

车辆网络多平台卸载智能资源分配算法

王汝言 梁颖杰* 崔亚平

(重庆邮电大学通信与信息工程学院 重庆 400065)

(重庆高校市级光通信与网络重点实验室 重庆 400065)

(泛在感知与互联重庆市重点实验室 重庆 400065)

摘要: 为了降低计算任务的时延和系统的成本, 移动边缘计算(MEC)被用于车辆网络, 以进一步改善车辆服务。该文在考虑计算资源的情况下对车辆网络时延问题进行研究, 提出一种多平台卸载智能资源分配算法, 对计算资源进行分配, 以提高下一代车辆网络的性能。该算法首先使用K临近(KNN)算法对计算任务的卸载平台(云计算、移动边缘计算、本地计算)进行选择, 然后在考虑非本地计算资源分配和系统复杂性的情况下, 使用强化学习方法, 以有效解决使用移动边缘计算的车辆网络中的资源分配问题。仿真结果表明, 与任务全部卸载到本地或MEC服务器等基准算法相比, 提出的多平台卸载智能资源分配算法实现了时延成本的显著降低, 平均可节省系统总成本达80%。

关键词: 车辆网络; 移动边缘计算; 资源分配; 强化学习

中图分类号: TN919.2

文献标识码: A

文章编号: 1009-5896(2020)01-0263-08

DOI: [10.11999/JEIT190074](https://doi.org/10.11999/JEIT190074)

Intelligent Resource Allocation Algorithm for Multi-platform Offloading in Vehicular Networks

WANG Ruyan LIANG Yingjie CUI Yaping

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

(Chongqing Key Laboratory of Optical Communication and Networks, Chongqing 400065, China)

(Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing 400065, China)

Abstract: In order to reduce the delay of computing tasks and the total cost of the system, Mobile Edge Computing (MEC) technology is applied to vehicular networks to improve further the service quality. The delay problem of vehicular networks is studied with the consideration of computing resources. In order to improve the performance of the next generation vehicular networks, a multi-platform offloading intelligent resource allocation algorithm is proposed to allocate the computing resources. In the proposed algorithm, the K-Nearest Neighbor (KNN) algorithm is used to select the offloading platform (i.e., cloud computing, mobile edge computing, local computing) for computing tasks. For the computing resource allocation problem and system complexity in non-local computing, reinforcement learning is used to solve the optimization problem of resource allocation in vehicular networks using the mobile edge computing technology. Simulation results demonstrate that compared with the baseline algorithms (i.e., all tasks offload to the local or MEC server), the proposed multi-platform offloading intelligent resource allocation algorithm achieves a significant reduction in latency cost, and the average system cost can be saved by 80%.

Key words: Vehicular networks; Mobile Edge Computing (MEC); Resource allocation; Reinforcement learning

收稿日期: 2019-01-25; 改回日期: 2019-07-16; 网络出版: 2019-09-20

*通信作者: 梁颖杰 liangyj10111@163.com

基金项目: 国家自然科学基金(61801065, 61771082, 61871062), 重庆市高校创新团队建设计划(CXTDX201601020)

Foundation Items: The National Natural Science Foundation of China (61801065, 61771082, 61871062), The Program for Innovation Team Building at Institutions of Higher Education in Chongqing (CXTDX201601020)

1 引言

近年来,无线通信和机器学习的深入研究促使车辆网络的不断发展。车辆网络通过实现车内、车与车、车与人、车与路、车与云服务平台的V2X (Vehicle to everything)全面网络架构,大大提升了车辆智能化,为辅助驾驶、自动驾驶等技术提供了数据支持和可靠性保障^[1,2]。在车辆网络中,每一辆智能汽车都配备有一台车载设备(Vehicle Equipment, VE),其功能类似于一台带有联网接口的小型计算机^[3]。车载设备可以通过车辆网络进行道路危险预警,停车位导航和视频娱乐等服务^[4]。但随着车辆服务爆炸式地增长,车载设备计算能力的有限性对车辆网络构成了重大挑战^[5]。基于云的车辆网络在远程云和车载设备间分配计算资源完成任务,一定程度上缓解了对计算资源的需求^[6]。然而由于云计算服务器部署在远离车辆的位置,很难满足一些服务的时延要求^[7]。因此,移动边缘计算(MEC)作为一种解决方案得以在车辆网络中进行应用^[8]。移动边缘计算通过在无线接入网边缘部署 MEC服务器的方法计算任务,既可以满足车辆计算能力的扩展需求,也可以克服在使用远程云时时延较大的缺点^[9,10]。

在现有研究中,文献^[11]提出了联合本地和移动边缘计算的资源分配方案,该方案考虑了移动边缘计算和本地共同协作,使用强化学习及深度强化学习方法来解决计算卸载问题。但是,考虑到 MEC服务器有限的计算资源和多样化的车辆任务类型,云计算将是一种节省 MEC服务器计算资源的途径。文献^[12]提出了联合移动边缘计算与云计算的方案,该方案考虑了移动边缘计算和云计算共同协作,使用深度强化学习方法,判断是否应该将计算任务卸载到 MEC服务器上。该工作忽视了车辆本身的计算能力。如果将车辆的计算能力考虑进去,则可以减少计算时延,这对时延敏感类业务尤其重要^[7]。文献^[13]方案考虑了云计算和本地计算协作的情况,提出了车辆云计算的概念,适当地协调由车辆的计算资源组成的车辆云和远程云,向用户提供实时服务。该工作将路边单元的计算资源也考虑加入车辆云中,使用半马尔科夫决策过程进行资源分配。但是,该工作没有将路边单元和车辆的计算资源明确进行利用与划分。文献^[14]提出了一个云计算-移动边缘计算-本地计算的3层网络架构,每一层可以处理不同的任务。但这项工作没有对各层处理的任务进行详细的划分。

将车辆的本地计算资源考虑到整体的资源分配问题中,是一种可行并且高效的方案。本文在文

献^[11-14]工作基础上,提出了一种联合云计算、移动边缘计算、本地计算的多平台卸载智能资源分配算法,旨在最小化时延,提高下一代车辆网络的性能。本文所提多平台卸载智能资源分配算法首先使用K邻近(K-Nearest Neighbor, KNN)算法确定任务应该卸载到云计算服务器、MEC服务器还是本地CPU。针对卸载到 MEC服务器和云计算服务器的情况,在同时考虑系统复杂性的情况下,使用强化学习算法,再次选择计算平台,进行计算资源分配。仿真结果表明,本文所提多平台卸载智能资源分配算法可以有效地减少系统的时延成本,相对于完全卸载到本地、完全卸载到 MEC等基准算法,提出的多平台卸载智能资源分配算法平均可节省系统总成本达80%。

本文的其余部分安排如下:第2节介绍系统模型;第3节将问题描述为优化问题,并在第4节中提出多平台卸载智能资源分配算法;在第5节中,对提出算法进行了仿真分析,并进行了讨论;最后,在第6节中阐述了结论和未来的工作。

2 系统模型

本节首先介绍了网络模型,其次是任务模型。最后,给出了本文使用的计算模型。

2.1 网络模型

考虑一个分为3层进行计算的车辆网络模型,车辆的计算任务可以在本地CPU, MEC服务器或云计算服务器进行,其中有多辆车存在计算任务,如图1所示。云计算层具有很高的计算能力,但距离车辆较远,延迟较高,可以计算一些对计算要求较高,对延迟不敏感的任务,例如路径规划等任

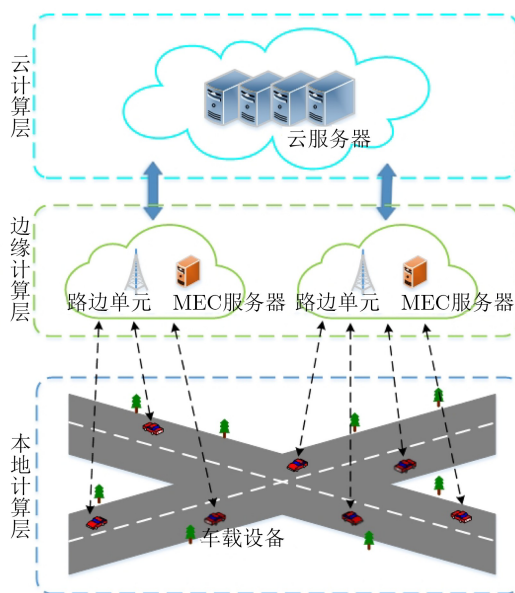


图1 网络模型

务。移动边缘计算层具有较强的计算能力，距离车辆较近，延迟较低，可以计算一些对计算要求一般，对延迟较敏感的任务，例如实时位置导航等任务。本地计算层计算能力较弱，但计算小型任务时，延迟很低。可以计算一些对计算要求很小，对延迟非常敏感的任务，例如道路危险预警等任务。这里定义所有路边单元和车辆的集合为 $K = \{1, 2, \dots, k\}$ 和 $N = \{1, 2, \dots, n\}$ ，MEC服务器的集合表示为 $M = \{1, 2, \dots, m\}$ 。

考虑车辆与其连接的路边单元之间的无线信道是实际的时变信道，所以建模为瑞利信道。假设每台车辆 n 都有一个计算任务需要完成。车辆 n 可以通过无线将任务卸载到云端或MEC服务器或在本地执行。将 W 定义为无线信道的带宽，假定多台车辆选择同时卸载任务，则将无线带宽平均分配给卸载车辆上传数据。根据文献[15]，车辆 n 可以实现的上传数据速率为

$$r_n = \frac{W}{N} \log_2 \left(1 + \frac{P_n h_n}{\frac{W}{N} N_0} \right) \quad (1)$$

其中， N 是车辆卸载的任务数量， P_n 是车辆 n 上传数据的传输功率， h_n 是车辆 n 在无线信道中的信道增益， N_0 是复高斯信道白噪声的方差[11]。

2.2 任务模型

假设每台车辆 n 都有一个计算任务 R_n ，定义为 $R_n = (B_n, D_n, \tau_n)$ ， R_n 可以卸载在车辆的本地CPU或MEC服务器或云计算服务器上计算。这里 B_n 表示计算 R_n 所需的输入数据的大小，包括程序代码和输入参数。 D_n 表示完成计算任务 R_n 所需的CPU周期总数，即为完成任务 R_n 所需的计算资源量。 τ_n 表示任务 R_n 的最大可容忍延迟，即完成每台车辆的任务的延迟不能超过 τ_n 。不同计算任务的3个参数不同，并且每个任务不能被划分。定义卸载决策向量为 $\mathbf{A} = [\alpha_1 \alpha_2 \dots \alpha_N]$ ，其中 $\alpha_n = \{\alpha_n^l, \alpha_n^m, \alpha_n^c\}$ ， $\alpha_n^l, \alpha_n^m, \alpha_n^c \in (0, 1)$ ， $\alpha_n^l + \alpha_n^m + \alpha_n^c = 1$ ，若卸载到本地CPU，MEC服务器或云计算服务器，则相应的 $\alpha_n^l, \alpha_n^m, \alpha_n^c$ 为1，否则为0。一个计算任务 R_n 只能卸载到一个平台进行计算。

2.3 计算模型

(1) 本地计算模型

如果车辆 n 选择在本地执行任务 R_n ，定义车辆 n 的本地执行延迟为 T_n^l ， T_n^l 仅包括本地CPU的处理延迟。然后，将 f_n^l 表示为车辆 n 的计算能力(每秒的CPU周期数)。任务 R_n 的本地执行延迟 T_n^l 为

$$T_n^l = \frac{D_n}{f_n^l} \quad (2)$$

定义本地计算总成本为 C_n^l

$$C_n^l = T_n^l \quad (3)$$

(2) MEC计算模型

如果车辆 n 选择在MEC服务器执行任务 R_n ，整个卸载将分为3个步骤。首先，车辆 n 需要通过无线接入网向路边单元上传 R_n 的输入数据，并且路边单元将数据转发给MEC服务器。然后，MEC服务器分配部分计算资源用来执行计算任务，最后MEC服务器将执行结果返回给车辆 n 。将 f_n^m 表示为MEC服务器的计算能力。根据上述步骤，将 $T_{n,t}^m$ 定义为车辆 n 的传输延迟

$$T_{n,t}^m = \frac{B_n}{r_n} \quad (4)$$

将 $T_{n,p}^m$ 定义为MEC服务器执行 R_n 的处理延迟

$$T_{n,p}^m = \frac{D_n}{f_n^m} \quad (5)$$

将 $T_{n,b}^m$ 定义为处理结果的下载延迟

$$T_{n,b}^m = \frac{B_n}{r_b} \quad (6)$$

根据文献[16]，下载数据速率总体上非常高，并且结果的数据大小远小于输入数据的数据大小，所以该步骤的延迟在本文的其余部分被忽略。定义MEC计算总成本为 C_n^m

$$C_n^m \approx T_{n,t}^m + T_{n,p}^m \quad (7)$$

(3) 云计算模型

如果车辆 n 选择在云服务器执行任务 R_n ，整个卸载将分为3个步骤。首先，车辆 n 需要通过无线接入网络向路边单元上传 R_n 的输入数据，并且路边单元将数据转发给云服务器；然后，云服务器分配部分计算资源用来执行计算任务，最后云服务器将执行结果返回给车辆 n ，将 f_n^c 表示为云服务器的计算能力。根据上述步骤，将 $T_{n,t}^c$ 定义为车辆 n 的传输延迟

$$T_{n,t}^c = T_{n,t}^m = \frac{B_n}{r_n} \quad (8)$$

车辆的传输延迟与在MEC模型处的传输延迟相同，因为这一步骤，都是通过无线接入网络向路边单元上传 R_n 的输入数据。

将 $T_{n,p}^c$ 定义为云服务器执行 R_n 的处理延迟

$$T_{n,p}^c = \frac{D_n}{f_n^c} \quad (9)$$

将 $T_{n,b}^c$ 定义为处理结果的下载延迟

$$T_{n,b}^c = \frac{B_n}{r_b} \quad (10)$$

由于在云计算服务器处理的任务数据较大,其结果数据也较大,所以在云计算模型部分,下载延迟不能被忽略。

考虑到以上的时延,定义云计算总成本为 C_n^c

$$C_n^c = T_{n,t}^c + T_{n,p}^c + T_{n,b}^c \quad (11)$$

整个系统的总成本定义为

$$C_{\text{all}} = \sum_{n=1}^N a_n^l C_n^l + a_n^m C_n^m + a_n^c C_n^c \quad (12)$$

3 问题描述

在本节中,将系统的任务卸载和资源分配建模为优化问题。本文的研究目标是 minimized 整个系统中所有车辆延迟的总成本。在任务的最大容忍延迟 τ_n 和MEC服务器的总计算资源 F 的限制下,问题描述为

$$\left. \begin{aligned} \min_{\mathbf{A}, \mathbf{f}} \sum_{n=1}^N a_n^l C_n^l + a_n^m C_n^m + a_n^c C_n^c \\ \text{s.t. C1: } \alpha_n^l, \alpha_n^m, \alpha_n^c \in (0,1), \alpha_n^l + \alpha_n^m + \alpha_n^c = 1 \\ \text{C2: } a_n^l T_n^l + a_n^m T_n^m + a_n^c T_n^c \leq \tau_n, \forall n \in N \\ \text{C3: } 0 \leq f_n^m \leq a_n^m F, \forall n \in N \\ \text{C4: } \sum_{n=1}^N a_n^m f_n^m \leq F, \forall n \in N \end{aligned} \right\} \quad (13)$$

式(13)中, $\mathbf{A} = [\alpha_1 \alpha_2 \cdots \alpha_N]$ 为卸载决策向量, \mathbf{f} 为MEC服务器的计算资源分配向量, 表示为 $\mathbf{f} = [f_1 f_2 \cdots f_N]$ 。

C1表示每台车辆 n 可以选择通过本地计算或MEC或云计算来执行其计算任务; C2表示完成每项任务的时间不应超过任务最大容忍延迟; C3表示为车辆任务分配的计算资源不能超过MEC服务器的总资源 F ; C4表示为车辆任务分配的计算资源总和不能超过MEC服务器的总资源 F 。

问题式(13)可以通过寻找最佳的卸载决策向量 \mathbf{A} 和计算资源分配向量 \mathbf{f} 来解决。卸载决策向量 \mathbf{A} 决定将车辆任务卸载到本地CPU或MEC服务器或云计算服务器其一进行计算。资源分配向量 \mathbf{f} 决定为任务分配多少MEC服务器的计算资源。由于卸载决策向量 \mathbf{A} 是一个2元向量,问题式(13)的可行集合和目标函数是非凸的。此外,由于车辆网络的特殊性,车辆数目是不断变化的,如果车辆任务数量增加,问题式(13)中的卸载决策向量和资源分配向量的大小将会以非常快地速度增加,而资源分配向量也会改变。一般的解决方案是将非凸问题近似为凸优化,使用传统方法解决近似凸优化问题。但是该方法可能造成损失,且对于大规模优化来说难

度较高,一般方法很难解决这种非凸问题^[11]。因此,本文提出了使用K临近和强化学习等机器学习的方法来寻找最优的 \mathbf{A} 和 \mathbf{f} ,通过确定最优的 \mathbf{A} 和 \mathbf{f} 来寻找问题式(13)的最优解。

4 多平台卸载智能资源分配算法

该部分提出了任务卸载及资源分配的智能算法。提出的算法首先考虑将任务卸载到哪个平台进行计算,然后使用强化学习方法解决MEC服务器的资源分配问题^[17]。

首先需要确定将任务卸载到哪个平台上。这是一个分类问题,需要将所有任务分为3类。本文使用KNN算法,因为该算法简单,高效,易于计算,并且可以快速分类任务。KNN算法考虑延迟 τ 和任务大小 B ,分别计算每个任务与云计算服务器、MEC服务器和本地CPU之间的欧几里德距离。距离计算公式表示为

$$D = \sqrt{(\tau_{R_j} - \tau_i)^2 + (B_{R_j} - B_i)^2} \quad (14)$$

其中, τ_{R_i} , τ_i , B_{R_i} , B_i 分别表示任务 R_i 的最大延迟,各平台的平均延迟,任务 R_i 的任务大小,各平台允许的最大任务大小。这里假定,任务的延迟和计算要求是一定会被满足的,我们的目的是寻找可以提供最小延迟的卸载平台。

本文只考虑单车辆本地计算,没有考虑车辆间的协作计算。当任务被选择分配到MEC服务器或云计算服务器时,需要考虑计算资源的分配问题。强化学习是一种算法,指从环境状态到行为映射的学习,从而最大化系统行为的环境累积奖励值。使用强化学习方法可以很好地适应车辆网络的环境并找到最佳的资源分配解决方案。路边单元负责收集MEC服务器的状态,然后将整个信息组装成系统状态。然后,路边单元将系统状态发送给代理程序,并获得用于为特定车辆分配资源的最优策略的反馈。在使用强化学习的非本地资源分配过程中,取得行动后,路边单元将发出通知,通知车辆,整个过程在算法1中表示。为了获得最优策略,有必要确定系统状态,行为和奖励函数。接下来,将分别介绍系统状态、行为和奖励函数。

(1) 状态

可用的路边单元的数量表示为 $k \in \{1, 2, \dots, K\}$, 可用的MEC服务器个数表示为 $m \in \{1, 2, \dots, M\}$, 对于车辆 n 在时间间隔 $t \in \{0, 1, \dots, T-1\}$ 内,状态向量可以描述为如下表示: $\mathbf{s} = (tc, ac)$, 其中 $tc = C_{\text{all}}$; ac 为MEC服务器的可用计算CPU周期数,可以定义为 $ac = F - \sum_{n=0}^N f_n$ 。

算法1 多平台卸载智能资源分配算法

阶段1: 初始化

- (1) 任务 R_i 的最大延迟 τ_{R_i} , 任务大小 B_{R_i} ;
- (2) 任务的当前位置 $P = 1$;
- (3) Q 矩阵, 参数 γ , 奖赏矩阵 R ;
- (4) 本地(l)、移动边缘计算(m)、云计算(c), 各平台平均延迟 τ_1, τ_2, τ_3 , 允许最大任务大小 B_1, B_2, B_3 。

阶段2: 选择任务卸载位置

计算 (τ_{R_i}, B_{R_i}) 和 (τ_1, B_1) 的欧式距离 D

$$D = \sqrt{(\tau_{R_i} - \tau_1)^2 + (B_{R_i} - B_1)^2}, P = 1$$

for $j=2, 3$ do计算 (τ_{R_i}, B_{R_i}) 和 (τ_j, B_j) 的欧式距离 d_j if $d_j < D$ then

$$D = d_j, P = j$$

end if

end for

if $P=1$

任务卸载到本地

if $P=2$ or 3

进行阶段3。

阶段3: 资源分配

if $P=3$ then在云计算服务器中计算任务 R_i

end if

if $P=2$ then

for 每次迭代 do

随机选择一个状态 s_t

for 每一步 do

从状态 s_t 的可能动作中随机选择动作 a 执行动作 a , 计算奖励 r , 进入下一状态 s'

计算

$$q(s, a) \leftarrow r(s, a) + \gamma \cdot \max[q(s', a')]$$

更新状态 $s \leftarrow s'$ until Q 矩阵稳定

end for

end for

end if

(2) 动作

在系统中, 动作被定义为卸载决策向量 $\mathbf{A} = [\alpha_1 \alpha_2 \dots \alpha_N]$ 和计算资源分配向量 $\mathbf{f} = [f_1 f_2 \dots f_N]$ 。

(3) 奖励

本文的优化问题是获得最小的总成本, 强化学习的目标是获得最大奖励, 奖励与总成本负相关, 故定义为 $\frac{tc_1 - tc(s, a)}{tc_1}$ 。这里, tc_1 为本地计算的总成本, $tc(s, a)$ 为系统的总成本。

在强化学习过程中, 本算法使用了Q-learning算法, 它是一种记录Q值的学习方法。每个状态动作对都有一个对应的值 $q(s, a)$ 。对于每一步, 代理计算 $q(s, a)$ 并存储在Q矩阵中, 这个值可以被认为是一个长期的奖励。 $q(s, a)$ 可以表示为

$$q(s, a) = r(s, a) + \gamma \cdot \max [q(s', a')] \quad (15)$$

其中, s, a 是当前状态和动作, s', a' 是下一个状态和动作。定义 γ 为学习参数, γ 值被定义为 $0 \leq \gamma \leq 1$ 。如果 γ 趋近于0, 这意味着代理主要考虑直接奖励, 如果 γ 趋近于1, 这意味着代理非常关注未来奖励。对于每一步, Q矩阵中的值被迭代。这样, 最后就可以得到最优的 f 。

5 仿真结果分析

本节将展示仿真结果用来评估所提方案的性能。对结果进行分析, 并进行讨论。

5.1 仿真结果

在仿真中, 假设的场景如下。考虑使用带宽 $W=10$ MHz的单个小小区, 并且使用MEC服务器部署的路边单元位于中心。车辆在距离路边单元100~1000 m的距离内随机分布。将所提出的算法与其他两个方法进行比较。“完全本地”表示所有车辆任务通过本地CPU执行任务。“完全MEC”表示所有车辆将其任务卸载到MEC服务器。

首先对车辆到路边单元的距离与系统总成本之间的关系进行分析。在图2中, 随着车辆与路边单元之间的距离不断增加, 多平台卸载智能资源分配算法及完全MEC算法的曲线都呈上升趋势, 而完全本地算法的曲线不随距离的增加而增加。这是因为车辆距离路边单元越远, 车辆任务的上传数据速率就越小, 处理任务的延迟就会增加, 系统的总成本必然增加。而完全本地算法是直接在本地进行任务计算的, 车辆到路边单元的距离不对其系统总成本产生影响。从图2中也可看出, 随着距离的不断增大, 本文算法与完全MEC算法的曲线距离不断增加。在400 m之后有了明显的差值, 这代表所

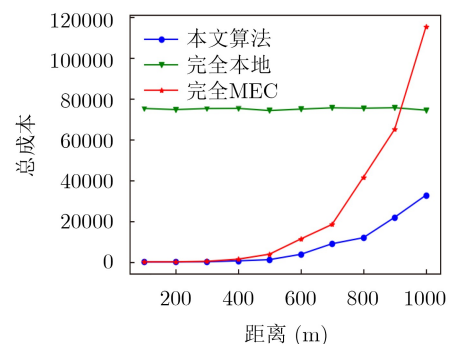


图2 总成本与距离关系图

提出的算法有着一定的适应性。在距离为700 m时,本文算法的系统总成本是10000,此时完全MEC的系统总成本是20000,所提算法节省了50%的成本。

图3为车辆数与总成本的关系图。随着横轴代表的车辆总数的增加,3种算法的曲线都呈上升趋势,即3种方法的总成本在不断增加。在相同车辆数下,本文所提多平台卸载智能资源分配算法的系统总成本最小。当车辆数为6时,多平台卸载智能资源分配算法的总成本为10000,完全MEC、完全本地的总成本分别为55000,35000。本文算法相对于完全MEC、完全本地算法分别节省了80%和70%的总成本。所提算法可以获得最佳结果,证明了本文算法的有效性。

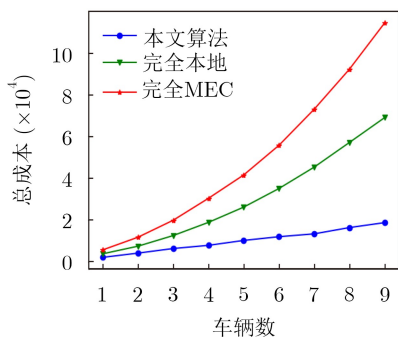
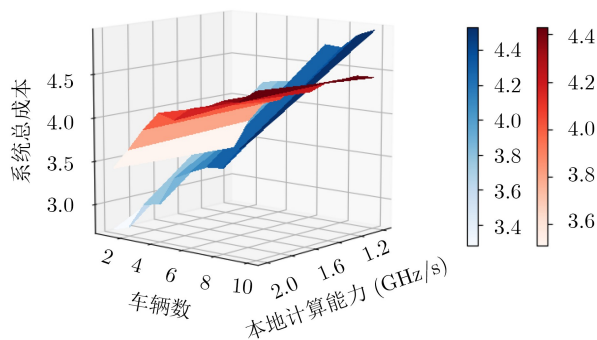


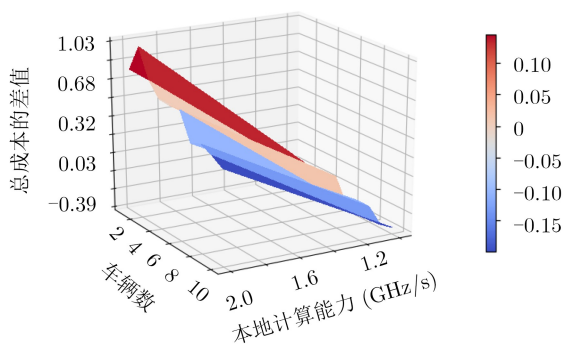
图3 总成本与车辆数关系图

图4为本地计算能力对总成本的影响关系图。图4(a)为使用多平台卸载智能资源分配算法和完全本地算法的系统总成本,红色代表本文算法,蓝色代表完全本地算法。随着本地计算能力的增加,蓝色面从高于红色面逐渐到低于红色面,这代表在相同车辆任务数下,完全本地算法的总成本随着本地计算能力的提升,逐渐低于多平台卸载智能资源分配算法的系统总成本。图4(b)作为图4(a)的补充说明,更加直观地说明了两个算法的系统总成本的关系。图4(b)是本文算法的系统总成本与完全本地算法的系统总成本之间的差值,该差值可定义为 $C_{4(b)} = C_1 - C_2$ 。 C_1 表示本文算法的系统总成本, C_2 表示完全本地算法的系统总成本。从图4(b)中可以看出,当由纵轴表示的本地计算能力增加时,平面从负值上升到正值,这表示使用多平台卸载智能资源分配算法的系统总成本从低于使用完全本地算法的系统总成本到高于使用完全本地算法的系统总成本,从两算法的差值证明了图4(a)的结论。结果表明,本文算法对不同的本地计算能力具有不同的适应性,适用于本地计算能力较弱的情况。

图5为卸载任务的数据大小 B_n 与总成本的关系。如图5所示,随着任务数据量的增加,3种算法



(a) 两算法的系统总成本



(b) 两算法的系统总成本差值

图4 总成本与车辆数及本地计算能力关系图

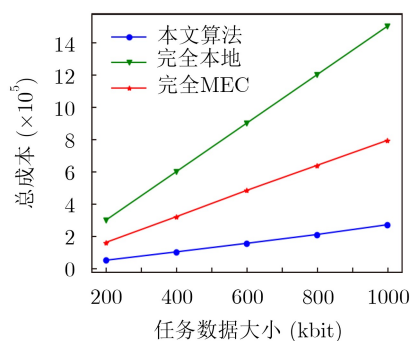


图5 总成本与任务数据大小关系图

的曲线都呈上升趋势,表示所有算法的总成本都随着数据量的增加而增加。因为处理越大的数据量会导致更多的时间消耗,增加系统的总成本。本文算法可以获得最佳结果,因为其增长曲线的斜率相较于其他两种算法的曲线的斜率最小,表示其增长的趋势最慢。随着数据量的增加,完全本地算法的曲线的斜率比其他两种算法的曲线斜率都要大,表示其系统总成本的增长速度最快。这表明计算任务的数据量越大,本地处理的时间越长,导致系统总成本增大。MEC服务器的处理能力比本地强,导致的系统总成本也比本地的总成本少。

5.2 讨论

通过对上述仿真结果的分析,可以看出本文所提多平台卸载智能资源分配算法可以有效地节省系

统总成本。这是因为本文算法从分层卸载协调和智能资源分配两方面提高了系统性能。首先是分层卸载协调。使用的KNN算法选择任务的卸载位置，进行了分层卸载协调，使得各层的计算资源都得到了充分的利用。其次是智能资源分配。使用的Q-learning算法在MEC层进行智能资源分配。它可以在任务的最大容忍延迟内，为任务分配最合适的计算资源，减少资源的浪费。综合这两部分，相较于其他传统算法，本文算法通过协调3层的计算资源，减少了完成 n 个任务的总时延，在协调了多平台的计算资源的同时又智能地分配了计算资源，使得系统的计算资源利用最大化，从而比传统算法更节省系统总成本。但同时需要指出的是，本文算法存在一定的适应性，适用于距离路边单元较远和本地计算能力相比较弱的情况。当车辆距离路边单元较近时，本文算法的优势较完全MEC算法卸载到本地时会有更少的总成本。在接下来的工作中，将会提出更具有普适性的算法，适用各种情况。

6 结束语

本文在联合移动边缘计算技术的车辆网络下提出了一个多平台卸载智能资源分配算法，可实现计算资源的动态分配，用以改善车辆网络的性能。本文算法同时考虑了云计算服务器、MEC服务器和本地CPU，以选择最适合的任务卸载平台。另外，当任务选择卸载到非本地时，使用了强化学习方法来解决其资源分配问题。仿真结果表明，本文提出的基于机器学习方法的多平台卸载智能资源分配算法可以有效地节省系统总成本，优化系统总性能。同时需要指出的是，本文算法有一定的适应性，适合车辆距离路边单元较远及本地计算能力较弱的情况。未来工作将考虑能耗等优化指标优化系统总成本，并考虑在多MEC服务器、多本地计算下的资源分配问题，提出更具有普适性的资源分配算法。

参考文献

- [1] WU Dapeng, ZHANG Feng, WANG Honggang, *et al.* Security-oriented opportunistic data forwarding in mobile social networks[J]. *Future Generation Computer Systems*, 2018, 87: 803–815. doi: [10.1016/j.future.2017.07.028](https://doi.org/10.1016/j.future.2017.07.028).
- [2] LUO Changqing, JI Jinlong, WANG Qianlong, *et al.* Channel state information prediction for 5G wireless communications: A deep learning approach[J]. *IEEE Transactions on Network Science and Engineering*, 2018. doi: [10.1109/TNSE.2018.2848960](https://doi.org/10.1109/TNSE.2018.2848960).
- [3] HUSSAIN R, SON J, EUN H, *et al.* Rethinking vehicular communications: Merging VANET with cloud computing[C]. The 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, Taipei, China, 2013: 606–609. doi: [10.1109/CloudCom.2012.6427481](https://doi.org/10.1109/CloudCom.2012.6427481).
- [4] LI Chunhai, WANG Siming, HUANG Xumin, *et al.* Parked vehicular computing for energy-efficient Internet of vehicles: A contract theoretic approach[J]. *IEEE Internet of Things Journal*, 2019, 6(4): 6079–6088. doi: [10.1109/JIOT.2018.2869892](https://doi.org/10.1109/JIOT.2018.2869892).
- [5] ZHANG Ke, MAO Yuming, LENG Supeng, *et al.* Optimal delay constrained offloading for vehicular edge computing networks[C]. 2017 IEEE International Conference on Communications, Paris, France, 2017: 1–6. doi: [10.1109/ICC.2017.7997360](https://doi.org/10.1109/ICC.2017.7997360).
- [6] ASHRAF M I, LIU Chenfeng, BENNIS M, *et al.* Dynamic resource allocation for optimized latency and reliability in vehicular networks[J]. *IEEE Access*, 2018, 6: 63843–63858. doi: [10.1109/ACCESS.2018.2876548](https://doi.org/10.1109/ACCESS.2018.2876548).
- [7] ZHANG Ke, MAO Yuming, LENG Supeng, *et al.* Contract-theoretic approach for delay constrained offloading in vehicular edge computing networks[J]. *Mobile Networks and Applications*, 2019, 24(3): 1003–1014. doi: [10.1007/s11036-018-1032-0](https://doi.org/10.1007/s11036-018-1032-0).
- [8] DING Qing, SUN Bo, and ZHNAG Xinming. A traffic-light-aware routing protocol based on street connectivity for urban vehicular Ad Hoc networks[J]. *IEEE Communications Letters*, 2016, 20(8): 1635–1638. doi: [10.1109/LCOMM.2016.2574708](https://doi.org/10.1109/LCOMM.2016.2574708).
- [9] ZHANG Ke, MAO Yuming, LENG Supeng, *et al.* Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading[J]. *IEEE Vehicular Technology Magazine*, 2017, 12(2): 36–44. doi: [10.1109/MVT.2017.2668838](https://doi.org/10.1109/MVT.2017.2668838).
- [10] HAN Guangjie, LIU Li, CHAN S, *et al.* HySense: A hybrid mobile Crowdsensing framework for sensing opportunities compensation under dynamic coverage constraint[J]. *IEEE Communications Magazine*, 2017, 55(3): 93–99. doi: [10.1109/MCOM.2017.1600658CM](https://doi.org/10.1109/MCOM.2017.1600658CM).
- [11] SOOKHAK M, YU F R, HE Ying, *et al.* Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing[J]. *IEEE Vehicular Technology Magazine*, 2017, 12(3): 55–64. doi: [10.1109/MVT.2017.2667499](https://doi.org/10.1109/MVT.2017.2667499).
- [12] LI Ji, GAO Hui, LÜ Tiejun, *et al.* Deep reinforcement learning based computation offloading and resource allocation for MEC[C]. 2018 IEEE Wireless Communications and Networking Conference, Barcelona, Spain, 2018: 1–6. doi: [10.1109/WCNC.2018.8377343](https://doi.org/10.1109/WCNC.2018.8377343).
- [13] HE Ying, ZHAO Nan, and YIN Hongxi. Integrated networking, caching, and computing for connected vehicles:

- A deep reinforcement learning approach[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(1): 44–55. doi: [10.1109/TVT.2017.2760281](https://doi.org/10.1109/TVT.2017.2760281).
- [14] LIN Chuncheng, DENG D J, and YAO C C. Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units[J]. *IEEE Internet of Things Journal*, 2018, 5(5): 3692–3700. doi: [10.1109/JIOT.2017.2690961](https://doi.org/10.1109/JIOT.2017.2690961).
- [15] MAO Yuyi, YOU Changsheng, ZHANG Jun, *et al.* A survey on mobile edge computing: The communication perspective[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(4): 2322–2358. doi: [10.1109/COMST.2017.2745201](https://doi.org/10.1109/COMST.2017.2745201).
- [16] ZHAO Pengtao, TIAN Hui, QIN Cheng, *et al.* Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing[J]. *IEEE Access*, 2017, 5: 11255–11268. doi: [10.1109/ACCESS.2017.2710056](https://doi.org/10.1109/ACCESS.2017.2710056).
- [17] ZHANG Ke, MAO Yuming, LENG Supeng, *et al.* Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks[J]. *IEEE Access*, 2016, 4: 5896–5907. doi: [10.1109/ACCESS.2016.2597169](https://doi.org/10.1109/ACCESS.2016.2597169).
- 王汝言: 男, 1969年生, 教授, 主要研究方向为泛在网络、多媒体信息处理等.
- 梁颖杰: 女, 1994年生, 硕士生, 研究方向为车联网、移动边缘计算.
- 崔亚平: 男, 1986年生, 讲师, 研究方向为毫米波通信、多天线技术、车联网等.