动态车辆网络场景中的协同空地计算卸载和资源优化

王俊华[①] 罗 菲[①] 高广鑫*[②] 李 斌[③]

[①](南京航空航天大学计算机科学与技术学院 南京 211106)

[②](南京航空航天大学经济与管理学院 南京 211106)

[③](南京信息工程大学计算机学院 南京 210044)

摘 要：针对移动用户数量迅猛增长和地面基础设施分布稀疏所带来的挑战，该文提出一种能量收集辅助的空地协同计算卸载架构。该架构充分利用无人机(UAVs)的灵活机动性和路侧单元(RSUs)及基站(BS)的强大算力，实现了任务计算的动态实时分发。特别地，无人机通过能量收集来维持其持续运行和稳定的计算性能。考虑到无人机与地面车辆的高动态性、车辆计算任务的随机性，以及信道模型的时变性，提出一个能耗受限的长期优化问题，旨在从全局角度有效降低整个系统的平均时延。为了解决这一复杂的混合整数规划(MIP)问题，提出一种基于改进演员-评论家(Actor-Critic)强化学习算法的计算卸载策略(IACA)。该算法运用李雅普诺夫优化技术，将长期系统时延优化问题分解为一系列易于处理的帧级子问题。然后，利用遗传算法计算目标Q值替代目标神经网络输出以调整强化学习进化方向，有效避免了算法陷入局部最优，从而实现动态车辆网络中的高效卸载和资源优化。通过综合仿真验证了所提计算卸载架构和算法的可行性和优越性。

关键词：空地一体化车联网；能量收集；计算卸载；强化学习；遗传算法

中图分类号：TN929.5; TP18 **文献标识码**：A **文章编号**：1009-5896(2025)01-0102-14

DOI: 10.11999/JEIT240464

1 引言

移动边缘计算(Mobile Edge Computing, MEC)作为一种创新技术，已经逐步渗透到车联网(Internet of Vehicles, IoV)的广阔领域中[1]。通过在网络边缘部署计算资源，MEC能够实时、高效地处理来自车辆的各类数据，极大地降低了数据传输的延迟，提升了整体响应速度。然而，在密集建筑或复杂地形环境下，数据传输的效率和可靠性会受到显著影响[2]。无人机(Unmanned Aerial Vehicles, UAVs)因其灵活部署的特性，成为了改善地面网络设备连通性和构建低成本、高可靠性网络环境的关键工具[3]。当前研究在无人机辅助车辆协同计算领域取得了显著进展，涵盖了从理论建模到算法设计再到实际应用优化的多个方面[4]。Wang等人[5]聚焦于长期视角下的UAV辅助车辆协同计算问题，通过应用Lyapunov优化技术，有效解耦了UAV的长期能量约束，并设计了多种协同任务卸载方案，旨在优化无人机的轨迹规划、任务分配及通信资源管理，以最小化车辆的任务延迟或能耗。类似地，He等人[6]则构建了包含多目标优化的无人机辅助车联网协同计算框架，通过迭代算法结合负载均衡标准，实现了无人机选择、网络资源分配及计算任务的高效管理，进而优化了卸载比率和计算资源利用。此外，文献[7]聚焦于无人机辅助多车队移动边缘计算系统，通过设计基于2维路径跟踪模型和Frenet框架的车队控制器，深入建模了空地通信与车载计算的耦合特性，并提出了基于序列2次规划法的优化算法，以最大化系统加权全局能效。

在算法应用层面，已有研究采用了强化学习算法及其多种变体算法联合优化无人机轨迹、任务分配和通信资源，以最小化任务延迟或能耗，提升空地网络服务质量[8–17]。其中，Yan等人[10]提出基于交通态势感知的无人机最优飞行轨迹算法，云计算中心预测实时交通状况并周期性分配无人机至不同任务区域。基于预测结果，构建飞行轨迹优化问题以最小化无人机成本，并考虑飞行和转弯能耗。最后提出基于深度强化学习的能效自主部署策略，获得各任务区域无人机最优悬停位置。Liu等人[14]研究了无人机辅助的车载边缘计算网络中的联合混合缓存与替换方案。该方案综合考虑路侧单元(Rode-Side Unit, RSU)和UAV的内容缓存与服务缓存，采用深度Q网络优化无人机混合缓存数据选择和用户任务卸载策略减少任务延迟。Yan等人[15]采用了强化学习算法解决基于软件定义网络(Software

收稿日期：2024-06-11；改回日期：2024-12-18；网络出版：2024-12-23

*通信作者： 高广鑫 gxgao@nuaa.edu.cn

基金项目：国家自然科学基金(62002166)，国家社会科学基金(22BGL113)

Foundation Items: The National Natural Science Foundation of China (62002166), The National Social Science Fund of China (22BGL113)

Defined Network, SDN)的无人机辅助车联网计算卸载优化问题，该算法充分考虑了车辆与UAVs的移动性、动态网络环境以及信息时效性、能耗和租赁价格之间的复杂关系。Zhang等人[16]则提出了迭代优化算法，针对异构无人机辅助的车载边缘计算问题，通过结合传统优化技术如二分法、连续凸逼近算法(Successive Convex Approximation, SCA)或启发式算法优化资源分配及无人机位置，旨在最小化系统延迟或能耗成本。

此外，随着能量收集(Energy Harvesting, EH)技术的进步，UAVs能够通过收集环境中的能量实现可持续能源供应[18–22]。Firozjae等人[18]研究了地面用户与作为飞行基站的UAVs之间的高能效上行链路通信，利用EH技术从UAVs的上行用户信号中收集能量，为UAVs的长时间运行提供了可能。Dai等人[22]研究了基于能量收集的UAVs辅助车载任务卸载的能量与任务分配优化问题。然而，目前的研究尚未将EH辅助的UAVs作为调度中心，与路侧单元(Road Side Units, RSUs)和基站(Base Station, BS)协同进行计算任务的卸载迁移。然而，以上研究尚未考虑长期视角下的动态快速智能空地协同计算——即考虑到无人机与地面车辆的高动态性、系统能耗动态特性，以及车载计算任务的随机到达特性等综合因素，如何从长期优化角度实现异构网络中的智能化快速卸载决策和资源分配以提升整体系统服务效率。

鉴于此，本文创新性地提出一种能量收集辅助的空地协作计算卸载架构，本架构实现了UAVs, RSUs和BS之间的紧密集成，以协同高效处理车辆产生的动态任务队列。通过引入EH技术，UAVs能够捕捉并转化环境中的可再生能量，确保持续供电与稳定算力。此外，考虑到信道条件的时变性和节点的高移动性，本研究建模了混合整数规划(Mixed Integer Programming, MIP)问题，并通过低成本地迭代调整卸载决策和计算资源分配，以实现系统性能的最优化。以下概述本文的主要贡献：

(1) 本文建模了能量收集辅助的动态空地协同计算卸载问题，充分考虑无人机与地面车辆的高移动性、无线信道的随机衰落以及任务的动态到达等复杂因素，旨在满足多种动态约束及长期能耗约束的同时，寻求最优的联合卸载决策与资源分配策略，以最小化全局服务时延。

(2) 本文首次提出一种基于强化学习的计算卸载策略——改进演员-评论家算法(Improved Actor-Critic Algorithm, IACA)以解决上述混合整数规划问题。本算法首先利用李雅普诺夫优化技术将问题分解为帧级确定性优化问题，然后利用遗传算法计算目标Q值替代目标神经网络输出以调整强化学习进化方向，有效避免了算法陷入局部最优，提高了求解效率和全局最优性。

(3) 本文通过详细的仿真实验验证了所提方案在任务处理时延和计算复杂度方面的显著优势。与现有的深度学习算法相比，本文提出的方案在车联网环境下的实时计算服务中展现出了更高的效率、更强的鲁棒性和更优的整体性能。

2　系统模型

图1展示了所提出的能量收集辅助的空地协同计算卸载架构，其包括一个由$U=\{1,2,\cdots,|U|\}$个UAV构成的空中网络，$R=\{1,2,\cdots,|R|\}$个RSU、$V=\{1,2,\cdots,|V|\}$辆车及1个BS b组成的地面网络。UAV不仅具备通信和计算能力，还能作为中继将计算任务卸载到RSU或BS进行处理。RSU和BS作为地面固定基础设施，拥有稳定的能源供应。当地面车辆行驶进入某UAV的覆盖区域时，它向该UAV发出计算任务请求。UAV进一步选择在本地处理器上直接计算，或者作为中继将任务转发给合适的RSU或BS进行计算。本文将系统时间细分为T个等长的时隙，每个时隙由t表示，长度为$\mathcal{T}$。任意UAV u在时刻t收到的任务请求用2元组$\langle C_u^t, D_u^t\rangle$来表示，其中，$C_u^t$表示该计算任务所需CPU周期数，而$D_u^t$则代表任务输入数据大小。

本文采用3维笛卡尔坐标系建模。地面网络中所有的车辆、RSU和BS都位于0高度。用$\boldsymbol{p}_v^t=(x_v^t,y_v^t)$表示车辆$v(v\in V)$在时刻$t$的位置，$\boldsymbol{p}_r^t=(x_r^t,y_r^t)$表示RSU $r(r\in R)$在时刻t的位置，$\boldsymbol{p}_b^t=(x_b^t,y_b^t)$表示 BS b在时刻t的位置。假设无人机$u(u\in U)$以固定高度H飞行，其坐标可以表示为$\boldsymbol{p}_u^t=(x_u^t,y_u^t,H)$。此外，UAV和RSU, BS之间的通信速度都与无线信道增益有关。本文引入了$h_{u,r}^t$和$h_{u,0}^t$两个参数，分别表示在第t个时隙内，UAV u与RSU r, BS b之间的无线信道增益。信道增益在

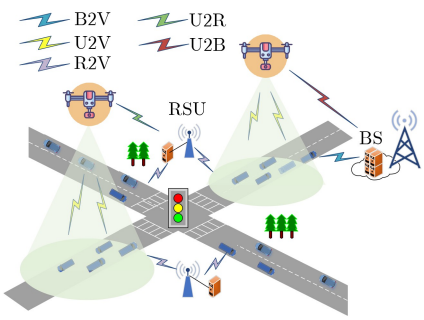


图 1 能量收集辅助动态车辆网络中的空地协同计算卸载

不同时隙间是独立且随机变化的，反映了无线通信环境的不确定性和动态性。

2.1 任务卸载模型

本文采用二进制变量表示任务的卸载决策。具体来说，定义任务卸载向量：$\boldsymbol{\alpha}^t = \{\alpha_{u,s}^t \mid u \in U, s \in S\}$表示UAV u是否卸载到服务器s上执行，其中$S = \{b\} \cup R = \{0, 1, \cdots, |R|\}$。本文用编号为“0”的服务器指代BS b。当$\forall s \in S, \exists \alpha_{u,s}^t = 0$时，表示UAV u在t时刻选择在其本地执行计算任务，即不进行任务卸载。相反，当$\alpha_{u,s}^t = 1$时，表示UAV u决定将任务卸载到RSU或BS上执行。此时，如果$s = 0$，则任务被卸载到BS上执行；若$s > 0$，则任务将被卸载到RSU上执行。鉴于UAV的计算频率和能量存储能力均有限，本文假定在每个时隙t内，UAV仅能接受并处理单个计算任务。

2.2 通信模型

由于实际通信环境中可能存在多种情况，如无人机和车辆之间的直接可视路径视距(Line of Sight, LoS)被障碍物遮挡时，通信信号将通过反射、衍射等非视距(Non Line of Sight, NLoS)路径进行传输。因此，本文假设UAV与车辆之间的无线信道包括LoS和NLoS部分[22]。车辆v和UAV u在时隙t的LoS信道概率$\varepsilon_{u,v}^t$表示为

$$\varepsilon_{u,v}^t = \frac{1}{1 + a\exp\left(-b\left(\theta - a\right)\right)} \tag{1}$$

其中，a和b均为环境相关参数。$\theta = \arctan\cdot(H/\|p_u^t - p_v^t\|)$表示UAV在为车辆提供服务时的仰角，其值随车辆和UAV的位置变化。因此，UAV与车辆之间的信道增益由式(2)给出

$$h_{u,v}^t = \frac{\tau_0\left(\varepsilon_{u,v}^t + \zeta\left(1 - \varepsilon_{u,v}^t\right)\right)}{\left(H^2 + \|\boldsymbol{p}_u^t - \boldsymbol{p}_v^t\|^2\right)} \tag{2}$$

其中，τ_0为参考距离1 m时的信道增益，ζ为NLoS信道的衰减因子。在此基础上，在时隙t将计算任务从车辆v卸载到UAV u，任务传输速率$R_{u,v}^t$可表示为

$$R_{u,v}^t = B_{u,v}^t \log_2\left(1 + \frac{h_{u,v}^t P_{u,v}^t}{\sigma_{u,v}^2}\right) \tag{3}$$

其中，$B_{u,v}^t$, $h_{u,v}^t$, $P_{u,v}^t$, $\sigma_{u,v}^2$分别表示在时隙t处车辆v和UAV u之间的信道带宽、信道增益、传输功率和噪声功率。相应地，车辆任务的传输时延表示为

$$T_{u,v}^t = \frac{D_u^t}{R_{u,v}^t} \tag{4}$$

其次，UAV u和RSU r之间的信道增益表示为

$$h_{u,r}^t = \vartheta \cdot A_{\mathrm{d}}\left(\frac{\mathrm{c}}{4\pi F_{u,r}\|\boldsymbol{p}_u^t - \boldsymbol{p}_r^t\|}\right)^{d_{\mathrm{e}}} \tag{5}$$

其中，ϑ表示UAV与RSU之间的路径损耗常数，A_{d}表示天线增益，$F_{u,r}$表示载波频率，d_{e}是路径损耗指数。因此，二者之间的数据传输速率$R_{u,r}^t$为

$$R_{u,r}^t = B_{u,r}^t \log_2\left(1 + \frac{h_{u,r}^t P_{u,r}^t}{\sigma_{u,r}^2}\right) \tag{6}$$

其中，$B_{u,r}^t, P_{u,r}^t$和$\sigma_{u,r}^2$分别表示在时隙t处UAV u和RSU r之间的信道带宽、传输功率及噪声功率。由此得到，UAV将任务卸载到RSU的传输时延为

$$T_{u,r}^t = \frac{D_u^t}{R_{u,r}^t} \tag{7}$$

最后，UAV和BS之间的信道增益表示为

$$h_{u,0}^t = g_0 \frac{d_0}{\left(\|\boldsymbol{p}_u^t - \boldsymbol{p}_0^t\|\right)^4} \tag{8}$$

其中，g_0表示UAV与BS之间的路径损耗常数。则二者之间的数据传输速率可以表示为

$$R_{u,0}^t = B_{u,0}^t \log_2\left(1 + \frac{h_{u,0}^t P_{u,0}^t}{\sigma_{u,0}^2}\right) \tag{9}$$

其中，$B_{u,0}^t$表示UAV u和BS b之间的信道带宽。$P_{u,0}^t$和$\sigma_{u,0}^2$分别为UAV u和BS b之间的传输功率和噪声功率。因而，UAV将任务卸载到RSU传输时延为

$$T_{u,0}^t = \frac{D_u^t}{R_{u,0}^t} \tag{10}$$

2.3 计算模型

2.3.1 UAV 本地计算模型

给定UAV u在时隙t时的CPU计算频率f_u^t，则UAV u的本地计算时延、及本地计算消耗能量为

$$T_{u,\mathrm{L}}^t = \frac{C_u^t}{f_u^t} \tag{11}$$

$$E_{u,\mathrm{L}}^t = \kappa_u (f_u^t)^2 C_u^t \tag{12}$$

其中，$\kappa_u > 0$表示能耗系数，它描述了UAV在计算过程中，其能耗与CPU计算频率之间的特定关系。

2.3.2 UAV 卸载模型

在本文的假设中，考虑到RSU和BS的计算能力和传输速率高于具有能量约束的UAV。此外，假设从RSU或BS返回给UAV的计算结果的数据量远小于UAV卸载给它们的原始数据量[23]。鉴于这些假设，当UAV收到无法处理的计算任务时，将会把这些任务卸载给地面的RSU或BS进行处理。为了量化这一过程，本文假设f_s^t表示地面服务器的计算频率，则其计算延迟可以表示为

$$T_{s,\mathrm{G}}^{t}=\frac{C_{u}^{t}}{f_{s}^{t}},s\in R\cup\{b\}\tag{13}$$

在UAV执行任务时，它需要悬停(Hover)以接收数据、执行本地计算或卸载任务到RSU和BS[22]。因此，悬停时长贯穿整个任务执行过程，包括车辆到UAV的任务传输时间、本地计算时间或者卸载任务并收到返回结果的时间。总悬停/总执行时间表示为

$$\begin{aligned}T_{u,\mathrm{H}}^{t}=T_{u,v}^{t}+\left(1-\sum_{s=0}^{|R|}\alpha_{u,s}^{t}\right)T_{u,\mathrm{L}}^{t}+\alpha_{u,r}^{t}(T_{u,r}^{t}+T_{r,\mathrm{G}}^{t})\\+\alpha_{u,0}^{t}(T_{u,0}^{t}+T_{0,\mathrm{G}}^{t}),r\in R\end{aligned}\tag{14}$$

悬停状态的能耗与悬停时间成正比[9,22]，即

$$E_{u,\mathrm{H}}^{t}=\psi_{u}^{t}T_{u,\mathrm{H}}^{t}\tag{15}$$

其中ψ_u^t为UAV在时隙t的每秒恒定能量消耗值。此外，在UAV以恒定速度沿固定轨迹飞行的过程中，会消耗推进能量，即飞行能耗。这一能耗通常与飞行速度、飞行距离、UAV重量及空气阻力等因素相关[5,22]。为了简化表达，其表示为

$$E_{u,\mathrm{F}}^{t}=\mu_{u}\left\|\boldsymbol{v}_{u}^{t}\right\|^{2}\tag{16}$$

其中，μ_u是与UAV重量相关的能耗系数，$\boldsymbol{v}_u^t$是UAV u在t时刻的飞行速度。

2.4 车辆移动模型

考虑到车辆的移动性和跟驰特性，在向UAV进行任务传输的过程中，车辆的行驶距离是动态变化的。为保障通信的稳定性和效率，车辆必须保持在UAV的有效通信范围内。这个通信范围通常被定义为以UAV为中心，半径为ℓ的圆形区域[24]。具体表示为

$$\left|\delta_{v}^{t}T_{u,v}^{t}\pm\sqrt{\left(x_{u}^{t}-x_{v}^{t}\right)^{2}}\right|\leq\ell\tag{17}$$

其中，δ_v^t为车辆 v在时隙t的行驶速度。为了确保车辆在直线道路上的行驶安全，本节遵循文献[25]中的假设，即在最坏情况下(当前方车辆突然减速至完全停止时)，后车与前车之间的最小距离间隙应不小于一个特定的安全间隙S_0。因此，可以推导出当前车辆的位置距离应满足

$$\left\|\boldsymbol{p}_{\mathrm{vl}}^{t}-\boldsymbol{p}_{v}^{t}\right\|\geq S_{0}+\delta_{v}^{t}\Delta t+\left(\left(\delta_{v}^{t}\right)^{2}-\left(\delta_{\mathrm{vl}}^{t}\right)^{2}\right)/2\delta_{\mathrm{b}}\tag{18}$$

其中，Δt是恒定的反应时间。p_{vl}^t, δ_{vl}^t和δ_{b}分别表示前车的位置、行驶速度和制动操作的减速度。公式的右侧由3部分组成：S_0表示基础安全间隙，确保即使在极端情况下，两车之间也能保持一定的安全距离；第2部分$\delta_v^t\Delta t$计算了在驾驶员的反应时间内，后车以当前速度δ_v^t行驶的距离；第3部分则计算了从驾驶员开始制动到车辆完全停止所需的距离。

2.5 UAV能量收集模型

电池续航时间的限制会显著影响UAV的任务卸载性能。作为一种解决方案，EH技术被应用于无人机辅助的任务卸载中，UAV通常通过收集无线射频能量来减轻其能量负担。然而，在实际场景中，由于路径损耗等因素，基于射频的EH可能会受到严重限制。因此，实际部署时可引入其他形式的可再生能源(如太阳能和风能)，作为补充手段以增强能量收集的效率与可靠性。根据文献[10,20,21]，本文在建模无人机能量收集过程时，将其抽象化为一系列连续到达的能量包，并假设能量包在每个时隙的到达服从独立且同分布。假设用e^t表示在时隙t开始时到达UAV的能量包。具体地

$$0\leq e_{u}^{t}\leq e_{\max}^{t},t\in\{1,2,\cdots,T\}\tag{19}$$

其中，$e_{\max}^t$表示 UAV在时隙t能够收集的最大可用能量包。假设UAV u在初始时隙被完全充电，其能量预算在整个任务执行期间被均匀分配。具体来说，每个时隙的能量预算可以表达为

$$\bar{E}_{u}=\frac{E_{\max}^{u}}{T}\tag{20}$$

其中，$E_{\max}^u$表示 UAV在完全充电状态下的能量阈值。在时隙t，UAV u的系统能耗由3部分组成：本地计算能耗$E_{u,\mathrm{L}}^t$、悬停能耗$E_{u,\mathrm{H}}^t$，以及飞行能耗$E_{u,\mathrm{F}}^t$。因此，系统能耗的表达式为

$$E_{u}^{t}=\left(1-\sum_{s=0}^{|R|}\alpha_{u,s}^{t}\right)E_{u,\mathrm{L}}^{t}+E_{u,\mathrm{H}}^{t}+E_{u,\mathrm{F}}^{t}\tag{21}$$

此外，为了确保UAV的能量可持续性，其长期能耗应满足约束为

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{E_{u}^{t}-e^{t}\right\}\leq\bar{E}_{u}\tag{22}$$

2.6 问题公式化

基于上述分析，可以得出在时隙t的系统总处理时延$T_{\mathrm{sys}}^t=\sum_{u=1}^{U}T_{u,\mathrm{H}}^t$。考虑到无线信道增益的时变性、以及无人机能量收集的动态性，本文提出对卸载策略$\boldsymbol{\alpha}^t=\left\{\alpha_{u,s}^t\mid u\in U,s\in S\right\}$以及计算资源分配$\boldsymbol{f}_u^t=\{f_u^t\mid u\in U\}$进行联合优化，以最小化系统的平均时延。这一优化问题可以表述为

$$
\left.\begin{aligned}
&\mathrm{P1}: \min_{\boldsymbol{\alpha}^t, \boldsymbol{f}^t} \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{T_{\mathrm{sys}}^t\right\}\\
&\text{s.t. C1}: 0 \le e_u^t \le e_{\max}^t\\
&\quad \mathrm{C2}: \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{E_u^t - e_u^t\right\} \le \bar{E}_u\\
&\quad \mathrm{C3}: \sum_{s=0}^{|R|}\alpha_{u,s}^t \le 1, \alpha_{u,s}^t \in \{0,1\}\\
&\quad \mathrm{C4}: \sum_{u=1}^{U}\left(1 - \sum_{s=0}^{|R|}\alpha_{u,s}^t\right)\cdot f_u^t \le f_u^{\max}\\
&\quad \mathrm{C5}: \left|\delta_v^t T_{u,v}^t \pm \sqrt{\left(x_u^t - x_v^t\right)^2}\right| \le \ell\\
&\quad \mathrm{C6}: \left\|\boldsymbol{p}_{\mathrm{vl}}^t - \boldsymbol{p}_v^t\right\| \ge S_0 + \delta_v^t \Delta t\\
&\qquad + \left(\left(\delta_v^t\right)^2 - \left(\delta_{\mathrm{vl}}^t\right)^2\right)/2\delta_{\mathrm{b}}\\
&\quad \mathrm{C7}: \forall v, \mathrm{vl} \in V, \forall u \in U, \forall s \in S,\\
&\qquad \forall t \in \{1,2,\cdots,T\}
\end{aligned}\right\} \tag{23}
$$

这里，约束C1表示由EH技术捕获的能量包具有最大值；约束C2对应长期能源消耗预算，确保UAV在执行任务过程中能够在预定的能耗限制内运行；约束C3确保每个任务在一个时间段内最多只能卸载到一个服务器；约束C4表明UAV进行本地计算时所消耗的计算资源不能超过最大可用计算容量，以保证任务能够在UAV本地得到有效处理；约束C5表示在任务传输过程中，车辆应保持行驶在UAV的覆盖范围内；约束C6表明后车与前车之间的距离间隙不能小于特定的安全间隙。

2.7 基于李雅普诺夫优化的问题解耦

李雅普诺夫优化理论的核心在于利用李雅普诺夫函数评估系统稳定性，通过最小化李雅普诺夫漂移(Lyapunov Drift)与性能惩罚项的加权和，在保障系统稳定性的同时优化系统性能[5]。其中，李雅普诺夫漂移是指在连续的时隙中李雅普诺夫函数值的变化量，它衡量了系统状态随时间的变化趋势；性能惩罚项是与系统性能目标相关的函数，它反映了当前系统状态与理想性能状态之间的差距。该理论能够将长期稳定性约束转化为每个时隙的决策优化问题。因此，本节采用李雅普诺夫优化理论来解决长期能量队列稳定性约束C2，将其解耦为逐帧(即单个时隙 t)的稳定性约束。首先，为每个UAV引入与之对应的虚拟能量队列$\{Y_u(t)\}_{u=1}^U$，这些队列是优化过程中的关键变量，反映出UAVs的能量状态。初始时，所有队列被设定为零，即$Y_u(1)=0$，随后按照以下规则进行更新，以确保系统能够动态地管理能量资源

$$
Y_u(t+1) = \max\left\{Y_u(t) + E_u^t - e_u^t - \bar{E}_u, 0\right\} \tag{24}
$$

为了量化并控制这些队列的稳定性，本文定义2次李雅普诺夫函数$L(\boldsymbol{Z}(t))$，该函数是队列状态向量$\boldsymbol{Z}(t)$的2次函数，能够反映系统偏离稳定状态的程度。同时，引入李雅普诺夫漂移函数$\Delta L(\boldsymbol{Z}(t))$，它衡量了在单个时隙内李雅普诺夫函数的变化量，是评估系统稳定性变化的关键指标

$$
\left.\begin{aligned}
&L(\boldsymbol{Z}(t)) = \frac{1}{2}\sum_{u=1}^{U}Y_u(t)^2\\
&\Delta L(\boldsymbol{Z}(t)) = \mathbb{E}\{L(\boldsymbol{Z}(t+1)) - L(\boldsymbol{Z}(t)) \mid \boldsymbol{Z}(t)\}
\end{aligned}\right\} \tag{25}
$$

为了在维持队列稳定性的同时最小化系统任务延迟，本文提出在每个时隙t内优化卸载决策和计算资源的策略。其核心思想是通过最小化漂移加惩罚函数$\Lambda(\boldsymbol{Z}(t))$来寻找最优策略。该函数考虑了李雅普诺夫漂移(即系统稳定性的变化)，同时通过引入惩罚权重V来平衡系统平均时延的重要性。这里的惩罚权重V是一个关键参数，它运行决策者在系统稳定性和任务延迟之间做出灵活的权衡。具体来说，V的值越大，系统越倾向于减少任务延迟而牺牲队列稳定性；反之亦然。

$$
\Lambda(\boldsymbol{Z}(t)) \triangleq \Delta L(\boldsymbol{Z}(t)) + V\cdot\sum_{u=1}^{U}\mathbb{E}\left\{T_u^t \mid \boldsymbol{Z}(t)\right\} \tag{26}
$$

为了推导$\Lambda(\boldsymbol{Z}(t))$的上界，即原问题P1在每个时隙的上界，本文首先通过不等式性质得到

$$
\begin{aligned}
Y_u(t+1)^2 \le\ & Y_u(t)^2 + 2Y_u(t)\left(E_u^t - e_u^t - \bar{E}_u\right)\\
&+ \left(E_u^t - e_u^t - \bar{E}_u\right)^2
\end{aligned} \tag{27}
$$

其次，通过对两边的U个队列求和，得到结果为

$$
\begin{aligned}
&\frac{1}{2}\sum_{u=1}^{U}Y_u(t+1)^2 - \frac{1}{2}\sum_{u=1}^{U}Y_u(t)^2\\
&\quad \le \frac{1}{2}\sum_{u=1}^{U}\left(E_u^t - e_u^t - \bar{E}_u\right)^2\\
&\qquad + \sum_{u=1}^{U}Y_u(t)\left(E_u^t - e_u^t - \bar{E}_u\right)
\end{aligned} \tag{28}
$$

接下来，通过对上述公式两边同时取期望，并结合式(26)，可以得到

$$
\Delta L(\boldsymbol{Z}(t)) \le \mathrm{I} + \sum_{u=1}^{U}Y_u(t)\cdot\mathbb{E}\left[E_u^t - e_u^t - \bar{E}_u \mid \boldsymbol{Z}(t)\right] \tag{29}
$$

其中，常数I代表了与队列状态无关的项，可表示为

$$
\begin{aligned}
&\frac{1}{2}\sum_{u=1}^{U}\mathbb{E}\left[\left(E_u^t - e_u^t - \bar{E}_u\right)^2\right]\\
&\quad \le \frac{1}{2}\sum_{u=1}^{U}\left[\left(E_u^{\max}\right)^2 + \left(\bar{E}_u\right)^2 + \left(e_u^t\right)^2\right] \triangleq \mathrm{I}
\end{aligned} \tag{30}
$$

最后，得到式(27)中漂移加惩罚表达式的上界

$$\mathrm{I}+\sum_{u=1}^{U}\left\{Y_u(t)\mathbb{E}\left[E_u^t-e_u^t-\bar{E}_u\mid\boldsymbol{Z}(t)\right]+V\cdot\mathbb{E}\left[T_{\mathrm{sys}}^t\mid\boldsymbol{Z}(t)\right]\right\}\tag{31}$$

这一上界表达式提供了对整体系统漂移和惩罚项的一个理论上界，也为后文进一步优化计算任务卸载策略提供了指导。通过将原始问题P1转化为一系列上界最小化子问题P2，本文能够采用更高效的算法来求解，从而实现更好的系统性能和资源利用率

$$\left.\begin{aligned}&\mathrm{P2}:\min_{\boldsymbol{\alpha}^t,\boldsymbol{f}^t}V\cdot T_{\mathrm{sys}}^t+Y_u(t)\cdot E_u^t\\&\text{s.t. C1,C3,C4,C5,C6,C7}\end{aligned}\right\}\tag{32}$$

3　基于改进Actor-Critic的计算卸载与资源优化算法

通过观察分析，可以发现问题P2是一个非凸的、非确定性多项式时间难(Nondeterministic Polynomial-hard, NP-hard)性质的混合整数规划(Mixed Integer Programming, MIP)问题，直接求解其最优方案是一项艰巨的任务。传统的优化方法在面对此类问题时，尤其是在需要迅速响应的信道相干持续时间内，往往难以找到最优或接近最优的卸载决策。鉴于此，本文提出基于改进的Actor-Critic强化学习算法IACA，旨在克服这些挑战并实现动态稳定的在线计算卸载。

3.1　MDP模型

结合本文的研究场景，本节旨在将问题P2转化为一个马尔可夫决策过程(Markov Decision Process, MDP)，以便更有效地优化网络中的平均时延。这个过程被细分为一系列的时间步，每个时间步中，系统根据当前状态进行决策，并相应地调整计算任务的调度和资源分配。在MDP的框架下，状态、动作和奖励的定义如下：

状态$\boldsymbol{S}_t$：在t时隙，系统状态$\boldsymbol{S}_t$全面反映了车载边缘计算网络的关键信息。状态主要包括时隙t的任务数据大小D_u^t，计算任务所需的CPU周期数C_u^t，信道增益状态$h_{u,s}^t$，以及虚拟能量队列$Y_u(t)$。因此，状态可以定义为$\boldsymbol{S}_t=\left\{\boldsymbol{S}_1^t,\boldsymbol{S}_2^t\cdots,\boldsymbol{S}_U^t\right\}$，其中$\boldsymbol{S}_u^t=\left\{D_u^t,C_u^t,h_{u,r}^t,h_{u,0}^t,Y_u(t)\right\}$。

动作$\boldsymbol{A}_t$：动作空间$\boldsymbol{A}_t$包含了UAV在t时隙对计算任务的所有可能操作。这些动作主要包括UAV对每个任务的卸载调度决策，以及UAV的计算频率分配。具体来说，$\alpha_{u,s}^t=1$时表示任务被UAV卸载到服务器 s计算。若$\sum_{s=0}^{|R|}\alpha_{u,s}^t=0$则表示任务在本地UAV u计算。f_u^t表示 UAV为任务分配的CPU频率。因此，t时隙的动作空间可以表示为$\boldsymbol{A}_t=\{\boldsymbol{\alpha}^t,\boldsymbol{f}^t\}=\left\{\alpha_{u,s}^t\mid u\in U,s\in S\right\}\cup\left\{f_u^t\mid u\in U\right\}$。

奖励R_t：借助李雅普诺夫优化理论，原始优化目标被解耦为一系列上界子问题，即最小化时延与能耗相关的函数值。因此，本文将奖励函数定义为与优化目标负相关，即$R_t=-\left\{V\cdot T_u^t+Y_u(t)\cdot E_u^t\right\}$。

3.2　IACA算法

为了解决上述决策问题，本文构建了两个深度神经网络 (Deep Neural Networks, DNNs)，即演员 (Actor) 和评论家 (Critic)，它们的参数分别用θ和ω表示。图2所示为IACA算法架构图。首先，Actor网络根据当前的环境状态$\boldsymbol{S}_t$来生成卸载策略π_θ。具体来说，它将状态$\boldsymbol{S}_t$作为输入，然后输出一个介于0～1的连续动作向量$\left\{\hat{\boldsymbol{\alpha}}^t\hat{\boldsymbol{f}}^t\right\}$，这些值分别代表了卸载决策和计算频率的松弛值。随后，这些连续动作会被量化为二进制值。之后，Critic网络负责评估Actor网络所生成的动作的质量。它根据当前的状态$\boldsymbol{S}_t$和量化后的动作$\boldsymbol{A}_t$来计算奖励R_t的近似值。对于Actor和Critic的详细描述如下。

演员Actor：它根据系统状态的集合作为输入来生成当前的计算卸载和计算频率决策$\hat{\boldsymbol{\alpha}}^t$, $\hat{\boldsymbol{f}}^t$。该网络的设计利用了通用近似定理的原理，即使用适当的激活函数并具备足够数量的隐藏神经元，神经网络就能够逼近任何连续函数。Actor网络采用了ReLU作为隐藏层的激活函数。ReLU函数通过简单的阈值操作$z=\max\{v,0\}$来定义神经元的输出。在输出层，为了确保输出的松弛动作值在(0,1)的范围内，本文采用了sigmoid激活函数，具体形式为$S=1/\left(1+\mathrm{e}^{-v}\right)$。接下来，为了将松弛卸载动作$\hat{\boldsymbol{\alpha}}^t$转化为实际的二进制卸载动作，本文采用了一种保序量化方法[26]。这种方法的核心思想是在量化过程中保持原始动作值之间的相对顺序不变，以确保决策的一致性。量化后的决策和卸载动作执行后，系统会给出奖励R_t并过渡到下一个状态$\boldsymbol{S}_{t+1}$。这些经验数据$\{\boldsymbol{S}_t,\boldsymbol{A}_t,R_t,\boldsymbol{S}_{t+1}\}$被存储在一个有限容量的经验回放缓冲池中，当达到容量上限时，旧数据将被新数据替换。

评论家Critic：它通过使用价值函数来估计给定状态下执行特定动作的长期奖励，从而指导Actor选择更优的动作。在本文中，经验回放技术被用来训练Actor和Critic网络，以迭代地改进策略π。

网络模型的训练包含如下两个步骤：

(1) Critic网络参数的更新。首先，智能体从经验回放缓冲池中随机选择多组数据$\{\boldsymbol{S}_t,\boldsymbol{A}_t,R_t,\boldsymbol{S}_{t+1}\}$作为该网络的输入。然后，智能体计算时间差分(Temporal Difference, TD)目标值λ_t，该值由当前奖励R_t和下一状态$\boldsymbol{S}_{t+1}$下的期望Q值组成，即

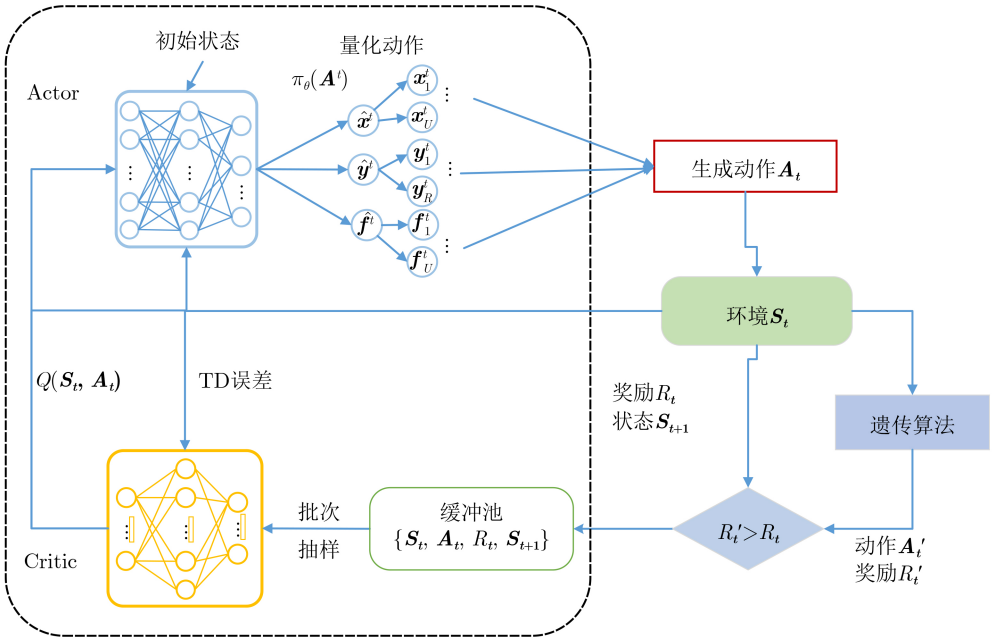


图 2 IACA算法架构图

$$\lambda_t = R_t + \gamma Q(\boldsymbol{S}_{t+1}, \boldsymbol{A}_{t+1}:\omega') \tag{33}$$

其中，γ为奖励折扣因子，ω'是Critic网络的目标网络参数。最后通过最小化Critic网络输出的当前状态值函数$Q(\boldsymbol{S}_t, \boldsymbol{A}_t:\omega)$与TD目标值$\lambda_t$的均方误差损失来更新该网络参数$\omega$，损失函数定义为

$$\text{Loss}(\omega) = \frac{1}{2}[Q(\boldsymbol{S}_t, \boldsymbol{A}_t:\omega) - \lambda_t]^2 \tag{34}$$

(2) Actor网络参数的更新。首先，Actor网络根据当前状态$\boldsymbol{S}_t$选择动作$\boldsymbol{A}_t$。接着，从Critic网络中得到该动作在当前状态下的Q值$Q(\boldsymbol{S}_t, \boldsymbol{A}_t)$。最后，并采用策略梯度的方法对Actor网络进行参数更新，以最大化期望奖励。损失梯度定义为

$$\text{Loss}(\theta) = \nabla_\theta \ln \pi_\theta(\boldsymbol{S}_t, \boldsymbol{A}_t)\, Q(\boldsymbol{S}_t, \boldsymbol{A}_t:\omega) \tag{35}$$

最后，为了增强强化学习算法的探索能力，避免在求解卸载决策时陷入局部最优，本工作引入了改进的遗传算法来辅助求解卸载决策$\hat{\boldsymbol{\alpha}}^t$。当遗传算法找到的卸载决策所获得的奖励$\bar{R}^t$优于Actor网络的生成决策所获得的奖励$R_t$时，这些更优的卸载决策将被用来替代Actor生成的决策，并加入到经验回放缓冲池中。遗传算法的实施步骤如下所述：

(1) 初始化种群：在初始化阶段，确定了种群规模N_{pop}、迭代次数item、交叉概率pc和变异概率pm。

(2) 计算种群的适应度：使用式(33)作为适应度函数计算每个个体的适应度值F_i。

(3) 选择：在该阶段，个体的选择是基于轮盘赌的方法，其中选择个体进行重组的概率与其适应值F_i 相关。轮盘赌的计算方法为

$$p_i = 1 - \frac{F_i}{\sum_{j=1}^{N_{\text{pop}}} F_j} \tag{36}$$

该选择策略使得那些适应度较高的个体更有机会参与到种群的重组过程中，提高种群的整体适应性。

(4) 交叉：该阶段将选择阶段中的个体进行组合，以期望产生更加多样化且具有更高适应度的后代个体，从而推动种群向着更优解的方向演进。

(5) 变异：基于概率选择，若个体进入变异阶段，则可以通过增强染色体基因突变的能力，更有效地向着更高适应度的方向转变，从而显著提高算法的局部搜索能力。

(6) 停止迭代：当达到设置的最大迭代次数或者种群中个体的适应度不再增加时，就停止迭代并返回当前最优结果。

当采取该算法生成的动作时，最大回报被计算为状态动作的值(即$\varphi(\boldsymbol{S}_t, \bar{\boldsymbol{A}}_t) = R'^t + Q(\boldsymbol{S}_{t+1}, \bar{\boldsymbol{A}}_{t+1}:\omega')$)。然后，使用它来替换预测的$Q$值(即$\lambda_t$)以更新网络参数，其表示为

$$\left.\begin{aligned}\text{Loss}(\omega) &= \frac{1}{2}\left[Q(\boldsymbol{S}_t, \bar{\boldsymbol{A}}_t:\omega) - \phi(\boldsymbol{S}_t, \bar{\boldsymbol{A}}_t)\right]^2 \\ \text{Loss}(\theta) &= \nabla_\theta \ln \pi_\theta(\boldsymbol{S}_t, \bar{\boldsymbol{A}}_t)\, Q(\boldsymbol{S}_t, \bar{\boldsymbol{A}}_t:\omega)\end{aligned}\right\} \tag{37}$$

综上所述，在联合优化卸载决策和计算资源分配时，采用本文所提出的算法框架，系统能够持续与环境互动，优化网络参数，直至达到理想的优化目标。算法1呈现了IACA算法的伪代码。

算法 1　基于改进Actor-Critic强化学习算法的计算卸载策略

输入：系统状态 $\boldsymbol{S}_t$，参数 V，奖励折扣因子 γ，Actor 网络结构，Critic 网络结构

输出：卸载决策$\hat{\boldsymbol{\alpha}}^t$，每个时间帧对应的最优计算频率分配$\hat{\boldsymbol{f}}^t$

(1) 初始化经验池, 网络模型参数以及系统环境参数;

(2) for episode $\leftarrow 1,2,\cdots$ do

(3)　获取当前环境系统初始状态 $\boldsymbol{S}_0$

(4)　Actor 生成一个0～1的松驰动作 $\hat{\alpha}_{u,s}^t, \hat{f}_u^t$;

(5)　将$\hat{\alpha}_{u,s}^t$和$\hat{f}_u^t$量化为二进制动作$\hat{\boldsymbol{\alpha}}^t$和满足约束条件的计算频率$\hat{\boldsymbol{f}}^t$，得到动作$\boldsymbol{A}_t$;

(6)　基于动作 $\boldsymbol{A}_t$ 得到下一个的状态 $\boldsymbol{S}_{t+1}$ 和当前奖励 R_t;

(7)　改进遗传算法生成卸载决策$\bar{\alpha}_{u,s}^t$,和奖励 R_t';

(8)　if $R_t' > R_t$ then

(9)　　$\boldsymbol{A}_t = \{\bar{\alpha}_{u,s}^t, f_u^t\}$

(10)　　$R_t = R_t'$

(11)　将 $\{\boldsymbol{S}_t, \boldsymbol{A}_t, R_t, \boldsymbol{S}_{t+1}\}$ 存储至缓冲池中;

(12)　for Agent do

(13)　　从经验池中随机采样批量数据 $\{\boldsymbol{S}_t, \boldsymbol{A}_t, R_t, \boldsymbol{S}_{t+1}\}$;

(14)　　通过 $\lambda_t = R_t + \gamma Q\left(\boldsymbol{S}_{t+1}, \boldsymbol{A}_{t+1} : \omega'\right)$ 计算 TD 目标值;

(15)　　计算损失值 $\mathrm{Loss}\,(\omega) = \dfrac{1}{2}[Q\,(\boldsymbol{S}_t, \boldsymbol{A}_t : \omega) - \lambda_t]^2$，更新 Critic 网络;

(16)　　计算损失值 $\mathrm{Loss}\,(\theta) = \nabla_\theta \ln \pi_\theta\,(\boldsymbol{S}_t, \boldsymbol{A}_t)\,Q\,(\boldsymbol{S}_t, \boldsymbol{A}_t : \omega)$ ，采用策略梯度更新 Actor 网络;

(17) for $t = 1,2,\cdots,T$ do

(18)　获取时隙t 的环境状态;

(19)　利用训练好的 Actor-Critic 模型，得到时隙t的最优卸载决策$\hat{\boldsymbol{\alpha}}^t$和计算频率$\hat{\boldsymbol{f}}^t$;

4　性能仿真与分析

4.1　仿真设置

本节使用Python3.11.7和PyTorch1.2.1实现了所提算法，并构建了仿真模型来评估其性能。仿真场景设定在一个20×20 km^{2}的区域内，该区域内的UAV, RSU以及车辆的数量均可根据具体的实验需求进行灵活配置。车辆行驶模型参考Green shield模型[27]，假设在不间断的交通流条件下，车辆速度与密度呈线性关系，即$\delta_v^t = u_0\,(1 - k/k_j)$，其中$u_0$表示自由流速度，$k$表示密度，$k_j$表示交通堵塞密度。车辆行驶过程中应保持与前车的最小距离为2 m，且其反应时间Δt和制动操作减速度δ_{b}分别为1.1 s和1 m/s^2。UAV的飞行高度固定为50 m[19]，半径ℓ的覆盖区域为200 m[24]，飞行速度v_u^t为25 m/s[20]。车辆与UAV的信道模型环境参数a和b分别设置为4.88和4.2[28]。设定UAV的重量为9.65 kg，总能量预算为500 kJ。此外，信道功率增益τ_0和衰减因子ζ分别设置为–50 dB和0.2[20]。UAV的悬停功率为220 W，能量系数κ_u设定为10^{-28}[7]。UAV, RSU和BS的计算能力f_u^t, f_r^t, f_b^t分别分布在[3,5] GHz, [5,8] GHz, [9,10] GHz[22]。IACA算法的模型参数以及其他实验参数的取值详见表1。

表 1　实验参数表

参数	值	参数	值
UAV计算能效系数 κ_u	10^{-28}	UAV飞行速度 v_u^t	25 m/s
可用带宽$B_{u,v}$	3 MHz	可用带宽 $B_{u,r}$	1 MHz
可用带宽 $B_{u,0}$	2.5 MHz	奖励折扣因子 γ	0.95
模型训练优化器	AdamOptimizer	批处理数量	512
Actor 学习率	0.001	Critic 学习率	0.002
天线增益A_d	3	载波频率$F_{u,r}$	915 MHz
路径损耗g_0	–40 dB	参考距离 d_0	1 m

此外，为验证所提空地协同动态卸载框架的有效性，本研究不仅采用了CPLEX和CVX工具作为基准求解方法，还引入了随机算法(Random Algorithm, RA)、基于单DNN的强化学习算法(Reinforcement Learning Algorithm based on Single DNN, RLASD)和基于优势动作评论的强化学习算法(Reinforcement Learning algorithm based on Advantage Actor-Critic, RLA2C)作为比较算法。

(1) CPLEX：在CPLEX求解卸载决策后，MIP问题转化为非凸子问题。本文采用逐次凸近似方法将其转化为凸问题，并利用CVX技术求解，以确保高效解决。

(2) RLASD[26]：在该算法中，首先通过单个DNN网络制定卸载决策。随后，利用对数障碍函数方法将计算资源优化问题转化为无约束形式，最后通过牛顿法迭代求解，得出优化后的计算频率决策。

(3) RA：每个时隙，UAV随机选择本地执行或卸载至RSU/BS。若资源或能耗受限，则随机换服务器。确定卸载决策后，用对数障碍函数方法和牛顿法优化资源问题。

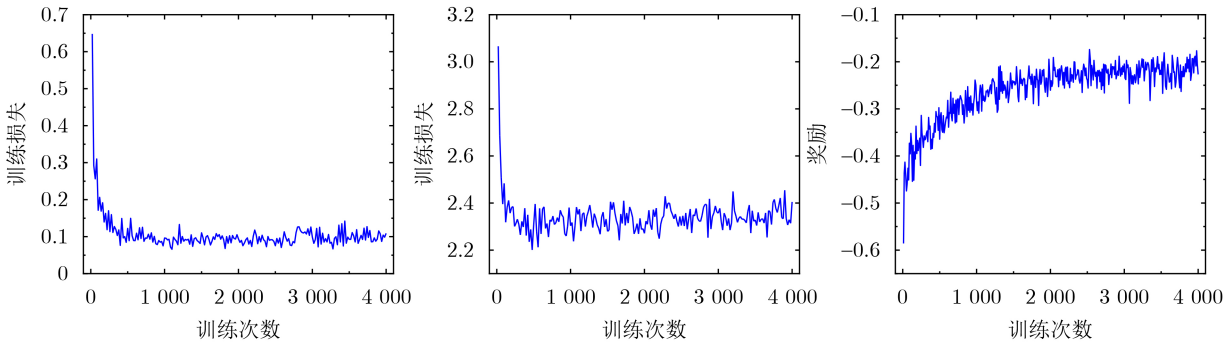
(4) RLA2C [29]：该算法运用基于优势动作评论的强化学习算法，求解本文提出的经李雅普诺夫优化技术分解的计算卸载和资源优化子问题，旨在最小化优化目标。

4.2 性能验证

图3展示了IACA算法框架中Actor和Critic网络的训练损失以及系统实现的奖励变化，验证了IACA算法的高效收敛性能。经过4000次训练回合，智能体通过不断与环境交互，优化了决策策略，并更新了网络参数。图3(a)和图3(b)显示，Actor和Critic网络的损失函数值随训练次数的增加而逐渐下降，这表明网络模型在逼近真实环境模型方面取得了进步。图3(c)则显示，奖励值随训练次数的增加而增大，并最终趋于稳定，表明智能体找到了更高效的决策策略。

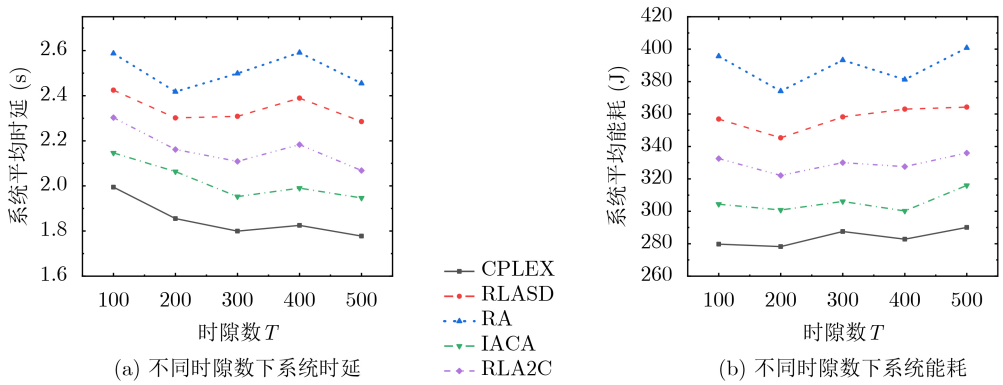
图4(a)和图4(b)揭示了系统平均时延与能耗随时隙数变化的趋势。随着时隙数递增，除RA算法外，各算法的系统平均延迟逐渐降低并趋于平稳。在此过程中，RA算法因随机卸载决策而表现出最高延迟。相比之下，RLA2C算法凭借引入的优势函数，在高效学习与训练中超越了RLASD算法的表现。而IACA算法，通过强化学习模型在动态环境中的反复训练，实现了接近CPLEX最优解的平均服务延迟。此外，IACA算法还通过最小化长期能耗约束下的李雅普诺夫漂移加惩罚函数上界，大幅度降低了系统平均能耗，在与RA和RLASD算法的对比中凸显出显著优越性。

图5(a)和图5(b)展示了任务输入数据大小对系统性能的影响。随着传输数据量从750 kbit增长至1 000 kbit，系统平均延迟与能耗均呈现出逐步攀升的趋势。本文提出的IACA算法凭借其与环境的有效交互以及采用的改进遗传算法，能够稳健地生成接近理想状态的最优解，无论是能耗还是时延均表现出色。相比之下，RLASD与RLA2C算法在处理大型任务时，由于训练环境的不稳定性，导致它们与CPLEX解之间的差距逐渐拉大。而采用RA算法求解问题时，系统平均时延与能耗则呈现出较大的波动性。



(a) Actor损失函数收敛性图 (b) Critic损失函数收敛性图 (c) 奖励函数收敛性图

图 3 不同时隙数下的训练损失和奖励

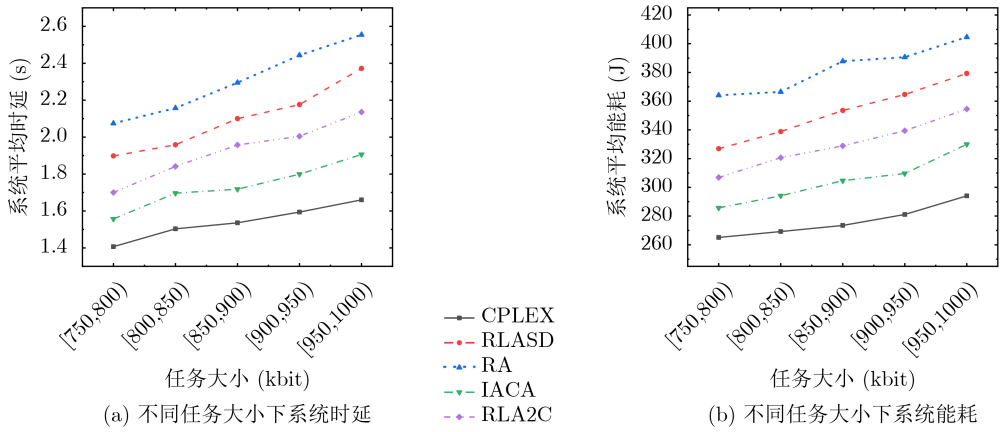


(a) 不同时隙数下系统时延 (b) 不同时隙数下系统能耗

图 4 不同时隙数对系统性能的影响

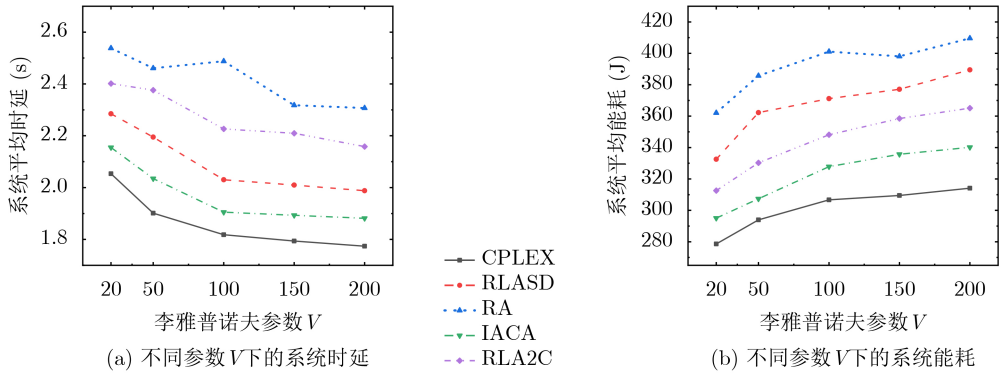
在图6(a)和图6(b)中展示了时隙数T为200时，李雅普诺夫控制参数惩罚权重V对系统平均时延和能耗的影响。通过引入V到李雅普诺夫漂移加惩罚函数中，本文实现了对系统性能的精细控制。随着V值的增加，四个算法的系统平均时延逐渐降低，而系统能耗则逐渐提高，且二者最终逐渐趋于稳定。所提出的IACA算法，由于在强化学习过程中采用了目标Q值改进机制，在系统时延和能耗优化方面具有更好的性能。

图7(a)和图7(b)揭示了UAV能耗阈值与数量对系统平均时延的影响。随着UAV能耗阈值的提升，其计算能力增强，本地计算效率提高，进而降低了系统总时延。提出的IACA算法有效避免了求解卸载决策时的局部最优问题，能灵活应对能耗阈值的变化，并通过稳定训练模型，实现了优于RLA2C, RLASD和RA算法的性能。另一方面，增加UAV数量初期能显著降低系统平均时延，但当数量过多且大数据量任务倾向于在UAV本地计算时，由于UAV计算能力相对RSU和BS较弱，系统时延会略有上升。此外，未结合遗传算法的强化学习算法RLASD和RLA2C，在稳定性和训练效果上均不及IACA算法。



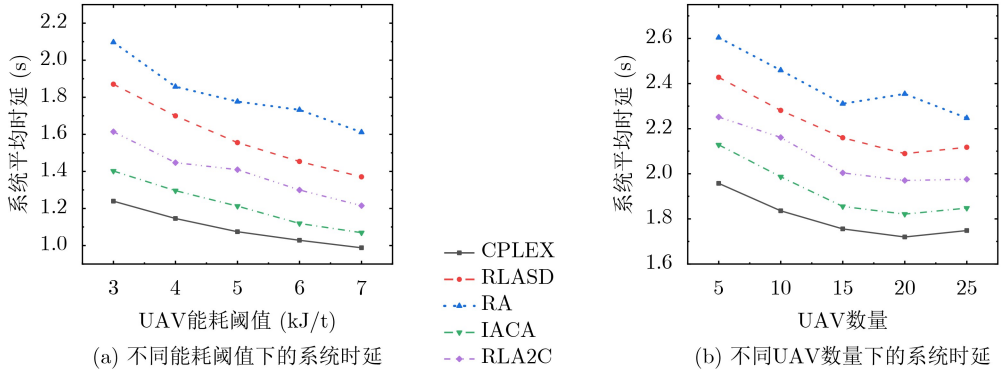
(a) 不同任务大小下系统时延　　(b) 不同任务大小下系统能耗

图 5 不同任务大小对系统性能的影响



(a) 不同参数V下的系统时延　　(b) 不同参数V下的系统能耗

图 6 不同李雅普诺夫控制参数V对系统性能的影响



(a) 不同能耗阈值下的系统时延　　(b) 不同UAV数量下的系统时延

图 7 不同UAV能耗阈值和数量对系统平均时延的影响

5 结论

本文深入研究了动态车辆网络中能量收集辅助的计算卸载和资源优化问题。其中，UAV作为任务调度中心与RSU和BS协同工作，为车辆提供计算服务，以降低系统平均计算时延。本文提出了IACA算法解决该复杂的MIP问题。首先，通过李雅普诺夫优化，将MIP问题分解为每一帧确定性子问题。然后，提出基于改进Actor-Critic强化学习算法的计算卸载算法来解决每一帧MIP问题。在迭代过程中，为扩大解空间且避免陷入局部最优解，本算法将改进遗传算法生成的高质量卸载决策加入到提供数据的缓冲池中。最后，通过全面的性能评估，验证了所提计算卸载架构和算法能够以较低的计算复杂度降低系统时延。

参 考 文 献

[1] ZHANG Haibo, LIU Xiangyu, XU Yongjun, *et al*. Partial offloading and resource allocation for MEC-assisted vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2024, 73(1): 1276–1288. doi: 10.1109/TVT.2023.3306939.

[2] LIU Qian, LIANG Hairong, LUO Rui, *et al*. Energy-efficiency computation offloading strategy in UAV aided V2X network with integrated sensing and communication[J]. *IEEE Open Journal of the Communications Society*, 2022, 3: 1337–1346. doi: 10.1109/OJCOMS.2022.3195703.

[3] YU Zhe, GONG Yanmin, GONG Shimin, *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. doi: 10.1109/JIOT.2020.2965898.

[4] HU Jinna, CHEN Chen, CAI Lin, *et al*. UAV-assisted vehicular edge computing for the 6G internet of vehicles: Architecture, intelligence, and challenges[J]. *IEEE Communications Standards Magazine*, 2021, 5(2): 12–18. doi: 10.1109/MCOMSTD.001.2000017.

[5] WANG Junhua, WANG Ling, ZHU Kun, *et al*. Lyapunov-based joint flight trajectory and computation offloading optimization for UAV-assisted vehicular networks[J]. *IEEE Internet of Things Journal*, 2024, 11(2): 22243–22256. doi: 10.1109/JIOT.2024.3382242.

[6] HE Yixin, ZHAI Daosen, ZHANG Ruonan, *et al*. A mobile edge computing framework for task offloading and resource allocation in UAV-assisted VANETs[C]. IEEE Conference on Computer Communications Workshop, Vancouver, Canada, 2021: 1–6. doi: 10.1109/INFOCOMWKSHPS51825.2021.9484643.

[7] DUAN Xuting, ZHOU Yukang, TIAN Daxin, *et al*. Weighted energy-efficiency maximization for a UAV-assisted multiplatoon mobile-edge computing system[J]. *IEEE Internet of Things Journal*, 2022, 9(19): 18208–18220. doi: 10.1109/JIOT.2022.3155608.

[8] ZHAO Nan, YE Zhiyang, PEI Yiyang, *et al*. Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing[J]. *IEEE Transactions on Wireless Communications*, 2022, 21(9): 6949–6960. doi: 10.1109/TWC.2022.3153316.

[9] ZHANG Liang, JABBARI B, and ANSARI N. Deep reinforcement learning driven UAV-assisted edge computing[J]. *IEEE Internet of Things Journal*, 2022, 9(24): 25449–25459. doi: 10.1109/JIOT.2022.3196842.

[10] YAN Ming, XIONG Rui, WANG Yan, *et al*. Edge computing task offloading optimization for a UAV-assisted internet of vehicles via deep reinforcement learning[J]. *IEEE Transactions on Vehicular Technology*, 2024, 73(4): 5647–5658. doi: 10.1109/TVT.2023.3331363.

[11] WU Zhiwei, YANG Zilin, YANG Chao, *et al*. Joint deployment and trajectory optimization in UAV-assisted vehicular edge computing networks[J]. *Journal of Communications and Networks*, 2022, 24(1): 47–58. doi: 10.23919/JCN.2021.000026.

[12] YANG Chao, LIU Baichuan, LI Haoyu, *et al*. Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks[J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(9): 9884–9895. doi: 10.1109/TVT.2022.3177664.

[13] PENG Haixia and SHEN Xuemin. Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks[J]. *IEEE Journal on Selected Areas in Communications*, 2021, 39(1): 131–141. doi: 10.1109/JSAC.2020.3036962.

[14] LIU Yinan, YANG Chao, CHEN Xin, *et al*. Joint hybrid caching and replacement scheme for UAV-assisted vehicular edge computing networks[J]. *IEEE Transactions on Intelligent Vehicles*, 2024, 9(1): 866–878. doi: 10.1109/TIV.2023.3323217.

[15] YAN Junjie, ZHAO Xiaohui, and LI Zan. Deep-reinforcement-learning-based computation offloading in UAV-assisted vehicular edge computing networks[J]. *IEEE Internet of Things Journal*, 2024, 11(11): 19882–19897. doi: 10.1109/JIOT.2024.3370553.

[16] ZHANG Wenqian, LÜ Zilong, GE Mengxia, *et al*. UAV-assisted vehicular edge computing system: Min-max fair offloading and position optimization[J]. *IEEE Transactions on Consumer Electronics*, 2024. doi: 10.1109/TCE.2024.3426513.

[17] HAN Zihao, ZHOU Ting, XU Tianheng, *et al*. Joint user

association and deployment optimization for delay-minimized UAV-aided MEC networks[J]. *IEEE Wireless Communications Letters*, 2023, 12(10): 1791–1795. doi: 10.1109/LWC.2023.3294749.

[18] FIROZJAE H M, MOGHADDAM J Z, and ARDEBILIPOUR M. A joint trajectory and energy harvesting method for an UAV enabled disaster response network[C]. 13th International Conference on Information and Knowledge Technology, Karaj, Iran, 2022: 1–5. doi: 10.1109/IKT57960.2022.10039000.

[19] ZHANG Ning, LIU Juan, XIE Lingfu, *et al*. A deep reinforcement learning approach to energy-harvesting UAV-aided data collection[C]. 2020 International Conference on Wireless Communications and Signal Processing, Nanjing, China, 2020: 93–98. doi: 10.1109/WCSP49889.2020.9299806.

[20] YANG Zheyuan, BI Suzhi, and ZHANG Y J A. Dynamic offloading and trajectory control for UAV-enabled mobile edge computing system with energy harvesting devices[J]. *IEEE Transactions on Wireless Communications*, 2022, 21(12): 10515–10528. doi: 10.1109/TWC.2022.3184953.

[21] CHANG Zheng, LIU Liqing, GUO Xijuan, *et al*. Dynamic resource allocation and computation offloading for IoT fog computing system[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(5): 3348–3357. doi: 10.1109/TII.2020.2978946.

[22] DAI Xingxia, XIAO Zhu, JIANG Hongbo, *et al*. UAV-assisted task offloading in vehicular edge computing networks[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(4): 2520–2534. doi: 10.1109/TMC.2023.3259394.

[23] WANG Feng, XU Jie, WANG Xin, *et al*. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[J]. *IEEE Transactions on Wireless Communications*, 2018, 17(3): 1784–1797. doi: 10.1109/TWC.2017.2785305.

[24] YUE Yuchen and WANG Junhua. Lyapunov-based dynamic computation offloading optimization in heterogeneous vehicular networks[C]. 2022 IEEE International Symposium on Product Compliance Engineering - Asia, Guangzhou, China, 2022: 1–6. doi: 10.1109/ISPCE-ASIA57917.2022.9971076.

[25] TREIBER M and KESTING A. Traffic Flow Dynamics: Data, Models and Simulation[M]. Berlin: Springer, 2013. doi: 10.1007/978-3-642-32460-4.

[26] HUANG Liang, BI Suzhi, and ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. *IEEE Transactions on Mobile Computing*, 2020, 19(11): 2581–2593. doi: 10.1109/TMC.2019.2928811.

[27] DAGANZO C F. Traffic flow theory[M]. DAGANZO C F. Fundamentals of Transportation and Traffic Operations. Oxford: Pergamon, 1997: 66–160. doi: 10.1108/9780585475301-004.

[28] SHI Weisen, LI Junling, CHENG Nan, *et al*. Multi-drone 3-D trajectory planning and scheduling in drone-assisted radio access networks[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 8145–8158. doi: 10.1109/TVT.2019.2925629.

[29] SUN Yukun and ZHANG Xing. A2C learning for tasks segmentation with cooperative computing in edge computing networks[C]. 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022: 2236–2241. doi: 10.1109/GLOBECOM48099.2022.10000948.

王俊华：女，副教授，研究方向为移动边缘计算、人工智能等.
罗　菲：女，硕士生，研究方向为车联网、边缘缓存和计算卸载等.
高广鑫：男，副教授，研究方向为算力网络管理、智慧交通等.
李　斌：男，副教授，研究方向为边缘计算与资源优化，深度强化学习算法等.

责任编辑：余　蓉

Collaborative Air-Ground Computation Offloading and Resource Optimization in Dynamic Vehicular Network Scenarios

WANG Junhua[①]　LUO Fei[①]　GAO Guangxin[②]　LI Bin[③]

[①](*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106, *China*)

[②](*College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106, *China*)

[③](*School of Computer Science, Nanjing University of Information Science and Technology, Nanjing* 210044, *China*)

Abstract:

Objective In response to the rapid growth of mobile users and the limited distribution of ground infrastructure, this research addresses the challenges faced by vehicular networks. It emphasizes the need for efficient computation offloading and resource optimization, highlighting the role of Unmanned Aerial Vehicles (UAVs), RoadSide Units (RSUs), and Base Stations (BSs) in enhancing overall system performance.

Methods This paper presents an innovative research methodology that proposes an energy harvesting-assisted air-ground cooperative computation offloading architecture. This architecture integrates UAVs, RSUs, and BSs to effectively manage the dynamic task queues generated by vehicles. By incorporating Energy Harvesting (EH) technology, UAVs can capture and convert ambient renewable energy, ensuring a continuous power supply and stable computing capabilities. To address the challenges associated with time-varying channel conditions and high mobility of nodes, a Mixed Integer Programming (MIP) problem is formulated. An iterative process is used to adjust offloading decisions and computing resource allocations at low cost, aiming to optimize overall system performance. The approach is outlined as follows:

Firstly, an innovative framework for energy harvesting-assisted air-ground cooperative computation offloading is introduced. This framework enables the collaborative management of dynamic task queues generated by vehicles through the integration of UAVs, RSUs, and BSs. The inclusion of EH technology ensures that UAVs maintain a continuous power supply and stable computing capabilities, addressing limitations due to finite energy resources.

Secondly, to address system complexities—such as time-varying channel conditions, high node mobility, and dynamic task arrivals—an MIP problem is formulated. The objective is to optimize system performance by determining effective joint offloading decisions and resource allocation strategies, minimizing global service delays while meeting various dynamic and long-term energy constraints.

Thirdly, an Improved Actor-Critic Algorithm (IACA), based on reinforcement learning principles, is introduced to solve the formulated MIP problem. This algorithm utilizes Lyapunov optimization to decompose the problem into frame-level deterministic optimizations, thereby enhancing its manageability. Additionally, a genetic algorithm is employed to compute target Q-values, which guides the reinforcement learning process and enhances both solution efficiency and global optimality. The IACA algorithm is implemented to iteratively refine offloading decisions and resource allocations, striving for optimized system performance.

Through the integration of these research methodologies, this paper makes significant contributions to the field of air-ground cooperative computation offloading by providing a novel framework and algorithm designed to address the challenges posed by limited energy resources, fluctuating channel conditions, and high node mobility.

Results and Discussions The effectiveness and efficiency of the proposed framework and algorithm are evaluated through extensive simulations. The results illustrate the capability of the proposed approach to achieve dynamic and efficient offloading and resource optimization within vehicular networks. The performance of the IACA algorithm is illustrated, emphasizing its efficient convergence. Over the course of 4 000 training episodes, the agent continuously interacted with the environment, refining its decision-making strategy and updating network parameters. As shown, the loss function values for both the Actor and Critic networks progressively decreased, indicating improvements in their ability to model the real-world environment. Meanwhile, a rising trend in reward values is observed as training episodes increase, ultimately stabilizing, which signifies that the agent has discovered a more effective decision-making strategy.

The average system delay and energy consumption relative to time slots are presented. As the number of slots increases, the average delay decreases for all algorithms except for RA, which remains the highest due to random offloading. RLA2C demonstrates superior performance over RLASD due to its advantage function. IACA, trained repeatedly in dynamic environments, achieves an average service delay that closely approximates CPLEX's optimal performance. Additionally, it significantly reduces average energy consumption by minimizing Lyapunov drift and penalties, outperforming both RA and RLASD.

The impact of task input data size on system performance is examined. As the data size increases from 750 kbit to 1 000 kbit, both average delay and energy consumption rise. The IACA algorithm, with its effective

interaction with the environment and enhanced genetic algorithm, consistently produces near-optimal solutions, demonstrating strong performance in both energy efficiency and delay management. In contrast, the performance gap between RLASD and RLA2C widens compared to CPLEX due to unstable training environments for larger tasks. RA leads to significant fluctuations in average delay and energy consumption.

The effect of the Lyapunov parameter V on average delay and energy consumption at T=200 is illustrated. With V, performance can be finely tuned; as V increases, average delay decreases while energy consumption rises, eventually stabilizing. The IACA algorithm, with its enhanced Q-values, effectively optimizes both delay and energy. Furthermore, the impact of UAV energy thresholds and counts on average system delay is demonstrated. IACA avoids local optima and adapts effectively to thresholds, outperforming RLA2C, RLASD, and RA. An increase in the number of UAVs initially reduces delay; however, an excess can lead to increased delay due to limited computing power.

Conclusions　The proposed EH-assisted collaborative air-ground computing offloading framework and IACA algorithm significantly improve the performance of vehicular networks by optimizing offloading decisions and resource allocations. Simulation results validate the effectiveness of the proposed methodology in reducing average delay, enhancing energy efficiency, and increasing system throughput. Future research could focus on integrating more advanced energy harvesting technologies and further refining the proposed algorithm to better address the complexities associated with large-scale vehicular networks. (While specific figures or tables are not referenced in this summary due to format constraints, the simulations conducted within the paper provide comprehensive quantitative results to support the findings discussed.)

Key words: Air-ground integrated vehicle networking; Energy Harvesting (EH); Computational offloading; Reinforcement learning; Genetic algorithms