移动车载边缘网络中基于递归深度强化学习的协作缓存接力算法

吴红海，王白冰，马华红，邢玲
（河南科技大学信息工程学院，河南 洛阳 471023）

摘　要：考虑无路侧单元覆盖的场景，充分利用车辆之间的协作来构建缓存系统，提出一种基于递归深度强化学习的协作缓存接力算法。考虑缓存决策的动态特性，将问题建模为部分可观察的马尔可夫决策过程，利用图神经网络预测车辆轨迹，并通过计算车辆间的连接稳定性度量，选择可作为缓存节点的车辆。此外，将长短期记忆网络嵌入深度确定性策略梯度算法中，以实现最终的缓存决策。仿真结果表明，所提算法在缓存命中率和时延方面优于传统缓存算法。

关键词：车载边缘网络；协作缓存接力；递归深度强化学习；马尔可夫决策
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024195

Recursive deep reinforcement learning-based collaborative caching relay algorithm in mobile vehicular edge network

WU Honghai, WANG Baibing, MA Huahong, XING Ling
College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Abstract: Considering scenarios without road side unit coverage, a recursive deep reinforcement learning-based collaborative caching relay algorithm was proposed to construct a caching system by leveraging the cooperation among vehicles. Recognizing the dynamic nature of caching decisions, the problem was modeled as a partially observable Markov decision process. Vehicle trajectories were predicted using graph neural network, and the connectivity stability between vehicles was measured to select those that could serve as caching nodes. In addition, long short-term memory network was integrated into the deep deterministic policy gradient algorithm to achieve the final caching decision. Simulation results demonstrate that the proposed algorithm outperforms traditional caching algorithms in terms of cache hit ratio and latency.

Keywords: vehicular edge network, collaborative caching relay, recursive deep reinforcement learning, Markov decision

0　引言

近年来，汽车行业发展迅猛，智能化和网联化已成为未来发展的重要趋势。随着车联网技术的普及和发展，车载边缘网络[1]作为一种新型通信架构得到广泛关注。车载边缘网络是由车辆与边缘节点构成的分布式网络，如图 1 所示。通过车辆之间的通信和边缘计算资源，车载边缘网络提供低时延、高带宽和实时性强的车联网服务。随着计算机技术和通信技术的快速发展，车辆间传输的数据量急剧

收稿日期：2024–07–05；修回日期：2024–10–22
通信作者：吴红海，honghai2018@haust.edu.cn
基金项目：国家自然科学基金资助项目（No.62272146, No.62071170, No.62171180, No.62072158, No.U23A20272, No.U22A2069）
Foundation Items：The National Natural Science Foundation of China (No.62272146, No.62071170, No.62171180, No.62072158, No.U23A20272, No.U22A2069)

增加，给车载网络带来了巨大压力[2]。为应对这一挑战，引入边缘缓存技术成为有效解决方案之一。边缘缓存是一种在系统边缘部署的缓存技术，在车载网络中引入边缘缓存技术，构建以路侧单元（RSU, road side unit）和智能汽车为缓存节点的协同缓存系统。当用户请求内容时，这些边缘设备能够直接通过无线通信为用户提供传输服务，从而减少从远程内容服务商获取内容的频率，大幅提高资源利用率[3]，有效降低内容获取时延，提高用户体验质量[4]。

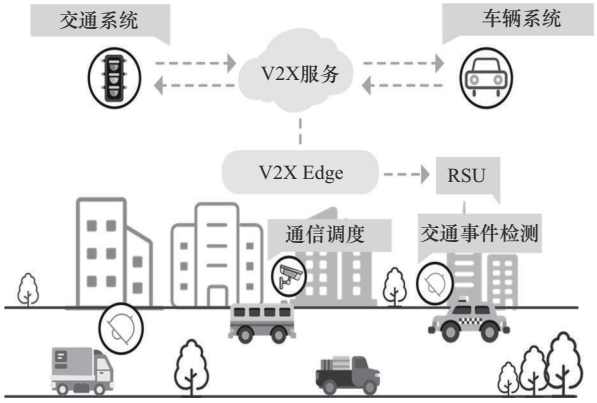

图1　车载边缘网络体系结构

现有的车载边缘缓存策略主要集中在车对基础设施（V2I, vehicle-to-infrastructure）通信，为了满足经过不同边缘节点的车辆对服务时延的要求，许多研究引入边缘协作缓存，并提出协作内容缓存方案[5]，这需要借助RSU之间的协作来完成数据缓存。通过这种方式，边缘服务器可以将高频访问的数据存储在RSU中，使车辆能够直接从RSU获取所需内容。然而，在车载边缘网络环境中，受限于部署成本[6-7]因素，无论是在城市还是乡村，RSU的覆盖范围往往有限。因此，RSU仅能提供断断续续的服务，这限制了通过V2I单跳传输实现的端到端服务，可能导致更高的通信时延。

随着智能汽车的发展，车辆开始具有一定的存储能力。一些研究者利用车与车（V2V, vehicle to vehicle）通信[8-10]进行车辆之间的缓存，无需借助RSU即可在车与车之间实现内容共享。然而，现有研究虽然利用V2V通信解决了RSU间歇性连接的问题，但由于车辆高速移动，车辆之间的连接时间非常短暂，这可能导致数据传输失败。

针对上述问题，考虑各区域RSU的有限性、车辆的高速移动性以及缓存内容的放置问题，本文设计了一种适用于移动车载边缘网络的车辆协作缓存系统，旨在从车辆轨迹预测、缓存节点选择和缓存内容更新3个方面解决车载应用和服务带来的相关问题。为应对动态交通环境，本文提出一种基于递归深度强化学习的协作缓存接力算法RDRL-CR。RDRL-CR使用图神经网络（GNN, graph neural network）预测车辆轨迹，并定义车辆之间的连接稳定性度量。根据预测的轨迹信息，构建预测权重邻接矩阵，以选择最优的车辆节点作为缓存车辆节点。最后，为了管理节点并协调缓存决策，本文提出一种基于Actor-Critic框架的缓存机制，通过递归深度强化学习算法求解最小化内容传递时延的目标函数，从而获得最优的缓存决策。

1　相关工作

基于固定基础设施的边缘缓存是指为边缘基础设施（如基站、RSU等）提供缓存资源，在基础设施上缓存内容，车辆通过V2I通信获取这些内容。Khanal等[11]研究了如何在RSU中优化热门文件的存储，以减少内容下载时延。为了缓解多个内容提供者对有限RSU缓存的竞争，Hu等[12]提出一种基于多目标拍卖的方法，并结合RSU重叠区域的特性制定一种基于缓存的切换决策机制，旨在使缓存的总体效益最大化。用户请求的不确定性是缓存问题的关键，对缓存命中率有显著影响。Bitaghsir等[13]引入基于多臂老虎机算法的内容缓存和分享策略，以评估未知内容的受欢迎程度，从而提升缓存效率。考虑车辆环境，Zhao等[14]设计了一种结合内容流行度预测的多级缓存机制，并运用动态马尔可夫模型预测车辆与RSU的连接概率，以便在RSU中预先缓存数据，增加流量卸载，并改进内容请求的响应时间。然而，这些边缘缓存策略的研究通常依赖于固定基础设施，并未充分考虑设施服务范围和缓存效率的限制。

实际上，车辆可以作为缓存节点，以扩展RSU有限的覆盖范围。为了提升车辆内容中心网络的效率，Qiao等[15]提出一种基于移动性预测的协同缓存方案，该方案利用基于部分匹配的预测方法，预测移动节点根据其过去轨迹到达不同热点区域的概率。Zhang等[16]设计了一种车辆内缓存的IV-Cache框架，考虑车辆的移动性，提出基于车辆间动态中

继的存储方案，将内容保存在指定的感兴趣区域，并通过擦除码引入数据冗余，以对抗V2V连接的脆弱性。Hu等[17]将高速缓存车辆与移动用户的交互建模为一个二维马尔可夫过程，以描述移动用户的网络可用性，并在此基础上提出一种在线车辆缓存算法，以最小化网络能耗。尽管上述研究考虑了车辆的移动性，但是忽略了车辆在复杂道路环境中的行驶情况。为了解决这一问题，现有工作通过深度强化学习算法识别复杂环境并提出有效的缓存算法。Wang等[18]提出一种具有请求预测的协作缓存策略，通过Q学习算法将车辆请求的内容预缓存到其他车辆，以减少内容获取时延。该算法通过使用*K*-means方法将附近车辆聚集在一起，仅在集群内提供内容，从而减少干扰。此外，还使用长短期记忆（LSTM, long short term memory）网络预测来自车辆的请求。为了更好地提高资源利用率，Xiong等[19]开发了一种基于深度强化学习（DRL, deep reinforcement learning）技术的增强算法，旨在最小化时延并增加请求资源量。然而，上述研究考虑用于缓存内容的DRL机制，该机制在多辆车参与决策的分布式环境中并不适用。Song等[20]通过考虑多智能体深度强化学习机制，研究合作场景中的联合内容缓存和内容共享，并通过设计ADMM解决多臂老虎机学习的问题。然而，这些方法在缓存节点之间大量交换信息，随着内容和缓存节点的增加，可能导致巨大的交换开销。

2　系统模型

2.1　网络体系结构

车辆协同缓存架构如图2所示，本文考虑由N个车辆和提供服务的云数据中心（CDC, cloud data center）组成的城市交通网络模型，在该模型中，将城市交通网络中的各个参与者车辆定义为节点。请求车辆（RV, requesting vehicle）节点指需要获取内容的车辆，而缓存车辆（CV, caching vehicle）节点指缓存了请求内容的车辆，这些内容通常从CDC提前下载以提供服务。由于车辆的机动性，车辆可能会离开当前缓存车辆的覆盖范围。当一个节点发现与原来的缓存节点断开连接时，会立即请求最近的缓存节点与自身建立连接。同时，如果车辆无法从当前缓存车辆获取所有请求的内容，则会移动到下一个缓存车辆的覆盖区域，以继续请求剩余内容。

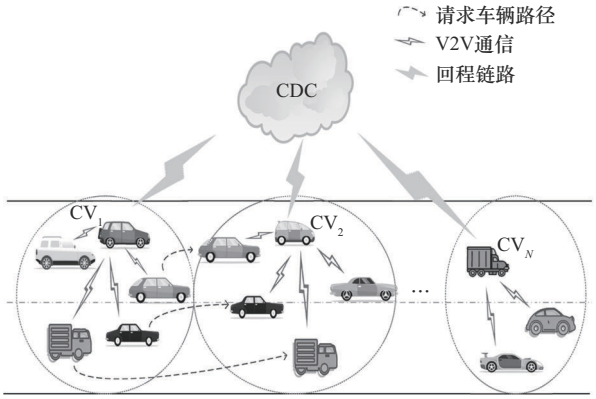

图2　车辆协同缓存架构

假设所有车辆的索引集为$Q_k^{\theta_i}\left(s_1^t,s_2^t,\cdots,s_M^t,a_1^t,a_2^t,\cdots,a_M^t\right)$，并且该路段所需的流行内容被分为$w$个内容块，集合$\mathcal{F}=\left\{f_1,f_2,\cdots,f_w\right\}$代表整个流行内容，每个内容块的大小为$L_w$。请求车辆向缓存车辆发出请求内容，而请求车辆所需要的内容块个数为Q且$Q\leqslant w$，用户在任何时间t对内容块的请求表示为$\mathrm{req}\mathcal{F}(t)=\left\{\mathrm{req}f_1(t),\cdots,\mathrm{req}f_q(t),\cdots,\mathrm{req}f_Q(t)\right\}$。

为了更清晰地建模车辆之间的交互，本文假设不同的缓存车辆的覆盖范围区域不重叠，并采用动态图$G=(V,E)$来描述车辆之间的交互。本文定义车辆状态集合为车辆在不同时间步长可能具有的所有状态。在每个时间步长t中，车辆集合为$V=\left\{v_i^{(t)}\middle|i=1,\cdots,n;\ t=1,\cdots,T\right\}$，表示交通场景中的一组车辆，而边集合为$E=\left\{e_{ij}^{(t)}\middle|\left(v_i^{(t)},v_j^{(t)}\right)\right\}$，表示车辆之间的交互关系。

2.2　通信模型

在本文提出的动态缓存策略中，由于请求车辆从缓存车辆获取内容的时延不同，为了减小因频繁切换而导致的信令开销，请求车辆仅选择具有最低时延的传输链路来获得内容。由于车辆请求的数据量相对于数据传输速率较小，因此可以忽略车辆发送请求所引起的传输时延。通过V2V通信链路，车辆可以将其缓存的内容传输到请求车辆。然而，由于传输范围的可变性和车辆的高移动性，V2V传输链路是不稳定的。当请求车辆与缓存车辆之间的距离超出通信范围时，该连接将断开，导致缓存车辆和请求车辆之间的通信中断。

在V2V通信范围内，RV与CV之间的数据传输速率为

$$r_{\mathrm{RV,CV}}(t)=B\mathrm{lb}\left(1+\frac{p_{\mathrm{CV}}h_{\mathrm{RV,CV}}}{\sigma^2+\sum\limits_{\mathrm{RV}'=\mathrm{RV}}p_{\mathrm{CV}}h_{\mathrm{RV}',\mathrm{CV}}}\right) \tag{1}$$

其中，B为传输带宽，σ^2为加性白高斯噪声，p_{CV}为CV的发射功率，$h_{\mathrm{RV,CV}}$为RV和CV之间的信道增益，$\sum\limits_{\mathrm{RV}'=\mathrm{RV}}p_{\mathrm{CV}}h_{\mathrm{RV}',\mathrm{CV}}$为其他车辆对RV和CV之间通信链路的干扰。数据传输时延用$T^r_{\mathrm{RV,CV}}$表示，由内容大小和数据传输速率共同决定

$$T^r_{\mathrm{RV,CV}}=\frac{f_q}{r_{\mathrm{RV,CV}}(t)} \tag{2}$$

2.3 问题建模

本文将车辆缓存问题表述为在每个时间周期内决定数据在缓存车辆中存储和更新的优化问题，目标是最小化在缓存存储容量限制下的平均数据传输时延。假设车辆网络运行在具有固定长度时隙的离散时间模型中，时隙长度为$\tau=\{0,1,2,\cdots,T\}$。在每个时隙$t\in\tau$中，车辆只允许请求一个内容，且全局缓存策略会定期更新。二进制指示符$X^t\in\{0,1\}$表示协作缓存替换方案，当$x^t_{f,i}\in X^t$为1时，表示内容f在时间t已经缓存在第i个CV中；否则，$x^t_{f,i}$为0。具体而言，将问题表述为

$$\begin{aligned}&\mathrm{P0}:\min\frac{1}{t}\sum_{t=1}^{\infty}\sum_{i=1}^{N}\lambda_i(t)T^r_{\mathrm{RV,CV}}\\&\text{s.t.}\\&\mathrm{C1}:\sum_{f=1}^{\mathcal{F}}L_f x^t_{f,i}\leqslant C_i,\quad\forall i\in V,t\in\{0,1,2\cdots\}\\&\mathrm{C2}:x^t_{f,i}\in\{0,1\},\quad\forall f\in\mathcal{F},t\in\{0,1,2\cdots\}\\&\mathrm{C3}:\sum_{f=1}^{\mathcal{F}}x^t_{f,i}\leqslant 1,\quad\forall i\in V\end{aligned} \tag{3}$$

其中，P0是目标函数，旨在决定数据在每个时间周期内存储在缓存中的方式，使平均数据传输时延最小化；$\lambda_i(t)$是数据i被请求的概率；$T^r_{\mathrm{RV,CV}}$是数据f在时间周期t的传输时延；约束条件C1确保CV缓存内容的总量不超过其容量限制；C2确保决策变量的非负性和完整性；C3保证缓存节点不允许缓存重复内容。

由于问题P0具有非凸性[21]，因此求解难度较大。此外，在实际场景中，可能无法获得所有时间周期内网络状态的完整信息，从而使得问题P0的全局优化面临挑战。在多车辆场景中，每个缓存车辆根据其本地缓存状态做出缓存k决策，并且无法获得其他（相邻）缓存节点的缓存状态信息。实际上，缓存节点无法观察到关于缓存状态和内容请求分布的完整系统信息，因此难以做出有效的缓存决策。为了解决这一问题，在下一节中将该问题表示为部分可观察的马尔可夫决策过程[22]，并使用深度强化学习提出一种内容协作缓存接力算法，以寻找最佳缓存决策，从而最小化内容传输时延。

3 基于递归深度强化学习的缓存接力策略

3.1 基于图神经网络的车辆移动性预测

考虑车辆轨迹的可预测性，本文提出一个基于GNN的框架，以预测所覆盖范围内下一周期的拓扑结构。在该框架中，采用图结构来建模空间交互行为。时间序列中的相互作用通过LSTM进行捕获和建模，LSTM描述了顺序输入的相关性和依赖性，并被应用于图的节点和边上，从而形成GNN结构。在本文中，从交互事件识别（IER, interaction event recognition）和轨迹预测（TP, trajectory prediction）2个角度对车辆交互行为进行建模和预测，分别通过层次框架中的IER和TP模块实现。

在IER模块中，通过GNN基于历史轨迹信息进行训练，以识别车辆之间的交互事件。在GNN中，每辆车被建模为图结构中的节点，而两辆车之间的交互则由连接它们空间分布的节点的空间边表示。LSTM被应用于每个节点和边，以描述事件序列中的相关性，IER模块可以用函数ϕ_{IER}表示为

$$\boldsymbol{O}_{\mathrm{IER}}=\phi_{\mathrm{IER}}\left(s_{t-n:t}\right) \tag{4}$$

其中，$s_{t-n:t}=\left[s^{t-n},\cdots,s^{t-1},s^t\right]$是一组从$t-n$到$t$的输入索引，$\boldsymbol{O}_{\mathrm{IER}}$是一组根据不同车辆的交互事件的识别结果。在时间$t$，输入的$s^t$特征定义为

$$s^t=\left(x^t,y^t,c^t\right) \tag{5}$$

在TP模块中，由IER模块识别的历史轨迹和交互事件被整合为GNN的输入。TP模块中的LSTM输出顺序轨迹，为了确保车辆表示的一致性，TP模块采用与IER模块相同的图形结构。TP模块的功能可以表示为

$$\boldsymbol{O}_{\mathrm{TP}}=\phi_{\mathrm{TP}}\left(\boldsymbol{O}_{\mathrm{IER}},s_{t-n:t}\right) \tag{6}$$

其中，$\boldsymbol{O}_{\mathrm{TP}}=\left[x^{t+1},y^{t+1};\cdots;x^{t+m-1},y^{t+m-1};x^{t+m},y^{t+m}\right]$是从时间$t+1$到$t+m$的节点预测轨迹。TP模块具有与IER模块相似的GNN结构，该模块输出预测轨迹为$[x_i^{t+1:t+\mathrm{pred}},y_i^{t+1:t+\mathrm{pred}}]$。

3.2　基于负载约束的车辆缓存节点选择算法

在预测下一时刻的车辆行驶轨迹后，定义车辆之间的链路定性度量，并构建预测权重邻接矩阵，以微观地描述车辆的拓扑关系。将节点在一个周期内可服务的节点个数作为负载约束，并根据最小支配集算法选出最优的缓存节点。

1) 连接稳定性度量

本文定义连接稳定性度量为$\omega_{i,j}^{t}$，$\forall i,j \in \left[1,N_{\text{veh}}\right]$，表示第$t$个周期车辆节点$i$和$j$之间归一化通信距离容差和归一化链路持续时间的加权和。

在第t个周期，假设车辆节点i和j在各自的通信范围内，且距离为$\varepsilon_{i,j}^{t}$，车辆节点的通信半径为R_{veh}，因此通信距离容差为$R_{\text{veh}}-\varepsilon_{i,j}^{t}$，从而获得归一化通信距离容差为

$$\tilde{\varepsilon}_{i,j}^{t}=\begin{cases}\dfrac{R_{\text{veh}}-\varepsilon_{i,j}^{t}}{R_{\text{veh}}}, & d_{\min}\leqslant\varepsilon_{i,j}^{t}<R_{\text{veh}}\\ 0, & \varepsilon_{i,j}^{t}\geqslant R_{\text{veh}}\end{cases}\tag{7}$$

变量$d_{\min}$表示在安全距离限制下两车的最小距离，$d_{\min}$值越大说明两车之间的距离越近，且在相同遮挡条件下信道质量越好。此外，在两车速度不变的情况下，链路持续时间也会更长。

假设周期间隔为Δt，在第t个周期，车辆节点i和j之间的相互覆盖时间长度被定义为链路持续时间$\tau_{i,j}^{t}$。由于链路持续时间在每个周期进行更新，当其值大于周期间隔时，链路持续时间的上限被设为Δt，并进行归一化，从而得到归一化的链路持续时间为

$$\tilde{\tau}_{i,j}^{t}=\begin{cases}\dfrac{\tau_{i,j}^{t}}{\Delta t}, & \tau_{i,j}^{t}<\Delta t\\ 1, & \tau_{i,j}^{t}\geqslant\Delta t\end{cases}\tag{8}$$

链路持续时间越长，说明两车的拓扑关系越稳定。将连接稳定性度量简化为归一化通信距离容差和归一化链路持续时间的加权和$\omega_{i,j}^{t}=\alpha\tilde{\varepsilon}_{i,j}^{t}+(1-\alpha)\tilde{\tau}_{i,j}^{t}$，其中$\alpha\in[0,1]$为加权因子。

2) 构建预测权重邻接矩阵

本文使用预测的轨迹信息来更新邻接矩阵的权重，从而生成预测权重邻接矩阵$\boldsymbol{W}^{t+1}=\left[\omega_{i,j}^{t+1}\right]_{N_{\text{veh}}\times N_{\text{veh}}},\forall i,j\in\left[1,N_{\text{veh}}\right]$，链路的权重$\omega_{i,j}^{t+1}$表示第$t+1$个周期车辆节点$i$和$j$之间连接稳定性度量的值。

根据预测的$\boldsymbol{W}^{t+1}$，采用最小支配集算法计算缓存节点集合$\mathcal{M}^{t+1}=\left\{m_1,m_2,\cdots,m_{N_M}\right\}$，其节点个数为$N_M$。为减少管理缓存节点的开销和缓存节点之间的信道竞争，应在满足最优性能的前提下，选择缓存节点的个数最少。

3) 最小支配集算法

针对预测权重邻接矩阵$\boldsymbol{W}^{t+1}$，定义3种车辆节点的状态，即状态待定节点、请求节点以及缓存节点，分别对应状态标志位0、1、2，从而构建第$t+1$个周期的节点标志位向量为

$$\boldsymbol{H}_{1\times N_{\text{veh}}}^{t+1}=\left(h_1^{t+1},h_2^{t+1},\cdots,h_{N_{\text{veh}}}^{t+1}\right)\tag{9}$$

其中，$\forall h_i^{t+1}\in[0,1,2]$，$i\in\left\{1,2,\cdots,N_{\text{veh}}\right\}$。根据预测权重邻接矩阵，计算每个节点连接状态未定节点的连接度，即该节点通信覆盖范围内连接的状态未定节点的数量。在给定数据帧长和数据传输速率的情况下，对于任何缓存节点m_k，$k\in\left\{1,2,\cdots,N_M\right\}$，每个周期的服务能力是有限的，因此将缓存节点最多能响应$N_{\max}$个请求节点的请求作为约束条件。

缓存节点m_k将在其通信半径范围内选择不多于$N_{\max}$个节点作为请求邻居节点，将缓存节点m_k的q个请求邻居节点集合记为$\chi_{m_k}^{t+1}=\left\{x_{m_k,1}^{t+1},x_{m_k,2}^{t+1},\cdots,x_{m_k,q}^{t+1}\right\}$，这些缓存节点能够覆盖通信范围内的所有普通节点。

为了实现无重复响应，每个普通节点只能与一个缓存节点建立连接，即在同一周期内，一个普通节点对一份文件的请求不能同时被2个或2个以上缓存节点响应。即任意2个缓存节点m_k和m_u的邻居节点集合交集为空，$\mathcal{X}_{m_k}^{t+1}\cap\mathcal{X}_{m_u}^{t+1}=\varnothing$，$k\neq u$。

在确保每个普通节点仅被一个缓存节点覆盖的前提下，每个缓存节点应选择其覆盖范围内链路权重最大的节点作为服务邻居节点。缓存节点应选择其服务邻居节点，以最大化平均链路权重。设缓存节点m_k的最优邻居集合为$\chi_{m_k}^{*t+1}$，对应的平均链路权重为$\bar{\boldsymbol{W}}_{m_k}^{*t+1}$，则目标函数为$\max\limits_{\left\{\chi_{m_k+1}\right\}}E\left[\sum\limits_{k=1}^{N_M}\bar{\boldsymbol{W}}_{m_k}^{*t+1}\right]$，$\forall m_k\in\mathcal{M}^{t+1}$。

3.3　基于递归深度强化学习的协作缓存替换算法

3.3.1　用于多车辆协作缓存的深度强化学习模型

由于每辆车的通信范围有限，请求车辆很难在

缓存车辆的覆盖范围内缓存完整的内容，尤其是在车辆速度较快或所需文件较大时。为了有效管理缓存空间，需要一种更灵活的协作缓存接力算法。在缓存接力阶段，缓存车辆更新其缓存内容，以提高缓存利用率和效率。基于此，本文采用深度强化学习算法来解决缓存车辆内容更新问题，首先构建马尔可夫决策过程。在此模型中，假设车辆可以完全观察到马尔可夫决策问题中的缓存状态，但单个车辆只能观察到自身的局部状态。因此，缓存决策问题采用MDP模型，MDP被定义为一个元组$\{S,\mathcal{A},R\}$。其中，S是系统状态，$\mathcal{A}$是系统动作，R是奖励函数。

1) 系统状态

系统状态空间表示为

$$s^t=\left(\boldsymbol{X}^t,\boldsymbol{B}^t,\boldsymbol{P}^t\right) \tag{10}$$

其中，$\boldsymbol{X}^t=\left[x_1^t x_2^t\cdots x_W^t\right]$表示缓存周期$t$内缓存车辆$m$上内容的缓存状态向量，$x_w^t$表示缓存周期$t$内缓存车辆$m$中流行内容分块$f_w$的缓存状态，如果缓存车辆缓存了流行内容分块$f_w$，则令$x_w^t=1$；否则，令$x_w^t=0$。

$\boldsymbol{B}^t=\left[B_1^t B_2^t\cdots B_H^t\right]$，$\boldsymbol{B}_h^t=\left[b_{h,1}^t b_{h,2}^t\cdots b_{h,W}^t\right]$表示截至缓存周期$t$，服务邻居车辆集合$\beta_m$中车辆$h$对流行内容的请求状态向量，$h\in\beta_m$。其中，$b_{h,w}^t$表示车辆$h$对流行内容分块$f_w$的请求状态，如果车辆$h$请求了流行内容分块$f_w$，则令$b_{h,w}^t=1$；否则，令$b_{h,w}^t=0$。

$\boldsymbol{P}^t=\left[P_1^t P_2^t\cdots P_2^t\right]$，$\boldsymbol{P}_h^t=\left[p_{h,1}^t p_{h,2}^t\cdots p_{h,W}^t\right]$表示截至缓存周期$t$，服务邻居车辆集合$\beta_m$中车辆$h$对流行内容的缓存状态向量。其中，$p_{h,w}^t$表示车辆$h$对流行内容分块$f_w$的缓存状态，如果车辆$h$缓存了流行内容分块$f_w$，则令$p_{h,w}^t=1$；否则，令$p_{h,w}^t=0$。

2) 系统动作

系统动作$\mathcal{A}$为下一个缓存周期$t+1$内缓存车辆m上内容的缓存状态向量$\boldsymbol{X}^{t+1}=\left[x_1^{t+1}x_2^{t+1}\cdots x_W^{t+1}\right]$，其中，$x_w^{t+1}$表示在下一个缓存周期$t+1$内，缓存车辆$m$中流行内容分块$f_w$的缓存状态。如果缓存车辆$m$需要缓存流行内容分块$f_w$，则令$x_w^{t+1}=1$；否则，令$x_w^{t+1}=0$。同时，缓存状态向量$\boldsymbol{X}^{t+1}=\left[x_1^{t+1}x_2^{t+1}\cdots x_W^{t+1}\right]$满足$\sum_{w=1}^{W}x_w^{t+1}L_w\leqslant\tau_m$，$\tau_m$表示缓存车辆$m$的缓存容量上限。

3) 奖励函数

对服务邻居车辆集合β_m中的每个车辆h，根据其请求状态向量$\boldsymbol{B}_h^t=\left[b_{h,1}^t b_{h,2}^t\cdots b_{h,W}^t\right]$、缓存状态向量$\boldsymbol{P}_h^t=\left[p_{h,1}^t p_{h,2}^t\cdots p_{h,W}^t\right]$和缓存车辆在下一个缓存周期$t+1$内的缓存状态向量$\boldsymbol{X}^{t+1}=\left[x_1^{t+1}x_2^{t+1}\cdots x_W^{t+1}\right]$，得到下一个缓存周期$t+1$内车辆$h$的可缓存流行内容集合$F_h$，计算得到可缓存流行内容集合$F_h$中所有内容的大小$L_h$，获取车辆$h$和缓存车辆$m$之间的传输速率$\text{rate}_{m,h}$，计算得到在下一个缓存周期$t+1$内车辆$h$可获取的最大传输数据量$g_h=\text{rate}_{m,h}\times\Delta t$。最后，采用式(11)计算得到缓存车辆$m$向车辆$h$传输内容的传输成本$\text{c_tr}_{m,h}$为

$$\text{c\_tr}_{m,h}=\begin{cases}\rho_1 L_h, & L_h\leqslant g_h\\ \rho_1 g_h, & L_h>g_h\end{cases} \tag{11}$$

其中，ρ_1为单位传输成本。

对比缓存状态向量$\boldsymbol{X}^t=\left[x_1^t x_2^t\cdots x_W^t\right]$和缓存状态向量$\boldsymbol{X}^{t+1}=\left[x_1^{t+1}x_2^{t+1}\cdots x_W^{t+1}\right]$，得到删除缓存内容集合$\chi_1$和增加缓存内容集合$\chi_2$，然后计算得到缓存车辆$m$的调整成本为

$$\text{c\_ad}_m=\sum_{w\in\chi_1}\rho_2 L_w+\sum_{w\in\chi_2}\rho_3 L_w \tag{12}$$

其中，ρ_2为删除缓存内容的单位成本，ρ_3为增加缓存内容的单位成本。此外，采用式(13)计算得到奖励函数R为

$$R=\frac{1}{\sum_{h\in\beta_m}\text{c\_tr}_h+\text{c\_ad}_m} \tag{13}$$

3.3.2 Actor-Critic框架

Actor-Critic算法提供了一个学习框架，包含2个独立的神经网络Actor和Critic。Actor负责学习最大化回报的策略，而Critic则对当前策略的价值函数进行估计，即评估Actor的优劣。在给定状态下，Actor根据策略选择一个动作。在与环境相互作用后，如果车辆做出一个动作，则环境更新状态并获得一定的奖励，每个奖励依赖于前一个奖励和动作。Critic评估当前策略的权重，并通过TD算法更新其价值函数。随后，Actor利用来自Critic的信息更新其策略。这个过程重复进行，直到该路段缓存任务完成。

1) Actor网络

Actor网络的输出是当前状态s对应动作$\mathcal{A}$的

缓存策略 $\pi=\{\pi_1,\pi_2,\cdots,\pi_M\}$，车辆 k 选择一个动作 A_k^t，取决于它的状态 s_k^t 和策略 $\pi_k^{\varnothing_k}$

$$A_k^t=\pi_k^{\varnothing_k}\left(s_k^t\right) \tag{14}$$

2) Critic 网络

在训练阶段，用于近似行动—价值函数 $V(S)$ 的 Critic 网络提供了基于状态 s_k^t 和全局状态 g 下采取动作 A_k^t 的总体回报。每个车辆在环境中执行动作 $A^t=\{A_1^t,A_2^t,\cdots,A_M^t\}$ 后，更新当前状态信息 s_k^t 以及从环境中获得的奖励 R^t，再将这些信息发送给 Critic 网络，该网络反馈下一时刻的状态信息 s^{t+1} 和奖励。对于每个车辆，Q 函数定义为 $Q_k^{\theta_i}\left(s_1^t,s_2^t,\cdots,s_M^t,a_1^t,a_2^t,\cdots,a_M^t\right)$，这解决了由非平稳环境引起的问题。考虑 M 个车辆，车辆 k 的策略为 π_k，则状态转移函数为

$$P\left(s_k^{t+1}s_k^t;\mathrm{Env}\right)=$$

$$P\left(s_k^{t+1}\middle|s_k^t;s_1^t,s_2^t,\cdots,s_M^t,A_1^t,A_2^t,\cdots,A_M^t,\pi_1,\pi_2,\cdots,\pi_M\right)=$$

$$P\left(s_k^{t+1}\middle|s_k^t;s_1^t,s_2^t,\cdots,s_M^t,A_1^t,A_2^t,\cdots,A_M^t,\pi_1',\pi_2',\cdots,\pi_M'\right) \tag{15}$$

通过最小化损失函数 $\mathcal{L}_{(\theta_k)}$，每个 Critic 更新其网络模型参数。损失函数 $\mathcal{L}_{(\theta_k)}$ 由一组参数 $\theta=\{\theta_1,\theta_2,\cdots,\theta_M\}$ 参数化得到，用于优化每个 Critic 网络的性能

$$\mathcal{L}\left(\theta_k\right)=E_{s^t,a^t}\left[\left(Q_k^{\theta_k}\left(s^t,A^t\right)-y^t\right)^2\right] \tag{16}$$

其中，$s^t=\{s_1^t,s_2^t,\cdots,s_M^t\}$，$A^t=\{A_1^t,A_2^t,\cdots,A_M^t\}$。

使用当前策略网络和目标值函数网络计算 TD 目标为

$$y^t=R^t+\gamma Q_k^{\theta_{k+1}}\left(s^t,A^t\right)\Big|_{A_k^t=\pi_k^{\varnothing_k'}\left(s_k^{t+1}\right)} \tag{17}$$

其中，γ 是未来累积奖励的折现因子，$0\leqslant\gamma\leqslant1$。

每辆车通过直接优化由参数 $\varnothing$ 参数化的策略来最大化奖励回报，因此，目标是最大化累积奖励函数

$$J\left(\varnothing_k\right)=E_{s^t,A^t}\left[Q_k^{\theta_k}\left(s^t,A^t\right)A_{s_k}^t=\pi_k^{\theta_k(k)}\right] \tag{18}$$

上述算法虽然能够处理合作任务，但无法处理部分可观测的环境或历史依赖的决策，因为每个节点需要跟踪来自所有边缘节点的先前交互。为了解决这个问题，本文机制引入了 LSTM 的多代理 Actor-Critic 架构，使得车辆在通信过程中能够记住上一次的信息，从而增强多车辆环境中的协作效果。

在 RDRL-CR 算法中，将递归神经网络 LSTM 加入 Actor-Critic 网络中，这样可以记住其他车辆的行动对奖励的影响。Actor 和 Critic 网络的历史信息分别用 h_a^t 和 h_c^t 表示，每个车辆基于先前的状态 h_k^t 选择动作，即 $A_k^t=\pi_k^{\varnothing_k}\left(h_k^t\right)$。然后 Q 函数变成 $Q_k^{\theta_k}\left(h_c^t,A^t\right)$，其中，$h_c^t=\{h_{c,1}^t,h_{c,2}^t,\cdots,h_{c,M}^t\}$。同样，Critic 网络中的损失函数 $\mathcal{L}_{(\theta_k)}$ 为

$$\mathcal{L}\left(\theta_k\right)=E_{h_c^t,A^t}\left[\left(Q_k^{\theta_k}\left(h_c^t,A^t\right)-y^t\right)^2\right] \tag{19}$$

目标函数表示为

$$y_k^t=r_k^t+\gamma Q_k^{\theta_k'}\left(h_c^{t+1},\pi_1^{\theta_1'}\left(h_{a,1}^{t+1}\right),\cdots,\pi_M^{\theta_M'}\left(h_{a,M}^{t+1}\right)\right) \tag{20}$$

重放缓存器在训练阶段负责存储经验信息，在每个训练步骤中，从重放缓存器中随机采样回放数据，以更新 Critic 网络和 Actor 网络，目标 Critic 网络和 Actor 网络参数分别用 θ' 和 $\varnothing'$ 表示。

缓存车辆接收用户请求并提取其特性，当前请求和缓存状态随后被传递给 Actor 网络，以获取相应的缓存操作。在执行策略产生的动作后，每个车辆都会得到奖励和下一个状态信息。为有效处理和存储这些信息，使用 LSTM 网络将从环境中接收的信息存储为历史记录

$$\begin{aligned}h_a^{t+1}&=\mathrm{LSTM}\left(h_a^t,s^{t+1}\right)\\h_c^{t+1}&=\mathrm{LSTM}\left(h_c^t,s^{t+1},A^t\right)\end{aligned} \tag{21}$$

其中，h_a^t 和 h_c^t 分别为 Actor 和 Critic 网络的历史记录。

Actor 和 Critic 网络将历史记录存储在重放记忆中，为了训练 Actor 和 Critic 网络，从回放内存中随机选择一个小批量的转换数据。对于一个样本，设置目标 Critic 网络为

$$\begin{aligned}y_{k,j}^t=r_{k,j}^t+\gamma Q_k^{\theta_k'}&\left(h_{c,1}^{t+1,j},\cdots,h_{c,M}^{t+1,j},\pi_1^{\theta_1'}\left(h_{a,1}^{t+1,j}\right),\cdots,\right.\\&\left.\pi_N^{\theta_N'}\left(h_{a,N}^{t+1,j}\right)\right)\end{aligned} \tag{22}$$

通过减少小批量数据上的损失函数来更新其参数 θ 的值

$$\mathcal{L}\left(\theta_k\right)=\frac{1}{S}\sum_{j\in S}\left(Q_k^{\theta_k}\left(h_{c,1}^{tj},\cdots,h_{c,M}^{tj},A_{1,j}^I,\cdots,A_{M,j}^I\right)-y_{i,j}^I\right)^2 \tag{23}$$

此外，Actor 网络利用批量梯度下降更新参数 $\varnothing$，并使用损失函数计算策略梯度

$$\begin{aligned}\nabla_{\varnothing_k}J\left(\varnothing_k\right)\approx\frac{1}{S}\sum_{j\in S}&\nabla_{\varnothing_{k,\pi_k^{\varnothing_k}}}\left(h_{a,k}^{tj}\right),\nabla_{a_k}Q_k^{\theta_k}\left(h_{c,1}^{tj},\cdots,h_{c,M}^{tj},\right.\\&\left.\pi_1^{\varnothing_1}\left(h_{a,1}^{tj}\right),\cdots,\pi_N^{\varnothing_N}\left(h_{a,N}^{tj}\right)\right)\end{aligned} \tag{24}$$

更新目标网络的 Actor 和 Critic 参数为

$$\theta_k' \leftarrow \tau\theta_k + (1-\tau)\theta'$$
$$\varnothing_k' \leftarrow \tau\theta_k + (1-\tau)\varnothing \tag{25}$$

4 仿真分析

本节从缓存系统性能的角度对本文提出的 RDRL-CR 算法的性能进行评价。本文模拟了一个由智能车辆组成的城市信息网络环境，该环境中不包含 RSU。仿真采用真实的城市交通网络数据集，以获取车辆的历史轨迹。具体而言，本文使用 Apolloscape 数据集[23]作为车辆轨迹数据集，这是一个公开的自动驾驶场景数据集，包括高清视频、稀疏激光点云、3D 物体检测和车辆轨迹数据。从 Apolloscape 数据集中选择城市街道场景的车辆轨迹数据，考虑网络中有 10 000 条内容，内容大小为 5~10 MB 不等。此外，每个 CV 的缓存大小设置为 300 MB 或 2 GB，使用以下通道模型进行仿真。路径损耗（dB）为 36.8+36.7 lg(d)，其中 d 为距离，单位为 m；对数正态遮蔽参数为 7 dB；天线增益为 5 dBi；小尺度衰落服从单位方差的瑞利分布；信道带宽为 20 MHz。

本节将本文提出的 RDRL-CR 算法与最不频繁使用（LFU, least frequently used）[24]、先进先出（FIFO, first-in-first-out）[25]、分布式协作缓存（DCC, distributed collaborative caching）策略[26]、基于深度强化学习的缓存替换策略（DRL）[27]以及基于机器学习的自适应缓存替换（ML-based ACCR, machine learning-based adaptive cache replacement）策略[28]进行比较。

1) LFU

LFU 是一种基于访问频率的机制，当缓存已满时，使用获取的文件来更新请求次数最少的文件。

2) FIFO

FIFO 是基于到达顺序的机制，当缓存已满时，使用获取的文件来更新最早缓存的文件。

3) DRL

DRL 是一种缓存替换决策机制，利用深度神经网络来学习缓存替换决策。在 DRL 中，每个 MEC 在本地观察的基础上独立执行缓存决策，而不考虑其他 MEC 的影响。

4) ML-based ACCR

基于机器学习的自适应缓存替换策略 ML-based ACCR 通过机器学习算法来优化缓存替换决策。该策略通过对历史的缓存数据和请求进行学习和分析，构建预测模型以预测未来的请求。

为了考虑缓存车辆不同的缓存容量对算法的影响，本文将缓存车辆的缓存容量选择范围控制在 4~20 GB。缓存容量对缓存命中率和内容访问时延的影响分别如图 3 和图 4 所示。

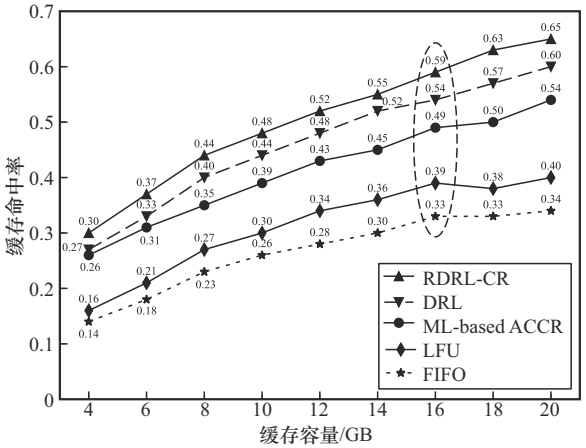

图 3 基于缓存容量的缓存命中率性能评估

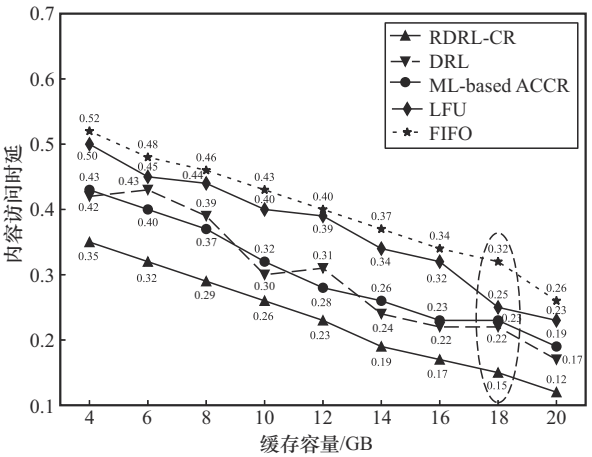

图 4 基于缓存容量的内容访问时延性能评估

图 3 展示了缓存容量对缓存命中率的影响，随着缓存容量的增加，缓存车辆能够缓存更多内容，从而更好地满足邻近请求车辆的用户请求。由图 3 可以看出，缓存命中率随着缓存容量的增加而上升。基于学习的机制相较于传统基于规则的替换机制更具优势，因为基于学习的机制能够从历史数据中捕获用户请求的特征。RDRL-CR 在性能上优于 DRL 和 ML-based-ACCR，DRL 缓存替换机制未考虑节点之间的合作，导致每个节点仅关注自身回报，而忽略其他节点的影响。与 FIFO、LFU、ML-based-ACCR

和DRL相比，本文提出的RDRL-CR算法的缓存命中率分别提高了26%、20%、10%和5%。

图4展示了缓存容量对内容访问时延的影响，仿真结果表明，随着缓存容量的增加，本文提出的RDRL-CR算法在内容访问时延方面始终优于其他算法。由于节点之间的合作，该机制相较于DRL表现出更高的稳定性。基于学习的方案明显优于FIFO和LFU，因为基于学习的方案能够根据用户请求来替换更合适的内容。与FIFO、LFU、ML-based-ACCR和DRL相比，RDRL-CR算法在内容访问时延方面分别降低了20%、17%、8%和7%。

如图5所示，内容数量对缓存命中率的影响显著。通过将内容数量从200个增加到1 000个来评估系统性能，由于缓存容量有限，缓存车辆需要频繁更新内容。仿真结果表明，RDRL-CR算法的性能优于其他参考算法。DRL算法不考虑节点之间的合作，每个节点都试图最大化自身的奖励而忽视其他节点，导致缓存命中率低于ML-based-ACCR和RDRL-CR。与FIFO、LFU、ML-based-ACCR和DRL相比，RDRL-CR算法的缓存命中率分别提高了25%、20%、10%和17%。

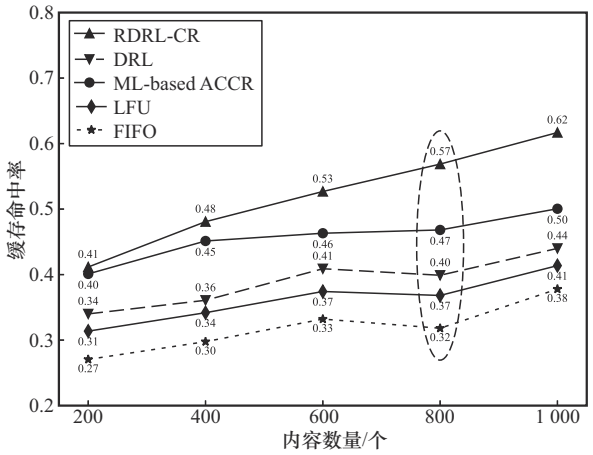

图5　基于内容数量的缓存命中率性能评估

如图6所示，内容数量对内容访问时延的影响显著。随着内容数量的增加，在缓存车辆的有限缓存容量下，缓存车辆上会产生更多的缓存更新。然而，本文提出的RDRL-CR算法的平均表现仍优于其他算法。与传统的缓存替换算法相比，基于学习的算法在时延方面表现更佳，因为基于学习的算法能够利用历史缓存数据预测可能被请求的内容，并优化缓存节点的内容替换策略，从而降低缓存未命中导致的时延。与FIFO、LFU、ML-based-ACCR和DRL相比，RDRL-CR算法在时延方面分别降低了19%、27%、8%和5%。

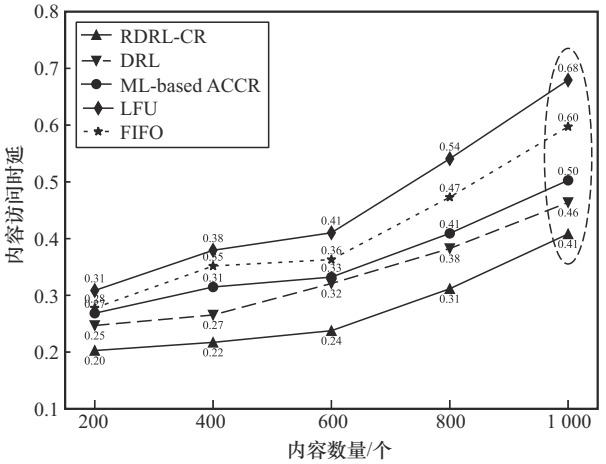

图6　基于内容数量的内容访问时延性能评估

5　结束语

本文考虑车辆间的协同缓存问题，提出一种基于递归深度强化学习的协作缓存接力算法，以减少往返负载和系统开销，解决车辆高速移动带来的缓存中断问题。通过将LSTM模型集成到递归DRL中，设计了一种适合多车协作的缓存中继算法，并详细讨论了Actor-Critic网络的更新机制。仿真结果表明，本文提出的算法在缓存命中率和内容访问时延方面优于其他基于学习和非学习（基于规则）的算法。

参考文献：

[1] 刘雷, 陈晨, 冯杰, 等. 车载边缘计算中任务卸载和服务缓存的联合智能优化[J]. 通信学报, 2021, 42(1): 18-26.
LIU L, CHEN C, FENG J, et al. Joint intelligent optimization of task offloading and service caching for vehicular edge computing[J]. Journal on Communications, 2021, 42(1): 18-26.

[2] LU Z J, QU G, LIU Z L. A survey on recent advances in vehicular network security, trust, and privacy[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(2): 760-776.

[3] SHOJAFAR M, CORDESCHI N, BACCARELLI E. Energy-efficient adaptive resource management for real-time vehicular cloud services[J]. IEEE Transactions on Cloud Computing, 2019, 7(1): 196-209.

[4] ZHU D J, DU H W, SUN Y D, et al. Massive files prefetching model based on LSTM neural network with cache transaction strategy[J]. Computers, Materials & Continua, 2020, 63(2): 979-993.

[5] SU Z, HUI Y L, XU Q C, et al. An edge caching scheme to distribute content in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2018, 67(6): 5346-5356.

[6] TAN L T, HU R Q, HANZO L. Twin-timescale artificial intelligence

aided mobility-aware edge caching and computing in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(4): 3086-3099.
[7] MUSA S S, ZENNARO M, LIBSIE M, et al. Mobility-aware proactive edge caching optimization scheme in information-centric IoV networks[J]. Sensors, 2022, 22(4): 1387.
[8] SU Z, HUI Y L, GUO S. D2D-based content delivery with parked vehicles in vehicular social networks[J]. IEEE Wireless Communications, 2016, 23(4): 90-95.
[9] ZHAO W C, QIN Y J, GAO D Y, et al. An efficient cache strategy in information centric networking vehicle-to-vehicle scenario[J]. IEEE Access, 2017, 5: 12657-12667.
[10] JAVED M A, ZEADALLY S. AI-empowered content caching in vehicular edge computing: opportunities and challenges[J]. IEEE Network, 2021, 35(3): 109-115.
[11] KHANAL S, THAR K, HUH E N. DCoL: distributed collaborative learning for proactive content caching at edge networks[J]. IEEE Access, 2021, 9: 73495-73505.
[12] HU Z W, ZHENG Z J, WANG T, et al. Roadside unit caching: auction-based storage allocation for multiple content providers[J]. IEEE Transactions on Wireless Communications, 2017, 16(10): 6321-6334.
[13] BITAGHSIR S A, DADLANI A, BORHANI M, et al. Multi-armed bandit learning for cache content placement in vehicular social networks[J]. IEEE Communications Letters, 2019, 23(12): 2321-2324.
[14] ZHAO Z L, GUARDALBEN L, KARIMZADEH M, et al. Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(8): 1786-1801.
[15] QIAO G H, LENG S P, MAHARJAN S, et al. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2020, 7(1): 247-257.
[16] ZHANG Y, LI C L, LUAN T H, et al. A mobility-aware vehicular caching scheme in content centric networks: model and optimization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(4): 3100-3112.
[17] HU B B, FANG L Y, CHENG X, et al. In-vehicle caching (IV-cache) via dynamic distributed storage relay (D^2SR) in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 843-855.
[18] WANG R Y, KAN Z W, CUI Y P, et al. Cooperative caching strategy with content request prediction in Internet of vehicles[J]. IEEE Internet of Things Journal, 2021, 8(11): 8964-8975.
[19] XIONG X, ZHENG K, LEI L, et al. Resource allocation based on deep reinforcement learning in IoT edge computing[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(6): 1133-1146.
[20] SONG J J, SHENG M, QUEK T Q S, et al. Learning-based content caching and sharing for wireless networks[J]. IEEE Transactions on Communications, 2017, 65(10): 4309-4324.
[21] ZHAO J J, LIU Y W, CHAI K K, et al. Joint subchannel and power allocation for NOMA enhanced D2D communications[J]. IEEE Transactions on Communications, 2017, 65(11): 5081-5094.
[22] XU W P, QIU R H, JIANG X Q. Resource allocation in heterogeneous cognitive radio network with non-orthogonal multiple access[J]. IEEE Access, 2019, 7: 57488-57499.
[23] HUANG X Y, WANG P, CHENG X J, et al. The Apolloscape open dataset for autonomous driving and its application[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(10): 2702-2719.
[24] MAHANANDA I G E, YOVITA L V, NEGARA R M. Performance of homogeneous and heterogeneous cache policy for named data network[C]//Proceedings of the 2022 10th International Conference on Information and Communication Technology (ICoICT). Piscataway: IEEE Press, 2022: 120-123.
[25] ZHOU H, WU T, ZHANG H J, et al. Incentive-driven deep reinforcement learning for content caching and D2D offloading[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(8): 2445-2460.
[26] LIU Y, MAO Y L, SHANG X J, et al. Distributed cooperative caching in unreliable edge environments[C]//Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 1049-1058.
[27] LIU X, XU S Y, YANG C, et al. Deep reinforcement learning empowered edge collaborative caching scheme for Internet of vehicles[J]. Computer Systems Science and Engineering, 2022, 42(1): 271-287.
[28] SETHUMURUGAN S, YIN J M, SARTORI J. Designing a cost-effective cache replacement policy using machine learning[C]//Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). Piscataway: IEEE Press, 2021: 291-303.

[作者简介]

吴红海（1979– ），男，河南洛阳人，博士，河南科技大学副教授，主要研究方向为移动多媒体计算、移动边缘计算。

王白冰（1998– ），女，河南南阳人，河南科技大学硕士生，主要研究方向为移动边缘缓存。

马华红（1979– ），女，河南洛阳人，博士，河南科技大学副教授，主要研究方向为人群传感网络、物联网。

邢玲（1978– ），女，河南洛阳人，博士，河南科技大学教授，主要研究方向为智能信息处理、信息语义分析、多媒体计算与网络智能信息处理、信息语义分析、多媒体计算与网络。