

基于移动边缘计算的 $GI/GI/1$ 排队建模与调度算法

张珂^{1,2} 张利国^{1,2}

摘要 针对车联网环境下路侧边缘计算节点部署不均衡、服务密度小、实时调度计算压力大等问题,提出一种基于智能车移动边缘计算(Mobile edge computing, MEC)的任务排队建模与调度算法,提供弹性计算服务,将具备感知、计算、控制功能的智能车作为移动边缘计算服务器,设计了车联网环境下的 MEC 体系架构.首先基于虚拟化技术对智能车进行虚拟化抽象,利用排队论对虚拟车任务构建了 $GI/GI/1$ 排队模型.然后基于云平台 Voronoi 分配算法对虚拟车任务进行分配绑定,进而实现了智能车的优化调度与分布式弹性服务,解决了边缘计算任务分配不均衡等问题.最后通过城市交通路网中的车辆污染排放的实时计算实验,验证了该方法的有效性.

关键词 车联网, 移动边缘计算, 排队模型, 调度, 分配算法

引用格式 张珂, 张利国. 基于移动边缘计算的 $GI/GI/1$ 排队建模与调度算法. 自动化学报, 2022, 48(7): 1737–1746

DOI 10.16383/j.aas.c190902

$GI/GI/1$ Queuing Model and Task Scheduling for Mobile Edge Computing

ZHANG Ke^{1,2} ZHANG Li-Guo^{1,2}

Abstract The edge computing of internet of vehicles is confronted with some challenges, such as the unbalanced arrangement, the service inflexible and the time delay for the real-time computing of roadside nodes. In this paper, a new queuing model and scheduling algorithm of mobile edge computing (MEC) is proposed based on intelligent vehicles integrating the sensing, computing and control together. The $GI/GI/1$ task queuing model is firstly set up for the distributed services of vehicular networks, in which intelligent vehicles are virtualized into virtual vehicles. Moreover, according to the Voronoi allocation algorithm, the tasks generated by virtual vehicles are allocated and bound to intelligent vehicles. The optimal scheduling and distributed elastic service of intelligent vehicles are presented to solve the problem of unbalanced distribution of tasks in edge computing. The simulation experiment of the vehicle pollutant emission illustrates the effectiveness of the proposed method.

Key words Internet of vehicles, mobile edge computing (MEC), queue model, scheduling, allocation algorithm

Citation Zhang Ke, Zhang Li-Guo. $GI/GI/1$ queuing model and task scheduling for mobile edge computing. *Acta Automatica Sinica*, 2022, 48(7): 1737–1746

车联网技术能够实现车与车、车与路、车与基础设施的信息交换,但对系统的实时性要求较高^[1-4].如果采用集中式的数据处理模式,即数据回传数据中心进行计算处理,则会产生较大时延.

最近兴起的边缘计算(Edge computing, EC)融合了网络、计算、存储等核心使能,分布式部署于靠近物或数据源头,是就近提供应用服务的新型计算模型^[5-9].车联网中的边缘计算^[10]主要是基于路侧单元(Road side units, RSU)进行的.将边缘计算

服务器直接部署于道路两侧,可以将更多的数据计算和存储从“数据中心”转移到“路侧单元”,部分数据不必再经过网络上传云端处理,而在本地完成数据交换及自主决策.从而降低了网络时延和负荷,有效减少网络传输量,避免网络拥塞,同时也提升了数据安全性和应用可靠性.然而,一方面由于车辆的移动特点,车载单元将频繁与不同的路侧单元进行信息交互,路侧单元间也频繁地转移计算任务,密集信息交互带来了不稳定性^[11-13],降低了任务传输的可靠性与安全性^[14];另一方面交通波动导致任务不均衡^[15],路侧单元部署规模大、成本高,有些路段可能没有部署边缘计算服务器却有较大的边缘计算需求.边缘计算任务的空间分布是不确定的,固定在路侧单元的边缘计算服务器无法对一些指定地点和时间的任务提供服务^[16],如何实现边缘计算服务器的空间优化配置成为了一项重要工作.

智能车的快速发展为上述问题提供了全新的解

收稿日期 2019-12-31 录用日期 2020-04-07

Manuscript received December 31, 2019; accepted April 7, 2020
国家自然科学基金(61873007),北京市自然科学基金(1182001)资助

Supported by National Natural Science Foundation of China (61873007) and Beijing Natural Science Foundation (1182001)

本文责任编辑 王占山

Recommended by Associate Editor WANG Zhan-Shan

1. 北京工业大学信息学部 北京 100124 2. 计算智能与智能系统北京市重点实验室 北京 100124

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124 2. Key Laboratory of Computational Intelligence and Intelligent Systems, Beijing 100124

决思路. 它们通常装备了大量的计算单元、通信设备、传感器和人机交互设备^[17-20]. 例如, 百度的无人驾驶汽车“阿波罗”装载了价值近百万元的计算机系统用于存储和运算, 用来分析车辆周边人、车、道路与环境^[21-22]. 此外, 在智能车与其他车辆行驶中将形成相对静止关系, 此时提供移动边缘计算 (Mobile edge computing, MEC) 将有效提升实时性与稳定性^[23-26].

本文提出一种车联网环境下的 MEC 架构. 与基于路侧单元的边缘计算相比, 此类 MEC 范式最大的不同之处在于: 其依赖于具有感知、计算、控制、通信能力的智能车在交通路网中提供弹性的边缘计算服务^[27-31]; 智能车作为分布式 MEC 节点, 执行局部、实时、短周期数据的处理与分析.

本文重点研究了 MEC 节点形成的移动拓扑结构, 按照计算需求的时间和空间约束进行计算任务的时空化研究, 建立了基于 $GI/GI/1$ 排队模型的虚拟车任务队列. 根据任务到达时间、任务地点、任务量等研究任务分配问题, 利用虚拟化概念最大化智能车的利用程度, 提出了基于云平台 Voronoi 的任务分配算法. 针对智能车局部任务队列, 引入虚拟服务时间、虚拟截止时间对任务执行模型进行约束. 最后以城市道路交通污染排放的实时计算为例, 讨论了分布式移动边缘计算方法的有效性.

本文结构如下: 第 1 节介绍了车联网环境下的 MEC 体系架构和任务分配算法; 第 2 节建立了 MEC 系统模型; 第 3 节介绍了道路交通污染排放计算任务; 第 4 节进行了仿真实验分析; 第 5 节总结与展望.

1 车联网环境下的 MEC 体系架构

本节设计了如图 1 所示的车联网环境下时空采样计算的 MEC 体系架构.

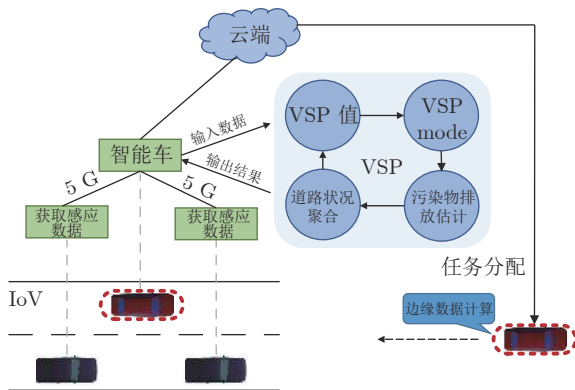


图 1 车联网环境下的 MEC 体系架构

Fig. 1 The MEC architecture for vehicle networks

该体系架构分为三层: 物理层、应用层、云端层. 物理层由道路车辆形成的车联网组成, 通过短程通信技术进行信息交换. 应用层由部署在城市交通路网中的智能车组成, 数据处理和交通信息计算服务部署于智能车上进行. 云端层拥有更多的计算和存储资源, 智能车动态感知交通信息后, 通过云端的边缘接入上传自身信息与交通信息, 云端利用移动拓扑结构对智能车进行调度.

假设交通路网中有 M 辆具有车载边缘计算服务器的智能车提供分布式服务, 记作

$$IV_s = \{IV_1, IV_2, \dots, IV_M\}, M \in \mathbf{N} \quad (1)$$

集合中下标表示智能车的编号, 所有智能车在路网中形成了移动拓扑结构, 而每辆智能车负责一个子区的计算任务. 由于智能车可以在交通路网中移动, 因而车载服务器具有空间属性和速度属性.

为实现多用户可以独立使用智能车而互相不影响, 应用云计算范式对智能车进行虚拟化, 则用户可在云端购买虚拟车 (Virtual vehicles, VV), 以此获得智能车的某些计算或控制服务, 该虚拟车反映了智能车的空间属性和速度属性. 记用户购买的 I 辆虚拟车集合

$$VV_s = \{VV_1, VV_2, \dots, VV_I\}, I \in \mathbf{N} \quad (2)$$

集合中下标表示虚拟车的编号. 对于用户而言, 事先并不知道智能车的分布情况, 只需要在云端购买一辆虚拟车, 指定特定地点的计算任务即可. 特定地点的计算任务通常可以表示为

$$Task = \{T^{arr}, task^X, T^s\} \quad (3)$$

其中, T^{arr} 表示任务到达时间; $task^X$ 表示任务在空间中的位置, 可以用经纬度 (lat, lon) 表示; T^s 表示任务量.

在多任务情况下, 虚拟车必须完全隔离, 因此需要对虚拟车进行时空建模, 描述其在时间和空间上的计算行为.

边缘计算中的核心使能技术是虚拟化, 通过虚拟化技术将物理层的 CPU、内存、传感器等硬件资源映射到虚拟层上, 用户任务通过虚拟车的时空推进程序, 利用智能车的传感器、计算资源、控制器、通信设备与周围环境进行交互. 则虚拟车任务可以看成一种离散的时空推进模型, 这里引入 Plotkin 符号语义^[32] 如下

$$\begin{aligned} \langle tc_1, t_1, x_1 \rangle &\rightarrow \langle tc_2, t_1 + k, x_1 + n \rangle \rightarrow \dots \rightarrow \\ &\langle tc_i, t_1 + (i-1)k, x_1 + \\ &(i-1)n \rangle \rightarrow \dots \end{aligned} \quad (4)$$

其中, tc_i 为任务程序, 例如采样其他车辆信息并做交通污染估计; $t_1, t_1 + k, \dots, t_1 + (i-1)k, \dots$ 为

时钟序列, 时间间隔为 k ; 箭头表示的流是时空推进; $x_1, x_1 + n, \dots, x_1 + (i-1)n, \dots$ 为空间序列, 空间间隔为 n .

当 $n = 0$ 时, 表示时间上推进而空间上没有推进, 此时虚拟车处于静止状态, 可以表示为

$$\langle tc_1, t_1 \rangle \rightarrow \langle tc_2, t_1 + k \rangle \rightarrow \dots \rightarrow \langle tc_i, t_1 + (i-1)k \rangle \rightarrow \dots \quad (5)$$

此时虚拟车可以在某一位置提供 MEC 服务.

同样, 有些空间工程采样并不需要时间参考, 此时虚拟车仅在空间上推进, 其空间推进模型可以表示为

$$\langle tc_1, x_1 \rangle \rightarrow \langle tc_2, x_1 + n \rangle \rightarrow \dots \rightarrow \langle tc_i, x_1 + (i-1)n \rangle \rightarrow \dots \quad (6)$$

利用时空推进模型可以指定智能车的行为, 为保证 MEC 云平台的调度系统能够快速响应, 需要进行 MEC 调度系统建模.

2 MEC 系统模型

MEC 中资源调度的实质是根据虚拟车产生的不同计算任务、任务到达时间、任务类型等, 云端调度系统选择合适的智能车前往特定地点执行计算任务, 为此需要研究调度模型以最大化利用智能车资源. 在 MEC 中已经实现了资源的虚拟化. 本节将进行 MEC 系统建模.

2.1 虚拟车任务队列模型

由第 1 节基本概念可知, 智能车集合 $IV_s = \{IV_m\}_{m=1}^M$, 虚拟车集合 $VV_s = \{VV_i\}_{i=1}^I$. 其中 M 是确定的, I 根据实际需求动态变化. 每辆智能车速度记为 v^{IV} , 虚拟车速度记为 v^{VV} , 虚拟车速度其实就是智能车在物理世界中行驶速度的映射.

虚拟车可以生成任务序列 $\langle Task_i(j) \rangle_{j=1}^{\infty}$, 表示第 i 辆虚拟车产生的第 j 个任务, 每个任务都包括了其到达时间 $T_i^{arr}(j)$, 任务位置 $task_i^X(j)$, 计算量 $T_i^s(j)$. 任务序列 $\langle Task_i(j) \rangle_{j=1}^{\infty}$ 由不同的虚拟车产生且相互独立, 因此 $Task_i(j)$ 关于 i 是相互独立的.

假设每个任务到达过程 $\{T_i^{arr}(j)\}_{j=1}^{\infty}$ 是一个更新过程. 则任务时间间隔 $I_i^{arr}(j) = T_i^{arr}(j) - T_i^{arr}(j-1)$ 关于 j 独立同分布. 其中 $T_i^{arr}(0) \equiv 0$, 所以每辆虚拟车的任务到达速率表示如下

$$\lambda_i^{VV} = \frac{1}{E[I_i^{arr}]} \quad (7)$$

式中, λ_i^{VV} 表示第 i 辆虚拟车的任务到达速率, I_i^{arr} 表示任务时间间隔的一般项, $E[I_i^{arr}]$ 表示期望.

任务位置 $task_i^X(j)$ 关于 i 和 j 独立同分布, 可

以用经纬度来表示每辆虚拟车产生的每个任务地理位置, $task_i^X(j) = (lat_{ij}, lon_{ij})$, 其中 lat_{ij} 表示纬度, lon_{ij} 表示经度. 同样, 任务计算量 $T_i^s(j)$ 也是关于 i 和 j 独立同分布, 如果将 T_i^s 作为 $T_i^s(j)$ 的一般项, 则有 $E[T_i^s] = E[T_i^s(j)]$.

虚拟车按照先到先服务 (First come first service, FCFS) 的策略执行任务序列 $\langle Task_i(j) \rangle_{j=1}^{\infty}$, 依次对指定地点 $task_i^X(1), task_i^X(2), \dots$ 执行任务. 并分别执行计算任务时间 $T_i^s(1), T_i^s(2), \dots$. 可以计算得到虚拟旅行时间

$$T_i^{Vtra}(j) = \frac{D_i^V(j)}{v^{VV}} \quad (8)$$

式中, $T_i^{Vtra}(j)$ 表示第 i 辆虚拟车前往第 j 个任务点的旅行时间; $D_i^V(j)$ 表示第 $j-1$ 次任务位置 $task_i^X(j-1)$ 与第 j 次任务位置 $task_i^X(j)$ 的路径距离, 其中 $task_i^X(0)$ 是指初始任务分配绑定到智能车时智能车的地理位置.

记任务 $Task_i(j)$ 的虚拟服务时间为 $T_i^{Vser}(j)$, 包括前往指定位置的旅行时间和执行任务时间, 则

$$T_i^{Vser}(j) = T_i^{Vtra}(j) + T_i^s(j) \quad (9)$$

事实上, 虚拟服务时间与物理世界中的智能车服务时间会有偏差, 因此设定虚拟截止时间作为对智能车的约束. 而虚拟截止时间的计算是以任务到达时间 $T_i^{arr}(j)$ 算起, 如果第 j 次任务到达时上次任务仍在进行中, 则应当以第 $j-1$ 次任务的虚拟截止时间算起, 两种任务虚拟截止时间计算情况如图 2 所示.

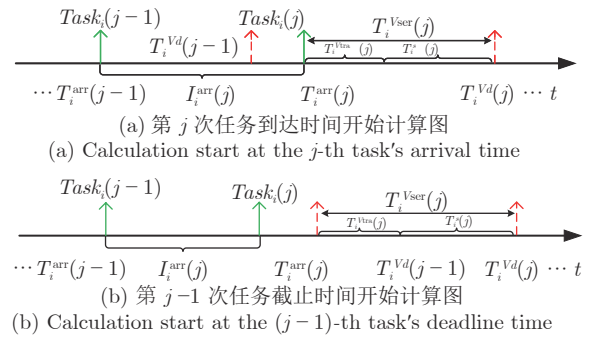


图 2 虚拟车任务虚拟截止时间计算示意图

Fig. 2 Virtual vehicle task's deadline time calculation schematic diagram

记第 i 辆虚拟车产生的第 j 个任务的虚拟截止时间为 $T_i^{Vd}(j)$, 则有

$$T_i^{Vd}(j) = \max \{T_i^{arr}(j), T_i^{Vd}(j-1)\} + T_i^{Vser}(j) \quad (10)$$

其中 $T_i^{Vd}(0) \equiv 0$, 表示虚拟车无任务时其默认值为

0. 当 $T_i^{\text{arr}}(j) > T_i^{Vd}(j-1)$ 时, 即图 2(a) 所示; 当 $T_i^{\text{arr}}(j) < T_i^{Vd}(j-1)$ 时, 即图 2(b) 所示.

因为每辆虚拟车产生的任务到达速率是 $\lambda_i^{VV} = 1/E[T_i^{\text{arr}}]$, 定义虚拟服务速率的一般形式为 μ^{VV} , 则有

$$\mu^{VV} = \frac{1}{E[T^{V\text{ser}}]} = \frac{1}{E[T^{V\text{tra}}] + E[T^s]} = \frac{1}{\frac{E[D^V]}{v^{VV}} + E[T^s]} \quad (11)$$

同时, 输出的虚拟截止过程 $\{T_i^{Vd}(j)\}_{j=1}^{\infty}$ 是一个随机变量. 则虚拟截止速率如式 (12) 所示

$$\lambda_i^{Vd} = \lim_{i \rightarrow \infty} \frac{1}{E[T_i^{Vd}(j) - T_i^{Vd}(j-1)]} \quad (12)$$

如果用户提交了任务, 产生了任务队列, 则虚拟车 VV_i 处于繁忙状态, 其工作状态记为 $S_i = 1$; 如果用户没有提交任务, 则虚拟车处于空闲状态, 记为 $S_i = 0$. 虚拟车 VV_i 将呈周期性循环这两种状态. 定义 $T_i^{V\text{busy}}(l)$ 为第 l 次繁忙状态时间, $T_i^{V\text{idle}}(l)$ 为第 l 次空闲状态时间, 则定义第 i 辆虚拟车的利用率为

$$u_i^V = \lim_{l \rightarrow \infty} \frac{E[T_i^{V\text{busy}}(l)]}{E[T_i^{V\text{busy}}(l)] + E[T_i^{V\text{idle}}(l)]} \quad (13)$$

因为任务到达时间间隔输入和服务时间都是独立同分布的变量, 则虚拟车的任务队列构成了 $GI/GI/1$ 队列模型, 用户按照自己选择的任务到达速率创建任务提交到虚拟车. 当 $\lambda_i^{VV} < \mu^{VV}$ 时, 虚拟截止率等于到达率, 即 $\lambda_i^{Vd} = \lambda_i^{VV}$. 如果用户提交任务频繁, 超过了约定速度确定的服务速率, 此时 $\lambda_i^{VV} \geq \mu^{VV}$, 那么虚拟截止速率等于服务速率, 即 $\lambda_i^{Vd} = \mu^{VV}$.

2.2 云平台调度模型

MEC 调度系统中有 M 辆智能车运行在路网中, 虚拟车产生的每个特定地点计算任务都需要由实体的智能车来完成, 也就是计算任务 $Task_i(j)$ 按照某种分配机制分配给智能车 $\{IV_m\}_{m=1}^M$ 中的某辆车去执行. 从第 2.1 节可知, 在时变环境中, 虚拟车产生的计算任务是一个排队过程, 是有时间限制的. 为了更高效地提供计算服务, 最小化任务的预期系统时间, 本节采用分布式自适应策略, 引入一种 Voronoi 分配算法来解决该问题.

智能车上的导航定位装置可实时获取其地理位

置坐标. 记 t 时刻 M 辆智能车的位置集合为 $p^{IV}(t) = \{p_m^{IV}(t)\}_{m=1}^M$, 其中 $p_m^{IV}(t) = (lat_m, lon_m)$, 表示编号为 m 的智能车的经纬度坐标. t 时刻虚拟车上生成的计算任务地点由第 2.1 节可知 $p^{VV}(t) = \{task_i^X(t)\}_{i=1}^I$, 其中第 i 辆虚拟车产生任务的地理位置坐标 $task_i^X(t) = (lat_i, lon_i)$.

给定时间 t , Voronoi 分配算法表示如下

$$r = \text{VorAllocation}(A, task_i^X(t), p^{IV}(t)) \quad (14)$$

式中, VorAllocation 为分配算法函数, 该函数接受三个参数; A 为智能车覆盖的服务区域, 包含 A 区域交通路网中的 node 节点和 edge 边的数据信息; $task_i^X(t)$ 是 t 时刻第 i 辆虚拟车到达任务的位置; $p^{IV}(t)$ 是 t 时刻智能车的地理位置; $r \in \{1, \dots, M\}$ 表示分配的智能车编号.

根据 A 区域中有 M 辆智能车, Voronoi 分配算法利用连续多中值均匀产生的 M 个点将 A 区域细分为 M 个子区域, 每个子区域是一个 Voronoi 单元, 则 $A = \{\text{VorCell}_m\}_{m=1}^M$. M 辆智能车分别被派到 M 个子区域中的中值点等待计算任务分配. 此时 M 个中值点即为智能车的初始位置 $p_m^{IV}(0)$. 则 Voronoi 单元表示如下

$$\text{VorCell}_m = \{y \in A : \|y - p_m^{IV}(0)\| \leq \|y - p_k^{IV}(0)\|, k = 1, \dots, M; k \neq m\} \quad (15)$$

式中, VorCell _{m} 表示第 m 个 Voronoi 单元; y 为 A 区域的任意一点; $p_m^{IV}(0)$ 为第 m 辆智能车的初始位置; $p_k^{IV}(0)$ 表示除第 m 辆智能车外的其他智能车初始位置, 故 $k \neq m$.

对于任意的 $t > 0$, $p_m^{IV}(t) \in \text{VorCell}_m$. 当虚拟车产生任务指定地点 $task_i^X(t)$ 时, 如果 $task_i^X(t) \in \text{VorCell}_m$, 那么该任务落入第 m 个 Voronoi 单元, 将该任务分配给该单元的智能车.

2.3 智能车局部队列模型

智能车针对落入该子区域的任务将生成一个新的任务序列 $\langle Task_m(n) \rangle_{n=1}^{\infty}$, 任务 $Task_m(n)$ 表示分配的第 n 个任务将由编号为 m 的智能车前往执行.

在实际任务执行中, 虚拟服务时间与物理世界中的智能车服务时间会有偏差. 智能车利用排队论的调度策略, 包括先到先服务 (First come first service, FCFS), 即最先到达队列的任务最先进行服务; 最早截止优先服务 (First deadline first service, FDFS), 即按照截止时间先后顺序重新进行排序, 依次进行服务; 信用优先服务 (Credit first service, CFS), 将信用分为 2、4、6、8、10 五个等级, 按照优

先级大的依次进行服务. 通过对比, 按照时间最优的策略执行任务, 令 Φ 表示调度策略, 则有

$$\Phi = \{\text{FCFS}, \text{FDFS}, \text{CFS}\}, \phi \in \Phi \quad (16)$$

其中, ϕ 为集合中的调度策略. 定义 $D_m^I(n)$ 为在使用调度策略后的第 $n-1$ 次任务和第 n 次任务间的旅行距离, 则可计算出智能车实际的旅行时间, 记为

$$T_m^{\text{tra}}(n) = \frac{D_m^I(n)}{v^{IV}} \quad (17)$$

式中, $T_m^{\text{tra}}(n)$ 表示第 m 辆智能车进行第 n 次任务的服务时间, v^{IV} 为平均速度. 记其任务执行时间为 $T_m^{\text{Is}}(n)$, 则智能车实际的服务时间就是

$$T_m^{\text{Iser}}(n) = T_m^{\text{tra}}(n) + T_m^{\text{Is}}(n) \quad (18)$$

令 $T_m^{\text{Iser}\phi}$ 表示在调度策略 ϕ 下的服务总时间, 则我们希望找到服务总时间最小的调度策略

$$\phi^{\text{optimal}} = \min_{\phi \in \Phi} T_m^{\text{Iser}\phi} \quad (19)$$

式中, ϕ^{optimal} 表示最优策略, 智能车将按照该调度策略执行任务. 记智能车实际完成任务的服务速率一般项为 μ^{IV} , 则有

$$\begin{aligned} \mu^{IV} &= \frac{1}{\text{E}[T^{\text{Iser}}]} = \\ &= \frac{1}{\text{E}[T^{\text{tra}}] + \text{E}[T^{\text{Is}}]} = \\ &= \frac{1}{\frac{\text{E}[D^I]}{v^{IV}} + \text{E}[T^{\text{Is}}]} \end{aligned} \quad (20)$$

同时记智能车实际完成任务时间为 $T_m^{\text{finish}}(n)$. 因为虚拟车显示的速度是物理世界中智能车速度的映射, 假设用户不会频繁提交任务, 这样系统便是稳定的. 在此定义 κ 为映射程度^[29], 其表示为

$$\kappa = \frac{\mu^{IV}}{\mu^{VV}} = \frac{\frac{\text{E}[D^I]}{v^{IV}} + \text{E}[T^{\text{Is}}]}{\frac{\text{E}[D^V]}{v^{VV}} + \text{E}[T^{\text{Is}}]} \quad (21)$$

式中, μ^{IV} 表示智能车服务速率, μ^{VV} 表示虚拟车服务速率. 若 κ 接近 1, 说明智能车与虚拟车服务速率匹配; 若 κ 较大, 则说明智能车的服务速率较差.

由此可以得到如图 3 所示的任务队列调度模型图, 一个调度模型通常由输入任务、调度决策、任务执行体组成. 已知虚拟车产生的任务到达速率 λ_i^{VV} 以及 Voronoi 分配算法, 则其调度流程如下.

1) 输入任务: MEC 资源池中的各个任务相互独立, 并且各个任务到达速率为 λ_i^{VV} , 即单位时间间隔内有 λ_i^{VV} 个任务到达. 所有虚拟车的任务到达云平台调度中心构成全局任务队列.

2) 调度决策: 任务到达云平台调度中心后, 根据 Voronoi 分配算法, 判断各个任务地点落入哪一 Voronoi 单元, 然后将该任务分配给该单元内的智能车, 即

$$\begin{aligned} &\text{if } \text{task}_i^X(t) \in \text{VorCell}_m, \\ &\text{then } IV_m \leftarrow \{\text{Task}_i(t)\} \end{aligned}$$

3) 任务执行体: 同一虚拟车的任务落入不同的 Voronoi 单元, 将由不同的智能车进行计算服务, 所以 M 辆智能车都有自己的局部任务队列. 智能车将按照局部调度策略对任务队列进行任务排序, 依次对指定地点执行 MEC 任务, 并在时间期限内完成计算任务.

3 道路交通污染排放计算

随着电动车、机动车、混合动力车等多能源结构汽车上路, 将呈现出混合交通流的趋势. 原来利

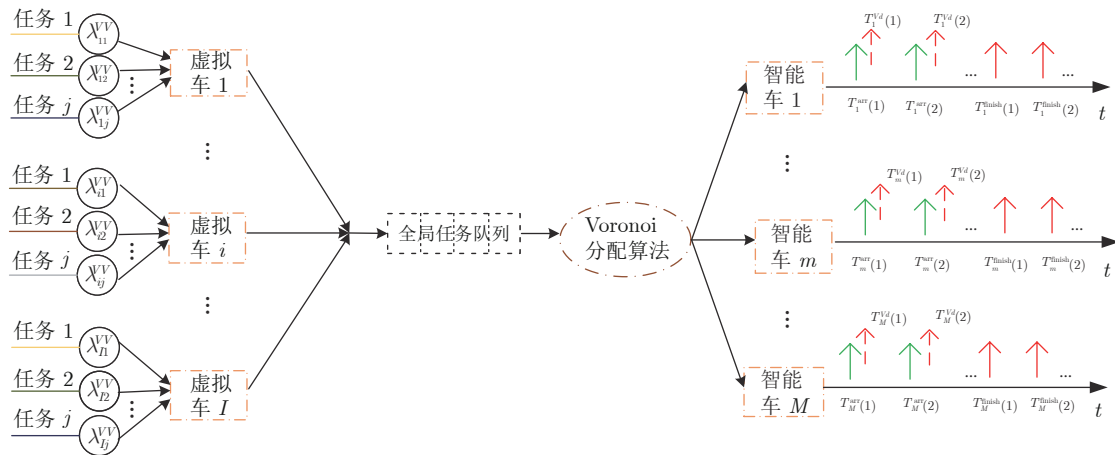


图 3 基于 Voronoi 分配算法的任务队列调度模型图

Fig. 3 Task queue scheduling model diagram based on Voronoi allocation algorithm

用宏观交通流数据作机动车尾气排放计算的方法已不再适用,而交通管理部门对准确反映且灵活地获取某一地点的交通污染排放情况有着强烈的需求.

为完成混合交通流下的移动交通污染排放计算任务,本文提出将 VSP (Vehicle specific power) 模型部署在智能车上.即交管部门提交计算任务地点与计算量后,系统根据任务地点落入哪一 Voronoi 单元而将任务分配给该单元内的智能车,智能车利用短程通信技术与指定路段的车辆建立通信,发起信息获取请求,收集道路上车辆类型、运行数据,利用 VSP 模型执行交通污染排放的动态测算任务.

VSP 理论的物理意义是瞬态机动车输出功率与机动车质量的比值,与车辆的瞬态排放具有较强的相关性,能够评价以秒为单位的瞬间排放量.不同的机动车类型具有不同的 VSP 计算公式,根据已有研究,我国城市小型轿车的交通污染排放可以简化为由速度、加速度和坡度组成的 VSP 表达式,如式 (22)所示

$$VSP = v [1.1a + 9.8 (\arctan (\sin (grade))) + 0.132] + 0.000302 \times v^3 \tag{22}$$

商务车 VSP 的计算公式,如式 (23) 所示

$$VSP = v [a + 9.8 (\arctan (\sin (grade))) + 0.09199] + 0.000169 \times v^3 \tag{23}$$

式中, v 为机动车瞬时速度 (m/s); a 为机动车瞬时加速度 (m/s²); $grade$ 为道路坡度,当机动车在城市道路中行驶时道路坡度取 0.

从式 (22)、式 (23) 可以看出,想要计算一辆机动车的 VSP 值需要获取车辆类型、瞬时速度、瞬时加速度等参数,而车辆移动传感技术 (如 GPS、陀螺仪、加速度计等微传感器) 迅猛发展使得获取这些参数成为可能.智能车获得该路段内的车辆类型、速度、加速度、坡度等信息后,根据 VSP 计算公式得到每辆车的 VSP 值,结合 BIN 方法便能够对排放因子进行分析计算,对照 VSP BIN 表 1 即可获得不同类型的机动车瞬时排放量.将该路段的车辆瞬

时排放进行累加便可得到路段的交通污染排放水平.

本节提出利用智能车 MEC 技术前往指定路段执行交通污染排放计算任务,为了验证系统的有效性和任务流程的可行性,将进行仿真实验进行相应的实验验证.

4 仿真实验

本文在 Eclipse 中搭建仿真项目,部署调度模型算法,验证系统模型的有效性.并在仿真软件 Aimsun 中进行交通污染排放计算任务调度实验,建立仿真地图、路网、交通路况等.通过 Aimsun API 接口编写算法控制智能车前往指定地点执行任务,记录各阶段时间,最后进行分析.

在 Eclipse 工程实验中,我们设计了如图 4 所示的 MEC 调度系统仿真模式类图,创建了 VirtualVehicle, TaskData, GlobalQueue, Scheduler, IntelligentVehicle 5 个类,从而建立了运行时仿真环境.其中 VirtualVehicle 可以创建多个线程实例,按照设定的到达规律生成任务 TaskData,这里时间间隔由 sleep (InterArrivalTime) 函数实现.TaskData 作为容器将保存虚拟车生成的任务信息.在运行时环境下,VirtualVehicle 将生成 TaskData 任务对象保存到自己的队列中,并发送到全局队列,全局队列由 ArrayBlockingQueue 实现.可以

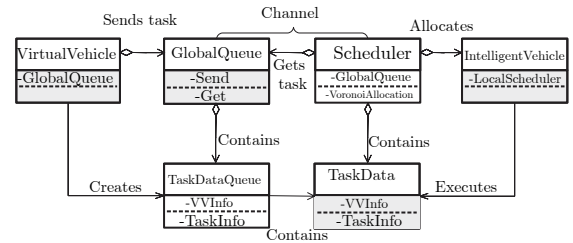


图 4 MEC 调度系统仿真模式类图

Fig.4 Simulation mode class diagram of MEC scheduling system

表 1 VSP 排放等级与平均排放清单
Table 1 VSP modes and the average modal emission rates of each

VSP 等级	VSP mode	CO ₂ (g/s)	CO (g/s)	NO _x (g/s)	HC (g/s)
VSP < -2	1	1.54369	0.01103	0.00101	0.00090
-2 ≤ VSP < 0	2	1.60441	0.00872	0.00104	0.00090
0 ≤ VSP < 1	3	1.13083	0.00468	0.00042	0.00084
...
28 ≤ VSP < 33	12	7.61770	0.24781	0.01438	0.00457
33 ≤ VSP < 39	13	8.32244	0.41307	0.01597	0.00570
39 ≤ VSP	14	8.47503	0.62466	0.01672	0.00716

实现 VirtualVehicle 和 Scheduler 的互斥操作, 保证全局队列中数据的安全性. 全局队列的大小是动态变化的, Scheduler 可以从全局队列中取出任务并按照任务地点落入 Voronoi 单元的编号将任务对象发送给对应的 IntelligentVehicle. GlobalQueue 和 Scheduler 在此承担了信息通道 (Channel) 的作用, IntelligentVehicle 拥有本地调度器, 将根据不同的调度规则计算服务总时间, 并得到最优的调度策略.

算法操作步骤如下:

- 1) 设定虚拟车数量 i 和智能车数量 m ;
- 2) 设定全局任务队列阈值;
- 3) 实例化 i 个 VirtualVehicle 线程实例 VV_i , m 个 IntelligentVehicle 线程实例 IV_m ;
- 4) 设定 VV_i 产生任务的时间到达规律参数、任务地点范围、任务量及范围;
- 5) VV_i 生成任务, TaskData 存储任务的到达时间、地理位置、任务量;
- 6) 按照式 (9) 计算虚拟服务时间;
- 7) 按照式 (10) 计算虚拟截止时间;
- 8) VV_i 将任务打包发送到 Channel 通道中的

全局任务队列;

9) 调度器 Scheduler 从全局任务队列取出任务, 按照式 (14)、式 (15) 将任务分配到对应的智能车 IV_m ;

10) 智能车 IV_m 将接收到的任务放入自己的局部队列, 按照式 (16) 调度策略及式 (17) ~ 式 (19) 分别计算 $T_m^{Iser\phi}$;

11) 重复步骤 4) 至步骤 10), 直至达到额定任务数量, 求得服务总时间最小的调度策略 $\phi^{optimal}$.

实验调用 API 从 Open Street Map 获取以北京工业大学为中心, 方圆 3000 米范围的交通路网拓扑图作为实验区域数据. 任务处理过程中的其中 6 个任务分配结果如图 5 所示, 最底层是交通路网, 9 个方块表示在 A 区域内提供边缘计算的智能车, 圆点表示虚拟车发出的计算任务位置, 通过 Voronoi 算法可以计算得到分界线, 将该区域分成了 9 个 Voronoi 单元, 区域索引与智能车索引相同. 创建 9 个 IntelligentVehicle 线程实例模拟 9 辆智能车, 随机产生 9 个均匀分布在路网中的地理位置坐标作为智能车初始位置坐标, 如图 5(a) 中的方块所示. 创建 16 个 VirtualVehicle 线程实例模拟 16 辆虚拟

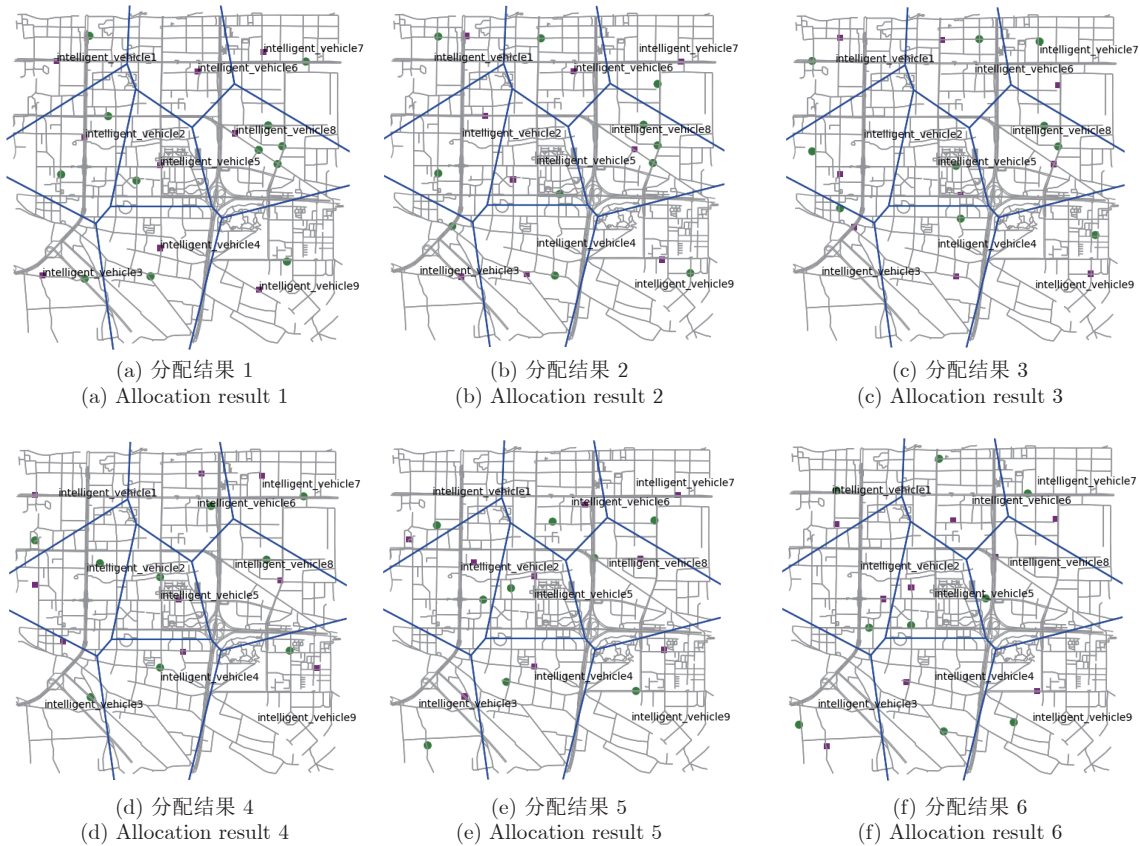


图 5 基于 Voronoi 算法的任务分配结果示意图
Fig.5 Allocation result based on Voronoi algorithm

车, 每辆虚拟车都将按照不同的分布规律产生 50 个任务, 任务被动态地发送到全局队列. Scheduler 按照式 (15) 将该区域分成 9 个 Voronoi 单元, 即 $A = \{VorCell_m\}_{m=1}^9$, 逐个将全局队列中的任务取出, 按照 Voronoi 分配算法进行分配.

智能车根据 FCFS, FDFS, CFS 调度策略对局部任务队列进行计算, 得到各自最优的调度策略. 在 FCFS, FDFS, CFS 三种调度算法下的智能车服务时间对比结果如图 6 所示.

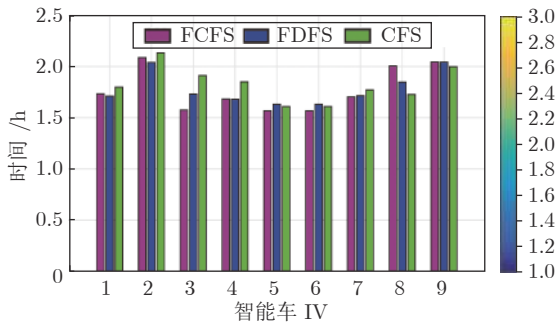


图 6 FCFS, FDFS, CFS 调度算法下智能车服务时间对比

Fig. 6 Service time comparison under scheduling algorithm of FCFS, FDFS, CFS

调度完成后在 Aimsun 中模拟智能车提供交通污染排放计算服务的任务场景. 利用 OSMnx 库可以获取北京市朝阳区的交通路网 GIS 数据, 利用 Aimsun 仿真软件中的插件 GIS Importer 将该数据导入 Aimsun 生成路网, 建立交通网络.

选择图 5 中的第 5 辆智能车所在区域 (中间区域) 作为实验区域, 根据 OD 矩阵将普通车辆放入仿真的真实场景中, 然后放入智能车 IV_5 作为控制对象. 通过 Aimsun API 编写程序模拟 IV 执行任务, 落入第 5 单元区域的任务被 IV 生成局部任务序列 $\langle Task_5^X(n) \rangle_{n=1}^6$, 其任务地点标注如图 7 中的圆点所示, 智能车 IV_5 初始位置如图 7 中车辆所示.

仿真程序按照 FCFS 调度策略执行任务. 读取第一个任务地点 $task_5^X(1)$, 智能车所在地点 $p_5^{IV}(0)$, 计算得到第一次执行任务地点与智能车初始位置之间的距离 $D_5^I(1)$, 通过获取路段平均车速可以计算出智能车到达任务点所在路段的实际旅行时间 $T_5^{Itra}(1)$. 同理计算出智能车通过任务地点所在路段的时间, 这段时间也是智能车执行任务时间, 即 $T_5^{Is}(1)$.

通过式 (18) 可以得到实际服务时间 $T_5^{Iser}(1)$, 即智能车需要提供服务的时间. 智能车物理移动前往任务地点, 按照虚拟截止时间约束完成计算任务.

实验中设置跟踪模式, 记录车辆实际旅行时间、

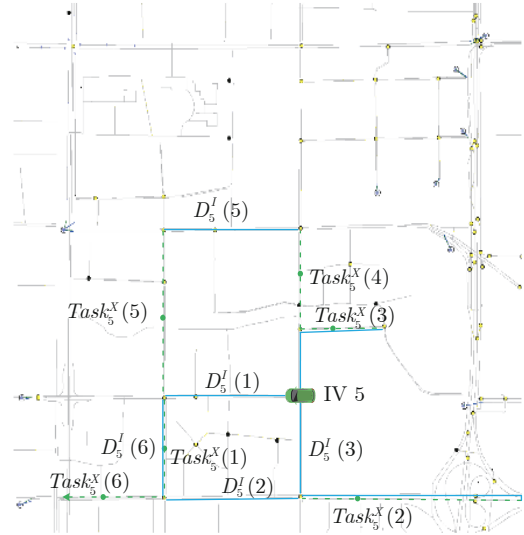


图 7 交通污染排放计算的智能车调度策略

Fig. 7 Intelligent vehicle scheduling strategy for traffic pollutant emission computing

实际执行任务时间、实际服务时间和实际完成任务时间. 第一个交通污染排放计算任务完成后, 程序将控制智能车前往第二个任务地点 $task_5^X(2)$ 执行交通污染排放计算任务. 同理, 智能车将依次按照任务分配前往 $task_5^X(3) \sim task_5^X(6)$ 执行任务. VV_s 发出的任务计算参数如表 2 所示.

执行交通污染排放计算任务时, 采用 Plotkin 时空推进式 (4), 设置程序时间间隔为 1 s, 同时仿真步长设置成 1 s. 智能车可以获得该路段所有车辆的动态信息, 通过获取不同车辆的瞬时速度 v 和加速度 a , 利用 VSP 模型可以得到每辆车的 VSP 值, 根据 BIN 方法得到每辆车的瞬时排放, 将计算得到的所有值进行累加得到该路段的整体交通污染排放情况. 在此过程中, 智能车的实际运行参数以及在每次任务中计算得到的交通污染排放情况如表 3 所示.

由式 (7)、式 (11) 可知, 任务到达速率 $\lambda^{VV} = 0.375$, 虚拟服务速率 $\mu^{VV} = 0.448$, 此时 $\lambda^{VV} < \mu^{VV}$. 经计算得到智能车真实服务速率 $\mu^{IV} = 0.476$, 通过式 (21) 得到 $\kappa = 1.0625$, 接近 1. 由文献 [29] 可知映射程度较好, 提高了智能车的计算资源利用率.

5 结论

本文设计了一种智能车联网环境下的 MEC 体系架构. 采用虚拟化技术对智能车计算资源进行了虚拟化抽象, 构建了虚拟车服务任务的 $GI/GI/1$ 排队模型, 同时基于云平台的 Voronoi 分配算法, 对虚拟车任务进行了分配绑定, 进而实现了智能车的

表 2 VVs 的任务计算参数
Table 2 VVs calculation parameters

$Task_i(j)$	$T_i^{arr}(j)$	$task_i^X(j)$	$T_i^{V^{tra}}(j)$	$T_i^s(j)$	$T_i^{V^{ser}}(j)$	$T_i^{V^d}(j)$
$i = 2, j = 2$	9:09:02	(39.8726, 116.466)	89	169	258	9:13:20
$i = 3, j = 3$	9:13:37	(39.8702, 116.476)	110	80	190	9:16:47
$i = 7, j = 1$	9:16:52	(39.875, 116.475)	200	60	260	9:21:12
$i = 9, j = 6$	9:21:14	(39.8787, 116.471)	0	52	52	9:22:06
$i = 12, j = 3$	9:22:06	(39.8767, 116.466)	78	60	138	9:24:24
$i = 15, j = 4$	9:24:34	(39.8705, 116.466)	40	43	83	9:25:57

表 3 IV 的实际运行参数
Table 3 Actual operating parameters of IV

$Task_m(n)$	$T_m^{tra}(n)$	$T_m^s(n)$	$T_m^{I^{ser}}(n)$	$T_m^{finish}(n)$	CO ₂ (g)	CO (g)	NO _x (g)	HC (g)
$m = 5, n = 1$	95	178	273	9:13:35	992.78	6.76	0.81	0.51
$m = 5, n = 2$	117	76	193	9:16:50	1144.42	7.76	0.79	0.63
$m = 5, n = 3$	202	58	260	9:21:12	426.80	2.96	0.29	0.24
$m = 5, n = 4$	0	50	50	9:22:04	590.63	4.11	0.41	0.33
$m = 5, n = 5$	83	63	146	9:24:32	1658.42	11.41	1.14	0.92
$m = 5, n = 6$	37	39	46	9:25:20	868.51	5.92	0.61	0.48

优化调度与分布式弹性服务, 解决了边缘计算任务分配不均衡问题. 下一步, 一个重要的研究方向是讨论 MEC 优化调度任务的实时性.

References

- Wang Xiao, Yao Ting-Ting, Han Shuang-Shuang, Cao Dong-Pu, Wang Fei-Yue. Parallel internet of vehicles: The ACP-based networked management and control for intelligent vehicles. *Acta Automatica Sinica*, 2018, **44**(8): 1391-1404 (王晓, 要婷婷, 韩双双, 曹东璞, 王飞跃. 平行车联网: 基于 ACP 的智能车辆网联管理与控制. *自动化学报*, 2018, **44**(8): 1391-1404)
- Zhou Yue-Zhi, Zhang Di. Near-end cloud computing: Opportunities and challenges in the post-cloud computing era. *Chinese Journal of Computers*, 2019, **42**(4): 677-700 (周悦芝, 张迪. 近端云计算: 后云计算时代的机遇与挑战. *计算机学报*, 2019, **42**(4): 677-700)
- Xia Yuan-Qing. Cloud control systems and their challenges. *Acta Automatica Sinica*, 2016, **42**(1): 1-12 (夏元清. 云控制系统及其面临的挑战. *自动化学报*, 2016, **42**(1): 1-12)
- Xu W B, Wang S, Yan S, He J H. An efficient wideband spectrum sensing algorithm for unmanned aerial vehicle communication networks. *IEEE Internet of Things Journal*, 2019, **6**(2): 1768-1780
- Yang B R, Wu D P, Wang R Y. CUE: An intelligent edge computing framework. *IEEE Network*, 2019, **33**(3): 18-25
- Zhou F H, Wu Y P, Hu R Q Y, Qian Y. Computation rate maximization in UAV-enabled wireless powered mobile-edge computing systems. *IEEE Journal on Selected Areas in Communications*, 2018, **36**(9): 1927-1941
- Ahmed A, Ahmed E. A survey on mobile edge computing. In: Proceedings of the 10th IEEE International Conference on Intelligent Systems and Control. Coimbatore, India: IEEE, 2016. 1-8
- Wang Fei-Yue, Zhang Jun, Zhang Jun, Wang Xiao. Industrial internet of minds: Concept, technology and application. *Acta Automatica Sinica*, 2018, **44**(9): 1606-1617 (王飞跃, 张军, 张俊, 王晓. 工业物联网: 基本概念、关键技术与核心应用. *自动化学报*, 2018, **44**(9): 1606-1617)
- Wang Fei-Yue, Zhang Jun. Industrial internet of minds: The concept, issues and platforms. *Acta Automatica Sinica*, 2017, **43**(12): 2061-2070 (王飞跃, 张俊. 物联网: 概念、问题和平台. *自动化学报*, 2017, **43**(12): 2061-2070)
- Zhang K, Mao Y M, Leng S P, He Y J, Zhang Y. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Vehicular Technology Magazine*, 2017, **12**(2): 36-44
- Shao C X, Leng S P, Zhang Y, Vinel A, Jonsson M. Performance analysis of connectivity probability and connectivity-aware MAC protocol design for platoon-based VANETs. *IEEE Transactions on Vehicular Technology*, 2015, **64**(12): 5596-5609
- Gill S S, Buyya R. Secure: Self-protection approach in cloud resource management. *IEEE Cloud Computing*, 2018, **5**(1): 60-72
- Dai Y Y, Xu D, Maharjan S, Zhang Y. Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 2019, **6**(3): 4377-4387
- Chen X, Jiao L, Li W Z, Fu X M. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 2016, **24**(5): 2795-2808
- Nunna S, Kousaridas A, Ibrahim M, Dillinger M. Enabling real-

- time context-aware collaboration through 5G and mobile edge computing. In: Proceedings of the 12th International Conference on Information Technology — New Generations. Las Vegas, USA: IEEE, 2015. 601–605
- 16 Yu R, Huang X M, Kang J W, Ding J F, Maharjan S, Gjessing S, et al. Cooperative resource management in cloud-enabled vehicular networks. *IEEE Transactions on Industrial Electronics*, 2015, **62**(12): 7938–7951
- 17 Liu Ye, Liu Lin-Feng, Zheng Long, Wang Hua-Feng. Study on the downlink performance of roadside unit in vehicular ad-hoc networks. *Journal of Software*, 2015, **26**(7): 1700–1710
(刘业, 刘林峰, 郑隆, 王华锋. 车联网 RSU 单元下行流量的性能研究. 软件学报, 2015, **26**(7): 1700–1710)
- 18 Jiang Yan, Wang Qi, Gong Jian-Wei, Chen Hui-Yan. Research on temporal consistency and robustness in local planning of intelligent vehicles. *Acta Automatica Sinica*, 2015, **41**(3): 518–527
(姜岩, 王琦, 龚建伟, 陈慧岩. 无人驾驶车辆局部路径规划的时间一致性与鲁棒性研究. 自动化学报, 2015, **41**(3): 518–527)
- 19 Guo Chun-Zhao, Yamabe Takayuki, Mita Seiichi. Drivable road boundary detection for intelligent vehicles based on stereovision with plane-induced homography. *Acta Automatica Sinica*, 2013, **39**(4): 371–380
(郭春钊, 山部尚孝, 三田诚一. 基于立体视觉平面单应性的智能车辆可行驶道路边界检测. 自动化学报, 2013, **39**(4): 371–380)
- 20 Zhu H, Yuen K V, Mihaylova L, Leung H. Overview of environment perception for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2017, **18**(10): 2584–2601
- 21 Love J, Jariyasumant J, Pereira E T, Zennaro M, Hedrick K, Kirsch C, et al. CSL: A language to specify and re-specify mobile sensor network behaviors. In: Proceedings of the 15th IEEE Real-time and Embedded Technology and Applications Symposium. San Francisco, USA: IEEE, 2009. 67–76
- 22 Sasaki K, Suzuki N, Makido S, Nakao A. Vehicle control system coordinated between cloud and mobile edge computing. In: Proceedings of the 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). Tsukuba, Japan: IEEE, 2016. 1122–1127
- 23 Li Li, Wang Fei-Yue. Ground traffic control in the past century and its future perspective. *Acta Automatica Sinica*, 2018, **44**(4): 577–583
(李力, 王飞跃. 地面交通控制的百年回顾和未来展望. 自动化学报, 2018, **44**(4): 577–583)
- 24 Li Li, Wang Fei-Yue, Zheng Nan-Ning, Zhang Yi. Research and developments of intelligent driving behavior analysis. *Acta Automatica Sinica*, 2007, **33**(10): 1014–1022
(李力, 王飞跃, 郑南宁, 张毅. 驾驶行为智能分析的研究与发展. 自动化学报, 2007, **33**(10): 1014–1022)
- 25 Liang D. Connected vehicle technology: Beyond smart driving — A discussion with scott McCormick, president of connected vehicle trade association. *IEEE Engineering Management Review*, 2018, **46**(1): 34–36
- 26 Boeing G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers Environment and Urban Systems*, 2017, **65**: 126–139
- 27 Krainer C, Kirsch C M. Cyber-physical cloud computing implemented as PaaS. In: Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-physical Systems. New York, USA: ACM, 2014. 15–18
- 28 Craciunas S S, Haas A, Kirsch C M, Payer H, Ock H, Rottmann A, et al. Information-acquisition-as-a-service for cyber-physical cloud computing. In: Proceedings of the 2th Usenix Conference on Hot Topics in Cloud Computing. Boston, USA: USENIX Association Berkeley, 2010. 14–19
- 29 Huang J C. From the Real Vehicle to the Virtual Vehicle [Ph. D. dissertation], University of California, America, 2013
- 30 Kirsch C M, Pereira E, Sengupta R, Chen H, Hansen R, Huang J C, et al. Cyber-physical cloud computing: The binding and migration problem. In: Proceedings of the 2012 Design, Automation and Test in Europe Conference and Exhibition (DATE). Dresden, Germany: IEEE, 2012. 1425–1428
- 31 Du B W, Huang R H, Xie Z P, Ma J H, Lv W F. KID model-driven things-edge-cloud computing paradigm for traffic data as a service. *IEEE Network*, 2018, **32**(1): 34–41
- 32 Plotkin G D. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 2004, **60**(1): 17–139



张珂 北京工业大学信息学部控制科学与工程专业硕士研究生. 2017 年获得河南理工大学自动化专业学士学位. 主要研究方向为智能交通系统, 移动边缘计算.

E-mail: zhgke766@126.com

(ZHANG Ke Master student in the control science and engineering, at Faculty of Information Technology, Beijing University of Technology. He received his bachelor degree from Henan Polytechnic University in 2017. His research interest covers intelligent transportation systems, mobile edge computing.)



张利国 北京工业大学信息学部控制科学与工程专业教授. 2011–2012 年国家公派美国加州大学伯克利分校访问学者. 主要研究方向为混杂系统, 智能交通系统和分布式参数控制系统. 本文通信作者.

E-mail: zhangliguo@bjut.edu.cn

(ZHANG Li-Guo Professor in the control science and engineering, at Faculty of Information Technology, Beijing University of Technology. He was a visiting scholar (2011–2012) of University of California, Berkeley. His research interest covers hybrid systems, intelligent transportation systems, and control of distributed parameter systems. Corresponding author of this paper.)