

一种车载服务的快速深度Q学习网络边云迁移策略

彭 军^① 王成龙^① 蒋 富^{*①} 顾 欣^② 牟玥玥^① 刘伟荣^①

^①(中南大学计算机学院 长沙 410083)

^②(中南大学自动化学院 长沙 410083)

摘 要: 智能网联交通系统中车载用户的高速移动, 不可避免地造成了数据在边缘服务器之间频繁迁移, 产生了额外的通信回传时延, 对边缘服务器的实时计算服务带来了巨大的挑战。为此, 该文提出一种基于车辆运动轨迹的快速深度Q学习网络(DQN-TP)边云迁移策略, 实现数据迁移的离线评估和在线决策。车载决策神经网络实时获取接入的边缘服务器网络状态和通信回传时延, 根据车辆的运动轨迹进行虚拟机或任务迁移的决策, 同时将实时的决策信息和获取的边缘服务器网络状态信息发送到云端的经验回放池中; 评估神经网络在云端读取经验回放池中的相关信息进行网络参数的优化训练, 定时更新车载决策神经网络的权值, 实现在线决策的优化。最后仿真验证了所提算法与虚拟机迁移算法和任务迁移算法相比能有效地降低时延。

关键词: 智能网联交通系统; 虚拟机迁移; 强化学习; 深度Q学习网络

中图分类号: TN929.52

文献标识码: A

文章编号: 1009-5896(2020)01-0058-07

DOI: [10.11999/JEIT190612](https://doi.org/10.11999/JEIT190612)

A Fast Deep Q-learning Network Edge Cloud Migration Strategy for Vehicular Service

PENG Jun^① WANG Chenglong^① JIANG Fu^① GU Xin^②

MU Yueyue^① LIU Weirong^①

^①(School of Computer Science and Engineering, Central South University, Changsha 410083, China)

^②(School of Automation, Central South University, Changsha 410083, China)

Abstract: The high-speed movement of vehicles inevitably leads to frequent data migration between edge servers and increases communication delay, which brings great challenges to the real-time computing service of edge servers. To solve this problem, a real-time reinforcement learning method based on Deep Q-learning Networks according to vehicle motion Trajectory Process (DQN-TP) is proposed. The proposed algorithm separates the decision-making process from the training process by using two neural networks. The decision neural network obtains the network state in real time according to the vehicle's movement track and chooses the migration method in the virtual machine migration and task migration. At the same time, the decision neural network uploads the decision records to the memory replay pool in the cloud. The evaluation neural network in the cloud trains with the records in the memory replay pool and periodically updates the parameters to the on-board decision neural network. In this way, training and decision-making can be carried out simultaneously. At last, a large number of simulation experiments show that the proposed algorithm can effectively reduce the latency compared with the existing methods of task migration and virtual machine migration.

Key words: Intelligent transportation system; Virtual machine migration; Reinforcement learning; Deep Q-learning Networks(DQN)

收稿日期: 2019-08-12; 改回日期: 2019-11-04; 网络出版: 2019-11-12

*通信作者: 蒋富 jiangfu0912@csu.edu.cn

基金项目: 国家自然科学基金(61873353, 61672539)

Foundation Items: The National Natural Science Foundation of China(61873353, 61672539)

1 引言

智能网联交通系统是一种集成了人工智能、传感技术、网络技术、自动控制技术及物联网等多项技术的交通信息智能决策调度系统^[1]。智能网联交通系统通过车辆、车路、基础设施、周边交通环境等多方信息的采集与交互，增强了车辆自身感知能力，为各类数据的综合分析与时处理创造了可能^[2]。同时它融合了云计算与移动边缘计算的技术优势，为行驶的车辆提供强大的感知计算能力，为自动驾驶与无人驾驶的落地提供有力支撑^[3]。但一直以来，交通系统环境的复杂性与车载用户的移动性成为制约其发展的一大瓶颈^[4]。车辆在边缘服务器之间的移动会不可避免地造成数据频繁迁移，从而产生额外的数据迁移时延，这给边缘服务器的实时计算服务带来了巨大的挑战。

近年来，为处理上述问题，国内外学者分别基于任务迁移和虚拟机迁移两种方式提出了不同解决方案。任务迁移是一种通过后向回传链路使任务在车载用户移动过程中不中断处理的方式，而虚拟机迁移是直接将处理任务的虚拟机迁移来保证任务的处理。在任务迁移方面，文献^[5]设计了一种基于遗传算法的任务迁移方法，一方面有效降低移动设备的能耗，另一方面保证了任务能及时处理，降低了任务迁移过程中的延时。文献^[6]综合考虑了用户移动设备的电池性能，提出了一种基于深度强化学习的计算卸载方法用于减少用户的延时和移动设备的能量消耗。文献^[7]提出了一种将任务从移动设备迁移到远程服务器的任务迁移决策算法，通过将任务迁移到多个服务器联合处理降低任务的处理时间。但基于任务迁移的方式随着车载用户远离，通信的回传链路会越来越长产生额外的延时。为克服任务迁移的缺点，文献^[8]给出了一种基于已知路径的虚拟机迁移算法。使用该算法可以提前将虚拟机迁移到预定位置，减少大量的传输延时。文献^[9]分析了由网络链路拥塞引起的用户服务质量下降。并设计了一种虚拟机迁移策略以避免拥塞。但基于虚拟机迁移的方法迁移虚拟机会占用大量的带宽，当任务负载很大的时候带来额外的排队延时。

虽然这些方案都能在一定程度上降低时延，而且这些方案大部分是离线的方法，不能根据真实环境实时地做出决策。因此，本文从保证边缘服务器的实时计算服务出发，提出了一种基于车辆运动轨迹的快速深度Q学习网络(Deep Q-learning Networks, DQN)边云迁移策略，将一个连续的车辆运行轨迹离散化为一系列的决策周期。在每个决策周期内，

根据实时的边缘服务器网络状态和通信回传时延在任务迁移或虚拟机迁移这两种方法中作出决策，本文更加具体的贡献可以概括如下：

(1) 本文提出了一种基于车辆运动轨迹的快速深度Q学习网络(Deep Q-learning Networks for Trajectory Process, DQN-TP)强化学习算法。提出的算法在每个决策周期根据当前边缘服务器网络状态信息，从虚拟机迁移和任务迁移中选择一个方法执行迁移决策。并在仿真中与任务迁移算法和虚拟机迁移算法比较，验证了所提算法的优越性。

(2) 本文将决策和评估神经网络分离，车载决策神经网络根据用户实时获取的边缘服务器网络状态参数和时延进行决策，同时把决策记录信息发送到云端；云端评估神经网络根据车载用户上传的决策信息进行训练，并将训练的网络参数更新到车载决策神经网络，实现评估和决策过程同时进行，增强算法实时性。

本文其余的部分组织如下，第2节介绍了本文的系统框架和模型，第3节分析了本文所提DQN-TP算法，第4节是本文的仿真结果分析，第5节给出了本文的结论。

2 系统框架和模型建立

如图1所示，车载用户通过接入路旁单元(Road Side Unit, RSU)卸载任务，同时随着车载用户离开RSU的覆盖范围，车载用户切换边缘服务器。每个RSU都配备一个小型的边缘服务器用于提供服务，边缘服务器之间可以通过后向回传网络(back-haul network)进行通信，每个边缘服务器同时也与中央云服务器相连支持远程调度。当车载用户卸载任务到边缘服务器时，边缘服务器会根据用户卸载的任务生成虚拟机去处理和执行任务。每个车载用户都配备一个车载决策神经网络用于实时决策，同时会将决策记录作为经验上传到云端，云端通过经验训练神经网络，每隔一段时间将训练得更加完备的神经网络反馈到车载神经网络中。

本文所用到的变量如表1所示，一个车载用户在一段时间内的行车轨迹可以被离散化为一系列的决策周期，且每个决策周期的时间长度为 σ 。在决策周期 t 车载用户的位置可以由二元组 ν_t 表示。假定区域内有 i 个边缘服务器并用集合 M 表示，与车载用户相似，使用二元组 μ_m 表示第 m 个($m \in M$)边缘服务器所在的位置。 r_m 代表边缘服务器 m 的覆盖范围。

由车载用户的位置和边缘服务器的位置，可以得到二者间的距离公式为

$$s_{(t,m)} = \|\nu_t - \mu_m\| \quad (1)$$

对于给定周期 t ，车载用户以概率 α 产生任务请

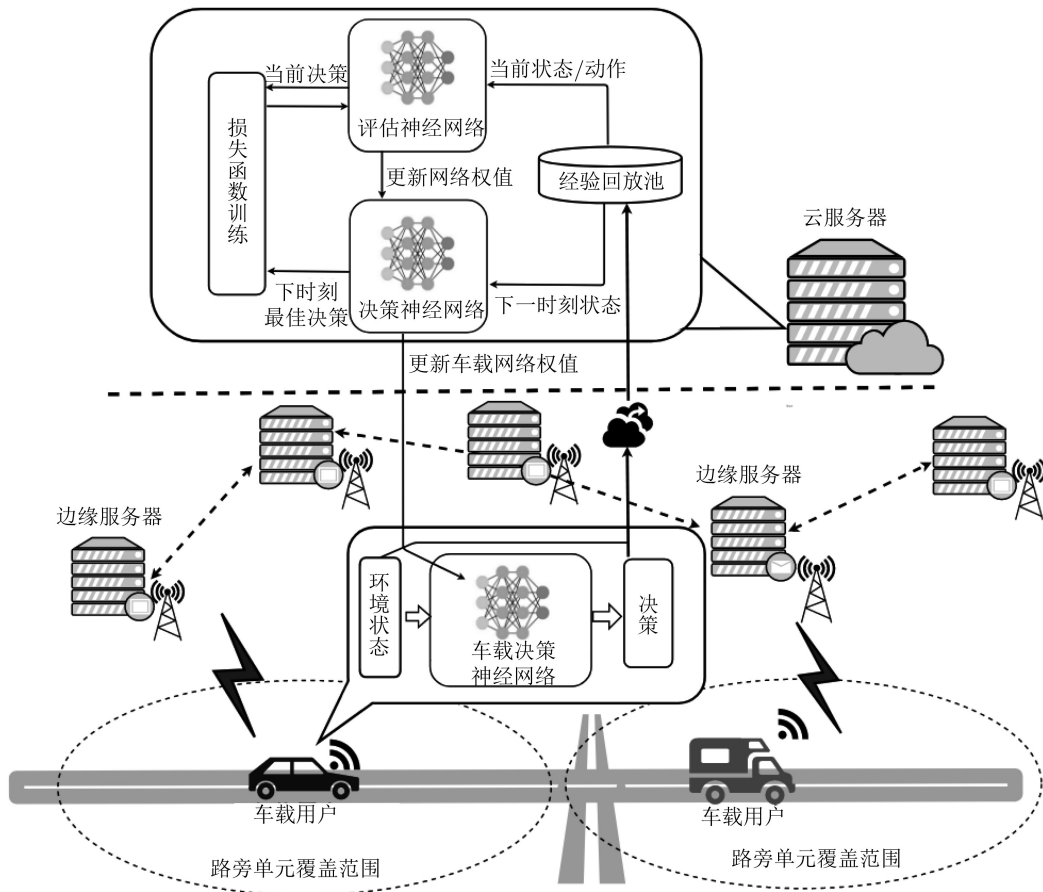


图1 智能网联交通系统结构示意图

表1 变量表

变量名	变量符号
决策周期长度	σ
决策周期	t
边缘服务器数量	i
车载用户位置	ν_i
边缘服务器位置	μ_m
路径损失参数	δ
路旁单元覆盖半径	r
任务大小	q_s
任务最大容忍时延	q_d
传输功率	P_s
时延	T
虚拟机所在位置	D

求 q 。对于每个任务都有两个关键参数，分别是任务的大小 q_s 和任务所能容忍最大延时 q_d 。每一个任务产生后会加入处理等待队列，处理原则遵循先入先出(FIFO)原则。处理方式可以被分为任务迁移处理和虚拟机迁移处理两种。

任务迁移^[10]的处理流程如下：当车载用户离开当前边缘服务器覆盖范围时，车载用户会在当前区

域所有可接入的边缘服务器中选择一个服务器建立连接，被选中的服务器接到车载用户提交的新请求后会向中央云服务器查询负责处理该用户服务的虚拟机的源边缘服务器位置，并把车载用户请求通过后向回传网络转交给源边缘服务器。任务迁移只做用户任务处理请求的转交，因此方法简单迁移量小，但随着车载用户远离源边缘服务器，转交任务的通信链路会变长，因而会带来额外的时延。

结合微软的虚拟硬盘技术和文献[11]的无缝虚拟机传输方法，虚拟机迁移的过程可以概括如下：当车载用户离开当前边缘服务器覆盖范围时，先进行任务迁移，与此同时新接入的服务器向虚拟机所在的源服务器发送虚拟机迁移请求，源服务器接到迁移请求后开始向当前服务器迁移虚拟机。当虚拟机迁移完成时，源服务器向云服务器发送虚拟机已经发生迁移的事件，云服务器将车载用户所对应的虚拟机所在位置修改为虚拟机迁移的目标服务器所在位置，同时目标服务器切断与源服务器的连接终止任务迁移成为新的源服务器，之后的处理都将会在新的源服务器上进行。虚拟机迁移会将任务处理的缓存、环境变量和上下文等一并传输到靠近用户侧的边缘服务器处理，因此延时较低，但频繁的迁

移会带来巨大的迁移量，尤其在任务和用户数量众多的情况下，频繁迁移会造成额外的边缘服务器排队时延。

接下来本文从任务迁移和虚拟机迁移两方面分析两种方法各自的时延消耗。首先是任务迁移，先给出车载用户传输任务到边缘服务器的传输速率 R_t 公式为

$$R_t = W \cdot \log_2 \left(1 + \frac{s_{(t,m)}^{-\sigma} P_s}{N} \right) \quad (2)$$

W 代表车载用户传输任务的带宽， P_s 代表传输功率， δ 是一个和距离有关的路径损耗参数^[12]，由式(2)可以得到传输时延为

$$T_t = \frac{q_s}{R_t} \quad (3)$$

对于任务迁移来说，总时延 T_s 应该包含传输时延 T_t 、虚拟机寻址时延 T_a 、任务的排队时延 T_q 和任务处理时延 T_e ， T_s 可以表示为

$$T_s = T_t + T_a + T_e + T_q \quad (4)$$

其中，虚拟机寻址时延 T_a 和通信的链路长度相关，处理时延 T_e 与任务大小 q_s 以及服务器的处理能力 f 相关，可以表示为

$$T_e = \frac{q_s}{\rho f} \quad (5)$$

其中， ρ 是一个与芯片材质相关的常数。如果当前连接的服务器正好就是虚拟机所在服务器，相当于直接将任务送至虚拟机处理没有寻址时延，那么总的时延可以用直连时延 T_d 表示为

$$T_d = T_t + T_e + T_q \quad (6)$$

对于虚拟机迁移的总等待时延，在虚拟机迁移完成之前是做的任务迁移，因此总等待时延在迁移完成前等于任务迁移的总时延 T_s ，当迁移完成后总时延就与直连时延 T_d 相等。

定义一个用于表示车载用户虚拟机所在位置的矩阵 \mathbf{D} ，如果在决策周期 t ，虚拟机在边缘服务器 m ，那么矩阵的对应元素 $\mathbf{D}_{(t,m)} = 1$ ，否则为0。

3 DQN-TP算法

本节将详细介绍提出的DQN-TP算法。一段时间内的车载用户轨迹被分为若干周期，其整个轨迹将被建模为一个马尔可夫决策过程。用 X 表示车载移动用户的状态集， $X_t = (v_t, \mathbf{D}_{(t,vm)}) \in X$ 为车载移动用户在周期 t 的一个状态。定义动作集为 A ， $a_t \in A$ 。车载用户在第 t 个时间周期的动作 a_t 代表选择任务迁移还是虚拟机迁移， $a_t = 1$ 代表进行虚拟机迁移， $a_t = 0$ 代表进行任务迁移。

为了让回报在满足约束条件下为正，超出约束条件为负，且偏离约束条件越多回报值在对应正负区间内越大， $U(\cdot)$ 是回报函数

$$U(T) = k \cdot (q_d \cdot T)^3 + b \quad (7)$$

其中， k 和 b 是常数，其值取决于约束的容忍程度。当总时延 T 超过任务能忍受的最大时延 q_d 代表超时，回报为负值。如果能在 q_d 内完成，回报为正值。

车载用户位置的概率与虚拟机放置的概率是相互独立的，只取决于状态 X_t 中上一周期选择的动作 a_t ，转移概率 $P_{X_t \rightarrow X_{t+1}}$ 可以被描述为

$$P_{X_t \rightarrow X_{t+1}} = P(\mathbf{D}_{(t+1,vm)} | \mathbf{D}_{(t,vm)}; \pi) \cdot P(v_{t+1} | v_t; \pi) \quad (8)$$

此外，状态值函数 $V_\pi(X)$ 和状态动作函数 $Q_\pi(X, a)$ 可以被描述为

$$V_\pi(X) = E_\pi \left[\sum_{t=0}^p \gamma^{t-1} \cdot U_t | X = X_0 \right] \quad (9)$$

$$Q_\pi(X, a) = E_\pi \left[\sum_{t=0}^p \gamma^{t-1} \cdot U_t | X = X_0, a = a_0 \right] \quad (10)$$

这里的 π 是车载移动用户当前选择的策略。 γ 表示折扣率。根据状态-动作值函数和贝尔曼公式的定义，可以获得式(11)

$$Q_\pi(X_t, a_t) = U_t + \gamma \sum P_{X_t \rightarrow X_{t+1}} \max_{a_{t+1}} Q_\pi(X_{t+1}, a_{t+1}) \quad (11)$$

接下来将建立两个具有相同结构的神经网络，评估神经网络和车载决策神经网络。评估神经网络放置到云端用于利用经验回放池中最新环境信息进行训练；车载决策神经网络用于车载用户实时决策，并将决策记录存入经验回放池中。

首先介绍车载用户的实时决策过程，决策神经网络参数来源于云端的评估神经网络的网络参数 θ ，并使用 ϵ -贪婪算法^[13] 进行动作的选择。它会以 ϵ 的概率随机选择一个动作去探索环境，也会以 $1-\epsilon$ 的概率用贪心思想选择当前状态下最好的动作。同时决策神经网络会将执行动作 a_t 后获得 U_t 和 X_{t+1} 作为一条记忆组成4元组 (X_t, a_t, U_t, X_{t+1}) 上传到云服务器的经验回放池中。

然后介绍云端的训练过程。评估神经网络的状态-动作值函数表示为 $Q_\pi(X, a; \theta)$ ，决策神经网络的状态-动作值函数表示为 $Q_\pi(X, a; \theta^-)$ ，参数 θ 和 θ^- 分别代表各自神经网络的网络参数。

训练开始时，从经验回放池中随机抽取一个大小为 n 的 mini-batch 训练数据，以当前时刻的状态 X_t 和动作 a_t 作为评估神经网络输入，下一个时刻的

状态 X_{t+1} 作为评估神经网络输入,根据式(11)进行训练。

由式(11)可知,下一时刻最大的 $Q_\pi(X_{t+1}, a_{t+1})$ 与 U_t 的和应该等于 $Q_\pi(X_t, a_t)$,因此损失函数 L 可以定义为二者的差值,具体如式(12)所示

$$L = \mathbb{E}_{(X_t, a_t, U_t, X_{t+1}) \in M} \cdot \left[\left(U_t + \gamma \max_a Q_\pi(X_{t+1}, a_{t+1}; \theta^-) - Q_\pi(X_t, a_t; \theta) \right)^2 \right] \quad (12)$$

对式(12)的损失函数求导可得损失函数的梯度公式(13),根据链式求导法则可知,损失函数的梯度变化是由动作-状态值函数的梯度变化造成的,而动作-状态值函数的梯度变化直接由网络参数 θ 决定,因此可以直接调节网络参数实现动作-状态值函数的拟合。

$$\nabla_\theta L = \mathbb{E}_{(X_t, a_t, U_t, X_{t+1}) \in M} \cdot \left[\left(U_t + \gamma \max_a Q_\pi(X_{t+1}, a_{t+1}; \theta^-) - Q_\pi(X_t, a_t; \theta) \right) \cdot \nabla Q_\pi(X_t, a_t; \theta) \right] \quad (13)$$

综上所述,根据损失函数的梯度采用梯度下降法更新评估神经网络的参数 θ ,再由更新的参数 θ 拟合新的动作-状态值函数 $Q_\pi(X, a; \theta)$ 得到式(14),并利用式(14)更新动作-状态值函数 $Q_\pi(X, a; \theta)$ 的值,其中 η 代表学习率。

$$Q_\pi^{j+1}(X_t, a_t; \theta) = Q_\pi^j(X_t, a_t; \theta) + \eta \cdot \left[U_t + \gamma \max_a Q_\pi^j(X_{t+1}, a_{t+1}; \theta^-) - Q_\pi^j(X_t, a_t; \theta) \right] \quad (14)$$

最后将整个训练过程重复,每隔 c 个训练周期,将云端的评估神经网络的参数 θ 更新到车载决策神经网络,指导车载神经网络更好的决策。

综合上述,表2中给出了详细的算法流程。提

出的算法将训练过程和决策过程分离,一方面把车载神经网络作为决策者与环境互动产生数据并将决策记录上传到云端经验回放池,另一方面云端利用车载端获得的最新数据对神经网络进行训练,并将训练好的参数更新到车载神经网络,让算法能做到实时决策,并且能根据实时路况更新决策,让决策质量越来越好。通过这种将方法,可以将训练过程和决策过程同时进行,并且在云端使用两个结构相同的神经网络可以在更新评估神经网络参数 θ 的同时不改变输出的稳定性。此外,由车载用户上传的决策经验可以为训练评估神经网络提供实时数据,随机抽样的mini-batch能够打破数据之间的时间连续性,从而进一步提升决策质量。

4 实验仿真

本文的仿真环境是在Window操作系统下的python环境进行的,硬件配置为i7-8750H+GTX1060+16G,在表3中给出了具体的变量参数配置信息。本文使用了SNIA云服务器记录轨迹数据集^[14],该数据集记录了真实环境下服务器的读写延时、真实任务的大小和任务对应的处理时间。初始的车载用户的路径轨迹和边缘服务器的位置信息在仿真环境中随机生成。

本节将从以下方面进行仿真验证,首先验证所提算法的收敛性,然后将提出算法与任务迁移算法(Only Service Transmission algorithm, ST-Only)和虚拟机迁移算法(Only Virtual Machine Migration algorithm, VMM-Only)在迁移流量和迁移时延的性能上进行比较,在下面给出被比较算法的定义。

任务迁移算法ST-Only:当车载用户离开边缘服务器覆盖范围时,只通过后向回传链路找到执行任务的虚拟机进行任务迁移,不进行虚拟机迁移。

虚拟机迁移算法VMM-Only:每当车载用户离

表2 DQN-TP算法

算法1: DQN-TP算法

- (1) Repeat:
- (2) 车载用户上传车载决策神经网络的经验 (X_t, a_t, U_t, X_{t+1}) 到经验回放池;
- (3) While $t \neq$ 最后一个周期do
- (4) 从经验回放池中随机抽取 n 个经验作为一个mini-batch;
- (5) 将 X_t, a_t 作为评估神经网络的输入获得 $Q_\pi(X_t, a_t; \theta)$,将 X_{t+1} 作为决策神经网络的输入获得 $Q_\pi(X_{t+1}, a_{t+1}; \theta^-)$;
- (6) 根据式(13)、式(14)训练神经网络;
- (7) End While
- (8) 每训练 c 次将云端的神经网络参数更新给车载神经网络 $\theta \rightarrow \theta^-$;
- (9) 车载用户使用 ϵ -贪婪算法选择动作-状态值函数最高的动作作为车载用户动作执行;
- (10) End

开边缘服务器覆盖范围就将虚拟机迁移到新接入的边缘服务器。

首先为了验证所提算法的收敛性，在图2中给出了损失函数随着训练周期变化的函数关系。从图2中可以看到，由于一开始的网络参数是随机初始化的，不能很好拟合真实的动作-状态值函数，所以损失很大，但随着训练周期的增加，损失函数逐渐减小并收敛，大约在第1200周期时，损失函数的值小于1。这说明随着训练的进行，所提算法通

表 3 仿真参数设定

参数名	参数符号	参数值
决策周期	σ	10^{-3} s
边缘服务器数量	i	10
路径损失参数	δ	1.5
带宽	W	4 MHz
路旁单元覆盖半径	r	500 m
效用函数参数	k	1.3
效用函数参数	b	0.1
记忆回放池最大存储数	o	3000
Mini-batch大小	n	500
参数更新间隔步长	c	80
神经网络层数	无	4
神经元总数	无	100

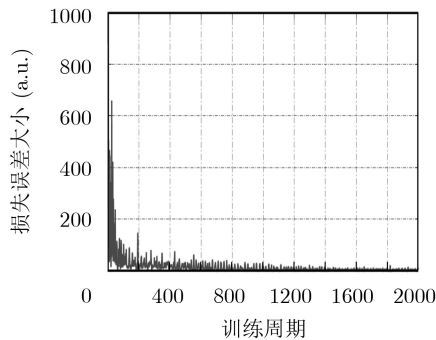
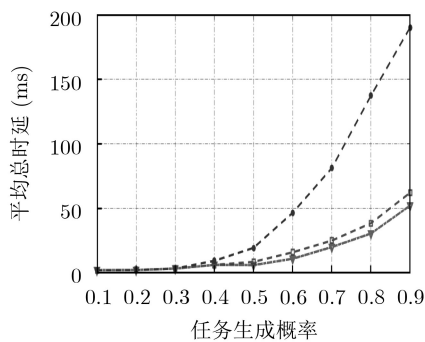


图 2 DQN-TP的收敛性

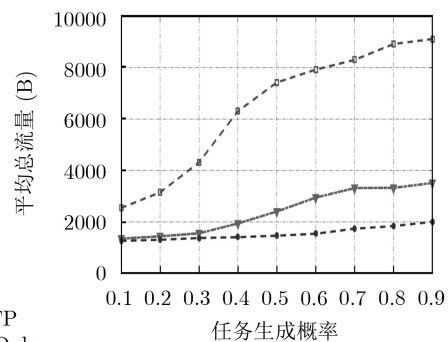


(a) 平均总时延与任务生成概率的关系

过损失函数逐步优化神经网络参数，逼近真实的动作-状态值函数。

3种算法的性能随着任务产生的概率变化趋势在图3中给出，任务产生概率的大小代表了任务数量的多少。3种算法的总时延比较如图3(a)所示，没做虚拟机迁移的ST-Only算法随着任务产生概率的增加延时会越来越高，其原因一方面是由于任务负载变大不可避免地造成算法性能下降，另一方面随着用户的移动，车载用户距离源服务器越来越远导致回传链路的长度越来越长，导致延时大幅地增加，而执行了虚拟机迁移的VMM-Only算法和DQN-TP算法通过将虚拟机迁移到邻近用户侧的边缘服务器处理就能很好地降低通信回传时延。可以看到VMM-Only算法和DQN-TP算法不管是在低任务负载的情况下还是在高任务负载的情况下都有较低的延时。对于VMM-Only算法来说，并不是每次虚拟机迁移都是有必要的，在任务负载较高时，每次都迁移虚拟机会带来大量的排队延时，因此随着任务负载的变大，算法性能退化严重。而本文提出的DQN-TP算法本质上是一种方法的选择，它会通过学习的方式判断什么时候执行任务迁移或虚拟机迁移并只在必要的时候才会迁移虚拟机，因此DQN-TP算法能获得更低的延时。

图3(b)展示的是3种算法在迁移量方面的性能比较。ST-Only算法由于不用将相对较大虚拟机迁移，因此流量的花费相比其他算法较小。VMM-Only算法相对于只做任务迁移ST-Only确实会有更多的迁移量开销，但这些开销对于降低时延提升服务质量是必要的。但相比起同样迁移了虚拟机的VMM-Only算法，DQN-TP能减少大量的迁移量。这表明所提出的DQN-TP算法能在保证服务质量的前提下尽可能地减少的迁移虚拟机，只在必要时迁移虚拟机，减少后向回传网络的流量负担。



(b) 迁移量与任务生成概率的关系

图 3 DQN-TP算法与其他迁移算法性能分析

5 结论

本文所提DQN-TP算法对现有方案从保证算法实时性和降低时延两方面进行改进。在算法实时性上, 本文将DQN网络中的决策神经网络和评估神经网络分离, 根据车辆的运动轨迹进行虚拟机或任务迁移的决策, 评估神经网络在云端读取经验回放池中的相关信息进行网络参数的优化训练, 定时更新车载决策神经网络的权值, 实现在线决策的优化; 在降低时延上, 本文综合虚拟机迁移和任务迁移两种方法, 根据用户给定环境信息进行智能决策, 只在必要时迁移虚拟机, 降低了网络时延和网络资源消耗。最后经过仿真验证, 与只进行任务迁移的ST-Only算法和只进行虚拟机迁移的算法VMM-Only相比都能获得最低的延时和相对较少的迁移量, 并在总体性能上优于其他算法。

参考文献

- [1] ZHU Li, YU F R, WANG Yige, *et al.* Big data analytics in intelligent transportation systems: A survey[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 20(1): 383–398. doi: [10.1109/TITS.2018.2815678](https://doi.org/10.1109/TITS.2018.2815678).
 - [2] D'OREY P M and FERREIRA M. ITS for sustainable mobility: A survey on applications and impact assessment tools[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2014, 15(2): 477–493. doi: [10.1109/TITS.2013.2287257](https://doi.org/10.1109/TITS.2013.2287257).
 - [3] 彭军, 马东, 刘凯阳, 等. 基于LTE D2D技术的车联网通信架构与数据分发策略研究[J]. *通信学报*, 2016, 37(7): 62–70. doi: [10.11959/j.issn.1000-436x.2016134](https://doi.org/10.11959/j.issn.1000-436x.2016134).
PENG Jun, MA Dong, LIU Kaiyang, *et al.* LTE D2D based vehicle networking communication architecture and data distributing strategy[J]. *Journal on Communications*, 2016, 37(7): 62–70. doi: [10.11959/j.issn.1000-436x.2016134](https://doi.org/10.11959/j.issn.1000-436x.2016134).
 - [4] GAO Kai, HAN Farong, DONG Pingping, *et al.* Connected vehicle as a mobile sensor for real time queue length at signalized intersections[J]. *Sensors*, 2019, 19(9): 2059. doi: [10.3390/s19092059](https://doi.org/10.3390/s19092059).
 - [5] KONG Yue, ZHANG Yikun, WANG Yichuan, *et al.* Energy saving strategy for task migration based on genetic algorithm[C]. 2018 International Conference on Networking and Network Applications, Xi'an, China, 2018: 330–336.
 - [6] CHEN Xianfu, ZHANG Honggang, WU C, *et al.* Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4005–4018. doi: [10.1109/JIOT.2018.2876279](https://doi.org/10.1109/JIOT.2018.2876279).
 - [7] SAHA S and HASAN M S. Effective task migration to reduce execution time in mobile cloud computing[C]. The 23rd International Conference on Automation and Computing, Huddersfield, UK, 2017: 1–5.
 - [8] GONÇALVES D, VELASQUEZ K, CURADO M, *et al.* Proactive virtual machine migration in fog environments[C]. 2018 IEEE Symposium on Computers and Communications, Natal, Brazil, 2018: 742–745.
 - [9] KIKUCHI J, WU C, JI Yusheng, *et al.* Mobile edge computing based VM migration for QoS improvement[C]. The 6th IEEE Global Conference on Consumer Electronics, Nagoya, Japan, 2017: 1–5.
 - [10] CHOWDHURY M, STEINBACH E, KELLERER W, *et al.* Context-Aware task migration for HART-Centric collaboration over FiWi based tactile internet infrastructures[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 29(6): 1231–1246. doi: [10.1109/TPDS.2018.2791406](https://doi.org/10.1109/TPDS.2018.2791406).
 - [11] LU Wei, MENG Xianyu, and GUO Guanfei. Fast service migration method based on virtual machine technology for MEC[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4344–4354. doi: [10.1109/JIOT.2018.2884519](https://doi.org/10.1109/JIOT.2018.2884519).
 - [12] WANG Yanting, SHENG Min, WANG Xijun, *et al.* Mobile-edge computing: Partial computation offloading using dynamic voltage scaling[J]. *IEEE Transactions on Communications*, 2016, 64(10): 4268–4282. doi: [10.1109/TCOMM.2016.2599530](https://doi.org/10.1109/TCOMM.2016.2599530).
 - [13] SUTTON R S and BARTO A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 1998: 25–42.
 - [14] SNIA trace data[EB/OL]. <http://iotta.snia.org/traces>, 2018.
- 彭军: 女, 1967年生, 教授, 研究方向为智能交通网络通信与安全控制、移动云服务、机器学习、智慧新能源管理。
王成龙: 男, 1996年生, 博士生, 研究方向为车联网、智能交通网络通信、机器学习。
蒋富: 男, 1983年生, 副教授, 研究方向为无线通信与移动网络、绿色认知无线传感器网络、协同通信、网络空间安全。
顾欣: 女, 1993年生, 博士生, 研究方向为无线网络优化、车联网通信技术。
牟玥玥: 女, 1996年生, 硕士生, 研究方向为图像处理、无线通信。
刘伟荣: 男, 1976年生, 教授, 研究方向为物联网、协同通信、CPS系统、云机器人情景感知。