叶尖角速度,  $v_0$  为悬停状态下平均旋翼诱导速度,  $d_0$  和  $s$  分别表示机身阻力比和旋翼实度,  $\rho$  和  $G$  分别表示空气密度和旋翼盘面积<sup>[21]</sup>。

(3) UAV 计算。  $F_{u, k}$  表示 UAV 分配给用户  $k$  的计算资源,  $\tau_u$  为 UAV 的 CPU 有效电容系数, 则 UAV 处理用户任务的时延  $t_{u, k}^c$  和能耗  $e_{u, k}^c$  分别表示为

$$t_{u, k}^c[n] = \frac{(1 - \alpha_k[n]) C_k[n] D_k[n]}{F_{u, k}} \quad (26)$$

$$e_{u, k}^c[n] = (1 - \alpha_k[n]) C_k[n] D_k[n] \tau_u F_{u, k}^2 \quad (27)$$

(4) BS 计算。由于 BS 通常有充足的能源供应并且计算资源丰富<sup>[18]</sup>, 因此仅考虑计算时延, 边缘服务器计算用户  $k$  卸载任务耗费的时延  $t_{\text{BS}, k}^c[n]$  可表示为

$$t_{\text{BS}, k}^c[n] = \frac{\alpha_k[n] C_k[n] D_k[n]}{F_{\text{BS}}} \quad (28)$$

式中  $F_{\text{BS}}$  表示 BS 计算资源。

为了定义用户计算时延, 本文做出如下假设: (1) 为了保证任务划分的准确性, UAV 只有在接受到完整的输入数据后, 才会对任务进行划分; (2) 为了保证计算结果的可靠性, UAV 只有与用户之间通信结束才会开始处理任务, 同样地, BS 处计算只有当 UAV 与 BS 之间通信结束才会执行任务计算; (3) UAV 上计算模块与通信模块通常是分离的, 因此 UAV 计算可与任务传输同时进行<sup>[8]</sup>。基于上述假设, 用户  $k$  完成计算任务耗费的总时延  $t_k[n]$  可表示为

$$t_k[n] = t_{k, lr}[n] + t_{k, \text{off}}[n] + \max \{ t_{u, k}^c[n], t_{u, kr}[n] + t_{\text{BS}, k}^c[n] \} \quad (29)$$

用户  $k$  总能耗  $e_k[n]$  包含压缩与卸载两部分, 可表示为

$$e_k[n] = e_{k, lr}[n] + e_{k, \text{off}}[n] \quad (30)$$

## 2 优化问题描述

本文通过联合优化 UAV 的发送功率  $P_u = \{P_{u,k}[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$ 、飞行轨迹  $q = \{q[n], \forall n \in \mathcal{N}\}$ 、用户发送功率  $P = \{P_k[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$ 、卸载比例  $\alpha = \{\alpha_k[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$ 、压缩比例  $\gamma = \{\gamma_k[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$ 、UAV 计算资源分配  $F_u = \{F_{u,k}[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$  和带宽资源分配  $B = \{B_k[n], \forall k \in \mathcal{K}, \forall n \in \mathcal{N}\}$  以实现用户能耗最小化。具体优化问题表述如下

$$\begin{aligned}
 & \min_{q, P_u, P, \alpha, \gamma, B, F_u} \sum_{n=1}^N \sum_{k=1}^K e_k[n] \\
 & \text{s.t. C1: } \|a[n]\| \leq a_{\max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C2: } \|v[n]\| \leq v_{\max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C3: } \|q[n+1] - q[n]\| \leq v_{\max} \delta \quad \forall n \in \mathcal{N} \\
 & \quad \text{C4: } H_{\min} \leq H[n] \leq H_{\max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C5: } P_{\min} \leq P_k[n] \leq P_{\max} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \\
 & \quad \text{C6: } P_{u, \min} \leq P_{u,k}[n] \leq P_{u, \max} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \\
 & \quad \text{C7: } \sum_{n=1}^N \sum_{k=1}^K e_{u,kr}[n] + e_{u,k}^c[n] + e_{\text{fly}}[n] \leq E_{u, \max} \\
 & \quad \text{C8: } \alpha_k[n] \in [0, 1] \quad \forall n \in \mathcal{N} \\
 & \quad \text{C9: } \gamma_k[n] \in [\gamma_{\min}, 1] \quad \forall n \in \mathcal{N} \\
 & \quad \text{C10: } \sum_{k=1}^K B_k[n] \leq B_{\max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C11: } B_{k, \min} \leq B_k[n] \leq B_{\max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C12: } \sum_{k=1}^K F_{u,k}[n] \leq F_{u, \max} \quad \forall n \in \mathcal{N} \\
 & \quad \text{C13: } 0 \leq F_{u,k}[n] \leq F_{u, \max} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \\
 & \quad \text{C14: } t_k[n] \leq t_k^{\max} \quad \forall k \in \mathcal{K}, \forall n \in \mathcal{N} \tag{31}
 \end{aligned}$$

式中:  $E_{u, \max}$  和  $B_{\max}$  分别表示 UAV 总能量和系统总带宽,  $a_{\max}$  为 UAV 最大加速度,  $B_{k, \min}$  表示分配给终端最小带宽资源,  $P_{\max}$  和  $P_{u, \max}$  分别表示用户和 UAV 的最大发射功率,  $P_{\min}$  和  $P_{u, \min}$  分别表示用户和 UAV 的最小发射功率,  $F_{u, \max}$  为 UAV 计算资源。约束条件 C1、C2、C3 和 C4 是 UAV 位移约束, 约束条件 C5 和 C6 是用户和 UAV 发送功率约束, 约束条件 C7 是 UAV 能耗约束, 约束条件 C8 和 C9 分别为卸载比例和压缩比例约束, 约束条件 C10 和 C11 为系统带宽资源分配约束, 约束条件 C12 和 C13 为 UAV 计算资源分配约束, 约束条件 C14 为任务处理时延约束。

式(31)是一个非凸优化问题且变量耦合度较高, 采用传统的优化算法难以求解。即使将问题分解为多个子问题, 也需要进行复杂的数学分析和公式推导。深度强化学习作为机器学习的重要分支, 可在不使用复杂的数值优化算法的情况下解决决策优化问题, 具有强大的数据处理能力。

## 3 基于 DRL 的用户总能耗优化方案

MDP 提供了强化学习中环境模型的形式化表示, 包括状态、动作、奖励以及状态转移概率等。通过建立与 MDP 相对应的强化学习问题, 可以使用强化学习算法来解决决策问题, 并找到最优的策略以最大化长期累积奖励。本节首先介绍强化学习中 MDP 的基本要素, 然后提出基于 SAC 的用户总能耗最小化算法。

### 3.1 MDP基本要素

深度强化学习由智能体、环境和强化学习算法3个部分组成,其目标问题通常被描述成一个马尔可夫决策过程,MDP由五元素 $(S, A, R, P, \gamma)$ 组成,其中 $S, A, R, P$ 和 $\gamma$ 分别表示状态空间、动作空间、奖励函数、状态转移概率函数和折扣因子。在本文场景中,智能体不需要任何关于环境的先验信息,只能从环境状态中获取因果信息,因此本文模型中的状态转移概率未知。 $\gamma \in [0, 1]$ 被定义为常数,用于权衡当前奖励和未来奖励的重要性。其余三要素( $S, A$ 和 $R$ )分别定义如下:

#### (1) 状态空间

状态是智能体决策的依据。首先,用户 $k$ 的压缩和卸载能耗与任务数据量大小 $D_k[n]$ 和计算密度 $C_k[n]$ 直接相关,其次,UAV位置受速度 $v[n]$ 影响并且UAV的位置会影响用户和UAV的传输速率,从而间接作用于用户卸载能耗。此外,UAV的电池容量有限,UAV剩余电量 $E_{\text{battery}}$ 也应考虑。因此,第 $n$ 个时隙的状态 $s_n$ 可以表示为

$$s_n = (\mathbf{q}[n], \mathbf{v}[n], E_{\text{battery}}, D_k[n], C_k[n], \forall k \in \mathcal{K}) \quad (32)$$

#### (2) 动作空间

动作是智能体决策的结果。在时隙 $n$ 中,UAV在执行飞行动作 $\mathbf{a}[n]$ 后,如果超出了指定区域,则将保持在其先前的位置。此外,智能体的动作还包括用户卸载功率 $P_k[n]$ 、UAV转发功率 $P_{u,k}[n]$ 、卸载比例 $\alpha_k[n]$ 、压缩比例 $\gamma_k[n]$ 、UAV计算资源分配 $F_{u,k}[n]$ 和带宽资源分配 $B_k[n]$ 。因此,第 $n$ 个时隙的动作可以表示为

$$\mathbf{a}_n = \{\mathbf{a}[n], p_k[n], p_{u,k}[n], \alpha_k[n], \gamma_k[n], F_{u,k}[n], B_k[n], \forall k \in \mathcal{K}\} \quad (33)$$

#### (3) 奖励函数

奖励是智能体执行动作后的反馈。奖励函数涉及优化问题的目标和不能满足约束的惩罚,其中,优化目标可表示为

$$E_{\text{sum}}[n] = \sum_{k=1}^K e_k[n] \quad (34)$$

式中 $E_{\text{sum}}[n]$ 表示时隙 $n$ 中所有用户总能耗。惩罚值则通过惩罚函数计算,表示为

$$P(x, a, b) = 2 - \exp(-[(x - a)/b]^+) \quad (35)$$

式中 $[x]^+ = \max(0, x)$ 。当约束条件C7不满足时,即UAV的能量不足时惩罚值 $P_E[n]$ 为

$$P_E[n] = P(E_{u,\text{sum}}[n], E_{u,\text{max}}, E_{u,\text{max}}) \quad (36)$$

式中 $E_{u,\text{sum}}[n]$ 表示前 $n$ 个时隙UAV总能耗,定义为

$$E_{u,\text{sum}}[n] = \sum_{i=1}^n \sum_{k=1}^K e_{u,kr}[i] + e_{u,k}^c[i] + e_{\text{fly}}[i] \quad (37)$$

当约束条件C14不满足时,即用户任务处理时延超过最大容忍时延惩罚值 $P_T[n]$ 为

$$P_T[n] = \frac{1}{K} \sum_{k=1}^K P(t_k[n], t_{k,\text{max}}, t_{k,\text{max}}) \quad (38)$$

因此,奖励函数可定义为

$$r[n] = -E_{\text{sum}}[n]P_T[n]P_E[n] \quad (39)$$

### 3.2 基于SAC的用户能耗最小化算法

深度强化学习通过与环境交互寻找最优策略,按动作空间离散化程度可分为两类:(1)离散动作空间,如深度Q网络、双重深度Q网络等;(2)连续动作空间,如深度确定性策略梯度算法、近端策略优化算法、SAC算法等。式(31)中卸载比例、计算资源分配等优化变量均是连续的,因此离散动作空间算法

并不适用。相较于其他连续动作空间算法,SAC算法具有以下优势:(1)与确定性策略梯度算法相比,SAC算法 Actor网络采用随机策略梯度来产生连续动作的概率分布,便于对动作空间的探索,避免潜在的局部最优;(2)该算法是异策略强化学习算法,具有更高的样本利用率;(3)该算法是一种基于最大熵框架的异策略强化学习算法,具有很强的探索性和鲁棒性,可应用于复杂的动态环境中。SAC算法的目标是最大化奖励与熵值加权后的期望值,最优策略  $\pi^*$  可表示为

$$\pi^* = \arg \max_{\pi} E_{(s_n, a_n) \sim \rho_{\pi}} \left[ \sum_{n=0}^T \gamma^n (r_n(s_n, a_n) + \alpha H(\pi(\cdot|s_n))) \right] \quad (40)$$

式中:  $\rho_{\pi}$  表示在策略  $\pi$  下状态和动作的分布,  $\alpha$  表示熵权重,用于平衡熵  $H$  和系统奖励之间的重要性。熵  $H$  的计算方式如下

$$H(\pi(\cdot|s_n)) = E_{a_n} [-\log \pi(a_n|s_n)] \quad (41)$$

基于 SAC 的 DRL 训练框架如图 2 所示,该算法的智能体包含 1 个 Actor 网络、2 个 Critic 网络和 2 个 Critic 网络的目标网络。Actor 网络用于近似策略函数  $\pi_{\varphi}(a_n|s_n)$ ,将状态输入 Actor 网络,输出高斯分布的均值和方差,智能体的动作采样自该分布。Critic 网络用于近似柔性  $Q_{\theta_i}(s_n, a_n)$  函数,作用是评估 Actor 网络的性能,目标网络输出目标值  $Q_{\bar{\theta}_i}(s_n, a_n)$ ,用于缓解训练过程中产生的 Q 值偏差问题,提高训练的稳定性。其中,  $i \in \{1, 2\}$ ,  $\varphi$ 、 $\theta_i$  和  $\bar{\theta}_i$  分别是 Actor 网络、Critic 网络和目标网络的参数。本文将离线训练过程部署在 BS 或其他计算资源丰富的节点上。在每个时隙中,智能体会根据系统当前状态利用 Actor 网络计算决策结果,UAV 与终端用户执行相应动作后,系统环境会反馈奖励并更新至新状态。智能体将与环境交互的记录存放在经验缓冲区  $D$ ,并在计算资源空闲时随机采样小批次样本  $B$  训练 Actor 网络和 Critic 网络。Critic 网络的损失函数  $L_Q(\theta_i)$  如下

$$L_Q(\theta_i) = E_{(s_n, a_n, s_{n+1}) \sim B, a_{n+1} \sim \pi_{\varphi}} \left[ \frac{1}{2} \left( Q_{\theta_i}(s_n, a_n) - \left( r(s_n, a_n) + \gamma \left( \min_{i=1,2} Q_{\bar{\theta}_i}(s_{n+1}, a_{n+1}) - \log \pi_{\varphi}(a_{n+1}|s_{n+1}) \right) \right) \right)^2 \right] \quad (42)$$

Actor 网络的损失函数  $L_{\pi}(\varphi)$  表示为

$$L_{\pi}(\varphi) = E_{s_n \sim B} \left[ E_{a_n \sim \pi_{\varphi}} \left[ \alpha \log \pi_{\varphi}(a_n|s_n) - \min_{i=1,2} Q_{\theta_i}(s_n, a_n) \right] \right] \quad (43)$$

根据损失函数  $L_Q(\theta_i)$  和  $L_{\pi}(\varphi)$ ,采用梯度下降法更新 Actor 网络和 Critic 网络的参数  $\varphi$  和  $\theta_i$ 。目标

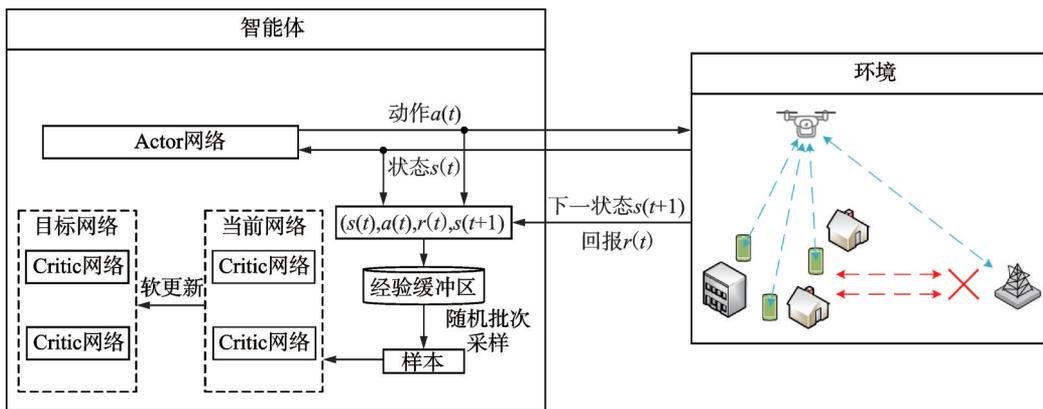


图 2 SAC 算法训练框架

Fig.2 Training framework of SAC algorithm

网络则进行“软”更新,即通过缓慢跟踪 Critic 网络参数以更新目标网络的参数,  $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$ , 其中,  $\tau$  为软更新学习步长。

此外,本文采用基于自适应梯度的方法来调整熵权重  $\alpha$ 。当智能体探索新区域时,最优动作尚不明确,增大  $\alpha$  可以探索更多空间,随着最优动作逐步确定,  $\alpha$  会逐渐减小。调节  $\alpha$  的损失函数  $L(\alpha)$  表示为

$$L(\alpha) = E_{a_n \sim \pi_\varphi} [-\alpha \log \pi_\varphi(a_n | s_n) - \alpha \bar{H}] \quad (44)$$

式中  $\bar{H}$  为预定义的最小熵阈值。基于以上叙述,本文所提基于 SAC 的用户总能耗最小化算法流程如表 1 所示。

表 1 基于 SAC 算法的用户总能耗最小化算法

Table 1 User total energy minimization algorithm based on SAC

输入:	初始化神经网络参数 $\varphi, \theta_1$ 和 $\theta_2, \bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ , 初始化经验缓冲区 $D \leftarrow \{\}$ , 随机初始化用户坐标
输出:	神经网络参数 $\varphi, \theta_1$ 和 $\theta_2$
(1)	for each episode do
(2)	初始化 UAV 状态和用户任务信息
(3)	for each environment step do
(4)	$a_n \sim \pi_\varphi(a_n   s_n)$
(5)	系统执行动作 $a_n$ 后进入下一状态 $s_{n+1}$ , 根据式(36)计算奖励 $r[n]$
(6)	$D \leftarrow \{(s_n, a_n, r[n], s_{n+1})\} \cup D$
(7)	end for
(8)	for each gradient step do
(9)	从经验缓冲区随机采样小批次经验样本 $B$ , 根据式(42~44), 计算损失函数 $L_Q(\theta_i), L_\pi(\varphi)$ 和 $L(\alpha)$ , 更新神经网络参数和熵权重
(10)	end for
(11)	end for

### 3.3 计算复杂度分析

在训练阶段,所提算法的计算复杂度主要取决于训练 Actor 和 Critic 神经网络所需的时间,一次迭代过程中 Actor 和 Critic 网络的梯度下降复杂度可表示为

$$O\left(N_B \left( \sum_{i=0}^{I-1} l_i l_{i+1} + \sum_{j=0}^{J-1} \hat{l}_j \hat{l}_{j+1} \right)\right) \quad (45)$$

式中:  $N_B$  为小批次样本的大小,  $I$  和  $J$  分别表示 Actor 和 Critic 网络中全连接层的数量,  $l_i$  和  $\hat{l}_j$  分别表示 Actor 网络第  $i$  层和 Critic 网络第  $j$  层神经元数量<sup>[18]</sup>。在执行阶段,只需要应用训练好的 Actor 网络,因此每个时隙的计算复杂度等于 Actor 网络前向传播的计算复杂度,可以表示为

$$O\left(\sum_{j=0}^{J-1} \hat{l}_j \hat{l}_{j+1}\right) \quad (46)$$

## 4 仿真结果与分析

### 4.1 参数设置

本文仿真环境基于 Python3.8 和 Pytorch1.11 编写。用户数量  $K$  为 15, 随机分散在地面上  $600 \text{ m} \times$

600 m的区域内。BS位置为[600, 600, 0] m,UAV的初始位置设置为[0, 0, 150] m,飞行高度限制在[100, 200] m,最大速度 $v_{\max}$ 为20 m/s,最大加速度 $a_{\max}$ 为5 m/s<sup>2</sup>。噪声功率谱密度为-174 dBm/Hz。任务量大小 $D_k[n] \in [1, 2.5]$  Mb,单位比特平均计算量 $C_k[n] \in [700, 1000]$  cycles/b。为简单起见,本文任务处理时延 $t_k^{\max}$ 设置为时隙长度。其他实验参数结合文献[19]和文献[21]设置,如表2所示。本文神经网络采用ReLU激活函数,设置两个全连接隐藏层,神经元数量分别为128和64,折扣因子为0.9,经验回放池 $D$ 设置为20 000,随机采样的小批次样本 $B$ 数量为256,Actor网络和Critic网络的学习率均为0.001,采用Adam优化器更新模型参数。

为评价基于SAC算法的性能,本文将其与以下基准算法进行比较。

(1) 近端策略优化(Proximal policy optimization,PPO)算法。PPO是一种同策略强化学习算法,其采用随机策略,策略网络输出高斯分布的均值和方差,实际动作采样自该分布。

(2) 深度确定性策略梯度(Deep deterministic policy gradient,DDPG)算法。DDPG是一种异策略DRL算法,策略网络直接输出动作值,通过采用噪声探索机制,更好地发现环境中的未知信息,提高算法的学习效率和性能。

(3) 双延迟确定性策略梯度(Twin delayed deep deterministic policy gradient,TD3)算法。TD3算法是在DDPG的基础上采用双Q网络、延迟更新和噪声调节技巧改进版本,具有更好的性能和稳定性。

## 4.2 数值分析

图3表示不同算法的奖励收敛性对比图。可以观察到,随着训练步数增加,4种算法奖励均呈上升趋势并且最终能收敛到较为稳定的奖励值,这表明强化学习智能体通过与环境交互能够学习到较优的策略以降低终端用户能耗。从收敛速度方面看,SAC和DDPG算法于 $8 \times 10^4$ 步左右收敛,PPO和TD3算法于 $6 \times 10^4$ 步左右收敛,在收敛速度方面所提方案略逊色于PPO和TD3算法。从收敛稳定性方面看,PPO算法最平稳,SAC算法次之,而TD3和DDPG算法收敛波动性较大,这是因为TD3和DDPG算法仅仅通过增加噪声来提高探索能力,可能会陷入局部最优解。从收敛效果方面看,本文采用的SAC算法拥有最好的奖励以及较平稳的收敛曲线,这是因为SAC算法是基于最大熵框架的强化学习算法,拥有更强的探索能力和较少的敏感超参数。

表2 相关参数设置

Table 2 Related parameter settings

参数	取值
周期长度 $T/s$	75
时隙数 $N$	50
用户最小和最大发射功率 $P/W$	0.1, 0.5
UAV最小和最大转发功率 $P_u/W$	0.1, 0.5
用户计算频率 $F_k/GHz$	1.5
UAV计算频率 $F_{u, \max}/GHz$	30
BS计算频率 $F_{BS}/GHz$	15
带宽资源 $B_{\max}/MHz$	30
终端最小带宽资源 $B_{k, \min}/MHz$	1
最小压缩比 $\gamma_{\min}$	0.5
UAV电池容量 $E_{u, \max}/J$	20 000
CPU电容系数 $\tau_k$	$10^{-29}$

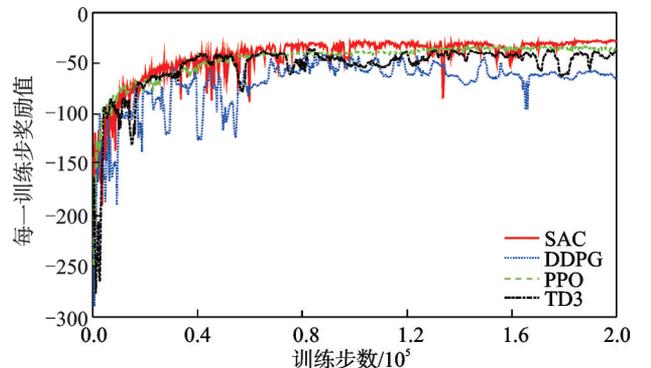


图3 不同算法收敛性

Fig.3 Convergence performance of different algorithms

图4对比了不同算法在不同任务量下的性能。除了PPO、TD3和DDPG算法外,还加入了“无压缩-SAC”对比方案,该方案除了不采用数据压缩技术以外,其余与所提方案一致。从该图可以观察到,任务数据量从[1, 1.25] Mb增大到[2.25, 2.5] Mb时,本文所提算法用户总能耗最低。相较于TD3、PPO和DDPG算法,总能耗分别降低了19.9%~21.3%、22.0%~25.1%和31.0%~34.0%,由此可见,SAC算法探索到的策略优于TD3、PPO和DDPG算法。相较于“无压缩-SAC”方案,用户总能耗降低了9.2%~11.6%,可以看出采用数据压缩技术可以有效降低用户总能耗并且随着任务数据量增加,数据压缩技术的效果愈加明显。

图5给出了终端用户能耗与用户数量之间的关系。随着用户数量增加,总能耗呈上升趋势,一方面是因为用户数量增加带来的总能耗提高;另一方面是随着用户数量增加,由于系统通信资源有限,导致终端竞争资源,从而造成能耗上升。与其他3种算法相比,本文采用的算法能够更合理地分配系统资源,总能耗始终保持最低。例如,当用户数量为18时,相较于TD3、PPO和DDPG算法,能耗分别降低了12.0%、24.3%和32.2%。

图6显示了终端总能耗与系统带宽资源的关系。随着系统带宽资源增加,总能耗呈下降趋势,原因在于,随着带宽资源增加,终端可获得更多的通信资源,能够实现更高的传输速率,从而降低卸载能耗。从图中可以观察到,与其他算法相比,本文方案可以更好地降低能耗。当系统带宽资源为60 MHz时,本文方法比TD3、PPO和DDPG算法分别降低了6.8%、26.7%和28.0%。

图7和图8展示了本文方案UAV飞行轨迹情况。结合两图可以观察到,UAV一方面升高飞行高度,另一方面会飞向终端用户分布的中心区域,尽量与所有用户保持较近的距离。这是因为靠近用户可以减少路径损耗,适当提高飞行高度可以增大用户至UAV的仰角,从而获得更大的有效衰落功率,进而实现更高的传输速率,降低用户卸载能耗。

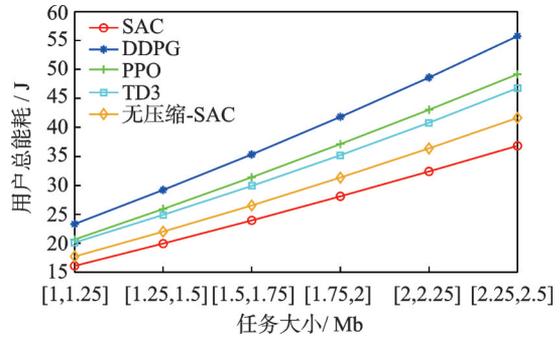


图4 不同任务量下算法性能对比

Fig.4 Comparison of algorithm performance under different tasks

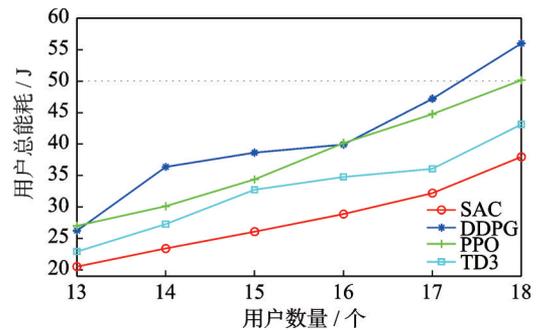


图5 不同用户数量下算法性能对比

Fig.5 Comparison of algorithm performance under different user numbers

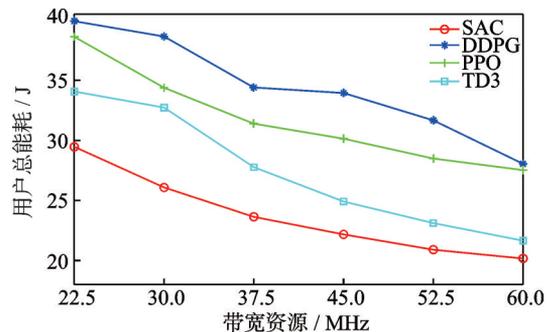


图6 不同带宽资源下算法性能对比

Fig.6 Comparison of algorithm performance under different bandwidth resources

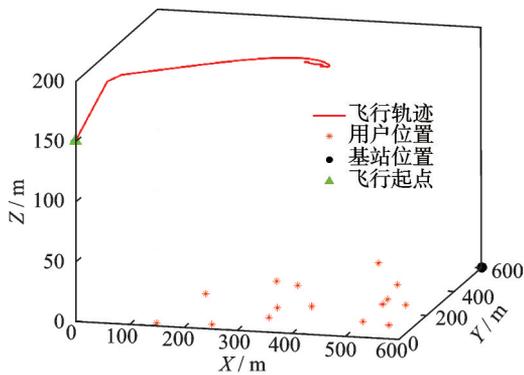


图7 无人机三维飞行轨迹

Fig.7 UAV 3D flight trajectory

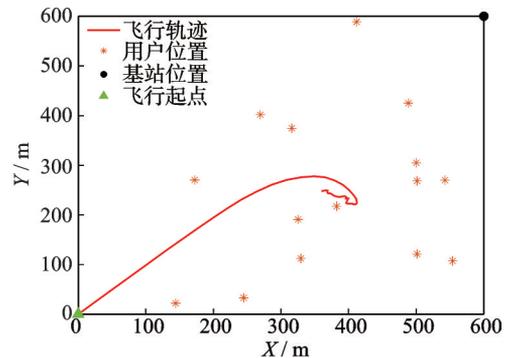


图8 无人机二维飞行轨迹

Fig.8 UAV 2D flight trajectory

## 5 结束语

本文研究了无人机中继辅助边缘计算中基于数据压缩的任务卸载方案。首先,通过联合考虑无人机轨迹、卸载比例、压缩比例、计算资源分配、带宽资源分配和功率控制,建立用户总能耗最小化问题;其次,将该问题描述成一个马尔可夫决策过程并采用SAC算法求解。仿真结果表明所提算法能有效降低用户总能耗。未来工作将考虑多跳MEC场景中多无人机协同工作与任务卸载问题。

## 参考文献:

- [1] CHENG K J, FANG X M, WANG X B. Energy efficient edge computing and data compression collaboration scheme for UAV-assisted network[J]. IEEE Transactions on Vehicular Technology, 2023, 72(12): 16395-16408.
- [2] 王心一, 陈志江, 雷磊, 等. 多无人机网络边缘智能计算卸载算法[J]. 数据采集与处理, 2023, 38(6): 1286-1298.  
WANG Xinyi, CHEN Zhijiang, LEI Lei, et al. Computation offloading algorithm for multi-UAV network based on edge intelligence[J]. Journal of Data Acquisition and Processing, 2023, 38(6): 1286-1298.
- [3] 黄斐, 李广侠, 王海超, 等. 高效节能的无人机多用户携能传输优化研究[J]. 数据采集与处理, 2022, 37(6): 1297-1313.  
HUANG Fei, LI Guangxia, WANG Haichao, et al. Optimization research on energy-efficient UAV-enabled multiuser SWIPT [J]. Journal of Data Acquisition and Processing, 2022, 37(6): 1297-1313.
- [4] DIAO X B, YANG W D, YANG L X, et al. UAV-relaying-assisted multi-access edge computing with multi-antenna base station: Offloading and scheduling optimization[J]. IEEE Transactions on Vehicular Technology, 2021, 70(9): 9495-9509.
- [5] LI Y Y, FANG Y, QIU L. Joint computation offloading and communication design for secure UAV-enabled MEC systems [C]//Proceedings of 2021 IEEE Wireless Communications and Networking Conference (WCNC). Nanjing: IEEE, 2021: 1-6.
- [6] ZHAN C, HU H, SUI X F, et al. Completion time and energy optimization in the UAV-enabled mobile-edge computing system[J]. IEEE Internet of Things Journal, 2020, 7(8): 7808-7822.
- [7] LIU Z J, TAN X, WEN M W, et al. An energy-efficient selection mechanism of relay and edge computing in UAV-assisted cellular networks[J]. IEEE Transactions on Green Communications and Networking, 2021, 5(3): 1306-1318.
- [8] 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.
- [9] HU X Y, WANG K K, YANG K, et al. UAV-assisted relaying and edge computing: Scheduling and trajectory optimization [J]. IEEE Transactions on Wireless Communications, 2019, 18(10): 4738-4752.
- [10] ZHANG L, ANSARI N. Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks[J]. IEEE Internet of Things Journal, 2020, 7(10): 10573-10580.
- [11] XU D, LI Q, ZHU H B. Energy-saving computation offloading by joint data compression and resource allocation for mobile-edge computing[J]. IEEE Communications Letters, 2019, 23(4): 704-707.

- [12] HOSSEINZADEH M, HUDSON N, ZHAO X, et al. Joint compression and offloading decisions for deep learning services in 3-tier edge systems[C]//Proceedings of 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). Los Angeles: IEEE, 2021: 254-261.
- [13] LU S F, XIA Q H, TANG X L, et al. A reliable data compression scheme in sensor-cloud systems based on edge computing [J]. IEEE Access, 2021, 9: 49007-49015.
- [14] REN J K, YU G D, CAI Y L, et al. Latency optimization for resource allocation in mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(8): 5506-5519.
- [15] WANG J B, ZHANG J Y X, DING C F, et al. Joint optimization of transmission bandwidth allocation and data compression for mobile-edge computing systems[J]. IEEE Communications Letters, 2020, 24(10): 2245-2249.
- [16] EI N N, ALSENWI M, TUN Y K, et al. Energy-efficient resource allocation in multi-UAV-assisted two-stage edge computing for beyond 5G networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 16421-16432.
- [17] YOU C S, ZHANG R. 3D trajectory optimization in rician fading for UAV-enabled data harvesting[J]. IEEE Transactions on Wireless Communications, 2019, 18(6): 3192-3207.
- [18] HEIDARPOUR A R, HEIDARPOUR M R, ARDAKANI M. Soft actor-critic-based computation offloading in multi-user MEC-enabled IoT—A lifetime maximization perspective[J]. IEEE Internet of Things Journal, 2023, 10(20): 17571-17584.
- [19] QI Q Y, SHI T, QIN K, et al. Completion time optimization in UAV-relaying-assisted MEC networks with moving users[J]. IEEE Transactions on Consumer Electronics, 2023. DOI: 10.1109/TCE.2023.3278470.
- [20] LI X Y, YOU C S, ANDREEV S, et al. Wirelessly powered crowd sensing: Joint power transfer, sensing, compression, and transmission[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(2): 391-406.
- [21] LIN NA, TANG H L, ZHAO L, et al. A PDDQNLP algorithm for energy efficient computation offloading in UAV-assisted MEC[J]. IEEE Transactions on Wireless Communications, 2023, 22(12): 8876-8890.

## 作者简介:



李斌(1987-),通信作者,男,教授,研究方向:无人机通信、边缘计算,E-mail: bin.li@nuist.edu.cn.



朱潇(1998-),男,硕士研究生,研究方向:移动边缘计算。



王俊义(1977-),男,研究员,研究方向:移动边缘计算、网络资源管控、图信号处理。

(编辑:夏道家)