

6-16-2022

Research and Simulation of Internet of Vehicles Task Offloading Based on Mobile Edge Computing

Peng Cheng

School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China; 2042201644@qq.com

Wenzhu Zhang

School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Shuhan Xie

School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Zixuan Yang

School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Research and Simulation of Internet of Vehicles Task Offloading Based on Mobile Edge Computing

Abstract

Abstract: In order to use the computing resources of edge devices to provide high-quality services, a joint resource allocation and task offloading mechanism is designed for the Internet of Vehicles architecture based on mobile edge computing. *In the mechanism, the original problem is decomposed into two sub-problems of resource allocation and offloading decision. The original problem is simplified into the resource allocation of maximizing system capacity, and the initial offloading set is obtained through the proportional resource allocation algorithm; the above problem is solved by the joint offloading decision-making and resource allocation mechanism.* The stable experimental results are obtained by iteration. The simulation results show that, compared with the previous methods, the mechanism reduces the complexity of the algorithm, and the user utility is always high, which greatly improves the service efficiency of the Internet of Vehicles system.

Keywords

mobile edge computing, internet of vehicles, resource allocation, offloading decision

Recommended Citation

Peng Cheng, Wenzhu Zhang, Shuhan Xie, Zixuan Yang. Research and Simulation of Internet of Vehicles Task Offloading Based on Mobile Edge Computing[J]. Journal of System Simulation, 2022, 34(6): 1304-1311.

基于移动边缘计算的车联网任务卸载研究与仿真

程鹏, 张文柱, 谢书翰, 杨子轩

(西安建筑科技大学 信息与控制工程学院, 陕西 西安 710055)

摘要: 为能够在接近用户端利用边缘设备的计算资源为人们提供高质量的服务, 基于移动边缘计算的车联网架构设计了一种联合资源分配和任务卸载的机制。将原问题分解为资源分配和卸载决策2个子问题分别求解。将原问题简化成系统容量最大化的资源分配问题, 通过比例资源分配算法得到初始卸载集合; 采用联合卸载决策与资源分配机制求解上述问题。通过迭代直到实验现象趋于平稳。仿真结果表明: 与以往方法相比该机制降低了算法的复杂度, 并且用户效用始终较高, 大大提升了车联网系统的服务效率。

关键词: 移动边缘计算; 车联网; 资源分配; 卸载决策

中图分类号: TP391; TN915.5

文献标志码: A

文章编号: 1004-731X(2022)06-1304-08

DOI: 10.16182/j.issn1004731x.joss.21-0015

Research and Simulation of Internet of Vehicles Task Offloading Based on Mobile Edge Computing

Cheng Peng, Zhang Wenzhu, Xie Shuhan, Yang Zixuan

(School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: In order to use the computing resources of edge devices to provide high-quality services, a joint resource allocation and task offloading mechanism is designed for the Internet of Vehicles architecture based on mobile edge computing. *In the mechanism, the original problem is decomposed into two sub-problems of resource allocation and offloading decision. The original problem is simplified into the resource allocation of maximizing system capacity, and the initial offloading set is obtained through the proportional resource allocation algorithm; the above problem is solved by the joint offloading decision-making and resource allocation mechanism.* The stable experimental results are obtained by iteration. The simulation results show that, compared with the previous methods, the mechanism reduces the complexity of the algorithm, and the user utility is always high, which greatly improves the service efficiency of the Internet of Vehicles system.

Keywords: mobile edge computing; internet of vehicles; resource allocation; offloading decision

引言

移动边缘计算(mobile edge computing, MEC)是一种结合5G架构的技术。MEC可以被看作是一种微型云服务平台, 通过接近用户端来完成计算、存储和通信等功能^[1-4]。因此一些对带宽和时

延有更加苛刻需求的计算任务可以在边缘服务器上处理。随着MEC技术的逐渐成熟, 车辆边缘计算(vehicle edge computing, VEC)成为研究热点, 车载应用对车联网技术提出了新的挑战。例如, 增强现实、自动驾驶等新型技术。VEC能够解决车辆网络中计算能力不足的缺陷^[5-6]。在城市交通

收稿日期: 2021-01-07

修回日期: 2021-03-11

基金项目: 国家自然科学基金(61473216); 陕西省重点研发计划(2021GY-066); 陕西省自然科学基金基础研究计划(2020JM-489)

第一作者: 程鹏(1994-), 男, 硕士生, 研究方向为边缘计算、车联网。E-mail: 2042201644@qq.com

拥堵的区域和停车场, 有大量闲置的车辆, 这些车辆有大量的计算资源未被利用。将车辆作为边缘节点来提供计算服务, 用户将任务卸载到网络的边缘去计算成为一种可行的方案^[7-9]。路侧单元 (road side unit, RSU)、微基站都可以作为服务节点去使用, 其中用户和服务节点之间通过专用短程通信技术进行通信。

面向边缘计算的车联网研究已经有了很多成果。文献[10]指出在一个多用户MEC场景中每个移动终端都有要执行的任务, 通过研究2个移动终端之间的相互依赖关系, 建立了任务卸载模型并且使移动终端时延与能耗的加权和最小。文献[11]充分利用了停放车辆中的计算资源, 应用博弈论得到了最优卸载决策集合。文献[12]利用停放车辆的闲置资源将VEC和容器技术相结合, 保证了停放车辆在执行任务时能够快速响应, 大大提高了车联网的资源利用率。文献[13]采用自主VEC计算模型, 通过收集和分析附近移动终端应用程序的数据, 采用启发式算法得出最优卸载策略。文献[14]基于李亚普诺夫的计算卸载策略, 根据当前的参数动态调整任务卸载策略, 最后决定任务在本地执行还是在MEC服务器上执行。文献[15]采用一种多目标帕累托优化算法解决了车联网能耗过大以及车辆高动态性所引起的任务间冲突的问题。文献[16]以最小化约束条件采用动态定价策略通过不断调整价格来提高MEC系统的服务质量。文献[17]提出了一种车辆雾计算的架构, 优化了单个移动终端的资源分配问题, 提高了车联网的服务效率。然而上述文献均没有充分考虑到任务卸载过程中由于资源分配不均导致资源浪费的问题。

本文提出了一种在VEC环境下的联合卸载决策与资源分配机制 (joint offloading decision and resource allocation, JODRA)。以往的方法都是离线优化的迭代非线性方法^[18], 本文提出了一种低复杂度的非迭代比例资源分配算法, 该方法不需要考虑子信道有较高信噪比的约束, 显著降低了复杂度。最后采用改进的粒子群算法 (improved

particle swarm optimization, IPSO) 得出了最优的卸载决策, 仿真结果验证了该机制的优越性。

1 系统仿真和优化问题的制定

1.1 系统仿真

图1是本文所提出的VEC系统场景。在该系统中, 道路一侧部署RSU, 并且相邻RSU的覆盖区域不重叠。因此可将道路划分为若干相同的区域。每一个RSU都配备了MEC服务器, 此时RSU充当接近用户端的微型云。将任务请求车辆用集合 $V = \{v_1, v_2, \dots, v_N\}$ 来表示, $v_i (i \in [1, N])$ 可以将任务卸载到附近的RSU上处理, 最后将计算结果返回给 v_i 。车辆 v_i 也可以将任务卸载至附近的闲置车辆进行处理, 由于闲置车辆和RSU均可提供计算服务, 因此将它们统一定义为服务节点, 表示为 $G = \{g_1, g_2, \dots, g_N\}$, 这样能够节约时间, 减少不必要的能量损耗。每个RSU的无线信道和计算资源都用于相应的区域。本文主要研究的是具有代表性的区域, 因此将本文的体系结构应用到其他场景也是非常容易的。 v_i 的计算任务可以被描述为 $OT_i = \{o_i, d_i, t_i^{\max}\}$, 其中, o_i 为输入数据的量; d_i 为所需的CPU周期数; t_i^{\max} 为可容忍的时延阈值。所有要卸载的任务是原子性的, 不能分离。在计算任务卸载过程中网络拓扑结构保持不变^[19]。

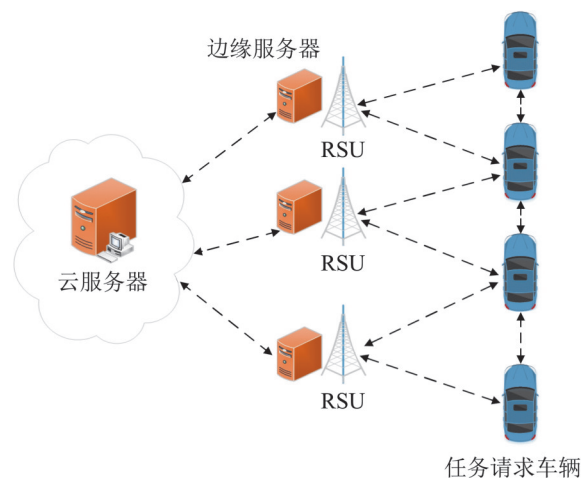


图1 VEC仿真场景

Fig. 1 VEC simulation scenario

1.2 通信模型

本文采用正交频分多址技术在用户和基站之间进行通信。用户数据区域的每个车辆 v_i 均可将其计算任务 OT_i 卸载到邻近的服务节点进行处理。且输入数据需要通过无线信道传输到服务节点，请求车辆的上行传输速率可以定义为

$$\gamma = \frac{B}{S} \ln \left(1 + \frac{p_i \eta^2}{\zeta \sigma^2} \right) \quad (1)$$

式中： B 为系统带宽； p_i 为分配给车辆 v_i 的传输功率； $\sigma^2 = (\vartheta_0 B)/S$ 为白色高斯噪声，这里 ϑ_0 是噪声功率谱密度， S 个子载波用 $s(1 \leq s \leq S)$ 来表示； η^2 为 v_i 到服务节点的信道增益。因此，任务请求车辆 v_i 的信噪比可以表示为 $SNR_{i,s} = \eta^2 / (\zeta \sigma^2)$ ，其中 ζ 是恒定的信噪比差距。

为有效地复用频谱，车辆与基础设施通信模式和车辆与车辆通信模式在同一频段工作。该系统还为RSU分配了正交频谱，从而可以忽略RSU间的干扰。

1.3 计算模型

在本地计算会涉及处理时延和计算能耗；在进行卸载计算时会涉及处理时延、传输时延和传输能耗。

1.3.1 本地计算

车辆 v_i 的计算能力为 c_i ，不同车辆所拥有的计算资源是不同的，则本地执行时延为

$$t_i^{\text{local}} = d_i / c_i \quad (2)$$

根据文献[20]，本地计算能耗为

$$e_i^{\text{local}} = k d_i c_i^2 \quad (3)$$

式中： k 为能耗参数^[21-22]，设置为 $k = 10^{-28}$ 。

1.3.2 卸载计算

当车辆 v_i 将其任务卸载至服务节点 g_j ， $j \in [1, M]$ 进行处理时的传输时延为

$$t_i^{\text{trans}} = o_i / \gamma_{i,j} \quad (4)$$

输出这些计算密集型任务的数据相对输入数

据非常小，因此本文不考虑下行传输时延。

传输能耗为

$$e_i^{\text{trans}} = p_i t_i^{\text{trans}} \quad (5)$$

卸载计算的能耗只考虑与传输有关的能耗。卸载完成后，服务节点 g_j 将对任务进行处理，处理过程所消耗的时间为

$$t_i^{\text{cal}} = d_j / c_j^i \quad (6)$$

式中： c_j^i 为服务节点 g_j 分配给任务请求车辆 v_i 的计算资源。

1.4 最优化问题制定

$\chi_i \in \{0, 1\}$ 代表卸载决策，当 $\chi_i = 1$ 时计算任务 OT_i 将被卸载，否则 OT_i 将在本地执行。因此，执行 OT_i 所产生的时延和能耗为

$$t_i = \chi_i (t_i^{\text{trans}} + t_i^{\text{cal}}) + (1 - \chi_i) t_i^{\text{local}} \quad (7)$$

$$e_i = \chi_i e_i^{\text{trans}} + (1 - \chi_i) e_i^{\text{local}} \quad (8)$$

定义用户效用作为 v_i 所产生的时延和能耗的权衡：

$$u_i(\chi_i, p_i) = \alpha \frac{t_i^{\text{local}} - t_i}{t_i^{\text{local}}} + \beta \frac{e_i^{\text{local}} - e_i}{e_i^{\text{local}}} \quad (9)$$

式中： α 与 β 分别为时延和能耗的权重因子，且满足 $\alpha + \beta = 1, 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$ 。如果任务很紧急，用户可以适当提高时延权重。否则，车辆在低电量情况下，可以适当提高能耗权重因子。由此，可将式(9)定义为一个最优化问题：

$$\begin{aligned} P_1: & \max_{A, P} \sum_{i=1}^N u_i(\chi_i, p_i) \\ \text{s.t.} & \\ C_1: & \chi_i \in \{0, 1\}, \forall i \in N \\ C_2: & 0 < p_i < p_{\max}, \forall i \in N \\ C_3: & \sum_{i=1}^N p_i = p_{\text{total}}, \forall i \in N \\ C_4: & (1 - \chi_i) t_i^{\text{local}} + \chi_i (t_i^{\text{trans}} + t_i^{\text{cal}}) \leq t_{\max}, \forall i \in N \\ C_5: & \psi_{i,s} \in \{0, 1\}, \forall i \in N, s \in S \end{aligned} \quad (10)$$

式中： $A = \{\chi_1, \chi_2, \dots, \chi_N\}$ 为卸载决策集合； $P = \{p_1, p_2, \dots, p_N\}$ 为车辆传输功率分配集合。约束条件： C_1 为车辆可以将其计算任务卸载给服务节点进行计算或在本地执行任务； C_2 为车辆传输功率约束； C_3 为总功率约束； C_4 为完成任务所需的时间不能超

过任务所能容忍的最大时延; C_5 为子载波约束条件, 其中 $\psi_{i,s} \in \{0, 1\}$ 是子载波分配指标系数, 当且仅当 $\psi_{i,s} = 1$ 时表示将子载波 s 分配且仅分配给用户。

2 解决方案

将所提出的问题分解成2个子问题: 资源分配和卸载决策。根据式(9)可知当任务在本地执行时不提供任何用户效用, 即 $\chi_i = 0$ 。因此可得

$$P_2: \max_{A, P} \sum_{i=1}^N u_i(1, p_i) \quad (11)$$

$$\text{s.t.} \\ C_1, C_2, C_3, C_4, C_5$$

2.1 资源分配

根据初始的卸载决策集集合 A , 将 P_2 简化成系统容量最大化的资源分配问题:

$$SP_1: \text{SysCap}(A) = \max_p \frac{B}{S} \sum_{i=1}^N \sum_{s=1}^S \psi_{i,s} \text{lb} \left(1 + \frac{p_i \eta^2}{\zeta \sigma^2} \right)$$

$$\text{s.t.} \\ C_2, C_3, C_5 \quad (12)$$

由式(12)可知本文所提出的是NP-hard组合优化问题, 很难用多项式时间算法来解决。因此需要对原始问题进行一些简化处理。每个 v_i 分配的功率可以用拉格朗日乘子法求解

$$\frac{1}{\Omega_1} \frac{S_1}{S} \left(\text{lb} \left(1 + \text{SNR}_{1,1} \frac{p_1 - R_1}{S_1} \right) + \text{lb} W_1 \right) = \\ \frac{1}{\Omega_k} \frac{S_i}{S} \left(\text{lb} \left(1 + \text{SNR}_{i,1} \frac{p_i - R_i}{S_i} \right) + \text{lb} W_i \right) \quad (13)$$

其中:

$$R_i = \sum_{s=2}^{S_i} \frac{\text{SNR}_{i,s} - \text{SNR}_{i,1}}{\text{SNR}_{i,s} \text{SNR}_{i,1}} \quad (14)$$

$$W_i = \left(\prod_{s=2}^{S_i} \frac{\text{SNR}_{i,s}}{\text{SNR}_{i,1}} \right)^{\frac{1}{S_i}} \quad (15)$$

式中: S_i 为分配给请求车辆 v_i 的子载波数; $\text{SNR}_{i,s}$ 为第 i 个用户在子载波 s 上接收到的信噪比; $\Omega_1: \Omega_2: \dots: \Omega_N$ 为归一化的比例常数并且 $\sum_{i=1}^N \Omega_i = 1$ 。由

参考文献[23]可知:

$$S_1: S_2: \dots: S_N \approx \Omega_1: \Omega_2: \dots: \Omega_N \quad (16)$$

因此, 式(13)可以用 S_i 代替 Ω_i , 联立方程组,

写成矩阵形式为

$$\begin{bmatrix} a_{N,N} & 0 & \dots & 1 \\ 0 & a_{N-1,N-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} p_N \\ p_{N-1} \\ \vdots \\ p_1 \end{bmatrix} = \begin{bmatrix} y_N \\ y_{N-1} \\ \vdots \\ p_{\max} \end{bmatrix} \quad (17)$$

然后对稀疏矩阵进行三角分解得到:

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \\ \frac{1}{a_{N,N}} & \frac{1}{a_{N-1,N-1}} & \dots & 1 \end{bmatrix} \quad (18)$$

$$U = \begin{bmatrix} a_{N,N} & 0 & \dots & 1 \\ 0 & a_{N-1,N-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 - \sum_{i=2}^N \frac{1}{a_{i,i}} \end{bmatrix} \quad (19)$$

其中:

$$a_{i,i} = -\frac{S_1}{S_i} \frac{\text{SNR}_{i,1} W_i}{\text{SNR}_{1,1} W_1} \quad (20)$$

$$y_i = \frac{S_1}{\text{SNR}_{1,1}} \left(W_i - W_1 + \frac{\text{SNR}_{1,1} R_i W_1}{S_1} - \frac{\text{SNR}_{i,1} R_i W_i}{S_i} \right) \quad (21)$$

由此可得:

$$p_1 = (p_{\text{total}} - \sum_{i=2}^N \frac{y_i}{a_{ii}}) / \left(1 - \sum_{i=2}^N \frac{1}{a_{ii}} \right) \quad (22)$$

$$p_i = (y_i - p_1) / a_{ii} \quad (23)$$

算法1: 比例资源分配算法

输入: 初始卸载决策集合 A 。

输出: 功率分配集合 P_{best} 。

while $i \leq N$ do

将式(23)带入式(11)计算初始功率分配集合 P , 形成最大系统效用集合 u_i , 将 u_i 所对应的功率 p_i 从 P 中删除

将 P 中剩余元素重新分配给任务车辆, 找到 P 中功率最大值, 并将其从 P 中移动到 u_i

依次为任务车辆分配最优功率, 计算最终每辆车所产生的系统效用

end while

return P_{best}

2.2 卸载决策

初始卸载集合情况下求得资源分配方案，将其数值解代入式(11)简化后得出最优化问题：

$$\begin{aligned} SP_2: \max_A Z(A) \\ \text{s.t.} \\ C_1, C_4 \end{aligned} \quad (24)$$

由于 $Z(A)$ 是非线性问题， SP_2 仍然是一个NP-hard问题。因此，本文采用IPSO求解上述问题。通过迭代直到实验现象趋于平稳或者达到迭代预设次数：

$$\mathbf{u}_i^{k+1} = \omega \mathbf{u}_i^k + \tau_1 \text{rand}_1(p_{\text{best}_i} - \mathbf{x}_i^k) + \tau_2 \text{rand}_2(\mathbf{g}_{\text{best}} - \mathbf{x}_i^k) \quad (25)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{u}_i^k \quad (26)$$

$$\mathbf{g}_i^k = \text{mod } l_i^k + 1 \quad (27)$$

式中： ω 为惯性权重； τ_1 和 τ_2 为学习因子； rand_1 和 rand_2 [0, 1]区间上服从均匀分布的独立随机变量； p_{best_i} 为粒子 i 的最佳位置； \mathbf{g}_{best} 为群体中所有粒子的最佳位置； \mathbf{u}_i^k 为第 k 次迭代粒子 i 的速度矢量； \mathbf{x}_i^k 为第 k 次迭代粒子 i 的位置矢量。由于IPSO的位置 \mathbf{x}_i 是连续的， \mathbf{v}_i 中任务分配是离散的集合，因此本文运用最小位置规则(small position value, SPV)把连续的位置矢量 $\mathbf{x}_i^k = [x_1^k, x_2^k, \dots, x_N^k]$ 转化成离散的矢量 $l_i^k = [l_1^k, l_2^k, \dots, l_N^k]$ ，然后将矢量 l_i^k 的每个元素映射到服务节点矢量 $\mathbf{G} = [g_1^k, g_2^k, \dots, g_M^k]$ ，具体通过式(27)来实现。

2.3 改进的粒子群算法

IPSO在迭代前期的效率较高，迭代后期时效率会逐渐变差，并且将会陷入局部最优和早熟现象。因此，文本采用粒距解决初始空间粒子分散不均匀的问题：

$$D = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{m=1}^M (p_{\text{best}_m} - \overline{p_{\text{best}_m}})^2} \quad (28)$$

式中： D 为粒距，表示粒子之间的分散程度， D 越大粒子间越分散； N 为种群规模； M 为空间维数； p_{best_m} 为粒子 i 在 m 维的最优值， $\overline{p_{\text{best}_m}}$ 为所有粒子在 m 维最优值的平均值。通过 D 来决定种群是

否变异，变异操作加强了局部搜索的能力，本文设定阈值 $h = 10$ 与 D 进行比较，若 $D < h$ ，则基本位变异，变异概率为0.3，从而避免早熟现象的发生。

算法2：联合卸载决策与资源分配机制 (JODRA)

输入：算法迭代次数 $Iteration$ ，计算任务 V ，服务节点 G ，粒子个数 N

输出：最优卸载决策集合 A_{best}

start

生成个体的初始种群

for $k=1$ to $Iteration$ do

for $i=1$ to N

更新粒子的速度和位置

end for

求解式(12)基于当前卸载决策集合得出每个请求用户的传输功率

变异条件判断，根据粒距和阈值 $h = 10$ 判断是否进行变异操作

对所有个体进行评估，选择产生下一代

do

return A_{best}

end procedure

3 仿真分析

本文使用Matlab平台对JODRA机制进行仿真实验，通过实验仿真结果来推断JODRA机制的性能。本节在IEEE 802.11p标准背景下展开。路边配备的RSU其覆盖范围是100 m，系统总功率为1 W，系统带宽为1 MHz，信噪比差距 $\zeta = 3.3$ ，详细仿真参数如表1所示。

3.1 平均CPU执行时间

本文将JODRA与遗传算法(genetic algorithm, GA)进行比较。在GA中，用户以分布式方式进行卸载决策，当有可能获得比本地计算更高的用户效用时，用户将进行卸载计算^[24]。

表1 仿真参数
Table 1 Simulation parameters

参数	描述	取值
o_i /kB	任务计算大小	400~500
d_i	任务所需CPU周期数	1 200~2 000
α, β	权重因子设置	0.5
c_i /GHz	车辆计算能力	1~2
c_j /GHz	边缘节点计算能力	5~6
p_{\max} /dBm	车辆最大发射功率	30

由图2可知,随着用户数目的增加,2种算法之间的性能差异逐渐明显。GA的执行时间比JODRA慢了一个数量级,并且JODRA为本实验提供了更高的实现能力。JODRA机制将原问题转化为2个子问题,相互迭代求解。由式(22)~(23)可知该过程需要一次除法和 $2(N-1)$ 次乘法和 $3(N-1)$ 次减法,因此复杂度是 $O(N)$ 。IPSO算法的时间复杂度为 $O(N^2)$,JODRA机制总体的时间复杂度为 $O(N^2 + N)$ 。而GA首先经过 $O(|N|\log^2|N|)$ 迭代后得到最优拉格朗日乘数,最后通过逆向归纳的复杂度为 $O(N)$ 。因此,GA整体的时间复杂度为 $O(N^2 \log^2 N)$ 这比本文所提出的算法复杂的多。

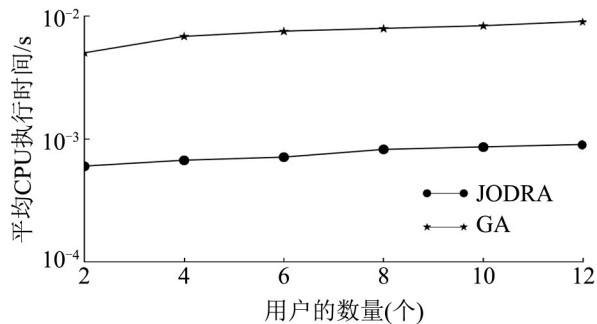


图2 用户的数量与平均CPU执行时间关系

Fig. 2 Relationship between the number of users and CPU execution time

3.2 平均用户效用

本节将JODRA与基本的粒子群算法(PSO)、全部卸载计算(all offload computing, AOC)进行比较。如图3所示,平均用户效用随着传输功率的增加而增加。受多个用户分集增益的影响,当用户数量较多时此增益更为明显。JODRA赋予了每个任务请求

车辆最大程度的数据传输速率。在子载波数为定值时,通过比较在不同传输功率限制下平均用户效用的变化趋势,可以看出JODRA所得到的用户效用始终较高。PSO由于自身算法的缺陷,局部搜索能力较差迭代搜索能力逐渐降低,因次,无法得出最优的卸载策略。

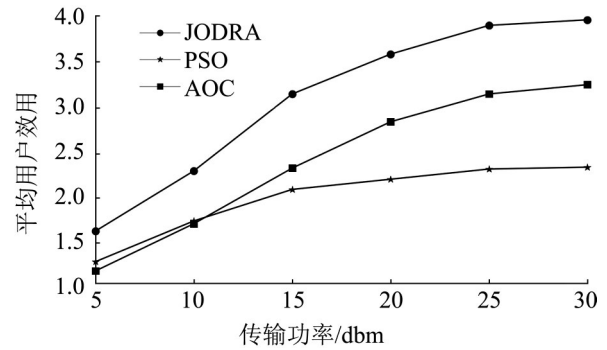


图3 传输功率与平均用户效用关系

Fig. 3 Relationship between transmission power and average users utility

图4展示了在不同用户数量的情况下每种算法的平均用户效用。当用户数量小于30时,JODRA与PSO所得到的平均用户效用非常相近。这是因为当用户数量较少时,有充足的服务节点去处理数据。随着用户数量的增多,PSO体现出了明显的劣势。AOC没有充分考虑时间约束问题,某些程序在处理过程中超过了预期的时间,导致卸载失败。JODRA机制改善了等功率分配系统效率低的缺点,并通过IPSO提高用户效用,随着用户数量的增加JODRA的优势越发突出。

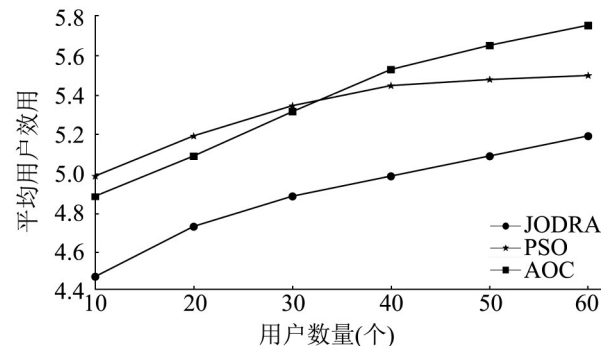


图4 用户数量与平均用户效用关系

Fig. 4 Relationship between number of users and average users utility

4 结论

本文设计了一种基于移动边缘计算的车联网系统架构, 构建了联合卸载决策和资源分配机制。在此基础上, 本文提出了一种低复杂度的比例资源分配算法, 并结合粒子群算法对卸载决策进行优化。实验表明该方案收敛速度快, 大大降低了计算复杂度。本文解决了车联网计算能力不足的缺陷, 在达到系统容量最大化的同时以最优卸载策略使车辆中的计算资源得到充分利用。通过多组实验对比表明该机制有效地降低了系统时延和能耗, 并且具有可行性和高效性。边缘计算过程中由于环境的不确定性, 未来边缘设备间的通信干扰是需要攻克的热点问题。

参考文献:

- [1] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69-89.
Shi Weisong, Zhang Xingzhou, Wang Yifan, et al. Edge Computing: Current Situation and Outlook[J]. Journal of Computer Research Development, 2019, 56(1): 69-89.
- [2] 赵梓铭, 刘芳, 蔡志平, 等. 边缘计算: 平台、应用与挑战[J]. 计算机研究与发展, 2018, 55(2): 327-337.
Zhao Ziming, Liu Fang, Cai Zhiping, et al. Edge Computing: Platforms, Applications and Challenges[J]. Journal of Computer Research Development, 2018, 55(2): 327-337.
- [3] Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. IEEE Communication Survey & Tutorials(S1553-877X), 2017, 19(3): 1628-1656.
- [4] Hu Y C, Patel M, Sabella D, et al. Mobile Edge Computing-A Key Technology Towards 5G[J]. ETSI White Papers, 2015, 11(11): 1-16.
- [5] Huang C M, Chen Y F, Xu S, et al. The Vehicular Social Network (VSN)-Based Sharing of Downloaded Geo Data Using the Credit-Based Clustering Scheme[J]. IEEE Access(S2169-3536), 2018, 54(6): 58254-58271.
- [6] Zhou H, Wang H, Chen X, et al. Data Offloading Techniques Through Vehicular ad Hoc Network: A Survey [J]. IEEE Access(S2169-3536), 2018, 6(9): 65250-65259.
- [7] Chen X. Decentralized Computation Offloading Game for Mobile Cloud Computing[J]. IEEE Transactions on Parallel and Distributed Systems(S1045-9219), 2015, 26(4): 97-4983.
- [8] You C S, Huang K B, Chae K J, et al. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading[J]. IEEE Trans on Wireless Commnca-Tions (S1558-2248), 2017, 16(3): 1397-1411.
- [9] Malandrino F, Casetti C, Chiasserini C F, et al. The Role of Parked Cars in Content Downloading for Vehicular Networks[J]. Vehicular Technology IEEE Transactions on (S1939-9359), 2014, 63(9): 4606-4617.
- [10] Yan J, Bi S Z, Ying J, et al. Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing With Inter-User Task Dependency[J]. IEEE Transactions on Wireless Communications(S1558-2248), 2020, 19(1): 235-250.
- [11] Huang X, Yu R, Liu J, et al. Parked Vehicle Edge Computing: Exploiting Opportunistic Resources for Distributed Mobile Applications[J]. IEEE Access (S2169-3536), 2018, 6(5): 66649-66663.
- [12] Huang X, Li P, Yu R. Social Welfare Maximization in Container-Based Task Scheduling for Parked Vehicle Edge Computing[J]. IEEE Communications Letters (S1558-2558), 2019, 23(8): 1347-1351.
- [13] Feng J, Liu Z, Wu C, et al. AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling[J]. IEEE Transactions on Vehicular Technology(S1939-9359), 2017, 66(12): 10660-10675.
- [14] Hu Y, Cui T, Huang X, et al. Task Offloading Based on Lyapunov Optimization for MEC-assisted Platooning[C]// IEEE 2019 11th Inter-National Conference on Wireless Communications and Signal Processing (WCSP). Piscataway: IEEE, 2019: 1-5.
- [15] Zheng X, Chen Y F, Alam M, et al. Multi-Task Scheduling Based on Classification in Mobile Edge Computing[J]. Electronics, 2019, 8(9): 938-954.
- [16] Han D, Chen W, Fang Y. A Dynamic Pricing Strategy for Vehicle Assisted Mobile Edge Computing Systems[J]. IEEE Wireless Communications Letters(S2162-2345), 2019, 8(2): 420-423.
- [17] Hou X, Li Y, Chen M, et al. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures[J]. IEEE Transactions on Vehicular Technology(S1939-9359), 2016, 65(6): 3860-3873.
- [18] Shen Z, Andrews J G, Evans B L. Optimal Power Allocation in Multiuser OFDM Systems[J]. IEEE Global Communications Conference(S1558-2558), 2003, 80(9): 337-341.
- [19] Wang C, Yu F R, Liang C, et al. Joint Computation Offloading and Interference Management in Wireless Cellular Networks with Mobile Edge Computing[J]. IEEE

- Transactions on Vehicular Technology(S1939-9359), 2017, 66(8): 7432-7445.
- [20] Guo S T, Xiao B, Yang Y, et al. Energy-Efficient Dynamic Offloading and Resource Scheduling in Mobile Cloud Computing[J]. IEEE Transactions on Mobile Computing(S1558-0660), 2019, 18(2): 319-333.
- [21] Pham Q, Leanh T, Tran N H, et al. Decentralized Computation Offloading and Resource Allocation for Mobile-Edge Computing: A Matching Game Approach[J]. IEEE Access(S2169-3536), 2018, 6(11): 75868-75885.
- [22] Wen Y, Zhang W, Luo H. Energy-Optimal Mobile Application Execution: Taming Resource-Poor Mobile Devices with Cloud Clones[C]//2012 Proceedings IEEE Annual Joint Conference: INFOCOM. IEEE Computer and Communications Societies, 2012: 2716-2720.
- [23] Wong I C, Shen Z, Evans B L, et al. A Low Complexity Algorithm for Proportional Resource Allocation in OFDMA Systems[C]//2004 IEEE Workshop on Signal Processing Systems Design and Implementation. Piscataway: IEEE, 2004: 1-6.
- [24] Zhang W, Wen Y, Wu D O. Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel[J]. IEEE Transactions on Wireless Communications(S1558-2248), 2015, 14(1): 81-93.