

## V2X 多节点协同分布式卸载策略

曹敦<sup>1,2</sup>, 张应宝<sup>1</sup>, 邹电<sup>1</sup>, 王进<sup>1</sup>, 汤强<sup>1</sup>, 冀保峰<sup>3</sup>

(1. 长沙理工大学计算机与通信工程学院, 湖南 长沙 410114;  
2. 南京邮电大学宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003; 3. 河南科技大学信息工程学院, 河南 洛阳 471023)

**摘要:** 为了应对车联网中计算资源密集、可分离型任务的卸载环境动态变化和不同协同节点通信、计算资源存在差异的问题, 提出了一种在 V2X 下多协同节点串行卸载、并行计算的分布式卸载策略。该策略利用车辆可预测的行驶轨迹, 对任务进行不等拆分, 分布式计算于本地、MEC 及协同车辆, 建立系统时延最小化的优化问题。为求解该优化问题, 设计了博弈论的卸载机制, 以实现协同节点串行卸载的执行顺序; 鉴于车联网的动态时变特性, 利用序列二次规划算法, 给出了最优的任务不等拆分。仿真结果表明, 所提策略能够有效减少计算任务系统时延, 且当多协同节点分布式卸载服务时, 所提策略在不同的参数条件下仍然能够保持稳定的系统性能。

**关键词:** 边缘计算; 可预测轨迹; 任务不等拆分; 分布式卸载

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022029

## Multi-node cooperative distributed offloading strategy in V2X scenario

CAO Dun<sup>1,2</sup>, ZHANG Yingbao<sup>1</sup>, ZOU Dian<sup>1</sup>, WANG Jin<sup>1</sup>, TANG Qiang<sup>1</sup>, JI Baofeng<sup>3</sup>

1. School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China  
2. Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China  
3. School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

**Abstract:** In order to cope with the dynamic changes of the offloading environment for computing resource-intensive and separable tasks in Internet of vehicle and deal with the problem that different collaborative nodes had different communication and computing resources, a distributed offloading strategy that multiple collaborative nodes had serial offloading mode and parallel computing mode in vehicle to everything (V2X) scenario was proposed. Utilizing the predictable motion trajectories of vehicle, the tasks were split into unequal parts, finally each part was computed on itself, mobile edge server, and vehicles in parallel. Then an optimization problem of the system time delay minimization was established. To solve the optimization problem, an offloading scheme based on the game theory was designed to determine the serial offloading execution order of the cooperative nodes. Considering the dynamic characteristics of Internet of vehicles, a sequential quadratic programming (SQP) algorithm was adopted to optimally split tasks. Finally, the simulation results show that the proposed strategy can effectively reduce system delay, and when multiple cooperative nodes offload in parallel, the proposed strategy can still maintain the stable system performance under the different parameter conditions.

**Keywords:** edge computing, predictable trajectory, unequal splitting of tasks, distributed offloading

收稿日期: 2021-10-26; 修回日期: 2022-01-11

基金项目: 国家自然科学基金资助项目 (No.61902041, No.61801170); 湖南省自然科学基金资助项目 (No.2021JJ30736); 长沙市自然科学基金资助项目 (No.kq2014112); 南京邮电大学宽带无线通信与传感网技术教育部重点实验室开放研究基金资助项目 (No.JZNY202102)

**Foundation Items:** The National Natural Science Foundation of China (No.61902041, No.61801170), The Natural Science Foundation of Hunan Province (No.2021JJ30736), The Natural Science Foundation of Changsha (No.kq2014112), The Open Research Fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications (No.JZNY202102)

## 0 引言

在这个万物互联的时代,人工智能和通信技术的快速发展能实现车辆与 X(即车、人、网络、基础设施等)的通信<sup>[1]</sup>,并提升车辆的自动驾驶能力,改善交通运营和服务智能化欠缺的现状。同时新型车载终端的应用也对通信质量和智能化要求日益严苛<sup>[2-3]</sup>。移动边缘计算(MEC, mobile edge computing)的出现能够有效地应对这一挑战。MEC是一种新型通信架构,将具有存储、计算、通信功能的服务平台部署在网络边缘位置,帮助移动终端用户将服务任务就近卸载到边缘节点上,进行协同处理<sup>[4-5]</sup>,从而减轻通信链路的传输压力,缩短服务的响应时间,减少移动设备的能耗和传输成本<sup>[6-7]</sup>。MEC技术作为5G移动通信中开始应用的一项重要技术,目前受到工业界和学术界的广泛关注<sup>[8]</sup>。而其在车联网中的应用能提供低时延的计算智能,使用户服务体验<sup>[9-10]</sup>有质的改变。

车联网服务类型可分为车载安全型、车载增强型和车载娱乐型三大类<sup>[11]</sup>。服务对时延、数据量、任务复杂度和子任务间耦合度等有不同的要求<sup>[12]</sup>。根据特征的不同,又可将任务分成不同的类型,如根据时延限制可分为时延敏感型和非时延敏感型,根据任务复杂度可分为计算资源密集型和稀疏型,根据子任务的耦合度分为可分离型和不可分离型<sup>[13]</sup>。时延敏感型的计算任务对数据的实时性要求极高,而计算密集型任务需要大量的存储资源,但时延要求相对宽松<sup>[14]</sup>。依据 Robertazzi 的可分离任务理论<sup>[12]</sup>,可分离型任务可分成若干任意大小的子任务,每个子任务之间没有依赖关系,可以按照任意顺序进行处理。而车载娱乐型服务中的视觉视频剪辑和动画渲染等任务具有非时延敏感、数据量大、复杂度高、耦合度低的特点。在 V2X(vehicle to everything)通信中,利用下沉到车辆及路侧单元的计算资源设计一种任务可拆分、多节点协同的卸载决策,可满足此类服务的服务质量(QoS, quality of service)要求。

然而,任务可拆分、多节点协同的卸载决策需面对一系列问题。在任务执行周期内,车载任务卸载环境是动态不确定的,网络拓扑结构及无线信道状态会快速变化,如何选择在任务执行周期内可用的协同节点是需要解决的问题之一,且考虑协同节点通信和计算资源差异,如何进行任务不等拆分,

使任务分布式计算执行时延最小化,以提高信道利用率是需解决的问题之二。本文面对上述 2 个问题,提出了一种 V2X 多节点协同分布式卸载策略。本文主要的研究工作总结如下。

1) 本文构建了 V2X 下本地、MEC 和协同车辆多节点协同计算模型,它联合了卸载策略和任务不等拆分问题,通过预测车辆行驶轨迹,将协同卸载问题建模成本地与协同节点分布式计算时间较大值最小化问题。

2) 本文采用博弈论中的纳什均衡来求解上述问题中服务的卸载策略,并将串行卸载问题公式化为带约束的高维非线性优化问题。

3) 在卸载策略确定的情况下,本文采用序列二次规划(SQP, sequential quadratic programming)算法求解上述优化问题,通过拉格朗日函数将原问题转化为二次规划子问题,从而确定任务的不等拆分比例,实现本地与卸载的工作时间均衡。

4) 实验结果表明,本文提出的算法具有良好的效能和优越性。与其他 4 种基准算法相比,本文提出的算法能够获得更低的系统时延,并具有良好的收敛性。

## 1 相关工作

国内外学者针对移动边缘计算的任务卸载问题开展了大量的研究。文献[15]研究了静止状态下的用户卸载,考虑到 MEC 的车载无线网络和计算能力有限的问题,设计了一种无线和计算分配联合优化算法用来求解用户的卸载决策。文献[16]提出了 LBB(linearization based branch and bound)和 CRI(closest rounding integer)算法分别解决在用户静止和速度恒定状态下的计算卸载策略,考虑到时变信道对任务卸载策略的影响,该研究提出了一种最接近四舍五入算法来解决固定时变频谱效率问题。文献[17]提出了在加速度不变的条件下多用户计算卸载决策,考虑到多个用户设备同时通过无线信道卸载计算任务时的信道干扰问题,设计了一种基于强化学习的进化博弈算法。文献[18]研究了车辆无线网络的移动感知计算卸载,考虑到车辆的随机移动和卸载过程中可能出现的越区切换问题,该研究提出了一种分析模型用来计算卸载决策。文献[19]提出了一种通过建立适用于移动和普通计算场景的三层体系架构来优化用户计算卸载决策的方法。

文献[20]研究了服务请求车辆发送不可拆分数

据包时的平均上行局部时延, 该研究利用最接近接收方模型确定单个边缘节点。文献[21]研究了每个用户通过无线多址传输将计算任务卸载到多用户协同计算, 考虑到非正交多址接入 (NOMA, non-orthogonal multiple access) 的共信道干扰和 MEC 的计算任务均等拆分问题, 该研究提出了联合最优任务卸载和资源分配的算法求解卸载决策。文献[22]通过研究多用户多输入多输出预编码和计算资源不等分配方法, 提出了一种利用半定松弛法和舍入法优化卸载决策。文献[23]提出了基于多臂强盗理论进行任务的不等拆分策略。文献[24]提出了在动态环境下基于自适应学习的多车辆等分任务卸载策略, 该研究以最小化平均卸载时延为目标, 提出了一种基于 V2V 通信的多车任务卸载系统模型。

综上所述, 当前已有的相关研究集中于以下 2 个方面: 1) 通过建立车辆的不同运动模型, 优化用户的卸载决策问题; 2) 根据服务任务类型的差异, 优化卸载决策问题。当前关于协同卸载策略的研究工作虽然很多, 但是聚集在车辆行驶轨迹预测下的不等任务拆分策略还较少。本文研究了 V2X 多节点协同串行卸载、分布式计算策略, 联合优化了卸载决策和任务不等分配比例。

## 2 系统模型

本文主要考虑高速直行道路场景。因反向行驶车辆相对宿主车辆有较大反向相对速度, 停留在宿主车辆通信范围的时间较短, 因此协同车辆仅考虑与宿主车辆同向行驶的车辆, 并且考虑车辆变道、变加速等行驶行为。为充分利用计算资源和减少卸载计算时延, 宿主车辆可以选择部分任务在本地执行, 其余任务利用 V2X 通信模式卸载到协同节点上协助处理。

### 2.1 网络模型

网络模型如图 1 所示。可提供边缘计算的路侧单元 (RSU, road side unit) 标记为  $v_0$ , 假设 RSU 覆盖范围内有  $N$  辆车呈泊松分布, 表示为  $V = \{v_1, v_2, \dots, v_i, \dots, v_N\}, i \in [1, N]$ 。假设车辆内都装有北斗卫星导航系统等定位设备, 可以实时获取车辆的轨迹信息<sup>[25]</sup>。集合  $V$  中车辆终端的轨迹信息用集合  $S$  表示,  $S = \{p_1^t, p_2^t, \dots, p_\alpha^t, \dots, p_N^t\}, \alpha \in [1, N]$ , 第  $\alpha$  辆车在第  $t$  时刻的轨迹信息为  $p_\alpha^t = (x_\alpha^t, y_\alpha^t, u_\alpha^t)$ , 其中  $u_\alpha^t$  为速度,  $s_\alpha^t = (x_\alpha^t, y_\alpha^t)$  为第  $\alpha$  辆车的位置坐标。当车辆  $v_i$  为宿主车辆时, 在卸载时延和通信范围的约束下,  $v_i$  有  $n \in [0, N]$  个可选择的协同节点  $M_i = \{m_{i,0}, m_{i,1}, \dots, m_{i,j}, \dots, m_{i,n}\}, (i \neq 0, j \neq i, j \in [0, n])$ , 其中,  $m_{i,j}$  表示宿主车辆  $v_i$  的第  $j$  个协同节点,  $m_{i,0} = v_0$  表示 RSU 可用。协同节点的卸载顺序为  $Q_{i,j} = \{q_{i,0}, q_{i,1}, \dots, q_{i,j}, \dots, q_{i,n}\}, (i \neq 0, j \neq i, j \in [0, n])$ , 其中,  $q_{i,j}$  表示  $v_i$  卸载到  $m_{i,j}$  的卸载顺序数,  $q_{i,j} \in \{1, 2, \dots, n+1\}$ , 例如  $q_{i,j}=4$  表示  $m_{i,j}$  为第 4 个执行卸载的协同节点。宿主车辆  $v_i$  的协同策略可通过协同节点集  $M_i$  根据  $Q_{i,j}$  排序后获得, 表示为  $A_i = \text{rank}(M_i, Q_{i,j})$ ,  $\text{rank}(\cdot)$  表示将  $M_i$  中元素  $m_{i,j}$  按  $Q_{i,j}$  中对应元素  $q_{i,j}$  的值升序排列。

为了便于阅读, 本文中主要变量及含义如表 1 所示。

### 2.2 通信模型

本文各节点采用正交频分多址 (OFDMA, orthogonality frequency division multiple access) 技术接入系统。任务车辆在执行边缘卸载时, 通过 V2R (vehicle to RSU)、V2V (vehicle to vehicle) 通信模式将计算任务卸载到路侧单元和协同车辆。本文假设

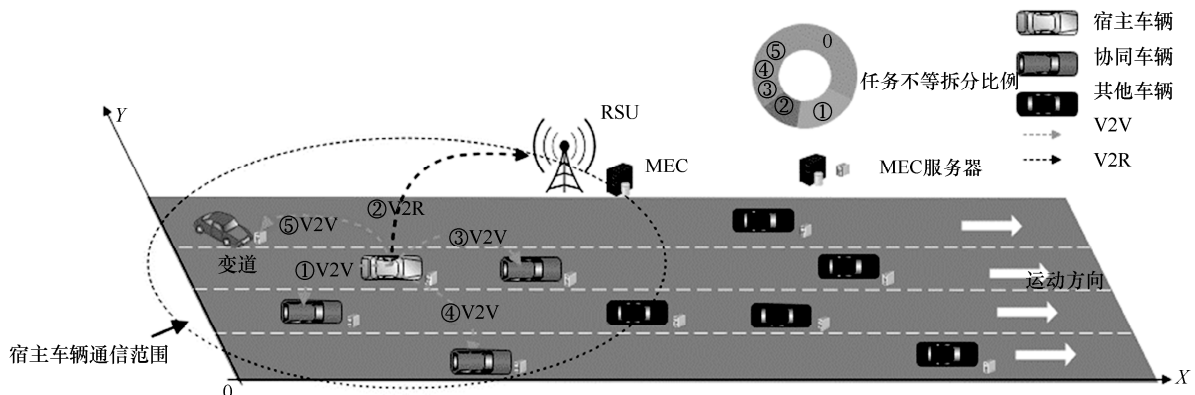


图 1 V2X 多节点协同分布式卸载策略网络模型

多协同节点干扰已通过正交频分多址技术消除,且车辆在任务上传时间内无线信道状态保持稳定<sup>[23]</sup>。不失一般性,假设 V2V 通信模式复用 V2R 通信模式的上行传输通道。根据香农定律可得,数据的上传速率为

$$r_{i,j} = w_* \text{lb} \left[ 1 + \frac{p_1^2 \left( \frac{1}{l_{i,j}^2} \right)^{p_2} P}{w_* \rho_0} \right] \quad (1)$$

其中, \* 可为 v 或 r,  $w_v$  和  $w_r$  分别表示 V2V 和 V2R 信道带宽;  $p_1$  表示上传链路信道衰落因子;  $p_2$  表示路径损耗因子;  $l_{i,j}$  表示从  $v_i$  到  $m_{i,j}$  的欧氏距离;  $\rho_0$  表示通信内部高斯噪声密度;  $P$  为发射功率。

表 1 主要变量及其含义

变量	含义
$N$	RSU 覆盖范围内的车辆数
$v_i$	第 $i$ 辆车
$n$	宿主车辆通信范围内车辆数
$x_\alpha$	车辆 $v_\alpha$ 对应二维坐标中的横坐标值
$y_\alpha$	车辆 $v_\alpha$ 对应二维坐标中的纵坐标值
$f_{i,j}^{\text{off}}$	协同节点 $m_{i,j}$ 的计算能力
$D_{i,j}^{\text{off}}$	宿主车辆 $v_i$ 卸载给协同节点 $m_{i,j}$ 的任务量
$f_i^{\text{comp}}$	宿主车辆 $v_i$ 的本地计算能力
$D_i^{\text{comp}}$	宿主车辆 $v_i$ 的本地计算任务量
$D_i^{\text{total}}$	宿主车辆执行的总任务量
$T_i^{\text{comp}}$	宿主车辆 $v_i$ 本地计算时延
$T_i^{\text{off}}$	宿主车辆 $v_i$ 总卸载计算时延
$\mu$	通信过程中重叠因子
$w_r$	V2R 通信模式信道带宽
$w_v$	V2V 通信模式信道带宽
$P$	车辆的发射功率
$p_1$	车辆上传时信道衰落因子
$p_2$	车辆上传时路径损耗因子
$m_{i,j}$	宿主车辆 $v_i$ 的第 $j$ 个协同节点
$r_{i,j}$	车辆 $v_i$ 到 $m_{i,j}$ 的上行传输速率
$\beta$	权重因子
$W$	车道道宽
$v_0$	路侧单元
$\partial_k$	搜索步长
$S^k$	搜索方向
$R$	车辆的通信范围

### 2.3 运动模型

本文构建行驶轨迹预测模型,通过  $t$  时刻的位置信息  $p_\alpha^t$  预测  $t + \Delta t$  时刻的位置信息  $p_\alpha^{t+\Delta t}$ 。假设本文

的行驶轨迹预测模型目标是估计用户短时间  $\Delta t$  内的移动轨迹。为方便描述,依据车辆行驶方向平行道路建立二维坐标,第  $\alpha$  辆车在  $t + \Delta t$  时刻的二维坐标为

$$\begin{cases} x_\alpha^{t+\Delta t} = x_\alpha^t + u_\alpha^t \Delta t + \frac{1}{2} \frac{du_\alpha^t}{dt} (\Delta t)^2, \alpha \neq 0 \\ y_\alpha^{t+\Delta t} = \text{int}[y_\alpha^t \pm \text{rand}(y_\alpha^t)]W \end{cases} \quad (2)$$

其中,  $\frac{du_\alpha^t}{dt}$  表示为第  $\alpha$  辆车在  $t$  时刻的加速度;  $\text{rand}(\cdot) \in \{0,1\}$  函数用来产生随机整数,即  $\Delta t$  时间内最多只允许变换一次车道;  $W$  表示车道宽度。

由 L2 范数可获得第  $\alpha$ 、 $\beta$  辆车在  $t + \Delta t$  时刻的距离为

$$l_{\alpha,\beta}^{t+\Delta t} = \left\| s_\alpha^{t+\Delta t}, s_\beta^{t+\Delta t} \right\|_2, (\alpha \neq \beta, \alpha, \beta \in N) \quad (3)$$

即

$$l_{\alpha,\beta}^{t+\Delta t} = \sqrt{(x_\alpha^{t+\Delta t} - x_\beta^{t+\Delta t})^2 + (y_\alpha^{t+\Delta t} - y_\beta^{t+\Delta t})^2} \quad (4)$$

### 2.4 计算模型

定义协同节点  $m_{i,j}$  被分配的计算任务为  $Z_{i,j} = \{D_{i,j}, q_{i,j}, t_i^{\text{max}}\}$ , 其中,  $D_{i,j}$  表示任务量大小,  $q_{i,j}$  表示任务  $D_{i,j}$  从  $v_i$  卸载到  $m_{i,j}$  的顺序标签值,  $t_i^{\text{max}}$  表示任务  $Z_{i,j}$  所能容忍的最大时延。一些计算任务具有较高的复杂性,无法及时得到计算结果。因此,需要决策  $Z_{i,j}$  中在本地及卸载计算的子任务量。

#### 1) 本地计算

假设宿主车辆  $v_i$  的本地计算能力为  $f_i^{\text{comp}}$ , 本地计算任务量为  $D_i^{\text{comp}}$  时, 本地计算时延为

$$T_i^{\text{comp}} = \frac{D_i^{\text{comp}}}{f_i^{\text{comp}}} \quad (5)$$

#### 2) 卸载计算

卸载计算时延包括传输时延、计算时延和回传时延,因任务完成结果的大小远小于任务的大小,并且考虑到传输的下行(回传)速度远大于上行(传输)速度,所以本文忽略结果回传的时延大小<sup>[22,26]</sup>,仅考虑上行传输时延和计算时延两部分的影响。

协同节点  $m_{i,j}$  的卸载计算时延为

$$T_{i,j}^{\text{off}} = \frac{\mu D_{i,j}^{\text{off}}}{r_{i,j}} + \frac{D_{i,j}^{\text{off}}}{f_{i,j}^{\text{off}}} \quad (6)$$

其中,  $D_{i,j}^{\text{off}}$  为宿主车辆  $v_i$  向  $m_{i,j}$  分配的子任务量;  $r_{i,j}$  为宿主车辆  $v_i$  传输任务到  $m_{i,j}$  的上传速率;  $\mu$  为通信过程中的重叠因子, 表示被执行的卸载到边缘节点数据比例, 若数据需要完全卸载才能执行, 则  $\mu=1$ 。V2V 通信中定义从宿主车辆到协同节点车辆的信号传输为上行传输, 而协同车辆将计算任务结果回传到宿主车辆的过程视为下行传输。

本文研究的是非时延敏感型任务的协同计算, 并考虑资源竞争的公平性, 宿主车辆使用单信道进行任务串行卸载, 同时为提高用户服务质量及无线资源的利用率、最小化系统时延, 设计各协同节点接受任务后分布式执行。在卸载过程中, 上行传输时间与执行时间存在一段重叠时间。因此卸载部分所用的时间  $T_i^{\text{off}}$  取决于所有协同节点上传时延之和, 再加最后被卸载协同节点的计算时延, 即

$$T_i^{\text{off}} = \sum_{j=0}^n \frac{\mu D_{i,j}^{\text{off}}}{r_{i,j}} + \frac{D_{i,n}^{\text{off}}}{f_{i,n}^{\text{off}}} \quad (7)$$

总任务量  $D_i^{\text{total}}$  被分成  $n+2$  份大小不等的子数据任务, 即  $D_i^{\text{total}} = D_i^{\text{comp}} + \sum_{j=0}^n D_{i,j}^{\text{off}}$ 。

## 2.5 问题形成

车联网中利用 MEC 进行卸载计算, 为提高通信及计算资源的利用率, 期望最小化任务卸载的计算时延。但任务卸载计算期间, 网络拓扑结构及信道质量因车辆的移动发生变化。因此, 需预测车辆轨迹, 选择时延容限内可用的协同节点, 并确定拆分任务方案、卸载节点及任务卸载顺序。需解决的串行卸载、并行计算问题为

$$\begin{aligned} \min_{A_i, D_i} T_{\text{vec}}(A_i, D_i) &= \min_{A_i, D_i} \{ \max(T_i^{\text{comp}}, T_i^{\text{off}}) \} = \\ \min_{A_i, D_i} \left\{ \max \left[ \frac{D_i^{\text{comp}}}{f_i^{\text{comp}}}, \sum_{j=0, j \neq i}^n \frac{\mu D_{i,j}^{\text{off}}}{r_{i,j}} + \frac{D_{i,n}^{\text{off}}}{f_{i,n}^{\text{off}}} \right] \right\} \\ \text{s.t. } C_1: & 0 < D_i^{\text{comp}}, 0 < D_{i,j}^{\text{off}}, \forall i, j \in N \\ C_2: & |l_{i,j}^{\Delta t}| \leq R, \forall i, j \in N, \Delta t \in [t, t + t_i^{\text{max}}] \\ C_3: & D_i^{\text{comp}} + \sum_{j=0}^n D_{i,j}^{\text{off}} = D_i^{\text{total}}, \forall i, j \in N \\ C_4: & \frac{D_{i,j}^{\text{off}}}{f_{i,j}^{\text{off}}} \leq \sum_{m=j+1}^n \frac{\mu D_{i,m}^{\text{off}}}{r_{i,m}} + \frac{D_{i,n}^{\text{off}}}{f_{i,n}^{\text{off}}}, \forall i, j, m \in N \end{aligned} \quad (8)$$

因为任务在本地和协同节点分布式执行, 所以

任务执行总时长取决于本地或协同节点任务执行时长的较大值。约束条件  $C_1$  表示任务在本地和协同节点分布式执行的约束边界;  $C_2$  表示在任务执行期  $t + t_i^{\text{max}}$  内  $v_i$  到  $m_{i,j}$  之间的相对距离小于或等于通信范围  $R$ ;  $C_3$  表示本地和卸载计算任务量之和等于总任务量  $D_i^{\text{total}}$ ;  $C_4$  表示当前协同节点  $m_{i,j}$  执行卸载任务的计算时延必须小于还未分配计算任务的协同卸载计算时延之和, 其目的在于保证最后一个协同节点将子任务的计算结果回传到宿主车辆后, 其余协同节点的计算结果均已完成回传。

## 3 算法设计

为了解决式(8)描述的优化问题, 本文提出了一种协同分布式卸载-SQP 策略 (CDOS-SQP, cooperative distributed offloading strategy-SQP)。在 CDOS-SQP 中, 分别提出了一种基于博弈论的卸载策略和基于 SQP 算法<sup>[27]</sup>的任务分配方法, 前者确定计算任务的卸载策略集  $A_i$ , 而后者根据  $A_i$  确定任务不等拆分集  $D_i$ 。

### 3.1 基于博弈论的卸载机制

本节确定宿主车辆协同卸载时可用的  $m_{i,j}$ , 可通过卸载顺序集  $Q_{i,j}$  获得协同策略集  $A_i$  来解决此问题。

在宿主车辆和协同节点的不断移动下, 确定车辆  $v_i$  为宿主车辆时的协同节点集  $M_i$  的准则是: 在任务执行期内协同车辆始终处于宿主车辆的通信范围  $R$  内。本文应用博弈论中的重复剔除严格劣战略来获得  $M_i$ 。

当宿主车辆无任何可用协同节点时,  $D_i^{\text{total}}$  全部在本地执行,  $m_{i,j}$  的加入不应使任务协同计算时间大于此值, 因此, 选择此情况下的计算时延作为最大时延容限, 表示为

$$t_i^{\text{max}} = \frac{D_i^{\text{total}}}{f_i^{\text{comp}}} \quad (9)$$

使用博弈论中的重复剔除严格劣战略, 将  $t + \Delta t$  时刻车辆间的距离大于通信范围  $R$  定义为劣战略, 终止条件为此时此刻集合  $M_i$  中不包含劣战略。从而确定从  $t$  时刻到  $t + t_i^{\text{max}}$  时刻始终处于  $R$  范围内的协同节点集  $M_i$ 。下面使用博弈论中的纳什均衡来确定最优卸载顺序集  $Q_{i,j}^*$ 。

首先, 定义  $m_{i,j}$  的  $t_{i,j}^{\text{recon}}$  为宿主车辆  $v_i$  卸载到  $m_{i,j}$  的上传时延及在  $m_{i,j}$  的计算时延之和, 即

$$t_{i,j}^{\text{recon}} = \frac{\mu D_{i,j}^{\text{recon}}}{r_{i,j}^{\text{min}}} + \frac{D_{i,j}^{\text{recon}}}{f_{i,j}^{\text{off}}}, (i \in N, j \in n) \quad (10)$$

其中,  $r_{i,j}^{\text{min}}$  表示在  $(t, t + t_i^{\text{max}})$  时间内从  $v_i$  到  $m_{i,j}$  的传输速率最小值;  $t_{i,j}^{\text{recon}}$  值越小,  $m_{i,j}$  越先被选择卸载。

然后, 使用博弈论中的纳什均衡来确定协同节点执行任务的卸载顺序。策略博弈<sup>[28]</sup>  $\Gamma$  涉及 3 个要素, 分别为参与者  $M_i$ 、策略  $Q_i$  以及效用函数  $U_i$ , 具体说明如下。

1) 参与者。宿主车辆  $v_i$  的  $n+1$  个参与博弈的协同节点集  $M_i$ 。

2) 策略。  $Q_i = \times_{j=0}^n Q_{i,j}$ 。其中,  $Q_i$  表示  $v_i$  的协同节点所有可能的串行卸载顺序集合;  $Q_{i,j}$  表示参与者可以选择的一个策略。  $Q_i$  可供选择策略的数量为  $(n+1)!$ ,  $q_{i,j}$  表示协同节点  $m_{i,j}$  选择的策略,  $q_{i,-j}$  表示除了  $m_{i,j}$  之外的  $n$  个策略参与者选择的策略, 则  $q_{i,j}$  和  $q_{i,-j}$  可组合成  $Q_{i,j}$ 。

3) 效用函数。当  $v_i$  的协同节点集  $M_i$  选择其串行卸载顺序集为  $Q_{i,j}$  时的代价函数为  $U_i(q_{i,j}, q_{i,-j})$ , 定义如式(11)所示。每个协同节点都希望选择最优卸载顺序获得最大效用。

$$U_i(q_{i,j}, q_{i,-j}) = \sum_{j=0}^n t_{i,j}^{\text{recon}} q_{i,j} \quad (11)$$

纳什均衡是指博弈中每个参与者都确信在其他参与者策略给定的情况下, 自己选择了最优策略, 从而使自己效用最大化。具体描述介绍如下。

**定义** 若对于  $\forall j \in n$  与  $\forall q_{i,j} \in Q_{i,j}$  而言, 满足条件  $U_i(q_i^*) \geq U_i(q_{i,j}, q_{i,-j}^*)$ , 可将  $q_i^* = (q_{i,j}^*, q_{i,-j}^*)$  组合的策略  $Q_{i,j}^*$  视作策略博弈  $\Gamma = \{M_i, Q_i, U_i\}$  的一个纳什均衡。

**证明** 已知不等拆分的子任务从  $v_i$  卸载到  $m_{i,j}$  的预估时延  $t_{i,j}^{\text{recon}}$  为已知值, 各个协同节点的预估时延大小不全相等,  $q_{i,j}$  为协同卸载执行的顺序标签数。为了使效用函数  $U_i(q_{i,j}, q_{i,-j})$  效益最大化, 当  $Q_{i,j}$  选择的策略集与  $\{t_{i,j}^{\text{recon}}\}$  从小到大的顺序标签值相同时, 存在策略  $Q_{i,j}^*$  为策略博弈  $\Gamma$  的一个纳什均衡。证毕。

通过以上最优策略原则, 计算出  $M_i$  的协同卸载最优顺序集  $Q_{i,j}^*$ 。随后, 联合集合  $M_i$  和  $Q_{i,j}^*$ , 通过  $\text{rank}(M_i, Q_{i,j}^*)$  函数计算出最优的  $A_i$ 。伪代码如算

法 1 所示。

**算法 1** 使用基于博弈论的卸载决策求解卸载策略集  $A_i$

**输入** V2V 或 V2R 通信模式信道带宽  $w_*$ , 车辆的通信范围  $R$ , 全部车辆终端初始轨迹信息集  $S$ , 处理的总任务量  $D_i^{\text{total}}$

**输出** 卸载策略集  $A_i$

1) 根据式(9)计算宿主车辆  $v_i$  的最大时延容限

$$t_i^{\text{max}}$$

2) 初始化集合  $E_i = \emptyset, H = \emptyset, G_i = \emptyset$

3) for  $\Delta t = 0, 1, 2, \dots, t_i^{\text{max}}$  do

4) 根据车辆行驶轨迹模型式(2), 计算宿主车辆  $v_i$  在  $t + \Delta t$  时刻的位置  $s_i^{t+\Delta t} = (x_i^{t+\Delta t}, y_i^{t+\Delta t})$

5) for  $j = 0, 1, 2, \dots, n, j \neq i$  do

6)  $v_j$  在  $t + \Delta t$  时刻的位置为  $s_j^{t+\Delta t}$

7) 计算  $l_{i,j} = \|s_i^{t+\Delta t}, s_j^{t+\Delta t}\|_2$ ,  $l_{i,j}^{\text{max}} = l_{i,0}$

8) 根据重复剔除严格劣战略

if  $l_{i,j} \leq R$  do

9)  $H = H \cup v_j$

10) if  $l_{i,j}^{\text{max}} \leq l_{i,j}$  do

11)  $m_{i,j}$  和  $v_i$  之间的最大距离  $l_{i,j}^{\text{max}} = l_{i,j}$

12) end if

13) end if

14) end for

15)  $t + \Delta t$  时刻的协同节点集  $E_{\Delta t} = H$

16) end for

17)  $M_i = E_0 \cap E_1 \cap \dots \cap E_{t_i^{\text{max}}}$

18) for  $k = 1, 2, 3, \dots, \text{length}(M_i)$  do

19) 由  $l_{i,j}^{\text{max}}$  计算出  $r_{i,j}^{\text{min}}$ , 然后根据式(10)计算

更新预估时延集  $G_i = G_i \cup t_{i,k}^{\text{recon}}$

20) end for

21) 由小到大排序集合  $G_i$  中的元素

22) 依据  $G_i$ , 根据纳什均衡定理计算  $Q_i$  中最优的集合  $Q_{i,j}^*$

23) 使用排序函数  $\text{rank}(M_i, Q_{i,j}^*)$ , 计算卸载策略集  $A_i$

24) return  $A_i$

### 3.2 基于 SQP 算法的任务分配

本节研究协同卸载下任务不等拆分集  $D_i$ 。现已通过卸载机制计算出  $A_i$ , 已知式(8)中的约束条件

$C_2$  仅影响策略集合  $A_i$ , 则在本节计算需确立集合  $D_i$  中任务的不等拆分比例, 使任务执行时延  $T_{\text{vec}}$  最小化时不考虑约束条件  $C_2$ 。由于集合  $D_i$  中任务可在本地和协同节点并行计算, 因此将式(8)转化为本地和卸载这两类并行执行时间差值  $\Delta T_{\text{vec}}$  最小化问题, 即

$$\min_{D_i} \Delta T_{\text{vec}}(D_i) = \min_{D_i} \left( \left| T_i^{\text{comp}} - T_i^{\text{off}} \right| \right) = \min_{D_i} \left[ \left| \frac{D_i^{\text{comp}}}{f_i^{\text{comp}}} - \left( \sum_{j=0}^n \frac{\mu D_{i,j}^{\text{off}}}{r_{i,j}} + \frac{D_{i,n}^{\text{off}}}{f_{i,n}^{\text{off}}} \right) \right| \right]$$

$$\text{s.t. } C_1: 0 < D_i^{\text{comp}}, 0 < D_{i,j}^{\text{off}}, \forall i, j \in N$$

$$C_2: D_i^{\text{comp}} + \sum_{j=0}^n D_{i,j}^{\text{off}} = D_i^{\text{total}}, \forall i, j \in N$$

$$C_3: \frac{D_{i,j}^{\text{off}}}{f_{i,j}^{\text{off}}} \leq \sum_{m=j+1}^n \frac{\mu D_{i,m}^{\text{off}}}{r_{i,m}} + \frac{D_{i,n}^{\text{off}}}{f_{i,n}^{\text{off}}}, \forall i, j, m \in N \quad (12)$$

因为协同策略集  $A_i$  中有多个协同节点, 所以任务不等拆分时按照协同节点数将其拆分成多份不等任务, 即式(12)的优化问题可描述为带约束条件高维非线性优化函数问题。SQP 算法是求解此类最有效的优化算法之一, 具有收敛性较好、边界搜索能力强、计算效率高等优点。本文采用 SQP 算法求解式(12)的优化目标问题。

优化问题的数学模型简化为

$$\min \Delta T_{\text{vec}}(X) \quad X \in \mathbf{R}^n$$

$$\text{s.t. } x_i \in X, 0 < x_i \leq D_i^{\text{total}}, D_i^{\text{total}} \in \mathbf{R}^n$$

$$h(X) = 0$$

$$g(X) \leq 0 \quad (13)$$

其中,  $X$  为  $\mathbf{R}^n$  中的一个非空多面体子集, 即任务不等拆分子集  $D_i$ ;  $\Delta T_{\text{vec}}(X)$  为优化目标函数,  $\Delta T_{\text{vec}}$  越接近 0, 分布式执行程度越高,  $T_{\text{vec}}$  越小; 约束条件中, 优化变量  $X$  取值范围的最大值为  $D_i^{\text{total}}$ , 最小值为 0;  $h(X)$  为式(12)中  $C_2$  描述的等式约束;  $g(X)$  为式(12)中  $C_3$  描述的不等式约束;  $\mathbf{R}^n$  表示  $D_i^{\text{total}}$  数值取自实数空间, 因此可将式(12)中的优化目标函数问题转化成求解式(13), 拉格朗日函数为

$$L(X, \lambda_1, \lambda_2) = \Delta T_{\text{vec}}(X) + \lambda_1 h(X) + \lambda_2 g(X) \quad (14)$$

其中,  $\lambda_1$  和  $\lambda_2$  为约束函数的加权因子。

在  $X^k$  点根据二阶泰勒公式近似展开为

$$L(X^{k+1}, \lambda_1^{k+1}, \lambda_2^{k+2}) = L(X^k, \lambda_1^k, \lambda_2^k) + (\nabla L(X^k, \lambda_1^k, \lambda_2^k))^T S^k + \frac{1}{2} (S^k)^T [\mathbf{B}] S^k \quad (15)$$

其中,  $S^k$  为优化问题的搜索方向;  $[\mathbf{B}]$  为拟牛顿法近似构造的海塞矩阵,  $[\mathbf{B}] = \nabla^2 L(X^k, \lambda_1^k, \lambda_2^k)$ ; 符号  $\nabla$  为梯度。拉格朗日函数的一阶导数为

$$\nabla L(X^k, \lambda_1^k, \lambda_2^k) =$$

$$\nabla(\Delta T_{\text{vec}}(X^k)) + (\nabla h(X^k))^T \lambda_1^k + (\nabla g(X^k))^T \lambda_2^k \quad (16)$$

函数  $g(X)$  在  $X^k$  点展开的二阶泰勒近似式为

$$g(X^{k+1}) = g(X^k) + (\nabla g(X^k))^T S^k = 0 \quad (17)$$

等式约束  $h(X) = 0$  在  $X^k$  点展开的二阶泰勒近似式为

$$h(X^{k+1}) = h(X^k) + (\nabla h(X^k))^T S^k = 0 \quad (18)$$

将式(15)~式(18)联合计算代入式(12), 经过整理得二次规划子问题为

$$\min(\nabla(\Delta T_{\text{vec}}(X^k)))^T S^k + \frac{1}{2} (S^k)^T [\mathbf{B}] S^k$$

$$\text{s.t. } h(X^k) + (\nabla h(X^k))^T S^k = 0$$

$$g(X^k) + (\nabla g(X^k))^T S^k \leq 0 \quad (19)$$

求解上述二次规划子问题<sup>[29]</sup>, 得到搜索方向  $S^k$ , 沿搜索方向进行一维搜索,  $\partial_k$  为第  $k$  次搜索的下降方向  $S^k$  的最优步长, 利用 Wolfe 步长规则获取, 按照式(20)进行迭代更新, 最终得到原问题的最优解为

$$X^{k+1} = X^k + \partial_k S^k \quad (20)$$

其中,  $[\mathbf{B}]$  为 Lagrange 函数的 Hessian 矩阵正定拟牛顿近似, 并通过稠密半定牛顿近似法 BFGS (Broyden-Fletcher-Goldfarb-Shanno) 进行计算, 作为不断更新的校正矩阵, 搜索方向  $S^k = -\mathbf{B}_k^{-1} \nabla L(X^k, \lambda_1^k, \lambda_2^k)$ , 迭代更新矩阵  $\mathbf{B}_k$  是  $\nabla^2 L(X^k, \lambda_1^k, \lambda_2^k)$  的近似, 更新式为

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{S}_k \mathbf{S}_k^T \mathbf{B}_k}{\mathbf{S}_k^T \mathbf{B}_k \mathbf{S}_k} + \frac{\mathbf{Z}_k \mathbf{Z}_k^T}{\mathbf{Z}_k^T \mathbf{S}_k} \quad (21)$$

其中,  $\mathbf{B}_k$  为  $n$  阶对称正定矩阵。

$$\mathbf{Z}_k = \theta_k y_k + (1 - \theta_k) \mathbf{B}_k \mathbf{S}_k$$

$$\mathbf{S}_k = \partial_k S^k$$

$$y_k = \nabla^2 L(X^{k+1}, \lambda_1^{k+1}, \lambda_2^{k+1}) - \nabla^2 L(X^k, \lambda_1^k, \lambda_2^k)$$

$$\theta_k = \begin{cases} 1, \mathbf{S}_k^T \mathbf{y}_k \geq 0.2 \mathbf{S}_k^T \mathbf{B}_k \mathbf{S}_k \\ \frac{0.8 \mathbf{S}_k^T \mathbf{B}_k \mathbf{S}_k}{\mathbf{S}_k^T \mathbf{B}_k \mathbf{S}_k - \mathbf{S}_k^T \mathbf{y}_k}, \mathbf{S}_k^T \mathbf{y}_k < 0.2 \mathbf{S}_k^T \mathbf{B}_k \mathbf{S}_k \end{cases}$$

本文使用 SQP 算法, 利用拟牛顿法(变尺度法)来近似构造海塞(Hessian)矩阵 $[\mathbf{B}]$ , 以建立二次规划子问题式(19), 又称约束变尺度法。通过求解二次规划子问题得到迭代的搜索方向 $\mathbf{S}^k$ , 沿搜索方向进行一维搜索, 找到迭代的步长 $\partial_k$ , 利用式(21)更新修正 $\mathbf{B}_{k+1}$ , 最后按照迭代式(20)最终得到问题的最优解 $D_i$ , 式(14)~式(21)为 SQP 的一般方法。

伪代码如算法 2 所示。

**算法 2** 基于 SQP 算法求解任务不等拆分集 $D_i$

输入 给定初始点参数 $X^0$ , 收敛精度参数 $\varepsilon$

输出 任务不等拆分集 $D_i$

初始化 对称正定矩阵 $\mathbf{S}_0$ , 迭代次数 $k = 0$

- 1) while true do
- 2) 根据式(15)~式(18), 将式(13)简化成序列二次规划问题式(19)
- 3) 利用式(19)求解二次规划子问题, 求出满足子问题的解 $\mathbf{S}_k$ 及其对应的拉格朗日乘子 $\lambda_1^k$ 和 $\lambda_2^k$
- 4) 从点 $X^k$ 沿方向 $\mathbf{S}_k$ 进行一维线性搜索确定步长 $\partial_k$ , 并令 $X^{k+1} = X^k + \partial_k \mathbf{S}^k$
- 5) if  $\Delta T_{\text{vec}}(X^{k+1}) \geq \varepsilon$  do
- 6) 修改 Hessian 矩阵 $[\mathbf{B}]$ , 利用式(21)更新修正 $\mathbf{B}_{k+1}$ , 令 $k = k + 1$
- 7) else
- 8) 将最优向量解 $X^{k+1}$ 赋值给集合 $D_i$ ,  
 $D_i = X^{k+1}$
- 9) break
- 10) end if
- 11) return 集合 $D_i$

通过 SQP 算法, 可以得到协同节点和宿主车辆所分配的任务不等拆分集 $D_i$ , 进一步可以计算出满足条件的最小计算时延。

## 4 仿真结果及性能分析

本节将评估所提算法在车辆移动下, 通过 SQP 算法将任务不等拆分后在分布式协同执行下的性能。使用各种度量来评估性能, 如系统总时延、迭代次数, 本文还研究了算法在不同参数设置下性能的表现。

### 4.1 仿真设置

本文采用 V2X 多节点分布式卸载策略, 定义

的道路场景如图 1 所示。在车辆数一定的情况下, 道路上车辆服从泊松分布, 主要实验参数如表 2 所示。为了获得方法的一般性能, 本文采用蒙特卡罗方法进行了 1 000 次仿真, 获得平均性能以衡量方法的性能。车辆在直道上行驶的最大速度符合高速公路最大安全速度及车辆间安全行驶速度的规定, 初始速度随机在 60 km/h 至最大速度之间选择, 并在单次仿真过程中保持不变。后续实验过程中在没有特殊说明情况下, 均采用表 2 中实验参数<sup>[30]</sup>。

表 2 主要实验参数

参数	取值
道路长度/m	900
宿主车辆的发射功率 P/W	0.2
通信范围 R /m	150
车道道宽 W /m	3
道路车道数	4
重叠因子 $\mu$	[0,1]
车载单元计算能力/GHz	1
RSU 服务器计算能力/GHz	3
高斯白噪声功率 $\rho_0$ /dBm	-114
V2V 通信信道带宽 $w_v$ /MHz	75
V2R 通信信道带宽 $w_r$ /MHz	100
总任务量/MB	50

为了比较说明本文提出的 CDOS-SQP 在协同卸载计算中的性能, 本文分别评估了文献[31]中的各种卸载策略模型, 即 $Q_i^{\text{opt}}$  (本地执行)、 $Q_j^{\text{opt}}$  (完全卸载到协同车辆执行)、 $Q_{i+\text{cloud}}^{\text{opt}}$  (本地和 RSU 分布式协同执行)、 $Q_{j+\text{cloud}}^{\text{opt}}$  (协同车辆和 RSU 分布式协同执行)、 $Q_{i+j+\text{cloud}}^{\text{opt}}$  (本地、协同车辆、RSU 分布式协同执行), 在本文实验参数环境下的性能, 上述对比卸载策略在进行任务不等拆分时均采用 SQP 算法。

### 4.2 仿真结果及分析

本节说明使用上述及本文所提协同卸载模型, 分别改变任务量、上传功率、车辆密度和车辆初始速度范围, 获得系统总时延, 比较分析 CDOS-SQP 的优越性。

车辆总数 $N$ 固定为 25 辆, 任务量与系统时延关系如图 2 所示。随着计算任务量 $D_i^{\text{total}}$ 增加, 本文所提 CDOS-SQP 的系统时延均低于对比策略。在任务量最大时, CDOS-SQP 策略系统时延相对最劣对比策略减少 67%, 相对最优对比策略减少 8%。当计算

任务量较大时, CDOS-SQP 的系统时延显著优于对比策略, 并且随着计算任务量的增加, CDOS-SQP 增长速度相比其他策略较缓。造成上述结果的原因是, 随着计算任务量的不断增加, 系统的最优策略会倾向于在边缘侧和本地联合计算。而 CDOS-SQP 相比对比策略可选择协同节点数最多, 因此 CDOS-SQP 的优势随着计算任务量的增加更加明显。

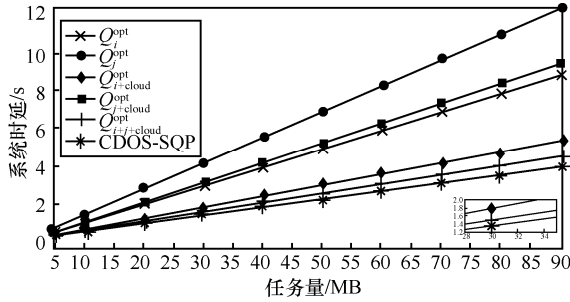


图 2 任务量与系统时延关系

计算任务量固定为 50 MB, 上传功率与系统时延关系如图 3 所示。从图 3 中可以看出,  $Q_i^{opt}$  策略只需在本地计算, 所以上传功率对  $Q_i^{opt}$  的系统时延大小并无影响, 而其余策略的任务卸载执行时延随着上传功率的增大而减小。这是因为宿主车辆上传功率增加会提高信噪比, 从而增加信道容量, 使采用边缘卸载的策略上传时延减小。同时 CDOS-SQP 相比对比策略可用的协同节点数最多, 且多协同节点采用分布式卸载策略并计算任务, 其系统时延是最小的。通过实验对比可知, 本文策略优于对比策略。

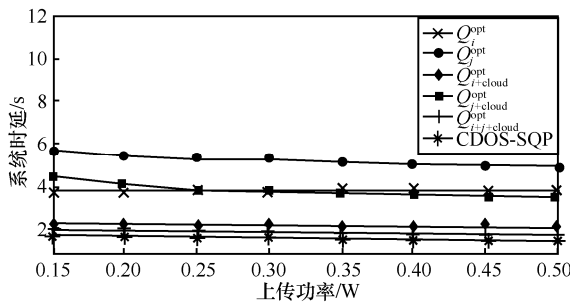


图 3 上传功率与系统时延关系

相同计算任务下车辆密度的变化对系统时延的影响如图 4 所示。从图 4 可知, 除  $Q_i^{opt}$  外, 其余策略随着车辆密度的增加, 系统时延不断减小。造成上述现象的原因是:  $Q_i^{opt}$  策略任务仅在本地图执行, 车辆密度的变化对其没有影响; 而在其他卸载策略中, 随着车辆密度的不断增大, 可用的协同节点也随着增多, 由式(1)~式(6)可知, 节点

间欧氏距离越小, 上传时延也将越小, 因此随车密度增加, 系统总时延减少。而 CDOS-SQP 可用的协同节点最多, 因此表现最优。

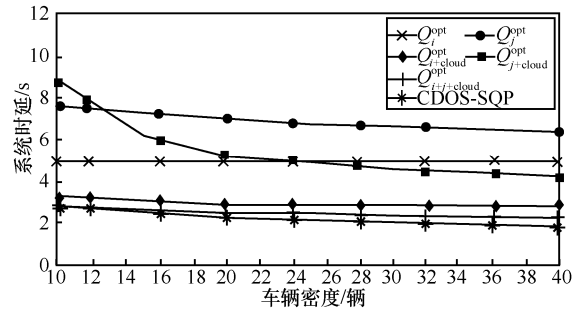


图 4 车辆密度与系统时延关系

车辆密度为 20 辆, 总任务量为 50 MB, 车辆初始速度变化范围与系统时延的关系如图 5 所示。结果表明, 随着初始速度变化范围的增加, 计算任务的系统时延不断增大。从图 5 中可看出, 车辆的运动速度对  $Q_i^{opt}$  没有产生影响, 而对  $Q_j^{opt}$  和  $Q_{j+cloud}^{opt}$  的结果会产生较大的波动。因为当车辆初始速度变化范围越大时, 宿主车辆与协同节点的连通时间将越短, 使可用的协同节点数也将越少。所以对于  $Q_j^{opt}$  和  $Q_{j+cloud}^{opt}$  算法而言, 全部任务只进行卸载计算影响最明显。CDOS-SQP 性能最优的原因是, 同时采用本地和协同节点分布式执行, 并且可利用多协同节点进行并行计算执行。

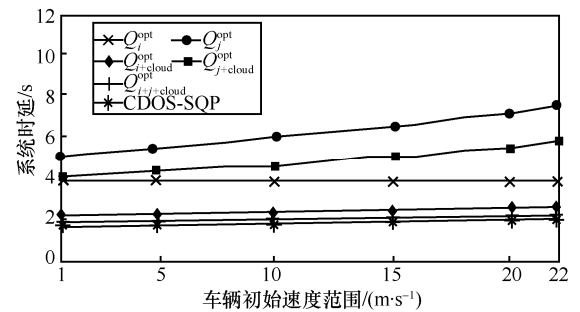


图 5 车辆初始速度变化范围与时延关系

图 6 给出了不同参数条件下迭代次数与收敛性的变化关系, 展示了 SQP 算法在不同参数下解决式(13)所示优化问题的迭代次数。通过图 6 可知, 当迭代 34 次后实验结果趋于平稳, 即趋于目标函数的最优解。随着迭代次数的增加, SQP 算法得到目标函数的值越来越接近满足约束条件的最优解。这表明 SQP 算法解决式(13)所示优化问题, 具有迭代次数较少、收敛性较好的优点。

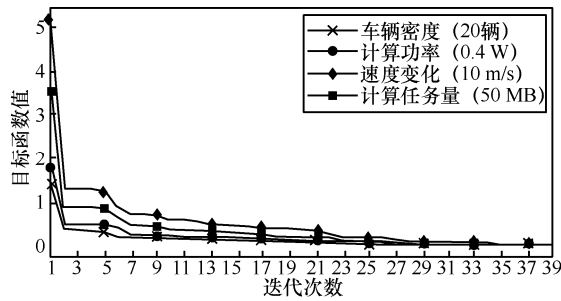


图 6 迭代次数与收敛性的变化关系

## 5 结束语

本文对可拆分任务的 V2X 多节点分布式卸载策略进行研究。考虑车辆行驶轨迹和最大时延约束，建立了基于博弈论的卸载策略和基于 SQP 算法的任务分配模型。利用博弈论中的重复剔除严格劣战略和纳什均衡确定出协同策略集；将原问题转换为带约束的高维非线性优化问题，根据 SQP 算法将该问题转换为求解二次规划子问题，利用拉格朗日和 BFGS 方法得到最优解，并分析了算法的收敛性。仿真结果表明，本文所提 CDOS-SQP 具有较好的收敛性和优越性。下一步的工作将研究更普遍的道路结构（如弯道、十字路口等多结构复杂道路场景）和考虑更精准的行驶轨迹预测的协同卸载问题。

## 参考文献:

[1] WANG S, HUANG J Q, ZHANG X Y. Demystifying millimeter-wave V2X: towards robust and efficient directional connectivity under high mobility[C]//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. Piscataway: IEEE Press, 2020: 1-14.

[2] MOUBAYED A, SHAMI A, HEIDARI P, et al. Edge-enabled V2X service placement for intelligent transportation systems[J]. IEEE Transactions on Mobile Computing, 2021, 20(4): 1380-1392.

[3] 郭辉, 芮兰兰, 高志鹏. 车辆边缘网络中基于多参数 MDP 模型的动态服务迁移策略[J]. 通信学报, 2020, 41(1): 1-14.

GUO H, RUI L L, GAO Z P. Dynamic service migration strategy based on MDP model with multiple parameter in vehicular edge network[J]. Journal on Communications, 2020, 41(1): 1-14.

[4] HE X Y, XING H, CHEN Y, et al. Energy-efficient mobile-edge computation offloading for applications with shared data[C]//Proceedings of 2018 IEEE Global Communications Conference. Piscataway: IEEE Press, 2018: 1-6.

[5] 黄永明, 郑冲, 张征明, 等. 大规模无线网络移动边缘计算和缓存研究[J]. 通信学报, 2021, 42(4): 44-61.

HUANG Y M, ZHENG C, ZHANG Z M, et al. Research on mobile edge computing and caching in massive wireless communication network[J]. Journal on Communications, 2021, 42(4): 44-61.

[6] MA Y, LIANG W F, GUO S. Mobility-aware delay-sensitive service

provisioning for mobile edge computing[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops. Piscataway: IEEE Press, 2019: 270-276.

[7] FENG J, RICHARD Y F, PEI Q Q, et al. Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(7): 6214-6228.

[8] HU X Y, WONG K K, YANG K. Wireless powered cooperation-assisted mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2018, 17(4): 2375-2388.

[9] GAO B, ZHOU Z, LIU F M, et al. An online framework for joint network selection and service placement in mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2021, PP(99): 1.

[10] GAO B, ZHOU Z, LIU F M, et al. Winning at the starting line: joint network selection and service placement for mobile edge computing[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1459-1467.

[11] CAO J, SU X. 5G edge computing enhanced mobile augmented reality[C]//Proceedings of 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). Piscataway: IEEE Press, 2021: 416-417.

[12] CHENG Y C, ROBERTAZZI T G. Distributed computation with communication delay (distributed intelligent sensor networks)[J]. IEEE Transactions on Aerospace and Electronic Systems, 1988, 24(6): 700-712.

[13] KHAN A U R, OTHMAN M, MADANI S A, et al. A survey of mobile cloud computing application models[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 393-413.

[14] XU W C, ZHOU H B, CHENG N, et al. Internet of vehicles in big data era[J]. IEEE/CAA Journal of Automatica Sinica, 2018, 5(1): 19-35.

[15] ZHU M, HOU Y Z, TAO X F, et al. Joint optimal allocation of wireless resource and MEC computation capability in vehicular network[C]//Proceedings of 2020 IEEE Wireless Communications and Networking Conference Workshops. Piscataway: IEEE Press, 2020: 1-6.

[16] LI S C, LIN S Y, CAI L, et al. Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing[J]. IEEE Transactions on Vehicular Technology, 2020, 69(3): 3384-3398.

[17] CUI Y Y, ZHANG D G, ZHANG T, et al. Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices[J]. AEU - International Journal of Electronics and Communications, 2020, 118: 153134.

[18] HUY H V, HO T M, LE L B. Mobility-aware computation offloading in MEC-based vehicular wireless networks[J]. IEEE Communications Letters, 2020, 24(2): 466-469.

[19] CARDELLINI V, DE-NITTO P V, DI-VALERIO V, et al. A game-theoretic approach to computation offloading in mobile cloud computing[J]. Mathematical Programming, 2016, 157(2): 421-449.

[20] WU Y Y, ZHENG J. Analysis of the uplink local delay in an MEC-based VANET[C]//Proceedings of 2019 IEEE Global Communications Conference. Piscataway: IEEE Press, 2019: 1-6.

[21] HANG B Q, ZHANG B L, WANG L, et al. A user association policy for UAV-aided time-varying vehicular networks with MEC[C]//Proceedings of 2020 IEEE Wireless Communications and Networking

Conference. Piscataway: IEEE Press, 2020: 1-6.

- [22] DING C F, WANG J B, ZHANG H, et al. Joint MU-MIMO precoding and resource allocation for mobile-edge computing[J]. IEEE Transactions on Wireless Communications, 2021, 20(3): 1639-1654.
- [23] SUN Y X, GUO X Y, SONG J H, et al. Adaptive learning-based task offloading for vehicular edge computing systems[J]. IEEE Transactions on Vehicular Technology, 2019, 68(4): 3061-3074.
- [24] QIN H, TAN G P, ZHOU S Y, et al. Adaptive learning-based multi-vehicle task offloading[C]//Proceedings of 2020 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2020: 1033-1038.
- [25] CHEN C, LIU L, QIU T, et al. ASGR: an artificial spider-web-based geographic routing in heterogeneous vehicular networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(5): 1604-1620.
- [26] REN J K, YU G D, HE Y H, et al. Collaborative cloud and edge computing for latency minimization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 5031-5044.
- [27] KHAN W U, ALI Z, WAQAS M, et al. Efficient power allocation with individual QoS guarantees in future small-cell networks[J]. AEU - International Journal of Electronics and Communications, 2019, 105: 36-41.
- [28] 张海波, 王子心, 贺晓帆. SDN 和 MEC 架构下 V2X 卸载与资源分配[J]. 通信学报, 2020, 41(1): 114-124.
- ZHANG H B, WANG Z X, HE X F. V2X offloading and resource allocation under SDN and MEC architecture[J]. Journal on Communications, 2020, 41(1): 114-124.
- [29] BOGGS P T, TOLLE J W. Sequential quadratic programming for large-scale nonlinear optimization[J]. Journal of Computational and Applied Mathematics, 2000, 124(1/2): 123-137.
- [30] 缪裕青, 徐伊, 张万桢, 等. 车联网中改进粒子群算法的任务卸载策略[J]. 计算机应用研究, 2021, 38(7): 2050-2055.
- MIAO Y Q, XU Y, ZHANG W Z, et al. Improved particle swarm algorithm for task offloading in vehicular networks[J]. Application Research of Computers, 2021, 38(7): 2050-2055.
- [31] ZHOU J S, TIAN D X, WANG Y P, et al. Reliability-optimal cooperative communication and computing in connected vehicle systems[J]. IEEE Transactions on Mobile Computing, 2020, 19(5): 1216-1232.

## [作者简介]



曹敦(1979-), 女, 湖南耒阳人, 博士, 长沙理工大学副教授、硕士生导师, 主要研究方向为车联网和 MIMO 无线通信等。



张应宝(1997-), 男, 甘肃天水人, 长沙理工大学硕士生, 主要研究方向为移动边缘计算。



邹电(1992-), 男, 湖南岳阳人, 长沙理工大学硕士生, 主要研究方向为数字孪生。

王进(1979-), 男, 江苏扬州人, 博士, 长沙理工大学教授、博士生导师, 主要研究方向为物联网、车联网和无线传感网等。

汤强(1982-), 男, 湖南岳阳人, 博士, 长沙理工大学讲师、硕士生导师, 主要研究方向为智能电网、无线网络和移动计算等。

冀保峰(1985-), 男, 河南洛阳人, 博士, 河南科技大学副教授, 主要研究方向为短距离协同传输毫米波 MIMO、自适应信号处理等。