Software Engineering and Applications 软件工程与应用, 2025, 14(2), 471-483
Published Online April 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.142042

Hans汉斯

基于车边云协同的服务迁移优化策略研究

张　奕，韩　韧

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月2日；录用日期：2025年4月19日；发布日期：2025年4月29日

摘　要

随着车联网的发展和智能交通系统的完善，移动边缘计算和边缘缓存技术不断发展，在车联网服务迁移问题上发挥了巨大作用。为了更加合理利用边缘缓存资源、减小时延，本文提出了一种联合优化时延和缓存策略的“车－边－云”协同服务迁移方案。在该架构下，网络中的车载终端、路侧单元和云服务器可以协同进行任务计算，进行动态服务迁移。通过使用基于Hawkes过程的方法，根据历史请求信息更新不同内容类型的流行度，选择合适的边缘计算节点和缓存内容。本文以最大化系统的长期效益为目标，降低时延和提高命中率，形式化服务迁移问题。这是一个具有挑战性的非凸问题。通过将该问题建模为马尔科夫决策过程，并引入深度强化学习求解最优问题，提出了一种结合长短期记忆网络和近端策略优化算法的服务迁移算法。仿真实验结果表明，提出的策略优化方法比其他策略有更好的性能。

关键词

边缘计算，车边云协同，服务迁移，马尔科夫决策过程，深度强化学习

Research on Service Migration Optimization Strategy Based on Vehicle-Edge-Cloud Collaboration

Yi Zhang, Ren Han

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 2nd, 2025; accepted: Apr. 19th, 2025; published: Apr. 29th, 2025

Abstract

With the development of the Internet of Vehicles (IoV) and the improvement of intelligent transportation system, mobile edge computing (MEC) and edge cache technology are constantly developing,

文章引用: 张奕, 韩韧. 基于车边云协同的服务迁移优化策略研究[J]. 软件工程与应用, 2025, 14(2): 471-483.
DOI: 10.12677/sea.2025.142042

which play a huge role in the service migration of the Internet of vehicles. In order to make more reasonable use of edge cache resources and reduce the delay, this paper proposes a "Vehicle-Edge-Cloud" collaborative service migration scheme which jointly optimizes the delay and cache strategy. In this architecture, vehicles, Roadside Units (RSUs) and cloud server in the network can collaborate on task computation and dynamic service migration. By using a Hawkes process-based approach, the popularity of different content types is updated based on historical request information, and appropriate edge compute nodes and cached content are selected. To maximize the long-term benefits of the system, reduce the delay and improve the hit rate, formalize the problem of service migration. This is a challenging non-convex problem. By modeling the problem as Markov Decision Process (MDP) and introducing deep reinforcement learning (DRL) to solve the optimal problem, a service migration algorithm combining Long Short-Term Memory (LSTM) and Proximal Policy Optimization (PPO) algorithm is proposed. Simulation results show that the proposed strategy optimization method has better performance than other strategies.

Keywords

Edge Computing, Vehicle-Edge-Cloud Collaboration, Service Migration, Markov Decision Process, Deep Reinforcement Learning

1. 引言

近年来，通信网络的广泛部署和交通基础设施的不断完善，使得智慧交通系统正发生着巨大的革新[1]。作为车联网(Internet of Vehicles, IoV)模式的进一步发展，基于“车–边–云”协同的模式将车辆、道路基础设施和中心平台更紧密地连接起来，实现了人、车、环境的协调发展[2] [3]。同时，车载终端各种延迟敏感型和计算密集型任务不断涌现，例如道路交通导航、辅助驾驶功能、通信娱乐应用等[4]。

然而，车辆的计算能力是有限的，难以满足日益增长的计算需求。传统的云计算虽然拥有强大的计算能力和丰富的存储资源，但由于无法避免的长距离传输瓶颈，也会导致车辆和云服务器之间产生较高的时延。移动边缘计算(Mobile Edge Computing, MEC)技术被认为是一个很有前途的解决方案[5] [6]。它将计算资源和存储资源以分布式的方式部署在距离用户层更近的边缘节点上，使这些边缘节点就近处理其覆盖区域内的相关业务，从而减轻链路的传输压力，并节约服务响应时间[7]。同时，传统的 MEC 框架下，车辆需等待路侧单元(Roadside Unit, RSU)接受并处理请求后才能获取计算结果，加大了时延和驾驶风险[8]。为此，边缘缓存技术逐渐被重视，用来解决该问题以提升响应请求速度。

多年来，国内外学者针对服务迁移策略优化问题做出了深入研究。文献[9]提出了基于预测的主动服务迁移，并将其与被动服务迁移结合。降低了单纯基于预测可能导致的误差，联合优化了车辆边缘计算的服务迁移和资源管理。文献[10]考虑了边缘服务器只能获得部分用户信息，并将服务迁移问题建模为部分可观察的 MDP，以减少用户时延和系统能耗。文献[11]提出了一种基于多智能体深度强化学习的缓存方案，解决因数据流量的剧增而导致的响应延迟。文献[12]设计了一种通过使用移动性预测和一致性哈希的联邦学习协同缓存方案，以捕捉 IoV 的动态变化。此外，传统的边缘缓存技术常采取一些经典方法，如最近最少使用(Least Recently Used, LRU)和最少频繁使用(Least Frequently Used, LFU)等方法[13]。但这些方案在特定场景下的相应表现并不理想。

尽管已有的研究在服务迁移策略优化方面取得了显著进展，但是仍需进行更深入的探索。本文提出了一种联合优化时延和缓存策略的“车–边–云”协同服务迁移方案。基于 Hawkes 过程的方法，根据历史请求信息更新不同内容类型的流行度，选择合适的边缘计算节点和缓存内容。以降低时延和提高命中率为优化目标，形式化服务迁移问题，建模为 MDP 问题。并提出了一种结合长短期记忆网络(Long Short-Term Memory, LSTM)和近端策略优化算法(Proximal Policy Optimization, PPO)的服务迁移算法。通过一系列不同参数设置的仿真实验，结果表明提出的策略优化方法有更好的性能。

2. 系统模型与问题描述

在智慧交通背景的车边云协同计算架构下，本文针对基于 5G 通信和 V2X (Vehicle to Everything)通信技术的自动驾驶车辆，在多车道行驶场景中进行了建模研究。

整体网络模型分为车辆层、路侧单元层、云层，具体架构如图 1 所示。在车辆层，车辆行驶在直行的多车道上。每辆车都配备一个车载单元(On board Unit, OBU)，用于与 RSU 通信和处理服务。在路侧单元层，道路两侧均与分布有多个 RSU，并分别连接边缘服务器。RSU 作为边缘计算节点，具有通信能力和计算能力。在每个时隙 τ 中，每个 RSU 为其范围内的车辆提供服务，并可以相互传输通信以进行协作。在云层中，设有一个中心云服务器，有强大的计算、存储和网络资源，可进行快速的任务处理。

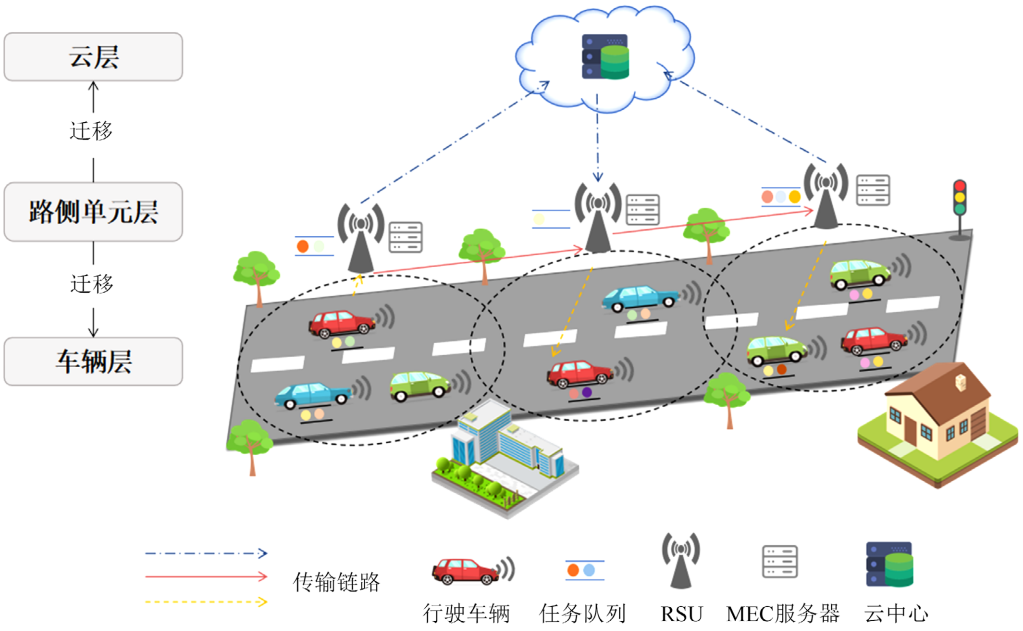

Figure 1. System architecture diagram based on Vehicle-Edge-Cloud collaboration
图 1. 基于车边云协同的系统架构图

本文采用时隙模型，即 $T=\{1,2,\cdots,\tau\}$。在每个时隙 τ 的开始，假设车辆可能对某个特定的场景产生服务请求。为了更好满足车辆的需求并提升服务质量，本文采用 RSU 主动感知道路环境及车辆请求的方式，并对待处理的服务任务进行计算迁移决策。在不损失通用性的情况下，将服务任务模型定义为式(1)。

$$S_r(\tau)=\left(d_{S_r(\tau)},\mu_{S_r(\tau)},o_{S_r(\tau)},h_{S_r(\tau)},e_{S_r(\tau)}\right) \tag{1}$$

其中，$d_{S_r(\tau)}$ 为任务的数据大小。$\mu_{S_r(\tau)}$ 为任务的难度系数，系数随着任务数据的增加而增加。$o_{S_r(\tau)}$ 为输出结果数据的大小。为了服务于高速行驶的车辆，需要将结果数据及时传输到 RSU 感知范围内的车辆。

因此需要施加时间限制，这里设置 $h_{S_r(\tau)}$ 为任务的有效时间。同时，考虑到任务内容的流行度因素，用 $e_{S_r(\tau)}$ 表示任务所属的类别。

2.1. 时延模型

每个 RSU 服务器能够实时获取行驶在其服务区内车辆的环境状态等信息。在感知到有待处理的任务时，系统首先会根据当前缓存和环境状况决策是否缓存该任务。若需要进行缓存，则从云层、路侧单元层、车辆层中选择一个合适的节点进行服务迁移并计算。该节点在计算完成后将结果传输给缓存的 RSU。若不需要缓存，则当感知到任务后不进行计算操作。

在迁移过程中，不仅需要将任务数据传输到目的节点，还需将计算结果返回至请求服务的车辆。通过优化迁移策略，系统能够在保证低延迟和高可靠性的同时，最大化资源利用效率，从而满足车联网环境下的多样化服务需求。在以上过程中，车辆、RSU、云服务器都会产生延迟，也就是车辆时延、RSU 时延和云端时延。按照时延产生的原因，又可以将时延划分为等待时延、计算时延和传输时延。

2.1.1. 车辆时延模型

当任务被迁移到车辆 m 上时，将计算时延定义为式(2)。其中，F_m 为车辆 m 的计算能力。

$$T_m^{comp}(\tau)=\frac{d_{S_r(\tau)}}{\mu_{S_r(\tau)}F_m} \tag{2}$$

式(3)中为任务在车辆 m 任务队列的等待延迟。假设每辆车的计算节点都有一个队列缓冲区，存储到达但未处理的计算任务。在时隙 τ，车辆 m 的任务队列长度为 L。在执行该任务之前，车辆 m 需要完成队列中前序排队的任务。

$$T_m^{wait}(\tau)=\sum_{l=1}^{L-1}\frac{d_{S_r(\tau)}^l}{\mu_{S_r(\tau)}^l F_m} \tag{3}$$

根据香农公式，可以得到车辆与 RSU 的传输率 $R_i^{m2r}(\tau)$，如式(4)所示。其中，B 为带宽，σ^2 为平均高斯白噪声[14]。

$$R_i^{m2r}(\tau)=B\log_2\left(1+\frac{p_m k_i^{m2r}}{\sigma^2}\right) \tag{4}$$

传输时延会出现在以下 2 种过程：服务请求从 RSU 到车辆，服务结果从车辆到 RSU。可得到车辆的传输时延为式(5)。

$$T_m^{tra}(\tau)=\frac{d_{S_r(\tau)}}{R_i^{m2r}(\tau)}+\frac{o_{S_r(\tau)}}{R_i^{m2r}(\tau)} \tag{5}$$

根据式(2)、式(3)和式(5)，可知车辆 m 在时隙 τ 的总时延为式(6)。

$$T_m^{total}(\tau)=T_m^{comp}(\tau)+T_m^{wait}(\tau)+T_m^{tra}(\tau) \tag{6}$$

2.1.2. RSU 时延模型

和车辆时延类似，RSU r 的计算时延和等待时延可定义为式(7)和式(8)。其中，F_r 为 RSU r 的计算能力。

$$T_r^{comp}(\tau)=\frac{d_{S_r(\tau)}}{\mu_{S_r(\tau)}F_r} \tag{7}$$

$$T_r^{wait}(\tau)=\sum_{l=1}^{L-1}\frac{d_{S_r(\tau)}^{l}}{\mu_{S_r(\tau)}^{l}F_r} \tag{8}$$

在考虑 RSU 传输时延时应分为 2 种情况。一种是请求任务的 RSU 和执行计算的 RSU 是同一个，即本地计算。此时计算结果将直接获取，不需要二次传输。另一种是该 RSU 作为边缘节点服务其他 RSU 的请求，计算结果 $o_{S_r(\tau)}$ 应传输到请求服务的 RSU。可合并写为式(9)。

$$T_r^{tra}(\tau)=\begin{cases}0, & \text{for } i\in R, i=r\\ \dfrac{d_{S_r(\tau)}}{R_i^{r2r}(\tau)}+\dfrac{o_{S_r(\tau)}}{R_i^{r2r}(\tau)}, & \text{for } i\in R, i\neq r\end{cases} \tag{9}$$

根据式(7)、式(8)和式(9)，可知 RSU r 在时隙 τ 的总时延为式(10)。

$$T_r^{total}(\tau)=T_r^{comp}(\tau)+T_r^{wait}(\tau)+T_r^{tra}(\tau) \tag{10}$$

2.1.3. 云端时延模型

和上述两种场景下的时延类似，云端时延也包括计算时延、等待时延和传输时延。由于中心云服务器上拥有丰富的资源和强大的计算能力，在提供服务方面有着巨大的优势。和边缘计算节点相比，可认为等待时延和计算时延忽略不计。最终云端时延为式(11)。

$$\begin{aligned}T_C^{total}(\tau)&=T_C^{comp}(\tau)+T_C^{wait}(\tau)+T_C^{tra}(\tau)\\&=T_C^{tra}(\tau)=\frac{d_{S_r(\tau)}}{R^{r2c}(\tau)}+\frac{o_{S_r(\tau)}}{R^{r2c}(\tau)}\end{aligned} \tag{11}$$

综上所述，任务的总延迟如式(12)所示。其中变量 $x_j(\tau)$ 表示服务迁移的决策变量，它被定义为 $x_j(\tau)\in\{0,1\}$。如果 $x_j(\tau)=1$，代表该不可分割的最小服务单元迁移到第 j 个节点。

$$T^{total}(\tau)=\sum_{j=1}^{M}x_j(\tau)T_m^{total}(\tau)+\sum_{j=M+1}^{M+R}x_j(\tau)T_r^{total}(\tau)+x_0(\tau)T_C^{total}(\tau) \tag{12}$$

2.2. 流行度模型

假设当前环境中存在 D 类不同的服务任务类型，即车辆会对 D 个类型的任务感兴趣。根据内容的类型差异，每类任务有各自的内容受欢迎程度 E_d。因此，受欢迎程度集合可表示为 $E_d\in\{E_1,E_2,\cdots,E_D\}$。内容流行度与车辆请求密切相关，会随着车辆流量和请求频率等因素动态变化。所以，为了提高服务质量，需及时响应过往车辆请求，需捕捉流行度的动态变化，对内容进行适时替换。

本文基于 Hawkes 过程构建内容流行度更新模型。它是一种特殊的自激线性模型。Hawkes 过程的核心思想在于：某事件的发生会增加后续事件再次发生的概率，这种影响具备激发性，并随着时间推移其效应会逐渐减弱。因此，构建指数衰减函数能够很好地描述这种自激励特性：当一个事件发生时，它会对未来产生一个瞬时的影响(由 φ 控制)，然后该影响会随着时间的推移逐渐衰减(由 δ 控制)。Hawkes 过程已被广泛应用于多个领域，例如地震学、神经科学、金融市场和犯罪行为建模等[15] [16]。

由于容量限制，RSU 需要选择内容进行缓存。且不同 RSU 的内容流行度的变化不同，需根据各自历史记录更新。在初始时刻 $\tau=0$ 时，假设所有内容受欢迎程度均为 0，即 $E_d(\tau)=0$，$d\in\{1,2,\cdots,D\}$。在每个时隙的开始，系统会对道路环境中车辆发出的所有请求内容进行收集和记录。利用 Hawkes 模型计算并更新受欢迎程度，公式为

$$E_d(\tau)=E_d(\tau-1)+\sum_{\tau_d\le\tau}\varphi e^{-\delta(\tau-\tau_d)} \tag{13}$$

$\varphi\ge 0$ 用来控制历史事件对未来影响的程度，在内容流行度变化较快的场景中通常会将值设置较大。δ 表示历史事件的时间衰减系数，当 δ 较小时说明历史事件的影响会持续较长时间，适合用于内容流行度较稳定的场景。在所有类流行度均计算完成后，将值进行归一化处理，存储至矩阵 E 中。基于当前流行度和历史信息，后续的缓存模型可根据 Hawkes 模型进一步预测未来时隙的内容热度。

2.3. 缓存模型

假设所有 RSU 具有相同的缓存容量上限，它们在每个时隙都会主动感知环境并做出缓存决策。为便于操作，将 RSU 内的缓存状态用矩阵表示为式(14)。

$$I_{r,d}=\left[i_{r,d}\right]_{R*D} \tag{14}$$

其中，$i_{r,d}\in\{0,1\}$ 表示 RSU r 对类别 d 结果的缓存状态。

每个时隙的开始，所有 RSU 会检查其缓存内容，并主动从缓存空间中移除已经过期的内容，更新缓存矩阵 $I_{r,d}$ 中对应位置为 0。当做出缓存决策 $c_{S_r(\tau)}$ 后，同样进行修正，更新公式为

$$i_{r,d}(\tau)=\begin{cases}i_{r,d}(\tau), & \text{for } i\in R, c_{S_r(\tau)}=0\\ H\left(c_{S_r(\tau)}=r\right), & \text{for } i\in R, c_{S_r(\tau)}\ne 0\end{cases} \tag{15}$$

其中，$H\left(c_{S_r(\tau)}=r\right)$ 是一个指示函数，$c_{S_r(\tau)}=r$ 时值为 1，否则为 0。

在缓存模型中，通过使用内容命中率衡量缓存操作有效性，以反应系统对车－边－云协同环境中车辆请求的响应情况。式(16)可计算出属于类别 k 的某结果命中率，其中，$g_r(\tau)$ 为 RSU 附近交通状况。

$$k_d(\tau)=\sum_{\tau_d\le\tau}E_d(\tau)i_{r,d}(\tau)g_r(\tau) \tag{16}$$

同时，根据前文引入的 Hawkes 模型，可预测未来时隙的车辆请求响应情况。通过考虑未来收益可优化当前缓存决策，未来命中率由式(17)可得。

$$\sum_{n=\tau+1}^{\tau+\rho}\hat{k}_d(n)=\sum_{n=\tau+1}^{\tau+\rho}\sum_{r\in R}\hat{E}_d(n)\hat{i}_{r,d}(n)\hat{g}_r(n) \tag{17}$$

其中，ρ 用来控制未来的时间长度。当 $c_{S_r(\tau)}\ne 0$ 时，当前命中率和未来命中率之间的权衡计算为

$$K(\tau)=k_d(\tau)+\sum_{n=\tau+1}^{\tau+\rho}\omega(n)k_d(n) \tag{18}$$

其中，$\omega(n)$ 为折扣因子，用于控制未来命中率的权重。

2.4. 问题描述

在每个时隙 τ 内，缓存和计算迁移决策极大地影响着时延和内容命中率。本文模型的目标是持续优化缓存和计算迁移，降低任务的时延和提高命中率，并在二者之间取得平衡。因此，本文构建了一个长期多目标优化问题，使系统长期总收益最大。考虑由于二者指标单位不同，此处做控制量级处理，以消除指标之间维度的影响。最后，根据上面描述的各类模型，优化目标公式可表示为式(19)，其中 ω_1 和 ω_2 是时延与命中率之间的权衡参数。

$$
\begin{aligned}
&\mathrm{P1}: \max_{x_j(\tau), c_{S_r(\tau)}} Z = \sum_{\tau=1}^{T} \left(-\omega_1 T^{total}(\tau) + \omega_2 K^{total}(\tau) \right) \\
&\text{s.t. } \mathrm{C1}: x_j(\tau) = 1 - H\left(c_{S_r(\tau)} = 0 \right) \\
&\quad \mathrm{C2}: \sum_{j=0}^{R+M+C} x_j(\tau) = 1 \\
&\quad \mathrm{C3}: T^{total}(\tau) \le h_{S_m(\tau)}, \tau < \infty \\
&\quad \mathrm{C4}: c_{S_r(\tau)} \in \{1, 2, \cdots, R\}
\end{aligned} \tag{19}
$$

其中，约束 C1 表示只有当缓存决策 $c_{S_r(\tau)}$ 不为 0 时才会分配计算节点。约束 C2 表示请求的每个服务单元只能在单个节点上执行。约束 C3 保证服务在有效时间内完成，即限定不允许超过的最大延迟。约束 C4 表示缓存决策的取值范围在 0 到 RSU 总数之间。

3. 基于强化学习的计算迁移与缓存优化

3.1. 马尔科夫决策问题

在“车–边–云”协同场景中，MDP 为建模和优化服务迁移决策提供了数学框架。通过将服务迁移问题形式化为一个序列决策问题，能够有效捕捉系统的动态性和不确定性，从而优化迁移策略。MDP 可由一个四元组 $\langle S, A, P, R \rangle$ 表示。 S 是状态集合； A 是动作集合； P 为时隙 τ 状态下的动作致时隙 $\tau+1$ 下一状态的概率； R 代表状态转移后的奖励[17]。

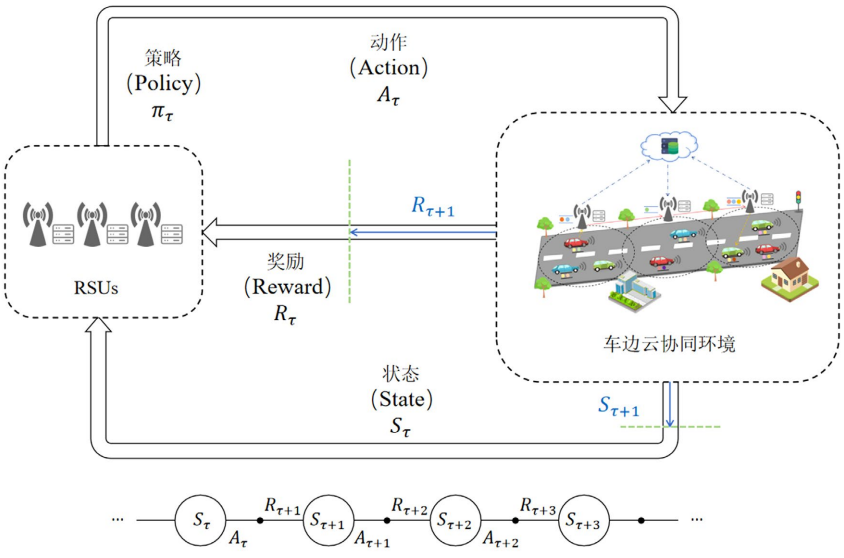

Figure 2. Service migration decision model based on Markov decision process
图 2. 基于马尔可夫决策过程的服务迁移决策模型

如图 2 所示，基于四元组构建 DRL 所需的关键因素。在每个时隙 τ 中，智能体对网络环境进行监控，并动态收集状态。设 S 为状态空间，各时隙 τ 的系统状态满足 $s_\tau \in S$ ，如式(20)所示。

$$
s_\tau = \left\{ S_r(\tau), E(\tau), I(\tau), g(\tau) \right\} \tag{20}
$$

每一个智能体从环境中观察到状态 s_τ 后，做出计算迁移和缓存决策，转化为下一个状态 $s_{\tau+1}$ 。动作空间收集所有决策 a_τ ，如式(21)所示。

$$a_\tau = \left\{ x_j(\tau), c_{S_r(\tau)} \right\} \tag{21}$$

奖励函数 r_τ 表示智能体在状态 s_τ 下执行服务迁移策略动作 a_τ 时获得的奖励。MDP 中以最大化系统的长期奖励为目标，即降低时延并提高内容命中率。因此，时延可被视为成本，为保持一致性，需以负值表示。所以如果满足约束条件 C1-C4，系统奖励设为式(22)。

$$r_\tau = -\omega_1 T^{total}(\tau) + \omega_2 K^{total}(\tau) \tag{22}$$

若不满足约束条件 C1-C4，则设为

$$r_\tau = -z \tag{23}$$

其中，z 是一个较大的数值，用来给予惩罚。

3.2. 算法设计

本文提出了一种结合 LSTM 和 PPO 的计算迁移与缓存优化算法。该方法是基于深度强化学习算法，在梯度策略算法的基础上结合了策略和价值的 Actor-Critic 算法。其原理是将策略参数化，表示为概率分布函数 $\pi_{\theta(a_\tau|s_\tau)}$，$\theta$ 为策略的参数。模型还包括价值网络 $V_{\phi(s_v)}$、重放缓存区以及环境。

针对传统 PPO 算法存在的高方差及裁剪过度问题，本文采用引入基线机制的广义优势估计(Generalized Advantage Estimation, GAE)方法，有效缓解了以上问题。通过引入超参数和基线函数，GAE 能在偏差和方差之间灵活权衡，适应不同的任务需求，从而提高策略梯度方法的稳定性。

此外，采用 LSTM 对传统全连接网络结构进行了改进。目的是解决一般递归神经网络在处理长时间依赖问题时的不足。这种网络架构能够有效地传递和表达长序列中的信息，避免忽视早期有用信息，从而提升动作网络和评价网络的性能。算法框架如图 3 所示。

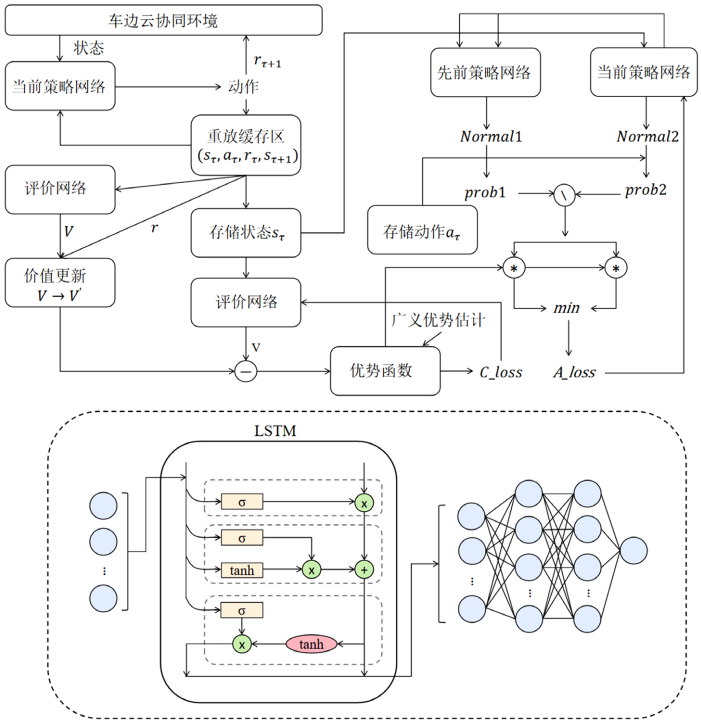

Figure 3. Algorithm framework
图 3. 算法框架图

在每个训练周期中，当前策略网络与环境进行交互，根据策略选择动作并得到奖励。随后，这些信息以元组形式$\left(s_\tau,a_\tau,r_\tau,s_{\tau+1}\right)$存入样本缓冲区。通过引入广义优势估计(GAE)，结合时序差分误差(TD-error)与蒙特卡洛方法(MC)，用不同长度的加权平均值进行估计。引入超参数λ平衡 TD 方法所引入的偏差与 MC 方法带来的方差。γ用于控制未来回报的重要性。时序差分误差的具体表达见式(24)。

$$\vartheta_\tau^V=r_\tau+\gamma V\left(s_{\tau+1}\right)-V\left(s_\tau\right) \tag{24}$$

由于策略梯度算法面临步长难以确定的问题，若步长不合适，参数更新后策略的回报函数值可能降低，导致算法无法收敛。因此在 PPO 中，策略更新的目标函数是通过重要性采样(Importance Sampling)来估计的。通过式(25)，可以将新旧策略网络的动作输出概率变化范围$r_\tau\left(\theta\right)$限制在一定区域内。

$$r_\tau\left(\theta\right)=\frac{\pi_{\theta(a_\tau|s_\tau)}}{\pi_{\theta_{old}}(a_\tau|s_\tau)} \tag{25}$$

为了限制更新幅度，PPO 对$r_\tau\left(\theta\right)$进行裁剪，将其限制在$\left[1-\varepsilon,1+\varepsilon\right]$内。该机制允许策略在一定范围内自由探索，同时避免更新幅度过大导致训练不稳定。最终可得到目标函数，如式(26)所示。

$$L_{clip}\left(\theta\right)=\hat{E}_\tau\min\left\{r_\tau\left(\theta\right)\hat{A}_\tau,clip\left(r_\tau\left(\theta\right),1-\varepsilon,1+\varepsilon\right)\hat{A}_\tau\right\} \tag{26}$$

其中，$\hat{E}_\tau$为时隙τ的经验期望。$\hat{A}_\tau$为优势函数，用于评估某个动作的好坏，如式(27)所示。$clip$为一个约束函数，用于对优势比率进行截断，确保其在区间$\left[1-\varepsilon,1+\varepsilon\right]$内。$\varepsilon$用来控制更新幅度。

$$\hat{A}_{\tau GAE(\gamma,\lambda)}=\sum_{i=\tau}\left(\gamma\lambda\right)^\tau\vartheta_\tau^V \tag{27}$$

4. 仿真实验与结果分析

4.1. 仿真实验

在仿真中，我们考虑并部署了一个 600 米范围内 $R=3$ 的多车道环境。假设两个相邻 RSU 之间的距离相同，覆盖范围相同。车辆会以规定范围内产生的随机速度匀速行驶在道路上。为了解决不同服务之间计算难度的差异，引入了 0.8 到 1 的难度系数作为难度分配。其他参数如表 1 所示。

Table 1. Parameter setting
表 1. 参数设置

参数	值
任务大小	[2, 10] MB
任务结果大小	[0.3, 3] MB
车辆速度	[36, 120] km/h
预测折扣因子	0.9
算法折扣因子	0.99
Hawkes 激励因子	0.9
Hawkes 延迟因子	0.1

4.2. 结果分析

首先，本文通过收益评估融合性能，如图 4 所示，根据不同奖励系数的收益值，可以观察到其总体效果在参数设置下的变化。在速度范围 36~120 km/h 内，根据式(22)的收益计算公式，可以计算得到每个

回合中所有时隙的 r_t 之和。随着迭代轮数的增加，在 120 轮左右时，已逐渐趋于收敛。PPO 通过引入剪切损失函数来控制每次策略更新的幅度，提高训练稳定。同时，经 LSTM 改进后的 PPO 策略网络能更好地理解历史信息，因而进行更优的策略决策，得到更高的收益值。

其中，奖励系数控制着时延和内容命中率的权重分配。当调整系数时，会对整体收益产生显著影响。随着 ω_2 的增大，ω_1 的减小，算法会更加侧重于内容命中率，对时延因素的重视下降，从而促命中率更快地向较大值收敛。同时，由于内容命中率所占较大比重，所以整体收益会上升。

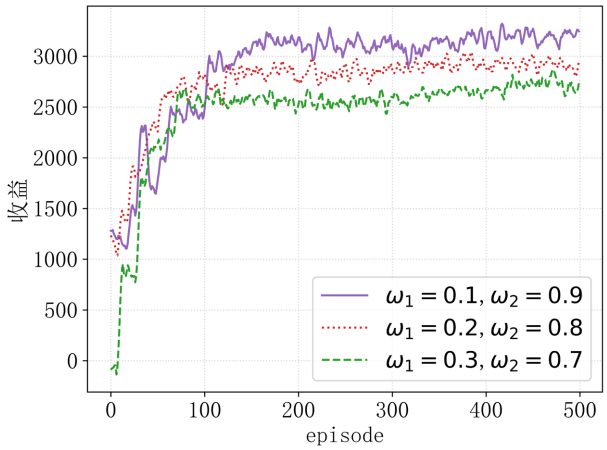

Figure 4. Reward for different tradeoff factors
图 4. 不同奖励系数下的收益

基于车边云协同场景对高并发和动态环境适应的要求，本文将着重考虑车辆请求响应率。其中，任务的有效时间直接影响着服务完成率和车辆请求响应率。任务时限的降低极大地提高了系统处理请求并进行决策的要求，若未能及时调度资源和计算迁移，则会无法及时响应请求。如图 5 所示，随着任务时限的降低，车辆请求响应率也快速下降。当任务时限减半后，响应率迅速下降了 22.6%，降至 54.2%。

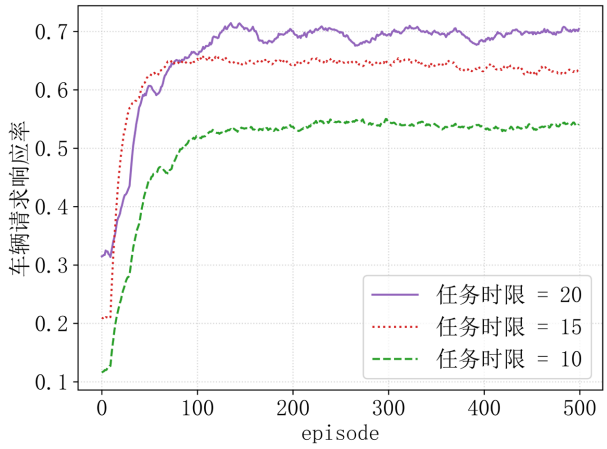

Figure 5. Vehicle request response rate under different deadlines
图 5. 不同任务时限下的车辆请求响应率

RSU 的缓存容量决定了其能存储的数据量。较大的缓存可以保存更多常用的数据，当车辆发送请求时，RSU 可以快速从缓存中提取所需数据，而无需向远程服务器请求。这种直接的数据访问显著提高了

请求的响应速度。如图 6 所示，随着缓存容量的上升，车辆请求响应率迅速增大。当边缘缓存容量为 4 MB 时，系统仍能响应约 69.1%的车辆请求。整体来看，当缓存容量增加 50%时，车辆请求响应率上升 24.6%。

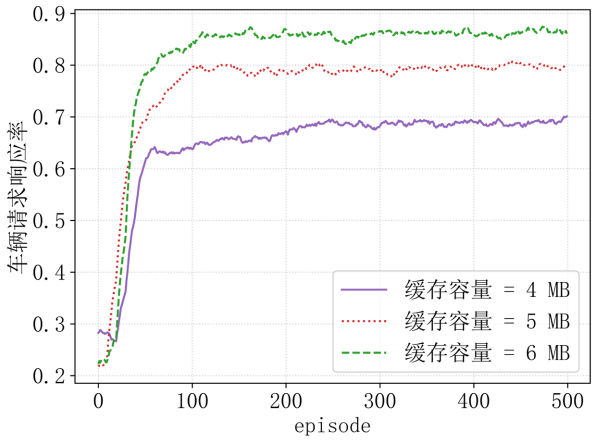

Figure 6. Vehicle request response rate under different cache capacities
图 6. 不同缓存容量下的车辆请求响应率

为了更好地评估提出算法的性能，观察不同缓存策略的效果，本文选取了几种算法进行比较。最少频繁使用(LFU)：当 RSU 的缓存容量满时，请求次数最少的内容将首先被替换。最近最少使用(LRU)：当 RSU 的缓存容量满时，最先替换未使用时间最长的内容。最优缓存算法：遍历所有可能的缓存节点，通过式(22)计算所有奖励值，并选择值最大的节点来进行缓存。

根据图 7 可观察到，车辆请求响应率最高的是最优缓存方法，因为该方法是通过遍历全部节点，以确保当前缓存策略是全局最优。然而，这可能导致时间和空间复杂度极高，特别是节点数量庞大时，计算成本和时延成本都会迅速增大，并不适合实际的场景。同样，作为经典的缓存替换算法，LFU 和 LRU 算法实现虽然较为简单，但是面对复杂场景不具备自适应能力，因此响应率相对较低。本文提出的算法弥补了以上算法的缺陷，通过收集更新不同类型的内容流行度，主动进行缓存迁移，更好地响应车辆请求，相较于 LFU 和 LRU 算法提高了 16.7%的效果。

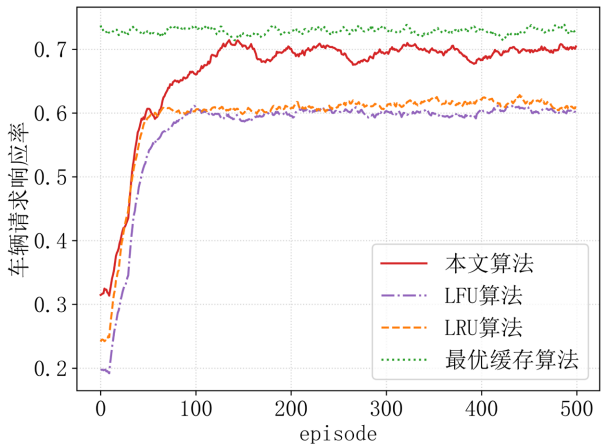

Figure 7. Vehicle request response rate under different algorithms
图 7. 不同算法的车辆请求响应率

5. 结束语

基于智慧交通迅速发展的背景下，本文提出了一种联合优化时延和缓存策略的“车–边–云”协同服务迁移方案。在该架构下，网络中的车载终端、RSU 和云服务器可以协同进行任务计算，进行动态服务迁移。通过使用基于 Hawkes 过程的方法，根据历史请求信息更新不同内容类型的流行度，选择合适的边缘计算节点和缓存内容。本文以降低时延和提高内容命中率为目标，形式化服务迁移问题。通过将该问题建模为 MDP，并引入深度强化学习求解最优问题，提出了一种结合 LSTM 和 PPO 的服务迁移算法。仿真实验结果表明，提出的策略优化方法比其他策略有更好的性能。未来的工作将致力于复杂交通场景下的服务迁移策略优化，结合通信环境、资源管理、任务要求等因素，综合提升服务的可靠性和效率。

参考文献

[1] Sun, Y., Hu, Y., Zhang, H., Chen, H. and Wang, F. (2023) A Parallel Emission Regulatory Framework for Intelligent Transportation Systems and Smart Cities. *IEEE Transactions on Intelligent Vehicles*, **8**, 1017-1020. https://doi.org/10.1109/tiv.2023.3246045

[2] Zhang, Y., Zhao, Y. and Zhou, Y. (2022) User-Centered Cooperative-Communication Strategy for 5G Internet of Vehicles. *IEEE Internet of Things Journal*, **9**, 13486-13497. https://doi.org/10.1109/jiot.2022.3143124

[3] Anwar, M.R., Wang, S., Akram, M.F., Raza, S. and Mahmood, S. (2022) 5G-Enabled MEC: A Distributed Traffic Steering for Seamless Service Migration of Internet of Vehicles. *IEEE Internet of Things Journal*, **9**, 648-661. https://doi.org/10.1109/jiot.2021.3084912

[4] Tang, H., Wu, H., Zhao, Y. and Li, R. (2022) Joint Computation Offloading and Resource Allocation under Task-Overflowed Situations in Mobile-Edge Computing. *IEEE Transactions on Network and Service Management*, **19**, 1539-1553. https://doi.org/10.1109/tnsm.2021.3135389

[5] Park, C. and Lee, J. (2021) Mobile Edge Computing-Enabled Heterogeneous Networks. *IEEE Transactions on Wireless Communications*, **20**, 1038-1051. https://doi.org/10.1109/twc.2020.3030178

[6] Peng, Y., Tang, X., Zhou, Y., Li, J., Qi, Y., Liu, L., *et al*. (2024) Computing and Communication Cost-Aware Service Migration Enabled by Transfer Reinforcement Learning for Dynamic Vehicular Edge Computing Networks. *IEEE Transactions on Mobile Computing*, **23**, 257-269. https://doi.org/10.1109/tmc.2022.3225239

[7] Wang, X., Ning, Z. and Guo, S. (2021) Multi-Agent Imitation Learning for Pervasive Edge Computing: A Decentralized Computation Offloading Algorithm. *IEEE Transactions on Parallel and Distributed Systems*, **32**, 411-425. https://doi.org/10.1109/tpds.2020.3023936

[8] Chen, X., Wu, C., Chen, T., Liu, Z., Zhang, H., Bennis, M., *et al*. (2022) Information Freshness-Aware Task Offloading in Air-Ground Integrated Edge Computing Systems. *IEEE Journal on Selected Areas in Communications*, **40**, 243-258. https://doi.org/10.1109/jsac.2021.3126075

[9] Lan, D., Taherkordi, A., Eliassen, F., Liu, L. and Yuan, X. (2023) Joint Optimization of Service Migration and Resource Management for Vehicular Edge Computing. 2023 19*th International Conference on Distributed Computing in Smart Systems and the Internet of Things* (*DCOSS-IoT*), Pafos, 19-21 June 2023, 155-160. https://doi.org/10.1109/dcoss-iot58021.2023.00036

[10] Chen, W., Chen, Y. and Liu, J. (2023) Service Migration for Mobile Edge Computing Based on Partially Observable Markov Decision Processes. *Computers and Electrical Engineering*, **106**, Article ID: 108552. https://doi.org/10.1016/j.compeleceng.2022.108552

[11] Zhang, D., Wang, W., Zhang, J., Zhang, T., Du, J. and Yang, C. (2023) Novel Edge Caching Approach Based on Multi-Agent Deep Reinforcement Learning for Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, **24**, 8324-8338. https://doi.org/10.1109/tits.2023.3264553

[12] Wu, H., Jin, J., Ma, H. and Xing, L. (2024) Federation-Based Deep Reinforcement Learning Cooperative Cache in Vehicular Edge Networks. *IEEE Internet of Things Journal*, **11**, 2550-2560. https://doi.org/10.1109/jiot.2023.3292374

[13] Zhou, H., Jiang, K., He, S., Min, G. and Wu, J. (2023) Distributed Deep Multi-Agent Reinforcement Learning for Cooperative Edge Caching in Internet-of-Vehicles. *IEEE Transactions on Wireless Communications*, **22**, 9595-9609. https://doi.org/10.1109/twc.2023.3272348

[14] Huang, J., Wan, J., Lv, B., Ye, Q. and Chen, Y. (2023) Joint Computation Offloading and Resource Allocation for Edge-Cloud Collaboration in Internet of Vehicles via Deep Reinforcement Learning. *IEEE Systems Journal*, **17**, 2500-2511. https://doi.org/10.1109/jsyst.2023.3249217

[15] 李媛媛, 付晓东, 刘骊, 等. 利用 Hawkes 过程模型的移动边缘计算服务质量预测[J]. 小型微型计算机系统, 2023, 44(7): 1571-1577.

[16] Ram, A. and Srijith, P.K. (2018) Accelerating Hawkes Process for Event History Data: Application to Social Networks and Recommendation Systems. 2018 10*th International Conference on Communication Systems & Networks* (*COMSNETS*), Bengaluru, 3-7 January 2018, 396-399. https://doi.org/10.1109/comsnets.2018.8328226

[17] Khamari, S., Rachedi, A., Ahmed, T. and Mosbah, M. (2023) Adaptive Deep Reinforcement Learning Approach for Service Migration in MEC-Enabled Vehicular Networks. 2023 *IEEE Symposium on Computers and Communications* (*ISCC*), Gammarth, 9-12 July 2023, 1075-1079. https://doi.org/10.1109/iscc58397.2023.10218103