引用格式：朱思峰, 许浩, 张青华, 等. 面向 SAGIN 场景的无人机缓存决策 [J]. 电子科技大学学报, 2026, 55(3): 350-360.
ZHU S F, XU H, ZHANG Q H, et al. Research on UAV cache decision in SAGIN scenario[J]. Journal of University of Electronic Science and Technology of China, 2026, 55(3): 350-360.

面向 SAGIN 场景的无人机缓存决策

朱思峰[1]，许　浩[1]，张青华[2]，张宗辉[1]，郝志鹏[1*]，鲍　磊[1]，
乔　蕊[3]，陈国强[4]，许蒙蒙[5]，朱　海[5]

（1. 天津城建大学计算机与信息工程学院，天津 300384；2. 天津城建大学图书馆，天津 300384；3. 周口师范学院计算机学院，周口 466001；
4. 河南大学计算机与信息工程学院，开封 475001；5. 河南工程学院计算机学院，郑州 451191）

摘要：针对移动边缘计算场景下数据流量高速增长带来的任务缓存时延过长的问题，提出基于深度强化学习算法的缓存优化方案。首先，在使用 5G 技术的空天地一体化网络场景架构下，以最小化任务缓存时延为目标，对缓存和通信模型进行建模；其次，使用 SAC（soft actor-critic）算法对局部最小缓存时延进行干扰，并根据一定概率接受新方案，从而获得全局最大缓存命中率；最后，对上述过程重复迭代，得到目标问题的一个最优解，确保了任务文件被预缓存在最优位置。仿真结果表明，在空天地一体化网络协同架构下，与 PPO 方案相比，该方案能够减少优化传输效率，降低缓存时延 5.30%，提高缓存命中率 3.90%。

关键词：移动边缘计算；深度强化学习；无人机；空天地一体化网络；缓存决策
中图分类号：TP393.1　　**文献标志码：**A　　DOI: 10.12178/1001-0548.2025155

Research on UAV cache decision in SAGIN scenario

ZHU Sifeng[1], XU Hao[1], ZHANG Qinghua[2], ZHANG Zonghui[1], HAO Zhipeng[1*], BAO Lei[1], QIAO Rui[3], CHEN Guoqiang[4], XU Mengmeng[5], and ZHU Hai[5]

(1. School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China;
2. Library, Tianjin Chengjian University, Tianjin 300384, China;
3. School of Computer, Zhoukou Normal University, Zhoukou 466001, China;
4. School of Computer and Information Engineering, Henan University, Kaifeng 475001, China;
5. School of Computer, Henan University of Engineering, Zhengzhou 451191, China)

Abstract: In the mobile edge computing scenario, the task cache delay becomes excessively long due to the rapid growth of data traffic. To address this issue, a cache scheme based on deep reinforcement learning algorithm in mobile edge computing scenarios is proposed. Firstly, under a 5G-based space-air-ground integrated network (SAGIN) architecture, a cache and communication model is established to minimize the content cache delay. Secondly, the SAC (soft actor critical) algorithm is used to interfere with the local minimum cache delay, and a new scheme is accepted with a certain probability, thereby achieving the global maximum cache hit rate. Finally, the above process is iterated repeatedly to obtain an optimal solution for the target problem, ensuring that the task files are pre-delayed in the optimal location. The simulation results show that under the SAGIN cooperation architecture, compared with the PPO (proximal policy optimization) scheme, the cache scheme can reduce the optimization transmission efficiency, reducing the cache delay by 5.30%, and improving the cache hit rate by 3.90%.

Key words: mobile edge computing; deep reinforcement learning; unmanned aerial vehicle; air-space-ground integration network; caching strategy

---

收稿日期：2025−07−24
基金项目：国家自然科学基金（62172457）；天津市自然科学基金重点项目（22JCZDJC00600）；河南省高校科技创新人才支持计划（23HASTIT029）；河南省科技攻关项目（242102210027）
作者简介：朱思峰，教授，主要从事空天地一体化网络、车联网、云边协同计算、人工智能算法及应用等方面的研究。
*通信作者 E-mail: hzpqina@163.com

近年来，新能源技术的发展推动着智能网联汽车的进步，随之带来用户的出行需求愈发纷繁多样[1]。智能网联汽车装载多个传感器以实现智能驾驶、道路检测、线路规划等任务。智能网联汽车任务量的增加以及任务难度的提升带来了算力需求与存储需求的急剧增加，但受限于智能汽车终端的计算能力和存储能力，传统智能汽车终端难以完成用户多样化的任务需求。相较于只依靠独立智能汽车终端完成任务处理或云中心计算模式，移动边缘计算（mobile edge computing, MEC）能够为智能汽车终端的任务计算给予强大的算力及庞大的存储能力，并提供可靠的任务缓存策略服务。传统的 MEC 是将用户的任务以及所需的任务文件缓存到道路两侧的边缘服务器（edg server, ES）辅助智能汽车进行任务处理[2-3]。通过将用户后续大概率所需要的任务文件提前存储至 ES 配置的存储器中，之后通过无线通信技术将内容输送至用户。由于无需频繁请求云中心，所以能够减少任务文件的传输时延、提高存储设备的利用率及增加存储设备的服务时长[4-5]。

在更加复杂的交通网络环境中，大量用户对当前热门任务的重复下载是移动数据流量急剧攀升的重要原因之一。为了应对未来海量数据请求带来的通信压力，MEC 使用边缘缓存技术将网络存储功能应用在 ES 的边缘计算节点，就近为用户提供所需的热门内容文件，从而缓解交通网络的传输压力。目前，具有更大优势的空天地一体化网络（space-air-ground integrated network, SAGIN）与 MEC 深度融合，地面通信网络和卫星通信网络加入无人机通信网络构成了 SAGIN[6-7]。无人机（unmanned aerial vehicle, UAV）作为低空载具拥有无视地形限制、便于部署、灵活移动的特点，因此可将 UAV 加入 SAGIN 与 MEC 深度结合。UAV 可携带微型服务器为智能汽车提供任务文件缓存服务，以及在 UAV 存储器中缓存当前用户群体依赖性更高热门内容文件，减少用户任务请求云中心的频率，从而更深层次地减轻通信拥塞程度和网络传输压力，使系统服务范围更广、服务质量更可靠、服务效率更高效。

深度强化学习（deep reinforcement learning, DRL）的发展使其展现出能与多领域融合的强大优势，大量学者将 DRL 应用在 MEC 领域中，并开始使用 DRL 解决用户处理任务过程的缓存问题。DRL 由于马尔可夫决策拥有了强大的自适应学习能力，为 MEC 提供了更为高效和智能的资源管理和任务文件缓存方案[8-9]。在 SAGIN 中，移动边缘计算的服务器需要实时分析海量庞杂的用户数据和及时处理用户纷繁多样的任务，以满足各种车载应用的多样化需求。而 DRL 结合深度学习和强化学习的优势，不断从环境中学习并优化自身决策能力，完成对实时状态的预测判断并有针对性地采取最优决策，使得 MEC 系统可以更好地应对动态变化的复杂场景，从而提高资源利用效率和缓存服务质量。因此，本文将研究如何利用 DRL 方法来优化具有 UAV 辅助的 MEC 系统，使其降低时延与能耗，提高缓存文件的命中率。

1　相关工作

传统的地面 MEC 会在网络边缘节点 ES 部署边缘服务器，将任务文件缓存到 ES 中；在传统云中心 MEC 中部署远程中央服务器并部署相应的任务文件，云端服务器通过远程通信对智能汽车进行辅助任务缓存，从而缓解智能汽车本地运行负载过大问题，但远程云端计算存在高时延的缺陷。

近年来，学者对 MEC 的缓存时延与能耗优化的问题进行了大量研究。文献 [10] 提出了 MEC 融合架构，该架构能动态分配任务至云/边缘节点，并且定义了混合架构资源协调机制，支持基于任务需求的动态请求，从而降低系统时延。但并未考虑缓存问题带来的影响。文献 [11] 定义移动 QoS 双层指标，分析了移动性影响，但是未分析卫星通信低轨卫星能力、未提及卫星链路长时延对 QoS 管理的影响。文献 [12] 针对缓存决策提出了两级协作缓存与动态决策模型。但是并没有考虑到卫星节点浮动因素，并且未利用卫星广覆盖能力处理偏远地区任务。文献 [13] 针对时延与能效问题提出了一种基于长期演进的移动边缘计算系统模型，通过将计算任务缓存到边缘服务器来减少延迟并提高 5G 系统的计算效率，但是并没有考虑到边缘服务器的计算能力和存储容量限制。文献 [14] 提出了四维联合优化框架，将资源发现顺序、分配、传输功率、缓存比例抽象为随机序列决策问题，实现端到端能耗最小化。

目前也有部分工作对服务器缓存文件放置问题进行了研究。文献 [15] 提出了缓存资源分配问题分解框架，将复杂问题拆分为基站缓存容量和请求匹配两个子问题，分别优化基站缓存部署和用户-基站匹配策略。文献 [16] 针对任务流行度对任务

缓存和服务器部署之间的耦合性，提出了一种基于用户的加权 K-means 服务器优化方案和基于 Q–leaning 算法的缓存放置优化算法。文献 [17] 针对区域性无线网络模型中的无人机调度和内容缓存策略的联合优化问题，以优化服务器节点的任务缓存成功率为目标，提出了一种 DC-DRL 策略。上述文献均未对无人机的终端设备关联问题进行优化。文献 [18] 针对用户和无人机服务器之间的关联性提出了一种点对点的转递匹配模型，并结合了用户满意度设计了一种耦合缓存策略与用户体验的近似最优的优化方案。但该方案并没有考虑到连续变化的用户会对服务器的部署带来影响，因此无法在特定的区域下得到近似最优的服务器部署方案。

上述方案多数无法兼顾用户任务生成的动态性与智能车辆定位信息的拓扑变换，且不能很好地从系统全局对 MEC 进行整体性能优化。为了提高缓存管理问题中的资源利用率，本文首先将问题分解为连续的无人机移动问题和离散的任务缓存问题，然后对 UAV 移动、任务缓存决策、用户目标进行解耦。将无人机的状态空间（如飞行角度、速度等）定义为连续变量，即状态之间的转换是平滑的，这些状态变量随时间发生平滑、连续的演变。将缓存管理的状态离散化，即其决策变量是离散的，如缓存内容文件与有限的存储器之间的缓存映射表等；其次，针对传统智能算法在高维混合决策空间中对任务关联性利用不高的缺点，本文使用 DRL 技术求解资源分配最优决策。

最后，本文通过对无人机的移动位置部署、任务文件缓存分配以及任务文件预缓存联合优化来达到最小化卸载时延、最大化缓存命中率的目的，从而提高整体的服务质量和系统性能。

2 系统模型

2.1 系统架构

基于 V2X 技术和无线通信技术，本文构建了空天地一体化网络场景下边缘计算系统结构，如图 1 所示。该系统架构中部署了一个搭载中央服务器的 LEO、多个装载微型服务器和缓存器的 UAV、多个具有任务文件需求的智能网联汽车、多个具备缓存能力的 ES。在该系统架构中，智能汽车会根据任务类型生成多样化的内容需求，优先在自身内容库中匹配，若未匹配成功则需要向 ES 或者 UAV 服务器发出内容请求。LEO 作为云中心处理信息中枢，其职能是收集 ES、UAV 和智能汽车的数据信息、各智能汽车所需的参数以及规划 UAV 运动轨迹，并通过采集各个 UAV 的定位信息、智能汽车的聚集状态以及生成任务的需求信息进行智能调度；每个 ES 服务器的覆盖区域有限，当智能汽车驶出当前 ES 服务器的通信范围时，需要距离最近的其他 ES 服务器建立新的通信连接。在建立新通信连接的过渡期，UAV 可为智能汽车提供连续的通信服务。UAV 集群能够与多个 ES 协同合作，服务某个区域的智能汽车。每个 UAV 将会根据场景的动态变换进行智能移动，从而提升整个系统的任务缓存效率和缓存流畅度。

本文设计的空天地一体化网络系统架构中包含 1 个具有计算能力和缓存能力的 LEO；U架具有计算能力和缓存能力的 UAV，UAV 集合表示为 $\mathcal{U}=\{U_i|1\leqslant i\leqslant U\}$；$E$个具有计算能力和缓存能力的 ES，ES 集合表示为$\mathcal{E}=\{E_i|1\leqslant i\leqslant E\}$；有$C$个智能汽车在智能交通系统道路上行驶，智能汽车集合表示为$\mathcal{C}=\{C_i|1\leqslant i\leqslant C\}$；智能汽车可能请求的所有内容文件集合表示为$\mathcal{F}=\{F_i|1\leqslant i\leqslant F\}$，该集合中内容文件的大小为$\mathcal{D}=\left\{D_f|f\in\mathcal{F}\right\}$；智能汽车与任务内容文件的关联变量用$O_{i,j}\in\{0,1\}$表示，若第$i$个智能汽车$C_i$向 ES 请求了内容任务$F_j$，则$O_{i,j}=1$，否则$O_{i,j}=0$。

本文利用三维笛卡尔坐标表示智能汽车、ES 和 UAV 的定位坐标。系统架构以离散时隙运行，将系统运行周期T分割为N个大小为$\delta_t=T/N$的时隙，将 UAV 在不同时隙的位置变化视为 UAV 的运动轨迹，时隙集合表示为$\mathcal{N}=\{N_i|1\leqslant i\leqslant N\}$，每个时隙内每一个智能汽车只与一个 UAV 和 ES 保持通信。C个智能汽车在规定的范围区域内移动，第i个智能汽车的位置信息用三维坐标$q_{\mathrm{Ve},i}=\left[x_{\mathrm{Ve},i},y_{\mathrm{Ve},i},0\right]$表示；UAV 服务器部署在规定范围的上方。假定同一个时隙内，UAV 和智能汽车视为相对静止，那么第u个 UAV 的位置信息可以表示为$q_{\mathrm{UAV},u}=\left[x_{\mathrm{UAV},u},y_{\mathrm{UAV},u},h_{\mathrm{UAV},u}\right]$。无人机的飞行速度记为$v_u$，重量记为$M_{\mathrm{UAV}}$，UAV 的飞行高度记为$h_{\mathrm{UAV},u}\in[h_{\min}，h_{\max}]$。第$e$个 ES 的位置信息可以表示为$q_{\mathrm{BS},e}=\left[x_{\mathrm{BS},e},y_{\mathrm{BS},e},h_{\mathrm{BS},e}\right]$。第$i$个智能汽车与第$u$个 UAV 的距离表示为$d_{i,u}=\left\|q_{\mathrm{Ve},i}-q_{\mathrm{UAV},u}\right\|$。第$i$个智能汽车与第$e$个 ES 的距离表示为$d_{i,e}=\left\|q_{\mathrm{Ve},i}-q_{\mathrm{BS},e}\right\|$。

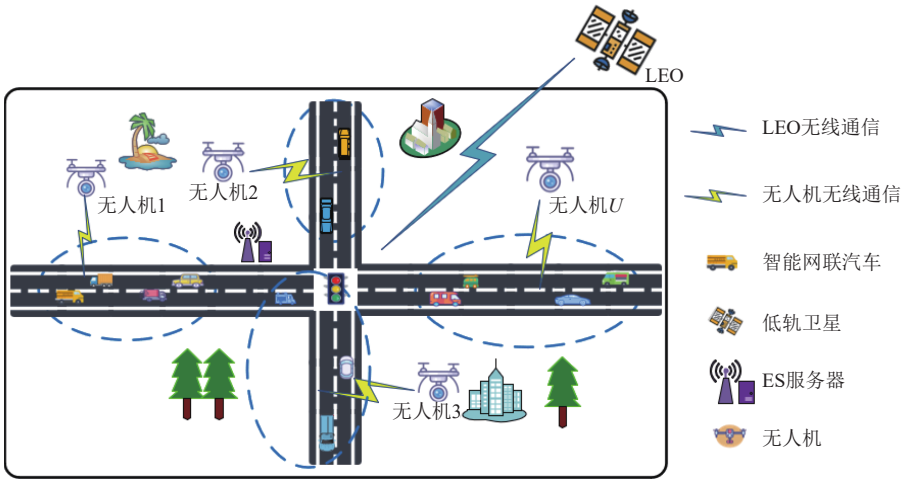


图 1　空天地一体化网络场景下的边缘计算系统架构

2.2　缓存模型

架构中智能汽车的任务文件需求是动态变化的，智能汽车的存储器容量也是有限的。当前时隙若智能汽车所需要的任务文件在本地存储器无法检索到时，车载会向附近的 ES 服务器发出任务文件缓存请求。接收到任务请求的 ES 服务器会检索自己的文件目录，若在 ES 存储器检索到请求文件，ES 则会将该文件发送至智能汽车，否则 ES 会向附近的 UAV 转发任务文件请求。接收到任务请求的 UAV 会检索自己的文件目录，若 UAV 在存储器检索到请求文件，UAV 则会将该文件输出至智能汽车，否则向 LEO 发出任务文件缓存请求。LEO 会调度已有文件的服务器将该缓存文件转发到该无人机并发送到智能汽车。

假设每台服务器的存储容量均为KMb 大小，LEO 包含了 ES 与 UAV 已有的内容文件目录信息。ES 与任务内容文件的关联变量用$Q_{i,f}\in\{0,1\}$表示，若第i个 ES 已缓存内容文件F_f至存储器，则$Q_{i,f}=1$，否则$Q_{i,f}=0$。UAV 与任务内容文件的关联变量用$Z_{i,f}\in\{0,1\}$表示，若第i个 UAV 已缓存内容文件F_f至存储器，则$Z_{i,f}=1$，否则$Z_{i,f}=0$。通过离散关联变量完成对缓存内容文件与存储器的离散化描述。

在某个时隙内，第i个智能汽车C_i向最近的 ES 服务器E_j和 UAV 服务器 U_m请求任务文件F_f缓存时的命中信号变量$Y_{i,f}$为：

$$Y_{i,f}=\begin{cases}1 & O_{i,f}=Q_{j,f}=Z_{m,f}=1\\ 0 & O_{i,f}=1;Q_{j,f}=0\text{或}Z_{m,f}=0\end{cases}\tag{1}$$

式中，$O_{i,f}$为第i个智能汽车C_i与任务文件F_f的关联变量；$Q_{j,f}$为第j个 ES 服务器E_j与内容文件F_j的关联变量；$Z_{m,f}$为第m个 UAV 服务器U_m与内容文件F_f的关联变量。$Y_{i,f}=1$表示在单个时隙内第i个智能汽车C_i向最近的 ES 服务器E_j和 UAV 服务器 U_m请求任务文件F_f缓存成功；$Y_{i,f}=0$表示缓存失败。

当智能汽车向就近的 ES 以及 UAV 请求缓存文件时，若 ES 或 UAV 存有该缓存文件，则视为智能汽车的请求内容文件成功命中，智能汽车请求缓存文件的平均缓存命中率为：

$$P_{\text{hit}}=\frac{\sum_{i=1}^{C}\sum_{n=1}^{F}Y_{i,n}}{C}\tag{2}$$

式中，C表示智能汽车的总量；$Y_{i,n}$表示在单个时隙内第i个智能汽车C_i向最近的 ES 服务器E_j和 UAV 服务器 U_m请求任务文件F_n缓存命中信号变量。

2.3　通信时延模型

本研究涉及一种集合地面、卫星和无人机一体化的通信架构，为了防止智能汽车通信受到干扰，假设通信网络拥有相同的频谱。当前架构下的带宽定义为B，且采用正交频分技术限制不同基站之间的干扰。智能汽车c和无人机u的信道增益为：

$$h_{c,u}=\frac{\beta_0}{d_{i,u}^2}=\frac{\beta_0}{\left\|q_{\text{Ve},i}-q_{\text{UAV},u}\right\|^2}\tag{3}$$

式中，β_0表示信道增益；$d_{i,u}$表示智能汽车i与无人机u的欧几里德距离。

由于将整个系统的运行周期分割为多个大小相同的时隙，因此假设 UAV 在每个时隙与智能汽车视为相对静止。在单个时隙内，智能汽车c向无人机u进行任务传输的速率为：

$$R_{c,u}=B\log_2\left(1+\frac{p_{c,u}h_{c,u}}{\sigma^2+b_{c,u}P_{\mathrm{NLoS}}}\right) \tag{4}$$

式中，B表示带宽；σ^2表示高斯噪声功率；$b_{c,u}$表示智能汽车 c与无人机u的通信遮挡信号；当智能汽车c与无人机u之间的通信无遮挡物时，$b_{c,u}=1$，否则$b_{c,u}=0$；$p_{c,u}$为智能汽车c向无人机u的发射功率，P_{NLoS}为智能汽车和服务器之间的障碍物产生的非视距（non line of sight, NLoS）。

智能汽车c和 ES 服务器e的信道增益为：

$$h_{c,e}=\frac{\beta_0}{d_{i,e}^2}=\frac{\beta_0}{\left\|q_{\mathrm{Ve},i}-q_{\mathrm{BS},e}\right\|^2} \tag{5}$$

式中，$d_{i,e}$表示智能汽车i与 ES 服务器e的欧几里德距离。

在单个时隙内，智能汽车c向 ES 服务器e进行任务传输的速率为：

$$R_{c,e}=B\log_2\left(1+\frac{p_{c,e}h_{c,e}}{\sigma^2+b_{c,e}P_{\mathrm{NLoS}}}\right) \tag{6}$$

式中，$b_{c,e}$为智能汽车c与 ES 服务器e的通信遮挡信号，当智能汽车c与 ES 服务器e之间无遮挡物时，$b_{c,e}=1$，否则$b_{c,e}=0$；$p_{c,e}$为智能汽车c向 ES 服务器e的发射功率。

假设：每个智能汽车在每个时隙内最多只生成一个任务，且该任务可以根据需求划分为不同大小的子任务；系统总时延由传输时延和替换时延两部分组成。智能汽车向 ES 服务器进行任务文件缓存时的传输时延为：

$$T_{c,e}^{f}=\frac{\sum_{f=1}^{F}O_{i,f}D_f}{R_{c,e}} \tag{7}$$

式中，$R_{c,e}$表示智能汽车与 ES 的传输速率；$O_{i,f}$表示智能汽车C_i与任务文件F_f的关联信号变量；D_f表示F_f的文件大小。

若智能汽车请求的缓存文件$f\in\mathcal{F}$在 ES 服务器检索失败时，则需要将该缓存请求协同至 UAV。UAV 协同缓存的传输时延为：

$$T_{c,u}^{f}=\frac{\sum_{s=1}^{S}\sum_{f=1}^{F}O_{i,f}\left(1-Q_{s,f}\right)D_f}{R_{c,u}} \tag{8}$$

式中，$Q_{s,f}$表示 ES 服务器与任务文件F_f的关联信号变量。

若智能汽车请求的缓存文件$f\in\mathbb{F}$在 ES 以及 UAV 检索失败时，则需要请求 LEO 对 UAV 进行缓存文件的更新替换。UAV 缓存文件替换时延为：

$$T_{\mathrm{ex}}^{f}=\frac{\sum_{e=1}^{E}\sum_{f=1}^{F}O_{i,f}\left(1-Q_{e,f}\right)\left(1-Z_{u,f}\right)D_f}{R_{\mathrm{ex}}} \tag{9}$$

式中，R_{ex}表示 ES 与 UAV 的传输速率；$Z_{u,f}$表示无人机U_u与任务文件F_f的关联信号变量。

因此，智能汽车C_i与文件缓存F_f所需要的总时延为：

$$T_{\mathrm{total}}^{f}=T_{c,e}^{f}+T_{c,u}^{f}+T_{\mathrm{ex}}^{f} \tag{10}$$

式中，$T_{c,e}^{f}$表示智能汽车向 ES 进行任务文件缓存时的传输时延；$T_{c,u}^{f}$表示 UAV 协同缓存的传输时延。

2.4 问题模型

本文的目标是最小化任务缓存时延、最大化缓存命中率。考虑到缓存决策、通信时延、计算时延以及轨迹优化，本文将 SAGIN 中的智能汽车任务缓存问题建模为一个联合优化问题：

$$\min T,\max P_{\mathrm{hit}}$$

$$\sum_{f=1}^{F}Q_{i,f}D_f\leqslant K \tag{11}$$

$$\sum_{f=1}^{F}Z_{i,f}D_f\leqslant K \tag{12}$$

$$h_{\mathrm{UAV},u}\in[h_{\min},h_{\max}] \tag{13}$$

$$O_{i,f}\in\{0,1\};Q_{i,f}\in\{0,1\};Z_{i,f}\in\{0,1\} \tag{14}$$

$$Y_{i,f}\in\{0,1\} \tag{15}$$

$$b_{c,u}\in\{0,1\},\forall c\in\mathrm{Vehicles},\forall u\in\mathrm{UAV}s \tag{16}$$

式 (11) 表示每台 ES 所缓存的内容文件不会超过其自身最大的存储上限；式 (12) 表示每台 UAV 所缓存的内容文件不会超过其自身最大的存储上限；式 (13) 表示每台无人机的飞行高度限制范围；式 (14) 表示智能汽车、ES 和 UAV 与内容文件F_f的联系标志变量，其中$O_{i,f}$表示智能汽车是否请求文件F_f，$Q_{i,f}$表示 ES 是否已缓存F_f，$Z_{i,f}$表示 UAV 是否已缓存F_f；式 (15) 表示智能汽车的请求内容是否在 ES 或 UAV 中；式 (16) 表示 UAV 与智能汽车的遮挡情况。

3　基于深度强化学习的缓存方案

3.1　深度强化学习

本研究中缓存时延、缓存命中率与 UAV 的轨迹移动是变量高度耦合的非线性规划问题。针对此问题，提出了一种基于 DRL 的缓存迁移方案来实现动态边缘车联网中的最优缓存策略，该方案采用 SAC（soft actor-critic）算法来联合优化系统缓存时延、缓存命中率与 UAV 轨迹移动，可以根据场景动态调整缓存策略。

本文方案将智能汽车的任务需求、环境状态作为输入；把缓存策略与 UAV 轨迹移动作为输出；在任务缓存的优化过程中，SAC 智能体会根据策略π在当前状态s_t下选择一个动作a_t，智能体做出所选动作a_t后会获得一个即时反馈奖励r_t，并且导致环境从s_t转移到新状态s_{t+1}。

3.1.1　状态空间

在单个时隙内 Agent 观察的状态空间为：

$$\begin{aligned}\mathcal{S}^t=\{&q^t_{\mathrm{Ve},1},\cdots,q^t_{\mathrm{Ve},C},q^t_{\mathrm{BS},1},\cdots,q^t_{\mathrm{BS},E},q^t_{\mathrm{UAV},1},\cdots,q^t_{\mathrm{UAV},C},\\&O^t_{1,1},\cdots,O^t_{1,F},\cdots,O^t_{C,1},\cdots,O^t_{C,F},Q^t_{1,1},\cdots,Q^t_{1,F},\cdots,\\&Q^t_{C,1},\cdots,Q^t_{C,F},Z^t_{1,1},\cdots,Z^t_{1,F},\cdots,Z^t_{C,1},\cdots,Z^t_{C,F},\cdots,\\&b^t_{1,1},\cdots,b^t_{1,u},b^t_{c,1},\cdots,b^t_{c,u}\}\end{aligned}\tag{17}$$

式中，$q^t_{\mathrm{Ve},c}$表示智能汽车c在t时隙时所处的位置；$q^t_{\mathrm{BS},e}$表示 ES 服务器e在t时隙所处的位置；$q^t_{\mathrm{UAV},u}$表示 UAV 服务器u在t时隙所处的位置；$O^t_{c,f}$表示在t时隙时用户c与任务文件f的关联变量；$Q^t_{e,f}$表示在t时隙时 ES 服务器e与任务文件f的关联变量；$Z^t_{u,f}$表示在t时隙时 UAV 服务器u与任务文件f的关联变量；$b_{c,u}(n)$表示在t时隙时 UAV 服务器u和用户c终端通信链路是否有遮挡。

3.1.2　动作空间

在单个时隙内 Agent 根据当前观察到的系统状态选择需要采取的动作，动作空间为：

$$\begin{aligned}\mathcal{A}^t=\{&\theta^t_1,\cdots,\theta^t_U,\ v^t_1,\cdots,v^t_U,h^t_1,\cdots,h^t_U,\\&w^t_{1,1},\cdots,w^t_{1,F},\cdots,w^t_{E,1},\cdots,w^t_{E,F},\\&l^t_{1,1},\cdots,l^t_{1,F},\cdots,l^t_{U,1},\cdots,l^t_{U,F}\}\end{aligned}\tag{18}$$

式中，θ^t_u表示 UVA 在t时隙内空中飞行的角度范围，$\theta^t_u\in[0,\ 2\pi]$；v^t_u表示 UAV 在时隙t空中飞行的速度，$v^t_u\in[0,v_{\max}]$；$w^t_{e,f}$表示 ES 服务器在时隙t中对内容文件f进行缓存替换；$l^t_{u,f}$表示 UAV 服务器在时隙t中对内容文件f进行缓存替换。

3.1.3　奖励函数

在单个时隙内 Agent 根据当前观测到的系统状态做出动作后，返回的瞬时奖励为：

$$r_t=-\mu T^f_{\mathrm{total}}-\eta P^f_{e,x}\tag{19}$$

$$P_{e,x}=1-P_{\mathrm{hit}}\tag{20}$$

式中，T^f_{total}表示在时隙t时智能汽车c进行请求缓存文件f的总时延，其作用是尽可能降低缓存模型所产生的时延；$P^f_{e,x}$表示在时隙t时刻内智能汽车进行请求缓存文件f的命中失败率，其作用是尽可能提升缓存文件的命中率；μ和η分别为请求缓存文件成功时延和缓存命中的权重。深度强化学习中的奖励函数可以引导 Agent 进行更好地学习训练，进而做出有利于降低智能汽车处理任务能耗和时延以及提高负载均衡的动作。

3.2　SAC 算法

SAC 算法是一种深度强化学习算法，该算法包含了环境（Environment）、智能体（Agent）、Actor 网络、Critic 网络和对应的两个 Target 网络，其框架如图 2 所示。SAC 算法在进行模型训练的过程中使用了缓冲经验回放机制，缓冲经验回放区负责保存并利用以往的训练样本(s^t,a^t,r^t,s^{t+1})，之后会对缓冲区的样本进行采样来降低彼此的关联性。

本文主要的优化目标是空天地异构网络中的缓存策略，由于缓存策略与无人机移动轨迹、处理时延、缓存命中率具有高度耦合性，所以需要使用能够在连续空间中处理多个目标的优化算法。SAC 的熵正则化（entropy regularization）能够在优化回报的同时保持策略的探索性，此算法通过策略熵的加权和以及最大化期望回报，保证智能体选择最优化动作的同时保持一定的探索。由于在移动边缘计算中的任务需求是动态变化的，SAC 算法相较于其他强化学习算法更能解决高频环境中的缓存策略优化问题，如 DQN 算法虽然在离散动作空间中表现较好，但在保持探索性与充分利用环境信息之间的平衡不如 SAC 算法，而 PPO 虽然适用于连续动作空间，但其探索性和策略的稳定性无法与 SAC 相提并论。

1）计算在当前状态s^t下的 TD-error：

$$\delta^t=r^t+\gamma Q_\pi(s^{t+1},a^{t+1})-Q_\pi(s^t,a^t)\tag{21}$$

式中，$Q_\pi(s^t,a^t)$表示智能体在状态s^t下使用策略π执行动作a^t后得到的预期回报奖励期望值；

$Q_\pi(s^{t+1},a^{t+1})$表示智能体在状态转移后的下一个状态中使用策略π执行动作a^{t+1}后得到的预期回报奖励期望值；γ为折扣系数，它表示之后所获得的奖励对当前状态的重要性；r^t表示 agent 在当前状态s^t下根据策略π选择动作a^t后得到的瞬时回报。

2）Actor 网络根据环境状态给出每个智能体可能执行动作的概率，智能体根据概率分布图确定一个动作并与环境进行交互。Actor 网络通过不断地模拟训练来逐渐逼近最优决策，在不同环境中给出最合适的指导动作，同时还可以通过分布式策略优化实现多智能体协作或竞争目标。在训练阶段，主要是通过全局 Critic 网络计算的优势函数评估动作的长期收益来反向传递优化 Actor 网络。

Actor 网络的优化过程是利用最大化剪切替代目标更新策略参数，即计算新旧策略的概率比值，使用优势值估算策略梯度，并通过剪切约束和熵正则化确保训练的稳定性，最终使用随机梯度下降迭代优化。

3）Critic 网络用于估计价值函数，在训练过程中不断拟合来对策略函数的选择进行评估，从而不断优化 Actor 网络。本文使用最小化损失函数更新 Critic 网络。

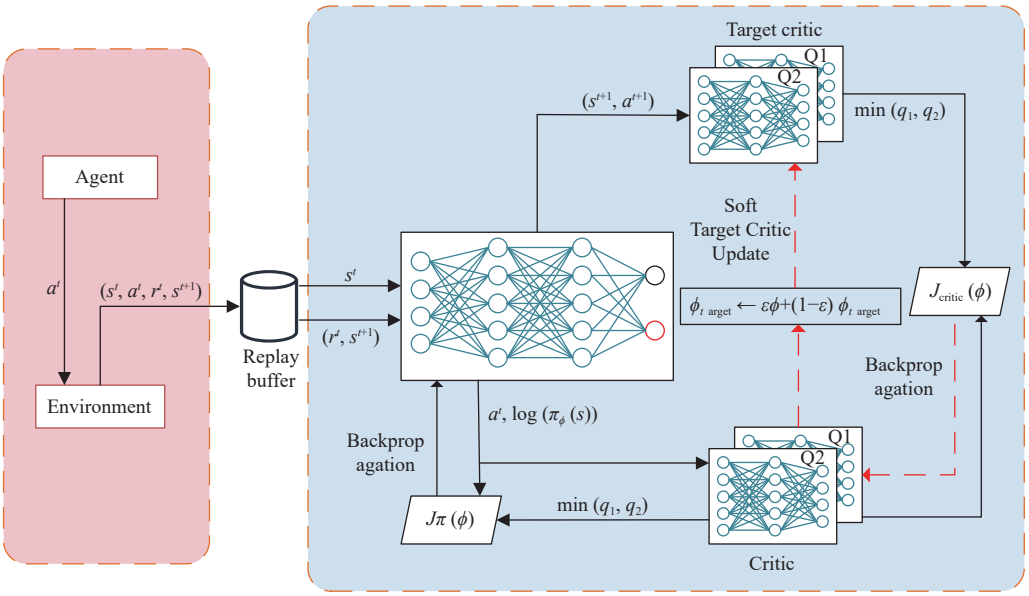


图 2 SAC 算法框架

3.3 基于 SAC 的缓存方案

为了保证训练的有效性，本文对智能体观察到的状态进行归一化预处理，并使用预处理后的数据进行 SAC 训练。

算法 1：SAC 缓存方案

初始化 ES 与 UAV 服务器数量E和 U，总训练 Episode 数为M，折扣因子为γ； Actor 网络、Critic 网络、Actor 目标网络、Critic 目标网络的参数；经验池为 buffer；Actor 网络学习率为lr_{Actor}；Criticr 网络学习率为$\text{lr}_{\text{Critic}}$

For Episode=1,2,⋯,M do

重置环境参数，初始化服务器状态，获取初始观测状态s^0

For i=1,2,⋯,N do

根据现有的策略π，状态s^t归一化成$\widehat{s^t}$，输入$\widehat{s^t}$，输出动作a^t

Agent 执行动作a_i，得到回报r_i和后继状态s_{i+1}，并计算 TD-error

下一时刻的状态s_{i+1}归一化成$\widehat{s_{i+1}}$

if 经验池存储空间剩余

把存储元组$(\widehat{s_i}, a_i, r_i, \widehat{s_{i+1}})$放到经验池中

else

将其置换到经验回放池

end if

从经验池中，根据采样概率P_k抽取样本$(\widehat{s_i}, a_i, r_i, \widehat{s_{i+1}})$

计算预测Q值

使用最小化损失函数反向传播更新 Critic 网络参数

使用 clip 函数更新 Actor 网络

软更新 Actor 网络和 Critic 网络

end for

end for

4　实验及分析

4.1　实验场景

仿真实验设备为 64 位的 Windows 10 系统，搭配 32 GB RAM、Intel i10 CPU 以及 GTX3060-Ti GPU。仿真实验使用的编程工具为 Python 3.12，神经网络的训练使用开源的 PyTorch 框架。

为了对后续的算法分析与结果讨论提供统一的数值基础，需将理论模型与仿真实验进行对接，通过对典型研究场景和参考文献的分析，特将系统参数具体化，并为通信、缓存等模块设置合理的功率、带宽与容量值。模拟环境设置终端用户设备、ES 服务器、UAV 服务器分别为 50 台、3 台、7 台[19-20]。传输功率、信道带宽、缓存存储器大小等数据参考表 1。为了保证终端的请求数据更合理，本文模拟实验中的请求序列遵循 Zipf 分布[21]。

表 1　实验参数设置

参数	含义	值
B/MHz	带宽	20
T	系统周期	320
N	时隙数目	40
t_{fly}/s	UAV飞行时间	1
$v_{\max}$/m·s^{-1}	UAV最大速度	20
f_u/GHz	UAV算力	20
buffer	经验池大小	5 000
batch	采样数量	128
β_0/dB	单位信道增益	−50
E_{UAV}/kJ	UAV初始电量	500
M_{UAV}/kg	UAV重量	50
$p_{c,e}$/W	ES传输功率	10
$p_{c,u}$/W	UAV传输功率	10
K/Mbit	缓存容量	160
$p_{c,s}$/W	LEO传输功率	50
H_{LEO}/km	LEO高度	790

4.2　SAC 参数调优实验

样组大小对 SAC 算法性能的影响如图 3 所示。

由图 3 可看出，样组为 64、128、256 时方案均能正常收敛；样组为 64 和 256 时其收敛程度均弱于取样组为 128 的表现。样组为 64 时波动较大，样组为 256 时收敛效果相较于 128 时收敛缓慢且收敛程度较差。本文认为这是因为在不同大小的经验池下，Agent 周期性地采集数据的情况不同。当取样组大小设置较小时，经验样本学习的内容较小，此时 Agent 采样的数据波动性比较大。当取样组设置较大时智能体学习内容的清澈度较低，由于取样组较大，会导致某些无关样本频繁以及相关性较小的样本被采样；因此，为使 SAC 缓存方案的收敛性能表现最佳，本文将取样组设为 128。

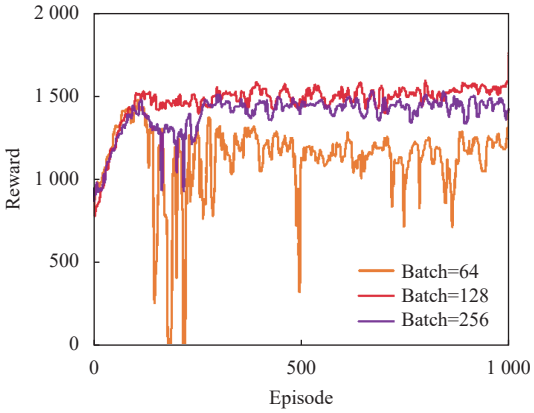


图 3　样组大小对 SAC 性能的影响

学习率变化对 SAC 性能的影响如图 4 所示。

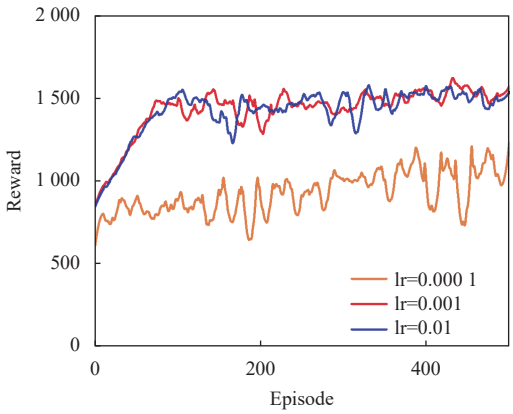


图 4　学习率变化对 SAC 性能的影响

由图 4 可看出，SAC 缓存方案在 lr = 0.01，0.001 时，最终结果都能成功收敛，而lr = 0.000 1时不能正常收敛；具体分析可以看出，lr = 0.001相较与lr = 0.01时，SAC 方案的收敛效果更好，波动性更小。当lr = 0.000 1时，本方案收敛较缓慢甚至不收敛。本文认为训练参数学习率设置大小会影响训练效果，过高的学习率会使训练更新步长过大，导致训练过程陷入局部最优，所以智能体无法探索新的更优方案。而学习率设置过低会使参数更新步长过小，从而迭代更多的次数才能达到收敛，甚至无法找到最优解，从而无法正常收敛；因此，为使 SAC 缓存方案的收敛性能表现最佳，本文设置 lr = 0.001。

4.3　SAC 方案性能的对比实验

为了进一步测试本文所提 SAC 方案的性能，

在相同的实验环境下对 SAC 缓存方案、PPO 缓存方案和 DQN 缓存方案在缓存时延、缓存命中率、Reward 指标上的表现进行对比实验。为保证实验结果的公平性，模拟预处理阶段针对 SAC、PPO 和 DQN 的关键超参数进行了适当调整，以确保它们能够在相同的实验设置下达到最理想的卸载效果；选择合适的样本大小和学习率，避免过高或过低的学习率和样本对收敛产生不利影响。同时为了保证有效公平对比，对所有方案的状态都进行了归一化处理。

3 种方案下迭代次数对 R 值的影响如图 5 所示。

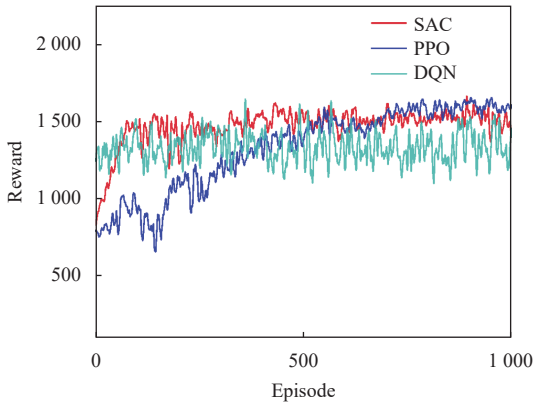


图 5 迭代次数对 R 值的影响

由图 5 可看出，SAC、PPO 方案最终都能成功收敛， 相较于 PPO 方案的收敛效果，SAC 方案收敛速度更快，且收敛效果更为稳定。DQN 方案的收敛波动较大，稳定性较差，难以收敛。本文认为可能是 SAC、PPO 均是使用 Actor-Critic 双重网络架构进行迭代训练，且是连续性动作，因此均能够正常收敛，但由于 PPO 采用的是单 Critic 网络，因此更新速度弱于 SAC。而 DQN 则是离散化的动作，针对连续化的场景表现更为吃力。因此得出：相较于其他缓存方案，SAC 缓存方案的性能更加符合对 SAGIN 结构的需求。在同样的实验环境下，缓存数量变化（缓存数量依次为 60、90、120）对命中率的影响如图 6 所示。

由图 6 可看出，随着缓存数量的增加，SAC、PPO 和 DQN 方案的缓存命中率均呈现不同程度的下降趋势，表明随着缓存数量的上升，用户请求任务命中的几率就会下降；本文挑选缓存数量在 90 时系统平均能耗的数据进行对比。可以看出，在相同的缓存数量下，SAC 的系统缓存命中率最高，分别比传统 PPO、DQN 的缓存方案提高 3.9%、39.6%。这可能是因为 SAC 缓存方案采用的是 Actor-Critic 双重网络架构，且 Critic 是双层网络模型，能够在规模更大的场景中分析用户的请求喜好，从而预缓存任务文件，进而提高整体的缓存命中率；而 PPO 由于网络结构的局限性以及 DQN 自身动作状态的限制，导致它们无法随着不断增大的环境状态进行有效训练；所以本文认为系统结构下 SAC 的缓存决策方案相较于其他方案能够更好地提高缓存命中率。

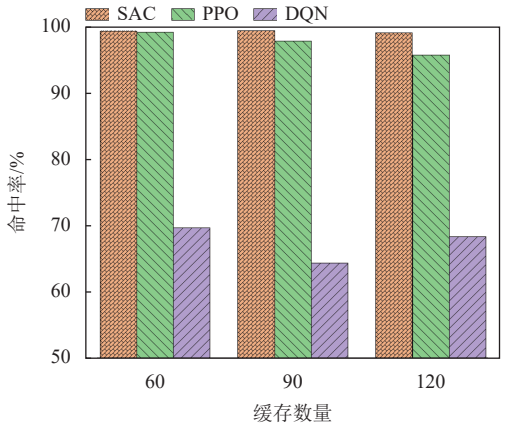


图 6 缓存数量变化对命中率的影响

在同样的实验环境下，缓存数量变化（缓存数量依次为 60、90、120）对时延的影响如图 7 所示。

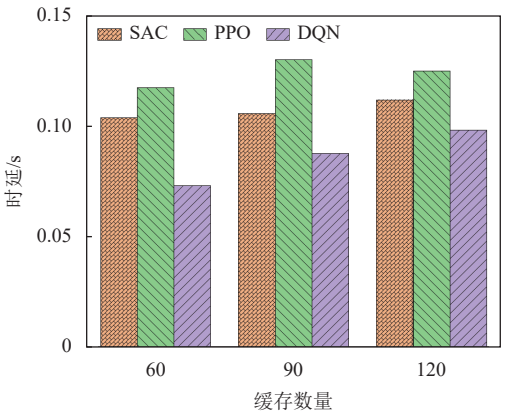


图 7 缓存数量变化对时延的影响

由图 7 可看出，随着智能汽车数量的增加，SAC、PPO、DQN 的平均时延均呈现上升的趋势，即随着缓存数量的上升，用户的缓存时延就会增加，缓存质量就会下降；本文挑选缓存数量在 90 时系统时延的结果进行对比。可以看出，在缓存数量为 90 的情况下，SAC 的系统时延较小，分别比 PPO 的缓存方案低 36.1%，比 DQN 的缓存方案提高 21.4%；本文认为 DQN 缓存方案相较于

SAC 缓存方案时延表现更好是因为 DQN 的文件命中率低，成功率较低导致任务完成量低，因此总体时延会相较于其他方案更低，但由于缓存命中率极低会导致终端用户任务无法完成，使用体验降低，无法在动态网络环境中进行实质性的应用；本文提出的系统结构下 SAC 的缓存方案相较于其他方案能够在提高缓存命中率的同时更好地降低系统平均时延。在同样的实验环境下，UAV 数量变化（UAV 数量依次为 4、6、8）对命中率的影响如图 8 所示。

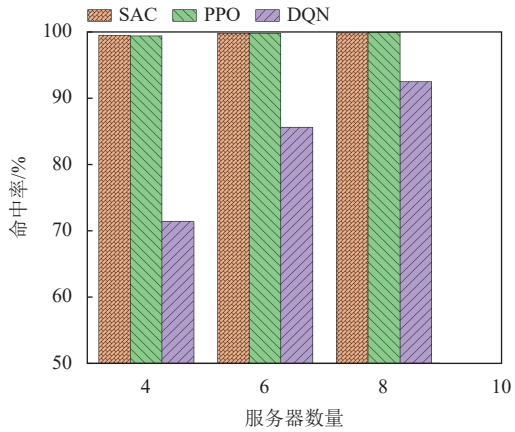


图 8　UAV 数量变化对命中率的影响

由图 8 可看出，随着服务器数量的增加，文件缓存命中率也呈现出不同的变化趋势，SAC 方案、PPO 方案和 DQN 方案命中率均随着服务器数量的增加而有所提高；本文挑选服务器数量在 8 时系统命中率的结果进行对比。可以看出，当服务器数量为 8 时，各个时间段的命中率表现接近，整体的命中率也表现最优。SAC 缓存方案相较其他缓存方案其系统命中率分别提高 2.7%、23.3%。这可能是因为随着服务器的增加，每个服务器服务的平均智能网联汽车数量在减少，即服务器会尽可能部署在所服务的智能网联汽车周边，同时每个智能汽车所分配到的平均文件存储空间也随之上升，从而导致任务缓存所产生的命中率提升；所以本文认为系统结构下 SAC 的缓存决策方案相较于其他方案能够更好地提高缓存命中率。在同样的实验环境下，UAV 数量变化（UAV 数量依次为 4、6、8）对时延的影响如图 9 所示。

由图 9 可看出，随着服务器数量的增加，文件缓存命中率也呈现出不同的变化趋势，SAC 方案、PPO 方案和 DQN 方案系统时延均随着服务器数量的增加而有所降低；本文挑选服务器数量在 8 时系统命中率的结果进行对比。可以看出，SAC 缓存方案相较其他 PPO 方案其系统时延低 5.3%、相较于 DQN 方案高 15.2%。本文认为 DQN 缓存方案相较于 SAC 缓存方案时延表现更好是因为 DQN 的文件命中率低，成功率较低导致任务完成量低，因此所产生的时延会相较于其他方案更低。但由于缓存命中率极低会导致终端用户任务无法完成，使用体验降低，无法在动态网络环境中进行实质性的应用。

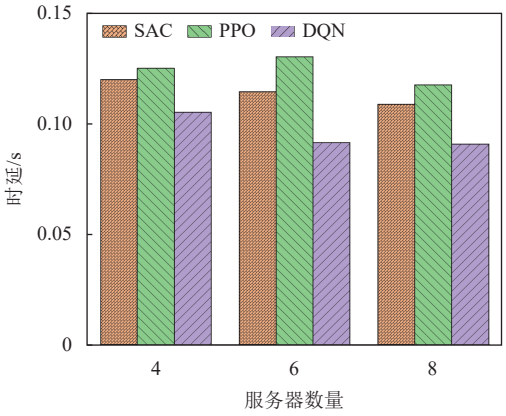


图 9　UAV 数量变化对时延的影响

5　结束语

本文针对 MEC 场景中的缓存时延和命中率优化问题构建了一种空天地异构网络架构，并提出一种 SAC 缓存优化方案。通过对场景的建模以及仿真实验模拟表明，该方案能够降低缓存时延以及提高缓存命中率，在 MEC 场景中具有较好的应用前景。然而在实际应用中 SAC 算法对海量数据进行计算时会占用大量的计算资源，并且在车载驶离当前 UAV 服务区域会造成卸载空白窗口，以上问题在落地应用时可能存在一定的影响。

针对方案可能存在的问题，未来将会从以下 3 个方面继续研究。1）结合实际通信数据对方案进行深入分析与验证，为方案的优化提供真实反馈。2）针对性优化 SAC 算法的计算问题，探索算法优化与资源节约的方法。3）发散式探讨场景深度定制的解决方案，对不同车联网应用场景进行定向优化以进一步提升方案的性能。

参 考 文 献

[1] MACH P, BECVAR Z. Mobile edge computing: A survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

[2] SONG Z Y, HAO Y Y, SUN X. Computation offloading and resource allocation algorithm for collaborative LEO satellite multi-access edge computing[J]. Acta Electronica Sinica, 2022, 50(3): 567-573.
[3] LI Z Y, WANG Q, CHEN Y F, et al. A survey on task offloading research in vehicular edge computing[J]. Chinese Journal of Computers, 2021, 44(5): 963-982.
[4] LIU Y, NI L Q, PENG M G. A secure and efficient authentication protocol for satellite-terrestrial networks[J]. IEEE Internet of Things Journal, 2022, 10(7): 5810-5822.
[5] NIU Z S, SHEN X M, ZHANG Q Y, et al. Space-air-ground integrated vehicular network for connected and automated vehicles: Challenges and solutions[J]. Intelligent and Converged Networks, 2020, 1(2): 142-169.
[6] JI B F, WANG Y A, SONG K, et al. A survey of computational intelligence for 6G: Key technologies, applications and trends[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 7145-7154.
[7] YANG L, PAN C S, KONG X G, et al. Review on 5G-satellite integrated network[J]. Journal on Communications, 2022, 43(4): 202-215.
[8] CUI H X, ZHANG J, GENG Y H, et al. Space-air-ground integrated network (SAGIN) for 6G: Requirements, architecture and challenges[J]. China Communications, 2022, 19(2): 90-108.
[9] TANG Q Q, LI B. Overview of mobile edge computing in space-air-ground integrated network[J]. Radio Communications Technology, 2021, 47(1): 25-35.
[10] ZHAO J, LI Q, GONG Y, et al. Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks[J]. IEEE Trans on Vehicular Technology, 2019, 68(8): 7944-7956.
[11] ZHOU F, HU R Q. Computation efficiency maximization in wirelesspowered mobile edge computing networks[J]. IEEE Trans on Wireless Communications, 2020, 19(5): 3170-3184.
[12] WANG Y, LANG P, TIAN D, et al. A game-based computation offloading method in vehicular multiaccess edge computing networks[J]. IEEE Internet of Things Journal, 2020, 7(6): 4987-4996.
[13] NING Z, DONG P, KONG X, et al. A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things[J]. IEEE Internet of Things Journal, 2019, 6(3): 4804-4814.
[14] ZHOU S, JADOON W. The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment[J]. Computer Networks, 2020, 178: 107334.
[15] BAHREINI T, BADRI H, GROSU D. Mechanisms for resource allocation and pricing in mobile edge computing systems[J]. IEEE Trans on Parallel and Distributed Systems, 2022, 33(3): 667-682.
[16] HU H, SONG W, WANG Q, et al. Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network[J]. IEEE Internet of Things Journal, 2022, 9(17): 15942-15956.
[17] SABUJ S R, ASIEDU D K P, LEE K J, et al. Delay optimization in mobile edge computing: Cognitive UAV-assisted eMBB and mMTC services[J]. IEEE Trans on Cognitive Communications and Networking, 2022, 8(2): 1019-1033.
[18] FENG J, RICHARD Y F, PEI Q, et al. Cooperative computation offloading and resource allocation for blockchain-enabled mobile-Edge computing: A deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(7): 6214-6228.
[19] LIU F, ZHANG Z, WANG Z, et al. ECC: Edge collaborative caching strategy for differentiated services load-balancing[J]. Computers, Materials and Continua, 2021, 69(2): 2045-2060.
[20] 刘亮, 荆腾祥, 段洁, 等. 空天地一体化网络中基于联邦深度强化学习的边缘协作缓存策[J]. 通信学报, 2025, 46(1): 93-107.
LIU L, JING T X, DUAN J, et al. Federated deep reinforcement learning-based edge collaborativecaching strategy in space-air-ground integrated network[J]. Journal on Communications, 2025, 46(1): 93-107.
[21] PENG K, NIE J, KUMAR N, et al. Joint optimization of service chain caching and task offloading in mobile edge computing[J]. Applied Soft Computing, 2021, 103: 107142.

责任编辑 税 红