



研究与开发

时延约束与能量感知的车辆边缘计算 协同资源分配与任务卸载

高文轩¹, 杨新杰^{1,2}, 杨家智¹, 马楠³

(1. 宁波大学信息科学与工程学院, 浙江 宁波 315211;

2. 宁波艾欧迪互联科技有限公司, 浙江 宁波 315021;

3. 北京邮电大学信息与通信工程学院, 北京 100876)

摘要: 在车联网中, 计算密集型 and 时延敏感型车载应用的出现对车辆的计算能力和处理速度提出了挑战。为此, 车辆边缘计算 (vehicular edge computing, VEC) 应运而生, 它通过利用周边车辆和路侧单元 (road-side unit, RSU) 的空闲计算资源来辅助车辆进行联合任务计算。然而, 如何有效管理 VEC 系统的计算资源以实现系统性能优化, 仍是一个亟待解决的问题。基于此, 研究 VEC 系统中的协同资源分配和任务卸载问题, 通过车辆到基础设施 (vehicle-to-infrastructure, V2I) 和车辆到车辆 (vehicle-to-vehicle, V2V) 的通信模式来实现部分任务的卸载, 同时协同优化选择决策、任务卸载比例, 以及车辆和 RSU 的计算资源分配, 在保证任务完成的前提下最小化系统总能耗。由于该问题是一个 NP 难的混合整数非线性规划问题, 将原问题解耦成多个子问题, 并提出一种融合蚁群系统 (ant colony system, ACS)、拉格朗日乘子和梯度法的优化方案。仿真结果表明, 所提方案相较于对比方案在降低任务未完成率和系统总能耗方面展现出更优越的性能。

关键词: 车辆边缘计算; 能耗; 计算卸载; 蚁群系统; 未完成率

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2025049

Joint resource allocation and task offloading for latency constraint and energy-sensing vehicular edge computing

GAO Wenxuan¹, YANG Xinjie^{1,2}, YANG Jiazhi¹, MA Nan³

1. Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

2. Ningbo Arrow Deep Internet Technology Co., Ltd., Ningbo 315021, China

3. School of Information and Communication Engineering, Beijing University of
Posts and Telecommunications, Beijing 100876, China

Abstract: In the Internet of vehicles, the emergence of computationally intensive and latency-sensitive vehicular applications poses great challenges for computing time and power consumption. Vehicle edge computing (VEC) is therefore proposed to take advantages of the computing capabilities of vehicles and roadside unit (RSU). However, the effi-

收稿日期: 2024-12-01; 修回日期: 2025-01-01

通信作者: 杨新杰, yangxinjie@nbu.edu.cn



cient resource management for VEC systems to achieve optimised performance remains unsolved. Based on this, the issue of collaborative resource allocation and task offloading in the VEC system was studied, with partial tasks being offloaded through vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication modes. At the same time, the selection decisions, task offloading ratios, and the allocation of computing resources among vehicles and RSUs were collaboratively optimized, with the aim of minimizing the total system energy consumption while ensuring task completion. As the problem was a NP-hard mixed-integer nonlinear programming problem, the optimization problem was decoupled into multiple subproblems, and subsequently solved by using ant colony system (ACS), Lagrange multipliers and gradient method. Extensive simulation results demonstrate that the proposed scheme significantly outperforms the benchmarking algorithms in terms of task incomplete rate and overall system energy consumption.

Key words: VEC, energy consumption, computation offloading, ACS, incomplete rate

0 引言

随着 5G 技术的应用, 物联网 (Internet of things, IoT) 和人工智能 (artificial intelligence, AI) 领域取得了显著的发展。车联网作为物联网的延伸, 随着日益增长的智能交通建设和车辆智能化需求, 其应用场景不断丰富, 产生了大量车辆计算任务^[1-2]。这些任务可分为两大类, 即时延敏感型任务和非时延敏感型任务。前者包括地图导航、路径规划、视觉增强、行驶决策和车辆感知等涉及安全信息、道路安全、实时决策的任务, 对处理时延的要求较高; 后者包括虚拟现实、语音识别、视频处理和在线游戏等任务, 对处理时延的要求相对宽松。

限于车辆的存储和计算能力, 数据量庞大的车辆计算任务难以得到有效的本地处理。传统的基于云的计算形式由于集中化的云服务器与用户车辆间的距离通常较远, 容易受到传输路径和网络拥塞影响, 处理时延较高, 在处理大规模复杂的计算密集型 and 时延敏感型车联网应用时面临巨大挑战^[3]。车辆边缘计算 (vehicle edge computing, VEC) 是近年来备受关注的具有发展前景的新范式^[4]。VEC 将计算能力扩展到车辆网络边缘, 通过车到基础设施 (vehicle-to-infrastructure, V2I) 通信方式将车辆任务迁移到路侧单元 (roadside unit, RSU) 或附近的边缘服务器上,

或通过车到车 (vehicle-to-vehicle, V2V) 通信方式迁移到其他具有空闲计算资源的车辆上。相较于传统的基于云的方案, VEC 给用户车辆提供了更灵活的计算资源、更短的任务传输距离和更快速的响应。然而, 对于大量不同的数据量和时延要求的异构任务, 任务卸载和资源分配策略仍旧是 VEC 亟待解决的问题, 尤其在考虑车辆的移动性导致的动态网络拓扑结构时, 这一问题变得更为复杂。在 VEC 环境下, 关于计算卸载的研究主要聚焦于如何利用 V2I 和 V2V 通信技术, 实现在 RSU 或可提供计算服务车辆之间高效的计算卸载服务, 旨在提升车联网应用中任务处理的效率, 进而优化系统的处理时延和能耗, 以及基于时延和能耗的系统效用等性能指标。然而, V2I 和 V2V 通信场景中的车辆边缘计算仍面临诸多挑战, 对此, 众多研究致力于探索 VEC 架构中任务卸载与资源分配问题的解决方案。

一些研究者选择采用 V2I 方式将任务卸载至 RSU 或基站进行计算。文献[5]基于每个车辆用户利用 V2I 方式与 RSU 进行通信和任务卸载, 提出一种基于博弈模型构造的分布式最优卸载算法, 确定每个车辆用户在多用户非合作计算卸载策略下的卸载概率, 旨在使每辆车的效用最大化。文献[6]考虑 V2I 通信方式将任务卸载到服务器, 研究 VEC 系统中多用户多边缘服务器的资源分配问题, 将负载均衡和卸载相结合形成一个系统效用

函数, 提出一个低复杂度的算法, 以协同优化卸载决策和资源分配, 从而最大化系统效用。文献[7]针对V2I的多用户多服务器VEC场景, 提出基于单亲遗传算法和启发式规则的混合智能优化算法, 用于决定任务执行的位置和执行顺序, 旨在最大化基于任务时延的系统卸载利用率。文献[8]为提高系统的效用和计算时间, 提出了一种云-移动边缘计算(mobile edge computing, MEC)协同计算卸载框架, 为协同优化计算卸载决策和计算资源分配, 设计了一种分布式计算卸载和资源分配算法。文献[9]为解决服务请求过多导致边缘服务器过载和服务质量(quality of service, QoS)降低的问题, 在一个多用户卸载系统中提出了一种基于深度强化学习(deep reinforcement learning, DRL)的数字孪生赋能车联网方案, 用于优化卸载决策, 提高系统的QoS水平。文献[10]构建了支持MEC车辆网络的任务卸载场景, 场景中的车辆可以通过多种数据传输模式将任务卸载至RSU或基站, 在给定时延约束和考虑数据传输故障情况下, 采用深度Q学习方法选择最优MEC服务器和数据传输方式来最大化基于时延的卸载系统效用和卸载可靠性。对典型VEC场景下的计算卸载调度问题, 文献[11]为避免车辆移动性导致车辆与RSU之间的频繁切换所引起的数据传输时间和能耗的动态变化, 提出应用DRL来优化卸载决策和调度时间, 以平衡任务执行时延和能耗。

上述工作并未考虑车辆之间通过V2V进行任务数据传输, 且忽略了周围车辆的闲置计算资源, 而随着车辆计算能力的不断提高, 可以利用其他车辆的闲置计算资源来实现高效计算卸载, 从而提高任务处理的性能。文献[12]研究了车辆间的计算任务分配问题, 考虑无线信道状态和空闲计算资源变化, 根据空闲车辆贡献的计算资源和获得的价格提出一种分布式动态定价方案, 并通过DRL来解决任务分配问题, 旨在最大化任务车辆卸载任务的累计奖励。文献[13]针对一个基

于V2V任务卸载的场景, 允许利用可提供服务的车辆的闲置计算资源, 采用基于学习的自适应卸载策略算法来确定卸载目标, 以降低任务车辆的执行时延。文献[14]构建了一个仅依靠V2V通信的任务卸载场景, 任务请求车辆综合考虑每辆提供服务的车辆的计算能力和最大服务时间来决定分配给每辆服务车辆的任务百分比, 并用粒子群优化算法最小化任务执行时间。

以上工作已经对V2I和V2V卸载模式进行了深入的研究, 但单一的卸载模式会限制系统性能的进一步提升。此外, 过度依赖RSU或其他基础设施来卸载计算任务, 可能会导致服务器负载过重, 而仅依赖V2V模式, 忽略了RSU等基础设施的辅助作用, 在面对高任务量和严格时延要求时, 车辆的移动性导致通信链路的不稳定性也可能对系统性能产生负面影响。因此, 同时利用V2I和V2V的卸载模式, 可以有效克服这些固定模式的局限性, 实现系统性能的优化, 并显著提升用户的QoS。

在探索将两种卸载模式结合进行优化方面, 当前已有多项研究取得进展。文献[15]利用V2I卸载模式, 通过将RSU作为中继, 将移动车辆与RSU集成, 为用户提供服务, 并提出了一种基于粒子群算法的可靠任务分配机制, 以部分卸载的方式最大化任务的可靠性。文献[16]构建了一种软件定义网络(software defined network, SDN)辅助的车载MEC网络架构, 该架构同时考虑V2V和V2I模式且假设任务不可分割, 将卸载决策表述为潜在博弈, 而计算资源、传输功率和上行传输子信道的分配则采用无状态Q学习方法优化, 在约束条件下最小化系统开销。文献[17]则在车联网场景下, 基于二进制卸载提出了一种协同任务卸载、信道分配和计算资源分配的方案, 车辆可灵活选择将任务本地处理、V2I模式或V2V模式, 该方案在任务时延限制和车辆通信链路维持时间约束下, 通过广义Benders分解(gen-



eralized Benders decomposition, GBD) 和重构线性化 (reformulated linearization, RL) 方法进行优化, 以最小化所有车辆的任务处理总时延。文献[18]构建了一个多边缘多端系统模型, 为差异化处理异构任务, 通过基于优先级和车到边缘距离的聚类方法来确定卸载策略, 并利用拉格朗日乘子法优化计算资源分配, 以降低系统总时延并提高执行效率。文献[19]构建了一个基于 V2V、V2I 和车辆到行人 (vehicle-to-pedestrian, V2P) 的智能超表面 (reconfigurable intelligent surface, RIS) 辅助的计算卸载车联网, 考虑所有可提供计算服务的车辆、行人和 MEC 服务器, 根据网络状态、任务优先级和任务量构建效用函数, 提出了一种基于马尔可夫博弈的新型多智能体深度强化学习解决方案, 用于优化任务卸载决策策略, 旨在最大化车联网的平均效用。

上述工作集中于协同优化卸载决策和任务分配比例或只是对某个特定方面如卸载决策或任务分配比例进行优化, 多数工作聚焦在二进制卸载模式下的卸载决策和传输功率分配的协同优化或信道资源分配和计算资源的协同优化, 少数研究则同时关注在二进制卸载模式下的卸载策略、信道资源分配、功率分配和计算资源分配的协同优化。然而, 在单一车辆需要同时执行多个异构任务时, 鲜有工作在考虑车辆移动性的同时对卸载策略、任务拆分以及 RSU 和车辆的空闲计算资源进行协同优化。此外, 随着车联网应用数量的增长, 能源消耗成为一个巨大挑战, 保证应用的 QoS 要求且最小化总能耗, 成为一个重要优化目标^[20]。车联网通常采用基于 IEEE 802.11p 无线通信技术的竞争接入方式^[21], 这为部分卸载方式提供了合理的实现方式。与二进制卸载相比, 部分卸载方式将任务数据划分成局部和边缘计算两部分, 可充分利用车辆自身资源和 VEC 服务器资源^[6,22]。因此, 本文探讨任务的分配比例和提供计算服务节点的计算资源可用性, 并将任务卸载

到最佳的车辆和 RSU 上, 从而最大限度降低系统总能耗。本文的主要贡献如下。

(1) 提出了一个由单个 RSU 和多个可提供计算服务的空闲车辆组成的场景, 首次探讨了一辆车会产生多个异构型任务的情况, 每个任务可灵活地选择在本地计算其任务或者通过 V2V/V2I 方式卸载一部分任务到空闲车辆/RSU 进行计算, 任务由数据量、时延容忍度和任务复杂度来描述。

(2) 对选择决策、任务卸载比例和计算资源分配进行协同优化, 旨在最小化系统能耗, 同时, 考虑了车辆移动性带来的通信链路有限的维持时间和任务的时延约束。优化问题为一个混合整数非线性规划 (mixed-integer nonlinear programming, MINLP) 问题, 由于求解难度较高, 将原问题解耦成多个子问题, 以迭代的方式求解, 提出了一种能量感知的协同选择决策、卸载比例和计算资源的优化方案。

(3) 针对凸性的计算资源的优化问题采用了拉格朗日乘子和梯度法进行求解; 针对选择决策的优化则通过将原问题转换为旅行商问题 (traveling salesman problem, TSP), 采用蚁群系统 (ant colony system, ACS) 算法进行求解; 针对任务卸载比例的优化, 推导出了相应的闭式表达式。由此得到了原问题的最优解, 仿真结果显示, 所提方案相较于 4 个对比方案具有明显优越的性能。

1 系统模型和问题形成

1.1 系统模型

本文考虑一个双向城市快速路系统场景, 包含一个布置于道路旁边并配置有边缘服务器的 RSU, 以及多辆具备 VEC 能力沿着公路双向行驶的移动车辆, 系统模型如图 1 所示。为更好地反映实际应用, 假设在某个很短的时间段内, 系统中有一个车辆生成多个数据量、计算复杂度以及时延约束存在差异的时延敏感型异构任务, 每个

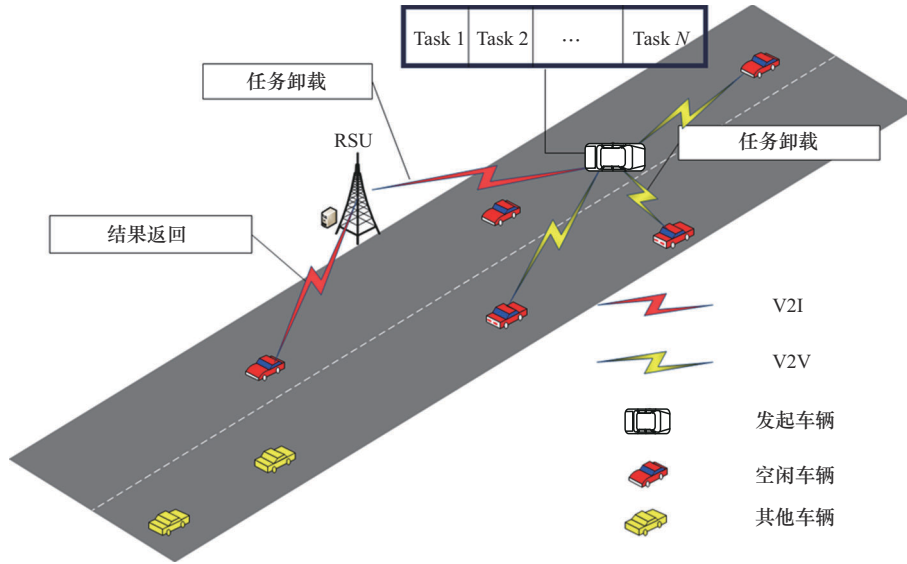


图1 系统模型

任务卸载的任务类型可以是各种车联网应用，包括图像或视频识别、导航复制、碰撞和障碍物检测、自动驾驶等。假设每个任务可拆分为两个部分，分别由发起车辆本地计算以及通过 V2I 卸载至 RSU 或通过 V2V 卸载至其他车辆并行执行。为清楚起见，本文定义了以下名词。

- 发起车辆：发起任务的车辆。
- 空闲车辆：提供计算服务的其他车辆。
- 计算实体：RSU 和空闲车辆的总称。
- 本地任务：一个任务拆分后，由发起车辆本地计算的部分，可以为零。
- 卸载任务：一个任务拆分后，通过卸载由计算实体计算的部分，可以为零。

假设发起车辆 u 生成了 N 个任务，任务编号集合为 $\mathcal{N} = \{1, \dots, N\}$ ，任务 $i (i \in \mathcal{N})$ 可用 $h_i = \{d_i, t_i^{\text{cons}}, c_i\}$ 表示， d_i 表示任务量，单位为 KB； t_i^{cons} 表示接受的最大任务处理时延，单位为 s； c_i 表示复杂度，定义为任务数据量所需的计算量，单位为 cycle/byte。

假设发起车辆 V2V 通信范围内有 M 个空闲车辆，计算实体的编号集合为 $\mathcal{V} = \{0, 1, \dots, M\}$ ，0 代表 RSU，其他值代表空闲车辆。对于任务 h_i ，定

义选择决策变量 $a_{ij} \in \{0, 1\}$ 和卸载比例 $\beta_{ij} \in \{0, 1\}$ ，当选择计算实体 $j (j \in \mathcal{V})$ 进行任务卸载时， $a_{ij} = 1$ ，否则， $a_{ij} = 0$ ；卸载任务的任务量为 $\beta_{ij} d_i$ ，本地任务的任务量为 $(1 - \beta_{ij}) d_i$ 。

每个任务的处理时延和能耗受到 3 个因素的影响，分别是车辆的移动性、任务的传输和任务的计算。车辆的移动性指的是空闲车辆从初始位置移动到发起车辆的通信覆盖范围外的时间，以及每个移动车辆（包括发起车辆）从初始位置移动到 RSU 的通信覆盖范围外的时间；任务的传输方面，发起车辆通过无线信道传输任务到计算实体 $j (j \in \mathcal{V})$ 所需的时间和能耗，取决于所分配的任务量大小；任务的计算，计算时间取决于每个计算实体分配的计算资源和分配的任务量。

1.2 车辆的移动性

假设 RSU 的坐标为原点 $(x_0, y_0) = (0, 0)$ ，车辆 i 的初始位置为 (x_i, y_i) ， $i \in \{u, 1, \dots, M\}$ 。 R 为 RSU 通信范围半径，车辆的位置值在 x_0 的左边为负，右边为正。假设车辆 i 以恒定速率 $|v_i|$ 行驶，当其从左向右沿着道路行驶的时候， v_i 的值是正的，否则为负值，可得到车辆 i 在驶出 RSU 范围



前的剩余距离:

$$e_i = \sqrt{R^2 - (y_i - y_0)^2} - x_i \frac{v_i}{|v_i|} \quad (1)$$

其中, $v_i/|v_i|$ 表示车辆 i 的移动方向。可得到车辆 i 和 RSU 之间 V2I 通信的最大持续时间:

$$t_{i0}^{\text{hold}} = \begin{cases} \frac{e_i}{|v_i|}, & e_i > 0 \\ 0, & e_i \leq 0 \end{cases} \quad (2)$$

假设任意两车之间 V2V 通信的最大距离为 r , 发起车辆 u 和其范围内的空闲车辆 j 之间通信的最大持续时间为:

$$t_{uj}^{\text{hold}} = \begin{cases} \frac{\sqrt{r^2 - (y_u - y_j)^2} - \frac{v_j}{|v_j|}(x_j - x_u)}{|v_j - v_u|}, & \frac{v_u}{|v_u|} + \frac{v_j}{|v_j|} = 0, j \in \mathcal{V} \setminus \{0\} \\ \frac{\sqrt{r^2 - (y_u - y_j)^2} - \frac{v_j}{|v_j|}(x_j - x_u)}{|v_j - v_u|}, & \frac{v_u}{|v_u|} + \frac{v_j}{|v_j|} \neq 0, |v_j| > |v_u|, j \in \mathcal{V} \setminus \{0\} \\ \frac{\sqrt{r^2 - (y_u - y_j)^2} - \frac{v_j}{|v_j|}(x_u - x_j)}{|v_j - v_u|}, & \frac{v_u}{|v_u|} + \frac{v_j}{|v_j|} \neq 0, |v_j| \leq |v_u|, j \in \mathcal{V} \setminus \{0\} \end{cases} \quad (3)$$

其中, 分段函数第一项表示两车的速度方向相反, 第二项表示两车速度方向相同且空闲车辆速度大于发起车辆, 第三项表示两车速度方向相同且空闲车辆速度小于发起车辆。

考虑车辆在执行任务期间以恒定速度行驶, 不失一般性, 假设车辆的速度服从均值为 ε , 方差为 σ^2 的截断高斯函数^[23-24]:

$$f(v) = \begin{cases} \frac{\exp\left(-\frac{(v-\varepsilon)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2} \left(\operatorname{erf}\left(\frac{v_{\max}-\varepsilon}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{v_{\min}-\varepsilon}{\sigma\sqrt{2}}\right) \right)}, & v_{\min} \leq |v| \leq v_{\max} \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中, $\operatorname{erf}(\cdot)$ 是高斯误差函数, $|v_i|$ 处在最大速度 $v_{\max}$ 和最小速度 $v_{\min}$ 之间。

1.3 通信模型

当发起车辆通过 V2I 或 V2V 将一个子任务卸载给 RSU 或空闲车辆计算时, 传输过程包含两部分时延: 任务卸载的传输时延和任务计算结果回传的传输时延。由于任务计算结果的数据很少,

本文未考虑任务结果回传的传输时延和能耗^[14]。

考虑文中系统为基于正交频分复用 (orthogonal frequency division multiplexing, OFDM) 无线传输系统。由于 OFDM 系统子信道之间的正交性, 各 V2I 和 V2V 信道之间不存在无线干扰。基于香农定律, 发起车辆上传子任务给 RSU 或空闲车辆的无线链路传输速率可表示为:

$$R_{ij} = B_{ij} \log(1 + \text{SNR}_{ij}), i \in \mathcal{N}, j \in \mathcal{V} \quad (5)$$

其中, B_{ij} 表示将任务 h_i 发送给计算实体 j 时分配的带宽。相应的信噪比 (signal-to-noise ratio, SNR) 表示为:

$$\text{SNR}_{ij} = \frac{P_i |\mathcal{H}_{uj}|^2 d_{ij}^{-\alpha}}{N_0} \quad (6)$$

其中, P_i 为发送任务 h_i 时的功率, $\mathcal{H}_{uj}$ 表示发起车辆 u 和计算实体 j 之间的小尺度瑞利衰落系数, 服从复高斯分布 $\text{CN}(0, 1)$, d_{ij} 为发起车辆和计算实体 j 之间的距离, α 为大尺度路径损耗因子, N_0 为加性高斯白噪声功率。

1.4 任务的传输和计算过程

关于处理任务 h_i 所需的时延, 具体如下。

(1) 处理本地任务所需的计算时延可表示为:

$$t_{iu}^{\text{loc}} = a_{ij} \frac{(1 - \beta_{ij}) c_i d_i}{f_{iu}}, i \in \mathcal{N}, j \in \mathcal{V} \quad (7)$$

其中, f_{iu} 表示发起车辆处理任务 h_i 的本地任务时的计算资源, 单位为 cycle/s, 并且 t_{iu}^{loc} 需要满足最大任务处理时延 t_i^{cons} :

$$\text{C1: } t_{iu}^{\text{loc}} \leq t_i^{\text{cons}}, i \in \mathcal{N} \quad (8)$$

(2) 处理卸载任务需要的时延包括发起车辆将任务 h_i 的卸载任务卸载给计算实体 j 的传输时延和计算实体 j 的计算时延, 表示为:

$$t_{ij}^{\text{off}} = a_{ij} \left(\frac{\beta_{ij} d_i}{R_{ij}} + \frac{\beta_{ij} c_i d_i}{f_{ij}} \right), i \in \mathcal{N}, j \in \mathcal{V} \quad (9)$$

其中, R_{ij} 表示将任务 h_i 的卸载任务传输给计算实体 j 的传输速率, f_{ij} 表示计算实体 j 处理卸载任务时的计算资源。

同样, 卸载任务时延 t_{ij}^{off} 也需要满足最大任务处理时延 t_i^{cons} 。当任务 h_i 利用 V2I 链路卸载给 RSU 时, t_{i0}^{off} 需要同时满足 t_i^{cons} 以及 V2I 链路的持续时间 t_{u0}^{hold} , 即:

$$\text{C2: } t_{i0}^{\text{off}} \leq \min(t_{u0}^{\text{hold}}, t_i^{\text{cons}}), i \in \mathcal{N} \quad (10)$$

当任务 h_i 利用 V2V 链路卸载给空闲车辆时, 需要保证传输时延满足 V2V 链路持续时间 t_{ij}^{hold} :

$$\text{C3: } a_{ij} \frac{\beta_{ij} d_i}{R_{ij}} \leq t_{ij}^{\text{hold}}, i \in \mathcal{N}, j \in \mathcal{V} \setminus \{0\} \quad (11)$$

在空闲车辆处理完任务 h_i 的卸载任务后, 可通过两种通信方式将计算结果传回, 即 V2V 通信原路传回或者 V2I 通过 RSU 作为中继返回结果, 后者需要保证发起车辆和空闲车辆都在 RSU 通信范围内。此时, t_{ij}^{off} 需要满足最大任务处理时延 t_i^{cons} , 同时也应满足发起车辆和空闲车辆之间 V2I 中继链路持续时间 $\min(t_{u0}^{\text{hold}}, t_{j0}^{\text{hold}})$ 和 V2V 链路持续时间 t_{ij}^{hold} 。

$$\text{C4: } t_{ij}^{\text{off}} \leq \min(t_i^{\text{cons}}, \max(t_{ij}^{\text{hold}}, \min(t_{u0}^{\text{hold}}, t_{j0}^{\text{hold}}))), \\ i \in \mathcal{N}, j \in \mathcal{V} \setminus \{0\} \quad (12)$$

卸载计算中消耗的总能量包括在发起车辆上用于任务传输的能量和在计算实体 j 和发起车辆上用于处理的能量。参考文献[25]可知, 计算实体 j 和发起车辆的处理器在处理各自任务时的功率分别为 $k(f_{ij})^3$ 和 $k(f_{iu})^3$, 单位为 J/s, 其中 k 为处理器芯片能耗系数^[26], 与任务处理时间进行相乘即可得出计算实体 j 处理任务的能耗 E_{ij}^{exe} 和发起车辆处理任务的能耗 E_{iu}^{exe} 为:

$$E_{ij}^{\text{exe}} = a_{ij} k (f_{ij})^2 c_i \beta_{ij} d_i \quad (13)$$

$$E_{iu}^{\text{exe}} = a_{ij} k (f_{iu})^2 c_i (1 - \beta_{ij}) d_i \quad (14)$$

传输任务所消耗的能量为发起车辆的发射功率乘以传输任务所需的时间:

$$E_{ij}^{\text{off}} = a_{ij} P_i \frac{\beta_{ij} d_i}{R_{ij}} \quad (15)$$

1.5 问题形成

通过协同优化选择决策、卸载比例和计算资源, 可形成一个基于能量感知的协同资源分配和选择决策的优化问题, 求解的目标是最小化所有任务的总能耗 E 。本文假设所有计算实体的实时的能量信息对系统是已知的, 即算法是基于能量感知的, 而能量感知的具体实现方式不在本文的研究范围内。定义 $\mathcal{A} = \{a_{ij}\}$ 为对计算实体进行选择的选择决策向量, $\mathcal{B} = \{\beta_{ij}\}$ 为任务 h_i 卸载给计算实体的任务量的比例向量, $\mathcal{F} = \{f_{ij}\}$ 为计算实体计算所分配任务的计算资源向量, $\mathcal{F} = \{f_{iu}\}$ 为发起车辆本地计算所分配任务时计算资源向量。最终, 形成的优化问题如下所示:

$$\begin{aligned} \text{P1: } \min E(\mathcal{A}, \mathcal{B}, \mathcal{F}, \mathcal{F}) &= \sum_{i \in \mathcal{N}} E_i(a_{ij}, \beta_{ij}, f_{ij}, f_{iu}) = \\ &= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \\ \text{s.t. } &\text{C1} \sim \text{C4} \\ \text{C5: } &\sum_{j \in \mathcal{V}} a_{ij} = 1, i \in \mathcal{N} \\ \text{C6: } &a_{ij} \in \{0, 1\}, i \in \mathcal{N}, j \in \mathcal{V} \end{aligned}$$



$$\begin{aligned}
\text{C7: } & 0 \leq \beta_{ij} \leq 1, i \in \mathcal{N}, j \in \mathcal{V} \\
\text{C8: } & \sum_{i \in \mathcal{N}} f_{iu} \leq f_u^{\max} \\
\text{C9: } & f_{iu} \geq 0, i \in \mathcal{N} \\
\text{C10: } & \sum_{i \in \mathcal{N}} a_{ij} f_{ij} \leq f_j^{\max}, j \in \mathcal{V} \\
\text{C11: } & a_{ij} f_{ij} \geq 0, i \in \mathcal{N}, j \in \mathcal{V}
\end{aligned} \tag{16}$$

约束C1~C4如上说明。约束C5确保任务只会选择一个且只有一个计算实体。约束C6将二进制变量 a_{ij} 限定在0或1。约束C7表示卸载比例范围为0~1。约束C8和约束C10分别确保发起车辆和计算实体为任务所分配的计算资源总和不超过它们的最大计算资源 $f_u^{\max}$ 和 $f_j^{\max}$ 。约束C9和约束C11分别为分配的计算资源，不会是负数。

在P1中，目标函数因存在二进制变量变得非线性，其他变量存在乘积关系，且约束C6也使得可行集变成了非凸。由此，问题变成一个MINLP问题，是非凸且NP难的，这使得求解该问题变得非常困难。

2 所提解决方法

原问题P1是一个MINLP问题，无法通过变换等方式转化为凸问题直接求解，通常可以将其解耦成子问题，通过迭代优化收敛来求解。需要注意的是，此时得到的解并不等价于原问题的最优解，而只是原问题的一个次优解。由于原问题P1的复杂性，可先将其解耦成4个子问题，分别为选择决策问题、计算实体计算资源分配问题、发起车辆计算资源分配问题、卸载比例问题 E ，再进行迭代优化求解。选择决策问题利用ACS算法优化，计算资源的分配问题则利用拉格朗日乘子和梯度法来优化，卸载比例问题根据其单调性通过闭式表达式得到。接下来对各子问题的解法进行详细阐述。

2.1 选择决策的优化

对于给定 $\mathcal{B}$ 、 $\mathcal{F}$ 、 $\mathcal{F}$ 之后的选择决策问题，参考问题P1后可形成问题P1.1：

$$\begin{aligned}
\text{P1.1: } & \min E(\mathcal{A}) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \\
& \left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \tag{17} \\
& \text{s.t. C1} \sim \text{C4, C5, C6, C10, C11}
\end{aligned}$$

ACS算法是一种元启发式算法，通常用来解决旅行商问题，即TSP，P1.1是一个广义指派问题（generalized assignment problem, GAP），可以将GAP转换为TSP，利用ACS算法来求解。ACS算法的基本原理：蚂蚁之间用信息素进行交流合作，在行进的途中会根据信息素对下一个路径进行选择，同时留下自己的信息素，并且信息素的浓度会随着时间推移挥发。所选择的路径越短，留下的信息素浓度越大，信息素浓的路径会吸引其他蚂蚁选择，而更多蚂蚁选择的路径的信息素会得到强化，表现出信息的正反馈现象，指导蚂蚁找到一条从蚁穴到食物源的最小路径。

TSP可表示为一个图论问题，其中每个城市表示图中的节点，城市之间的距离映射为边权重。问题目标是确保每个节点仅被访问一次，且回路的总权重达到最小。ACS算法中，蚂蚁每次根据信息素浓度选择下一个节点，从而在当前节点与所选节点间形成一条边，遍历所有节点后，就创建出了一条完整的合法路径，形成一条回路。

为了用ACS算法来解决GAP，需要先将GAP转为ACS算法能够解决的TSP。首先，定义集合 $\mathcal{N} \cup \mathcal{V}$ 属于 $\{1, \dots, N+M+1\}$ ， $1 \sim N$ 表示任务节点编号， $(N+1) \sim (N+M+1)$ 表示计算节点编号。GAP表述成一个二分图 $G = (\mathcal{N} \cup \mathcal{V}, \mathcal{N} \cup \mathcal{V}, \psi)$ ，将集合 $\mathcal{N}$ 中的任务、集合 $\mathcal{V}$ 中的计算实体都作为图中的节点， $\psi = \{\psi_{\mathcal{I}\mathcal{J}} | \mathcal{I} \in \mathcal{N} \cup \mathcal{V}, \mathcal{J} \in \mathcal{N} \cup \mathcal{V}\}$ 表示连接节点 $\mathcal{I}$ 和节点 $\mathcal{J}$ 的边权重。任务节点到计算实体节点的边权重设定为任务卸载的能耗，计算实体节点到任务节点的边权重设定为0，以诱导算法增加此类

状况的选择概率,任务与任务或者计算实体与计算实体之间的边权重设定为一个非常大的常数,以诱导算法减少此类状况的选择概率。

解决问题P1.1的ACS算法的框架表述如下。

(1) 信息的初始化

η_{IJ} 为启发信息,表示从节点 I 到节点 J 这条边的启发级别。当 $I \in [N+1, N+M+1]$ 或 $J \in [1, N]$ 时,取值可以是一个很大的常数,其他情况取值为 $\frac{1}{\psi_{IJ}}$ 。用 τ_{IJ} 表示节点 I 到节点 J 这条边的信息素浓度,它在各条边对应的初始值设定有如下表示:

$$\tau_{IJ} = \begin{cases} \frac{1}{N \cdot \psi_{IJ}}, & I \in [1, N], J \in [N+1, N+M+1] \\ 1, & I \in [N+1, N+M+1], J \in [1, N] \\ 0, & \text{其他} \end{cases} \quad (18)$$

假设蚁群有 m 只蚂蚁,编号表示为 $\mathcal{L}(\mathcal{L}=1, 2, \dots, m)$,将它们随机置于代表任务的节点序号上作为起始位置,初始化每只蚂蚁的节点编号列表,即此时列表中包含的是蚂蚁 $\mathcal{L}$ 未选择的节点序号。

(2) 边的构建

每一次的循环迭代中,每只蚂蚁 $\mathcal{L}$ 根据边上的信息素浓度和启发信息独立选择需要访问的下一个节点。为避免多次访问同一个任务节点,满足一个任务只能分发给一个实体的要求,当蚂蚁 $\mathcal{L}$ 选择的下一个节点是任务节点时,将该节点序号从节点编号列表中删除,不作为下一个节点的选择对象。

选择策略:蚂蚁 $\mathcal{L}$ 从剩余的节点列表中选择下一个未访问节点,按照如下规则选择:

$$\mathcal{J} = \begin{cases} \arg \max_{s \in V_{\mathcal{L}}(I)} \left[(\tau_{Is})^\alpha (\eta_{Is})^\beta \right], & q \leq q_0 \\ \text{根据式(20),其他} \end{cases} \quad (19)$$

其中, $V_{\mathcal{L}}(I)$ 表示蚂蚁 $\mathcal{L}$ 在访问节点 I 时可以选择的未访问节点的集合, α 为信息素影响因子,反映了蚂蚁在选择边时积累的信息素的影响程度, β 为启发信息影响因子, q 为 $[0,1]$ 均匀分布的随机数, q_0 为算法给定的概率参数。

算法根据各条边上的信息素浓度和启发信息来计算状态转移概率,并以轮盘赌的方式确定访问节点。蚂蚁 $\mathcal{L}$ 由已访问节点 I 到未访问节点 J 的状态转移概率 $p_{IJ}^{\mathcal{L}}$ 可以表示为:

$$p_{IJ}^{\mathcal{L}} = \begin{cases} \frac{\left[(\tau_{IJ})^\alpha (\eta_{IJ})^\beta \right]}{\sum_{s \in V_{\mathcal{L}}(I)} \left[(\tau_{Is})^\alpha (\eta_{Is})^\beta \right]}, & J \in V_{\mathcal{L}}(I) \\ 0, & \text{其他} \end{cases} \quad (20)$$

(3) 更新信息素

更新局部信息素:当蚂蚁 $\mathcal{L}$ 从节点 I 到下一个节点 J 后,边 (I, J) 上的信息素进行如下更新^[27]:

$$\tau_{IJ} = (1 - \gamma) \cdot \tau_{IJ} + \gamma \cdot \tau_0 \quad (21)$$

其中, γ 为给定参数,范围是 $(0, 1)$, τ_0 为各边的初始信息素。每当蚂蚁选中一个节点后,按照式(21)进行更新,增加其他蚂蚁选择该节点的概率。

更新全局信息素:为实现算法探索性和收敛性的有效折中,考虑子边在形成最优合法回路中的独立作用,在每次迭代后不仅更新本次迭代最优回路的子边的信息素,同时应更新经过蚂蚁数量多的子边的信息素,采取迭代最优和排序较优子边相结合的更新信息素方式^[28]。

在蚂蚁数量为 m 的蚁群中,假设子边 edge_{IJ} 经过的蚂蚁数为 edge_{IJ}^m ,对各子边已经过的蚂蚁数量进行排序,取前 l 条较优子边进行信息素更新,更新的全局信息素和子边的信息素增量按照式(22)计算^[28]:



$$\begin{cases} \tau_{IJ} = (1 - \rho) \cdot \tau_{IJ} + \rho \cdot \Delta\tau_{IJ} \\ \Delta\tau_{IJ} = \frac{1}{E_{\text{iterbest}}} + \zeta \end{cases} \quad (22)$$

$$\zeta = \begin{cases} Q, \text{ 小于 } l \\ 0, \text{ 其他} \end{cases} \quad (23)$$

其中, $Q = \frac{1}{E_{\text{iterbest}}}$, E_{iterbest} 为在迭代过程中当前代的最佳回路所形成的系统能耗, ρ 为信息素挥发系数, $\Delta\tau_{IJ}$ 表示本次迭代中节点 I 到节点 J 这条边的信息素增量, ζ 表示排序后子边信息素增量。

信息素增量的计算和信息素更新流程如下。

步骤1 初始化记录各子边的蚂蚁经过数量的计数矩阵, 矩阵元素设置为0。

步骤2 本文只对任务节点到计算实体节点的边计数。遍历所有蚁群个体, 用计数矩阵对各个蚂蚁走的回路中所包含的子边进行计数, 经过的子边在计数矩阵相应的位置加1, 否则不做变更。

步骤3 将计数矩阵中各边按照各自经过的蚂蚁数量从大到小排列, 取前 l 条子边按照式 (22) 更新相应子边的信息素。

步骤4 计数矩阵元素清零, 流程结束。

在进行完排序较优子边的信息素更新后, 再在本次迭代最优回路的子边上按照式 (22) 进行信息素的更新。

在ACS算法中, 信息素分布是种群有效进化的外在因素, 前文基于较优子边排序的主要作用是微调整体信息素布局, 因此为促进算法收敛和避免局部最优, 可以使用自适应机制对信息素调控。在算法初期, 通过多数的较优子边参与信息素的更新使解空间的信息素均衡分布, 趋于全局搜索; 在算法后期, 通过少量较优子边参与信息素更新, 提高算法收敛速度^[5]。 l 的取值定义为:

$$l = \left(1 - \frac{\text{iter}}{\text{iterMax}}\right) \cdot \Psi \quad (24)$$

其中, Ψ 为当前代蚂蚁走过的子边的数量, iter 为当前迭代次数, iterMax 代表最大迭代次数。

详细的选择决策算法过程如算法1所示。当任务节点 $I \in [1, N]$ 选择计算实体节点 $J \in [N+1, N+M+1]$ 时, f_{J-N}^r 表示计算实体 ($J-N$) 的剩余计算资源, $f_{I(J-N)}$ 表示计算实体 ($J-N$) 分配给任务 I 的计算资源。

算法1 ACS优化A

输入 给定可行解 $\mathcal{B}$ 、 $\mathcal{F}$ 、 $\mathcal{F}$, 初始化边权重 ψ 、启发信息 η_{IJ} 、信息素浓度 τ_{IJ}

输出 选择决策 $\mathcal{A}$

设定迭代最大次数 iterMax , 参数 γ , 信息素挥发系数 ρ , 信息素影响因子 α , 启发信息影响因子 β

while ($\text{iter} \leq \text{iterMax}$)

将 m 只蚂蚁的初始点随机放在代表任务的节点上, 并生成 m 只蚂蚁各自的回路节点列表

repeat

各计算实体计算资源初始化为各自具有的最大计算资源

while (1)

I 为蚂蚁 $\mathcal{L}$ ($\mathcal{L} = 1, 2, \dots, m$) 回路节点列表中当前所处的节点序号, Ω 表示当前可访问的节点序号集合

for $J \in \Omega$

if $I \in [1, N] \&\& J \in [N+1, N+M+1]$

if $f_{J-N}^r - f_{I(J-N)} \geq 0$

if J 为RSU节点序号

同时满足约束C1和C2, 启发信息 η_{IJ} 值不变; 否则, η_{IJ} 取一个很小的常数

else

同时满足约束C1、C3和C4, 启发信息 η_{IJ} 值不变; 否则, η_{IJ} 取一个很小的常数

end

```

else
     $\eta_{\mathcal{I}\mathcal{J}}$ 取一个很小的常数
end
else
    启发信息  $\eta_{\mathcal{I}\mathcal{J}}$ 值不变
end
end
end
使用式 (19) 确定下一个节点, 并存在
蚂蚁  $\mathcal{L}$  回路节点列表中
if  $\mathcal{I} \in [1, N]$  &&  $\mathcal{J} \in [N+1, N+M+1]$ 
    使用式 (21) 更新信息素, 并减去
    分配给任务  $\mathcal{I}$  的计算资源  $f_{\mathcal{I}(\mathcal{J}-N)}$ , 得到计算实体
    ( $\mathcal{J}-N$ ) 最终剩余的计算资源  $f_{\mathcal{J}-N}^r$ 
    else if  $\mathcal{I} \in [N+1, N+M+1]$  &&  $\mathcal{J} \in [1, N]$ 
        在集合  $\mathcal{Q}$  中剔除掉此时  $\mathcal{J}$  代表的任
        务节点序号
    end
    if  $\mathcal{Q}$  中无任务节点序号
        break;
    end
end
until 遍历  $m$  只蚂蚁
    找出当前代蚁群中最优蚂蚁的回路和相
    应的系统能耗值, 并与当前全局最优的回路比
    较, 更新全局最优蚂蚁的回路。
    采用上文提到的迭代最优和排序较优子边相
    结合的方式更新全局信息素。
    iter  $\leftarrow$  iter + 1;
end
Return 根据最优回路得出最优选择
决策  $\mathcal{A}$ 

```

2.2 卸载比例的优化

在所给定的 $\mathcal{A}$ 、 $\mathcal{F}$ 、 $\mathcal{F}$ 基础上形成的卸载比
例优化问题可由问题 P1.2 表示:

$$\begin{aligned}
 \text{P1.2: } \min E(\mathcal{B}) &= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \\
 &\left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \quad (25) \\
 \text{s.t. } &\text{C1} \sim \text{C4}, \text{C7}
 \end{aligned}$$

引理 1:

问题 P1.2 中的约束 C1、C2、C4, 可以有以
下表示:

$$-\frac{f_{iu} t_i^{\text{cons}}}{a_{ij} c_i d_i} \leq \beta_{ij} \leq \frac{T_{ij}^{\text{cons}} \cdot f_{ij} R_{ij}}{a_{ij} d_i (c_i R_{ij} + f_{ij})}, i \in \mathcal{N}, j \in \mathcal{V} \quad (26)$$

其中, 当 $j \in \mathcal{V} \setminus \{0\}$ 时, $T_{ij}^{\text{cons}} = \min(t_i^{\text{cons}}, \max(t_{uj}^{\text{hold}}, \min(t_{u0}^{\text{hold}}, t_{j0}^{\text{hold}})))$; 当 $j \in \{0\}$ 时, $T_{ij}^{\text{cons}} = \min(t_{u0}^{\text{hold}}, t_i^{\text{cons}})$ 。

证明:

给定一个卸载比例 β_{ij} , 当卸载给计算实体
时, 卸载计算所需时间为:

$$a_{ij} \left(\frac{\beta_{ij} d_i}{R_{ij}} + \frac{\beta_{ij} c_i d_i}{f_{ij}} \right) \leq T_{ij}^{\text{cons}} \Rightarrow \beta_{ij} \leq \frac{T_{ij}^{\text{cons}} \cdot f_{ij} R_{ij}}{a_{ij} d_i (c_i R_{ij} + f_{ij})}, i \in \mathcal{N}, j \in \mathcal{V} \quad (27)$$

发起车辆本地计算所需时间为:

$$a_{ij} \frac{(1 - \beta_{ij}) c_i d_i}{f_{iu}} \leq t_i^{\text{cons}} \Rightarrow \beta_{ij} \geq -\frac{f_{iu} t_i^{\text{cons}}}{a_{ij} c_i d_i}, i \in \mathcal{N}, j \in \mathcal{V} \quad (28)$$

式 (27) 和式 (28) 合并得到式 (26), 证毕。

此时, 问题 P1.2 可以分解成独立优化的 N 个
子问题, 利用引理 1, 可得到如下问题:

$$\begin{aligned}
 \text{P1.2.1: } \min_{\beta_{ij}} E_i(\beta_{ij}) &= \\
 &\sum_{j \in \mathcal{V}} a_{ij} \left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \\
 \text{s.t. } &-\frac{f_{iu} t_i^{\text{cons}}}{a_{ij} c_i d_i} \leq \beta_{ij} \leq \frac{T_{ij}^{\text{cons}} f_{ij} R_{ij}}{a_{ij} d_i (c_i R_{ij} + f_{ij})}, \\
 &i \in \mathcal{N}, j \in \mathcal{V}, \text{C3}, \text{C7} \quad (29)
 \end{aligned}$$



对问题 P1.2.1 的目标函数求关于 β_{ij} 的一阶导数:

$$\frac{\partial E_i(\beta_{ij})}{\partial \beta_{ij}} = a_{ij} k f_{ij}^2 c_i d_i + a_{ij} P_i \frac{d_i}{R_{ij}} - a_{ij} k f_{iu}^2 c_i d_i \quad (30)$$

$$\beta_{ij}^* = \begin{cases} \max \left\{ 1 - \frac{f_{iu} t_i^{\text{cons}}}{a_{ij} c_i d_i}, 0 \right\}, E_{ij}^{\text{off}} \geq E_{iu}^{\text{loc}}, i \in \mathcal{N}, j \in \mathcal{V} \\ \min \left\{ \frac{T_{ij}^{\text{cons}} f_{ij} R_{ij}}{a_{ij} d_i (c_i R_{ij} + f_{ij})}, \frac{t_{ij}^{\text{hold}} R_{ij}}{a_{ij} d_i}, 1 \right\}, E_{ij}^{\text{off}} < E_{iu}^{\text{loc}}, i \in \mathcal{N}, j \in \mathcal{V} \setminus \{0\} \\ \min \left\{ \frac{T_{ij}^{\text{cons}} f_{ij} R_{ij}}{a_{ij} d_i (c_i R_{ij} + f_{ij})}, 1 \right\}, E_{ij}^{\text{off}} < E_{iu}^{\text{loc}}, i \in \mathcal{N}, j \in \{0\} \end{cases} \quad (31)$$

其中, $E_{ij}^{\text{off}} = a_{ij} k f_{ij}^2 c_i d_i + a_{ij} P_i \frac{d_i}{R_{ij}}$ 表示卸载计算所需能耗, $E_{iu}^{\text{loc}} = a_{ij} k f_{iu}^2 c_i d_i$ 表示任务在本地计算所需能耗。

2.3 计算资源的优化

给定 $\mathbf{A}$ 、 $\mathbf{B}$ 、 $\mathbf{F}$ 之后形成的计算实体计算资源优化问题由问题 P1.3 表示:

$$\begin{aligned} \text{P1.3: } \min E(\mathcal{F}) &= \sum_{i \in \mathcal{N}} E_i(f_{ij}) = \\ &\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \\ \text{s.t. C2, C4, C10: } &\sum_{i \in \mathcal{N}} a_{ij} f_{ij} \leq f_j^{\max} \\ j \in \mathcal{V}, \text{ C11: } &a_{ij} f_{ij} \geq 0, i \in \mathcal{N}, j \in \mathcal{V} \end{aligned} \quad (32)$$

P1.3 的目标函数 $E(\mathcal{F})$ 为凸函数, 具体证明由附录 1 所示, 可行集为凸集, 因此, P1.3 是凸问题, 可以利用拉格朗日乘子和梯度法求解。

P1.3 的拉格朗日函数由式 (33) 给出:

$$\begin{aligned} \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda) &= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \\ &\left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) + \\ &\sum_{j \in \mathcal{V}} \zeta_j \left(\sum_{i \in \mathcal{N}} a_{ij} f_{ij} - f_j^{\max} \right) + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} \xi_{ij} (-a_{ij} f_{ij}) + \\ &\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} \lambda_{ij} \left(\frac{a_{ij} \beta_{ij} c_i d_i}{f_{ij}} + \frac{a_{ij} \beta_{ij} d_i}{R_{ij}} - T_{ij}^{\text{cons}} \right) \end{aligned} \quad (33)$$

由式 (30) 可知, 问题 P1.2.1 的目标函数根据一阶导数的正/负号单调递增/递减。通过单调性和可行域, 最优卸载比例 β_{ij}^* 可由式 (31) 得到:

其中, ζ 、 ξ 和 λ 为拉格朗日乘子。拉格朗日对偶问题由式 (34) 给出:

$$\begin{aligned} \max_{\zeta, \xi, \lambda} \min \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda) \\ \text{s.t. } \zeta, \xi, \lambda \geq 0 \end{aligned} \quad (34)$$

本文使用梯度法通过对偶问题来求解原问题。为获得计算实体计算资源最优分配, 求 $\mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)$ 关于 f_{ij} 的一阶导数, 并令其为 0:

$$\frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial f_{ij}} = a_{ij} \zeta_j - a_{ij} \xi_{ij} + \frac{a_{ij} \beta_{ij} c_i d_i}{f_{ij}^2} (2 k f_{ij}^3 - \lambda_{ij}) = 0 \quad (35)$$

因为获得式 (35) 的闭式解是困难的, 所以本文采用二分法来获得计算实体最优计算资源, 具体操作如算法 2 所示。

算法 2 利用二分法求解 f_{ij}

输入 计算实体的最大计算资源 $f_j^{\max}$

输出 最优 f_{ij}

设置 $f_{ij}^{ub} = f_j^{\max}$, $f_{ij}^{lb} = 0$

while $(|f_{ij}^{ub} - f_{ij}^{lb}| > 0.001)$

$\mathbb{X} = \frac{f_{ij}^{ub} + f_{ij}^{lb}}{2}$;

if $\frac{\partial \mathcal{L}(f_{ij}^{lb})}{\partial f_{ij}} \cdot \frac{\partial \mathcal{L}(\mathbb{X})}{\partial f_{ij}} < 0$

$f_{ij}^{ub} = \mathbb{X}$;

else

$$f_{ij}^{lb} = \mathbb{X}$$

end

end

Return f_{ij}

在通过算法2获得计算实体最优计算资源后，应用梯度法对拉格朗日乘子进行迭代更新，如下所示：

$$\begin{aligned}\zeta_j(t+1) &= \left[\zeta_j(t) + s1 \frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \zeta_j} \right]^+ \\ \xi_{ij}(t+1) &= \left[\xi_{ij}(t) + s2 \frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \xi_{ij}} \right]^+ \\ \lambda_{ij}(t+1) &= \left[\lambda_{ij}(t) + s3 \frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \lambda_{ij}} \right]^+\end{aligned}\quad (36)$$

其中， $s1$ 、 $s2$ 、 $s3$ 为步长，为正数， $[\cdot]^+$ 表示 $\max(0, \cdot)$ 。拉格朗日乘子梯度由式 (37) 给出：

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \zeta_j} &= \sum_{i \in \mathcal{N}} a_{ij} f_{ij} - f_j^{\max} \\ \frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \xi_{ij}} &= -a_{ij} f_{ij} \\ \frac{\partial \mathcal{L}(\mathcal{F}, \zeta, \xi, \lambda)}{\partial \lambda_{ij}} &= \frac{a_{ij} \beta_{ij} c_i d_i}{f_{ij}} + \frac{a_{ij} \beta_{ij} d_i}{R_{ij}} - T_{ij}^{\text{cons}}\end{aligned}\quad (37)$$

梯度法优化计算实体计算资源分配的具体步骤如算法3所示。

算法3 梯度法优化 $\mathcal{F}$

输入 计算实体的最大计算资源 $f_j^{\max}, j \in \mathcal{V}$

输出 最优 $\mathcal{F}$

$t=0$ ，初始化拉格朗日乘子 ζ, ξ, λ

repeat

for $i=1:N$

使用式 (36) 更新乘子；

使用算法2求 f_{ij} ；

end

$$\Theta(t) = \sum_{i \in \mathcal{N}} E_i(f_{ij});$$

$t \leftarrow t+1$;

until $|\Theta(t) - \Theta(t-1)| < 0.01$

Return $\mathcal{F}$

给定 $\mathcal{A}$ 、 $\mathcal{B}$ 、 $\mathcal{F}$ 之后形成的发起车辆计算资源优化问题，如问题 P1.4 所示：

$$\begin{aligned}\text{P1.4: } \min E(\mathbb{F}) &= \sum_{i \in \mathcal{N}} E_i(f_{iu}) = \\ &\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) \\ \text{s.t. } &\text{C1, C8, C9}\end{aligned}\quad (38)$$

P1.4 的目标函数 $E(\mathbb{F})$ 为凸函数，具体证明由附录2给出，其可行域为凸集，因此，问题 P1.4 是凸问题。类似地，可采用拉格朗日乘子和梯度法求解。P1.4 的拉格朗日函数由式 (39) 给出：

$$\begin{aligned}L(\mathbb{F}, \omega, \chi, \mu) &= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{V}} a_{ij} \\ &\left(k f_{ij}^2 \beta_{ij} c_i d_i + P_i \frac{\beta_{ij} d_i}{R_{ij}} + (1 - \beta_{ij}) k f_{iu}^2 c_i d_i \right) + \\ &\omega \left(\sum_{i \in \mathcal{N}} f_{iu} - f_u^{\max} \right) + \sum_{i \in \mathcal{N}} \chi_i (-f_{iu}) + \\ &\sum_{i \in \mathcal{N}} \mu_i \left(\frac{a_{ij} (1 - \beta_{ij}) c_i d_i}{f_{iu}} - t_i^{\text{cons}} \right)\end{aligned}\quad (39)$$

其中， ω 、 χ 和 μ 为拉格朗日乘子。相应的拉格朗日对偶问题由式 (40) 给出：

$$\begin{aligned}\max_{\omega, \chi, \mu} \min L(\mathbb{F}, \omega, \chi, \mu) \\ \text{s.t. } \omega, \chi, \mu \geq 0\end{aligned}\quad (40)$$

此处仍旧通过求解对偶问题来获得原问题 P1.4 的解，为获得发起车辆计算资源最优分配问题，求 $L(\mathbb{F}, \omega, \chi, \mu)$ 关于 f_{iu} 的一阶导数，并令其为0：

$$\frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial f_{iu}} = \omega - \chi_i + \frac{a_{ij} (1 - \beta_{ij}) c_i d_i}{f_{iu}^2} (2k f_{iu}^3 - \mu_i) = 0\quad (41)$$

采用二分法来获得发起车辆最优计算资源，具体操作如算法4所示。

算法4 利用二分法求解 f_{iu}



输入 发起车辆的最大计算资源 $f_u^{\max}$

输出 最优 f_{iu}

设置 $f_{iu}^{ub} = f_u^{\max}$, $f_{iu}^{lb} = 0$

while ($|\frac{f_{iu}^{ub}}{f_{iu}^{lb}} - 1| > 0.001$)

$$\mathbb{X} = \frac{f_{iu}^{ub} + f_{iu}^{lb}}{2};$$

$$\text{if } \frac{\partial L(f_{iu}^{lb})}{\partial f_{iu}} \cdot \frac{\partial L(\mathbb{X})}{\partial f_{iu}} < 0$$

$$f_{iu}^{ub} = \mathbb{X};$$

else

$$f_{iu}^{lb} = \mathbb{X};$$

end

end

Return f_{iu}

在通过算法4获得发起车辆最优计算资源后,应用梯度法对拉格朗日乘子进行迭代更新,如下所示:

$$\begin{aligned} \omega(t+1) &= \left[\omega(t) + s4 \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \omega} \right]^+ \\ \chi_i(t+1) &= \left[\chi_i(t) + s5 \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \chi_i} \right]^+ \\ \mu_i(t+1) &= \left[\mu_i(t) + s6 \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \mu_i} \right]^+ \end{aligned} \quad (42)$$

其中, $s4$ 、 $s5$ 、 $s6$ 为步长, 为正数。拉格朗日乘子梯度由式(43)给出:

$$\begin{aligned} \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \omega} &= \sum_{i \in \mathcal{N}} f_{iu} - f_u^{\max} \\ \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \chi_i} &= -f_{iu} \\ \frac{\partial L(\mathbb{F}, \omega, \chi, \mu)}{\partial \mu_i} &= \frac{a_{ij}(1 - \beta_{ij})c_i d_i}{f_{iu}} - t_i^{\text{cons}} \end{aligned} \quad (43)$$

梯度法优化发起车辆计算资源分配的具体步骤如算法5所示。

算法5 梯度法优化 $\mathbb{F}$

输入 发起车辆的最大计算资源 $f_u^{\max}$

输出 最优 $\mathbb{F}$

$t=0$, 初始化拉格朗日乘子 ω , χ , μ

repeat

for $i=1:N$

使用式(42)更新乘子;

使用算法4求 f_{iu} ;

end

$$\Theta(t) = \sum_{i \in \mathcal{N}} E_i(f_{iu});$$

$$t \leftarrow t+1;$$

until $|\Theta(t) - \Theta(t-1)| < 0.01$

Return $\mathbb{F}$

2.4 协同选择决策、卸载比例和计算资源的迭代优化

鉴于问题P1的NP难属性,将P1拆解成选择决策、卸载比例、计算实体计算资源和发起车辆计算资源4个子问题,分别优化再进行迭代求解,每次迭代中,计算相应的优化变量的最优值,在达到最大迭代次数后,优化停止。

因此,本文设计了一个能量感知的协同选择决策、卸载比例和计算资源的优化方案(energy-sensing joint selection decision, offloading ratio and computation resource optimization scheme, EJSOC),如算法6所示。

算法6 EJSOC

初始化各优化变量 $\mathcal{A}$, $\mathcal{B}$, $\mathcal{F}$, $\mathbb{F}$

设定 $t=0$, 最大迭代次数 $t_{\max}$;

repeat

使用算法1优化 $\mathcal{A}$;

使用算法5优化 $\mathbb{F}$;

使用式(31)优化 $\mathcal{B}$;

使用算法3优化 $\mathcal{F}$;

$$\Theta(t) = \sum_{i \in \mathcal{N}} E_i(a_{ij}, \beta_{ij}, f_{ij}, f_{iu})$$

$$t \leftarrow t+1;$$

until $t > t_{\max}$

Return 最优 $\mathcal{A}$, $\mathcal{B}$, $\mathcal{F}$, $\mathcal{F}$

3 性能分析

仿真系统参照图 1, 由一条城市双向快速路、一个发起车辆、一个 RSU 和多个空闲车辆组成。车辆随机向左或右行驶, 假设发起车辆的速度为 16 m/s, 空闲车辆的速度在 $v_{\min}$ 和 $v_{\max}$ 之间随机分布。除非有特别说明, 仿真参数设置见表 1。

表 1 仿真参数设置

参数	值
任务大小 $d_i, i \in \mathcal{N}$	[80, 320] KB
最小速度 $v_{\min}$	16 m/s
最大速度 $v_{\max}$	20 m/s
复杂度 $c_i, i \in \mathcal{N}$	[4000, 7000] cycle/byte
最大容忍时延 $t_i^{\text{cons}}, i \in \mathcal{N}$	[0.35, 0.75] s
空闲车辆 $j, j \in \mathcal{V} \setminus \{0\}$ 计算能力	[3, 8] GHz
RSU 计算能力	15 GHz
发起车辆 u 计算能力	6 GHz
子任务数量 N	8
空闲车辆数量 M	6
发起车辆的传输功率 $P_i, i \in \mathcal{N}$	24 dBm
处理器芯片能耗系数 k	10^{-26}
加性高斯白噪声功率 N_0	-100 dBm
V2V 分配带宽 $B_{ij}, j \in \mathcal{V} \setminus \{0\}$	2 MHz
V2I 分配带宽 $B_{ij}, j \in \{0\}$	3 MHz
RSU 通信范围半径 R	300 m
V2V 通信范围半径 r	150 m
路径损耗因子 α	2

将所提 EJSOC 与以下 4 种参考方案进行对比, 为了综合评估各方案的性能, 定义任务未完成率为在限定的任务时间内未完成的任务的比例, 与系统能耗共同反映系统性能。

- COO (computation offloading optimization) 方案: 该方案优化了计算资源和任务的卸载比例, 而对于 RSU/空闲车辆数量的选择则是随机的^[29]。
- RCO (RSU-assisted computation only)

方案: 该方案只利用 V2I 对 RSU 进行卸载计算, 同时对计算资源和任务卸载比例进行优化^[5-6]。

- EON (entire offloading or not) 方案: 该方案对整个任务本地计算或全部卸载到计算实体, 没有任务分解。
- FCR (fixed computing resource) 方案: 该方案对卸载比例和计算实体选择进行优化, 固定使用的计算资源, 对计算资源不进行优化^[30]。

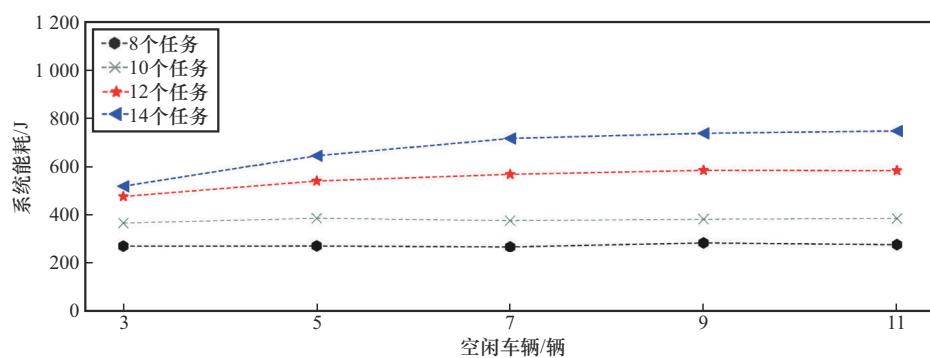
3.1 任务数量和空闲车辆数量对能耗和任务未完成率的影响

针对所提 EJSOC, 评估不同任务数量和空闲车辆数情况下能耗特性及任务未完成率, 如图 2 所示, 当发起车辆生成的任务数量超过 10 个时, 系统整体能耗随着空闲车辆数增加而增加并趋于平稳, 与此同时, 任务未完成率则持续下降。这是因为较少的空闲车辆无法提供充足的计算资源, 尽管能耗较低, 但却以较高的任务未完成率为代价; 随着空闲车辆数增加, EJSOC 在保持能耗相对稳定的前提下, 可以逐步降低任务未完成率, 尤其当任务数量小于 10 个时, 系统能耗趋于平稳。因此, EJSOC 对不同任务负载具有良好适应性。

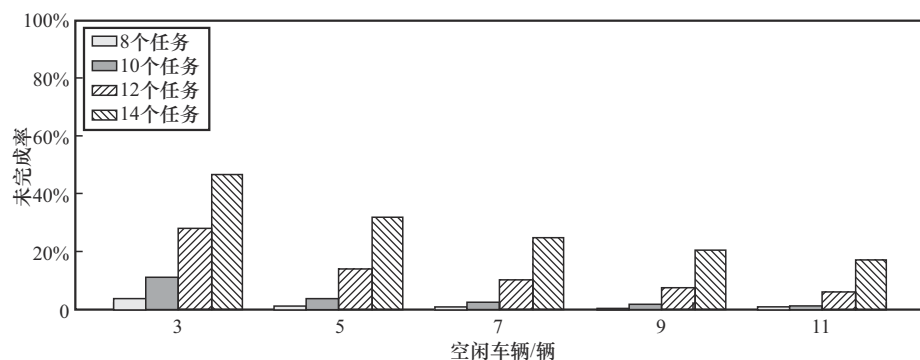
3.2 不同任务数据量对能耗和任务未完成率影响

不同任务数据量情况下能耗和任务未完成率如图 3 所示, 横坐标为当前任务数据量区间与默认区间的比值, 此处将任务数据量默认区间设为 [60, 240] KB。由图 3 可知, 各方案的能耗和任务未完成率均随着任务数据量的增大而增加。这是因为数据量的增大会导致更多的中央处理器 (central processing unit, CPU) 计算能耗, 同时导致任务处理时间变长且传输时延变高, 任务未完成率相应增加。

对于计算密集型和时间敏感型的车联网应

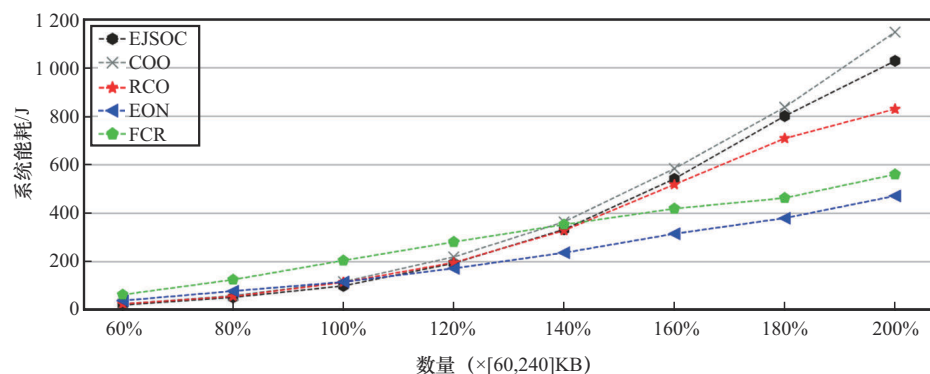


(a) 不同空闲车辆数下的系统能耗

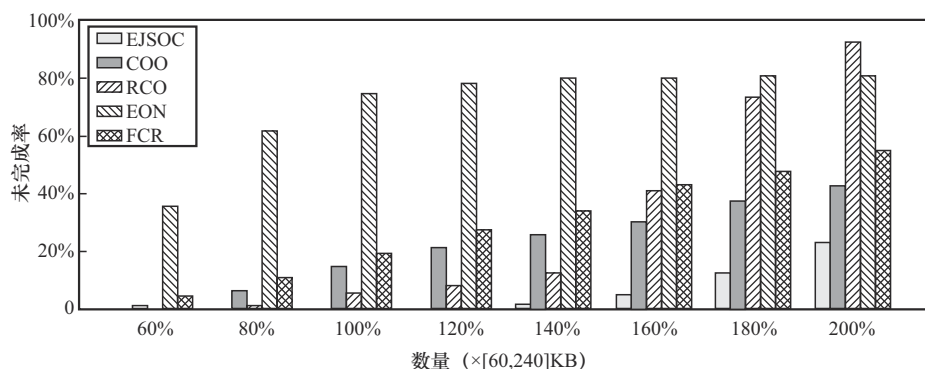


(b) 不同空闲车辆数下的任务完成率

图2 不同任务数量和空闲车数量情况下能耗和任务未完成率



(a) 不同数据量下的能量消耗



(b) 不同数据量下的任务未完成率

图3 不同任务数据量情况下能耗和任务未完成率

用,任务按时完成至关重要,EJSOC对不同的任务数据量均可实现最低的未完成率。相比COO方案,EJSOC凭借优化选择决策机制,在任务未完成率控制和能耗管理上均表现出优越性,其余对比方案尽管在任务数据量较大时能耗普遍低于EJSOC,但其任务未完成率却高出EJSOC数倍。因此,EJSOC通过协同优化选择决策、计算资源配置以及调整任务卸载比例,确保了任务的高效完成,尤其在数据量较小时,EJSOC能够完成所有任务并保持与其他方案相当的能耗水平。

3.3 任务时延限制对能耗和任务未完成率的影响

本文将最大容忍时延小于0.5 s的任务定义为时延敏感型任务。任务时延限制对能耗和任务未完成率的影响如图4所示。由图4可知,各方案的能耗与任务未完成率都随着时延敏感型任务占比的减少而降低,而EJSOC始终实现了最低的未完成率。

当占比为100%时,EJSOC的能耗虽最高,但任务未完成率较其他方案低约10%~25%;当占比降至80%时,EJSOC能耗开始优于EON;当占比为60%和40%时,EJSOC能耗接近FCR;当占比降至20%时,EJSOC的能耗表现与COO及RCO相当且任务未完成率降至近乎为零。这是因为随着时延敏感型任务占比的降低,任务执行的时延要求放宽,从而减少了所需的处理周期数,使得计算能耗得以降低。EJSOC在确保最优任务未完成率的同时,实现了相对较低的系统能耗,体现了最佳的性能平衡。

3.4 空闲车辆速度对能耗和任务未完成率的影响

本文将车速高于23 m/s的空闲车辆定义为高速车辆,并考虑发起车辆即将驶离RSU范围的极端情况。空闲车辆移动速度对能耗和任务未完成率的影响如图5所示。由图5可知,随着高速车辆

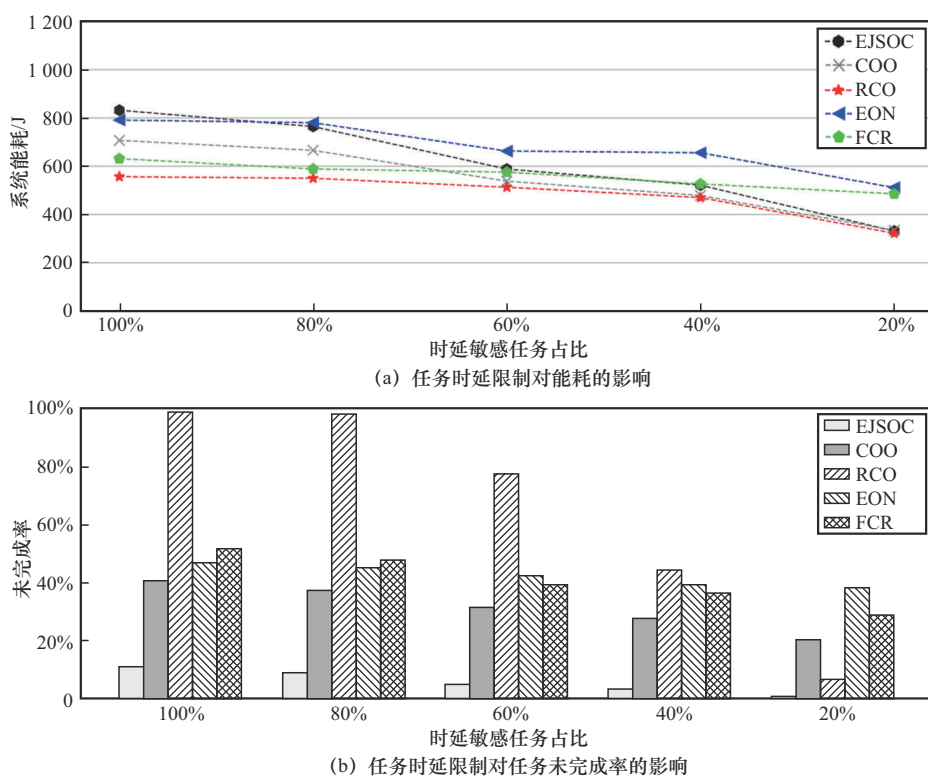
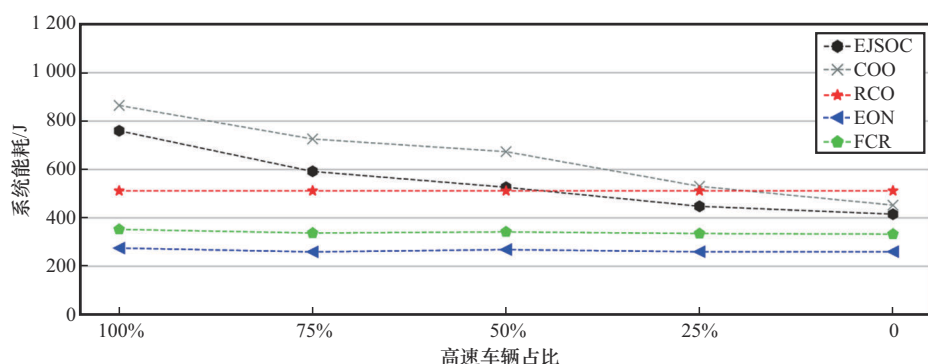
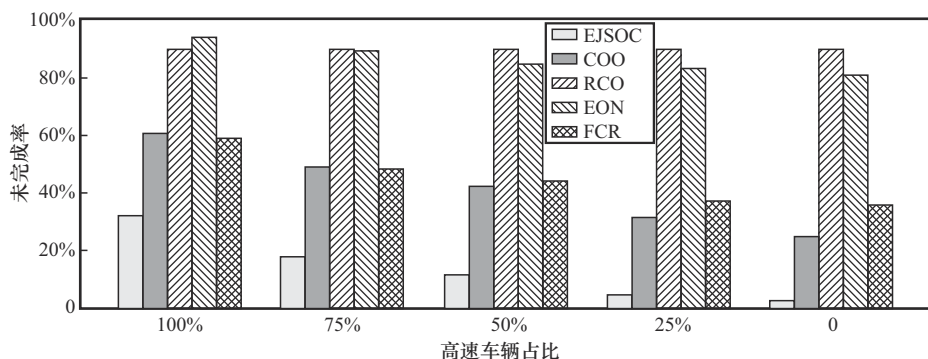


图4 任务时延限制对能耗和任务未完成率的影响



(a) 空闲车辆移动速度对系统能耗的影响



(b) 空闲车辆移动速度对任务未完成率的影响

图5 空闲车辆移动速度对能耗和任务未完成率的影响

占比下降, 由于RCO并不利用空闲车辆, 其能耗和任务未完成率保持不变, 其他方案的能耗和任务未完成率也随之降低, EJSOC始终保持最低未完成率, 且能耗下降明显, 优于COO, 而EON和FCR能耗优化效果不明显。这是因为当高速车辆占比高时, 高速车辆在发起车辆和RSU的通信范围内的等待时间会减少, 任务分配受到限制, 发起车辆承担过多的任务量则导致了任务未完成率的增加。高速车辆占比低时, 任务时延限制变得相对宽松, 需要的计算周期相对减少, 能耗也跟着降低。可以看出, EJSOC通过联合优化选择决策、任务分配和计算资源分配, 达到了最低的任务未完成率和较优能耗, 展现出其综合性能的优势。

3.5 空闲车辆计算资源对能耗和任务未完成率影响

空闲车辆计算资源对能耗和任务未完成率的影响如图6所示, 横坐标的百分比表示空闲车辆

计算资源区间与默认区间的比例。由图6可知, 当比例较小时, 空闲车辆计算资源较小造成大多任务只能卸载至RSU计算, 导致RSU计算资源不足, 随着比例增加, 可以更加优化地卸载, 各方案的任务未完成率随之下降, 此外, 更多的计算资源可以满足更严格的任务时延限制, 但这需要更多的计算周期来实现, 导致各方案的能耗随之增加。所提方案EJSOC相比其他方案始终保持着最低的任务未完成率, 且能耗性能优于部分对比方案。

4 结束语

本文考虑了一个由单个RSU、多个空闲车辆和单个发起车辆组成的车联网系统, 并分析了多个并发异构任务及车辆移动性对系统性能的影响。通过任务车辆的卸载选择决策、任务卸载比例和计算资源的协同优化, 本文旨在使系统能耗最小化, 并针对这一优化问题提出了

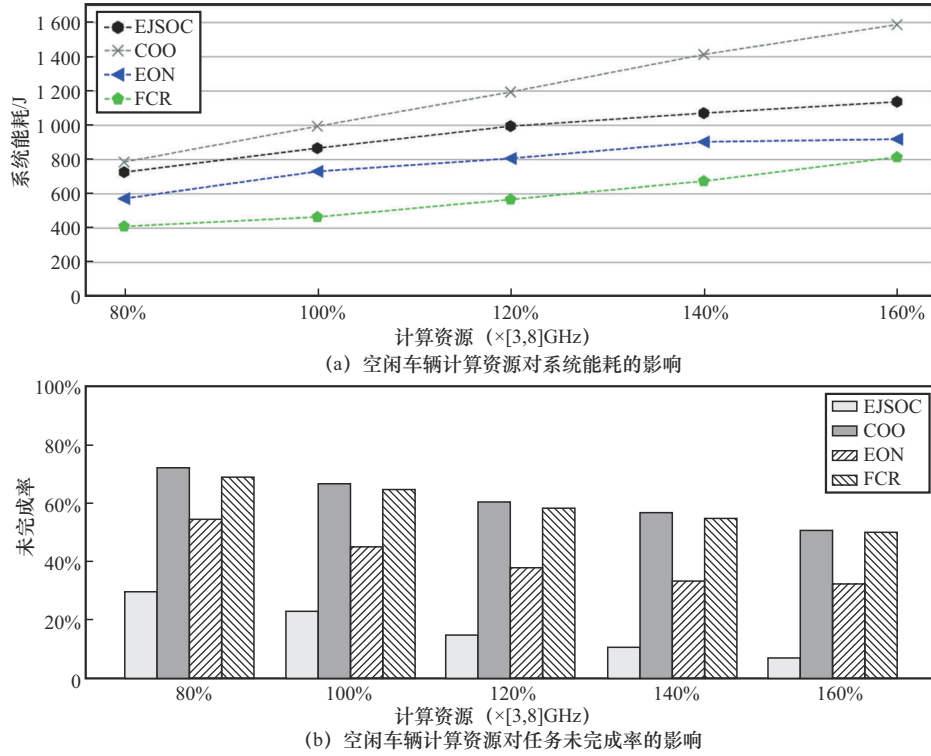


图6 空闲车辆计算资源对能耗和任务未完成率的影响

一种基于ACS算法、拉格朗日乘子和梯度法的优化方案。仿真结果表明,提出的EJSOC可以在保持任务未完成率较低的同时,使系统能耗最小化,综合评估后,其性能优于其他4个基准方案。

附录1

检查问题P1.3的凸性, $E(\mathcal{F})$ 关于 $\mathcal{F}$ 的梯度如下所示:

$$\frac{\partial E(\mathcal{F})}{\partial \mathcal{F}} = [2a_{1j1}kf_{1j1}\beta_{1j1}c_1d_1, \dots, 2a_{iji}kf_{iji}\beta_{iji}c_id_i, \dots, 2a_{NjN}kf_{NjN}\beta_{NjN}c_Nd_N]^T$$

接着求 $E(\mathcal{F})$ 关于 $\mathcal{F}$ 的海塞矩阵:

$$\mathbf{H}(E(\mathcal{F})) = \frac{\partial^2 E(\mathcal{F})}{\partial \mathcal{F}^2} = \text{diag}\left(\frac{\partial E(\mathcal{F})}{\partial \mathcal{F}}\right) = \begin{pmatrix} 2a_{1j1}k\beta_{1j1}c_1d_1 & 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 2a_{iji}k\beta_{iji}c_id_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 2a_{NjN}k\beta_{NjN}c_Nd_N \end{pmatrix}$$

其中, ji 表示编号为 ji 的计算实体来承担发起车辆的任务 h_i 的计算任务。很明显,所求的海塞矩阵 $\mathbf{H}(E(\mathcal{F}))$ 为对角矩阵,且 $a_{iji}k\beta_{iji}c_id_i \geq 0$,其所有顺序主子式大于或等于0。因此, $\mathbf{H}(E(\mathcal{F}))$ 是半正定的, $E(\mathcal{F})$ 是凸函数。

附录2

检查问题P1.4的凸性, $E(\mathcal{F})$ 关于 $\mathcal{F}$ 的梯度如下所示:

$$\frac{\partial E(\mathcal{F})}{\partial \mathcal{F}} = [2a_{1j1}kf_{1u}(1-\beta_{1j1})c_1d_1, \dots, 2a_{iji}kf_{iu}(1-\beta_{iji})c_id_i, \dots, 2a_{NjN}kf_{Nu}(1-\beta_{NjN})c_Nd_N]^T$$

随后求 $E(\mathcal{F})$ 关于 $\mathcal{F}$ 的海塞矩阵:

$$\mathbf{H}(E(\mathcal{F})) = \frac{\partial^2 E(\mathcal{F})}{\partial \mathcal{F}^2} = \text{diag}\left(\frac{\partial E(\mathcal{F})}{\partial \mathcal{F}}\right) = \begin{pmatrix} 2a_{1j1}k(1-\beta_{1j1})c_1d_1 & 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 2a_{iji}k(1-\beta_{iji})c_id_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 2a_{NjN}k(1-\beta_{NjN})c_Nd_N \end{pmatrix}$$



所求得的海塞矩阵 $\mathbf{H}(E(\mathbf{F}))$ 为对角矩阵, 且 $(1 - \beta_{ji}) \geq 0$, $a_{ji}k(1 - \beta_{ji})c_id_i \geq 0$, 其所有顺序主子式大于或等于 0。因此, $\mathbf{H}(E(\mathbf{F}))$ 是半正定的, $E(\mathbf{F})$ 是凸函数。

参考文献:

- [1] ZHUANG W H, YE Q, LYU F, et al. SDN/NFV-empowered future IoV with enhanced communication, computing, and caching[J]. *Proceedings of the IEEE*, 2020, 108(2): 274-291.
- [2] WANG J, FENG D Q, ZHANG S L, et al. Computation offloading for mobile edge computing enabled vehicular networks[J]. *IEEE Access*, 2019(7): 62624-62632.
- [3] CHEN X F, ZHANG H G, WU C, et al. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning[J]. *IEEE Internet of Things Journal*, 2018, 6(3): 4005-4018.
- [4] LIU L, CHEN C, PEI Q Q, et al. Vehicular edge computing and networking: a survey[J]. *Mobile Networks and Applications*, 2021, 26(3): 1145-1168.
- [5] WANG Y P, LANG P, TIAN D X, et al. A game-based computation offloading method in vehicular multiaccess edge computing networks[J]. *IEEE Internet of Things Journal*, 2020, 7(6): 4987-4996.
- [6] DAI Y Y, XU D, MAHARJAN S, et al. Joint load balancing and offloading in vehicular edge computing and networks[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4377-4387.
- [7] SUN J N, GU Q, ZHENG T, et al. Joint optimization of computation offloading and task scheduling in vehicular edge computing networks[J]. *IEEE Access*, 2020(8): 10466-10477.
- [8] ZHAO J H, LI Q P, GONG Y, et al. Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 7944-7956.
- [9] XU X L, SHEN B W, DING S, et al. Service offloading with deep Q-network for digital twinning-empowered Internet of vehicles in edge computing[J]. *IEEE Transactions on Industrial Informatics*, 2022, 18(2): 1414-1423.
- [10] ZHANG K, ZHU Y X, LENG S P, et al. Deep learning empowered task offloading for mobile edge computing in urban informatics[J]. *IEEE Internet of Things Journal*, 2019, 6(5): 7635-7647.
- [11] ZHAN W H, LUO C B, WANG J, et al. Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing[J]. *IEEE Internet of Things Journal*, 2020, 7(6): 5449-5465.
- [12] SHI J M, DU J, WANG J, et al. Distributed V2V computation offloading based on dynamic pricing using deep reinforcement learning[C]//*Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC)*. Piscataway: IEEE Press, 2020: 1-6.
- [13] 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.
- [14] CHEN C, CHEN L L, LIU L, et al. Delay-optimized V2V-based computation offloading in urban vehicular edge computing and networks[J]. *IEEE Access*, 2020(8): 18863-18873.
- [15] HOU X W, REN Z Y, WANG J J, et al. Reliable computation offloading for edge-computing-enabled software-defined IoV[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 7097-7111.
- [16] ZHANG H B, WANG Z X, LIU K J. V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks[J]. *China Communications*, 2020, 17(5): 266-283.
- [17] FAN W H, SU Y, LIU J, et al. Joint task offloading and resource allocation for vehicular edge computing based on V2I and V2V modes[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(4): 4277-4292.
- [18] LIU J X, WANG Y T, ZHANG W, et al. A novel offloading and resource allocation scheme for time-critical tasks in heterogeneous Internet of vehicles[C]//*Proceedings of the 2023 2nd International Conference for Innovation in Technology (INOCON)*. Piscataway: IEEE Press, 2023: 1-7.
- [19] HAZARIKA B, SINGH K, BISWAS S, et al. Multi-agent DRL-based task offloading in multiple RIS-aided IoV networks[J]. *IEEE Transactions on Vehicular Technology*, 2024, 73(1): 1175-1190.
- [20] NGUYEN L D, TUAN H D, DUONG T Q. Energy-efficient signalling in QoS constrained heterogeneous networks[J]. *IEEE Access*, 2016(4): 7958-7966.
- [21] The Institute of Electrical Engineers. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6: wireless access in vehicular environments: IEEE 802.11[S]. Piscataway: IEEE Press, 2010.
- [22] REN C L, ZHANG G A, GU X H, et al. Computing offloading in vehicular edge computing networks: full or partial offloading? [C]//*Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*. Piscataway: IEEE Press, 2022: 693-698.
- [23] ALNAGAR Y, HOSNY S, EL-SHERIF A A. Towards mobility-aware proactive caching for vehicular ad hoc networks[C]//*Proceedings of the 2019 IEEE Wireless Communications and Net-*

working Conference Workshop (WCNCW). Piscataway: IEEE Press, 2019: 1-6.

- [24] YU Z X, HU J, MIN G Y, et al. Mobility-aware proactive edge caching for connected vehicles using federated learning[J]. IEEE Transactions on Intelligent Transportation Systems, 22(8): 5341-5351.
- [25] BI S Z, ZHANG Y J. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6): 4177-4190.
- [26] WANG Y T, SHENG M, WANG X J, et al. Mobile-edge computing: partial computation offloading using dynamic voltage scaling[J]. IEEE Transactions on Communications, 2016, 64(10): 4268-4282.
- [27] 殷人昆, 吴阳, 张晶炜. 蚁群算法解决指派问题的研究和应用[J]. 计算机工程与科学, 2008, 30(4): 43-45, 112.
YIN R K, WU Y, ZHANG J W. Research and application of the ant colony algorithm in the assignment problem[J]. Computer Engineering & Science, 2008, 30(4): 43-45, 112.
- [28] DENG X Y, YU W L, ZHANG L M. A new ant colony optimization with global exploring capability and rapid convergence[C]// Proceedings of the 10th World Congress on Intelligent Control and Automation. Piscataway: IEEE Press, 2012: 579-583.
- [29] YU Y, ZHANG J, LETAIEF K B. Joint subcarrier and CPU time allocation for mobile edge computing[C]//Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2016: 1-6.
- [30] WANG X, NING Z, GUO S, et al. Imitation learning enabled task scheduling for online vehicular edge computing[J]. IEEE Transactions on Mobile Computing, 2020, 21(2): 598-611.

[作者简介]



高文轩 (1998-), 男, 宁波大学信息科学与工程学院硕士生, 主要研究方向为车联网系统的资源管理和性能优化。



杨新杰 (1971-), 男, 宁波大学信息科学与工程学院教授、硕士生导师, 主要研究方向为下一代移动通信系统架构、移动物联网接入技术、协作中继网络性能等。



杨家智 (2001-), 男, 宁波大学信息科学与工程学院硕士生, 主要研究方向为强化学习、车联网、边缘计算与资源分配。



马楠 (1979-), 男, 北京邮电大学信息与通信工程学院教授, 北京无线通信测试技术重点实验室主任, 主要研究方向为无线通信理论、测试技术和大数据应用。