

深度强化学习驱动的电力物联网任务卸载优化

武鹏荣¹ 李振文¹ 雷 强¹

WU Pengrong LI Zhenwen LEI Qiang

摘 要

在能源数字化转型背景下, 电力物联网 (power internet of things, PIoT) 面临数据处理需求暴增与偏远区域异构网络覆盖的挑战。为此, 文章提出一种基于无人机协同的天空地一体化 (space-air-ground, SAG) PIoT 架构, 通过联合优化任务卸载、无人机轨迹与资源管理, 将其建模为时变网络状态驱动的长期混合整数非线性规划问题, 并解耦为任务卸载-轨迹规划、计算资源管理两个子问题。针对前者, 提出融合异步训练、联邦学习等技术的异步联邦多智能体近端策略优化 (asynchronous federated PPO, AFMAPPO) 算法, 使系统总延迟降低 65.4%; 对后者设计了基于贪心算法的资源管理策略, 将任务完成率提升至 89.5%。实验结果相较于 MAPPO、PPO、GA 等传统基准方法, 所提方法在任务积压、队列延迟和任务完成率等方面表现更佳, 为动态时变 SAG-PIoT 系统提供可行方案。

关键词

天空地一体化; 电力物联网; 任务卸载; 异步联邦多智能体学习; 资源管理

doi: 10.3969/j.issn.1672-9528.2025.12.033

0 引言

随着数字化转型进程加速, 前沿的数据采集与传感技术持续升级, 电力物联网^[1] (power internet of things, PIoT) 借助先进的通信技术可实现发-输-用全环节的实时状态监控与能效优化, 在保障电力可靠供应、提高系统能效等方面发挥着重要作用^[2]。然而, 在高比例可再生能源并网背景下, 设备产生大量数据需实时传输和处理, 这对现有网络资源和系统安全性提出严峻挑战^[3]。此外, 用于监测可再生能源系统的 PIoT 设备部署于偏远地区, 其地面通信基础设施薄弱、数据传输速率低, 极大阻碍系统的高效运行。

在任务卸载领域中, 传统解决方法主要包括基于数学规划与凸优化的精确求解或匹配理论建模^[4]; 然而这些方法往往依赖全局精确模型, 计算开销大、难以满足毫秒级实时响应, 很难自适应网络波动与设备异构。相比之下, 深度强化学习 (deep reinforcement learning, DRL) 通过 end-to-end 的 model-free 学习, 可在高维状态下实现毫秒级在线决策, 并借助持续交互自动适应环境变化, 同时在奖励函数中联合优化时延、能耗与可靠性, 显著提升系统的实时性、鲁棒性。

针对上述挑战, 本文提出一种融合卫星广域覆盖、无人机 (unmanned aerial vehicle, UAV) 灵活中继与地面网络协同的“天-空-地”三维异构网络体系——SAG-PIoT (space-air-ground PIoT)。通过利用高通量、低时延的低

轨卫星实现核心站与边缘节点的数据交互; UAV 在地面网络出现盲区时, 可快速部署中继节点以恢复通信, 并可根据负载动态调整 UAV 部署策略; 地面终端承载日常状态监测与本地预处理功能。三者协同发挥不同层面的资源优势, 形成全方位、立体式的通信网络。

本文的主要贡献如下:

(1) 系统框架设计: 针对“云-边-端”任务卸载场景, 构建了无人机辅助的三级协同 SAG-PIoT 架构, 并在长期队列约束下建立了以时延、能耗及队列稳定为目标的系统成本最小化问题。

(2) 算法设计优化: 将问题转化为长期混合整数非线性规划, 利用李雅普诺夫优化解耦。进而, 提出基于异步联邦的多智能体优化算法 (AFMAPPO) 进行任务决策, 并设计队列感知的贪心算法实现动态资源管理。

(3) 实验验证: 仿真实验表明, 本文方案在任务积压、队列延迟等方面性能出色, 验证了所提算法的性能增益。

1 相关工作

面对海量 PIoT 任务, 移动云计算 (mobile cloud computing, MCC) 延迟大、效率低; 而移动边缘计算 (mobile edge computing, MEC) 将资源迁移到边缘, 缓解了设备计算能力不足。现有研究主要通过云边协同与智能算法提升卸载计算效能。架构设计方面, 文献 [5] 构建 MEC 云边协作架构, 采用改进遗传算法动态调整卸载策略, 以降低时延。文献 [6] 则将服务缓存与计算卸载联合建模,

利用博弈论优化边缘节点策略,提高资源效率。文献[7]提出边缘缓存与计算资源迁移机制,通过实时负载预测触发资源重分配,实现快速恢复与性能平稳并提升了任务处理效率。然而,这些研究多假设静态网络与充分覆盖,未充分考虑偏远区域边缘节点部署不足的核心瓶颈,导致在覆盖盲区场景下系统可靠性下降。算法层面,文献[8]设计了基于改进粒子群优化算法的任务分配机制,优化多边缘节点间的协作效率。文献[9]在智能电网需求侧通信中通过频谱与计算资源联合优化降低系统成本,但其未考虑任务到达速率的时变性。文献[10]提出能量感知任务卸载方案以平衡能耗与时延,但未考虑异构节点的影响。

近年来,研究者利用低路径损耗、高视距概率的 UAV 空中服务器来扩展地面网络。文献[11]在无线能量传输场景中优化 UAV 轨迹以最大化地面设备能量获取效率。文献[12]进一步融合 DRL 与优先经验回放技术,在 UAV-MEC 网络中实现能效-时延联合优化,但其未考虑跨层资源异构性对算法收敛性的影响。文献[13]通过大时间尺度服务器选择与小时间尺度信道分配提升能效。尽管现有研究取得一定成果,但存在无法区分实时轨迹调整与长期资源规划的需求与计算资源异构性的耦合效应等问题。

2 系统模型

本文构建的系统模型由 PlIoT 设备、地面基站 (GBS)、UAV、MEC、云服务器和 LEO 卫星组成,包含 I 个 PlIoT 设备, $J+K$ 个基站 (J 个 UAV 和 K 个 GBS),如图 1 所示。

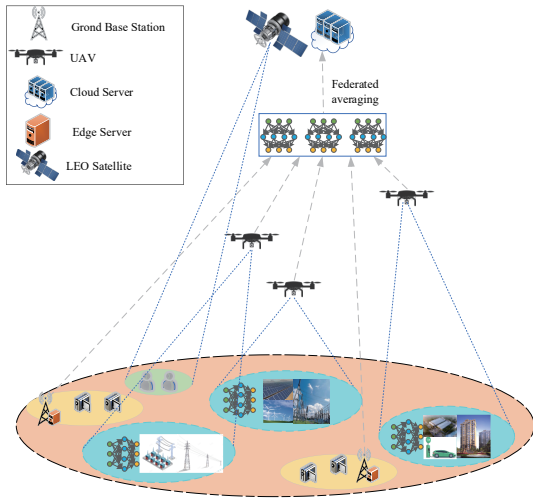


图 1 系统模型

设备和基站集合分别为 $U = \{u_1, u_2, \dots, u_I\}$ 和 $S = \{s_1, s_2, \dots, s_{J+K}\}$ 。设备生成大量任务,但受资源限制无法全部本地处理。UAV 搭载 MEC 可根据关联设备的分布进行优化配置。若任务在 UAV 通信范围内,其可卸载给 UAV 进行处理。配备云服务器的 LEO 卫星确保完整覆盖系统设备。

2.1 任务计算模型

设备任务由系统随机生成,用二元组 $u = (f_u, \text{task}_u')$ 表示, f_u 代表任务设备 u_i 的本地 CPU 频率;其中 $\text{task}_u' = (\text{id}_i, C_i, D_i, \tau_i)$, id_i 为任务的 id; C_i 为计算一位数据所需要的 CPU 周期; D_i 为任务的数据量大小; τ_i 为任务的截止时间。

系统时隙集定义为 $T = \{1, 2, \dots, t, \dots, T\}$, 时隙长度为 τ , 设备 u_i 在时隙 t 的位置为 $b_i(t) = \{x_i^u(t), y_i^u(t)\}$ 。UAV 以恒定速度 v 在高度 H 飞行,且功耗恒定。UAV s_j ($j \in 1, 2, \dots, J$) 在时隙 t 的位置为 $g_j(t) = \{x_j^s(t), y_j^s(t)\}$, 飞行方向为 $d_j(t)$, 满足 $|d_j(t)|=1$ 。该模型可视为准静态过程,在系统环境中 $g_j(t)$ 、 $d_j(t)$ 在时隙内保持稳定,在多个时隙间变化。 s_j 的位置与飞行方向关系为:

$$g_j(t+1) - g_j(t) = v\tau d_j(t) \quad (1)$$

s_j 对于 u_i 的可用性定义为 $a_{i,j}(t) \in \{0, 1\}$ 。其中 $a_{i,j}(t)=1$ 为时隙 t 时 s_j 可用于 u_i 。卸载决策由两个要素组成:①计算方式选择;② BS 选择。 u_i 决策为 $x(t) = \{x_{i,j}^o(t), x_i^c(t), u_i \in U, s_j \in S\}$ 。其中 $x_{i,j}^o(t)=1$ 为 u_i 的任务选择卸载计算, $x_{i,j}^o(t)=0$ 为任务选择本地计算; $x_i^c(t)=1$ 为 u_i 的任务选择云计算。

2.1.1 本地计算

本地计算模型的处理总时延为计算时延,总能耗为计算能耗。任务本地处理时,其时延与能耗为:

$$T_l = C_i D_i / f_u \quad (2)$$

$$E_l = P_{UE} T_l \quad (3)$$

式中: f_u 为设备计算频率; P_{UE} 为设备功率。

2.1.2 卸载计算

由于设备生成的任务数量大,当本地资源不足时,可将任务卸载到服务器处理。

(1) 任务卸载至 MEC

①设备-GBS 模型

若任务卸载至 GBS,则 $a_{i,j}(t)=0$ 。根据香农定理,任务在设备与 GBS 之间的传输速率为:

$$R_{i,j} = x_{i,j}^o(t) B_{i,j}(t) \log_2 \left(1 + \frac{P_i h_{i,j}(t)}{I_{i,j}^e(t) + \sigma^2} \right) \quad (4)$$

式中: $h_{i,j}(t)$ 、 $B_{i,j}(t)$ 、 P_i 、 $I_{i,j}^e(t)$ 分别为 u_i 和 s_j 间的信道增益、带宽、发射功率和电磁干扰功率; σ^2 为噪声功率。

②设备-UAV 模型

若任务卸载至 UAV,则 $a_{i,j}(t)=1$ 。鉴于 UAV 的移动性,考虑视距 (LoS) 与非视距 (NLoS) 损耗作用于数据传输,构建相应的任务卸载通信模型。数据传输中路径损耗为:

$$L_{i,j,d} = 20 \log_{10} \left(4\pi f_c \sqrt{H^2 + \|g_j(t) - b_i(t)\|^2} / c \right) + p_{i,j,d}^{\text{LoS}} \eta_{i,j,d}^{\text{LoS}} + (1 - p_{i,j,d}^{\text{LoS}}) \eta_{i,j,d}^{\text{NLoS}} \quad (5)$$

式中: $\|g_j(t) - b_i(t)\| = \sqrt{(x_j^s(t) - x_i^u(t))^2 + (y_j^s(t) - y_i^u(t))^2}$ 为欧氏范数; f_c 和 c 分别为载流子频率和光速; $\eta_{i,j,d}^{\text{LoS}}$ 和 $\eta_{i,j,d}^{\text{NLoS}}$ 分别为传播损耗对 LoS

和 NLoS 链路的附加损耗; $p_{i,j,t}^{\text{LoS}}$ 为设备与 LoS 连接的概率。

$$p_{i,j,t}^{\text{LoS}} = \frac{1}{1 + \omega_1 \exp\left\{-\omega_2 \left[\tan^{-1}\left(\frac{H}{\|g_j(t) - b_i(t)\|}\right) - \omega_1\right]\right\}} \quad (6)$$

式中: $(\omega_1, \omega_2, \eta_{i,j,t}^{\text{LoS}}, \eta_{i,j,t}^{\text{NLoS}})$ 由环境决定。 u_i 到 s_j 的传输速率为:

$$R_{i,j} = x_{i,j}^O(t) a_{i,j}(t) B_{i,j}(t) \log_2 \left(1 + \frac{P_i 10^{-L_{i,j,t}/10}}{I_{i,j}^e(t) + \sigma^2} \right) \quad (7)$$

任务卸载至 MEC 的总时延和总能耗为:

$$T_o = T_{\text{up}}^O + T_m = \frac{C_i D_i}{R_{i,j}(t)} + \frac{C_i D_i}{f_j} \quad (8)$$

$$E_o = P_{\text{UE}} T_{\text{up}}^O + P_j T_m \quad (9)$$

式中: f_j 为 MEC 频率; P_j 为 MEC 功率。

(2) 任务卸载至云服务器

任务选择卸载至云服务器时, 其传输速率为:

$$R_{\text{up}}^C(t) = x_{i,j}^O(t) x_i^C(t) W \log_2 \left(1 + \frac{P_i H_C}{I_{i,c}^e(t) + \sigma^2} \right) \quad (10)$$

式中: W 、 H_C 、 $I_{i,c}^e(t)$ 为设备与云服务器间的信道带宽、增益、电磁干扰。其总时延和总能耗为:

$$T_c = T_{\text{up}}^C + T_C = \frac{C_i D_i}{R_{\text{up}}^C(t)} + \frac{C_i D_i}{f_c} \quad (11)$$

$$E_c = P_{\text{UE}} T_{\text{up}}^C + P_C T_C \quad (12)$$

式中: f_c 、 P_C 为云服务器计算频率和计算功率。

2.2 任务队列模型

若设备 u_i 在时隙 t 到达的任务数为 $A_i(t)$, 且存在用于存储卸载任务的缓冲队列 $Q_i(t)$ 。

$$Q_i(t+1) = \{Q_i(t) - W_i(t)\}^+ + A_i(t) \quad (13)$$

式中: $\{e\}^+$ 为指标函数; $W_i(t)$ 为离开队列的任务量。

$$W_i(t) = \sum_{i=1}^I \min\{Q_i(t), \tau(x_{i,j}^O(t) R_{i,j} + x_{i,j}^C(t) R_{\text{up}}^C(t))\} \quad (14)$$

对服务器存储来自卸载任务的缓冲队列, 设服务器存储卸载任务的队列积压为 $Y_{i,j}^G(t)$ 、 $Y_{i,j}^O(t)$ 、 $Y_i^C(t)$ 。

$$\begin{cases} Y_{i,j}^G(t+1) = \{Y_{i,j}^G(t) - W_{i,j}^G(t)\}^+ + x_{i,j}^O(t) W_i(t) \\ Y_{i,j}^O(t+1) = \{Y_{i,j}^O(t) - W_{i,j}^O(t)\}^+ + x_{i,j}^O(t) a_{i,j}(t) W_i(t) \\ Y_i^C(t+1) = \{Y_i^C(t) - W_i^C(t)\}^+ + x_{i,j}^C(t) W_i(t) \end{cases} \quad (15)$$

式中: $W_{i,j}^G(t)$ 、 $W_{i,j}^O(t)$ 、 $W_i^C(t)$ 分别为 t 时隙设备 u_i 卸载并由 GBS、UAV、云服务器计算的任务数。

由于长期队列延迟在总延迟中占据主要地位, 其对卸载任务能否有效执行起到决定性作用。根据排队论中利特尔定律可知, 队列延迟与队列积压成正比, 与平均任务到达率成反比。将任务卸载的队列延迟记为 τ_i^O , 且存在约束条件:

$$\tau_i^O = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \frac{Q_i(t)}{A_i(t)} \leq \tau_{i,\max}^O \quad (16)$$

$$\tilde{A}_i(t) = \frac{1}{t} \sum_{y=0}^{t-1} A_i(y) \quad (17)$$

式中: $\tau_{i,\max}^O$ 为时间上界; $\tilde{A}_i(t)$ 为队列 $Q_i(t)$ 的平均任务到达率。

服务器侧队列延迟为 $\tau_{i,j}^{Y,G}$ 、 $\tau_{i,j}^{Y,O}$ 、 $\tau_i^{Y,C}$, 其中 $\tau_{i,j}^{Y,G,\max}$ 、 $\tau_{i,j}^{Y,O,\max}$ 、 $\tau_{i,\max}^{Y,C}$ 分别为 MEC、MCC 处理队列延迟上界, $\tilde{W}_{i,j}^G(t)$ 、 $\tilde{W}_{i,j}^O(t)$ 、 $\tilde{W}_i^C(t)$ 为各自队列的平均任务到达率, 有约束条件为:

$$\begin{cases} \tau_{i,j}^{Y,G} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \frac{Y_{i,j}^G(t)}{\tilde{W}_{i,j}^G(t)} \leq \tau_{i,j,\max}^{Y,G} \\ \tau_{i,j}^{Y,O} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \frac{Y_{i,j}^O(t)}{\tilde{W}_{i,j}^O(t)} \leq \tau_{i,j,\max}^{Y,O} \\ \tau_i^{Y,C} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \frac{Y_i^C(t)}{\tilde{W}_i^C(t)} \leq \tau_{i,\max}^{Y,C} \end{cases} \quad (18)$$

3 多目标联合优化

基于第二章构建的计算模型和队列模型, 本章提出面向时延-能耗-队列多重约束的联合优化框架, 通过动态协调卸载决策与异构资源分配, 获得最优策略。

3.1 问题公式化

本文的目标是在系统队列稳定、时延能耗约束下, 最小化长期成本。由此得到目标函数 P1:

$$\text{P1: } \min_Q \mathbb{C} = \Delta L(Q) + V(\omega_1 T_{\text{total}} + \omega_2 E_{\text{total}}) \quad (19)$$

$$\begin{cases} C_1: x_{i,j}^O(t) \in \{0,1\}, x_i^C(t) \in \{0,1\}, \forall u_i \in U, \forall s_j \in S \\ C_2: \sum_{u_i \in U} x_{i,j}^O(t) = 1, \forall u_i \in U, \forall s_j \in S \\ C_3: T_{\text{total}} = T_i + T_o + T_c, E_{\text{total}} = E_i + E_o + E_c \\ C_4: \sum_{i=1}^I f_{i,j}(t) \leq f_{j,\max}, \forall s_j \in S \\ C_5: \sum_{i=1}^I x_i^C(t) f_{i,c}(t) \leq f_{c,\max} \\ \text{s.t. } C_6: \text{公式 (1)} \\ C_7: \|g_j(t) - g_m(t)\| \geq \varepsilon, \forall s_j, s_m \in S, j \neq m \\ C_8: \text{公式 (13)、(15)} \\ C_9: 0 \leq \omega_1, \omega_2 \leq 1, \omega_1 + \omega_2 = 1 \end{cases} \quad (20)$$

式中: C_1 、 C_2 为卸载指标; C_3 为总时延和总能耗; C_4 、 C_5 为计算资源约束; C_6 为 UAV 位置与飞行方向约束; C_7 为 UAV 安全约束; C_8 为长期队列延迟约束; C_9 为权重因子约束。

由于长期排队延迟约束与短期决策耦合, 本文采用虚拟队列法将约束 C_8 转化为队列稳定性约束。针对 4 个队列延迟约束提出 4 个虚拟队列 $Z_i^O(t)$ 、 $Z_{i,j}^{Y,G}(t)$ 、 $Z_{i,j}^{Y,O}(t)$ 、 $Z_i^{Y,C}(t)$ 。

$$\begin{cases} Z_i^O(t+1) = \max\left\{Z_i^O(t) + \frac{Q_i(t)}{\tilde{A}_i(t)} - \tau_{i,\max}^O, 0\right\} \\ Z_{i,j}^{Y,G}(t+1) = \max\left\{Z_{i,j}^{Y,G}(t) + \frac{Y_{i,j}^G(t)}{\tilde{W}_{i,j}^G(t)} - \tau_{i,j,\max}^{Y,G}, 0\right\} \\ Z_{i,j}^{Y,O}(t+1) = \max\left\{Z_{i,j}^{Y,O}(t) + \frac{Y_{i,j}^O(t)}{\tilde{W}_{i,j}^O(t)} - \tau_{i,j,\max}^{Y,O}, 0\right\} \\ Z_i^{Y,C}(t+1) = \max\left\{Z_i^{Y,C}(t) + \frac{Y_i^C(t)}{\tilde{W}_i^C(t)} - \tau_{i,\max}^{Y,C}, 0\right\} \end{cases} \quad (21)$$

若虚拟队列平均速率稳定, 则队列延迟公式 C_8 自动满足。

因此, P1 可转换成 P2:

$$P2: \min_Q \mathbb{C} = \Delta L(Q) + V(\omega_1 T_{\text{total}} + \omega_2 E_{\text{total}}) \quad (22)$$

$$\text{s.t.} \begin{cases} C_1 \sim C_7, C_9 \\ C_8: \text{公式(21)} \end{cases} \quad (23)$$

3.2 Lyapunov 优化

$\Theta(t) = [Q(t), Z_i^O(t), Y_{i,j}^G(t), Z_{i,j}^{Y,G}(t), Y_{i,j}^O(t), Z_{i,j}^{Y,O}(t), Y_i^C(t), Z_i^{Y,C}(t)]$ 定义为队列积压和虚拟队列的连接向量。Lyapunov 函数定义为:

$$L(\Theta(t)) = \frac{1}{2} \left\{ \sum_{i,j \in I, K} Q(t)^2 + Z_i^O(t)^2 + Y_{i,j}^G(t)^2 + Z_{i,j}^{Y,G}(t)^2 + Y_{i,j}^O(t)^2 + Z_{i,j}^{Y,O}(t)^2 + Y_i^C(t)^2 + Z_i^{Y,C}(t)^2 \right\} \quad (24)$$

Lyapunov 漂移定义为函数 $L(\Theta(t))$ 在相邻时隙函数变化期望 $\Delta L(\Theta(t)) = \mathbb{E}[L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)]$ 。其表示队列积压变化, 绝对值小表示连续两个时隙间队列积压变化小, 队列稳定。

目标函数将队列稳定性与性能指标相结合, 通过调整参数 V 可在不同目标间权衡性能最优和队列稳定。根据 Lyapunov 优化, 系统需在每个时隙最小化系统的漂移加惩罚。通过区分优化变量, 可将 P2 解耦为两个子问题:

(1) 任务卸载 -UAV 轨迹规划 (SP1):

$$SP1: \max_{x(t), d(t)} \Psi_i(t) = \Psi_L(t) - \Psi_G(t) - \Psi_O(t) - \Psi_C(t) \quad (25)$$

$$\begin{cases} \Psi_L(t) = Q_i(t)W_i(t) + \frac{Z_i^O(t)W_i(t)}{\tilde{A}_i(t+1)} \\ \Psi_G(t) = \sum_{j=1}^K x_{i,j}^O(t) \left[Y_{i,j}^G(t)W_{i,j}^G(t) + Z_{i,j}^{Y,G}(t) \frac{Y_{i,j}^G(t) + x_{i,j}^O(t)W_{i,j}^G(t)}{\tilde{W}_{i,j}^G(t+1)} \right] \\ \Psi_O(t) = \sum_{j=1}^J x_{i,j}^O(t) a_{i,j}^O(t) \left[Y_{i,j}^O(t)W_{i,j}^O(t) + Z_{i,j}^{Y,O}(t) \frac{Y_{i,j}^O(t) + x_{i,j}^O(t)W_{i,j}^O(t)}{\tilde{W}_{i,j}^O(t+1)} \right] \\ \Psi_C(t) = \left\{ x_{i,j}^C(t) \left[Y_i^C(t)W_i^C(t) + Z_i^{Y,C}(t) \frac{Y_i^C(t) + x_{i,j}^C(t)W_i^C(t)}{\tilde{W}_i^C(t+1)} \right] \right\} \end{cases} \quad (26)$$

SP1 是混合整数非凸优化问题, 传统凸优化难以求解。为此, 第四部分提出一种基于 DRL 的解决方案。

(2) 服务器计算资源分配 (SP2):

$$SP2: \max_{f(t)} \Phi_j(t) = \sum_{i=1}^I \Phi_{i,j}^G(t) + \Phi_{i,j}^O(t) + \Phi_i^C(t) \quad (27)$$

$$\begin{cases} \Phi_{i,j}^G(t) = \frac{(t+1)\tau f_{i,j}(t)}{\lambda_i \sum_{j=1}^{t-1} W_{i,j}^G(t)} + Y_{i,j}^G(t) \times \frac{\tau f_{i,j}(t)}{\lambda_i} \\ \Phi_{i,j}^O(t) = \frac{(t+1)\tau f_{i,j}(t)}{\lambda_i \sum_{j=1}^{t-1} W_{i,j}^O(t)} + Y_{i,j}^O(t) \times \frac{\tau f_{i,j}(t)}{\lambda_i} \\ \Phi_i^C(t) = \frac{(t+1)\tau f_{i,c}(t)}{\lambda_i \sum_{j=1}^{t-1} W_i^C(t)} + Y_i^C(t) \times \frac{\tau f_{i,c}(t)}{\lambda_i} \end{cases} \quad (28)$$

针对 SP2, 本文提出基于贪婪策略的解决方案。

4 任务卸载与资源管理算法设计

基于第三章优化模型, 由于子问题的异构性, 传统集中式算法难以满足实时性需求。为此, 本章提出 AFMAPPO 算法解决联合决策子问题, 以贪心算法解决资源管理子问题。

4.1 基于 AFMAPPO 的任务卸载决策

AFMAPPO 算法融合异步训练、联邦学习 (federated learning, FL) 和多智能体近端策略优化 (MAPPO) 等技术, 在提高系统效率时保护智能体的隐私。将 SP1 表述为 $(S, \mathcal{A}, P, R, \gamma)$ 的马尔可夫决策过程 (MDP), 其中 S 和 $\mathcal{A}$ 分别为状态空间和动作空间, P 为状态转移概率 $\Pr(S(t+1)|S(t), \mathcal{A}(t))$, 其取决于状态 $S(t)$ 和动作 $\mathcal{A}(t)$ 。 R 为状态映射到智能体动作的奖励函数, γ 是奖励折现因子。本文将 I 个 PIoT 设备和 J 个 UAV 作为智能体, 系统 MDP 为:

(1) 状态空间: 本文设置系统状态空间为 $S(t) = \{b_i(t), g_j(t), Q_i(t), Y_{i,j}^G(t), Y_{i,j}^O(t), \tilde{A}_i(t), \tilde{W}_{i,j}^G(t), \tilde{W}_{i,j}^O(t), \tilde{W}_i^C(t)\}$ 。时隙 t 智能体的观测值为 $O_i(t) = \phi_i(S(t))$, 则:

$$\begin{cases} O_i(t) = \{Q_i(t), \{\tilde{W}_{i,j}^G(t), \tilde{W}_{i,j}^O(t)\}_{\forall s_j \in S}, \tilde{W}_i^C(t)\} \\ O_{i+j}(t) = \{b_i(t), Y_{i,j}^O(t)\}_{\forall u_j \in U}, g_j(t) \end{cases} \quad (29)$$

(2) 动作空间: 时隙 t 智能体动作 $\mathcal{A}_i(t) = \pi_i(O_i(t))$ 。将任务卸载决策与飞行方向定义为动作空间:

$$\begin{cases} \mathcal{A}_i(t) = (x_{i,j}^O(t), x_i^C(t), a_{i,j}(t)) \\ \mathcal{A}_{i+j}(t) = (d_j(t)) \end{cases} \quad (30)$$

(3) 奖励: 智能体执行动作后, 获得相应的奖励 $R(t) = \{R_1(t), R_2(t), \dots, R_{I+J}(t)\}$ 。

$$R_i(t) = \delta \Psi_i(t) \quad (31)$$

UAV 的奖励需综合考虑其飞行轨迹能够最大化所关联设备的总效用 $\Gamma(t) = \sum_{i=1}^I x_{i,j}^O(t) a_{i,j}(t) \Psi_i(t)$ 和违反安全约束的处罚 r^s , UAV 违反安全约束时 $s_j^s = 1$:

$$R_{I+J}(t) = \delta \Gamma(t) - \zeta_j^s r^s \quad (32)$$

MAPPO 算法中智能体在集中训练时利用全局优化策略, 在分布式执行时根据局部采取行动, 目标是实现最优策略 π_i 。独立 actor 网络 $\pi_i(O_i(t); \theta_i^\pi)$ 根据局部信息和参数输出动作; 独立 critic 网络 $C_i(O(t), \mathcal{A}(t); \theta_i^c)$ 根据训练的状态和动作评估 actor 网络的动作。此外, 队列积压会随着迭代不断更新, 训练良好的智能体借助深度神经网络进行决策。智能体 critic 网络的均方误差损失为 $L_i(\theta_i^c)$, 其中 $y_i(t)$ 为临界目标。

$$L_i(\theta_i^c) = \mathbb{E}[C_i(O(t), \mathcal{A}(t); \theta_i^c) - y_i(t)]^2 \quad (33)$$

$$y_i(t) = R_i(t) + \gamma C_i(O(t+1), \mathcal{A}(t+1); \theta_i^c) \Big|_{\mathcal{A}_i(t+1) = \pi_i(O_i(t+1); \theta_i^\pi)} \quad (34)$$

更新参数 θ_i^c , 其中 ξ^c 为 critic 网络的学习率:

$$\theta_i^c = \theta_i^c - \xi^c \nabla_{\theta_i^c} L_i(\theta_i^c) \quad (35)$$

更新 actor 网络使得损失最小化, 使 critic 网络的评价输出最大, 其中 ξ^π 为 actor 网络学习率。

$$\theta_i^\pi = \theta_i^\pi - \xi^\pi \nabla_{\theta_i^\pi} J_i \quad (36)$$

在模型训练过程中, 需要共享大量数据来提升模型性能,

而数据的发送会占用通信资源，同时在通信中还存在信息泄露的风险。为此，在 MAPPO 框架中引入 FL 可在保证数据隐私安全的前提下实现模型性能的优化。各智能体训练本地模型，仅共享模型参数而非原始数据，避免敏感数据的泄露和丢失，能够减少通信开销、动态适应局部环境变化。

异步联邦学习（asynchronous federated learning, AFL）可允许客户端分流上传收敛模型并训练全局模型。在传统联邦平均（FedAvg）中，服务器需等待所有客户端完成训练后同步聚合参数，而异步 FedAvg 采用按批次进行聚合。服务器累积一定数量参数后，统一加权平均。当服务器接收到新模型时，采用软更新对模型参数进行更新，以减少新模型中过时参数的影响，参数聚合用公式表示为：

$$D(t+1) = D(t) + \alpha \sum_{m=1}^M \omega_m \mu^m (D_m(t) - D(t)) \quad (37)$$

式中： α 为全局学习率，控制聚合更新的幅度； $\omega_m = n_m / \sum_{m=1}^M n_m$ 为批次内客户端权重； μ^m 为延迟衰减因子，其可降低过时参数对全局更新的影响。算法流程如算法 1 所示。

算法 1 基于 AFMAPPO 的任务卸载算法

输入：各智能体 actor 和 critic 网络及队列信息， $T, I, J, V, D, \{Q(t), Y_{i,j}^G(t), Y_{i,j}^O(t), Y_i^C(t), \tilde{A}(t), \tilde{W}_{i,j}^G(t), \tilde{W}_{i,j}^O(t), \tilde{W}_i^C(t)\}$ 。

输出：最优决策 $\{x_{i,j}^O(t), x_i^C(t), a_{i,j}(t), d_j(t)\}$ 。

初始化： $\{\theta_i^x, \theta_i^c, \theta_i^{x'} = \theta_i^x, \theta_i^{c'} = \theta_i^c\}$ 。

Step1：每批次训练开始，客户端从中央服务器下载全局模型参数 $D(t)$ ，并更新 $D_{i,j}(t) = D(t)$ 。

Step2：设备基于观测 $O(t)$ 与环境交互，生成动作 $A_i(t)$ 。

Step3：UAV 基于观测 $O_{i,j}(t)$ 与动作 $A_{i,j}(t)$ 维持安全区域。

Step4：计算每个智能体的奖励 $R_i(t)$ ，更新状态 $S(t+1)$ ，根据算法 2 分配边缘计算资源。

Step5：更新参数 θ_i^x, θ_i^c 。

Step6：更新本地模型参数 $D_{i,j}(t+1)$ ，并将其模型参数异步向中央服务器传输，汇总本批次客户端参数 $D(t+1) = D(t) + \alpha \sum_{m=1}^M \omega_m \mu^m (D_m(t) - D(t))$ ，更新全局模型参数。

4.2 计算资源管理

针对 SP2，本文提出基于贪心算法的资源管理方法。每个服务器在第 t 时隙获得与其关联的 PiOT 设备集，服务器优先将计算资源分配给 $\Phi(\rho_{i,j}(t))$ 最大的 u_i ，并将 u_i 从 U_i 中移除，直到任务都分配到资源或可用资源 $F(t)=0$ 结束进程，算法流程如算法 2 所示。

算法 2 计算资源分配算法

输入： $\{\lambda_i, Y_{i,j}^G(t), Y_{i,j}^O(t), Y_i^C(t), f_{j,\max}^G, f_{j,\max}^O, f_{\max}^C\}$

输出：最优计算资源分配方案

while $t=1 \sim T$ do

while $s_j \in S$ do

$U_i = \{u_i \in U | Y(t) > 0\}; F(t) = \{F_j^G = f_{j,\max}^G, F_j^O = f_{j,\max}^O, F^C = f_{\max}^C\}$

while $U_i \neq \emptyset; F(t) \neq \emptyset$ do

$$\rho_{i,j}(t) = \min\{F(t), \lambda_i Y(t)/\tau\}$$

$$u_i^* = \arg \max_{u_i \in U} \Phi(\rho_{i,j}(t))$$

$$f_{i,j}^*(t) = \rho_{i,j}(t)$$

$$F(t) = F(t) - f_{i,j}^*(t)$$

$$U_i = U_i \setminus u_i^*$$

end while

end while

end while

5 实验结果与分析

本节通过模拟所提系统环境来评估 AFMAPPO 算法的有效性，本研究基于 Python3.8 搭建 PiOT 仿真环境，集成 PyTorch1.10 实现 DRL 模型，采用 Flower 1.0 框架支持 FL 过程。用多架 UAV 和多个设备模拟 1 000 m×1 000 m PiOT 区域。UAV 飞行高度 $H=150$ m，速度 $v=10$ m/s。UAV 之间的安全距离 $\varepsilon=3$ m，到达任务数 $A_i(t)$ 在 $[15.7, 26.3] \times 10^4$ bit 内随机分布，服务器最大可用资源 $f_{\max}$ 在 $[78.3, 123.2]$ GHz 内随机分布。总时间 T 设为 100 s，每个时隙 $\tau=0.1$ s。将本文算法与 MAPPO、PPO、遗传算法（genetic algorithm, GA）进行比较。

图 2 为两架 UAV 和 2 个设备的场景中 UAV 轨迹，（a）（b）分别为二维、三维轨迹图。每架 UAV 根据最优的卸载方式执行飞行，从初始位置飞到目标设备，接受其卸载任务。

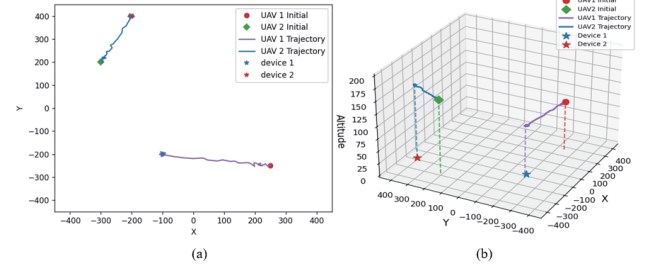


图 2 飞行轨迹图

图 3 表示各层队列积压情况。系统中设备和 UAV 可根据实时任务队列动态调节任务卸载策略和飞行轨迹，实现最优的卸载和飞行决策，从而减少任务积压。

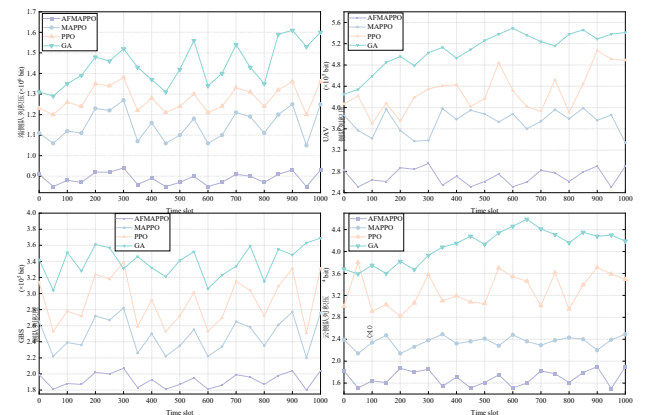


图 3 各层队列积压情况

图4表示任务卸载总队列延迟变化。与基准算法相比, AFMAPPO 将总排队延迟分别降低了 47.8%、65.4% 和 68.5%。其通过异步上传本地模型和放弃异常的局部模型来减轻通信影响, 达到最低的峰值增量和总队列延迟。

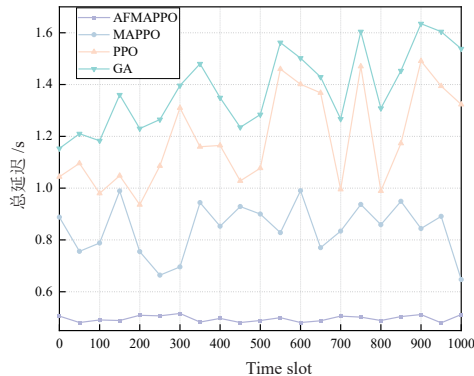


图4 总队列延迟

图5为不同设备数量时, 不同算法的任务完成率。随着设备的增加, 需要将有限的资源分配给更多的设备, 使得任务完成率下降。与其他算法相比, AFMAPPO 根据任务类型为设备分配计算资源, 将任务完成率提升至 89.5%。

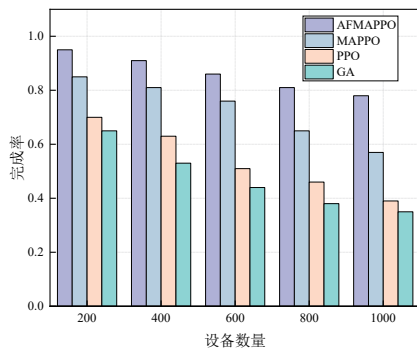


图5 任务完成率

6 结论

本文研究 SAG-PIoT 系统中任务卸载、UAV 轨迹和资源分配的联合优化问题。设计基于队列信息与性能指标的集中训练算法 AFMAPPO, 以动态调整卸载决策与飞行轨迹, 并采用贪心算法实现最优资源分配。实验表明, 该方案在任务积压、队列延迟及完成率上均取得最佳性能。未来将进一步拓展系统以适应更多不同场景。

参考文献:

- [1] 贺超, 汪超凡, 张思睿, 等. 电力物联网自适应任务卸载策略[J]. 重庆邮电大学学报(自然科学版), 2025, 37(4): 507-515.
- [2] 丁忠林, 李洋, 曹委, 等. 基于深度 Q 学习的电力物联网任务卸载研究[J]. 计算机与现代化, 2022(11): 75-80.
- [3] 曾琦, 刘友波, 李梓玮, 等. 支持新型电力系统的 5G/6G

高可靠低时延电力物联网通信设计与分析[J]. 中国电机工程学报, 2024, 44(21): 8435-8444.

- [4] 毛嘉辉, 朱雪琼, 胡成博, 等. 基于改进遗传算法的电力物联网安全通信协作干扰策略[J]. 计算机与现代化, 2025(3): 66-70.
- [5] 李宁, 于晓清, 陈炜, 等. 电力物联网移动边缘计算任务卸载策略[J]. 电测与仪表, 2024, 61(4): 155-160.
- [6] ZHOU H, ZHANG Z Y, LI D W, et al. Joint optimization of computing off loading and service caching in edge computing-based smart grid[J]. IEEE transactions on cloud computing, 2022, 11(2): 1122-1132.
- [7] 刘向军, 周凯, 孙毅, 等. 面向新型电力系统的边缘缓存与计算资源迁移策略[J]. 电力系统保护与控制, 2023, 51(24): 134-142.
- [8] WANG Q J, SHAO S J, GUO S Y, et al. Task allocation mechanism of power internet of things based on cooperative edge computing[J]. IEEE access, 2020, 8: 158488-158501.
- [9] SUN M Y, YUAN Y Z, MA K, et al. Spectrum allocation and computing resources optimization for demand-side cooperative communications in smart grid[J]. IEEE transactions on smart grid, 2021, 13(3): 1967-1975.
- [10] LI J L, DAI M H, SU Z. Energy-aware task offloading in the Internet of things[J]. IEEE wireless communications, 2020, 27(5): 112-117.
- [11] YUAN X P, YANG T Y, HU Y L, et al. Trajectory design for UAV-enabled multiuser wireless power transfer with nonlinear energy harvesting[J]. IEEE transactions on wireless communications, 2020, 20(2): 1105-1121.
- [12] WEI Q, ZHOU Z, CHEN X. DRL-based energy-efficient trajectory planning, computation offloading, and charging scheduling in UAV-MEC network[C]//2022 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE, 2022: 1056-1061.
- [13] HE C Z D, WU Y Y, HUANG M X, et al. Machine learning-based two-stage task offloading optimization for power distribution internet of things[J]. Wireless communications and mobile computing, 2022, 2022(1): 3365279.

【作者简介】

武鹏荣(2001—), 男, 山西朔州人, 硕士研究生, 研究方向: 电力物联网任务卸载。

李振文(1996—), 男, 山西忻州人, 硕士研究生, 研究方向: 车辆边缘计算任务卸载。

雷强(1998—), 男, 江西丰城人, 硕士研究生, 研究方向: 微电网调度。

(收稿日期: 2025-06-26 修回日期: 2025-12-05)