

车辆聚类及 UAV 轨迹优化辅助下基于 MADDPG 的边缘卸载算法

陈赓, 夏聪慧, 孔令志, 曾庆田

(山东科技大学电子信息工程学院, 山东 青岛 266590)

摘要: 为应对现有边缘计算中路侧单元过载、地面节点覆盖盲区等问题, 提出了车辆聚类及无人机 (UAV) 轨迹优化辅助下基于多智能体深度确定性策略梯度 (MADDPG) 的边缘卸载优化算法 (VCTOEM)。首先, 构建了“车-路-空”联合边缘卸载系统, 建立了多目标优化模型。其次, 设计了车辆聚类机制, 将有剩余资源的车辆聚合成群。此外, 引入了改进粒子群优化算法优化无人机轨迹, 降低飞行能耗与任务卸载的传输时延。最后, 设计了基于多智能体学习的协同卸载算法进行最终决策。仿真结果表明, 所提算法能够有效适应车辆高速移动带来的网络动态变化, 且在高负载场景下保持优异的收敛稳定性, 其平均奖励分别高于其他 6 种基线算法约 10.5%、29.7%、22.8%、9.1%、6.7%、9.9%。

关键词: 边缘卸载; 聚类机制; 轨迹优化; 多智能体学习

中图分类号: TN929.5

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2026058

MADDPG-based edge offloading algorithm assisted by vehicle clustering and UAV trajectory optimization

Chen Geng, Xia Conghui, Kong Lingzhi, Zeng Qingtian

College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Abstract: In order to solve the problems of overload of roadside units and blind area coverage of ground nodes in existing edge computing, a multi-agent deep deterministic policy gradient (MADDPG)-based edge offloading optimization algorithm assisted by vehicle clustering and unmanned aerial vehicle (UAV) trajectory optimization (VCTOEM) was proposed. Firstly, a “vehicle-road-air” joint edge offloading system was constructed, and a multi-objective optimization model was established. Secondly, a vehicle clustering mechanism was designed to aggregate vehicles with remaining resources into groups. In addition, an improved particle swarm optimization (PSO) algorithm was introduced to optimize the trajectory of UAV, ensuring efficient coverage of ground computing power blind spots while reducing flight energy consumption and transmission delays during task offloading. Finally, a collaborative offloading algorithm based on multi-agent learning was designed for final decision-making. Simulation results showed that the proposed algorithm could effectively adapt to the dynamic changes in the network caused by high-speed vehicle movement, maintaining excellent convergence stability in high load scenarios. The average rewards were about 10.5%, 29.7%, 22.8%, 9.1%, 6.7% and 9.9% higher than the other six benchmark algorithms, respectively.

Keywords: edge offloading, clustering mechanism, trajectory optimization, multi-agent learning

收稿日期: 2025-12-19; 修回日期: 2026-02-15

通信作者: 陈赓, gengchen@sdust.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61701284); 山东省自然科学基金资助项目 (No.ZR2022MF226); 山东科技大学青年教师人才培养计划 (No.BJ20221101); 青岛市应用基础研究计划基金资助项目 (No.19-6-2-1-cg); 山东科技大学菁英计划基金资助项目 (No.skr21-3-B-048); 山东省泰山学者计划基金资助项目 (No.tstp20250506)

Foundation Items: The National Natural Science Foundation of China (No.61701284), The Natural Science Foundation of Shandong Province (No.ZR2022MF226), The Talented Young Teachers Training Program of Shandong University of Science and Technology (No.BJ20221101), The Innovative Research Foundation of Qingdao (No.19-6-2-1-cg), The Elite Plan Project of Shandong University of Science and Technology (No.skr21-3-B-048), The Taishan Scholar Program of Shandong Province (No.tstp20250506)

0 引言

随着5G通信技术与智能网联汽车的深度融合,城市交通场景中涌现出自动驾驶、高清地图实时更新、车路协同感知等一批计算密集型与时延敏感型任务^[1]。这类任务对处理节点的算力、响应速度提出了严苛要求,而车辆本地计算单元受限于体积、功耗与成本,往往难以独立满足需求,因此车辆边缘计算(vehicular edge computing, VEC)成为解决这一矛盾的核心技术路径。通过将任务卸载至路侧单元(roadside unit, RSU)、邻近车辆等边缘节点,可有效降低任务处理时延,提升服务质量^[2]。VEC能够打破“云端集中计算”与“车辆本地计算”的二元对立,将边缘节点部署于交通场景核心区域,既具备接近车辆终端的低传输时延优势,又能通过资源聚合满足计算密集型任务需求。

传统VEC卸载方式大多是车辆到基础设施(vehicle-to-infrastructure, V2I),V2I通过车辆与交通信号灯、路侧单元、停车场设备等基础设施的信息交互^[3],为车辆提供实时交通路况、通行建议、停车位引导等服务,助力提升交通管理效率与通行流畅度,但在动态交通场景中却逐渐暴露出显著局限。一方面,RSU作为固定边缘节点,其覆盖范围与算力资源有限,在早晚高峰等车辆密集时段易出现资源过载问题,导致任务排队时延剧增;另一方面,车辆的高速移动特性使车联网拓扑呈现动态变化,传统基于静态节点的卸载策略对车辆间连接不稳定问题难以适配,易出现任务传输中断或重传,进一步加剧系统能耗与服务时延^[4]。现有研究表明,由于车辆计算能力不断提升,许多车辆出现了剩余资源,因此,可以将一辆车的任务卸载到另一辆车上处理,有效扩大系统的计算能力,减轻RSU的负载。

车辆到车辆(vehicle-to-vehicle, V2V)是VEC的关键技术之一,它依托短程无线通信技术实现车辆间直接的数据交互,能实时传递车辆的位置、速度、行驶方向、制动状态等核心信息,为VEC场景下的任务卸载、资源协同提供通信支撑。通过V2V可快速识别周边有空闲算力的车辆,建立临时数据传输链路,将部分计算任务卸载至邻近车辆处理^[5],同时保障车辆移动过程中任务传输的基本连续性,是VEC利用分散车载资源、缓解RSU压力的重要手段。Guo等^[6]提出了一种用于软

件定义车辆协作网络的V2V任务卸载策略,实现了轨迹预测与卸载优化,但是未充分考虑当周边可提供任务卸载服务的车辆资源处于稀缺状态时,系统的任务卸载策略与实施路径。Sun等^[7]提出了一种基于云边缘计算的联合机载任务卸载和作业调度方法,减小系统时延、能耗,实现工作负载平衡,但未充分考虑车辆动态移动导致的资源连接中断的情况。近年来,V2V边缘卸载技术围绕效率优化、安全保障与复杂场景适配持续突破,相关算法从基础多目标优化逐步向智能协同与环境适配演进。Liu等^[8]提出的分布式在线不稳定性感知卸载启发式算法,聚焦车辆移动导致的连接不稳定问题,通过服务路径匹配机制,为任务车辆匹配路径重合度最高的服务车辆,减少因车辆移动导致的卸载中断与切换开销。Shi等^[9]突破了单纯算法优化的局限,将深度强化学习(DRL)与区块链技术结合用于V2V边缘卸载,利用区块链记录车辆的任务执行记录与信用评级,再通过DRL算法学习车辆移动规律、资源空闲状态等动态信息,制定最优卸载决策。Zeng等^[10]提出的融合可重构智能表面与联邦强化学习的方案,能够适配复杂通信环境下的V2V边缘卸载,保护车辆信息安全。上述研究虽在连接稳定性、安全保障与复杂环境适配方面逐步优化,但未充分考虑车辆资源与任务需求的空间分布特性,难以高效整合区域内零散算力,在车辆密集或分布不均场景下易出现资源匹配冗余与调度效率瓶颈。

车辆到无人机(vehicle-to-unmanned aerial vehicle, V2U)是VEC场景下灵活高效的任务卸载技术,它借助空地一体化无线通信技术搭建车辆与无人机间的直接数据传输通道,实时交互车辆任务需求、无人机算力状态及空域位置等关键信息,为动态场景下的任务分流提供支撑,是VEC场景下弥补地面车辆资源稀缺、拓展算力覆盖范围的重要手段^[11]。Yang等^[12]提出在连续时隙内解耦控制决策的扰动李雅普诺夫优化(PLOT)在线算法,在保障任务队列稳定的同时,平衡系统成本与任务处理效率,但未充分结合车辆高速移动带来的通信链路动态变化。Wang等^[13]考虑到无人机(UAV)的功耗和移动用户的意愿,将UAV-边缘计算(MEC)多接入服务器和移动用户之间的交互建模为斯坦伯格博弈,将UAV作为领导者制定卸载服务定价策

略, 车辆作为跟随者根据自身任务需求与成本预算选择卸载方案。Dai等^[14]提出了李雅普诺夫优化+马尔可夫近似的在线无人机辅助VEC卸载策略, 能够解决长期能量约束、降低任务时延问题, 但是没有考虑无人机空域部署与移动路径的动态适配。Bai等^[15]提出的多UAV与边缘-云协同的卸载策略, 未局限于单节点卸载, 而是通过协同策略将车辆复杂任务拆分后, 合理分配至不同UAV以及地面边缘云节点, 解决了早期单UAV难以承接大规模车辆任务的短板。上述文献虽在能量约束、队列稳定等方面实现了优化, 但对无人机的空域部署与服务覆盖适配性设计不足, 未充分联动车辆移动特性与无人机的空间位置调整, 难以在动态场景中持续保障通信链路质量与资源利用效率。

针对上述问题, 本文开展了车-路-空联合边缘卸载系统的建模与优化研究, 主要工作与贡献如下。

1) 建立了车-路-空联合边缘卸载架构, 明确了V2V、V2I、V2U 3类通信链路的传输速率计算方法, 以及不同卸载路径下的时延与能耗量化模型, 为后续优化提供了理论基础。

2) 建立了“最大化系统总效用、最小化总时延与总能耗”的多目标优化问题模型, 设计了激励机制鼓励无人机与车辆集群参与卸载。将优化问题形式化为非线性混合整数规划 (mixed-integer non-

linear programming, MINLP) 问题, 并通过问题归约证明了其NP-难特性, 为算法选择提供了依据。

3) 提出了分层优化策略, 先基于车辆运动特征设计动态聚类机制, 再引入改进粒子群优化算法优化无人机轨迹, 最后构建车辆聚类及轨迹优化辅助下基于多智能体深度确定性策略梯度 (MADDPG) 的边缘卸载优化算法 (VCTOEM), 将卸载决策与资源分配转化为多智能体协同学习问题, 实现动态拓扑下的低时延、低能耗卸载。

4) 通过仿真实验验证了所提算法的有效性。与其他6种基线算法相比, 此算法在长期平均网络奖励、效用、时延、能耗方面均表现出显著优势, 验证了其优异的性能。

1 系统模型和问题建模

1.1 系统模型

在人口密集的城市中心, 因大量设备抢占资源, 导致路边RSU资源长期处于饱和状态, 难以持续为车辆提供稳定高质量服务。尤其在早晚高峰、突发交通事件等应急场景下, RSU极易出现资源过载、服务中断的情况, 为此, 本文采用车辆聚类机制聚合分散车载资源, 同时引入无人机作为应急空中边缘节点, 快速补位覆盖盲区、任务密集区, 形成车-路-空联合边缘卸载架构, 如图1所示。

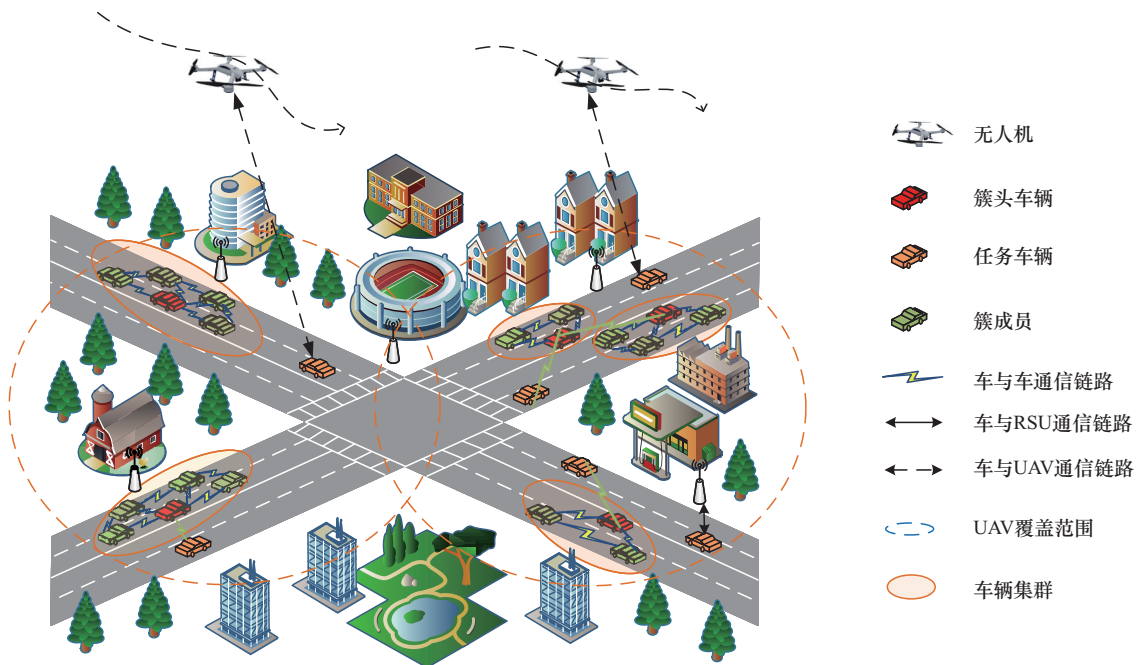


图1 车-路-空联合边缘卸载架构

计算任务有 3 种卸载途径：①卸载到车辆集群；②卸载到路侧单元；③卸载到无人机空中基站。这些边缘节点处理任务时会产生相应的时延、能耗。

本文主要符号和变量如表 1 所示。设连续时间 $t \in [0, T]$ ，将无人机的覆盖范围内的辅助车辆集合表示为 $\hat{V} = \{1, 2, \dots, v, \dots, V\}$ ，任务车辆集合表示为 $\hat{K} = \{1, 2, \dots, k, \dots, K\}$ ，车辆集群集合表示为 $\hat{C} = \{1, 2, \dots, c, \dots, C\}$ ，对应的簇头集合表示为 $\hat{CH} = \{CH_1, CH_2, \dots, CH_n, \dots, CH_N\}$ ，簇成员集合则为 $\hat{CM} = \{CM_1, CM_2, \dots, CM_n, \dots, CM_N\}$ ，路边单元集合表示为 $\hat{I} = \{1, 2, \dots, i, \dots, I\}$ ，无人机集合表示为 $\hat{U} = \{1, 2, \dots, u, \dots, U\}$ 。

引入一个二元变量 $a_{k,c}(t)$ 作为时间 t 任务车辆 k 与车辆集群 c 的关联因子，如果任务车辆 k 与车辆集群 c 关联，则 $a_{k,c}(t) = 1$ ，如果不关联则 $a_{k,c}(t) = 0$ 。引入一个二元变量 $b_{k,u}(t)$ 作为时间 t 车辆 k 与无人机 u 的关联因子，如果车辆 k 与无人机 u 关联，则 $b_{k,u}(t) = 1$ ，否则 $b_{k,u}(t) = 0$ 。引入一个二元变量 $x_{k,i}(t)$ 作为时间 t 车辆 k 与路边单元 i 的关联因子，如果车辆 k 与路边单元 i 关联，则 $x_{k,i}(t) = 1$ ，否则 $x_{k,i}(t) = 0$ 。

1.2 车辆移动模型

车辆 v 在时间 t 的瞬时位置由二维坐标表征，记为 $p_v(t) = \{x_v(t), y_v(t)\}$ 。设车辆集群含 N 个节点，则时间 t 其瞬时速度集合可定义为 $V =$

$\{v_1, v_2, \dots, v_N\}$ 。对于集合中任意两车辆 v_p 和 v_q ，瞬时距离为 $D_{p,q}(t) = \|p_p(t) - p_q(t)\|_2$ ，两辆车行进方向的夹角为 $\theta_{p,q} = \arccos\left(\frac{v_p \cdot v_q}{|v_p| \cdot |v_q|}\right)$ 。

1.3 无人机巡航模型

在空中层部署多架无人机，以固定高度飞行的无人机作为具备较强运算能力的计算服务器，核心功能是处理各类任务。设时间 t 无人机 u 的三维位置坐标为 $w_u(t) = \{x_u(t), y_u(t), h\}$ ，巡航速度为 $v_u(t)$ ，无人机相对于正北方向的航向角为 $\theta_u(t)$ ，无人机圆形覆盖区域的半径为 $r_u(t)$ ，无人机剩余计算资源为 $C_u(t)$ 。无人机 u 和其他无人机 u' 的相对距离可表示为

$$D_{uu'}(t) = \sqrt{(x_u(t) - x_{u'}(t))^2 + (y_u(t) - y_{u'}(t))^2} \quad (1)$$

1.4 任务模型

任务采用“入口-核心-出口”3级分层结构，设任务集合为 $TK = \{TK_{in}, TK_{core}, TK_{out}\}$ 。入口任务 TK_{in} 负责原始数据预处理，出口任务 TK_{out} 承担结果整合与反馈，核心任务为计算密集型主体，包含多个具有依赖关系的子任务。其中核心任务 TK_{core} 可分解为 s 个子任务，即 $\hat{S} = \{1, 2, \dots, s, \dots, S\}$ 。

入口任务和出口任务的属性由三元组 $A = (D, C, T^{\max})$ 描述，分别代表在时间 t 时，任务的总比特大小、完成该任务所需的周期数，以及此任务所允许的最大时延。具有依赖型关系的子任务属性

表 1 主要符号和变量

参数	含义	参数	含义
\hat{K}	任务车辆集合	\hat{U}	无人机集合
\hat{V}	辅助车辆集合	\hat{I}	路侧单元集合
\hat{C}	车辆集群集合	\hat{CH}	簇头集合
TK	任务集合	\hat{CM}	簇成员集合
\hat{S}	子任务集合	T_k^l	本地处理时延
$T_{s,c}$	子任务 s 卸载到车辆集群 c 的时延	$R_{k,c}$	任务车辆 k 到车辆集群 c 的传输速率
$T_{s,u}$	子任务 s 卸载到无人机 u 的时延	$R_{k,u}$	任务车辆 k 到无人机 u 的传输速率
$T_{s,i}$	子任务 s 卸载到路侧单元 i 的时延	$R_{k,i}$	任务车辆 k 到路侧单元 i 的传输速率
$E_{s,c}$	子任务 s 卸载到车辆集群 c 的能耗	$E_{s,u}$	子任务 s 卸载到无人机 u 的能耗
$E_{s,i}$	子任务 s 卸载到路侧单元 i 的能耗	$a_{k,c}$	任务车辆 k 与车辆集群 c 的关联因子
$b_{k,u}$	任务车辆 k 与无人机 u 的关联因子	$x_{k,i}$	任务车辆 k 与路侧单元 i 的关联因子
E_u^{circ}	无人机 u 推进能耗	U_k	系统效用

由五元组 $B = (D_s, I_s, C_s, O_s, T_s^{\max})$ 描述, 其中, D_s 为子任务 s 的自身固有的总比特大小; I_s 为子任务 s 的输入数据大小, 这些数据来源于其前驱任务的输出 O_{s-1} , 代表了任务链中流动的、需要在连续子任务间传递的中间计算结果; C_s 为完成子任务 s 所需的周期数; O_s 为子任务 s 的输出数据大小; T_s^{\max} 为完成子任务 s 所允许的最大时延。 D_s 和 I_s 的核心区别在于, D_s 是任务本身的传输开销, 而 I_s 是任务依赖所带来的数据流开销。子任务之间存在线性任务依赖关系, 第 $s-1$ 个子任务的输出是第 s 个子任务的输入, 即 $I_s = O_{s-1}$ 。对于第 1 个子任务, $I_1 = O_0 = 0$ ^[16]。子任务依赖关系如图 2 所示。本文引入 α 、 β 、 γ 分别为子任务 s 对车辆集群、无人机、路侧单元的卸载决策, 决策为 1 则卸载, 决策为 0 则不进行卸载。

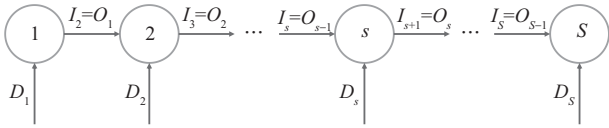


图2 子任务依赖关系

1.5 通信模型

在边缘卸载系统中, 任务车辆可通过 V2V、V2I 及 V2U 3 类链路进行数据传输。为准确刻画各链路的通信能力, 本文基于香农公式构建了传输速率模型。考虑到各边缘节点的带宽资源有限且需在多个关联车辆间共享, 在公式中引入了带宽分配比例因子 η 。该因子由任务车辆和边缘节点的实时关联关系决定, 能够将节点的总带宽在其当前服务的所有车辆间进行分配。

用 p_k 表示第 k 个任务车辆的发射功率, B_c 表示集群 c 的总带宽, σ_1^2 表示干扰功率, 则与集群 c 关联的任务车辆 k 的可达到速率表示为

$$R_{k,c}(t) = \eta_{k,c}(t) B_c \ln \left(1 + \frac{p_k |H_{k,c}(t)|^2}{\sigma_1^2} \right) \quad (2)$$

其中, $\eta_{k,c}(t) = \left[\sum_{k=1}^K a_{k,c}(t) \right]^{-1}$ 表示集群带宽的分配比例, 由在时刻 t 与集群 c 关联的任务车辆个数决定; $H_{k,c}(t)$ 为从任务车辆 k 到车辆集群 c 的信道增益。

路侧单元的总带宽为 B_1 , 则 t 时任务车辆 k 与路侧单元 i 之间的传输速率为

$$R_{k,i}(t) = \eta_{k,i}(t) B_1 \ln \left(1 + \frac{p_k |H_{k,i}(t)|^2}{\sigma_2^2} \right) \quad (3)$$

其中, $\eta_{k,i}(t) = \left[I \cdot \sum_{k=1}^K x_{k,i}(t) \right]^{-1}$ 表示路侧单元带宽的分配比例, 由路侧单元数量以及在时刻 t 与 RSU i 关联的任务车辆个数决定, I 为 RAU 的数量, $H_{k,i}(t)$ 为从任务车辆 k 到路侧单元 i 的信道增益。

任务车辆到 UAV 的通信链路考虑了视距 (LoS) 与非视距 (NLoS) 两种通信场景的分析^[17]。同时考虑了这两类场景的出现概率及其对应的路径损耗情况。任务车辆 k 和无人机 u 之间的 LoS 和 NLoS 通信的发生概率为

$$\begin{cases} P_{k,u}^{\text{LoS}} = \frac{1}{\mu \exp \left(-\omega \left(\left(\frac{180}{\pi} \right) \arcsin \left(\frac{h_c}{d_{k,u}} \right) - \mu \right) \right)} \\ P_{k,u}^{\text{NLoS}} = 1 - P_{k,u}^{\text{LoS}} \end{cases} \quad (4)$$

其中, μ 和 ω 是取决于环境的常数值, h_c 为无人机高度, $d_{k,u}$ 为任务车辆与无人机的距离。因此, 任务车辆 k 和无人机 u 之间的路径损耗模型可以表示为

$$PL_{k,u}^{\zeta} = L_{k,u}^{\text{FSPL}} + \eta_{\zeta} \zeta \in \{ \text{LoS}, \text{NLoS} \} \quad (5)$$

其中, $L_{k,u}^{\text{FSPL}}$ 表示自由空间路径损耗, η_{ζ} 是 LoS 或 NLoS 链路的过度路径损耗。因此, 任务车辆 k 和计算无人机 u 之间的平均路径损耗可以表示为

$$\overline{PL}_{k,u} = PL_{k,u}^{\text{LoS}} \cdot P_{k,u}^{\text{LoS}} + PL_{k,u}^{\text{NLoS}} \cdot P_{k,u}^{\text{NLoS}} \quad (6)$$

任务车辆 k 和计算无人机 u 之间的信道增益可以表示为

$$g_{k,u} = \frac{1}{\sqrt{\overline{PL}_{k,u}}} \quad (7)$$

无人机的总带宽为 B_2 , 则 t 时任务车辆 k 与无人机 u 之间的传输速率为

$$R_{k,u}(t) = \eta_{k,u}(t) B_2 \ln \left(1 + \frac{p_k g_{k,u}}{\sigma_3^2} \right) \quad (8)$$

其中, $\eta_{k,u}(t) B_2$ 表示无人机 u 分配给每个设备的带宽, σ_3^2 为干扰功率。

1.6 计算模型

任务车辆 k 自身的 CPU 计算频率为 f_k^l 。车辆用户 k 的入口任务和出口任务在本地计算, 因此, 任务在本地计算的时延和能耗分别为

$$T_k^l = \frac{C_{in}}{f_k^l} + \frac{C_{out}}{f_k^l} \quad (9)$$

$$E_k^l = t_k(C_{in} + C_{out})(f_k^l)^2 \quad (10)$$

其中, t_k 为与硬件相关的影响因子。

若任务车辆 k 的子任务 s 卸载到车辆集群 c 中进行处理, 则传输和处理子任务的时延和能耗分别为

$$T_{s,c} = \frac{D_s + I_s}{R_{k,c}} + \frac{C_s}{f_c} \quad (11)$$

$$E_{s,c} = p_k \frac{D_s + I_s}{R_{k,c}} + t_c C_s (f_c)^2 \quad (12)$$

其中, $f_c = f_m^{\max} \cdot \theta_{k,c}(t)$ 表示车辆集群 c 分配给子任务 s 的计算资源, f_m^{\max} 是集群中算力最大的簇成员的计算资源, $\theta_{k,c}(t)$ 是资源分配比例, t_c 是与硬件相关的影响因子。

若任务车辆 k 的子任务 s 卸载到无人机 u 中进行处理, 则传输和处理子任务的时延和能耗分别为

$$T_{k,u} = \frac{D_s + I_s}{R_{k,u}} + \frac{C_s}{f_u} \quad (13)$$

$$E_{s,u} = p_k \frac{D_s + I_s}{R_{k,u}} + t_u C_s (f_u)^2 \quad (14)$$

其中, $f_u = \frac{1}{S} (f_u^{\text{total}} \cdot \theta_{k,u}(t))$ 表示无人机 u 分配给子任务 s 的计算资源, f_u^{total} 是无人机剩余的总计算资源, $\theta_{k,u}(t)$ 是资源分配比例, t_u 为与硬件相关的影响因子。

若任务车辆 k 的子任务 s 卸载到路侧单元 i 中进行处理, 则传输和处理子任务的时延和能耗分别为

$$T_{k,i} = \frac{D_s + I_s}{R_{k,i}} + \frac{C_s}{f_i} \quad (15)$$

$$E_{s,i} = p_k \frac{D_s + I_s}{R_{k,i}} + t_i C_s (f_i)^2 \quad (16)$$

其中, $f_i = \frac{1}{S} (f_i^{\text{total}} \cdot \theta_{k,i}(t))$ 表示路侧单元 i 分配给子任务 s 的计算资源, f_i^{total} 是路侧单元剩余的总计算资源, $\theta_{k,i}(t)$ 是资源分配比例, t_i 为与硬件相关的影响因子。

资源分配比例代表了边缘节点将自身剩余总计算资源分配给特定任务的份额, 体现了边缘节点内部存在多任务共享资源的情况。

因此, t 时刻系统完成任务的总时延和总能耗分别为

$$T^{\text{total}}(t) = \sum_{k=1}^K (\max(T_k^l, \sum_{c=1}^C a_{k,c} (\sum_{s=1}^S \alpha T_{s,c})) + \sum_{u=1}^U b_{k,u} (\sum_{s=1}^S \beta T_{s,u}) + \sum_{i=1}^I x_{k,i} (\sum_{s=1}^S \gamma T_{s,i})) \quad (17)$$

$$E_k^{\text{total}}(t) = \sum_{k=1}^K (E_k^l + \sum_{c=1}^C a_{k,c} (\sum_{s=1}^S \alpha E_{s,c})) + \sum_{u=1}^U b_{k,u} (\sum_{s=1}^S \beta E_{s,u}) + \sum_{i=1}^I x_{k,i} (\sum_{s=1}^S \gamma E_{s,i}) + E_u^{\text{circ}} \quad (18)$$

其中, E_u^{circ} 为无人机推进能耗, α 、 β 、 γ 分别为子任务对 3 种边缘节点的卸载决策, $a_{k,c}$ 、 $b_{k,u}$ 、 $x_{k,i}$ 分别为任务车辆与 3 种边缘节点的关联因子。

1.7 问题建模

在边缘卸载中, 网络效用能够评估资源利用与服务质量。向任务车辆 k 收取的任务传输单价为 p_1 , 租赁无人机、车辆和路侧单元的频谱成本分别为 p_2 、 p_3 、 p_4 , 则任务车辆 k 的通信效用为

$$U_k^{\text{tra}} = p_1 D_{\text{core}} - b_{k,u}(t) p_2 R_{k,u} - a_{k,c}(t) p_3 R_{k,c} - x_{k,i}(t) p_4 R_{k,i} \quad (19)$$

向任务车辆 k 收取的任务计算单价为 p_5 , 而租赁无人机、车辆和路侧单元的计算资源成本分别为每赫兹 p_6 、 p_7 和 p_8 , 则任务车辆 k 的计算效用为

$$U_k^{\text{com}} = p_5 C_{\text{cove}} - b_{k,u}(t) p_6 f_u - a_{k,c}(t) p_7 f_c - x_{k,i}(t) p_8 f_i \quad (20)$$

因此, 对于任务车辆 k 产生任务的总效用 U_k 可以计算为

$$U_k = U_k^{\text{tra}} + U_k^{\text{com}} \quad (21)$$

本文的主要优化目标是通过优化任务卸载策略、计算资源分配, 最大化系统效用, 同时最小化任务处理的总时延、总能耗, 兼顾任务截止时间约束与智能体资源限制, 实现边缘卸载效率最大化与资源利用最优化。于是目标函数可以表示为

$$F(t) = U_k - \phi T^{\text{total}}(t) - \psi E^{\text{total}}(t) \quad (22)$$

其中, ϕ 、 ψ 为时延和能耗所占权重因子。优化问题可以表示为

$$\text{P1: } \max_{f,\alpha,\beta,\gamma} \sum_{t=1}^T F(t)$$

$$\text{s.t. C1: } a_{k,c}(t), b_{k,u}(t), x_{k,i}(t) \in \{0,1\},$$

$$\forall k \in \hat{K}, \forall c \in \hat{C}, \forall u \in \hat{U}, \forall i \in \hat{I}$$

$$\text{C2: } \alpha, \beta, \gamma \in \{0,1\}$$

$$\text{C3: } \phi + \psi = 1$$

$$\begin{aligned}
\text{C4: } & \sum_{c=1}^C a_{k,c}(t) + \sum_{u=1}^U b_{k,u}(t) + \sum_{i=1}^I x_{k,i}(t) \leq 1, \\
& \forall k \in \hat{K}, \forall c \in \hat{C}, \forall u \in \hat{U}, \forall i \in \hat{I} \\
\text{C5: } & \sum_{k=1}^K a_{k,c}(t) f_c \leq f_c^{\max}, \forall k \in \hat{K}, \forall c \in \hat{C} \\
\text{C6: } & \sum_{k=1}^K b_{k,u}(t) f_u \leq f_u^{\max}, \forall k \in \hat{K}, \forall u \in \hat{U} \\
\text{C7: } & \sum_{k=1}^K x_{k,i}(t) f_i \leq f_i^{\max}, \forall k \in \hat{K}, \forall i \in \hat{I} \\
\text{C8: } & T_k^{\text{total}} \leq T_k^{\max}, \forall k \in \hat{K} \quad (23)
\end{aligned}$$

其中, $f = \{f_c, f_u, f_i\}$, C1表示任务车辆与车辆集群、任务车辆与无人机、任务车辆与路侧单元的关联因子, C2表示子任务卸载决策, C3表示时延与能耗的权重之和为1, C4表示一个任务车辆最多只可关联1个边缘节点, C5、C6、C7分别表示与车辆集群、无人机、路侧单元关联的所有任务所用计算资源之和不超过它们的最大计算资源, C8表示任务处理时延约束。

上述问题不仅是一个MINLP问题,而且是一个NP-难问题。具体而言,车辆的高速移动使网络拓扑持续变化,路侧单元的固定覆盖范围限制与无人机的动态巡航轨迹进一步加剧了系统状态的时变性,导致任务用户与边缘节点的关联关系随时间不断重构,资源分配的约束条件也因此始终处于动态调整中。从边缘节点的资源调度视角出发,可将该问题归约为经典的0-1背包问题^[18],将边缘节点的总可用资源,如计算能力、通信带宽等定义为“背包容量” C ;将待处理的计算任务视为“物品”,每个任务对应两个核心属性——“重量”与“价值”,其中“重量”等价于任务卸载至该节点所需消耗的资源量,如计算周期、带宽占用等,“价值”则是通过系统效用提升以及任务处理的时延降低量、能耗节省量的加权值综合表征。决策目标是从所有任务中选择一个子集,在总资源消耗不超过 C ,即“重量”总和 $\leq C$ 的约束下,实现总价值的最大化,这一目标与0-1背包问题的优化逻辑完全一致。由于0-1背包问题已被证明为NP-难,而车-路-空边缘卸载场景下的计算卸载与资源分配问题可通过上述映射等价于该问题,因此该场景下的优化问题同样属NP-难问题。

问题P1的NP-难特性决定了其难以直接求得实

时最优解,为此,将这一复杂联合优化问题P1分解为3个子问题,分别为辅助车辆资源整合子问题、无人机任务覆盖子问题以及协同卸载与资源分配决策子问题。

首先将无人机位置看作固定值,并将路侧单元视为静态资源。此时,P1中与车辆间协作相关的内容可独立提取,并结合车辆集群的相关约束,形成辅助车辆资源整合问题。该问题的目标是优化V2V卸载链路与集群内资源分配,将分散的车载计算资源整合为V2V资源池。第一个子任务的优化问题可表示为

$$\begin{aligned}
\text{P1.1: } & \max_{f_c, a} \sum_{t=1}^T F(t) \\
\text{s.t. } & \text{C1: } a_{k,c}(t) \in \{0,1\}, \forall k \in \hat{K}, \forall c \in \hat{C} \\
& \text{C2: } \sum_{c=1}^C a_{k,c}(t) \leq 1, \forall k \in \hat{K}, \forall c \in \hat{C} \\
& \text{C3: } \sum_{k=1}^K a_{k,c}(t) f_c \leq f_c^{\max}, \forall k \in \hat{K}, \forall c \in \hat{C} \\
& \text{C4: } D_{hm} \leq R_h, \forall h \in \hat{C}H, \forall m \in \hat{C}M \\
& \text{C5: } T_c \geq T_k^{\min}, \forall c \in \hat{C} \quad (24)
\end{aligned}$$

其中,C1、C2、C3为问题P1的车辆集群部分,C4表示簇成员必须在簇头通信的一跳范围内,C5表示簇头与簇成员的连接时间不能低于任务处理的最小时间。

在获得地面车辆集群的结构信息及其服务能力后,可在此基础上优化空中资源,使其尽可能地覆盖更多的任务车辆。因此可将车辆聚类结果视为固定值,并将问题P1中与无人机及V2U链路相关的内容提取,结合无人机飞行约束,形成无人机任务覆盖问题,旨在优化无人机轨迹从而最大化其覆盖任务与通信质量。第二个子任务的优化问题可表示为

$$\begin{aligned}
\text{P1.2: } & \max_{f_u, b} \sum_{t=1}^T F(t) \\
\text{s.t. } & \text{C1: } b_{k,u}(t) \in \{0,1\}, \forall k \in \hat{K}, \forall u \in \hat{U} \\
& \text{C2: } \sum_{u=1}^U b_{k,u}(t) \leq 1, \forall k \in \hat{K}, \forall u \in \hat{U} \\
& \text{C3: } \sum_{k=1}^K b_{k,u}(t) f_u \leq f_u^{\max}, \forall k \in \hat{K}, \forall u \in \hat{U} \\
& \text{C4: } x_{\min} \leq x_u(t) \leq x_{\max}, y_{\min} \leq y_u(t) \leq y_{\max}, \\
& \quad \forall u \in \hat{U}
\end{aligned}$$

$$\begin{aligned}
 & C5: v_u(t) \leq V_{\max}, \forall u \in \hat{U} \\
 & C6: D_{uu'}(t) \geq D_u^{\min}, \forall u \in \hat{U}
 \end{aligned} \quad (25)$$

其中, C1、C2、C3 为问题 P1 的无人机相关部分, C4 为无人机边界约束, C5 为无人机速度约束, C6 为无人机防撞约束。

在前两个子问题的基础上, 最终形成给定车辆集群及无人机轨迹的协同卸载与资源分配决策问题。第三个子任务的优化问题可表示为

$$\begin{aligned}
 & P1.3: \max_{f, \alpha, \beta, \gamma} \sum_{t=1}^T F(t) \\
 & \text{s.t. } C1: \alpha, \beta, \gamma \in \{0, 1\} \\
 & \quad C2: \phi + \varphi = 1 \\
 & \quad C3: \sum_{k=1}^K a_{k,c}(t) f_c \leq f_c^{\max}, \forall k \in \hat{K}, \forall c \in \hat{C} \\
 & \quad C4: \sum_{k=1}^K b_{k,u}(t) f_u \leq f_u^{\max}, \forall k \in \hat{K}, \forall u \in \hat{U} \\
 & \quad C5: \sum_{k=1}^K x_{k,i}(t) f_i \leq f_i^{\max}, \forall k \in \hat{K}, \forall i \in \hat{I} \\
 & \quad C6: T_k^{\text{total}} \leq T_k^{\max}, \forall k \in \hat{K}
 \end{aligned} \quad (26)$$

P1.3 的决策环境不再是原始的动态随机网络, 而是经过 P1.1 和 P1.2 预处理后的、相对稳定且资源与覆盖信息更明确的系统状态。

2 优化策略

本文在第 1 节将 NP-难问题分解为 3 个子问题, 对于第一个子问题, 设计了基于车辆运动特性的动态聚类机制, 将分散且动态的车载计算资源整合为相对稳定的资源单元, 解决地面网络拓扑动态性问题。对于第二个子问题, 引入了改进的粒子群优化 (PSO) 算法优化无人机轨迹, 实现空中节点覆盖与任务需求的实时匹配。对于第三个子问题, 在前两层优化的基础上构建融合集群信息与轨迹参数的 MADDPG 任务卸载策略, 提升动态场景下的决策适配性, 在保证方案可行性与实时性的基础上, 逼近原复杂优化问题的近似最优解。

2.1 车辆聚类机制

本文提出了“动态簇数量+多指标簇头选择+约束化成员筛选”的车辆聚类机制, 通过自适应调整簇规模、综合评估簇头能力、严格筛选成员, 提升地面网络的稳定性与资源聚合效率, 车辆集群由簇头和簇成员组成。任务在车辆集群中以子任务的形式串行处理。子任务卸载至集群后, 由簇头节点

接收, 并依据集群内各成员的资源状态, 将其调度至当前算力最强的成员车辆进行处理。车辆聚类机制如图 3 所示。

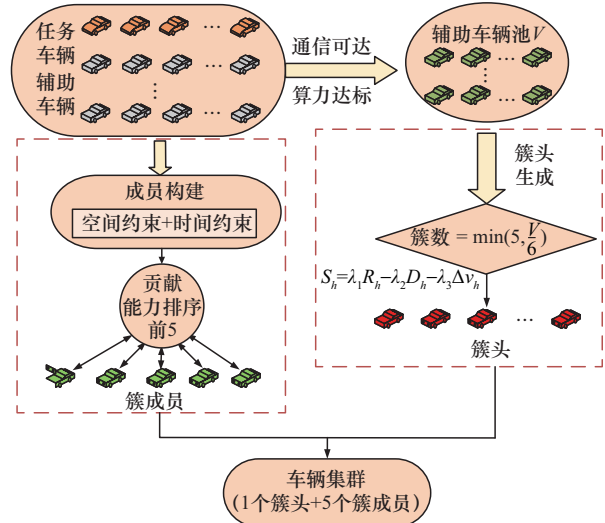


图3 车辆聚类机制

簇头选择综合考虑车辆的通信覆盖能力、计算资源与位置稳定性。设任务车辆 k 的通信半径为 r_k , 无人机覆盖半径为 r_u , 辅助车辆 v 剩余算力为 $C_v(t)$, 子任务 s 的最小计算量为 C_s^{\min} , 则任务车辆 k 与辅助车辆 v 的相对距离为

$$D_{k,v}(t) = \sqrt{(x_k(t) - x_v(t))^2 + (y_k(t) - y_v(t))^2} \quad (27)$$

任务车辆通过 V2V 通信获取周边辅助车辆的实时状态, 仅将满足“通信可达”与“资源有效”的车辆纳入候选簇头池。

通信可达: 辅助车辆必须在无人机的覆盖范围内, 并且必须在任务车辆的通信范围内。

资源有效: 辅助车辆的剩余算力必须能够满足单个子任务的最小计算需求。

因此, 当 $D_{k,v}(t) \leq \min(r_k, r_u)$ 且 $C_v(t) \geq C_s^{\min}$ 时, 辅助车辆进入簇头候选池 \tilde{V} , 进行最后的综合评分。簇头的综合评分 S_h 融合了资源聚合能力、通信覆盖范围与位置稳定性 3 个维度, 定义为

$$S_h = \lambda_1 \cdot R_h - \lambda_2 \cdot D_h - \lambda_3 \cdot \Delta v_h \quad (28)$$

其中, $R_h = C_h(t) + \frac{1}{|\Omega_h|} \sum_{n \in \Omega_h} C_n(t)$ 为簇头资源聚合能力, 即簇头自身算力与通信范围内所有潜在成员的平均算力之和; Ω_h 为簇头通信范围内的所有潜在成员的集合; D_h 为簇头与任务车辆的相对距

离; Δv_h 为簇头与任务车辆的相对速度; λ_1 、 λ_2 和 λ_3 为权重系数, 满足 $\lambda_1 + \lambda_2 + \lambda_3 = 1$ 。按评分由高到低选取 K_{ch} 个簇头, 从而生成车辆 k 的 K_{ch} 个候选簇。

$$K_{ch} = \min\left(5, \left\lfloor \frac{|\tilde{V}|}{6} \right\rfloor\right) \quad (29)$$

其中, \tilde{V} 为簇头候选池。

簇头确定后, 需为每个簇头匹配满足“连接稳定”与“资源充足”的簇成员, 避免因成员移动或算力不足导致的任务传输中断。令簇头的坐标为 (x_h, y_h) , 簇成员的坐标为 (x_m, y_m) , 簇成员随车辆移动实时变化, 成员筛选遵循以下两个条件。

距离: 簇成员必须在簇头通信的一跳范围内, 即 $D_{hm} \leq r_h$ 。

连接时间: 簇头与簇成员的有效连接时间 T_c 需大于任务处理的最小时延需求 T_k^{\min} 。有效连接时间 T_c 的计算需区分簇头与簇成员的相对运动方向, 假设簇头与簇成员的平均行驶速度分别为 v_h 和 v_m , 二者的相对位置存在两种情形。

1) 当簇头处于簇成员沿行驶方向的后方位置时, 簇成员要脱离簇头通信范围所需移动的距离为

$$d_m = \sqrt{r_h^2 - B^2} - A \quad (30)$$

其中, r_h 代表簇头的单跳通信覆盖半径, B 为两车间纵向坐标差的绝对值 $|y_m - y_h|$, A 为两车间横向坐标差的绝对值 $|x_m - x_h|$ 。

2) 当簇头处于簇成员沿行驶方向的前方位置时, 簇成员要脱离簇头通信范围所需移动的距离为

$$d_m = \sqrt{r_h^2 - B^2} + A \quad (31)$$

因此, 簇头与簇成员的连接时间为

$$T_c = \frac{d_m}{|v_h - v_m|} \quad (32)$$

满足上述条件的成员则可以进入成员候选池, 候选成员的贡献能力由车辆的剩余计算资源与其能和簇头连接的时间决定, 即

$$\text{贡献能力} = C_m(t) \times T_c \quad (33)$$

按“贡献能力”降序排序, 取5个为簇成员。每个车辆集群的最终结构为1个簇头和5个簇成员, 簇头负责与任务车辆以及各个簇成员进行信息交互和数据传输, 簇成员负责提供空闲资源, 共同处理

卸载的任务。车辆聚类机制以固定的卸载决策周期为基准周期性执行, 每个集群从这一周期聚类成功开始, 在整个周期内保持结构稳定, 直至下一周期开始时解散并重新组建, 簇生命周期为系统卸载周期时长 10 s。车辆聚类机制如算法 1 所示。

算法 1 车辆聚类机制算法

输入 任务车辆 k 坐标 $p_k(t)$ 、通信半径 r_k , 任务时延阈值 T^{\max} , 子任务最小计算量 C_s^{\min} , 辅助车辆坐标 $p_v(t)$ 、算力 $C_v(t)$, 速度 $v_v(t)$, 通信半径 r_v , 无人机覆盖半径 r_u

输出 任务车辆 k 的候选簇集合 \tilde{C} (每个簇含 1 个簇头+5 个簇成员)

1) 初始化簇头候选池 \tilde{V}

2) 初始化簇集合 \tilde{C}

3) for 1 to V

4) 根据式(27)计算 $D_{k,v}$

5) if $D_{k,v}(t) \leq \min(r_k, r_u)$ and $C_v(t) \geq C_s^{\min}$

6) 将 v 加入 \tilde{V}

7) end if

8) end for

9) 计算簇数 $K_{ch} = \min\left(5, \left\lfloor \frac{|\tilde{V}|}{6} \right\rfloor\right)$

10) 计算辅助车综合评分

11) 按评分降序选前 K_{ch} 个作为簇头

12) 遍历每个簇头

13) 遍历剩余辅助车辆

14) if 满足空间时间约束

15) 进入候选池

16) 按贡献能力排序, 取前 5 个成员

17) end if

18) 构建簇加入 \tilde{C}

19) 返回 \tilde{C}

2.2 基于改进 PSO 的无人机轨迹优化

为了让无人机边缘节点能够覆盖更多待执行卸载任务的车辆, 同时缩短无人机与各任务车辆之间的通信距离, 本文为无人机开展了轨迹优化工作。

将无人机 u 在时间 t 可成功处理的、满足时延约束的子任务总数定义为无人机 u 覆盖区域内的“有效任务服务量” $Q(t)$, 则

$$Q_u(t) = \sum_{k=1}^K b_{k,u}(t) \sum_{s=1}^S \beta \quad (34)$$

为了提高无人机覆盖任务量并降低数据传输过程中的链路损耗与传输时延，可将问题P1.2转化为

$$\begin{aligned}
 & \text{P2: } \max_{v_u, p_u} J = \omega_1 Q_u(t) - \omega_2 E_u^{\text{circ}}(t) - \omega_3 \overline{\text{PL}}_{k,u}(t) \\
 & \text{s.t. } \text{C1: } x_{\min} \leq x_u(t) \leq x_{\max}, y_{\min} \leq y_u(t) \leq y_{\max}, \\
 & \quad \forall u \in \hat{U} \\
 & \text{C2: } v_u(t) \leq V_{\max}, \forall u \in \hat{U} \\
 & \text{C3: } D_{uu'}(t) \leq D_u^{\min}, \forall u \in \hat{U} \quad (35)
 \end{aligned}$$

其中， $p_u = (x_u, y_u)$ ， J 为无人机轨迹优化的综合收益， ω_1 、 ω_2 、 ω_3 为权重因子。问题P2的优化目标以“最大化无人机覆盖区域内的任务服务能力”为核心，兼顾无人机巡航能耗与通信链路质量。

为了解决该问题，本文引入了改进的PSO算法，算法核心逻辑是通过“群体协作”寻找最优解。算法中，每个待优化问题的潜在解被称为“粒子”，所有粒子构成“粒子群”。每个粒子有两个核心属性：位置和速度。无人机轨迹优化如图4所示。

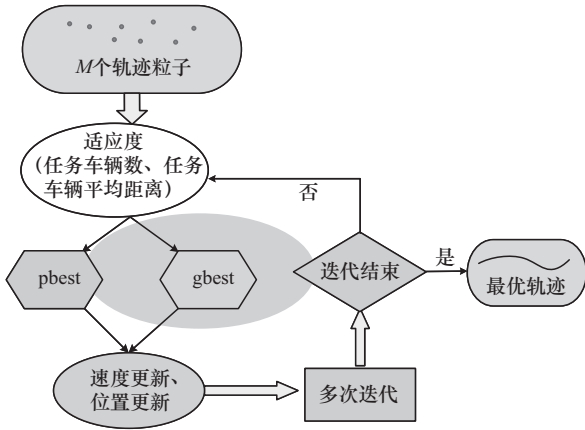


图4 无人机轨迹优化

起初会根据每个粒子初始位置覆盖范围内的任务车辆数量，以及与所有任务车辆的平均距离计算初始适应度，根据适应度挑选出自身历史最优位置 (pbest) 和群体历史最优位置 (gbest)，并根据这两个指标进行速度和位置的更新。

速度更新式为

$$v(t+1) = c_0 \cdot v(t) + c_1 \cdot r_1 \cdot (pbest(t) - p_u(t)) + c_2 \cdot r_2 \cdot (gbest(t) - p_u(t)) \quad (36)$$

其中， c_0 是惯性权重，用于平衡全局探索和局部开发； c_1 、 c_2 是学习因子； r_1 、 r_2 是随机数，增加搜索随机性。

位置更新公式为

$$p(t+1) = p(t) + v(t+1) \quad (37)$$

接下来进行多次迭代，每一轮迭代都会依据当前粒子的自身历史最优、群体全局最优位置，动态更新各粒子的速度与位置。随着迭代次数的增加，粒子群将不断收敛于更优解，最终为无人机规划出能够最大化综合收益的最优轨迹。无人机轨迹优化如算法2所示。

算法2 基于改进PSO的无人机轨迹优化算法

输入 任务车辆位置集合 $\{w_k(t)\}_{k \in \hat{K}}$ 、子任务需求集合 $\{D_s, C_s, T_s^{\max}\}_{s \in \hat{S}}$ 、无人机初始位置 $\{x_u(0), y_u(0), h\}$ 、权重 ω_1 、 ω_2 、 ω_3 ，迭代次数 G

输出 无人机 u 的最优轨迹 $\{x_u^*(t), y_u^*(t), h, \}$

- 1) 初始化粒子群：每个粒子代表一组轨迹参数 $\{x, y, h\}$ ，粒子数量为 M ，随机生成初始位置与速度
- 2) 计算初始适应度：对每个粒子，代入综合收益 J 公式，计算适应度值，记录每个粒子的个体最优 pbest 与群体最优 gbest
- 3) 初始化迭代次数 $g = 1$
- 4) while $g \leq G$
- 5) 根据式(36)更新粒子速度
- 6) 根据式(37)更新粒子位置，若位置超出约束范围，通过“边界回弹”调整至约束内
- 7) 根据更新后的粒子位置，重算适应度
- 8) 若当前粒子适应度优于 pbest，则更新 pbest；若所有粒子的 pbest 优于 gbest，则更新 gbest；
- 9) end while
- 10) 选取最终适应度最高粒子的轨迹为无人机 u 的最优轨迹

2.3 基于MADDPG的联合边缘卸载优化算法

2.1节的车辆聚类与2.2节的无人机轨迹优化分别处理了资源聚合与任务覆盖问题，本文最终要解决的问题在于，各个任务车辆应如何为其子任务选择卸载目标，并确定所需分配的计算资源比例。因此，本文提出了基于MADDPG的联合边缘卸载优化方案。该方案在DDPG的基础上，融入“集中式Critic-分布式Actor”架构与多智能体联合经验回放机制，不仅能解决多智能体互动时的“信息不对

称”问题,还能通过全局价值评估引导各智能体协同优化^[19]。MADDPG将每个任务车辆视为一个独立的智能体,其状态观测信息来源于前两阶段输出的稳定化环境。通过集中式训练,Critic网络能够利用全局状态信息评估联合动作的长期价值,从而引导各个Actor网络学习到相互协作的策略。通过分布式执行,每个任务车辆可仅依据自身局部观测做出快速决策,保障了算法的实时性。

从DRL理论框架出发,可将车-路-空协同场景中任务车辆节点的计算任务卸载问题建模为多智能体马尔可夫决策过程(MMDP)。该MMDP可形式化定义为如下四元组。

$$(S^K, A^K, P^K_{(s_{t+1}|s_t, a_t)}, R^K_{(s_t, a_t)}) \quad (38)$$

其中, S^K 代表任务卸载过程中的状态空间, A^K 代表任务车可执行的动作空间, $P^K_{(s_{t+1}|s_t, a_t)}$ 表示在时间 t 处于状态 s_t 并执行动作 a_t 后, 在下一时间 $t+1$ 转移至状态 s_{t+1} 的概率, $R_{(s_t, a_t)}$ 则对应在该状态动作下的即时奖励, K 为任务车辆的数量。

在时间 t , 系统的状态空间可表示为

$$S^K(t) = [p_1(t), D_1(t), C_1(t); \dots; p_K(t), D_K(t), C_K(t); p_v(t), C_v(t); p_u(t), C_u(t); p_i(t), C_i(t); a_{k,c}(t), b_{k,u}(t), x_{k,i}(t)] \quad (39)$$

包含任务车辆的位置、任务大小和计算量, 任务车辆通信范围内各边缘节点的位置剩余计算资源, 以及各个关联因子。

在时间 t , 系统的动作空间可表示为

$$A^K(t) = A_1(t) \times A_2(t) \times \dots \times A_K(t) \quad (40)$$

其中, 第 k 个智能体的动作空间 $A_k(t)$ 定义为

$$A_k(t) = [o_k(t), \theta_{k,c}(t), \theta_{k,u}(t), \theta_{k,i}(t)] \quad (41)$$

其中, $o_k(t)$ 表示任务车辆 k 的卸载目标选择, $\theta_{k,c}(t)$ 、 $\theta_{k,u}(t)$ 、 $\theta_{k,i}(t)$ 分别表示车辆集群 c 、无人机 u 、路边单元 i 分配给车辆 k 的计算资源比例。

在上述动作空间的框架下, 各智能体基于局部观测生成卸载与资源分配决策。考虑到车辆自身计算资源受限, 智能体将采用轻量化的策略网络结构, 能够显著降低决策的计算复杂度。此外, 智能体生成决策后, 任务卸载与资源分配开始执行, 与此同时智能体准备下一决策周期的计算。该设计能够避免策略决策时间对任务完成总时延产生影响。

奖励函数以最大化系统总效用、最小化总时延

与总能耗为目标, 平衡收益和成本的关系, 引导智能体生成低时延、低能耗且高收益的卸载策略, 可表示为

$$R_t(S_t, A_t) = \sum_{k=1}^K (U_k - \phi T_k^{\text{total}} - \psi E_k^{\text{total}}) \quad (42)$$

为降低训练过程中的价值估计振荡, 目标Critic网络 Q_{target} 与目标Actor网络 π_{target} 的参数不直接参与梯度更新, 而是通过当前网络参数缓慢迭代更新, 公式可表示为

$$\theta_{\text{target}} \leftarrow \tau \cdot \theta_{\text{current}} + (1 - \tau) \cdot \theta_{\text{target}} \quad (43)$$

其中, θ_{target} 与 θ_{current} 分别为目标网络与当前网络的参数, τ 为软更新系数。该机制通过限制目标网络参数的更新幅度, 确保价值估计的平滑性, 适配车-路-空场景中资源状态的动态变化。这一缓慢更新的特性, 能让目标网络输出的动作与价值估计保持稳定, 为后续目标 Q 值的计算提供了可靠的未来价值依据, 避免目标 Q 值因参数突变而出现剧烈波动。

基于时序差分(TD)误差原理, 目标 Q 值融合即时奖励与未来最优动作的价值估计, 其公式可表示为

$$Q_t = R_t + \chi \cdot Q_{\text{target}}(S_{t+1}, \pi_{\text{target}}(S_{t+1})) \cdot (1 - \text{done}) \quad (44)$$

其中, Q_t 为时刻 t 的目标 Q 值, χ 为折扣因子, Q_{target} 为目标Critic网络参数, $\pi_{\text{target}}(S_{t+1})$ 为目标Actor网络在状态 S_{t+1} 下输出的最优联合动作, done 为终止标志。

目标 Q 值是Critic网络的训练基准, 校准其价值评估精度后, Critic会引导Actor网络调整参数, 输出最大化 Q 值的卸载动作, 最终形成兼顾即时收益与长期效率的策略。

VCTOEM框架如图5所示。VCTOEM框架的核心基于MADDPG构建, 融合了车辆聚类模块与无人机轨迹优化模块进行前置状态感知。框架中的每个车辆智能体配备有独立的Actor网络与共同的Critic网络, 在探索噪声的辅助下生成适配场景的卸载决策。接下来会根据任务完成的时延、能耗及系统效用产生相应任务奖励。系统状态、卸载决策与任务奖励共同构成经验数据存入经验回放缓冲区, 为各智能体的Actor网络和Critic网络提供训练数据, 同时借助软更新机制优化目标网络参数, 实现智能体在动态复杂环境中持续迭代优化。

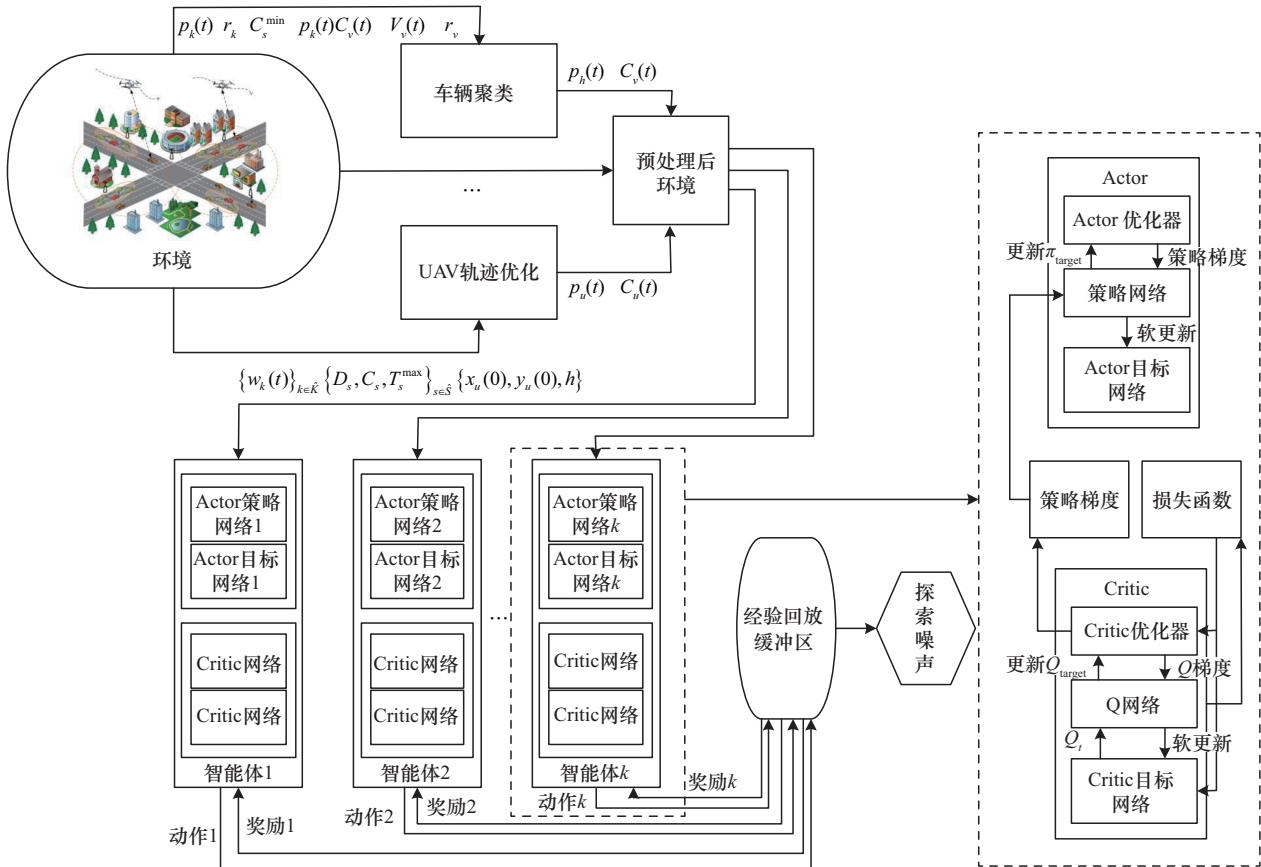


图5 VCTOEM框架

基于MADDPG的边缘卸载优化算法如算法3所示。

算法3 基于MADDPG的边缘卸载优化算法

输入 MADDPG参数、状态空间数据

输出 卸载和资源分配决策 δ

- 1) 初始化 K 个 Actor、1 个 Critic 及对应目标网络, 经验池 D 为空
- 2) 获取初始车辆集群与无人机轨迹
- 3) for 迭代 $n = 1$ 到 N do
- 4) for 时间 $t = 0$ 到 T do
- 5) 采集全局状态 $S^K(t)$
- 6) 各智能体生成动作 $A_k = \delta(S_k) + \text{噪声 } \sigma$
- 7) 执行 A^K , 得奖励 R , 下一状态 S' , 终止标志 done
- 8) 将 $(S^K, A^K, R, S', \text{done})$ 存入 D
- 9) 若 $|D| \geq B$
- 10) 采样 B 条经验, 按式(44)计算目标 Q 值
- 11) 更新 Critic
- 12) 按式(43)软更新目标网络
- 13) 各智能体按策略梯度更新 Actor

14) 定时更新车辆集群与无人机轨迹

15) end for

16) end for

3 仿真实验

3.1 仿真场景与仿真参数设置

本文以人口密集的十字交叉路口为核心环境, 构建了车辆任务协同卸载系统。场景中包含 X 方向与 Y 方向各 1 000 m 的公路, 交叉点坐标为 [500, 500], 两条公路总宽度均为 24 m, 满足双向车道通行需求。交通流方面, 设置 5 辆任务车辆与 200 辆辅助车辆, 任务车辆生成待处理任务, 辅助车辆用于承接卸载任务, 所有车辆越界后从道路入口重新生成, 确保交通流连续稳定。在道路一侧均匀部署了 8 个 RSU, 并部署 2 架无人机, RSU 覆盖范围为 125 m, 无人机初始巡航速度为 8 m/s, 固定高度为 50 m, 通信覆盖范围为 300 m, 通过改进 PSO 算法动态优化巡航轨迹, 提升任务覆盖效率。此外, 参考文献[20], 将任务上传功率设置为 0.1 W, 仿真参数设置如表 2 所示。

表 2 仿真参数设置

参数	预设值	参数	预设值
辅助车辆数量/辆	200	辅助车辆计算能力/GHz	0.7~0.9
任务车辆数量/辆	5	RSU 计算能力/GHz	2
辅助车辆速度/(m·s ⁻¹)	15~30	UAV 计算能力/GHz	8
服务车辆速度/(m·s ⁻¹)	15~30	租用集群频谱单价/Hz ⁻¹	2 × 10 ⁻⁷
子任务大小/bit	400~450	租用 UAV 频谱单价/Hz ⁻¹	1 × 10 ⁻⁵
总的任务大小/bit	1 200~2 250	租用 RSU 频谱单价/Hz ⁻¹	0.5 × 10 ⁻⁵
地面总的频谱带宽/MHz	5	目标网络软更新系数	0.01
车辆发射功率/W	0.1	车辆集群计算单价/Hz ⁻¹	1.5 × 10 ⁻⁷
RSU 计算单价/Hz ⁻¹	0.2 × 10 ⁻⁵	UAV 计算单价/Hz ⁻¹	0.7 × 10 ⁻⁵
折扣因子	0.95	UAV 固定高度/m	50
任务传输奖励/Hz ⁻¹	2 × 10 ⁻⁵	任务计算奖励/bit ⁻¹	1.5 × 10 ⁻⁵

本文基于 MADDPG 的车辆集群协同卸载算法，需要多智能体协同的动作探索，因此采用带衰减的高斯噪声进行动作选择，并参考文献[21-23]进行了以下参数设置：初始噪声标准差为 0.1，逐渐衰减到 0.05，噪声衰减参数为 0.000 05，以平衡探索与利用过程。将折扣因子定义为 0.95，用于对未来奖励进行折扣计算。在神经网络中建立了两个隐藏层，神经元节点数分别为 256 和 128。在训练过程中，最小批大小和回放缓冲器容量分别是 32 和 10 000，同时设置目标网络软更新系数 τ 为 0.01，以实现参数的平滑更新。

3.2 仿真结果分析

3.2.1 参数分析

为了筛选出最优的超参数组合，本文对比了不同 actor 与 critic 学习率对网络平均奖励的影响，结果如图 6 所示。从图 6 可以看出，不同的 actor 与 critic 学习率组合下，网络的平均奖励变化差异显著，这体现了超参数对强化学习模型训练效果的关键影响。当学习率组合相对较大时，模型前期的平均奖励提升速度较快，但收敛后的奖励也偏小，收敛效果有所欠缺；当学习率组合过小时，模型的平均奖励在训练前期的增长节奏偏缓，整体的收敛进程被拉长，训练效率有所降低。相比之下，处于中间区间的学习率组合，既能让平均奖励在训练前期快速上升，又能在后续过程中保持较为平稳的增长趋势，最终的奖励水平也处于较高区间，兼顾了训

练效率与收敛稳定性。因此，最终可选择 actor 学习率为 0.000 1、critic 学习率为 0.003 的组合，以在效率与稳定性之间取得较好的平衡。

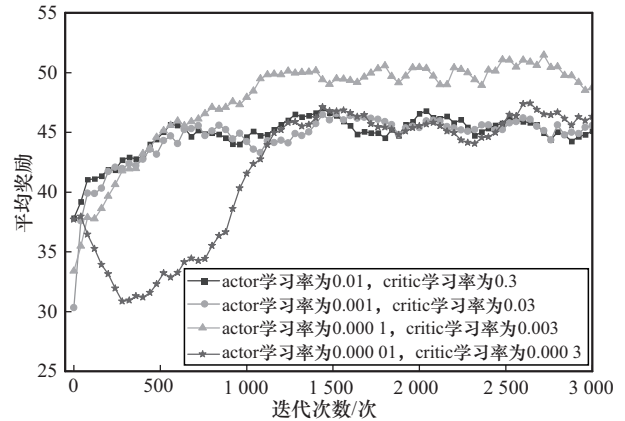


图 6 超参数对网络奖励的影响

图 7 为稳定后的时延和能耗随时延权重变化的曲线。从图 7 可以看出，当时延权重为 0.3 时，平均能耗达到 7.63，处于较高水平，平均时延则为 6.75；当时延权重调整到 0.4 时，平均能耗明显下降，平均时延也同步降低，两者都处于相对理想的状态；随着时延权重继续增大，平均时延虽有小幅起伏但整体保持稳定，平均能耗却出现了明显的回升，性能表现有所下滑。因此，时延权重在 0.4 附近时，这两项指标均处于较低水平，实现了较好的平衡，因此后续实验中将该值作为默认设置。

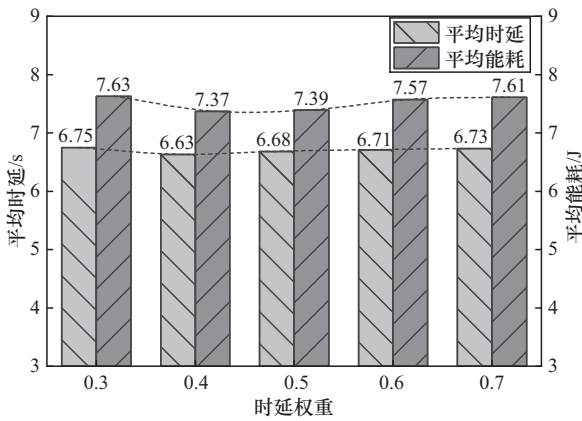


图7 稳定后的时延和能耗随时延权重变化的曲线

3.2.2 卸载方案的性能分析

为了更全面地评估本文所提 VCTOEM 在边缘计算优化场景下的性能, 选取 6 种算法从 3 个角度展开对比实验。

1) 消融实验

为验证本文车辆聚类机制与无人机轨迹优化两个核心模块的可行性, 对其展开了消融实验。

无轨迹优化 (NT-MADDPG): 在 VCTOEM 的基础上, 移除无人机轨迹优化模块, 无人机的位置固定不变。

无车辆聚类 (NC-MADDPG): 在 VCTOEM 的基础上, 移除车辆聚类机制。作为替代, 将离散的辅助车辆作为边缘节点的一种。

2) 训练模型对比

为验证多智能体场景的可行性, 以及 MADDPG 对于此场景的适配性, 本文进行了训练模型对比。

深度确定性策略梯度算法 (DDPG): 为比较“多智能体协同”与“单智能体集中控制”的差异, 选取了文献[24]中的 DDPG 进行训练。此算法是 VCTOEM 的单智能体简化版本, 使用单一的智能体为所有任务车辆做出集中式的卸载与资源分配决策, 而非多智能体协同。

多智能体软演员-评论家算法 (MASAC): 与 MADDPG 同为多智能体模型, 本文保留了原文的场景与方法。在训练时, 选取了文献[25]的 MASAC 模型进行训练。MASAC 的核心参数参照了文献[25]中的参数以及与本文场景进行了适配性设置。

3) 卸载方法对比

为验证卸载方法的可行性, 本文选取了文献[26]中集成 MADDPG、LQR 和 CVXPY 求解器的智能

算法 (MADDPG-LC) 以及文献[27]的联邦多智能体深度确定性策略梯度算法 (Fed-MADDPG) 进行了对比。

MADDPG-LC: 该算法运用了基于多智能体协同的混合优化卸载策略, 使用 LQR 与 CVXPY 求解器进行实时联合优化轨迹设计和资源分配。

Fed-MADDPG: 该算法采用基于联邦训练的无状态信息共享任务卸载方法, 基于联邦训练框架与 MADDPG 构建。

以上 6 种对比算法保证了评估维度的全面性, 接下来将对 7 种算法各个指标的收敛性进行比较。

1) 网络平均奖励收敛性比较

图 8 为不同算法的平均奖励收敛性对比。其中, NT-MADDPG 和 NC-MADDPG 因分别缺少无人机轨迹优化和车辆聚类, 其收敛后的奖励分别小于 VCTOEM 约 10.5% 和 29.7%; 单智能体 DDPG 因无法实现多智能体协同优化, 收敛后的奖励值小于 VCTOEM 约 22.8%; 改变训练模型的 MASAC 与改变轨迹优化方法的 MADDPG-LC 收敛后的奖励值分别小于 VCTOEM 约 9.1% 和 6.7%。其中 VCTOEM 与 NC-MADDPG 之间的显著性能差异, 体现出了车辆聚类机制的关键作用。该机制通过周期性的稳定聚类, 在每个卸载周期内构建出连接可靠、资源聚合的车辆集群, 相较于单一车辆节点方案, 显著增强了高动态场景下任务卸载的稳定性。

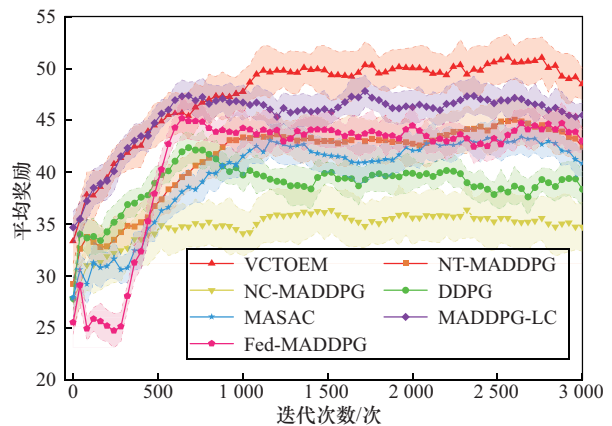


图8 不同算法的平均奖励收敛性对比

2) 平均时延收敛性比较

图 9 为不同算法的平均时延收敛性对比。初始阶段, 所有算法的平均时延均处于较高状态, 差异

较小,但随着迭代推进,VCTOEM的时延下降速度更快,在500次迭代后已明显低于其他算法。NT-MADDPG、NC-MADDPG降幅弱于VCTOEM,这一差异可归因于,NT-MADDPG缺失轨迹优化、NC-MADDPG仅支持单车辆卸载,功能模块的不足限制了二者的时延优化能力;单智能体的DDPG初始时延较低,但收敛后明显高于多智能体的VCTOEM,体现出多智能体框架在边缘卸载任务中的协作优势;MASAC作为同类多智能体算法,其时延表现与NT-MADDPG接近,但仍高于VCTOEM;MADDPG-LC和Fed-MADDPG时延收敛后的水平仍高于VCTOEM,说明VCTOEM的轨迹优化与聚类结合的方案更适配该场景的时延优化需求。收敛后VCTOEM的时延分别低于其他6种算法约4.07%、8.89%、6.03%、3.47%、3.17%、3.01%。

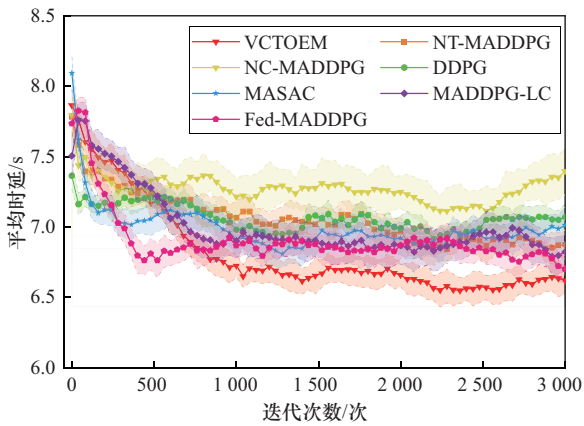


图9 不同算法的平均时延收敛性对比

3) 平均能耗收敛性比较

图10为不同算法的平均能耗收敛性对比。从图10可以看出,初始阶段各算法平均能耗接近7.75~8.00,但随着迭代次数增加,VCTOEM的能耗下降趋势更显著且最终稳定在较低水平,明显优于其他算法。虽然NT-MADDPG以及NC-MADDPG能耗也呈下降趋势,但由于前者缺少无人机轨迹优化,后者省略了车辆聚类,能耗表现始终弱于VCTOEM;单智能体的DDPG能耗下降幅度有限,体现出多智能体框架在边缘卸载任务中的优势;MASAC收敛后的能耗较高则体现了MADDPG对场景的适配性;MADDPG-LC和Fed-MADDPG的能耗表现虽优于NC-MADDPG,却仍不及VCTOEM,这也说明VCTOEM能更高效地降低边

缘计算系统的平均能耗,验证了其在多智能体边缘计算优化场景下的性能优势。收敛后VCTOEM的能耗较其他6种算法分别低约1.56%、2.85%、1.42%、1.36%、1.29%、1.25%。

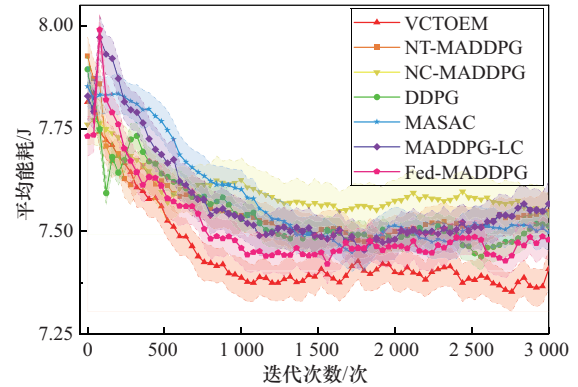


图10 不同算法的平均能耗收敛性对比

4) 系统效用收敛性比较

图11为不同算法的系统效用收敛性对比。VCTOEM在迭代过程中平均系统效用始终领先,随训练推进稳定提升,收敛后效用显著优于其余算法。这得益于其融合了无人机轨迹优化与车辆聚类,精准适配了边缘卸载场景的特性。NT-MADDPG因缺失轨迹优化模块,效用表现明显逊色;NC-MADDPG作为未进行车辆聚类的算法,系统效用与收敛性始终处于较差状态,体现出车辆集群对系统效用的直接影响;DDPG的收敛性印证了多智能体协同对边缘卸载效用提升的必要性;MASAC的效用仍不及VCTOEM,反映其场景适配性不足;MADDPG-LC与Fed-MADDPG使用了MADDPG模型训练,但效用仍落后,凸显了VCTOEM卸载方法的可行性。

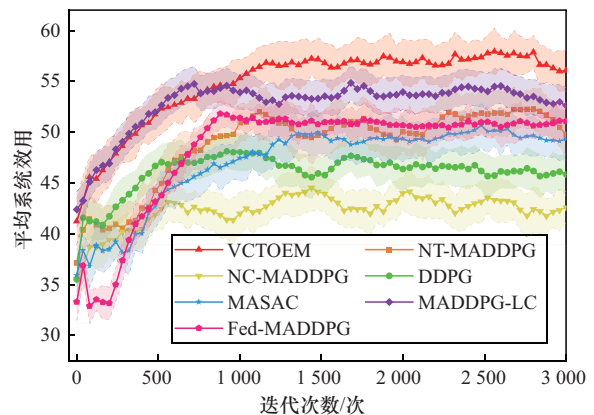


图11 不同算法的系统效用收敛性对比

3.2.3 不同参数的适用性分析

1) UAV 的计算能力

图 12 和图 13 分别为 UAV 计算能力对不同算法奖励和时延的影响。

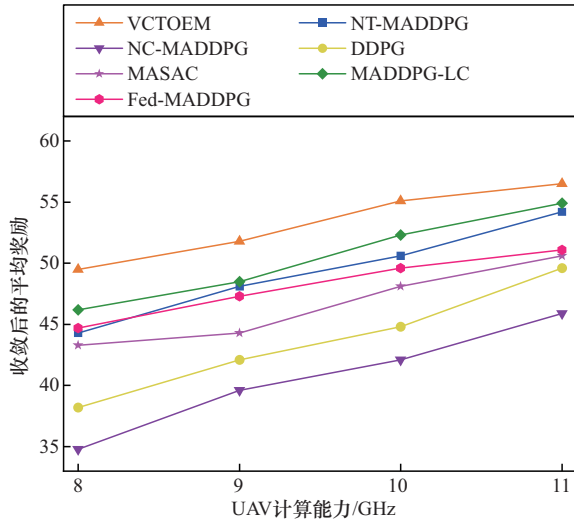


图 12 UAV 计算能力对不同算法奖励的影响

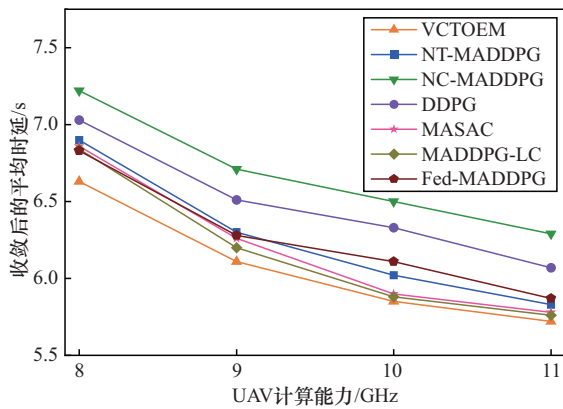


图 13 UAV 计算能力对不同算法时延的影响

从图 12 可以看出，随着 UAV 计算能力从 8 GHz 增至 11 GHz，各算法间的性能差距呈现收窄趋势。在 8 GHz 时，VCTOEM 比表现最差的算法奖励高了约 29.7%；在 11 GHz 时，这一差距缩小至约 17.6%，其他算法之间的奖励差距也在逐渐减小。这一现象是因为高算力对算法短板的弥补效应，当 UAV 算力较低时，VCTOEM 的无人机轨迹优化与车辆聚类能更高效地分配有限资源，凸显协同优势；当算力充足时，硬件性能的冗余降低了对精细化调度的依赖，即使是性能较差的算法，也能通过提升计算速度减少时延与能耗损失。VCTOEM 始终保持最优的性能，为更好地对

比各算法之间的差异，本文将 UAV 的计算能力定义为 8 GHz。

从图 13 可以看出，随着 UAV 计算能力的提高，7 种算法的平均时延均呈现下降趋势，这一变化得益于高算力对任务处理效率的直接提升。低算力阶段，UAV 处理能力有限，VCTOEM 的轨迹优化可以减少任务传输耗时，车辆聚类能整合分散算力，二者通过协同高效分配资源，大幅压缩总时延，策略优势显著；当算力充足时，硬件快速处理任务的能力覆盖了调度差异，即便基准算法缺失轨迹优化或改变轨迹优化方法，也能凭借冗余算力快速完成任务，与 VCTOEM 的时延差距因此缩小。在此过程中，VCTOEM 的时延始终处于最低状态。

2) 辅助车辆的计算能力

由于辅助车辆数量较多，为了保证场景真实性，本文令辅助车辆的计算能力在一个区间里随机生成。图 14 为辅助车辆计算能力对系统收敛后平均奖励的影响。随着辅助车辆计算能力提升，各算法的奖励曲线逐渐靠拢，性能差异持续缩小。这是因为在低算力阶段，VCTOEM 的协同优化设计能高效适配资源约束，NC-MADDPG 因缺少车辆聚类，任务分配效率受限，与 VCTOEM 的差距明显；当算力充足后，NC-MADDPG 的单一车辆节点卸载模式不再受资源瓶颈限制，其劣势得到有效缓解，同时 NT-MADDPG、DDPG 等算法的任务处理效率也随算力提升而改善，各算法间的奖励差距逐步收窄。

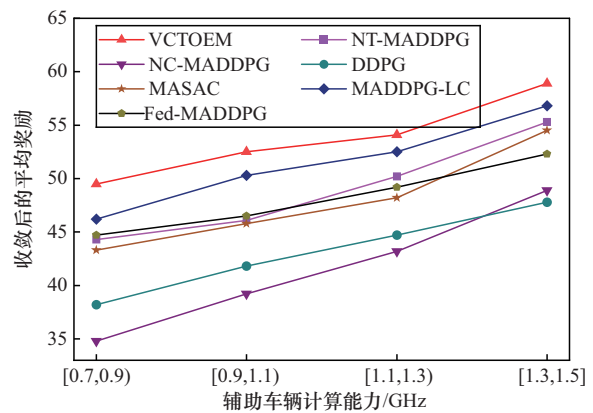


图 14 辅助车辆计算能力对系统收敛后平均奖励的影响

3) 辅助车辆的速度

图 15 为辅助车辆速度对不同算法奖励的影响，

能够充分展现各算法在高动态环境下的适应性。从图 15 可以看出,在车辆速度较低的区间,各算法的奖励水平相对接近, VCTOEM 略占优势,但差距并不显著。而随着车辆速度提升,各算法的奖励呈现明显分化, VCTOEM 的奖励降幅最小,仍保持在较高水平, NC-MADDPG 的奖励则大幅下滑。这说明在车辆高速移动导致的高动态场景中,车辆聚类机制通过连接时间筛选车辆,整合多车辆资源,相比单一车辆作为卸载节点,能够增强系统对高动态环境的适应能力,提升系统在高动态场景下的稳定性。由此可见, VCTOEM 相较于其他方案具有更强的鲁棒性。

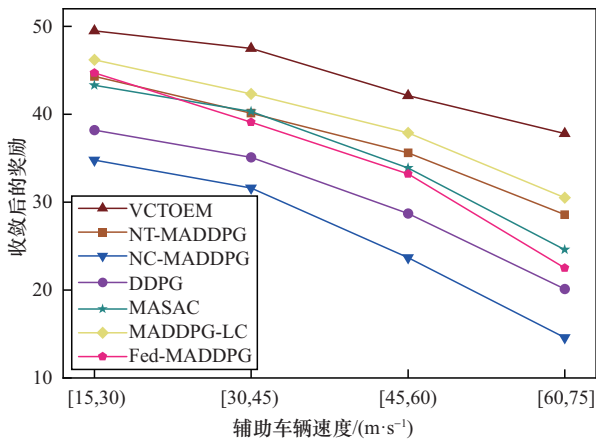


图 15 辅助车辆速度对不同算法奖励的影响

4) 任务大小

图 16 为任务大小对不同算法时延的影响,由于任务较多,为保证场景真实性,将任务大小在一个区间里随机生成。从图 16 可以看出,随着任务大小的增加,7 种算法的时延均出现了增长,当任务大小分布在较小区间时,各算法的时延差异并不明显,这是因为任务量较小时,系统计算与通信资源未达到饱和状态,无论是什么算法,均能满足低负载下的任务卸载需求,算法间的性能优势难以体现。而当任务量变大时,任务对资源的消耗大幅增加,各算法的时延差异显著拉大, VCTOEM 此时展现出优越的性能。这是因为在大任务场景下, VCTOEM 的无人机轨迹优化模块可提升通信效率,车辆聚类环节能通过集群协作分摊计算压力,两者协同作用下,系统资源的利用效率被充分激发,最终在高负载场景下展现出更优的时延表现,这也证明了本文算法更适合当前的场景。

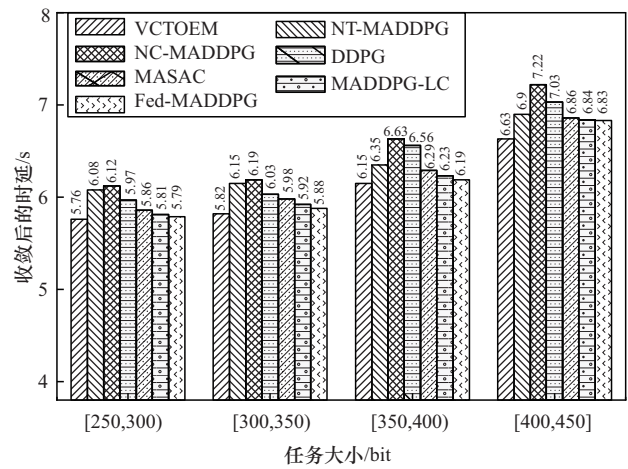


图 16 任务大小对不同算法时延的影响

5) 辅助车辆密度

图 17 为辅助车辆密度对不同算法奖励的影响。从图 17 可以看出,当辅助车辆密度处于较高水平时,7 种算法性能均处于较优状态,尤其是 VCTOEM 展现出了较高的奖励。随着辅助车辆密度逐步降低,所有算法的奖励均呈现明显下降趋势,当密度降至 20 辆时, VCTOEM 的性能甚至低于部分不采用车辆聚类的算法。其原因在于车辆密度过低时无法形成有效簇,任务卸载被迫转向 UAV 与 RSU,这一转变间接加剧了 UAV 的任务负担,引发 UAV 计算资源的瓶颈效应,最终导致算法性能显著下滑。不过本文场景设定在人口密集的城市中心,因此这种情况在实际中发生概率极低。即使出现了这种情况,本文的多边缘节点协同卸载机制也能使性能维持在合理范围内。

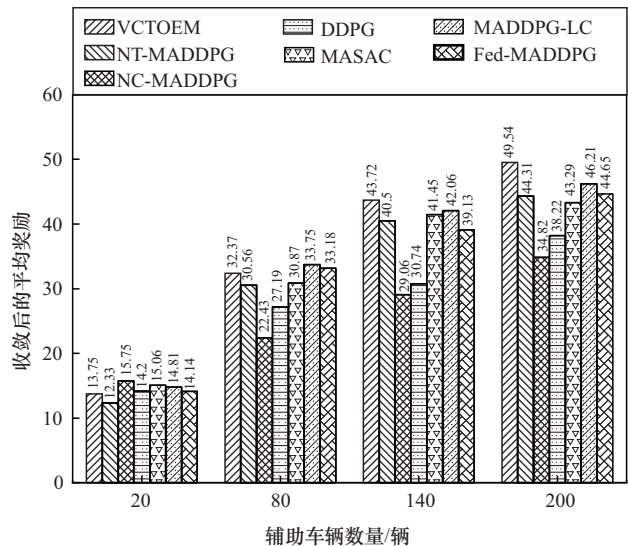


图 17 辅助车辆密度对不同算法奖励的影响

4 结束语

本文针对车联网高动态拓扑下计算密集型、时延敏感型任务的边缘卸载难题,提出了一种车辆聚类及无人机轨迹优化辅助下基于MADDPG的边缘卸载优化算法。该算法采用“车辆聚类-轨迹优化-卸载决策”三级协同框架,首先通过动态车辆聚类机制聚合分散车载资源,提升地面节点协作稳定性,再利用改进PSO算法优化无人机轨迹,填补地面覆盖盲区并降低传输损耗,最后基于MADDPG的多智能体协同决策,实现系统效用、时延与能耗的多目标平衡。然后将VCTOEM与NT-MADDPG、NC-MADDPG、DDPG、MASAC、MADDPG-LC、Fed-MADDPG这6种基线算法进行对比,仿真结果表明,VCTOEM在平均奖励、平均系统效用、平均时延及平均能耗等关键指标上均展现出显著优势,该算法能够有效适配车辆高速移动带来的网络动态变化,在高负载场景下保持优异的收敛稳定性,能够为车-路-空联合边缘计算场景的高效任务卸载提供可靠解决方案。

参考文献:

- [1] Ye W H, Zheng K, Wang Y Y, et al. Federated double deep Q-learning-based computation offloading in mobility-aware vehicle clusters[J]. *IEEE Access*, 2023, 11: 114475-114488.
- [2] Waheed A, Ali Shah M, Mohsin S M, et al. A comprehensive review of computing paradigms, enabling computation offloading and task execution in vehicular networks[J]. *IEEE Access*, 2022, 10: 3580-3600.
- [3] Huang M T, Shen Z R, Zhang G L. Joint spectrum sharing and V₂V/V2I task offloading for vehicular edge computing networks based on coalition formation game[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2024, 25(9): 11918-11934.
- [4] Fan W H, Su Y, Liu J, et al. Joint task offloading and resource allocation for vehicular edge computing based on V2I and V2V modes[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(4): 4277-4292.
- [5] Han X, Tian D X, Sheng Z G, et al. Reliability-aware joint optimization for cooperative vehicular communication and computing[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(8): 5437-5446.
- [6] Guo H, Rui L L, Gao Z P. V2V task offloading algorithm with LSTM-based spatiotemporal trajectory prediction model in SVCNs[J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(10): 11017-11032.
- [7] Sun Y L, Wu Z Y, Meng K, et al. Vehicular task offloading and job scheduling method based on cloud-edge computing[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(12): 14651-14662.
- [8] Liu F M, Chen J, Zhang Q X, et al. Online MEC offloading for V2V networks[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(10): 6097-6109.
- [9] Shi J M, Du J, Shen Y, et al. DRL-based V2V computation offloading for blockchain-enabled vehicular networks[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(7): 3882-3897.
- [10] Zeng M, Zhao Y B, Wang J, et al. Multiple reconfigurable intelligent surfaces aided V2X offloading networks: a federated reinforcement learning-based approach[J]. *IEEE Transactions on Vehicular Technology*, 2026, 75(1): 1281-1294.
- [11] Zhao M X, Zhang R Q, He Z L, et al. Joint optimization of trajectory, offloading, caching, and migration for UAV-assisted MEC[J]. *IEEE Transactions on Mobile Computing*, 2025, 24(3): 1981-1998.
- [12] Yang Z Y, Bi S Z, Zhang Y 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.
- [13] Wang M L, Zhang L, Gao P, et al. Stackelberg-game-based intelligent offloading incentive mechanism for a multi-UAV-assisted mobile-edge computing system[J]. *IEEE Internet of Things Journal*, 2023, 10(17): 15679-15689.
- [14] Dai X X, Xiao Z, Jiang H B, et al. UAV-assisted task offloading in vehicular edge computing networks[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(4): 2520-2534.
- [15] Bai Z Y, Lin Y F, Cao Y, et al. Delay-aware cooperative task offloading for multi-UAV enabled edge-cloud computing[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(2): 1034-1049.
- [16] Xu B, Kuang Z F, Gao J, et al. Joint offloading decision and trajectory design for UAV-enabled edge computing with task dependency[J]. *IEEE Transactions on Wireless Communications*, 2023, 22(8): 5043-5055.
- [17] Guo H Z, Wang Y T, Liu J J, et al. Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond[J]. *IEEE Transactions on Wireless Communications*, 2024, 23(1): 347-359.
- [18] Chen G, Zhou Y X, Xu X J, et al. A multi-aerial base station assisted joint computation offloading algorithm based on D3QN in edge VANETs[J]. *Ad Hoc Networks*, 2023, 142: 103098.
- [19] Zhang Y, Mou Z Y, Gao F F, et al. UAV-enabled secure communications by multi-agent deep reinforcement learning[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(10): 11599-11611.
- [20] Li H, Qu L, Chen W F, et al. Fairness-aware joint optimization of 3D trajectory and task offloading in multi-UAV edge computing systems[C]// *Proceedings of the 2025 21st International Conference on the Design*

of Reliable Communication Networks (DRCN). Piscataway: IEEE Press, 2025: 1-5.

- [21] Kumar A S, Zhao L, Fernando X. Task offloading and resource allocation in vehicular networks: a Lyapunov-based deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2023, 72(10): 13360-13373.
- [22] Chen G, Zhang Y M, Song Z H, et al. Distributed grade-clustering assisted computing offloading optimization for cooperative vehicular-edge networks with near-field communications[J]. IEEE Transactions on Cognitive Communications and Networking, 2026, 12: 4089-4104.
- [23] Peng H X, Shen X M. 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.
- [24] Qiu B, Wang Y X, Xiao H L, et al. Deep reinforcement learning-based adaptive computation offloading and power allocation in vehicular edge computing networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(10): 13339-13349.
- [25] Wu D P, Liu T, Li Z D, et al. Delay-aware edge-terminal collaboration in green Internet of vehicles: a multiagent soft actor-critic approach[J]. IEEE Transactions on Green Communications and Networking, 2023, 7(2): 1090-1102.
- [26] Wang Z W, Wang H W, Liu L H, et al. Dynamic trajectory design for multi-UAV-assisted mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2025, 74(3): 4684-4697.
- [27] Lei S X, Tang H J, Li C Y, et al. Federated MADDPG-based collaborative scheduling strategy in vehicular edge computing[J]. IEEE Transactions on Mobile Computing, 2026, 25(1): 54-66.

[作者简介]



陈赓 (1984-), 男, 山东潍坊人, 博士, 山东科技大学教授、博士生导师, 主要研究方向为异构网络、泛在网络和软件定义移动网络无线资源管理和优化算法。



夏聪慧 (2002-), 女, 山东菏泽人, 山东科技大学硕士生, 主要研究方向为车联网和移动边缘计算中的计算卸载。



孔令志 (2001-), 男, 山东德州人, 山东科技大学硕士生, 主要研究方向为车联网和移动边缘计算中的计算卸载。



曾庆田 (1976-), 男, 山东潍坊人, 博士, 山东科技大学教授、博士生导师, 主要研究方向为Petri网、过程挖掘和知识管理。