

数字孪生辅助下基于 D3QN 的车载网络协同卸载算法

陈赓, 宋政翰, 夏聪慧, 曾庆田

(山东科技大学电子信息工程学院, 山东 青岛 266590)

摘要: 为应对车辆边缘计算中拓扑高动态、任务多样性及低延迟约束等挑战, 提出一种数字孪生辅助下基于 D3QN 的车载网络协同卸载算法。首先, 构建数字孪生网络以实现交通状态的动态建模, 融合车辆时空与资源信息, 区别于传统静态聚类策略, 提升协作分簇稳定性并缩小策略搜索空间。其次, 基于任务的可分解性构建 2 种任务模型, 设计混合卸载策略以精准适配动态现实需求。此外, 设计基于 D3QN 的车辆集群协同卸载算法, 构建双网络架构解耦动作-目标评估以抑制 Q 值偏差, 加速策略收敛, 实现效用-时延均衡。仿真结果表明, 所提方案在高动态与高负载场景中可显著降低任务处理时延, 相较于其他 6 种基线算法分别实现 5.32%、8.54%、1.47%、11.2%、68.51% 和 103.15% 的平均系统效用提升。

关键词: 车辆边缘计算; 数字孪生; 车辆集群; 深度强化学习; 任务多样性

中图分类号: TN929.5

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025144

D3QN-based collaborative offloading algorithm for vehicular networks assisted by digital twins

CHEN Geng, SONG Zhenghan, XIA Conghui, ZENG Qingtian

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

Abstract: To address the challenges of highly dynamic topologies, task diversity, and low-latency constraints in vehicular edge computing, a D3QN-based collaborative offloading algorithm assisted by digital twin technology was proposed. Firstly, a digital Siamese network was constructed to realize the dynamic modeling of traffic state, which integrated vehicle spatio-temporal and resource information, was different from the traditional static clustering strategy, improved the stability of collaborative clustering and reduced the strategy search space. Next, based on task decomposability, two types of task models were established, and a hybrid offloading strategy was devised to accurately adapt to dynamic real-world demands. Furthermore, a D3QN-based collaborative offloading algorithm for vehicle clusters was developed, utilizing a dual-network architecture to decouple action and target evaluations, thereby suppressing Q -value bias, accelerating policy convergence, and achieving a balance between utility and latency. The simulation results demonstrate that the proposed scheme can significantly reduce task processing latency in high-dynamic and high-load scenarios, and achieves an average system utility improvement of 5.32%, 8.54%, 1.47%, 11.2%, 68.51%, and 103.15% compared to the other six baseline algorithms, respectively.

Keywords: vehicular edge computing, digital twin, vehicle cluster, deep reinforcement learning, task diversity

收稿日期: 2025-05-27; 修回日期: 2025-08-14

通信作者: 曾庆田, qtzeng@sust.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基金资助项目(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

车辆边缘计算 (VEC, vehicular edge computing) 作为移动边缘计算 (MEC, mobile edge computing) 在智能交通领域的重要应用, 近年来受到了学术界和工业界的广泛关注。VEC 通过将计算任务从车辆卸载到路边单元 (RSU, roadside unit), 有效缓解了车辆本地计算资源的不足, 为智能交通系统提供了强大的计算支持^[1]。然而传统卸载到 RSU 的方法往往带来额外的时延和能量消耗, 车辆的卸载经常会造成 RSU 计算资源分配不均, 使 RSU 的计算负担过重, 增加额外的任务等待时间^[2]。同时路边有许多有额外计算资源的车辆, 因此计算资源有限的车辆可以通过车辆通信将任务卸载到周围的空闲车辆中来执行, 提高资源的利用率。此外, 与固定位置的 RSU 相比, 在同一方向上行驶的 2 辆车之间的链接持续时间更长, 因为车辆之间的相对速度更小^[3]。

车辆到车辆 (V2V, vehicle-to-vehicle) 计算卸载作为一种有前途的机制能有效地减轻 RSU 的工作负载^[4], 然而, 它仍然面临着一些挑战。交通场景中的网络拓扑结构高度动态, 车辆和 RSU 的连接状态随时间快速变化, 这对任务卸载的稳定性和实时性提出了极高要求^[5]; 并且大多数现有的车辆应用需要实时响应, 这意味着任务是时间敏感的^[6]。更为重要的是, 道路上的车辆总是愿意帮助他人完成任务是不现实的, 因此车辆在享受边缘计算服务时, 需要向 RSU 或服务车辆支付一定的费用, 这使任务卸载不仅要考虑时延优化, 还需要兼顾成本控制。文献[5]提出了一种基于资源预测的卸载策略, 考虑了邻近车辆的空闲计算资源和车辆的移动性, 并将卸载问题描述为顺序决策问题, 从而解决了所提出的资源分配困境, 但预测精度有限, 难以适应高度动态的环境。文献[7]提出了一种基于邻近车辆的任务卸载框架, 考虑了因车辆位置变化而导致的迁移成本问题, 并引入贝叶斯推理结合深度强化学习 (DRL, deep reinforcement learning) 算法来解决所提出的顺序决策问题, 但未考虑车辆高速移动导致的网络连接不稳定问题。此外, 邻近车辆参与任务卸载的意愿往往受到资源消耗和缺乏激励的限制, 文献[8]设计了一种基于车辆类型的契约激励机制, 激励车辆自行共享计算资源。文献[9]提出了一种基于信用积分的激励机制,

通过奖励参与任务的车辆用户信用积分以兑换服务或资源, 从而提升其参与度。但现有的 V2V 研究大多只研究如何将 V2V 卸载问题转化为一个减少时延的序贯决策问题, 而没有从根本上考虑如何解决上述问题, 卸载效率仍然受到动态车辆网络的限制。

为了解决上述问题, 在任务卸载之前构建指定的预备空间作为预处理过程是至关重要的^[10]。数字孪生 (DT, digital twin) 技术可以创建物理对象的多尺度数字制图^[11], 为 VEC 任务卸载提供了新的解决方案。文献[12]提出了一种自适应 DT 赋能的 VEC 架构, 通过构建车辆与 RSU 的虚拟模型并引入双闭环机制实现实时映射与智能调度, 利用 DRL 优化任务卸载和资源分配, 从而显著降低整体任务处理时延。文献^[13]提出了一种面向工业自动化的 DT 赋能 MEC 架构, 通过迭代优化物联网设备的发射功率、用户关联和任务卸载策略, 最小化端到端时延。文献^[14]提出了一种新型工业物联网架构, 通过将 DT 捕获的工业特征与联邦学习相结合, 提升了资源受限环境下的学习性能。事实上, 上述针对 VEC 的 DT 解决方案还处于早期阶段, 还有很大的提升空间, 当前, 仍然缺乏使用 DT 进行预测以优化卸载决策的研究, 且基于 DT 的卸载策略多为二元模式, 无法充分利用资源的灵活性。

在 VEC 场景中, 任务的多样性和复杂性对卸载策略提出了更高要求。传统研究通常将任务视为单一整体处理, 忽视了任务的可分割性和非线性特性。近年来, 研究者开始关注任务的可分割性, Ma 等^[15]提出的基于任务分割的卸载框架, 通过将大规模计算任务拆分为多个子任务并行处理, 显著降低了任务完成时间。然而, 这类研究主要针对线性可分割任务, 未能充分考虑非线性任务的特性。针对这一问题, Ye 等^[16]提出了一种基于图分割的任务分解方法, 将非线性任务表示为有向无环图, 并通过优化子任务间的依赖关系提高处理效率。Guo 等^[17]则进一步研究了任务可分离性和子任务间的数据依赖性, 提出了 DT 辅助的混合任务卸载方案, 结合启发式贪婪和基于深度 Q 网络 (DQN, deep Q-network) 方法优化资源分配。这些研究为处理复杂任务提供了多样化解决方案。上述文章将任务简单分为计算密集型任务和数据密集型任务, 缺少任务处理的公平性和处理时延的解释, 更多地

集中于线性任务的卸载优化，因此未能充分考虑实际场景中任务的多样性。

针对以上挑战，本文主要的研究工作如下。

1)设计了DT辅助下基于D3QN (dueling double deep Q-network) 的车载网络协同卸载算法，该算法以平均网络效用和时延为优化目标，建立了一个非凸、非线性的NP-hard优化问题。该框架基于任务的可分解性建立2种任务模型，设计混合卸载策略，以适应实际生活中不同的计算需求，为VEC环境下的任务卸载提供了科学、灵活的解决方案。

2)基于云端构建的DT虚拟网络，设计了一种基于万有引力公式的车辆聚类算法，通过将车辆间的相对位置、相对速度和当前负载等关键因素量化为引力值，显著提高了车辆聚类的准确性和稳定性，减小了决策空间，为后续任务卸载决策提供了更可靠的网络拓扑。

3)提出了一种基于D3QN的车辆集群协同卸载算法。该算法结合在云端构建的DT模型，RSU从云服务器中提取交通网络状态、任务特征和资源可用性等关键信息。利用这些信息，构建基于D3QN算法双网络架构的优化模型，通过分离动作评估与目标值计算抑制Q值偏差，提高学习效率，并将贪心策略和付费服务相结合，使智能体与环境不断交互，进行探索和学习，优化任务卸载策略和计算资源分配，实现平均系统效用与时延的双重目标优化。

4)通过仿真实验验证了本文算法的有效性。与

其他6种基线算法相比，本文算法在长期平均网络效用和处理时延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

DT增强型车辆边缘计算网络如图1所示，在一个计算任务密集的场景中，为了减少RSU的计算压力，考虑将部分计算任务卸载到周围空闲车辆，车辆协同处理任务，可以实现资源的共享与优化，提高整体计算效率。车辆集合表示为 $N = \{u_1, u_2, \dots, u_N\}$ ，任务车辆可将计算任务卸载至空闲车辆集群或路侧单元RSU，其中任务车辆集群表示为 $TV = \{TV_1, TV_2, \dots, TV_R\}$ ，服务车辆表示为 $SV = \{SV_1, SV_2, \dots, SV_F\}$ ，路侧单元RSU集合用 $Q = \{r_1, r_2, \dots, r_Q\}$ 表示。在DT网络的构建过程中，各RSU会以固定周期通过车与基础设施链路采集系统状态，并通过光纤链路实时上传至云服务器，由云端构建和维护DT模型。车辆 u_n 在数字孪生网络 (DTN, digital twin network) 中表示为 $\{C_n, v_n, l_n, F_n\}$ ，分别表示任务负载、速度、位置和最大处理能力，该状态向量作为聚类算法的输入基础，上传至云端孪生模块进行聚类。RSU的DT信息可以表示为 $\{l_q, F_q\}$ ，表示位置和当前计算能力。在进行车辆聚类后，车辆集群表示为 $VD = \{VD_1, VD_2, \dots, VD_K\}$ ，

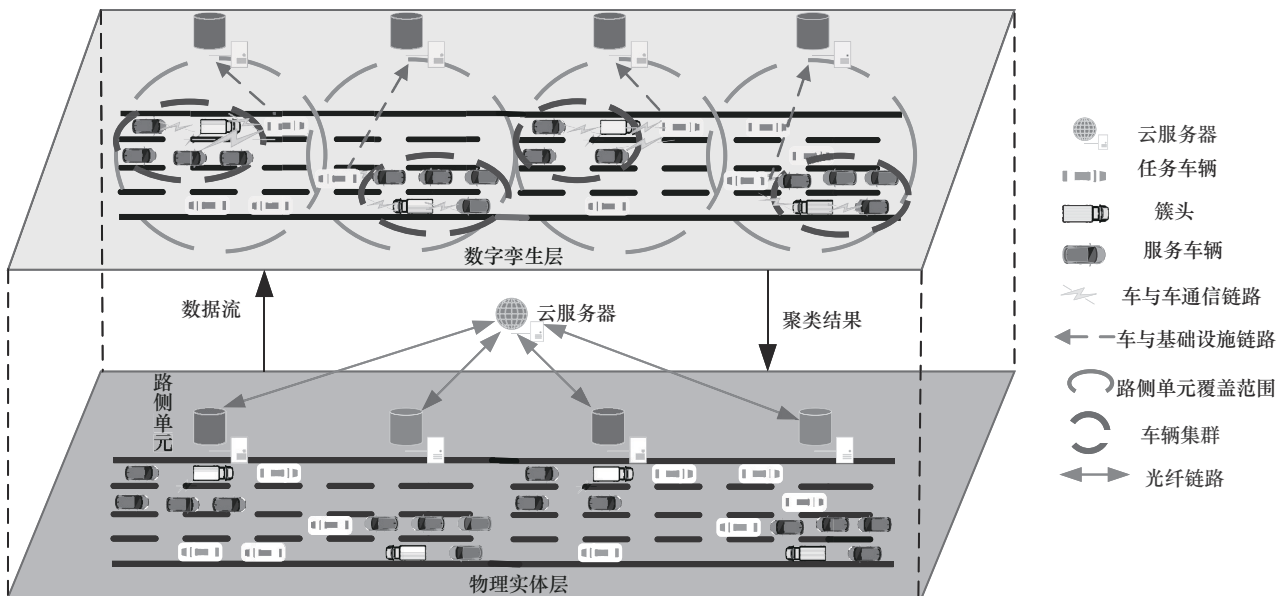


图1 DT增强型车辆边缘计算网络

每个集群 VD_k 包含 W 个成员和一个簇头 CH_k , 簇成员表示为 $VD_k = \{u_{k1}, u_{k2}, \dots, u_{kw}\}$, 该聚类结构由云端构建的 DT 模型生成并回传至 RSU, RSU 根据聚类结果优化任务的调度与资源分配, 从而实现高效的计算卸载。同时在云端, 孪生模块会对来自多个 RSU 的原始数据流进行统一聚合与对齐, 这一过程不仅确保了数据的一致性和时效性, 还能为全局任务调度提供更加全面的视图。考虑到 RSU 上报存在异步性和网络时延, 系统引入全局统一时间戳机制 τ_t , 所有车辆状态在上传后会基于当前轮次的 τ_t 进行时间同步, 避免因不同时间点的数据上传导致的冲突或不一致性。晚于该时间戳的状态会缓存在缓存区, 早于 $\tau_t - \delta$ 的信息则被丢弃, 以确保后续聚类使用的数据时空一致。同时孪生模块还使用滑动时间窗口机制: 对每辆车 u_n , 记录其在最近 T_w 个时间片内的连续状态, 构建多时刻状态序列向量 $\psi_n = [s_n(t - T_w + 1), s_n(t - T_w + 2), \dots, s_n(t)]$, 用于短期行为趋势建模。该设计中 DT 不仅起到了数据融合与聚类计算的执行载体作用, 更在系统层面保障了多源动态数据的时序协同与结构一致性。相比仅依赖本地 RSU 的方法, DT 通过统一建模机制有效弥补了分布式感知时延带来的信息差异, 增强了聚类策略对车辆高动态行为的适应能力。主要符号和变量如表 1 所示。

1.2 任务模型

在实际场景中, 可分任务和非线性任务广泛存在。可分任务如交通流量预测、地图构建和车载内

容分发, 可拆分为多个子任务并行处理, 以提高计算效率和资源利用率^[18]。而非线性任务如实时路径规划和紧急事件响应等, 受到严格的时序约束, 需要整体处理以确保任务的完整性。这 2 类任务需要高效的资源调度和卸载策略来优化任务执行。定义 $e = \{e_1, \dots, e_R\}$ 为任务车辆产生的任务集合。系统在时隙中运行, 每个用户在一个时隙中只产生一个任务, 任务可分成多个不同类型的子任务。本文假设非线性任务的输入输出任务在本地处理, 因此任务 e_{r0} 和 e_{rn} 的大小数据量在本文忽略, 旨在聚焦集群并行计算性能优化, 减少通信过程对调度策略的干扰, 确保任务执行的完整性。定义 $e_r = \{e_{r1}, \dots, e_{rG}\}$ 为 e_r 的子任务集合。其中, e_{ri} 代表可分任务, 可按比例分解; e_{rj} 代表不可分的非线性任务, 其输入输出可忽略, 且本文仅考虑并行计算的非线性任务模型, 以支持高效的任务卸载与资源调度。任务车辆产生包含 I 个可分割子任务和 J 个不可分割子任务, 总任务数为 G 。任务 e_r 可用组元 $\{D_r, \tau_r, C_r\}$ 描述, 分别表示子任务数据大小、最大容忍时间和所需 CPU 周期数。简化的任务结构如图 2 所示。

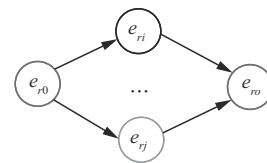


图 2 简化的任务结构

表 1 主要符号和变量

参数	含义	参数	含义
TV	任务车辆集合	SV	服务车辆集合
N	车辆集合	Q	路侧单元集合
VD	车簇集合	CH _k	车簇 VD _k 的簇头
e	任务集合	e _{ri}	e _r 中的可分割任务
e _{rj}	e _r 中的非线性任务	t _r ^{loc}	本地处理时延
t _{rgkw} ^{tran}	卸载到车辆集群时子任务 e _{rg} 的传输时间	r _{ij}	车辆 i 和节点 j 传输速率
t _{rg} ^{kw}	卸载到车辆集群时子任务 e _{rg} 的计算时间	t _{rg}	卸载到车辆集群时子任务 e _{rg} 处理时间
t _{rgq}	卸载到 RSU 时子任务 e _{rg} 的处理时间	t _{rg}	子任务 e _{rg} 的处理时间
t _r	任务 e _r 的处理时间	U _{com} ^r	任务计算效用
U _{tra} ^r	通信效用	c _{rkw}	车辆 u _{kw} 贡献的计算资源比例
d _{rg}	RSU 贡献的计算资源比例	b _r	表示任务车辆 TV _r 是否选择卸载
b _{rg}	子任务 e _{rg} 的卸载选择	U _r [']	TV _r 生成任务总效用

1.3 任务卸载模型

任务车辆可以将任务卸载到 RSU 和车簇进行计算, 或者进行车辆的本地计算。当任务进行本地处理时, 任务车辆 TV_r 本地可用计算资源的 CPU 频率为 f_r , 因此, 车辆的本地处理时延可表示为

$$t_r^{\text{loc}} = \frac{\sum_{i=1}^w C_{ri}}{f_r} \quad (1)$$

任务车辆选择将可分任务 e_{ri} 卸载到车簇时, 仅选择通信范围内的簇内车辆直接卸载, 以提高传输效率并降低时延。车簇中与任务车辆 TV_r 直接相连的车辆有 w_1 辆, 任务分配比为 $\rho = \{\rho_1 \cdots \rho_{w_1}\}$ 。对于不可分割的非线性任务 e_{ij} , 当子任务数量超过直接通信车辆数量时, 部分子任务需通过簇头中继转发至其他簇内节点完成卸载。任务传输时间为

$$t_{rgkw}^{\text{tran}} = \begin{cases} \frac{\rho_{rw} D_{rg}^{kw}}{r_{rkw}}, g \in I \\ \frac{D_{rg}^{kw}}{r_{r\text{CH}_k}} + \frac{D_{rg}^{kw}}{r_{kw\text{CH}_k}}, g \in J \end{cases} \quad (2)$$

为了明确数据传输速率的计算, 定义第 i 个车辆与边缘节点 j 之间的传输速率为 r_{ij} , 可表示为

$$r_{ij} = B_{ij} \text{lb} \left(1 + \frac{p_i d_{ij}^\beta h_{ij}^2}{N_0 + I_{ij}} \right) \quad (3)$$

其中, B_{ij} 为带宽, p_i 是车辆 u_i 的数据传输功率, d_{ij} 为距离, h_{ij} 为信道增益, N_0 表示信道内的高斯噪声功率, I_{ij} 为干扰项。在考虑了等待时间后, 服务车辆的计算时间可表示为

$$t_{rg}^{kw} = \begin{cases} t_{\text{wait}} + \frac{\rho_{rw} C_{rg}^{kw}}{c_{rkw} f_{kw}}, g \in I \\ t_{\text{wait}} + \frac{C_{rg}^{kw}}{c_{rkw} f_{kw}}, g \in J \end{cases} \quad (4)$$

其中, f_{kw} 表示服务车辆 u_{kw} 的计算资源, c_{rkw} 表示贡献计算资源的比例。由于任务在车辆集群中是并行计算的, 任务 e_{rg} 的处理时间为

$$t_{rkg} = \max(t_{rg}^{kw} + t_{rgkw}^{\text{tran}}), \forall w \in W \quad (5)$$

当任务卸载到 RSU 时, 由于在计算频率固定的情况下, 并行处理的时间通常大于或等于串行处理的时间, 因此将并行处理自动转换为串行处理, 以最大化任务处理效率。RSU 处理 2 种类型子任务的过程基本一致, 因此 RSU 计算子任务 e_{rg} 所需的

时间为

$$t_{rgq} = \frac{C_{rg}}{d_{rq} f_{rq}} + t_{rq}^{\text{wait}} + \frac{D_{rg}}{r_{rq}} \quad (6)$$

其中, f_{rq} 是 RSU 的计算资源, d_{rq} 表示 RSU 为任务车辆分配的计算资源比例, t_{rq}^{wait} 为任务等待时间。

综上所述, 当任务车辆 TV_r 决定进行任务卸载时, 子任务 e_{rg} 的处理时间为

$$t_{rg} = b_{rg} t_{rkg} + (1 - b_{rg}) t_{rgq} \quad (7)$$

任务车辆 TV_r 的任务 e_r 的处理总时间为

$$t_r = b_r \max(t_{rg}) + (1 - b_r) t_r^{\text{loc}} \quad (8)$$

其中, b_r 表示任务车辆 TV_r 是否选择卸载, $b_r \in \{0, 1\}$, 当 $b_r = 0$ 时, 表示车辆选择本地处理子任务; 当 $b_r = 1$ 时, 表示车辆选择卸载处理子任务。此外, $b_{rg} \in \{0, 1\}$, 当 $b_{rg} = 1$ 时, 表示车辆将子任务 e_{rg} 卸载给车辆集群; 反之, 则卸载给地面 RSU。

1.4 问题建模

为激励车辆和 RSU 积极参与任务卸载, 设计了奖励机制, 将效用分为通信效用和计算效用。车辆卸载任务的价格包括计算任务所需的价格和租用计算资源的价格。计算效用为

$$U_{\text{com}}^r = p_1 C_r - b_r [(1 - b_{rg}) d_{rq} f_{rq} (p_2 + p_q) + \sum_{w=1}^{w_2} b_{rg} c_{rkw} f_{kw} (p_3 + p_{sv})], \forall g \in G \quad (9)$$

其中, p_1 为用户 TV_r 产生任务的单位价格, p_2 和 p_3 为 RSU 和服务车辆的计算资源的回报单价, p_q 和 p_{sv} 为 RSU 和服务车辆的计算资源租用单价。通信效用为

$$U_{\text{tra}}^r = p_4 D_r - b_r \left[\sum_{w=1}^{w_2} b_{rg} p_5 r_{rw} + (1 - b_{rg}) p_6 r_{rq} \right], \forall g \in G \quad (10)$$

其中, p_4 是用户向服务器传输任务的传输单价, p_5 和 p_6 是服务车辆和 RSU 租用频谱资源的单价。综上, 车辆的 TV_r 生成任务的总效用 U_r' 为

$$U_r' = U_{\text{com}}^r + U_{\text{tra}}^r \quad (11)$$

本文的优化目标是在 RSU 代理中完成任务的卸载决策和计算资源的分配决策, 最终使系统在卸载处理任务的过程中获得最大的效用, 同时使总时延最小。在时隙中对任务卸载的总效用和总时延进行加权求和, 优化问题可表述为

$$\begin{aligned}
J: \max_{b, c_{rkw}, d_{rq}} \lim_{t \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T [(1-\alpha) \sum_{r=1}^R U_r' - at_r], \alpha \in (0,1) \\
\text{s.t. } C_1: b_r, b_{rg} \in \{0,1\}, \forall r \in R \\
C_2: c_{rkw}, d_{rq} \in \{0,1\}, \forall r \in R \\
C_3: \sum_{r=1}^R d_{rq} f_{rq} \leq f_q, \sum_{r=1}^R c_{rkw} f_{rkw} \leq f_{kw}, \forall r \in R, \\
\forall w \in W \\
C_4: tr \leq \tau_r, \forall r \in R \\
C_5: p_{kw}^{\text{com}} < p_q^{\text{com}} \quad (12)
\end{aligned}$$

其中, α 是用于调整车辆效用和时延优化权重的归一化因子; C_1 表示用户的卸载决策; C_2 表示分配给任务车辆 TV_r 的服务车辆和 RSU 的计算资源比率; C_3 确保服务车辆和 RSU 分配给用户的计算资源不超过其总计算资源; C_4 表示任务的完成时间应小于任务的最大容忍时间; C_5 表示 RSU 计算资源的价格高于服务车辆, 以鼓励用户充分利用周围车辆资源, 避免 RSU 过载。

一般来说求解目标函数 J 非常具有挑战性, 该问题属于 NP-hard 问题, 需要同时优化问题包含离散卸载决策变量 ($b_r, b_{rg} \in \{0,1\}$) 与连续资源分配变量 ($c_{rkw}, d_{rq} \in (0,1)$), 且目标函数 J 包括多个非线性耦合项, 例如 U_r' 中的 $b_r(1-b_{rg})d_{rg}$ 等项。此外, 该问题还受到多维资源约束 (如 RSU 和服务车辆的计算能力) 与时延约束的共同限制 (对应约束 C_3 和 C_4)。为证明该问题的 NP-hard 性, 采用从经典多维背包问题^[19] (MDKP, multidimensional knapsack problem) 出发的多项式时间归约方法: 将每个卸载任务类比为背包问题中的一个物品, 假设背包容量为 C_{a_m} , 共有 N 物品, 每个物品都有相应的价值 B_{l_n} 和重量 W_{e_n} , 每个物品可以选择是否放入背包 (1 或 0)。目标是找到一个包含在 N 中的子集 N' , 使总价值 $\sum_{n \in N'} B_{l_n}$ 最大, 同时背包的总重量 $\sum_{n \in N'} W_{e_n}$ 不超过背包的总容量 C_{a_m} , 即 $\sum_{n \in N'} W_{e_n} \leq C_{a_m}$ 场景中放入背包的选择为 $b_r = 1$, 否则不会占用背包空间。 B_{l_n} 的值表示为生成任务的总效用 U_r' 和时延 t_r 的加权和。同时, 选择放入背包的物品重量相当于 RSU 或服务车辆分配给任务车辆的计算资源, 系统资源 $d_{rq} f_{rq}$ 和 $c_{rkw} f_{rkw}$ 对应于背包的容量限制, 背包的总重量对应的是基站的有限资源。最终目标策略的子集被发现, 允许最大限度地利用有限资

源, 以最大限度地提高总价值。MDKP 已被计算复杂性理论证明为 NP-hard 问题, 且上述归约过程在输入规模为任务数和服务车辆数的多项式时间内完成, 因此可严格推得车载网络协同计算卸载优化问题亦属于 NP-hard 问题。

2 DT 辅助下基于 D3QN 的车载网络协同卸载算法

2.1 DT 增强型车辆聚类算法

本文设计了一种 DTN 辅助下的增强型车辆聚合方案。该系统部署于边缘云, 每个 RSU 实时收集其周围车辆的计算资源、任务处理状态和通信拓扑信息, 并通过光纤链路上传至云端, 云端对这些多源异构数据经过聚合、时间同步与简单预测处理, 用于在云端执行车辆聚类计算, 生成每辆车的簇归属方案, 并将结果回传至对应 RSU, 作为后续协同卸载算法的输入。借鉴牛顿万有引力公式, 构建基于“引力”的车辆集群机制^[20]。为确保车辆集群的稳定性, 只有行驶方向一致的车辆可划入同一集群。传统依赖 RSU 本地感知的方式, 易受到网络异步上报、状态滞后等因素影响, 难以实时掌握跨区域车辆的整体资源分布和运动趋势。DT 系统通过集中整合跨区域车辆动态信息, 并引入相对移动性等趋势性指标, 实现了更稳定、更前瞻的聚类决策。相比单纯基于位置或瞬时状态的聚类方式, DT 支撑下的聚类过程更具前瞻性与连续性, 可显著减少簇结构频繁变动引起的卸载策略波动。

具体成簇过程如下。首先, 为任务车辆 TV_r 建立其通信范围内服务车辆的列表 $VU_r = \{u_{r1}, u_{r2}, \dots, u_{rH}\}$, 共 H 辆车。计算每辆车的簇头因子, 对于服务车辆 SV_f 其通信范围内的车辆表示为 $JU_r = \{u_1^f, u_2^f, \dots, u_j^f\}$, 共 J 辆车。其中服务车辆 SV_f 与邻居节点的相对移动性和相对距离分别表示为

$$RV_f = \frac{1}{J} \sum_{j=1}^J |v_f - v_j|, RD_f = \frac{1}{J} \sum_{j=1}^J |d_{fj}| \quad (13)$$

簇头因子计算为

$$\begin{aligned}
C_f &= \alpha_1 RV_f + \alpha_2 RD_f + \alpha_3 \frac{1}{J} \\
\alpha_1, \alpha_2, \alpha_3 &\in (0,1), \alpha_1 + \alpha_2 + \alpha_3 = 1 \quad (14)
\end{aligned}$$

其中, $\alpha_1, \alpha_2, \alpha_3$ 为调整相对移动性和相对距离权重的因子。按 C_f 值排序, 选择前 K 辆车作为簇头, 形

成初始簇集合 $G = \{g_1, g_2, \dots, g_K\}$ 。基于引力模型为初始簇分配成员，这里定义车辆 u_i 和 u_j 间引力表示为

$$F_{ij} = \frac{\beta M_i M_j}{(RV_{ij} + RD_{ij})^2} \quad (15)$$

其中， M_i 、 M_j 为有效质量，且 $M = \frac{F}{C + \varepsilon}$ ， ε 为极小常数； β 为引力系数，用于平衡分子和分母之间的差距。遍历所有服务车辆，计算它们与通信范围内簇头之间的引力，选择引力最大的簇头加入。继续遍历剩余未加入集群的车辆，最终形成稳定的聚合组。具体 DT 增强型车辆聚类算法如算法 1 所示。

算法 1 数字孪生增强型车辆聚类算法

输入 DT 状态信息 $\{C_n, v_n, l_n, F_n\}$, $\{l_q, F_q\}$

输出 车辆集群 VD

- 1) 初始化，通过 DT 网络获取车辆状态
- 2) 重复计算步骤 3)~步骤 12)
- 3) for $TV_r = TV_1:TV_R$
- 4) 构建 TV_r 通信范围内的服务车辆列表 JU_f
- 5) for $u_j^f = u_1^f:u_f^f$
- 6) 根据式(14)计算簇头因子 C_f
- 7) end for
- 8) 按 C_f 降序排序，选择前 K 辆车作为簇头
- 9) for $SV_f = SV_1:SV_F$
- 10) 根据式(15)计算 SV_f 与范围内簇头引力值
- 11) 将 SV_f 分配给引力最大的簇头
- 12) end for
- 13) 检测剩余未加入集群的车辆，直至所有车辆加入某个簇
- 14) 直至所有车辆集群稳定
- 15) 输出最终集群集合 VD

算法 1 主要由候选服务车辆搜索、簇头因子计算、簇头筛选与成员分配等步骤构成。设任务车辆数为 Q ，服务车辆总数为 S ，每辆服务车辆通信范围内的平均可感知车辆数为 G ，目标簇头数量为 C 。首先，候选集构建阶段需遍历任务车辆与服务车辆，时间复杂度为 $\mathcal{O}(Q \cdot S)$ ；在簇头因子计算复杂度为 $\mathcal{O}(S \cdot G)$ ；簇头筛选排序的复杂度为

$\mathcal{O}(S \cdot \text{lb}S)$ ；簇成员根据引力分配给 C 个簇头，复杂度为 $\mathcal{O}(S \cdot C)$ ；因此，总体的时间复杂度为 $\mathcal{O}(Q \cdot S + S \cdot G + S \cdot \text{lb}S + S \cdot C)$ 。空间复杂度主要用于存储车辆状态、候选列表和中间变量，约为 $\mathcal{O}(Q + S + C)$ 。

2.2 马尔可夫决策过程

对于目标函数 J 的边缘计算任务卸载模型，可将其转化为一个马尔可夫决策过程 (MDP, Markov decision process)。该过程的状态空间和动作空间都具有很高的维度，用一个四元组 (S, A, P, R) 表示，其中， S 表示智能体所有状态组成的状态空间， A 表示智能体所有可能采取的动作集合， P 表示转移概率， R 表示奖励函数^[21]。

状态空间 S 表示整个系统的状态。RSU 作为智能代理通过 DT 模型实时获取表征系统的全局动态信息，该状态空间是通过 DT 辅助聚类后的环境特征，为深度强化学习的策略优化提供多维状态输入。它包括每个聚合组 VD_k 中的服务车辆 u_{kw} 的位置、任务车辆 TV_r 的实时位置、任务车辆 TV_r 产生的任务 e_r 的大小和计算资源需求、系统剩余任务总量。因此，系统状态可以简记为 $S = \{\text{loc}_{kw}, \text{loc}_r, D_r, C_r, D_{re}\}$ 。

动作空间 A 表示所执行动作的集合。RSU 每获取一个状态，便会根据贪心策略选取并执行该动作。对于该网络边缘计算卸载和资源分配问题，主要包括卸载目的地的选择和计算资源分配 2 个部分，表示为 $A = \{a_{rkw}, b_r, b_{rg}, c_{rkw}, d_{rq}\}$ ，其中， b_r 表示任务车辆 TV_r 的整体卸载策略 ($b_r = 1$ 为卸载，否则为本地处理)； b_{rg} 表示子任务 e_{rg} 的卸载目标选择 ($b_{rg} = 1$ 为车辆集群，否则为 RSU)； $a_{rkw} \in \{0, 1\}$ 表示在车辆集群 VD_k 中选择服务车辆 u_{kw} ； c_{rkw} 和 d_{rq} 分别表示服务车辆和 RSU 为子任务分配的计算资源比例，需满足资源约束 $\sum c_{rkw} \leq 1$ 和 $\sum d_{rq} \leq 1$ 。该动作空间通过分层决策机制，支持任务卸载与资源分配的联合优化，聚类引力模型提供了结构化的状态表示，而 D3QN 则在此基础上实现对高效动作的学习与选择，二者形成协同互补的决策机制。

转移概率 P 。在当前时隙下信道增益不变，在代理执行动作后，下一个时隙的 MDP 环境将会发生变化，这也意味着任务车辆与服务车辆和 RSU 之间的传输速率会改变，因此得到转移概率的定

义,且概率值和为1。

$$P(s_{t+1}|s_t, a_t) = \Pr\{s_t = s_{t+1}|s_{t-1} = s_t, a_{t-1} = a_t\}$$

$$\sum P(s_{t+1}, R_t | s_t, a_t) = 1 \quad (16)$$

其中, s_t, s_{t+1} 分别表示当前时隙状态和执行动作 a_t 后的下个时隙的状态。

奖励函数 \mathcal{R} 表示智能体与环境交互后所得到的奖励回报。通常来说,这个奖励回报能直接反映选取动作的好坏^[22]。系统效用和时延都是评估指标,需要综合考虑,因此,奖励值可以描述为完成任务的网络效用值减去所花费的时延,差值越大越好,表示为

$$R(S, A) = U_r = (1 - \alpha) \sum_{r=1}^R U'_r - \alpha t_r, \alpha \in (0, 1) \quad (17)$$

其中, α 为权重因子,用于平衡效用与时延的优化目标。该奖励函数通过最大化效用与最小化时延的权衡,驱动智能代理学习高效的卸载策略与资源分配方案。

2.3 基于D3QN的车辆集群协同卸载算法

该方案采用基于D3QN的算法,结合了双DQN(DDQN, double deep Q-network)和Dueling DQN(dueling deep Q-network)的优势。双DQN通过双网络结构减少估计偏差,Dueling网络将 Q 值分解为状态值和动作优势,提升状态理解和动作评估能力。D3QN在复杂环境中表现出色,能够高效推导出接近最优的动作策略。

为了解决DQN的 Q 值的过度估计,Double DQN不再从 Q 目标网络中寻找每个动作的最大 Q 值,而是先在当前 Q 网络中找到最大 Q 值对应的动作 A_{\max} ^[23],然后将其放入 Q -目标网络中计算目标 Q 值,即

$$Q^{\text{target}} = R_{t+1} + \gamma Q(S_{t+1}, \arg \max_A Q(S_{t+1}, A, \theta_t), \theta_t^-) \quad (18)$$

其中, θ_t 表示 Q 网络在时间步长 t 处的参数。该方法在任务卸载和计算资源分配的问题中仅依赖 Q 值函数学习,仍然存在局限,尤其在复杂决策场景中做了决策但任务仍然无法完成时,导致学习效率低下甚至误导性更新。为此,Dueling DQN引入状态值函数和动作优势函数,将 Q 值分解为2个部分,使智能体能够更精准地评估状态价值和动作优势,从而提升学习效果。在状态 S 中选择动作 A 的 Q 值输出为

$$Q(S, A, \theta, \omega, \eta) = V(S, \theta, \omega) + A(S, A, \theta, \eta) \quad (19)$$

直接使用式(19)会导致 $A(s, a)$ 的平均值偏离0,从而影响模型的稳定性^[24]。因此,Dueling DQN采用了一种去均值的 Q 值计算方式,以保证在不同动作间仍然能保持相对优势,即

$$\tilde{A}(S, A, \theta, \eta) = A(S, A, \theta, \eta) - \frac{1}{|A|} \sum_{a \in A} A(S, a, \theta, \eta) \quad (20)$$

因此,在状态 S 中选择动作 A 输出的 Q 值可以表示为

$$Q(S, A, \theta, \omega, \eta) = V(S, \theta, \omega) + [A(S, A, \theta, \eta) - \frac{1}{|A|} \sum_{a \in A} A(S, a, \theta, \eta)] \quad (21)$$

D3QN算法的 Q 值表达式与Dueling DQN算法基本相同,其状态值和动作优势值由Double DQN的双深度 Q 网络计算^[25],D3QN算法的损失函数可以表示为

$$L(\theta) = \mathbb{E}[(y_t - Q(s_t, a_t; \theta))^2] \quad (22)$$

以上损失函数作为当前网络参数与样本数据拟合程度的度量, $\mathbb{E}(\cdot)$ 表示期望运算。通过最小化损失函数,可以优化网络的参数 θ ,从而提升 Q 值估计的准确性,指导智能体选择最优行为策略。算法采用了基于 ε -greedy策略的动作选择机制。基于D3QN的车辆集群协同卸载算法如算法2所示。

算法2 基于D3QN的车辆集群协同卸载算法

输入 D3QN算法参数、DT网络状态数据

输出 卸载和资源分配决策

- 1) 初始化 当前 Q 网络参数 θ 、目标 Q 网络参数 θ' 、经验回放单元
- 2) for $i=0:T-1$
- 3) 重置环境,获取初始状态 S_0
- 4) for $t = 0:T$
- 5) 对状态 S_t 利用DT辅助聚类后得到 \hat{S}_t
- 6) 输入 \hat{S}_t 进入当前 Q 网络,计算每个动作 A_t 的 Q 值,使用 ε -greedy选择动作 A_t
- 7) 获得奖励、下一状态 S_{t+1} 和is_terminal
- 8) 将 $(\hat{S}_t, A_t, R_t, \hat{S}_{t+1}, \text{is_terminal})$ 存储到 D
- 9) 从经验回放单元随机采样 m 个样本,根据式(21)计算目标 Q 值
- 10) 根据式(22)计算误差反向更新参数 θ
- 11) 更新状态 S_{t+1} 和Reward

- 12) 判断 $t = T - 1$ 或是否完成
 13) end for
 14)end for

该算法的时间复杂度主要取决于训练轮数 E 和每轮中的时间步数 T 。同时算法复杂度也受神经网络规模的影响。设网络共有 L^{fw} 层前向传播层和 L^{bw} 层反向传播层，每层的输入输出神经元数量分别为 n_i^l 、 n_o^l ，则算法 2 网络本身的计算复杂度为 $\mathcal{O}(N) = 2 \cdot \mathcal{O}(\sum_{l=0}^{L^{\text{fw}}} n_i^l \cdot n_o^l + \sum_{l=0}^{L^{\text{bw}}} n_i^l \cdot n_o^l)$ 。在训练过程中，从经验回放池中采样的时间复杂度为 $\mathcal{O}(B)$ ，其中 B 为每次训练所采样的 batch 大小。因此，整个算法 2 的总时间复杂度可以近似表示为 $\mathcal{O}(E \cdot T \cdot (2 \sum_{l=0}^{L^{\text{fw}}} n_i^l \cdot n_o^l + \sum_{l=0}^{L^{\text{bw}}} n_i^l \cdot n_o^l + B))$ 。空间复杂度由网络参数和经验回放池所占内存共同决定。由于 D3QN 包含 2 个网络，其参数存储复杂度为 $\mathcal{O}(2 \cdot W)$ ，其中 W 为单个网络的权重数量。设 R 和 C 分别表示经验池与中间缓存的空间占用，则总空间复杂度为 $\mathcal{O}(2 \cdot W + R + C)$ 。

3 仿真实验

3.1 仿真场景与仿真参数设置

本文考虑一个长度为 2 000 m 的双向高速公路场景，它包含 4 个车道。在满足基本任务卸载需求的同时尽可能贴近真实场景，场景包括了 2 个与云服务器有线相连的 RSU，任务车辆数量设置为 3 辆。

同时系统生成的总任务量不变，每个时隙 3 个不同位置的任务车辆生成 2 个不同类型的子任务。仿真实验模块主要包括系统环境边缘车载自组网的配置和 RSU 中的智能代理 (Agent) 两部分。考虑到现实城市道路环境，车辆以恒定速度行驶。区域内设置 50 辆服务车辆，聚类后每个聚合组至少包含 5 辆车，速度大小设置为 18~22 m/s 以恒定速度行驶。关于信道模型，参考了文献[26]，采用地面总的频谱带宽为 $W=5$ MHz 来传输任务。此外，参考了文献[27]，将车辆上传数据的功率设置为 $P=0.1$ W。其他的仿真参数设置如表 2 所示。关于收费单价的设置基于系统模型，参考文献[19]、文献[24]和文献[28]。

本文基于 D3QN 的车辆集群协同卸载算法，需要连续的动作探索，因此使用 ϵ -greedy 进行动作选择，设置贪婪指数为 0.9，逐渐衰减到 0.05，并将折扣衰减因子 r 大小定义为 0.99，以对预期奖励进行折扣。在神经网络中建立了 3 个隐藏层，神经元节点数分别为 256、128 和 20。在训练过程中，最小批和回放缓冲器大小分别是 256 和 4 000。

3.2 仿真结果分析

3.2.1 参数分析

为了确定超参数的最佳值，比较了超参数对网络奖励的影响，结果如图 3 所示，超参数选择对神经网络的训练效果有显著影响。学习率控制着模型每轮参数更新的幅度，当学习率较大时，虽然初期训练速度较快，但容易造成模型震荡，导致收敛不

表 2 仿真参数设置

参数	预设值	参数	预设值
服务车辆数量/辆	[30,50]	车辆计算资源/GHz	0.1
任务车辆数量/辆	3	RSU 计算资源/GHz	0.2
任务车辆速度/(m·s ⁻¹)	20	完成 e_{rg} CPU 周期	$0.2 \times 10^9 \sim 0.4 \times 10^9$
服务车辆速度/(m·s ⁻¹)	18~22	每 Hz 用户传输单价	8×10^{-5}
e_{rg} 数据大小/Mbit	0.4~0.5	每 Hz 租车服务频谱资源单价	0.4×10^{-5}
总的任务大小/Mbit	[40,70]	每 bit 任务车辆计算单价	2×10^{-8}
地面总频谱带宽/MHz	5	每 Hz 租车服务计算资源单价	0.4×10^{-5}
车辆发射功率/W	0.1	每 Hz 车辆服务奖励返回单价	0.8×10^{-5}
e_{ri} 数量	[0,2]	e_{ri} 数量	[0,2]
每 Hz 服务车辆奖励返回单价	1.2×10^{-5}	每 Hz 租赁 RSU 计算资源	1.2×10^{-5}
每 Hz 租赁 RSU 频谱资源单价	0.2×10^{-5}	每 Hz RSU 奖励返回单价	0.6×10^{-5}

稳定;当学习率较小时,模型虽然收敛更平稳,但整体训练时间显著增加,影响效率。由于算法采用了贪婪策略,智能体在探索环境时优化决策,因此贪婪值会随着训练过程逐步衰减。从图 3 可以看出,衰减因子越小,收敛速度越慢。经对比后选择学习率为 0.003,衰减因子为 5×10^{-4} ,以实现快速收敛和稳定性。

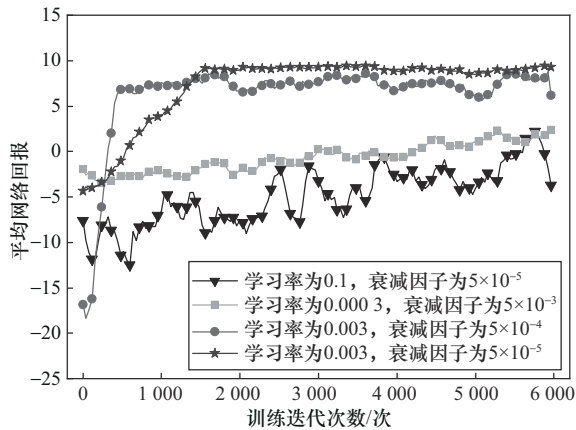


图 3 超参数对网络奖励的影响

不同权重对系统性能的影响如图 4 所示,奖励函数中权重系数 α 对系统性能表现具有显著影响。随着 α 的增加,即效用部分的权重降低、时延惩罚的权重增加,平均网络效用呈下降趋势,而平均时延整体逐步减小。当 $\alpha = 0.87 \sim 0.89$ 时,平均网络效用都处于较高水平,但平均时延也相对较高;当 $\alpha = 0.89 \sim 0.91$ 时,平均网络效用下降至 105.52,平均时延处于较低水平。在综合考虑系统性能前提下, $\alpha = 0.89$ 附近表现出较好的平衡性,因此在后续实验中选择该值作为默认设置。

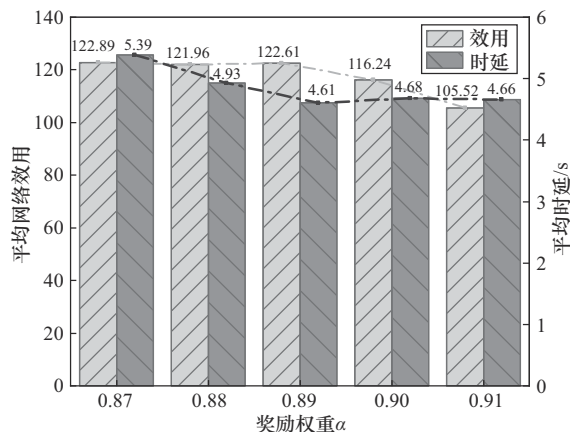


图 4 不同权重对系统性能的影响

3.2.2 卸载方案的性能分析

为了更好地评估本文算法 DV-D3QN 在优化平均系统效用和时延的性能,将其与 6 种基线算法进行比较。其中, DV-D3QN、DV-Dueling DQN 和 DV-DDQN 这 3 种算法均采用了本文构建的 DT 辅助车辆聚类机制,同时选取文献[12]的 DT-A3C 算法和文献[20]的 GK-DDPG 算法作为对照。文献[12]虽未在聚类过程中引入 DT 技术,但利用 DT 优化车辆任务卸载与资源分配;文献[20]则采用 GK-means 算法进行辅助聚类。作为进一步对照, NC-D3QN 未使用任何聚类手段,也未引入 DT,仅基于个体车辆的局部状态进行卸载决策; K-D3QN 未引入 DT,仅采用传统 K-means 算法按车辆的空间位置聚类,在聚类过程不考虑车辆的速度、资源负载等动态信息。此外,本文关注卸载任务在策略执行阶段的响应时延,平均时延统计不包含 DT 模块在云端聚类与决策过程中的通信与计算时延。

1) 平均网络奖励收敛性比较

不同算法的平均网络奖励如图 5 所示。NC-D3QN 和 K-D3QN 因未能引入 DT 技术进行动态环境建模和实时聚类优化,导致收敛较慢且奖励波动较大。DT-A3C 虽利用 DT 优化任务卸载与资源分配,但未在聚类过程中引入 DT 辅助聚类,因此在收敛速度与稳定性方面均逊于 DV 系列算法。GK-DDPG 虽采用基于 DT 的 GK-means 算法进行车辆聚类,但其基于连续动作空间的训练过程收敛速度偏慢。

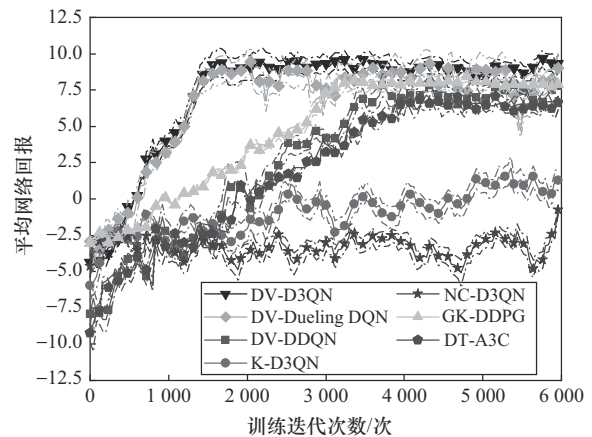


图 5 不同算法的平均网络奖励

DV-Dueling DQN 和 DV-DDQN 由于引入了 DT 技术进行动态聚类,系统能够实时模拟网络状

态, 进而优化车辆行为的决策过程和系统的动态适用性, 因此收敛速度和稳定性有所提升。但 DV-DDQN 收敛速度仍慢于 DV-D3QN 约 45%, DV-Dueling DQN 收敛后稳定性较差, 本文 DV-D3QN 算法相比这 2 种算法优化了目标函数和最优动作计算, 使智能体在约 1 800 次训练找到最优策略, 相较于 Dueling DQN, 收敛速度提高了近 3 倍。本文算法在全局范围的平均网络奖励相比其他基线算法, 其收敛速度更快, 最终奖励值更高, 进一步验证了该算法的优越性。

2) 平均网络效用对比

当总任务大小为 50 Mbit 时, 不同算法的平均网络效用如图 6 所示。

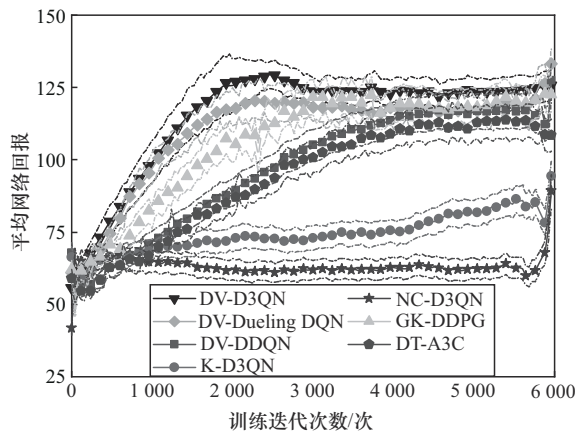


图6 不同算法的平均网络效用

由于智能体在初期处于探索阶段, 因此网络效用值较低。但随着训练的持续进行, 本文 DV-D3QN 算法与 DV-Dueling DQN、DV-DDQN 相比, 在训练后期表现出更小的波动性, 保证了更稳定的效用收敛值。与之相比, DT-A3C 虽然利用 DT 构建模型, 逐步改善网络效用, 但由于未使用 DT 辅助聚类, 其优化效率仍然不高。GK-DDPG 采用 GK-means 聚类优化任务卸载, 但未考虑车辆动态信息, 具有无法适应复杂网络环境的局限性。K-D3QN 和 NC-D3QN 在决策过程中无法实时感知网络状态变化, 导致决策空间过大、优化效率低下。相比之下, 本文算法的平均网络效用分别提高了 5.32%、8.54%、1.47%、11.2%、68.51% 和 103.15%。因此, 在本文研究的环境下, 本文算法相较于其他基线方法, 能够更快地优化网络效用, 且收敛后有更稳定的效用波动, 验证了该算法的优势和适用性。

3) 平均网络时延对比

图 7 展示了不同算法在训练过程中平均时延的变化情况。训练初期, 由于智能体的探索行为, 所有算法的时延都较高且波动较大。随着训练的深入, 智能体逐渐学习到更优的任务卸载和资源分配策略, 大部分算法的平均时延开始下降并趋于稳定。

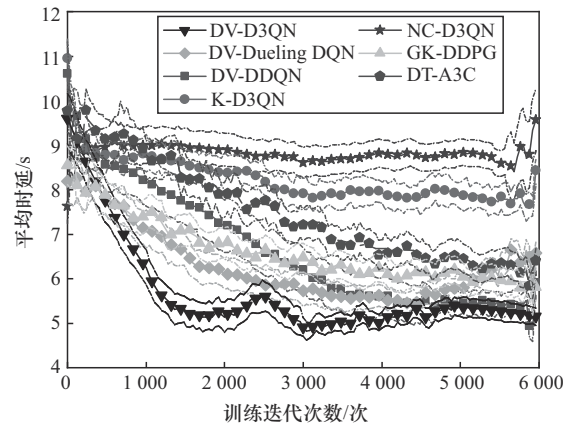


图7 不同算法的平均时延

与其他基线方法相比, 本文 DV-D3QN 算法在收敛后表现出最低的平均时延和较小的波动幅度。这得益于 DV-D3QN 结合了 Dueling DQN 和 DDQN 的优势, 通过更精确的 Q 值更新和更稳定的决策, 减少了任务卸载的不确定性, 合理地利用计算资源。相比于其他基线方案, 本文 DV-D3QN 算法的处理时延分别减少了 14.94%、13.51%、19.3%、25.4%、38.53%、46.52%。同时, DT 辅助聚类的引入缩小了卸载空间, 提升了车辆间的任务协作效率, 进一步降低了计算和通信时延; 相比 NC-D3QN 和 K-D3QN, DV-D3QN、DT-A3C 能够更快找到最优策略, 减少决策失误, 显著降低平均时延。虽然 GK-DDPG 通过 GK-means 聚类优化任务卸载, 但其聚类过程中未考虑车辆的动态信息, 导致在高负载和高动态环境下, 时延表现不如本文算法。综上所述, DV-D3QN 算法在任务卸载决策和资源分配方面表现优异, 能够更快学习最优策略, 在收敛速度、稳定性和性能上均优于其他算法, 验证了其有效性和优越性。

3.2.3 不同参数的适用性分析

1) 不同任务类型对时延的影响

在不同车辆数量 (从 40~70 辆) 下, 混合任

务、可分割任务和非线性任务的平均时延变化情况如图 8 所示。仿真结果表明,本文 DV-D3QN 算法对不同任务类型的处理时延均表现出显著优势。在可分割任务场景下,DT 技术通过实时采集车辆位置、速度和负载等多维数据,使聚类算法能够动态感知网络状态变化,从而智能调整任务卸载策略,使时延稳定维持在 4.0 s 左右。这种稳定性验证了聚类算法在资源调度方面的有效性,即使车辆规模从 40 辆增至 70 辆,仍能保持高效的并行处理能力。

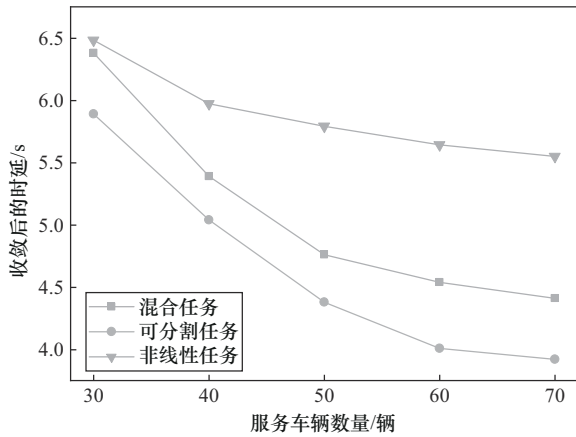


图 8 不同种类任务在服务车辆变化下的时延比较

对于非线性任务,本文 DV-D3QN 算法借助 DT 技术实现的动态拓扑感知能力,时延从 5.0 s 优化至 4.0 s。这一改进主要源于聚类算法对任务依赖关系的智能识别,通过建立最优的卸载序列,减少了任务间的等待时间。混合任务场景下 4.5 s 的均衡表现,则体现了算法在复杂任务组合中的鲁棒性,说明本文算法能根据任务特性动态调整资源分配策略,实现整体性能优化。

本文的仿真任务设置考虑到实际需求的多样性,都采用混合任务的设置。

2) 车辆 CPU 计算能力对车辆效用和时延的影响

图 9 和图 10 展示了不同算法在不同车载计算能力 (0.09~0.12 GHz) 下的平均效用和时延的表现。随着计算能力提升,两者均呈现正向影响。

在效用方面,本文 DV-D3QN 算法在所有情况下表现最佳,表明本文算法通过更高效的资源调度策略,充分利用计算资源,实现了更优的任务卸载和计算分配。随着计算能力增强 (由 0.09 GHz 增

加到 0.12 GHz),各算法效用差距逐渐缩小,表明硬件性能的提升部分弥补了算法效率的不足,但本文算法仍保持最优,凸显其策略的鲁棒性。

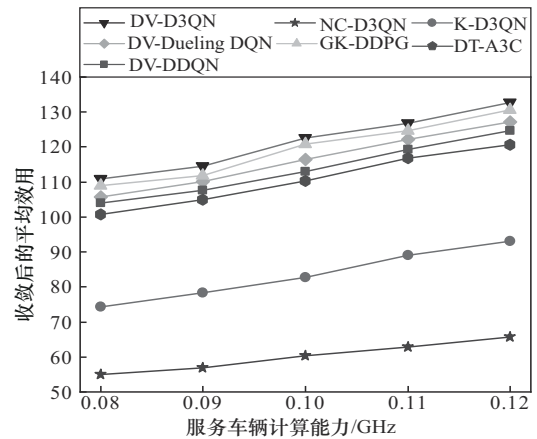


图 9 不同算法在服务车辆计算能力上的平均效用对比

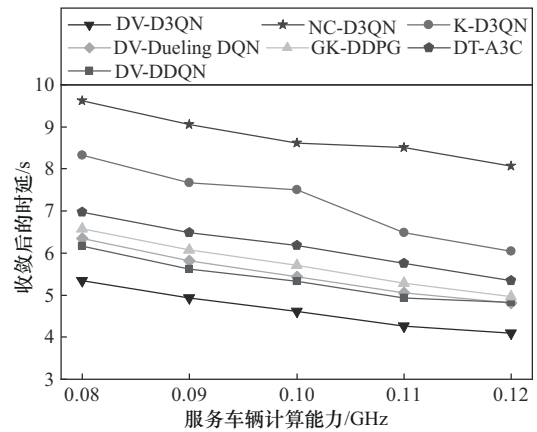


图 10 不同算法在服务车辆计算能力上的时延对比

在时延方面,计算能力的提升直接降低了任务处理时间,本文 DV-D3QN 算法通过智能调度进一步减少了资源竞争带来的等待时延,尤其在低算力场景下其调度优势更为显著,当计算能力达到较高水平时,硬件计算能力成为主导因素,算法差异对时延的影响趋于弱化。这表明本文算法的核心价值在于计算资源受限场景下的精细化调度能力,能为边缘计算中动态异构环境提供有效的优化路径。

3) 不同总任务大小的平均效用

不同算法在总任务大小上的平均效用对比如图 11 所示。本文 DV-D3QN 算法在所有任务规模下均表现最佳,比 DV-Dueling DQN、NC-D3QN 分别高 13.6%、91.4%。

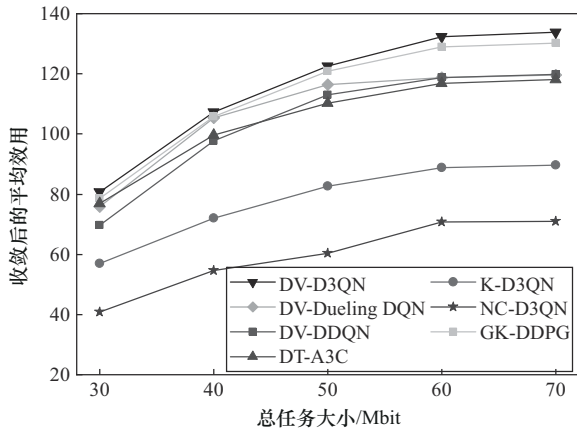


图 11 不同算法在总任务大小上的平均效用对比

随着任务规模增加，各算法的效用整体提升，但 NC-D3QN 和 K-D3QN 受限于未充分优化卸载策略，增益较小。本文 DV-D3QN 算法通过 DT 增强型聚类优化了任务卸载空间，减少了冗余决策的计算开销，同时结合 D3QN 的双目标网络和状态值分解机制，提升了策略稳定性和 Q 值估计精度。随着任务规模的增大，NC-D3QN 因未充分优化聚类导致决策波动加剧，本文算法通过动态调整车辆集群的计算资源分配和卸载决策，有效降低了任务处理时延并提高了资源利用率，从而在各类任务规模下均表现出更优的效用收敛性和稳定性。GK-DDPG 虽然通过 GK-means 聚类优化任务卸载，但未考虑车辆的动态信息，导致在大任务规模下的效用提升较小。DT-A3C 虽利用 DT 优化任务卸载与资源分配，但由于未结合 DT 辅助聚类，其任务协作效率不如本文算法。

4) 不同服务车辆数量的平均效用和时延

图 12 和图 13 从车辆数量对平均效用和时延的影响进行对比分析，不同算法的性能差异随车辆规模扩大呈现显著分化。

在效用方面，当车辆数量从 40 辆增至 70 辆时，使用 DT-VC 聚类算法的 3 种算法展现出较强的扩展性，聚类算法能够减少不必要的计算资源竞争，使任务分配更加均衡，从而提高效用值。相比其他算法，本文 DV-D3QN 算法通过 DT 技术实现动态聚类优化，实时融合多维车辆数据提升聚类精度，结合 D3QN 的双目标网络抑制 Q 值过估计，最终在复杂场景中达到最高效用。GK-DDPG 虽然采用 GK-means 聚类优化任务卸载，但未考虑车辆动态状态，效用提升有限。DT-A3C 虽然利用 DT 技术优

化任务卸载，但未结合 DT 辅助聚类，相较于本文算法其任务协作能力有所不足，无法达到同样的协同效益。

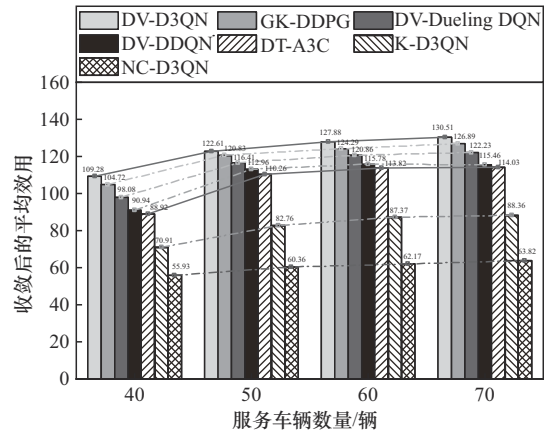


图 12 不同算法在服务车辆数量上的平均效用对比

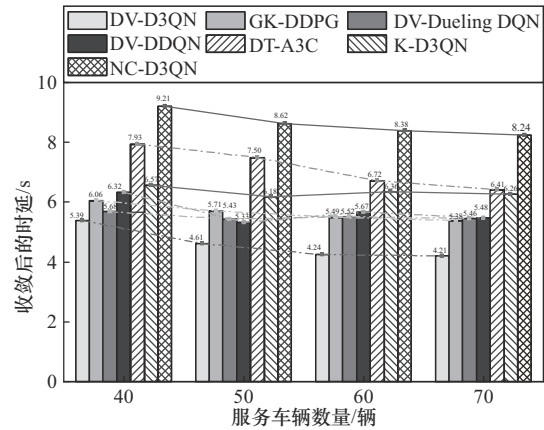


图 13 不同算法在服务车辆数量上的时延对比

在时延方面，本文 DV-D3QN 算法在所有情况下均表现出最低的时延，DV-Dueling DQN 和 DV-DDQN 因未完全解决 Q 值偏差问题，时延分别增加了 10.8% 和 5.9%。GK-DDPG 通过 GK-means 聚类优化卸载任务，但未能有效控制动态环境下的时延。DT-A3C 虽利用 DT 优化资源分配，但缺乏辅助聚类，导致高负载时延较高。而 NC-D3QN 和 K-D3QN 因聚类效率低下和决策冲突，导致部分任务在高负载情况下仍被错误调度，增加了任务队列时延，时延分别升高了 31.9% 和 40.7%。本文算法通过动态调整资源分配优化和精准卸载决策机制，在车辆规模扩大时仍保持最低时延性能。实验表明，本文算法在动态车辆网络中具备强鲁棒性，为高密度场景提供了可靠解决方案。

5) 任务车辆速度对车辆效用的影响

为了验证 DT 增强型车辆聚类算法在高动态环境下的适应性, 将本文 DV-D3QN 算法与只使用传统 k-means 算法的 K-D3QN 和没有车辆聚类的 NC-D3QN 算法进行比较, 对比结果如图 14 所示。在其他条件不变的情况下, 随着任务车辆速度的增加, 所有算法的效用均下降, 但本文算法下降最缓慢且始终保持最高效用。由于 DT-VC 聚类通过借助 DT 实时感知车辆资源状态 F_n 、相对距离 RD_{ij} 和行驶方向, 从而构建稳定的协作集群, 有效减缓高速移动引发的拓扑频繁变化对卸载策略的干扰。相比之下, K-D3QN 采用传统 k-means 聚类, 仅基于静态位置划分车辆, 无法适应高速场景下的资源动态分布, 导致集群内任务协作效率下降; 而 NC-D3QN 因未采用聚类机制, 在高速环境下决策空间爆炸式增长, 策略波动剧烈。

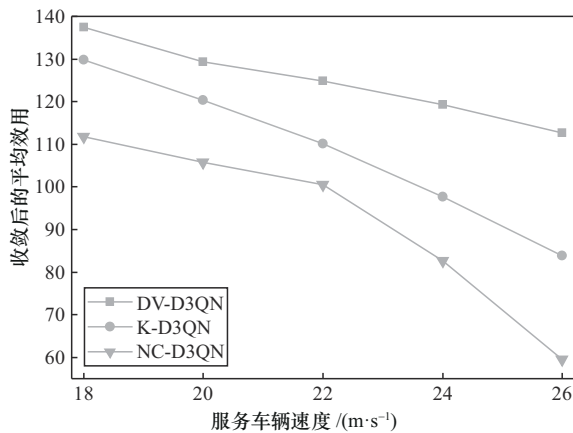


图 14 不同算法在任务车辆速度上的平均效用对比

4 结束语

本文针对车联网高动态环境下多类型任务的边缘计算卸载问题, 提出了一种 DT 辅助下基于 D3QN 的车载网络协同卸载算法。该算法采用双阶段优化框架, 首先设计 DT 增强型车辆聚类算法对车辆辅助聚类, 显著增强卸载决策可靠性。随后设计了基于 D3QN 的车辆集群协同卸载算法, 实现时延和系统效用的多目标平衡。将本文 DV-D3QN 算法与 DV-DDQN、DV-Dueling DQN、GK-DDPG、DT-A3C、K-D3QN 和 NC-D3QN 这 6 种基线算法进行对比, 仿真结果表明, 本文 DV-D3QN 算法在平均系统效用、时延等关键指标上均展现出显著优势, 该方案能够有效适应高动态的网络拓扑变化,

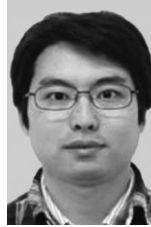
在高负载场景下保持卓越的稳定性, 并支持多类型任务的差异性优化, 为车联网边缘计算场景的任务卸载提供了一种高效可靠的解决方案。

参考文献:

- [1] ZHANG J X, ZHAO L, YU K P, et al. A novel federated learning scheme for generative adversarial networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(5): 3633-3649.
- [2] ZHAO L, YANG K Q, TAN Z Y, et al. A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(6): 3664-3674.
- [3] 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.
- [4] DU J, JIANG C X, ZHANG H J, et al. Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(10): 2202-2217.
- [5] SHI J M, DU J, WANG J, et al. Deep reinforcement learning-based V2V partial computation offloading in vehicular fog computing[C]//Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2021: 1-6.
- [6] WANG X J, NING Z L, GUO S, et al. Imitation learning enabled task scheduling for online vehicular edge computing[J]. IEEE Transactions on Mobile Computing, 2022, 21(2): 598-611.
- [7] ZHAO L, ZHANG E C, WAN S H, et al. MESON: a mobility-aware dependent task offloading scheme for urban vehicular edge computing[J]. IEEE Transactions on Mobile Computing, 2024, 23(5): 4259-4272.
- [8] LU Y L, MAHARJAN S, ZHANG Y. Adaptive edge association for wireless digital twin networks in 6G[J]. IEEE Internet of Things Journal, 2021, 8(22): 16219-16230.
- [9] PENG X T, OTA K, DONG M X. Multiattribute-based double auction toward resource allocation in vehicular fog computing[J]. IEEE Internet of Things Journal, 2020, 7(4): 3094-3103.
- [10] DAI Y Y, ZHAO J T, ZHANG J, et al. Federated deep reinforcement learning for task offloading in digital twin edge networks[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(3): 2849-2863.
- [11] KHAN L U, SAAD W, NIYATO D, et al. Digital-twin-enabled 6G: vision, architectural trends, and future directions[J]. IEEE Communications Magazine, 2022, 60(1): 74-80.
- [12] DAI Y Y, ZHANG Y. Adaptive digital twin for vehicular edge computing and networks[J]. Journal of Communications and Information Networks, 2022, 7(1): 48-59.
- [13] DO-DUY T, HUYNH D V, DOBRE O A, et al. Digital twin-aided intelligent offloading with edge selection in mobile edge computing[J]. IEEE Wireless Communications Letters, 2022, 11(4): 806-810.
- [14] SUN W, LEI S Y, WANG L, et al. Adaptive federated learning and digital twin for industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2021, 17(8): 5605-5614.
- [15] MA X D, SU Z, XU Q C, et al. Edge computing and UAV swarm coop-

- erative task offloading in vehicular networks[C]//Proceedings of the 2022 International Wireless Communications and Mobile Computing (IWCMC). Piscataway: IEEE Press, 2022: 955-960.
- [16] 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.
- [17] GUO H Z, ZHOU X Y, WANG J D, et al. Intelligent task offloading and resource allocation in digital twin based aerial computing networks[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(10): 3095-3110.
- [18] PENG Y J, SONG X Q, LIU F, et al. Joint task partition and computation offloading for latency-sensitive services in mobile edge networks[C]//Proceedings of the 2022 18th International Conference on Mobility, Sensing and Networking (MSN). Piscataway: IEEE Press, 2022: 191-196.
- [19] 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.
- [20] ZHAO L, ZHAO Z J, ZHANG E C, et al. A digital twin-assisted intelligent partial offloading approach for vehicular edge computing[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(11): 3386-3400.
- [21] ZHAO N, LIANG Y C, NIYATO D, et al. Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks[J]. IEEE Transactions on Wireless Communications, 2019,
- [22] ZHANG L, ZHANG Z Y, MIN L, et al. Task offloading and trajectory control for UAV-assisted mobile edge computing using deep reinforcement learning[J]. IEEE Access, 2021, 9: 53708-53719.
- [23] TANG H J, WU H M, QU G J, et al. Double deep Q-network based dynamic framing offloading in vehicular edge computing[J]. IEEE Transactions on Network Science and Engineering, 2023, 10(3): 1297-1310.
- [24] YANG C, LIU B C, LI H Y, 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.
- [25] WANG Y H, CHEN L, ZHOU Y F, et al. Resource allocation and trajectory design in UAV-assisted jamming wideband cognitive radio networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(2): 635-647.
- [26] YU Z, GONG Y M, GONG S M, 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.
- [27] TIAN S J, DENG X H, CHEN P P, et al. A dynamic task offloading algorithm based on greedy matching in vehicle network[J]. Ad Hoc Networks, 2021, 123: 102639.
- [28] XUE J B, WU Q Q, ZHANG H J. Cost optimization of UAV-MEC network calculation offloading: a multi-agent reinforcement learning method[J]. Ad Hoc Networks, 2022, 136: 102981.

[作者简介]



陈康 (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方面无线资源管理和优化算法。



宋政翰 (2000-), 男, 山东济宁人, 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。



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。



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。