

基于计算重用的无人机辅助边缘计算系统能耗优化

李斌^{*①} 蔡海晨^① 赵传信^② 王俊义^③

^①(南京信息工程大学计算机学院 南京 210044)

^②(安徽师范大学计算机与信息学院 芜湖 241002)

^③(桂林电子科技大学信息与通信学院 桂林 541004)

摘要: 针对复杂地形下时延敏感任务对终端用户的计算需求激增问题, 该文提出一种无人机(UAV)辅助的移动边缘计算可重用任务的协同计算卸载方案。首先, 通过联合优化用户卸载策略、用户传输功率、无人机上服务器分配、用户设备的计算频率和无人机服务器的计算频率以及无人机的飞行轨迹, 构建满足时延约束下最小化系统平均总能耗的系统模型。其次, 通过深度强化学习求解该优化问题, 并提出了基于柔性动作-评价(SAC)的优化算法。该算法采用最大熵的策略来鼓励探索, 以增强算法的探索能力并加快训练的收敛速度。仿真结果表明, 基于SAC的算法能有效降低系统的平均总能耗, 并具有较好的收敛性。

关键词: 无人机; 移动边缘计算; 计算重用; 资源分配; 柔性动作-评价算法

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2024)06-0001-08

DOI: 10.11999/JEIT231061

Energy Optimization for Computing Reuse in Unmanned Aerial Vehicle-assisted Edge Computing Systems

LI Bin^① CAI Haichen^① ZHAO Chuanxin^② WANG Junyi^③

^①(School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China)

^②(School of Computer and Information, Anhui Normal University, Wuhu 241002, China)

^③(School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China)

Abstract: To address the high computational performance demands of delay-sensitive tasks in complex terrains, the collaborative computation offloading scheme for reusable tasks in mobile edge computing with the assistance of Unmanned Aerial Vehicle (UAV) is proposed. Firstly, the minimization of the average total energy consumption is formulated by jointly optimizing user offloading, user transmission power, server assignment on UAV, computation frequencies of users and UAV servers, as well as UAV flight trajectory, while meeting the latency constraints. Secondly, a deep reinforcement learning approach is employed to solve the optimization problem, and a Soft Actor-Critic (SAC) based optimization algorithm is introduced. The SAC algorithm utilizes a maximum entropy policy to encourage exploration that enhances the algorithm's exploration capabilities and accelerates the training convergence speed. Simulation results demonstrate that the proposed SAC algorithm effectively reduces the average total energy consumption of the system while exhibiting good convergence.

Key words: Unmanned Aerial Vehicle (UAV); Mobile edge computing; Reusable tasks; Resource allocation; Soft Actor-Critic (SAC) algorithm

收稿日期: 2023-10-07; 改回日期: 2024-01-19; 网络出版: 2024-02-05

*通信作者: 李斌 bin.li@nuist.edu.cn

基金项目: 国家自然科学基金(62101277, 62371149), 认知无线电与信息处理教育部重点实验室基金(CRKL230203)

Foundation Items: The National Natural Science Foundation of China (62101277, 62371149), The Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education (CRKL230203)

1 引言

移动边缘计算(Mobile Edge Computing, MEC)已经成为解决计算密集型业务的一种新范式[1,2]。当前MEC主要依赖于地面基础通信设施,通过部署固定式边缘服务器进行任务卸载。但在地质灾害等自然因素导致地面基础设施受到破坏甚至摧毁的情况下,现有的MEC技术将无法发挥其作用[3]。无人机(Unmanned Aerial Vehicle, UAV)因具有灵活机动、易于部署、成本低等优点[4],被广泛应用到无线通信网络中,可通过在UAV上放置服务器为用户提供MEC服务[5]。

业界对UAV辅助的MEC开展了一些不凡的研究工作。譬如,文献[6]通过联合优化计算资源、卸载决策、带宽分配和UAV轨迹,实现系统计算效率的最大化。文献[7]以最小化系统能耗为目标,对无人机部署和任务调度进行联合优化。但上述两种方案均假设UAV飞行在固定高度,只优化了UAV的2维飞行轨迹。文献[8]对UAV的轨迹调度和区域划分进行联合优化以实现UAV总能耗最小化,但在动态场景下该方案并不适用。文献[9]以最小化用户的平均能耗为目标,对用户的卸载策略和UAV飞行轨迹进行联合优化。该方案未考虑UAV的能耗。

工业物联网应用(如可重用的工业模块和游戏模块)可通过模块化的设计将其分解为一组独立的任务[10],任务的输入和结果与任务代码分离,每个独立的任务模块可以接受不同的参数并产生不同的执行结果。因此,这些任务可以在用户设备之间重复使用,被称为计算重用。尽管模块化的设计简化了应用,但执行可重用任务仍然需要大量的计算资源和能量,这对于资源有限的用户设备来说是一大挑战。最近,MEC可重用任务的相关研究备受关注。文献[11]研究了协同多人游戏组件的计算卸载问题,利用组件的可重用性最小化玩家的总能耗。文献[12]将可重用任务的协同卸载过程建模为联盟博弈,进而实现成本节约最大化。文献[13]研究了地面MEC系统中可重用任务的协作卸载问题,并提出了混合平均奖励近端策略优化算法。

对于执行可重用任务的设备来说,忽略用户设备之间的协作会导致任务数据的重复传输,进而增加额外的时延和能耗开销。此外,由于任务的可重用性、任务到达的不确定性、共享带宽和服务器等待队列的变化等问题,使得求解卸载问题存在一些困难。(1)任务的本地执行、数据传输和远程执行是并行事件,不合理的卸载策略可能会导致负载不均衡。(2)在共享通信环境中,分配给设备的带宽

随着参与卸载的设备数量的增加而减少,所以要鼓励用户设备协作传输。鉴于此,UAV辅助MEC的高动态环境下如何有效地对可重用任务的卸载问题进行求解是一大难题。

针对上述问题,本文的主要工作如下:(1)考虑到任务的可重用性,构建了一个最小化UAV辅助MEC系统平均总能耗的问题,涉及到用户卸载策略、用户的传输功率、UAV的服务器分配、用户设备的计算频率和UAV服务器的计算频率以及UAV飞行轨迹的联合优化。(2)为了求解该非凸优化问题,首先将用户的卸载计算过程建模为马尔可夫决策过程(Markov Decision Process, MDP),探索可重用任务在UAV辅助MEC系统中的卸载方案,采用柔性动作-评价(Soft Actor-Critic, SAC)方案求解上述优化变量,提出一种系统平均总能耗最小化的卸载决策算法。

2 系统模型

如图1所示,本文所考虑的UAV辅助MEC系统由 K 个用户 $\forall k \in \mathcal{K} \triangleq \{1, 2, \dots, K\}$,一个配备了 Q 个服务器 $\forall q \in \mathcal{Q} \triangleq \{1, 2, \dots, Q\}$ 的UAV以及 M 个可重用任务 $\forall m \in \mathcal{M} \triangleq \{1, 2, \dots, M\}$ 组成,任务 m 的数据量为 D_m 。每个用户在一段时间内处理一系列可重复使用的任务,因此假设系统在离散时隙运行,在每个时隙的开始,用户以一定概率从可重用任务集中接收一个任务,任务到达服从均匀分布。

假设每个时隙到达用户的任务都能在该时隙内完成。利用3维笛卡尔坐标可以表示用户和UAV的具体位置。 K 个用户随机分布在目标区域内,第 k 个用户的位置用坐标表示为 $\mathbf{p}_k = (x_k, y_k, 0)$,UAV在目标区域上空一定高度范围内飞行,在时隙 n ,UAV的位置用坐标表示为 $\mathbf{p}_{\text{uav}}[n] = (x_{\text{uav}}[n], y_{\text{uav}}[n], H_{\text{uav}}[n])$,用户 k 与UAV的距离为 $d_k[n] = \|\mathbf{p}_k - \mathbf{p}_{\text{uav}}[n]\|_2$ 。

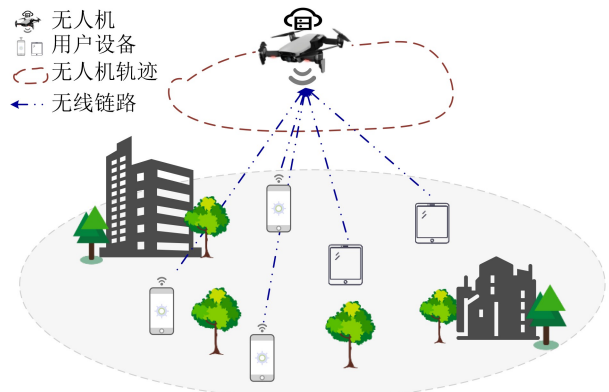


图1 系统模型图

2.1 通信模型

在UAV辅助MEC系统中，用户与UAV之间的通信链路主要是视距链路，因此信道被建模为自由空间路径损耗模型。在时隙 n ，用户 k 与UAV之间的信道增益可以建模为 $g_k[n] = \beta_0 d_k^{-2}[n]$ ，其中 $d_k[n] = \|\mathbf{p}_k - \mathbf{p}_{\text{uav}}[n]\|_2$ 为用户 k 与UAV之间的欧氏距离， β_0 为参考距离为1 m时的信道功率增益。UAV同时接收多个用户的计算任务数据，总无线带宽 B 被用户等分，用户设备间存在信号干扰。当共享无线链路的用户数量发生变化时，带宽被重新分配。用户 k 与UAV之间的上行链路传输速率计算为

$$u_k[n] = \frac{B}{\sum_{k \in \mathcal{K}} I(a_k[n] > 0)} \cdot \log_2 \left(1 + \frac{p_k^n[n] g_k[n]}{\sum_{i \in \mathcal{K} \setminus \{k\}} p_i^n[n] g_i[n] + \sigma^2} \right) \quad (1)$$

其中，函数 $I(\cdot)$ 为指示函数，当括号中条件为真时取值1，为假时取值0，此处用来判断用户是否需要共享无线链路卸载任务， $p_k^n[n]$ 为用户 k 在时隙 n 的传输功率， σ^2 为高斯白噪声功率。

2.2 计算模型

本文中的UAV辅助MEC系统采用部分卸载模式，定义 $a_k[n]$ 为用户 k 在时隙 n 卸载到UAV上计算的任务比例， $1 - a_k[n]$ 为在本地计算的任务比例。

2.2.1 本地执行部分

在时隙 n ，用户 k 本地计算任务 m 的时延为

$$t_{k,m}^l[n] = \frac{D_m(1 - a_k[n])C}{\sum_{i \in \mathcal{K}_m[n]} I(o_i[n] = m) f_i^l[n]} \quad (2)$$

其中， C 是任务每比特数据需要CPU运行的周期频率数， $\mathcal{K}_m[n]$ 是在时隙 n 内参与处理任务 m 的用户集合， $o_k[n]$ 是时隙 n 到达用户 k 的任务， $f_i^l[n]$ 是用户 i 在时隙 n 的计算频率。用户 k 本地计算任务 m 的能耗为

$$e_{k,m}^l[n] = \frac{\mu_k C D_m (1 - a_k[n]) (f_k^l[n])^3}{\sum_{i \in \mathcal{K}_m[n]} I(o_i[n] = m) f_i^l[n]} \quad (3)$$

其中， μ_k 为用户设备CPU的有效开关电容。

2.2.2 远程执行部分

在时隙 n ，用户 k 上传任务 m 的传输时延为

$$t_{k,m}^u[n] = D_m a_k[n] X_{k,m}[n] u_k^{-1}[n] \quad (4)$$

其中， $X_{k,m}[n] \in [0, 1]$ 表示在时隙 n 内用户 k 协作传输任务 m 的比例。用户 k 上传任务 m 的能耗为

$$e_{k,m}^u[n] = p_k^n[n] t_{k,m}^u[n] \quad (5)$$

在时隙 n ，任务 m 的卸载部分上传完成时间为 $t_m^u[n] = \max\{t_{k,m}^u[n] | k \in \mathcal{K}_m[n]\}$ 。在任务 m 的卸载部分全部到达UAV后，才会根据UAV上服务器的分配情况，选择服务器进行计算。因此，当参与协同传输的设备的传输完成时间相等时，用户的任务上传成本最小。在时隙 n ，任务 m 在UAV服务器 q 上远程执行的计算时延为

$$t_{m,q}^r[n] = D_m a_k[n] C (f_q^r[n])^{-1} \quad (6)$$

计算能耗为

$$e_{m,q}^r[n] = \mu_q C D_m a_k[n] (f_q^r[n])^2 \quad (7)$$

其中， μ_q 为UAV搭载的服务器上CPU的有效开关电容，主频为 $f_q^r[n]$ 。任务 m 上传完成且目标服务器空闲后，才能执行。卸载到UAV上远程执行产生的时延包括任务的传输时延和计算时延，因此在时隙 n 内，任务 m 的卸载部分在UAV服务器 q 上远程执行的总时延为

$$t_{m,q}^{\text{total}}[n] = \max\{t_m^u[n], t_q^w[n]\} + t_{m,q}^r[n] \quad (8)$$

当任务 m 决定在服务器 q 上执行后，服务器 q 的等待时间 $t_q^w[n]$ 立即更新为 $t_q^w[n] = \max\{t_m^u[n], t_q^w[n]\} + t_{m,q}^r[n]$ 。更新 $t_q^w[n]$ 的次数与时隙 n 内在服务器 q 上执行的任務数量相同。在时隙 n ，任务 m 执行的总时延为

$$T_m[n] = \max \left(\max_{k \in \mathcal{K}_m[n]} t_{k,m}^l[n], t_{m,q}^{\text{total}}[n] \right) \quad (9)$$

任务 m 执行的总能耗为

$$e_m[n] = \sum_{k \in \mathcal{K}_m[n]} e_{k,m}^l[n] + \sum_{k \in \mathcal{K}_m[n]} e_{k,m}^u[n] + e_{m,q}^r[n] \quad (10)$$

2.2.3 无人机飞行能耗

UAV在计算用户卸载部分任务的同时保持飞行状态，因此，还要考虑UAV的飞行能耗。在时隙 n ，UAV飞行产生的能耗为 $e_{\text{fly}}[n] = p_{\text{fly}}[n] \tau$ ，其中 τ 为一个时隙的长度， $p_{\text{fly}}[n]$ 为UAV的飞行功率，根据文献[14]表示为

$$p_{\text{fly}}[n] = p_{\text{rot}} \left(1 + \frac{3 \|\mathbf{v}[n]\|^2}{v_{\text{tip}}^2} \right) + p_{\text{hov}} \left(\sqrt{1 + \frac{\|\mathbf{v}[n]\|^4}{4v_{\text{hov}}^2}} - \frac{\|\mathbf{v}[n]\|^2}{2v_{\text{hov}}^2} \right)^{\frac{1}{2}} + \frac{1}{2} \varepsilon \delta \lambda A \|\mathbf{v}[n]\|^3 \quad (11)$$

其中， p_{rot} 为UAV叶片旋转功率， p_{hov} 为UAV悬停

功率, v_{tip} 为 UAV 叶片尖端速度, v_{hov} 为 UAV 悬停时平均转子速度, δ, A, ε 和 λ 分别表示空气密度、转子盘面积、机身阻力比和转子稳定度, $\mathbf{v}[n]$ 为 UAV 时隙 n 的飞行速度, 可以通过表达式 $\mathbf{v}[n] = (\mathbf{p}_{uav}[n+1] - \mathbf{p}_{uav}[n])\tau^{-1}$ 求出。因此, 每个时隙内, 系统的总能耗为

$$E[n] = \sum_{m \in \mathcal{M}} e_m[n] + e_{fly}[n] \quad (12)$$

3 优化问题描述

本文联合优化用户卸载策略、用户的传输功率、UAV 的服务器分配、用户设备的计算频率和 UAV 服务器的计算频率以及 UAV 的飞行轨迹以最小化系统的平均总能耗。问题表述为

$$\begin{aligned} & \min_{a_k[n], p_k^u[n], \beta_m[n], f_k^l[n], f_q^s[n], \mathbf{p}_{uav}[n]} \frac{1}{N} \sum_{n=1}^N E[n] \\ \text{s.t. C1 : } & \max_{m \in \mathcal{M}} T_m[n] \leq \tau \\ \text{C2 : } & a_k[n] \in [0, 1], \forall k \in \mathcal{K} \\ \text{C3 : } & p_k^{\min} \leq p_k^u[n] \leq p_k^{\max}, \forall k \in \mathcal{K} \\ \text{C4 : } & f_k^{\min} \leq f_k^l[n] \leq f_k^{\max}, \forall k \in \mathcal{K} \\ \text{C5 : } & \sum_{m \in \mathcal{M}} (e_{k,m}^l[n] + e_{k,m}^u[n]) \leq e_k^{\max}, \forall k \in \mathcal{K} \\ \text{C6 : } & \beta_m[n] \in \mathcal{Q}, \forall m \in \mathcal{M} \\ \text{C7 : } & f_q^{\min} \leq f_q^s[n] \leq f_q^{\max}, \forall q \in \mathcal{Q} \\ \text{C8 : } & \sum_{m \in \mathcal{M}} e_{m,q}^r[n] \leq e_q^{\max}, \forall q \in \mathcal{Q} \\ \text{C9 : } & \sum_{m \in \mathcal{M}} \sum_{q \in \mathcal{Q}} e_{m,q}^r[n] + e_{fly}[n] \leq e_{uav}^{\max} \\ \text{C10 : } & \mathbf{p}_{uav}[1] = \mathbf{p}_{uav}[N] \\ \text{C11 : } & x_{\min} \leq x_{uav}[n] \leq x_{\max} \\ \text{C12 : } & y_{\min} \leq y_{uav}[n] \leq y_{\max} \\ \text{C13 : } & H_{\min} \leq H_{uav}[n] \leq H_{\max} \\ \text{C14 : } & \|\mathbf{p}_{uav}[n] - \mathbf{p}_{uav}[n-1]\|_2 \leq V_{\max}\tau \end{aligned} \quad (13)$$

其中, p_k^{\min} 和 p_k^{\max} 分别表示用户设备 k 的最小和最大传输功率, f_k^{\min} 和 f_k^{\max} 分别表示用户设备 k 的最小和最大计算频率, f_q^{\min} 和 f_q^{\max} 分别表示 UAV 上服务器 q 的最小和最大计算频率。约束条件 C1 为任务的计算时延限制, 约束条件 C2 为时隙 n 用户的卸载比例, 约束条件 C3 为用户传输功率限制, 约束条件 C4 和 C7 为用户设备和 UAV 上服务器计算频率限制, 约束条件 C5 为用户设备的能耗限制, 约束条件 C6 为 UAV 服务器分配序号, 约束条件 C8 和 C9 为 UAV 上单个服务器和整个 UAV 的能耗限制, 约束条件 C10 为 UAV 飞行轨迹的起点和终点, 约束条件 C11,

C12 和 C13 为 UAV 的飞行区域限制, 约束条件 C14 为 UAV 的飞行速度限制。非线性的目标函数和非凸的约束条件使得问题成为一个多变量耦合的非凸优化问题, 该类问题的复杂程度过高, 利用传统凸优化技术难以直接求解。为求解此问题, 本文提出一种基于 SAC 的深度强化学习算法。

4 优化问题求解

由于动态卸载问题涉及到高维状态空间和动作空间, 传统的强化学习方法可能会面临维度灾难和收敛速度缓慢的问题。因此, 本文采用深度强化学习方法来解决此问题。

4.1 SAC 算法

SAC 是一种 off-policy 强化学习算法, 与传统的强化学习算法不同, SAC 算法不仅考虑到累积奖励的最大化, 还将策略的熵作为一个重要的目标, 该算法的目标是最大化累积奖励和策略熵的加权和, 鼓励智能体在选择动作时具有更多的随机性。这意味着尽可能随机地采取行动, 更有利于找到最佳的解决方案。

SAC 算法采用最大熵策略, 以最大化策略的熵来鼓励探索, 该策略同时最大化回报和熵的期望, 最优策略 π^* 可以表示为

$$\pi^* = \arg \max_{\pi} E_{(s^t, a^t) \sim \rho_{\pi}} \left[\sum_{i=t}^{\infty} \gamma^{i-t} (r^i(s^t, a^t) + \alpha H(\pi(\cdot|s^t))) \right] \quad (14)$$

其中, ρ_{π} 表示动作概率分布函数, γ 为折扣因子, $H(\pi(\cdot|s^t)) = E_{a^t}[-\log_2 \pi(a^t|s^t)]$ 是策略 π 的熵, α 为调整系统奖励和探索未知动作比重的温和系数。 $V^t = \sum_{i=t}^{\infty} \gamma^{i-t} (R^i + \alpha H(\pi(a^i|s^t)))$ 表示在状态 s^t 下熵增强的累积回报, 也称为软回报。卸载策略 π 的软 Q 函数定义为

$$Q(s^t, a^t) = r^t + \alpha E_{s^{t+1} \sim \rho_{\pi}} [V^{t+1}] \quad (15)$$

该公式表示当前卸载策略 π 在状态 s^t 下, 联合卸载动作 a^t 的期望软回报, 用于评估卸载策略的预期回报值。

最优的最大熵卸载策略通过软策略评估和更新迭代交替进行得到。在固定卸载策略的情况下, 通过迭代软贝尔曼算子 \mathcal{T}_{π} 计算软 Q 值。软卸载策略的训练目标是找到一个比当前卸载策略 π_{old} 更好的卸载策略 π_{new} 。为了确保更新后的策略 π_{new} 具有更高的软 Q 值, 通过最大化软 Q 函数的指数形式来更新策略。SAC 使用深度神经网络逼近软 Q 函数 $Q(s, a)$ 和卸载策略 $\pi(a|s)$, 将软 Q 函数和卸载策略

参数化为 $Q_\theta(s, a)$ 和 $\pi_\varphi(a|s)$ ，其中， θ 和 φ 是深度神经网络的参数。软 Q 函数的参数通过最小化软贝尔曼残差从而训练得到

$$L_Q(\theta) = E_{(s^t, a^t) \sim \mathcal{B}} \left[\frac{1}{2} (Q_\theta(s^t, a^t) - \hat{Q}(s^t, a^t))^2 \right] \quad (16)$$

其中， \mathcal{B} 表示用于存储历史经验样本的经验回放数组； $\hat{Q}(s^t, a^t)$ 是目标软 Q 函数。类似地，通过最小化期望KL散度，可以训练卸载策略的参数 φ 。具体而言，可以通过最小化式(17)来训练卸载策略的参数

$$L_\pi(\varphi) = E_{s^t \sim \mathcal{B}} [E_{a^t \sim \pi_\varphi} [\alpha \log_2 \pi_\varphi(a^t|s^t) - Q_\theta(s^t, a^t)]] \quad (17)$$

其中，对 s^t 的期望可以通过从经验回放数组 \mathcal{B} 中进行采样来近似计算。

在模型的训练过程中，系统奖励的变化会导致温和系数 α 的动态改变，从而使得模型的训练过程不稳定，故采用固定的温和系数是不合理的。因此，在训练过程中，更新网络参数的同时也要自动调整温和系数 α 的值。当动作空间尚未被充分探索，最优卸载动作还未确定时，适当提高温和系数有利于探索更多的空间。当动作空间已探索完成，最优卸载动作已经确定时，温和系数应该适当降低。本文通过最小化损失 $L(\alpha) = E_{s \sim \mathcal{B}} E_{a \sim \pi} [-\alpha \log_2 \pi_\varphi(a|s) - \alpha H_0]$ 来调整温和系数，其中， H_0 是目标熵数值。

4.2 基于SAC的卸载策略算法设计

本文考虑的模型被建模为马尔可夫决策过程，

MDP的3要素：状态空间、动作空间、奖励函数定义如下。基于SAC的卸载决策算法结构如图2所示。

(1) 状态空间： $S[n]$ 表示智能体在时隙 n 的状态， $S[n] = \{\text{task}_k[n], p_{\text{uav}}[n-1], d_{\text{uav}}[n-1]\}$ 。其中 $\text{task}_k[n]$ 是时隙 n 到达用户 k 的任务序号， $p_{\text{uav}}[n-1]$ 是UAV在时隙 $n-1$ 时的位置， $d_{\text{uav}}[n-1] = \{d_{\text{uav}}^1, d_{\text{uav}}^2, \dots, d_{\text{uav}}^K\}$ 是时隙 $n-1$ 时UAV与每个用户之间的距离向量。

(2) 动作空间：动作空间包括用户的传输功率、用户任务的卸载比例、UAV的服务器分配、用户设备的计算频率、UAV上服务器的计算频率和无人机的飞行位置，表示为 $A[n] = \{p_k^u[n], a_k[n], \beta_m[n], f_k^l[n], f_q^r[n], p_{\text{uav}}[n]\}$ ，其中 $p_k^u[n]$ 为用户 k 在时隙 n 的传输功率， $a_k[n]$ 为用户 k 在时隙 n 的卸载比例， $\beta_m[n]$ 为时隙 n 任务 m 分配到的UAV服务器， $f_k^l[n]$ 为用户 k 在时隙 n 的计算频率， $f_q^r[n]$ 为UAV上服务器 q 在时隙 n 的计算频率， $p_{\text{uav}}[n]$ 为UAV在时隙 n 的位置坐标。

(3) 奖励函数： $r[n] = -E[n] - P$ 。其中 $E[n] = \sum_{m \in \mathcal{M}} e_m[n] + e_{\text{fly}}[n]$ 为时隙 n 用户设备是到达的所有任务执行完成的计算能耗和UAV飞行能耗的总和。当约束条件C1不满足时，即任务执行超时，设置惩罚 punish_t 。当约束条件C5, C8和C9不满足时，分别为用户设备能耗超出预算、UAV上单个服务器能耗超出预算和UAV总能耗超出预算时，设置惩罚 punish_e 。当约束条件C14不满足时，即UAV飞行速度超过最大飞行速度限制时，设置

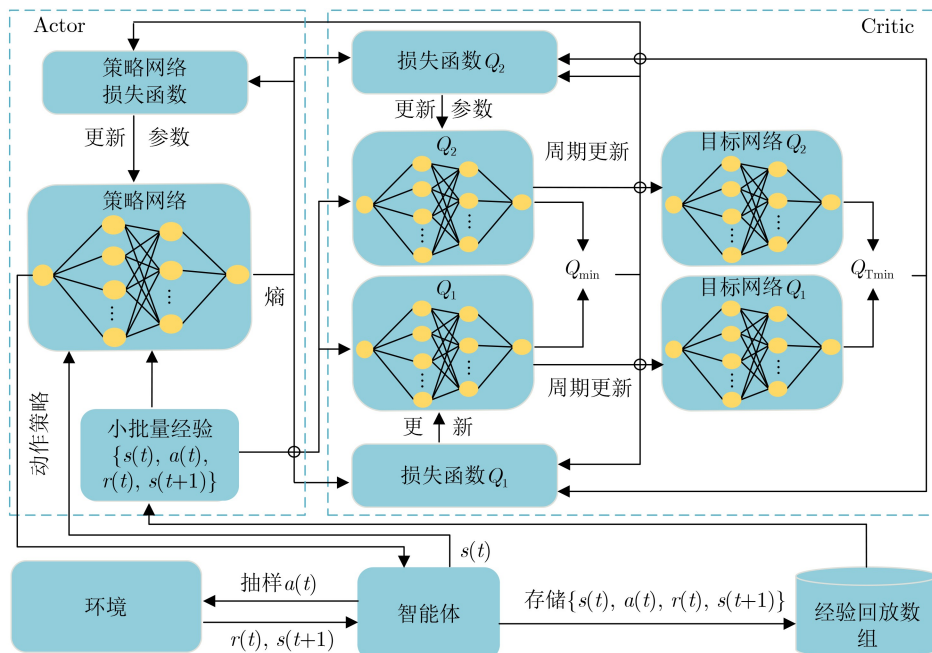


图2 基于SAC的卸载决策算法结构图

惩罚 punish_v 。因此, $P = \text{punish}_t + \text{punish}_e + \text{punish}_v$ 是任务执行超时、能耗超限和UAV飞行速度超限的惩罚值总和。

基于上述定义说明, 本文所提基于SAC的系统平均总能耗最小化算法具体流程如算法1所示。

在深度强化学习算法中, 参数更新通常涉及到大量的矩阵相乘操作, 故在分析时间复杂度时会考虑到算法迭代过程中乘法的计算次数。因此, 算法1的复杂度为 $O(\sum_l^L n_l n_{l-1})$, 其中 n_l 是隐藏层 l 的神经单元数量。本文所用的SAC算法中隐藏层的大小接近, 算法1的时间复杂度可以近似为 $O((l-1)n_l^2) \sim O(n_l^2)$ 。

5 仿真结果

本节利用Python 3.7和Pytorch框架进行仿真并验证所提出方案的性能。6个用户随机分布在边长20 m的正方形区域内, UAV在给定的服务范围内飞行, 并将初始位置和终点位置都设为(0,0,10)。任务的数据量 $D_m \in [8 \times 10^6, 8.5 \times 10^6]$ bit, 处理单位bit任务所需CPU周期数 C 为650 cycles/bit, 总带宽 B 为 6×10^7 Hz, 信道功率增益 β_0 为-30 dB, 高斯白噪声功率 σ^2 为-60 dBm, 用户设备最大传输功率 p_k^{\max} 和最小传输功率 p_k^{\min} 为30 dBm和28 dBm。用户设备的能耗预算 e_k^{\max} 为10 J, 最大计算频率 f_k^{\max} 和最小计算频率分别为0.7 GHz和0.6 GHz。UAV上服务器的能耗预算 e_q^{\max} 为30 J, 最大计算频率 f_q^{\max} 和最小计算频率 f_q^{\min} 分别为1.8 GHz和1.6 GHz。UAV上搭载服务器数量为6个。用户设备CPU电容系数 $\mu_k = 10^{-26}$, UAV上服务器CPU电容系数 $\mu_q = 10^{-27}$ 。UAV的最大飞行速度 V_{\max} 为2 m/s。本文参数设置参考文献[15], SAC的训练参数设置如表1所示, 无人机飞行功率公式中相关参数如表2所示。

算法1 基于SAC的系统平均总能耗最小化算法

输入: 最大回合数 E , 学习率 β , 折扣回报 γ , 时隙数 N 。
 步骤1 初始化经验数组, Actor网络, Critic网络及目标网络参数, 随机生成用户坐标以及计算任务信息;
 步骤2 for episode=1: E
 初始化无人机初始坐标以及初始状态 $s[0]$;
 for slot=1: N
 更新时隙 t 用户设备上到达任务的信息和带宽分配情况;
 根据当前策略 π_φ 和状态 $s[t]$ 选择动作 $a[t]$;
 根据奖励函数计算 $r[t]$, 观察下一个状态 $s[t+1]$, 并将 $\{s[t], a[t], r[t], s[t+1]\}$ 存储到经验回放数组;
 从经验回放数组中随机采样一组经验样本, 根据式(16)和式(17)分别计算损失函数 $L_Q(\theta), L_\pi(\varphi)$, 并更新Q网络参数 θ, V 网络参数 φ 和温因子 α ;
 每隔 Z 步更新目标网络参数 φ^- ;
 步骤3 输出用户卸载策略网络参数 φ 。

图3对比了不同方案的收敛性能。可以观察到, 随着训练的进行, 3种方案的reward均逐渐趋于稳定。本文所提算法在 1.5×10^4 步后趋于稳定并收敛, 而基于A2C的方案则在 2.5×10^4 步后趋于稳定并收敛。相较于A2C算法, SAC算法具有更快的收敛速度, 并且在收敛后可以取得更好的reward。这是因为SAC算法采用了熵正则化和双重价值函数的方法, 促进了探索, 避免了策略陷入局部最优解, 从而加快了算法的收敛速度。双重价值函数使得SAC算法能够更准确地估计reward, 并引导策略的更新, 从而在收敛后实现更好的reward。而随机卸载策略收敛速度最慢且收敛后取得的reward最差, 原因在于随机的卸载策略不能根据系统实时情况做出合理的卸载决策。

图4显示了在不同用户数下系统平均总能耗的对比。由图4可知, 在不同用户数量的场景下基于SAC方案的计算总能耗低于其他对比方案, 基于A2C的方案次之, 随机卸载方案的计算总能耗最高, 性能表现最差。上述结果证明了联合优化UAV飞行轨迹和用户卸载方案在最小化系统平均总能耗方面的

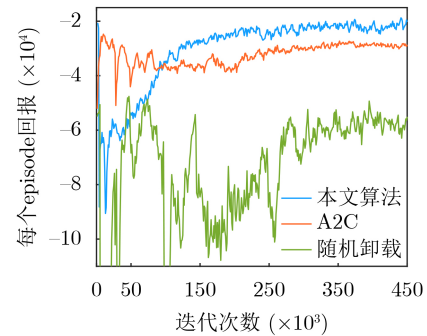


图3 不同方案性能对比图

表1 SAC训练参数

参数	值	参数	值
隐藏层数量 L	3	惩罚值 C	8
折扣回报 γ	0.99	目标网络更新频率 Z	320
最大回合数 E	10^3	温因子 α 初始值	0.005
学习率 β	10^{-4}	贪婪策略比例	0.2
批次经验大小	64	经验回放数组大小	10^6

表2 无人机飞行功率相关参数

参数	值	参数	值
UAV叶片旋转功率 p_{rot}	59.03 W	空气密度 δ	1.225 kg/m ³
UAV悬停功率 p_{hov}	79.07 W	转子盘面积 A	0.5030 m ²
UAV叶片尖端速度 v_{tip}	120 m/s	机身阻力比 ε	0.6
UAV悬停时平均转子速度 v_{hov}	3.6 m/s	转子稳定度 λ	0.05

重要性和必要性，且SAC算法的性能表现要优于A2C算法。在用户数较多的情况下，本文所提的方案仍能取得比较好的效果。

在图5中，用户数目设置为8个，所有方案的系统平均总能耗随着用户任务数据量的增加而增大。相比之下，基于SAC方案的计算总能耗最低，随机卸载方案的计算总能耗最高，且方案之间的能耗差距比较大，这说明利用SAC算法可以较好地优化UAV飞行轨迹和任务卸载决策等，使得用户任务能够以更高效的方式完成计算，从而减少了系统平均总能耗。

图6为基于SAC方案的UAV 3维飞行轨迹图，UAV从起始位置出发，沿着一条相对直接的路径

飞到用户中心位置，并逐渐向用户较为密集的区域靠近，并在用户较为密集的区域上空来回盘旋，为用户提供MEC服务。基于SAC方案的UAV最优3维飞行轨迹较为平顺，这有助于提高飞行的稳定性并降低飞行能耗。并且UAV在每个时隙内与需要提供计算服务的地面用户保持相对合理的距离，在用户较为密集的区域，调整UAV的飞行高度可以一定程度上减弱用户之间的信号干扰，从而提高用户的远程卸载效率。通过选择合适的飞行高度，UAV可以更好地满足用户的计算需求。

6 结束语

本文研究了UAV辅助的可重用任务MEC系统中的计算任务卸载方案。通过联合优化用户卸载策略、用户传输功率、UAV上服务器分配、计算频率以及UAV的飞行轨迹以最小化系统的平均总能耗，并提出了一种基于SAC的可重用任务卸载决策算法。仿真结果表明，所提算法可有效求解可重用任务的协同计算卸载问题。在未来工作中，考虑用户移动场景下多UAV协作的MEC可重用任务方案。

参考文献

- [1] LIU Yaqiong, PENG Mugen, SHOU Guochu, *et al.* Toward edge intelligence: Multiaccess edge computing for 5G and internet of things[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 6722–6747. doi: [10.1109/JIOT.2020.3004500](https://doi.org/10.1109/JIOT.2020.3004500).
- [2] 贾淑霞, 郝万明, 高梓涵, 等. 任务迁移的移动边缘计算系统中资源分配和任务卸载策略[J]. *电讯技术*, 2022, 62(8): 1037–1043. doi: [10.3969/j.issn.1001-893x.2022.08.003](https://doi.org/10.3969/j.issn.1001-893x.2022.08.003).
JIA Shuxia, HAO Wanming, GAO Zihan, *et al.* Resource allocation and task offloading strategy in mobile edge computing system for task migration[J]. *Telecommunication Engineering*, 2022, 62(8): 1037–1043. doi: [10.3969/j.issn.1001-893x.2022.08.003](https://doi.org/10.3969/j.issn.1001-893x.2022.08.003).
- [3] TUN Y K, DANG T N, KIM K, *et al.* Collaboration in the sky: A distributed framework for task offloading and resource allocation in multi-access edge computing[J]. *IEEE Internet of Things Journal*, 2022, 9(23): 24221–24235. doi: [10.1109/JIOT.2022.3189000](https://doi.org/10.1109/JIOT.2022.3189000).
- [4] SUN Geng, LI Jiahui, LIU Yanheng, *et al.* Time and energy minimization communications based on collaborative beamforming for UAV networks: A multi-objective optimization method[J]. *IEEE Journal on Selected Areas in Communications*, 2021, 39(11): 3555–3572. doi: [10.1109/JSAC.2021.3088720](https://doi.org/10.1109/JSAC.2021.3088720).
- [5] HSU Y H and GAU R H. Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks[J]. *IEEE Transactions on Mobile Computing*, 2022, 21(1): 306–320. doi: [10.1109/TMC.2020](https://doi.org/10.1109/TMC.2020).

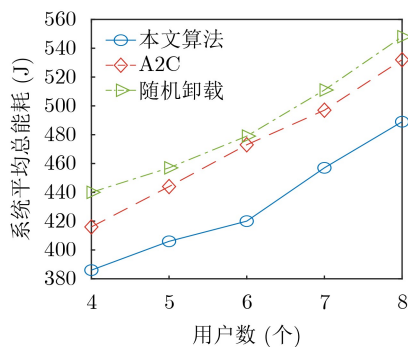


图4 用户数与系统平均总能耗关系图

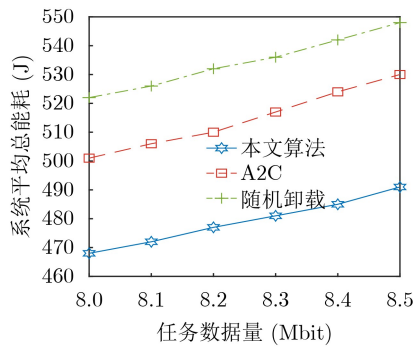


图5 任务数据量与系统平均总能耗关系图

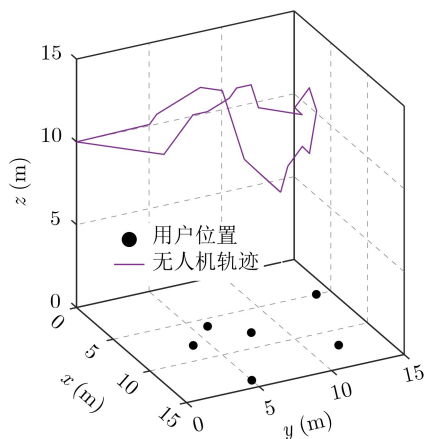


图6 无人机3维飞行轨迹图

- 3003639.
- [6] WANG Longjie, ZHOU Qiang, and SHEN Yan. Computation efficiency maximization for UAV-assisted relaying and MEC networks in urban environment[J]. *IEEE Transactions on Green Communications and Networking*, 2023, 7(2): 565–578. doi: [10.1109/TGCN.2022.3222398](https://doi.org/10.1109/TGCN.2022.3222398).
- [7] WANG Yong, RU Zhiyang, WANG Kezhi, *et al.* Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing[J]. *IEEE Transactions on Cybernetics*, 2020, 50(9): 3984–3997. doi: [10.1109/TCYB.2019.2935466](https://doi.org/10.1109/TCYB.2019.2935466).
- [8] WANG Di, TIAN Jie, ZHANG Haixia, *et al.* Task offloading and trajectory scheduling for UAV-enabled MEC networks: An optimal transport theory perspective[J]. *IEEE Wireless Communications Letters*, 2022, 11(1): 150–154. doi: [10.1109/LWC.2021.3122957](https://doi.org/10.1109/LWC.2021.3122957).
- [9] 张广驰, 何梓楠, 崔苗. 基于深度强化学习的无人机辅助移动边缘计算系统能耗优化[J]. *电子与信息学报*, 2023, 45(5): 1635–1643. doi: [10.11999/JEIT220352](https://doi.org/10.11999/JEIT220352).
ZHANG Guangchi, HE Zinan, and CUI Miao. Energy consumption optimization of unmanned aerial vehicle assisted mobile edge computing systems based on deep reinforcement learning[J]. *Journal of Electronics & Information Technology*, 2023, 45(5): 1635–1643. doi: [10.11999/JEIT220352](https://doi.org/10.11999/JEIT220352).
- [10] WESTERA W, PRADA R, MASCARENHAS S, *et al.* Artificial intelligence moving serious gaming: Presenting reusable game AI components[J]. *Education and Information Technologies*, 2020, 25(1): 351–380. doi: [10.1007/s10639-019-09968-2](https://doi.org/10.1007/s10639-019-09968-2).
- [11] YANG Xuemei, LUO Hong, SUN Yan, *et al.* Energy-efficient collaborative offloading for multiplayer games with cache-aided MEC[C]. The IEEE International Conference on Communications, Dublin, Ireland, 2020: 1–7. doi: [10.1109/ICC40277.2020.9148751](https://doi.org/10.1109/ICC40277.2020.9148751).
- [12] YANG Xuemei, LUO Hong, SUN Yan, *et al.* Coalitional game-based cooperative computation offloading in MEC for reusable tasks[J]. *IEEE Internet of Things Journal*, 2021, 8(16): 12968–12982. doi: [10.1109/JIOT.2021.3064186](https://doi.org/10.1109/JIOT.2021.3064186).
- [13] YANG Xuemei, LUO Hong, SUN Yan, *et al.* A novel hybrid-ARPPPO algorithm for dynamic computation offloading in edge computing[J]. *IEEE Internet of Things Journal*, 2022, 9(23): 24065–24078. doi: [10.1109/JIOT.2022.3188928](https://doi.org/10.1109/JIOT.2022.3188928).
- [14] LIU Boyang, WAN Yiyao, ZHOU Fuhui, *et al.* Resource allocation and trajectory design for MISO UAV-assisted MEC networks[J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(5): 4933–4948. doi: [10.1109/TVT.2022.3140833](https://doi.org/10.1109/TVT.2022.3140833).
- [15] CHEN Peipei, LUO Xueshan, GUO Deke, *et al.* Secure task offloading for MEC-aided-UAV system[J]. *IEEE Transactions on Intelligent Vehicles*, 2023, 8(5): 3444–3457. doi: [10.1109/TIV.2022.3227367](https://doi.org/10.1109/TIV.2022.3227367).
- 李 斌: 男, 副教授, 硕士生导师, 研究方向为移动边缘计算、无人机通信网络.
- 蔡海晨: 男, 硕士生, 研究方向为移动边缘计算.
- 赵传信: 男, 教授, 博士生导师, 研究方向为物联网、智能信息处理.
- 王俊义: 男, 研究员, 博士生导师, 研究方向为边缘计算、无线资源分配.

责任编辑: 余 蓉