

车载多接入边缘网络中联合资源分配和动态任务卸载方案*

薛建彬, 安悦, 关向瑞, 安亚宁

(兰州理工大学计算机与通信学院, 甘肃兰州 730050)

摘要: 针对车联网中车辆移动速度过快产生的任务卸载失败问题, 设计了一个有效的任务卸载风险评估模型, 并提出了联合资源分配的动态任务卸载方案。将时间、能耗和风险共同建模为系统效用, 通过联合优化卸载决策、资源分配来最大化系统效用。优化问题被公式化为混合整数非线性规划, 在给定卸载决策的情况下, 利用凸优化技术解决计算资源分配问题, 功率分配通过分式规划技术来优化。仿真分析了车辆移动性对系统效用的影响, 证明了所提方案的合理性。

关键词: 移动边缘计算; V2V 通信; 资源分配; 动态任务卸载

中图分类号: TN929.5 文献标志码: A 开放科学(资源服务)标识码(OSID):

文章编号: 1001-2486(2022)06-061-09



与作者互动
听语音
聊科研

Joint resource allocation and dynamic task offloading scheme in vehicle multi-access edge network

XUE Jianbin, AN Yue, GUAN Xiangrui, AN Yaning

(School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: Aiming at the problem of task offloading failure caused by the excessively fast vehicle moving in the Internet of vehicles, an effective task offloading risk assessment model was designed, and a dynamic task offloading scheme based on joint resource allocation was proposed. Time, energy consumption, and risk were jointly modeled as system utility, and the system utility was maximized through joint optimization of offloading decision and resource allocation. The optimization problem was formulated as a mixed integer nonlinear programming. Convex optimization techniques were used to solve the computational resource allocation problem under a given offloading decision, power allocation was optimized through fractional programming technology. The simulation analyzes the impact of vehicle mobility on system utility, and the rationality of the proposed scheme is proved.

Keywords: mobile edge computing; V2V communication; resource allocation; dynamic task offloading

随着移动网络和物联网技术的快速发展, 专家预测: 至 2022 年, 汽车产业年销量将达到 600 万辆, 全球汽车电子市场规模达 3 800 亿美元, 继续呈加速增长态势, 智能网联车热潮已经袭来^[1]。车辆激增涌现出了越来越多的时延敏感型和计算密集型的新型应用, 例如自动驾驶、车辆社交网络^[2]和交互式游戏^[3], 为了应对不断激增的数据流量和海量的设备连接, 5G 将渗透在未来生活的各个领域。车联网技术与 5G 的落地应用, 将为未来无人驾驶汽车提供可靠的技术支持, 并将车联网带入一个新时代——无人驾驶时代。但是, 这些新型应用给人们的生活带来巨大便利的同时, 大量的计算密集型应用会消耗更多的能

量, 给车载终端造成巨大的压力^[4]。此外, 由于移动车载终端的计算资源有限, 会增加消息处理的时延^[5], 这对车联网技术的研究产生了巨大的挑战。

面对这些挑战, 提出了支持多接入边缘计算(multi-access edge computing, MEC)^[6]的车载网络架构。MEC 服务器有强大的计算和存储能力^[7], 基于 MEC 的解决方案, 可以提供低时延、高带宽并切合各应用场景的计算处理能力, 具有很强的适用性, 在这种协作架构下, 车辆产生的任务可以卸载至路边基础设施单元(road side unit, RSU)上的 MEC 服务器执行或者卸载至通信范围内的其他车辆上处理, 从而节省了传输时延和处

* 收稿日期: 2021-01-10

基金项目: 国家自然科学基金资助项目(61841107); 甘肃省自然科学基金资助项目(18J3RA153, 20JR10RA182); 甘肃省教育厅创新基金资助项目(2022A-215)

作者简介: 薛建彬(1973—), 男, 甘肃会宁人, 教授, 博士, 博士生导师, E-mail: xuejb@lut.edu.cn; 安悦(通信作者), 女, 硕士研究生, E-mail: 2237908411@qq.com

理能耗,解决了车辆自身计算资源不足的问题^[8]。

针对车辆计算资源不足的问题,建议将车辆任务卸载到其通信范围内的其他车辆执行,考虑到车辆的移动性,在动态网络下定义了一个风险函数来评估任务卸载失败的可能性,并联合系统风险提出了卸载效用函数。

1 相关工作

随着以大数据为主要特征的各种先进技术在交通领域的应用,车联网成为必然的发展趋势。然而现今的车载设备有限的计算和存储能力难以满足大量高计算需求和低时延的限制,因此如何平衡应用时延高要求和车载终端计算能力,成为近些年研究的重点。多接入边缘计算作为 5G 演进的关键技术,有助于克服车联网领域中的发展瓶颈,为其发展提供有力支持。

针对车辆自身资源不足的问题,Hou 等^[9]提议将道路空闲车辆作为基础设施,让这些车辆提供卸载所需的计算资源。此外,文献[10]引入了 $M/M/1$ 排队模型,设计了一种基于 MEC 负载状态的预测性卸载策略,通过车与车通信方式进行任务卸载。文献[11]不仅考虑将车辆产生的任务卸载至边缘服务器,还进一步把任务划分为可分割和不可分割,再分别进行卸载,以减少系统能耗。

前期研究任务卸载的目的是为了优化终端设备的能耗或者节省任务处理时延,但随着研究的深入,在计算卸载过程中,除了上述优化目标外,还有许多学者研究了卸载过程中的无线电资源分配和计算资源分配^[12]。例如,Wang 等^[13]在考虑系统总收入的条件,联合优化卸载决策和资源分配,以提高无线网络的性能。此外,车联网的服务质量和网络成本也值得重点关注。文献[14]提出了一种时延预测的组合方式,将任务自适应地卸载到 MEC 服务器,节省了计算成本。Si 等^[15]将车辆视为边缘节点,利用车辆的潜在资源,减轻了网络中的数据拥塞问题,提升了用户的服务质量。

总的来说,大部分已有研究将任务卸载处理可以节省成本^[16],或提高用户的卸载效用^[17],能够弥补车辆自身资源匮乏的缺点。但这些方案都有一些局限性,以上研究都假设系统是准静态的,并没有考虑到车辆速度过快导致车与车或者车与基础设施通信过程中连接中断的问题^[18]。一旦在任务卸载期间车辆或是 MEC 服务器与产生任

务的终端车辆断开连接,将会增大网络时延和能耗成本。因此,为了提高网络性能,在车辆卸载过程中,应考虑到车辆的移动性对系统的影响。

2 系统模型

考虑如图 1 所示的多个任务车辆、多个处理车辆的系统,处理车辆向资源受限的移动车辆提供计算卸载服务。将移动系统中的任务车辆集以及处理车辆集分别表示为 $\hat{N} = \{1, 2, \dots, N\}$ 和 $\hat{J} = \{1, 2, \dots, J\}$ 。

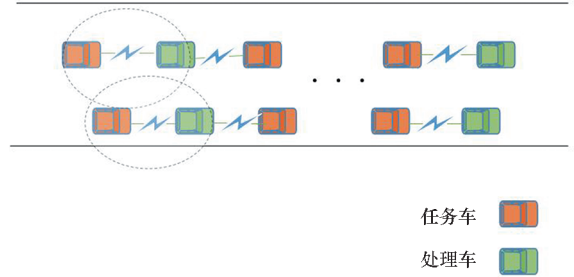


图 1 基于 V2V 通信的计算任务卸载

Fig. 1 Computing task offloading based on V2V communication

2.1 本地计算任务模型

考虑到每个移动车辆 $n \in \hat{N}$ 只有一个待处理任务 K_n ,任务不可分割为子任务, K_n 可表示为 $\{d_n, c_n\}$, d_n 表示输入数据的大小, c_n 是完成任务所需要的 CPU 周期数。每个任务 K_n 可以在本地处理,也可卸载至其他空闲邻近车辆处理。 K_n 在本地处理的任务时延为:

$$t_n^l = \frac{c_n}{f_n^l} \quad (1)$$

其中, f_n^l 是车载设备 n 本地计算能力, $f_n^l > 0$ 。本地计算的能耗为:

$$e_n^l = \varepsilon (f_n^l)^2 c_n \quad (2)$$

其中, ε 为能量系数,它的值取决于芯片结构。

2.2 任务卸载模型

任务节点 n 卸载它的任务 K_n 到其中一个处理车辆 j ,产生的延迟由三部分组成:传输延迟,在处理节点的计算延迟,输出结果回传延迟。但是,由于输出数据远小于输入数据,因此,第三部分的延迟经常可以忽略。

在上行链路传输过程考虑正交频分多址 (orthogonal frequency division multiple access, OFDMA) 接入技术,把工作频带 B 分成 M 个子信道, $\hat{M} = \{1, 2, \dots, M\}$ 表示子信道的集合,每个子信道的大小为 $W = B/M$ 。一个任务车辆只可占用

一个子信道,定义卸载变量 $a_{nj} = \{0, 1\}$, $\forall n \in \hat{N}$, $j \in \hat{J}$,其中, $a_{nj} = 1$ 表示任务 K_n 卸载至处理车辆 j 执行,否则, $a_{nj} = 0$ 表示任务在本地计算。因此,卸载策略必须满足以下约束条件:

$$\sum_{j \in \hat{J}} a_{nj} \leq 1, \forall n \in \hat{N} \quad (3)$$

定义 $N_j = \{n \in \hat{N} | a_{nj} = 1\}$ 为卸载到处理车辆 j 的任务车辆集合,则所有进行卸载处理的任务车辆集合表示为 $N_{\text{off}} = \cup_{j \in \hat{J}} N_j$, $N_{\text{off}} = |N_{\text{off}}|$ 表示进行卸载处理的车辆总数,定义 $\mathbf{Q} = \{a_{nj} | n \in \hat{N}, j \in \hat{J}\}$ 为卸载决策矢量, $\mathbf{Z} = \{a_{nj} = 1 | n \in \hat{N}, j \in \hat{J}\}$ 为所有任务卸载策略集合。任务 K_n 卸载至处理车辆 j 的传输速率为:

$$r_{nj} = W \log_2 \left(1 + \frac{p_n h_{nj}}{\sigma^2} \right), \forall n \in \hat{N}, j \in \hat{J} \quad (4)$$

其中, h_{nj} 为任务车辆 n 和处理车辆 j 之间的信道增益, p_n 为任务车辆 n 的发射功率, σ^2 为背景噪声方差。

则上行链路发送任务 K_n 的时间为:

$$t_{\text{off}}^n = \sum_{j \in \hat{J}} \frac{a_{nj} d_n}{r_{nj}}, \forall n \in \hat{N} \quad (5)$$

任务在处理车辆处花费的时间为:

$$t_{\text{exe}}^n = \sum_{j \in \hat{J}} \frac{a_{nj} c_n}{f_{nj}}, \forall n \in \hat{N} \quad (6)$$

定义计算资源分配策略为 $F = \{f_{nj} | n \in \hat{N}, j \in \hat{J}\}$,其中, $f_{nj} > 0$ 是处理车辆 j 分配给任务车辆 $n \in N_j$ 的计算资源。每个处理车辆能够同时向多个任务车辆提供计算卸载服务, f_{nj} 为处理车辆 j 分配给任务车辆 n 的计算资源。应该满足:

$$\sum_{n \in \hat{N}} f_{nj} \leq F_j, \forall j \in \hat{J} \quad (7)$$

则卸载的总时延为:

$$T_n = t_{\text{off}}^n + t_{\text{exe}}^n = \sum_{j \in \hat{J}} a_{nj} \left(\frac{d_n}{r_{nj}} + \frac{c_n}{f_{nj}} \right), \forall n \in \hat{N} \quad (8)$$

上行链路的能量消耗为:

$$E_n = p_n t_{\text{off}}^n = p_n d_n \sum_{j \in \hat{J}} \frac{a_{nj}}{r_{nj}}, \forall n \in \hat{N} \quad (9)$$

2.3 卸载风险模型

定义当卸载时间大于车辆之间连接时间为风险。设计一个风险评估模型来估计车辆卸载失败的可能性,风险被公式化为卸载时间与车辆连接时间的比例。

$$L_n(T) = \begin{cases} \sum_{j \in \hat{J}} \frac{a_{nj} T_n}{\frac{R - D_{nj}}{|v_j - v_n|}}, \forall n \in \hat{N}, T_n > \frac{R - D_{nj}}{|v_j - v_n|} \\ 0, T_n \leq \frac{R - D_{nj}}{|v_j - v_n|} \end{cases} \quad (10)$$

其中: R 为车辆的通信范围; D_{nj} 为车辆 n 与车辆 j 的欧式距离, $D_{nj} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2}$; v_j 和 v_n 分别为车辆 n 和车辆 j 的速度。

2.4 卸载效用模型

定义卸载效用函数为:

$$U_n = H(a, p, f) - \varphi \sum_{n \in \hat{N}} L_n(T) \quad (11)$$

其中: $H(a, p, f) = \phi_n \sum_{j \in \hat{J}} a_{nj} \left(\alpha \frac{t_n^l - T_n}{t_n^l} + \beta \frac{e_n^l - E_n}{e_n^l} \right)^{[20]}$,

$\forall n \in \hat{N}$, $\frac{t_n^l - T_n}{t_n^l}$ 和 $\frac{e_n^l - E_n}{e_n^l}$ 分别表示时间和能耗的改善^[19]; $\alpha, \beta \in [0, 1]$, $\alpha + \beta = 1$, α, β 分别表示用户对任务完成时间和能耗的偏好。当 $H_n \leq 0$ 时表示任务车辆 n 不卸载任务到处理车辆, $\phi_n \in (0, 1]$, $\forall n \in \hat{N}$ 代表资源提供商对车辆 n 的偏好, $\varphi \in (0, 1]$ 表示风险因子。

3 问题建模和解耦

3.1 问题建模

对于给定的卸载决策和资源分配,定义系统的卸载效用为所有用户卸载效用的加权和。本节将优化问题公式化为混合整数非线性规划(mixed integer nonlinear programming, MINLP),目标函数为:

$$\max_{\{a_{nj}\}, \{p_n\}, \{f\}} \sum_{n \in \hat{N}} U_n \quad (12)$$

$$\text{s. t. } a_{nj} = \{0, 1\}, \forall n \in \hat{N}, j \in \hat{J} \quad (13)$$

$$\sum_{j \in \hat{J}} a_{nj} \leq 1, \forall n \in \hat{N} \quad (14)$$

$$0 < p_n \leq P, \forall n \in N_{\text{off}} \quad (15)$$

$$f_{nj} > 0, \forall n \in N_j, j \in \hat{J} \quad (16)$$

$$\sum_{n \in \hat{N}} f_{nj} \leq F_j, \forall j \in \hat{J} \quad (17)$$

约束式(13)是一个二进制卸载变量,表示任务可以在本地计算,也可以卸载到处理车辆执行。式(14)表示一个任务至多卸载至一个处理车辆执行,式(15)表示传输功率约束,式(16)表示每个卸载任务都应该分配计算资源,式(17)表示分配的总的计算资源应该在处

理车辆的服务能力之内。

3.2 问题分解

给定一个满足约束式(13)、式(14)的任务卸载决策,整理可将目标函数重写为:

$$\max_{\{a_{nj}\}, \{p_n\}, \{f\}} \sum_{j \in \hat{J}} \sum_{n \in N_j} \phi_n(\alpha + \beta) - \Gamma(a, p, f) \quad (18)$$

对于给定的卸载决策,式(18)中的第一项为常数。其中

$$\Gamma(a, p, f) = \sum_{n \in N_j} \sum_{j \in \hat{J}} \varphi \left(\frac{\alpha T_n}{t_n^l} + \frac{\beta E_n}{e_n^l} \right) + \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{c_n \varphi}{R - D_{nj}} + \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\varphi D_n}{w \log_2(1 + \sigma^2)} \quad (19)$$

所以目标变成了

$$\min_{p, f} \Gamma(a, p, f) \quad (20)$$

$$\text{s. t. } \begin{cases} 0 < p_n \leq P, \forall n \in N_{\text{off}} \\ f_{nj} > 0, \forall n \in N_j, j \in \hat{J} \\ \sum_{n \in N_j} f_{nj} \leq F_j, \forall j \in \hat{J} \end{cases}$$

根据式(8)、式(9)、式(19)。可以得到

$$\Gamma(a, p, f) = \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n + \lambda_n}{\log_2(1 + \frac{p_n h_{nj}}{\sigma^2})} + \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\eta_n + \mu_n}{f_{nj}} \quad (21)$$

$$\text{s. t. } \begin{cases} 0 < p_n \leq P, \forall n \in N_{\text{off}} \\ f_{nj} > 0, \forall n \in N_j, j \in \hat{J} \\ \sum_{n \in N_j} f_{nj} \leq F_j, \forall j \in \hat{J} \end{cases}$$

其中, $\tau_n = \frac{\varphi \alpha d_n}{W t_n^l}$, $\varepsilon_n = \frac{\phi_n \beta d_n}{W e_n^l}$, $\lambda_n = \frac{\varphi d_n}{W \frac{R - D_{nj}}{|v_j - v_n|}}$,

$\eta_n = \phi_n \alpha f_n^l$, $\mu_n = \frac{c_n \varphi}{R - D_{nj}}$ 。式(21)第一项是关于

功率 p 的函数,第二项只依赖于计算资源的分配,此外,约束条件式(15) ~ (17) 是独立的,换句话说,给定确定的卸载决策,通信资源和计算资源的优化是无关系的。因此可以把这个问题解耦为功率分配和计算资源分配问题。

3.2.1 功率分配问题

$$\text{P1: } \min_p T(a, p, f) = \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n + \lambda_n}{\log_2(1 + \frac{p_n h_{nj}}{\sigma^2})} \quad (22)$$

$$\text{s. t. } 0 < p_n \leq P, \forall n \in N_{\text{off}}$$

由于目标函数是非凸函数,问题 P1 是非凸优化问题,然而 P1 可以归入非线性分式规划问题^[20],因此可以用分式规划来解决它。假设存在一个集合 $\zeta (\zeta \neq 0)$ 为问题式(22) 的可行解。定义一个最优解 p_n^* 和最优值 v^* , 可得:

$$v^* = \min_p \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n + \lambda_n}{\log_2(1 + p_n H_{nj})} \quad (23)$$

$$= \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n^* + \lambda_n}{\log_2(1 + p_n^* H_{nj})}$$

$$\text{s. t. } 0 < p_n \leq P, \forall n \in N_{\text{off}}$$

其中, $H_{nj} = \frac{h_{nj}}{\sigma^2}$ 。

引理 1 达到最优值 v^* 的条件是,当且仅当

$$\min_p \sum_{n \in N_j} \sum_{j \in \hat{J}} [(\tau_n + \varepsilon_n p_n + \lambda_n) - v^* \log_2(1 + p_n H_{nj})] \quad (24)$$

$$= \sum_{n \in N_j} \sum_{j \in \hat{J}} [(\tau_n + \varepsilon_n p_n^* + \lambda_n) - v^* \log_2(1 + p_n^* H_{nj})]$$

$$= 0$$

该定理的证明可在文献[21]中得到。

上述定理表示问题 P1 可以通过它的等价问题式(24)得到。此外,由于 v^* 一般是未知的,故用 v 代替^[22],如算法 1 所示。

问题式(23)可以改成以下形式:

$$\text{P2: } \min_p \sum_{n \in N_j} \sum_{j \in \hat{J}} [(\tau_n + \varepsilon_n p_n + \lambda_n) - v \log_2(1 + p_n H_{nj})] \quad (25)$$

$$0 < p_n \leq P, \forall n \in N_{\text{off}}$$

为了解决 P2,注意到问题 P2 为凸函数和线性函数的加权和,同时约束条件为凸条件,因此问题 P2 为凸函数^[23]。该问题满足斯莱特条件,可以使用拉格朗日对偶分解和梯度下降法来解决^[24]。利用拉格朗日乘子法可将问题转化为:

$$\max_k \min_p L(p, k) \quad (26)$$

该问题可以通过一个主问题和多个子问题的迭代来解决^[24]。子问题为:

$$D(k) = \min_p L(p, k) \quad (27)$$

拉格朗日函数为:

$$L(p, k) = \sum_{n \in N_j} \sum_{j \in \hat{J}} [(\tau_n + \varepsilon_n p_n + \lambda_n) - v \log_2(1 + p_n H_{nj})] - \sum_{n \in N_j} k_n (P - p_n) \quad (28)$$

其中, $k = (k_1, k_2, \dots, k_{N_{\text{off}}}) \geq 0$ 。通过 KKT (Karush-Kuhn-Tucker) 条件,求函数 $L(p, k)$ 对于 p_n 的微分,可以获得每个用户的最优传输功率分配。

$$p_n^* = \left\{ \frac{v}{(\varepsilon_n + k_n) \ln 2} - \frac{1}{H_{nj}} \right\}^+, n \in N_j, j \in \hat{J} \quad (29)$$

算法1 迭代功率分配解决 P1

Alg.1 Iterative power allocation to solve P1

初始化:

1. 设置最大迭代次数 i_{\max} 和精度 δ
 2. 设置初始迭代指标 $i=0$ 和初始最优值 $v^i=1$
- 迭代:
3. **While** $i < i_{\max}$ **do**
 4. **For** 给定 v^i , 解决 P2 获得 p^i 的值
 5. **if** $\left| \sum_{n \in \hat{N}_j} \sum_{j \in \hat{J}} [(\tau_n + \varepsilon_n p_n^i + \lambda_n) - v \log_2(1 + p_n^i H_{nj})] \right| < \delta$ **then**
 6. $p^* = p^i$
 7. $v^* = \sum_{n \in \hat{N}} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n^i + \lambda_n}{\log_2(1 + p_n^i H_{nj})}$
 8. **else**
 9. 设置 $v^{i+1} = \sum_{n \in \hat{N}} \sum_{j \in \hat{J}} \frac{\tau_n + \varepsilon_n p_n^i + \lambda_n}{\log_2(1 + p_n^i H_{nj})}$
 10. **break**
 11. **end if**
 12. **end for**
 13. **end while**

主问题通过拉格朗日乘子来解决,利用梯度下降法更新拉格朗日乘子, k 的表达式为:

$$k(i+1) = k(i) - \delta_k \left(\sum_{n \in \hat{N}} P - p_n^* \right) \quad (30)$$

其中, i 是迭代指标, δ_k 是正步长, $\delta_k = \frac{1}{10^i}$ 。算法流程如算法2所示。

算法2 次优功率分配算法解决 P2

Alg.2 Suboptimal power allocation algorithm solves P2

初始化:

1. 设置 k, i^{\max} 和精度 δ , 设置 $i=0$
- 迭代:
2. **while** $i < i^{\max}$ **do**
 3. 根据式(28)分配传输功率 $p_n(t)$
 4. 根据式(29)更新拉格朗日乘子
 5. **if** $\|k(i+1) - k(i)\|_2 < \delta$ **then**
 6. $p_n^* = p_n(i)$
 7. **break**
 8. **else**
 9. $i = i + 1$
 10. **end if**
 11. **end while**
 12. 输出: $p_n^* = (p_1^*, \dots, p_{N_j}^*)$

3.2.2 计算资源分配问题

$$P3: \min_f \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\eta_n + \mu_n}{f_{nj}} \quad (31)$$

$$\text{s. t. } f_{nj} > 0, \forall n \in N_j, j \in \hat{J} \quad (32)$$

$$\sum_{n \in \hat{N}} f_{nj} \leq F_j, \forall j \in \hat{J} \quad (33)$$

注意到式(32) ~ (33) 两个约束条件都是凸条件,将目标函数 P3 用 $Y(\mathbb{Z}, A)$ 来表示。 $Y(\mathbb{Z}, A)$ 关于 f_{nj} 的二阶导数为:

$$\frac{\partial^2 Y^2(\mathbb{Z}, A)}{\partial f_{nj}^2} = \frac{2(\eta_n + \mu_n)}{f_{nj}^3} > 0, \forall n \in N_j, j \in \hat{J} \quad (34)$$

因此 P3 是一个凸优化问题,可以用 KKT 来解决。

引理2 问题 P3 的最优计算资源分配 f_{nj}^* 和最优解 $Y(\mathbb{Z}, A^*)$ 分别为:

$$f_{nj}^* = \sqrt{\frac{\eta_n + \mu_n}{\sum_{j \in \hat{J}} \gamma_j^*}}, \forall n \in N_j, j \in \hat{J} \quad (35)$$

$$Y(\mathbb{Z}, A^*) = \sum_{j \in \hat{J}} \frac{1}{F} \left(\sum_{n \in N_j} \sqrt{\eta_n + \mu_n} \right) \quad (36)$$

证明:问题 P3 的拉格朗日函数为:

$$L(a, f, \gamma) = \sum_{n \in N_j} \sum_{j \in \hat{J}} \frac{\eta_n + \mu_n}{f_{nj}} + \sum_{j \in \hat{J}} \gamma_j \left(\sum_{n \in N_j} f_{nj} - F_j \right) \quad (37)$$

$\gamma = (\gamma_1, \dots, \gamma_j)$ 是拉格朗日乘子,对 f_{nj} 求一阶导数,可得:

$$\frac{\partial L(a, f, \gamma)}{\partial f_{nj}} = -\frac{\eta_n + \mu_n}{f_{nj}^2} + \gamma_j, \forall n \in N_j, j \in \hat{J} \quad (38)$$

令式(38)等于0,求解出 f_{nj} ,然后可得到最优的计算资源分配。

$$f_{nj} = \sqrt{\frac{\eta_n + \mu_n}{\sum_{j \in \hat{J}} \gamma_j^*}}, \forall n \in \hat{N}, j \in \hat{J} \quad (39)$$

其中, $\gamma_j^* > 0$, 满足

$$\sum_{n \in \hat{N}} f_{nj}^* = F_j, \forall j \in \hat{J} \quad (40)$$

通过将式(39)代入式(40),可得到最优的拉格朗日乘子 γ_j^* 。注意当 $n \notin N_j$ 时, $f_{nj}^* = 0$ 。

$$\gamma_j^* = \frac{\sum_{n \in N_j} (\eta_n + \mu_n)}{\sum_{j \in \hat{J}} F_j^2}, \forall j \in \hat{J} \quad (41)$$

把式(41)代入式(39)可得:

$$f_{nj}^* = \frac{F_j \sqrt{\eta_n + \mu_n}}{\sum_{n \in N_j} \sqrt{\eta_n + \mu_n}}, \forall j \in \hat{J}, n \in N_j \quad (42)$$

最后,目标函数 P3 可表示为:

$$Y(\mathbb{Z}, A^*) = \sum_{j \in \hat{J}} \frac{1}{F} \left(\sum_{n \in \hat{N}_j} \sqrt{\eta_n + \mu_n} \right) \quad (43)$$

3.2.3 联合任务卸载调度和资源分配

在上述分析中,对于一个给定的卸载决策,获得了功率分配和计算资源分配,根据式(18)、式(21)和式(36)可得:

$$U^*(F) = \sum_{n \in \hat{N}_j} \sum_{j \in \hat{J}} \varphi(\alpha + \beta) - T(a, p^*) - Y(\mathbb{Z}, F^*) \quad (44)$$

$$\text{s. t. } a_{nj} = \{0, 1\}, \forall n \in \hat{N}, j \in \hat{J} \quad (45)$$

$$\sum_{j \in \hat{J}} a_{nj} \leq 1, \forall n \in \hat{N} \quad (46)$$

其中, p^* 可以通过算法 1 获得, F^* 可以通过式(35)得到。原始问题可重写为:

$$\max_a \sum_{n \in \hat{N}_j} \sum_{j \in \hat{J}} \varphi(\alpha + \beta) - T(a, p^*) - Y(\mathbb{Z}, F^*) \quad (47)$$

解决问题式(38)的一个直接方法是穷举所有可能的卸载决策,进而得到最优的卸载决策,但此种方法复杂度为 $O(2^n)$, 其中 $n = N \times J$, 显然,使用穷举法是不切实际的,因此提出一种次优卸载决策的方法来降低复杂度^[25]。程序初始时,假设卸载集合 $\mathbb{Z} = \emptyset$, 找到效用最大的集合 \mathbb{Z} , 如果存在一个决策 $a_{nj} \in \mathbb{Z}$, 将它删除后集合的效用比删除之前的集合效用大,那么执行删除操作,从集合中删除这个决策,直到再也找不出这样的决策就停止删除操作;如果存在一个决策 $a_{nj} \in \mathcal{Q}/\mathbb{Z}$, 将该决策加入集合 \mathbb{Z} 中,使得系统效用的值变大,那么执行交换操作,直到再也找不出这样的决策,程序结束。最后,找到了使系统效用最好的卸载决策集合 \mathbb{Z} , 具体步骤如算法 3 所示。

算法 3 删除和交换求解卸载决策

Alg. 3 Delete and swap solution offloading decisions

初始化: $\mathbb{Z} = \emptyset$

1. 寻找 $a_{we} = \arg \max_{n \in \hat{N}, j \in \hat{J}} U^*(\{a_{nj}\})$
2. 设置 $\mathbb{Z} \leftarrow \{a_{we}\}$
3. **if** 存在 $a_{nj} \in \mathbb{Z}$ such that $U^*(\text{remove}(\mathbb{Z}, a_{nj})) > \left(1 + \frac{1}{\Omega(q, \varepsilon)}\right) U^*(\mathbb{Z})$ **then**
4. 设置 $\mathbb{Z} \leftarrow \text{remove}(\mathbb{Z}, a_{nj})$
5. 返回到第 3 步
6. **else if** 存在 $a_{nj} \in \mathcal{Q}/\mathbb{Z}$ such that $U^*(\text{exchange}(\mathbb{Z}, a_{nj})) > \left(1 + \frac{1}{\Omega(q, \varepsilon)}\right) U^*(\mathbb{Z})$ **then**
7. 设置 $\mathbb{Z} \leftarrow \text{exchange}(\mathbb{Z}, a_{nj})$
8. 回到第 4 步
9. **end if**
10. 输出 \mathbb{Z}

4 仿真分析

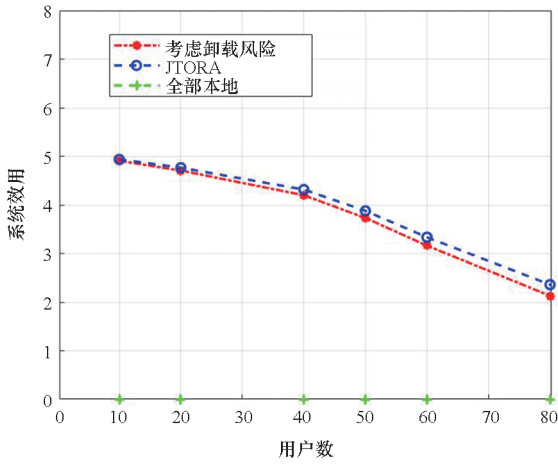
仿真结果评估了所提联合动态任务卸载调度和资源分配 (joint dynamic task offloading and resource allocation, JDTORA) 策略的性能。系统仿真参数的设置: $P = 20$ dBm, 假设处理车辆和本地车辆 CPU 能力分别为 $F_j = 20$ GHz, $f_n^l = 1$ GHz, 根据文献[26], 设置能量系数为 5×10^{-27} , 设置偏好参数 $\alpha = 0.2, \beta = 0.8, \varphi = 1$, $\forall n \in \hat{N}$, 车辆速度 v 在 50 ~ 110 km/h 之间取值。

图 2(a) ~ (c) 所展示的是用户数和系统效用之间的关系, 将用户数逐步增加, 并在三种不同工作负载的情况下进行比较。从图 2(a) ~ (c) 可以看出, 随着任务量的增加, 所有方案的性能也显著增加。这是因为, 当任务需要更多的计算资源时, 用户将从把任务卸载到处理车辆中获益更多。还发现, 随着用户数的增加, 系统效用也在增加。这是因为, 当有许多用户竞争无线资源和计算资源来卸载他们的任务时, 发送任务和在处理车辆上执行任务的开销会更高, 从而降低了卸载效用。从图中还可以看出, 在用户数相同的情况下, 本文所提 JDTORA 方案与联合任务卸载和资源分配 (joint task offloading and resource allocation, JTORA) 方案的系统效用性能极其接近, 差值趋于 0。这说明在考虑卸载风险的情况下, 即对影响系统卸载的因素考虑更为全面时, 系统效用依旧可以趋于不考虑风险的优解, 验证了所提方案的有效性。与传统的全部本地计算方案进行了比较, 本地计算时的卸载效用值为 0, 因此, 当用户数相同的情况下, 本文所提方案的效用大于本地卸载方案的效用。

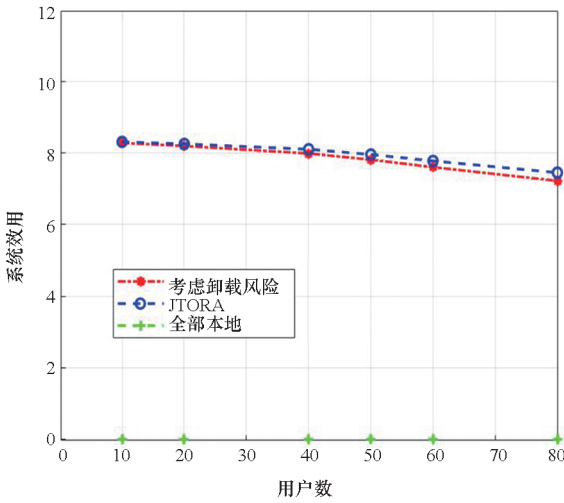
图 3(a) ~ (b) 表示当用户对时间的偏好因子从 0.1 到 0.9 变化时, 用户的平均时间消耗和能量消耗变化趋势。从图中可以看出, 当用户对时间的偏好 α 增加时, 平均时间消耗降低, 相反, 平均能量消耗随之增加。此外, 图中显示当用户数更多时, 用户会经历更大的时间消耗和能量消耗, 这是因为当争夺有限资源的用户数更多时, 用户从卸载的任务中获益的机会更小。

图 4 显示的是用户数与系统风险之间的关系。本文的风险是由于任务卸载引起的, 从图中可以直观地看出, 当任务不进行卸载时 ($a_{nj} = 0$), 系统风险为 0; 而当用户数增加时, 系统的风险也

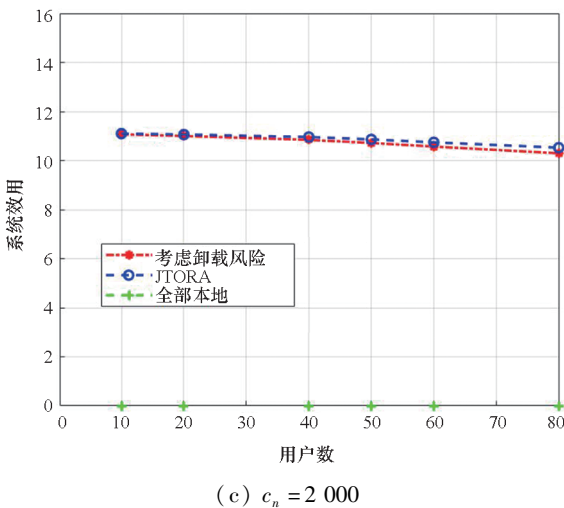
随之增大。这是因为当更多的用户卸载它们的任务时,增大了系统的卸载风险。



(a) $c_n = 1\ 000$



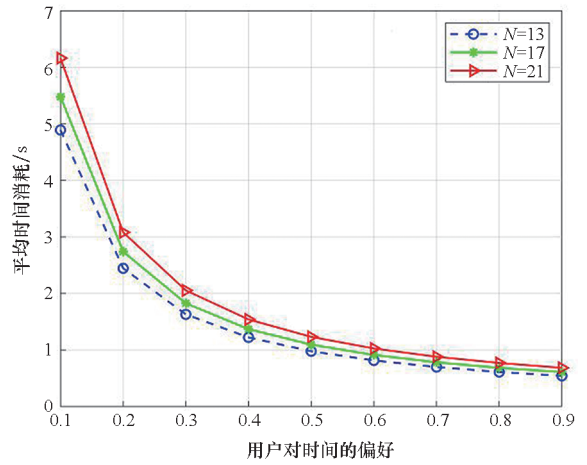
(b) $c_n = 1\ 500$



(c) $c_n = 2\ 000$

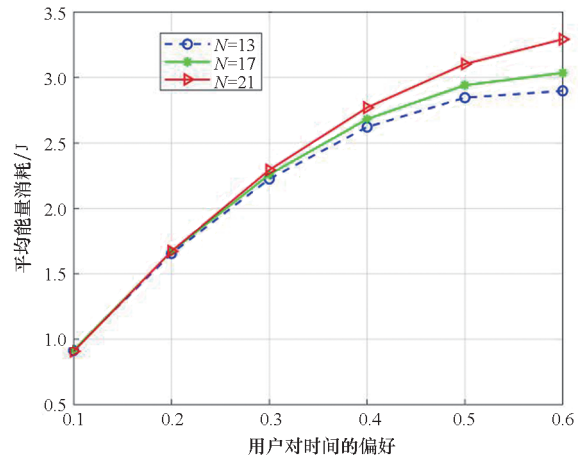
图2 不同工作负载情况下,针对不同用户数的平均系统效用比较

Fig.2 Comparison of the average system utility of different numbers of users under different workloads



(a) 平均时间消耗

(a) Average time consumption



(b) 平均能量消耗

(b) Average energy consumption

图3 使用 JDTORA 获得的所有用户的平均时间消耗和能量消耗

Fig.3 Average time consumption and energy consumption of all users obtained by using JDTORA

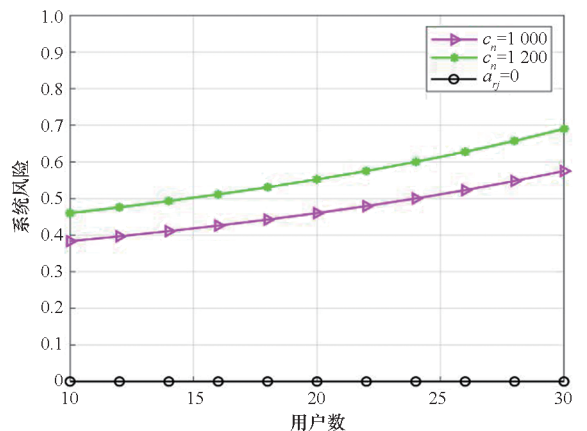


图4 不同工作负载情况下,针对不同用户数的系统风险比较

Fig.4 Comparison of system risks with different numbers of users under different workloads

5 结论

通过卸载决策和资源分配的联合优化,设计了边缘车联网系统中基于考虑卸载风险的效用最大化问题。优化问题被表述为混合整数非线性规划问题,此问题是难以解决的,因此将原始问题解耦成两个子问题,分别使用凸优化技术和分式规划技术解决。最后,联合两个子问题的解在多项式时间内求出了最优的卸载决策。仿真结果表明所提方案在有大量用户车辆的实际环境中是有效的。与其他研究不同的是,本文考虑到车辆的移动性,设计了联合资源分配的动态任务卸载方案,建模了一个风险函数来评估车辆连接中断的风险。本文的主要贡献如下:

1) 设计了一个边缘车辆网络中的动态卸载模型,为了在动态网络中成功卸载任务,定义了一个风险函数来评估任务卸载失败的可能性。

2) 联合系统风险定义了一个新的卸载效用函数,并将此问题建模为任务卸载决策、功率分配和计算资源分配的联合优化问题,以最大化系统的总效用。

3) 混合整数非线性规划问题是难以解决的,通过计算发现,可以将问题解耦成资源分配问题和具有特定解的卸载决策问题。进一步地,将资源分配问题分解为功率分配和计算资源分配,分别使用分式规划和凸优化技术进行求解。

4) 仿真结果表明所提动态卸载方案能够得到近似最优解,这证明了该方案的有效性。

参考文献 (References)

- [1] CHEN H D, ZHAO T T, LI C M, et al. Green Internet of vehicles: architecture, enabling technologies, and applications[J]. IEEE Access, 2019, 7: 179185 – 179198.
- [2] PAPADIMITRIOU A, KATSAROS D, MANOLOPOULOS Y. Social network analysis and its applications in wireless sensor and vehicular networks [C]//Proceedings of International Conference on e-Democracy, 2009.
- [3] SOOKHAK M, YU F R, HE Y, et al. Fog vehicular computing: augmentation of fog computing using vehicular cloud computing[J]. IEEE Vehicular Technology Magazine, 2017, 12(3): 55 – 64.
- [4] ZHAO H T, ZHU Q X, CHEN Y, et al. A research of task-offloading algorithm for distributed vehicles[C]//Proceedings of 2020 IEEE International Conference on Communications Workshops, 2020.
- [5] WANG C, LI Y, JIN D P, et al. On the serviceability of mobile vehicular cloudlets in a large-scale urban environment[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(10): 2960 – 2970.
- [6] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey [J]. IEEE Internet of Things Journal, 2018, 5(1): 450 – 465.
- [7] SUN C Y, SUN Y, WANG W Z. A survey of MEC: 1998 – 2001 [C]//Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2002.
- [8] ZHANG K, MAO Y M, LENG S P, et al. Predictive offloading in cloud-driven vehicles: using mobile-edge computing for a promising network paradigm [J]. IEEE Vehicular Technology Magazine, 2017, 12(2): 36 – 44.
- [9] HOU X S, LI Y, CHEN M, et al. Vehicular fog computing: a viewpoint of vehicles as the infrastructures [J]. IEEE Transactions on Vehicular Technology, 2016, 65(6): 3860 – 3873.
- [10] 万鹏. 基于移动边缘云计算的车联网计算卸载调度[D]. 武汉: 华中科技大学, 2019.
- WAN P. Computation offloading scheduling in vehicular networks based on mobile edge computing [D]. Wuhan: Huazhong University of Science and Technology, 2019. (in Chinese)
- [11] LIU J, WANG S B, WANG J T, et al. A task oriented computation offloading algorithm for intelligent vehicle network with mobile edge computing [J]. IEEE Access, 2019, 7: 180491 – 180502.
- [12] KHALILI A, ZARANDI S, RASTI M. Joint resource allocation and offloading decision in mobile edge computing[J]. IEEE Communications Letters, 2019, 23(4): 684 – 687.
- [13] WANG C M, LIANG C C, YU F R, et al. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing [J]. IEEE Transactions on Wireless Communications, 2017, 16(8): 4924 – 4938.
- [14] ZHANG K, MAO Y M, LENG S P, 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.
- [15] SI P B, HE Y, YAO H P, et al. DaVe: offloading delay-tolerant data traffic to connected vehicle networks[J]. IEEE Transactions on Vehicular Technology, 2016, 65(6): 3941 – 3953.
- [16] LI H T, LI X J, WANG W G. Joint optimization of computation cost and delay for task offloading in vehicular fog networks[J]. Transactions on Emerging Telecommunications Technologies, 2020, 31(2): e3818.
- [17] DAI Y Y, XU D, MAHARJAN S, et al. Joint offloading and resource allocation in vehicular edge computing and networks[C]//Proceedings of IEEE Global Communications Conference, 2018.
- [18] YANG F C, WANG S G, LI J L, et al. An overview of Internet of vehicles [J]. China Communications, 2014, 11(10): 1 – 15.
- [19] LYU X C, TIAN H, SENGUL C, et al. Multiuser joint task offloading and resource optimization in proximate clouds[J].

- IEEE Transactions on Vehicular Technology, 2017, 66(4): 3435 – 3447.
- [20] DINKELBACH W. On nonlinear fractional programming[J]. Management Science, 1967, 13(7): 492 – 498.
- [21] DU J B, ZHAO L Q, FENG J, et al. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee[J]. IEEE Transactions on Communications, 2018, 66(4): 1594 – 1608.
- [22] LIU Y F, WANG Y, SUN R J, et al. Energy efficient downlink resource allocation for D2D-assisted cellular networks with mobile edge caching[J]. IEEE Access, 2018, 7: 2053 – 2067.
- [23] LEMARÉCHAL C. S. Boyd, L. vandenbergh, convex optimization, Cambridge university press, 2004 hardback, 65 US \$, ISBN 0 521 83378 7 [J]. European Journal of Operational Research, 2006, 170(1): 326 – 327.
- [24] WU W, WANG Q, WU X L, et al. Joint offloading and resource allocation for scalable vehicular edge computing[C]// Proceedings of the 92nd IEEE Vehicular Technology Conference (VTC2020 – Fall), 2020.
- [25] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856 – 868.
- [26] LIWANG M H, DAI S J, GAO Z B, et al. A computation offloading incentive mechanism with delay and cost constraints under 5G satellite-ground IoV architecture [J]. IEEE Wireless Communications, 2019, 26(4): 124 – 132.