

面向车辆边缘计算任务卸载的延迟与能耗联合优化方法

李文旺, 周浩浩, 邓苏, 马武彬, 吴亚辉

引用本文

李文旺, 周浩浩, 邓苏, 马武彬, 吴亚辉. [面向车辆边缘计算任务卸载的延迟与能耗联合优化方法](#)[J]. 计算机科学, 2024, 51(11A): 231000080-7.

LI Wenwang, ZHOU Haohao, DENG Su, MA Wubin, WU Yahui. [Joint Optimization of Delay and Energy Consumption of Tasks Offloading for Vehicular Edge Computing](#) [J]. Computer Science, 2024, 51(11A): 231000080-7.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于端边协同的节点部署和资源分配联合优化方法](#)

Joint Optimization Method for Node Deployment and Resource Allocation Based on End-Edge Collaboration

计算机科学, 2024, 51(11A): 240200010-7. <https://doi.org/10.11896/jsjcx.240200010>

[面向回收信息的线上线下载多源异构数据融合系统](#)

Online and Offline Multi-source Heterogeneous Data Fusion System for Recycling Information

计算机科学, 2024, 51(11A): 240100095-7. <https://doi.org/10.11896/jsjcx.240100095>

[目标个数不规则变化的动态多目标优化算法](#)

Dynamic Multi-Objective Optimization Algorithm with Irregularly Varying Number of Objectives

计算机科学, 2024, 51(11A): 231000079-11. <https://doi.org/10.11896/jsjcx.231000079>

[基于遗传算法的低碳导向的物流中心配送优化](#)

Optimization of Low-carbon Oriented Logistics Center Distribution Based on Genetic Algorithm

计算机科学, 2024, 51(11A): 231200035-6. <https://doi.org/10.11896/jsjcx.231200035>

[基于演化多目标聚类的SAR图像变化检测](#)

Change Detection in SAR Images Based on Evolutionary Multi-objective Clustering

计算机科学, 2024, 51(9): 140-146. <https://doi.org/10.11896/jsjcx.230800014>

面向车辆边缘计算任务卸载的延迟与能耗联合优化方法

李文旺 周浩浩 邓 苏 马武彬 吴亚辉

国防科技大学信息系统工程全国重点实验室 长沙 410073

(liwenwang22@nudt.edu.cn)

摘要 车联网(IoV)与联网自动驾驶汽车(CAV)的结合推动了自动驾驶技术的飞速发展,但也带来了计算资源的巨大需求,给资源受限的车辆带来了挑战。车辆边缘计算(VEC)的出现,提供了一种全新的解决方案,通过将任务卸载到路侧单元中的边缘服务器上,能够以更高效的方式为车联网提供服务。然而,多个车辆同时发出卸载请求时会产生资源抢占,增大任务处理延时,如何高效调度资源以最大化服务质量是一个亟待解决的问题。为此,文中旨在从多目标优化的角度,详细分析VEC计算卸载的延迟和能耗,使延迟和成本最小化,并提出了名为NSGA2TO的基于改进非支配排序遗传算法的任务卸载算法。该算法能够寻找到多目标优化问题的Pareto最优解,大量仿真结果验证了NSGA2TO的优越性能。此外,还探究了Pareto最优解所涉及的延迟与能耗之间的关系,有助于更好地理解车辆任务卸载问题的复杂性。通过合理平衡延迟和能耗,将能够进一步提升车联网系统的性能和效率,为用户提供更安全、更便捷的出行体验。

关键词: 车辆边缘计算;任务卸载;多目标优化;NSGA-II;帕累托最优解

中图分类号 TP311

Joint Optimization of Delay and Energy Consumption of Tasks Offloading for Vehicular Edge Computing

LI Wenwang, ZHOU Haohao, DENG Su, MA Wubin and WU Yahui

National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China

Abstract The combination of the Internet of Vehicles(IoV) and connected autonomous vehicles(CAV) has promoted the rapid development of autonomous driving technology, but it has also created a huge demand for computing resources, which is challenging to resource-constrained vehicles. Vehicular edge computing(VEC) offers an entirely new solution. By offloading tasks to edge servers deployed in the roadside unit(RSU), we are able to service the IOV in a more efficient way. However, resource pre-emption will occur when multiple vehicles send offloading requests at the same time, which will increase the task processing delay. How to efficiently dispatch resources to maximize the quality of service is an urgent problem to be solved. To solve this problem, we treat it as a multi-objective optimization problem and propose a task offloading algorithm named NSGA2TO based on non-dominated sorting genetic algorithm-II. The algorithm can find the Pareto optimal solution of multi-objective optimization problems, and extensive simulation results verify that NSGA2TO outperforms counterparts. In addition, we also explore the relationship between the delay and energy consumption involved in the Pareto optimal solution, which helps to better understand the complexity of the vehicle task offloading problem. By properly balancing delay and energy consumption, we will be able to further improve the performance and efficiency of the connected autonomous system, providing users with a safer and more convenient travel experience.

Keywords Vehicular edge computing, Tasks offloading, Multi-objective optimization, NSGA-II, Pareto optimal solution

1 引言

近年来,随着城市化进程的推进,交通拥堵、事故频发等问题日益凸显,给人们的出行带来了极大的不便和安全隐患^[1]。面对这一局面,车联网(Internet of Vehicles, IoV)技术和联网自动驾驶汽车(Connected Automated Vehicles, CAV)应运而生,成为了改善交通状况的重要措施。车联网技术旨在通过车辆之间、车辆与基础设施之间的互连互通,实现信息

的共享和交互,提供实时的车辆和道路信息,使驾驶员可以更准确地了解交通流量、路况、停车位等情况,从而做出更明智的驾驶决策,提高行车的安全性和效率^[2]。联网自动驾驶汽车则是在车联网技术的基础上发展起来的一种创新交通方式,其利用车联网技术收集和大量数据,结合传感器、人工智能等技术,实现车辆的自主感知、决策和控制^[3]。车联网IoV与联网自动驾驶汽车CAV的结合推动了自动驾驶技术的飞速发展,提高了安全性、舒适性和效率,为人们的出行提

基金项目:国家自然科学基金(61871388)

This work was supported by the National Natural Science Foundation of China(61871388).

通信作者:周浩浩(haohaozhou@nudt.edu.cn)

供了全新的体验。然而,尽管车联网技术和联网自动驾驶汽车有着广阔的发展前景,但在其落地实施过程中仍存在挑战和不足之处,其需要庞大的数据生成和传输,例如,一辆联网汽车每小时大约产生 25 GB 的数据^[4],而智能交通系统等物联网应用对延迟十分敏感,传输延迟要求小于 1ms^[5]。这对资源受限的车辆提出了重大挑战。

车辆边缘计算(Vehicular Edge Computing, VEC)的出现提供了新的解决方案,它通过将计算密集型任务卸载到路边单元(Road Side Unit, RSU)上配备的边缘服务器(Edge Server, ES),从而实现低延迟和减少能耗^[6]。在 VEC 中,每辆车设备同时只能关联一个路边单元^[7],一个 RSU 则需要为多辆车服务,如何高效、经济地利用有限资源寻找车辆任务卸载的最优解,以最小化延迟并降低能耗是非常具有挑战性的。

在车辆边缘计算领域,任务卸载(计算卸载)问题已经得到广泛的研究。这些研究通常将任务卸载定义为延迟或能耗的最小化优化问题,提出了分布式资源分配方法(如博弈论方法^[8])、基于最优化的集中式资源分配方法^[9]、启发式算法^[10-11]等。此外,近期也有学者提出了基于人工智能^[12-13]或深度强化学习^[14-16]的方法来解决这些问题。现有的工作主要集中在单一性能指标上,如降低车辆设备能耗、降低任务卸载延迟、最小化任务平均延迟等。然而,面对多样化的需求,任务卸载通常需要综合考虑多个因素。例如,为了减少能耗,可能倾向于将任务卸载到最近的 RSU,但由于 RSU 处理任务数量有限,可能会导致任务等待时间增加,影响卸载决策及服务质量,因此任务卸载需要综合考虑不同需求和性能约束,实现多目标优化。也有学者对延迟与能耗进行联合优化^[17-18],将两个优化目标加权后作为一个目标,每次只能得到一种权值下的最优解,并且在实际应用中权值难以科学定义。本文针对任务卸载延迟和能耗最小化,从多目标优化的角度处理车辆边缘计算中的任务卸载问题。为了解决这一问题,提出了一种基于改进非支配排序遗传算法的任务卸载算法(Non-dominated Sorting Genetic Algorithm-II Based Task Offloading Algorithm, NSGA2TO),用于获得 pareto 最优解,该算法能够找到最优的任务卸载方案,将总延迟和能耗最小化。本文的主要工作总结如下:

1) 在车辆边缘计算基础上,分析任务卸载的过程,在此基础上提出车辆任务卸载问题。

2) 考虑车辆综合性能,将车辆任务卸载问题表述为一个多目标优化问题,最小化总时延与能耗。

3) 提出一种基于改进非支配排序遗传算法的任务卸载算法(NSGA2TO),以获得 pareto 最优解。

4) 结合仿真实验验证 NSGA2TO 的性能,在得到 pareto 最优解的基础上探究延迟与能耗之间的关系。

本文第 2 章介绍了相关工作;第 3 章描述了系统模型;第 4 章介绍了所提出的 NSGA2TO 算法;第 5 章讨论了实验结果;最后总结全文。

2 相关工作

车辆任务卸载问题是车联网中日益关注的问题,其中的关键是研究如何有效地分配通信和计算资源,以降低延迟和减少能耗。

大多数物联网应用对延迟十分敏感,必须考虑降低车辆任务卸载的时延问题。Tang 等^[19]在考虑 VEC 系统的时空相关性的基础上,提出了一种时空计算卸载方案,从而降低时延。Liu 等^[20]通过将计算卸载决策问题表述为多用户博弈,提出了一种分布式计算卸载方案,有效减低了延迟。为了减少云边协作系统中所有设备的加权和延迟,Ren 等^[21]研究了云计算和边缘计算之间的协作,提出的协同云和边缘计算方案优于传统方案的时延性能。Pan 等^[22]提出了一种基于贪婪算法的方法,用于 5G 边缘计算的延迟服务卸载策略。

为了降低计算卸载过程中的能耗,Zhang 等^[23]提出了一种通过优先级分配和类型分类的三阶段高效计算卸载方案。Jang 等^[24]考虑了由于车辆速度不同而导致的通信环境变化,共同优化多车的卸载比例和资源分配,最小化车辆的总能耗。Yang 等^[25]考虑了车辆机动性的影响,提出了 BOCO 算法,首次将贝叶斯优化应用于 VEC 的任务卸载中。

上述研究都是有效的解决方案,它们通常是制定一个单目标优化问题,如最小化延迟或能耗。但是面对多样化的需求,任务卸载需要考虑不同需求和综合性能,实现多目标优化。本文在已有研究成果的基础上,进一步进行了研究,具体来说,考虑了不同计算能力要求的任务,将任务卸载建立在一个具有通信和计算的统一框架中,同时考虑了系统的综合性能指标,提出了一种多目标优化问题,使卸载过程中的延迟和成本最小化。此外,通过本文提出的 NSGA2TO 算法得到了问题的 pareto 最优解。

3 系统模型

车辆边缘计算网络框架如图 1 所示,本文考虑路边有 M 个路边单元 $RSU: R = \{r_1, r_2, \dots, r_M\}$,路边单元 RSU 主要由无线接入点(Access Point, AP)和边缘服务器 ES 组成,RSU 负责接收车辆请求、处理任务和数据传输,由于 ES 强大的计算能力,部署了 ES 的 RSU 可以完成任务处理操作,部署了 AP 的 RSU 负责数据传输。本文考虑 W 个 ES: $S = \{s_1, s_2, \dots, s_W\}$, $W < M$,即 M 个 RSU 中有 W 个部署了 ES。路边共有 J 辆汽车,汽车 j 产生任务 t_j , $t_j \triangleq \{\eta_j, \Pi_j, I_j, \theta_j\}$,其中 η_j 代表任务大小; Π_j 代表任务长度,即所需要的计算量; I_j 代表任务 t_j 的类型,可以分为传输任务(如故障警告传播)或计算任务(如实时定位与导航计算); θ_j 代表任务的输出数据量与输入数据量之比。本文的目标是以最小的时间成本和能耗成本将任务发送到 AP 进行传输和发送到 ES 进行计算。

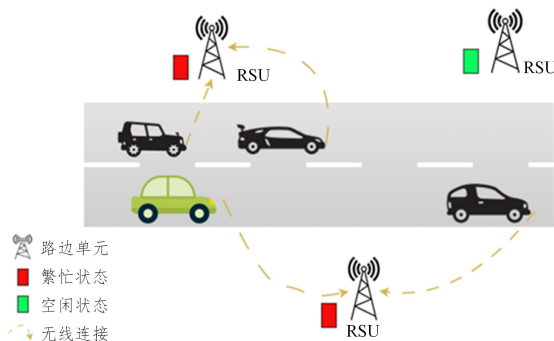


图 1 车辆边缘计算网络框架图

Fig. 1 Diagram of vehicular edge computing network frame

3.1 时间成本模型

车辆的输出数据需要通过 RSU 传输到云端,VEC 的任务是将 j 个车辆任务传输到 RSU 中,如果车辆任务类型为传输任务,则该任务需要传输到安装 AP 的 RSU 处理;如果车辆任务类型为计算任务,则该任务需要传输到安装 ES 的 RSU 处理,当任务 t_j 传输第 m 个 RSU(用 r_m 表示)时,由于有些 RSU 与任务类型不匹配或 r_m 处于繁忙状态,则需要判断在 r_m 的任务 t_j 是否需要迁移到其他与该任务匹配的 RSU。如果需要,则 $\psi_{j,m}=1$,否则 $\psi_{j,m}=0$ 。

$$\psi_{j,m} = \begin{cases} 1, & t_j \text{ needs to be transmitted in } r_m \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

需要判断车辆任务迁移到哪个 RSU,如果处于 r_m 的任务 t_j 迁移到 r_n ($m \neq n$),则 $\varphi_{m,n}=1$,否则 $\varphi_{m,n}=0$ 。

$$\varphi_{m,n} = \begin{cases} 1, & t_j \text{ in } r_m \text{ is transmitted to } r_n \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

对于每个车辆任务 t_j 需要计算它传输到 RSU 的时间。假设车辆任务大小为 η_j ,车辆 j 的传输速率为 f_j ($0 \leq f_j \leq F_v$),其中 F_v 代表车辆的最大传输速率,车辆 j 与 RSU 之间的距离为 $\epsilon_{j,m}$,则任务 t_j 的传输时间为:

$$A_j = \sum_{m=1}^M (1 - \psi_{j,m}) \frac{\eta_j}{f_j} \epsilon_{j,m} \quad (3)$$

当 RSU 与任务类型不匹配或 r_m 处于繁忙状态需要进行迁移时,我们需要计算迁移时间。假设 r_m 和 r_n 之间距离为 $\xi_{m,n}$,RSU 的数据传输速率为 ϑ_a ($0 \leq \vartheta_a \leq F_r$),其中 F_r 代表 RSU 的最大传输速率,则任务 t_j 的迁移时间为:

$$B_j = \sum_{m=1}^M \sum_{n=1}^M \psi_{j,m} \varphi_{m,n} \frac{\eta_j}{\vartheta_a} \xi_{m,n}, m \neq n \quad (4)$$

如果车辆任务为计算任务,则还需要考虑任务的计算时间。如果任务 t_j 在第 w 个 ES(用 s_w 表示)进行计算,则 $\gamma_{j,w}=1$,否则 $\gamma_{j,w}=0$ 。

$$\gamma_{j,w} = \begin{cases} 1, & t_j \text{ is computed in } s_w \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

本文用 ES 中空闲虚拟机 (VM) 的数量表示 ES 的计算能力,当车辆任务到达目标 ES 时,服务器将任务分配给 s_w 中的空闲虚拟机进行处理。设任务长度为 Π_j , s_w 中虚拟机数量为 u_w , u_w 的计算性能为 δ ,则任务 t_j 的计算时间为:

$$C_j = \sum_{w=1}^W \gamma_{j,w} \frac{\Pi_j}{u_w \delta} \quad (6)$$

当任务在 s_w 中处理完后需要回传给车辆,任务 t_j 的输出数据量与输入数据量之比为 θ_j ,车辆 j 与 ES 之间的距离为 $\zeta_{j,w}$,则任务 t_j 的回传时间为:

$$D_j = \sum_{w=1}^W \gamma_{j,w} \theta_j \frac{\eta_j}{f_j} \zeta_{j,w} \quad (7)$$

根据车辆的任务卸载过程,任务的 t_j 的时间成本 T_j 为该任务的传输时间、迁移时间、计算时间、回传时间之和,即:

$$T_j = A_j + B_j + C_j + D_j \quad (8)$$

3.2 能耗成本模型

与时间成本模型类似,车辆任务的传输、迁移、计算和回传都需要消耗能量。假设车辆 j 的传输功率为 p_j ,则任务 t_j 的传输能耗为:

$$X_j = \sum_{m=1}^M p_j (1 - \psi_{j,m}) \frac{\eta_j}{f_j} \epsilon_{j,m} \quad (9)$$

设 RSU 的传输功率为 λ_a ,则任务 t_j 的迁移能耗为:

$$Y_j = \sum_{m=1}^M \sum_{n=1}^M \lambda_a \psi_{j,m} \varphi_{m,n} \frac{\eta_j}{\vartheta_a} \xi_{m,n}, m \neq n \quad (10)$$

设 ES 一个 CPU 周期内的计算能耗为 β ,任务 t_j 的计算能耗为:

$$Z_j = \sum_{w=1}^W \gamma_{j,w} \Pi_j \beta \quad (11)$$

在任务计算结果的回传过程中,我们将回传能耗定义为:

$$K_j = \sum_{w=1}^W \lambda_a \gamma_{j,w} \theta_j \frac{\eta_j}{f_j} \zeta_{j,w} \quad (12)$$

根据车辆的任务卸载过程,任务 t_j 的能耗成本 E_j 为该任务的传输能耗、迁移能耗、计算能耗、回传能耗之和,即:

$$E_j = X_j + Y_j + Z_j + K_j \quad (13)$$

3.3 问题描述

对于给定的车辆任务 t_j ,由于任务类型不同,处理它会产生不同的时延和能耗,对于传输任务,只需要考虑传输和迁移过程中的时延和能耗,而对于计算任务,还需要考虑计算和回传过程中的时延和能耗。本文考虑在计算资源和通信资源的限制下,使所有车辆的时延和能耗最小化,需要对 $\psi_{j,m}$, $\varphi_{m,n}$ 和 $\gamma_{j,w}$ 进行优化,多目标优化问题可表示为:

$$\begin{aligned} \min_{(\psi_{j,m}, \varphi_{m,n}, \gamma_{j,w})} \quad & T = \sum_{j=1}^J A_j + B_j + C_j + D_j \\ \min_{(\psi_{j,m}, \varphi_{m,n}, \gamma_{j,w})} \quad & E = \sum_{j=1}^J X_j + Y_j + Z_j + K_j \\ \text{s. t.} \quad & W < M \\ & \psi_{j,m}, \varphi_{m,n}, \gamma_{j,w} \in \{0, 1\}, \forall j \in J, m \in M, w \in W \\ & \sum_{w=1}^W \gamma_{j,w} = 1, \forall j \in J, w \in W \end{aligned} \quad (14)$$

根据时延成本模型和能耗成本模型,原问题可表述为:

$$\begin{aligned} \min_{(\psi_{j,m}, \varphi_{m,n}, \gamma_{j,w})} \quad & T = \sum_{j=1}^J \sum_{m=1}^M (1 - \psi_{j,m}) \frac{\eta_j}{f_j} \epsilon_{j,m} + \sum_{j=1}^J \sum_{w=1}^W \gamma_{j,w} (\theta_j \frac{\eta_j}{f_j} \zeta_{j,w} + \frac{\Pi_j}{u_w \cdot \delta}) + \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^M \psi_{j,m} \varphi_{m,n} \frac{\eta_j}{\vartheta_a} \xi_{m,n} (m \neq n) \\ \min_{(\psi_{j,m}, \varphi_{m,n}, \gamma_{j,w})} \quad & E = \sum_{j=1}^J \sum_{m=1}^M p_j (1 - \psi_{j,m}) \frac{\eta_j}{f_j} \epsilon_{j,m} + \sum_{j=1}^J \sum_{w=1}^W \gamma_{j,w} (\lambda_a \theta_j \frac{\eta_j}{f_j} \zeta_{j,w} + \beta \Pi_j) + \sum_{j=1}^J \sum_{m=1}^M \sum_{n=1}^M \lambda_a \psi_{j,m} \varphi_{m,n} \frac{\eta_j}{\vartheta_a} \xi_{m,n} (m \neq n) \\ & W < M \\ & \psi_{j,m}, \varphi_{m,n}, \gamma_{j,w} \in \{0, 1\}, \forall j \in J, m \in M, w \in W \\ & \sum_{w=1}^W \gamma_{j,w} = 1, \forall j \in J, w \in W \end{aligned} \quad (15)$$

4 基于改进非支配排序遗传算法的任务卸载算法 (NSGA2TO)

4.1 多目标与帕累托优化

式(15)的目标是使时延和能耗同时最小化,这是一个多目标优化问题。为了处理多目标优化问题,主要有两种方法。一种方法是通过目标聚合将问题视为单目标优化问题,常规

方法是基于每个目标分配加权和的权重。但是,权重大小需要提前确定,并且需要通过大量的实验进行调整,这是非常难以实现的。此外,加权求和法只能近似于帕累托面为凸的情况,如果帕累托面为非凸,则该方法不能等价于原多目标优化问题。最后,这种方法只能获得一个折衷的解,但不能反映多目标之间的冲突特征。另一种方法采用多目标优化算法求解,通过求解多目标优化问题会得到一组最优解,这些最优解的元素称为帕累托最优解或非支配解,其中包含了大量的信息,如多目标之间的相关性和帕累托支配关系。最重要的是,即使系统发生变化,多目标优化算法生成的非支配解仍然有效,因此其更适合解决车辆任务卸载问题。

4.2 NSGA-II 算法

NSGA-II 算法的灵感来自于生物进化,它适用于复杂的多目标优化问题^[26]。与 NSGA 算法相比,该算法实现了更快、更准确的搜索性能,具有更好的收敛性、多样性和效果,得到了广泛的应用。其主要有 3 点改进:1)快速非支配排序;2)拥挤度比较;3)精英保留策略。为了明确地描述本文提出的 NSGA2TO,首先介绍了 NSGA-II 的一些术语。

1)个体(Individual):代表一个候选解的染色体,它由一串基因(Gene)组成。在多目标优化问题中,个体通常表示为一个向量,包含多个优化目标的值。

2)种群(Population):由一组个体构成的集合。种群表示了当前搜索空间中的解集合。

3)非支配(Non-dominated):一个解被称为非支配解,如果它在多个目标函数下都不被其他解支配。换句话说,没有其他解在所有目标函数上同时取得更好的结果。

4)支配(Dominance):如果一个解在某个目标函数上的取值比另一个解更好,并且在其他目标函数上的取值至少有一个比另一个解更好,那么这个解就被称为支配解。

5)非支配排序(Non-dominated Sorting):将种群中的解划分为多个等级(前沿),使得每个等级都包含非支配解。这个排序过程用于保持种群中的多样性以及帮助选择下一代解

6)交叉(Crossover):通过交换两个个体的基因片段来产生新的个体。在 NSGA-II 中,交叉是增加种群多样性和搜索空间的重要操作。

7)变异(Mutation):通过改变个体的某些基因来产生新的个体。变异有助于在搜索过程中引入新的解,增加搜索空间的广度。

8)拥挤距离(Crowding Distance):用于衡量解在搜索空间内的局部分布密度。拥挤距离较大的解说明其周围区域有较少的解存在,有助于保持种群中的多样性。

9)锦标赛选择(Tournament Selection):用于从种群中选择个体进行交叉和变异,在每个选择操作中保留较好的个体,并逐步优化种群中的解,能够有效地维持种群的多样性,并有助于提高优秀个体的生存和遗传到下一代的概率。

10)精英保留(Elitism):保存前几代中优秀的解到下一代种群中,以确保优秀解不丢失。

4.3 NSGA2TO 算法的整体流程

Step1 得到 Pareto 解集

1)初始化:初始化算法中的参数,包括:生成大小为 M 的随机初始种群 P_0 ,最大迭代次数 N ,交叉概率 p_1 ,变异概率

p_2 。在染色体 C_i 中,对每个基因对应的变量进行初始化。

2)评估个体适应度:根据问题的目标函数,计算每个个体的适应度值。

3)基于非支配排序和拥挤度距离计算对个体进行排序:根据个体的适应度值进行非支配排序,将个体划分为一系列不同的非支配等级。在每个等级中,通过计算个体的拥挤度距离来进行排序,拥挤度距离表示个体周围的分布密度,用于维持多样性。

4)选择:根据等级和拥挤距离,使用锦标赛选择方法从初始种群中选择父代种群。

5)生成新种群:对父代种群 $P(n)$ 进行交叉和变异操作,生成子代种群 $O(n)$ 。将父代种群和子代种群合并,生成大小为 $2M$ 的种群 $PO(n)$ 。通过精英保留,选择最优的大小为 M 的种群。

6)迭代:重复执行步骤 2)到步骤 4),直到达到最大迭代次数 N 。

7)存储:在所有迭代之后,保存 Pareto 解集 Z ,这些结果直接用于后续决策。

Step2 生成最优解

在得到 Pareto 解集后 Z 后, Z 表示满足最小时延和能耗的 M 个任务卸载决策,为了满足不同性能指标,我们提出了一个灵活的 P 模型,根据该模型选择最优卸载策略。

$$P_z = \frac{a(T_{\max} - T_z)}{T_{\max} - T_{\min}} + \frac{b(E_{\max} - E_z)}{E_{\max} - E_z} \quad (z \in Z) \quad (16)$$

其中, T_{\max} 和 E_{\max} 是解集 Z 中的最大时延和最大能耗, T_m 和 E_m 是策略 m 的时延和能耗, a 和 b 分别表示延迟和能耗的权重,两者之和等于 1,相对较小的 a 表示对延迟的容忍度较高,而较大的 a 则表示更关注实时性能;相对较小的 b 表示对能耗的容忍度较高,而较大的 b 则表示更关注节能。

算法 1 Non-dominated Sorting Genetic Algorithm-II Based Task Offloading Algorithm(NSGA2TO)

input:种群大小 M ;最大迭代次数 N ;交叉概率 p_1 ;变异概率 p_2

output:基于 NSGA-II 得到的 Pareto 解集 Z ;基于 P 模型得到的最优解 P_z

1. Initialize: $P_0 \leftarrow M, PO_1 \leftarrow P_0$
2. for $n \leftarrow 1$ to N do:
3. Calculate the fitness value T_j and E_j according to Formula(8) and (14)
4. $PO(n) \leftarrow$ Non-dominated sorting($PO(n)$)
5. $PO(n) \leftarrow$ Crowding distance($PO(n)$)
6. $P(n+1) \leftarrow$ Tournament Selection($PO(n)$)
7. $O(n+1) \leftarrow$ Crosses and mutates($P(n+1)$)
8. $PO(n+1) \leftarrow P(n+1) \cup O(n+1)$
9. $PO(n+1) \leftarrow$ Elite strategy selects($PO(n+1)$)
10. $n \leftarrow n+1$
11. endfor
12. Get the Pareto-optimal solution set Z
13. for z in Z do
14. $P_z \leftarrow P_{\text{model}}(\lambda, \mu)$
15. $P_{\text{best}} \leftarrow \min(P_z)$
16. endfor
17. return P_{best}

5 实验结果及讨论

本章将对本文提出的 NSGA2TO 方法进行评估,并与以下方案进行比较。

1) Benchmark: 将每个等待任务分配给邻近的 RSU, 它遵循临近原则, 不断重复搜索最近的 RSU。

2) VTO^[27]: 基于 SPEA-II 提出的一种高效的车辆任务卸载方法。

5.1 实验设置

本文考虑车辆数为 30~120, 每辆车包含车辆 ID 和位置信息, 实验参数如表 1 所列。实验环境是基于 12th Intel(R) Core(TM) i7-12700 CPU @ 2.10GHz × 12 的 Windows 11 系统, 我们使用的软件环境为 Python 3.9.2。

表 1 仿真参数

Table 1 Simulation parameters

Description	Parameter	Value
Number of Vehicles	J	30~120
Number of RSU	M	20
Number of ES	W	5
Number of VM	u_w	1~5
CPU capacity of a VM	δ	1~10
Power of RSU	λ_a	1~5
Computing energy consumption in a CPU cycle of ES	β	1~5
Length of Tasks	Π_j	1~100
Filesize of Tasks	η_j	1~500
Power of Vehicles	p_j	1~5

5.2 实验结果

5.2.1 帕累托解集

本文使用不同数量的车辆进行实验, 结果如图 2 所示, 很容易地观察到, 延迟目标和能耗目标的运行方向相反, 这意味着一个目标的提高可能导致另一个目标的下降。因为为了减少能耗, 可能倾向于将任务卸载到较近的 RSU, 但由于 RSU 处理任务数量有限, 这可能会导致任务等待时间增加, 从而增加延迟, 反之亦然。图 2 给出了不同车辆数量下 Benchmark, VTO 和 NSGA2TO 的性能比较。图 2(a) 中的车辆数为 30, 图 2(b) 中的车辆数为 60, 图 2(c) 中的车辆数为 90, 图 2(d) 中的车辆数为 120, 绿色

数据为 Benchmark 的解集, 红色数据为 VTO 的解集, 蓝色数据为 NSGA2TO 的解集。结果表明, Benchmark, VTO 和 NSGA2TO 三者之间的 pareto 最优解差异较大。对于 Benchmark 来说, 将每个等待任务分配给邻近的 RSU, 能耗固定, 但是延迟很高。VTO 和 NSGA2TO 的性能均优于 Benchmark, 且本文提出的 NSGA2TO 比 VTO 的效果更好。

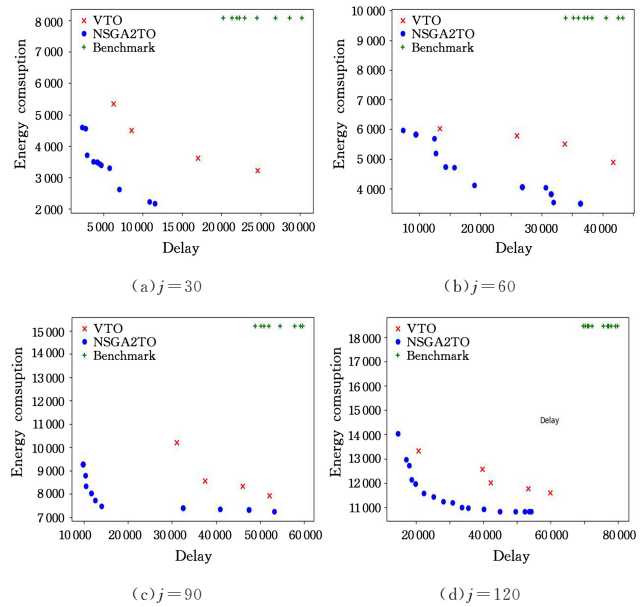


图 2 不同车辆数量下 Benchmark, VTO 和 NSGA2TO 帕累托最优解的比较(电子版为彩图)

Fig. 2 Comparison of Pareto-optimal solutions among Benchmark, VTO and NSGA2TO with different number of vehicles

另外我们比较了不同车辆数的情况下, 同一方法的差别。Benchmark 的实验结果如图 3(a) 所示, VTO 的实验结果如图 3(b) 所示, NSGA2TO 的实验结果如图 3(c) 所示, 红色数据为车辆数为 30 的解集, 蓝色数据为车辆数为 60 的解集, 绿色数据为车辆数为 90 的解集, 黄色数据为车辆数为 120 的解集。我们发现随着车辆数 J 的增加, 延迟和能耗都在增加。这是因为车辆越多, 需要处理的任務越多, 由于 RSU 的通信和计算资源有限, 延迟和能耗也就越高。

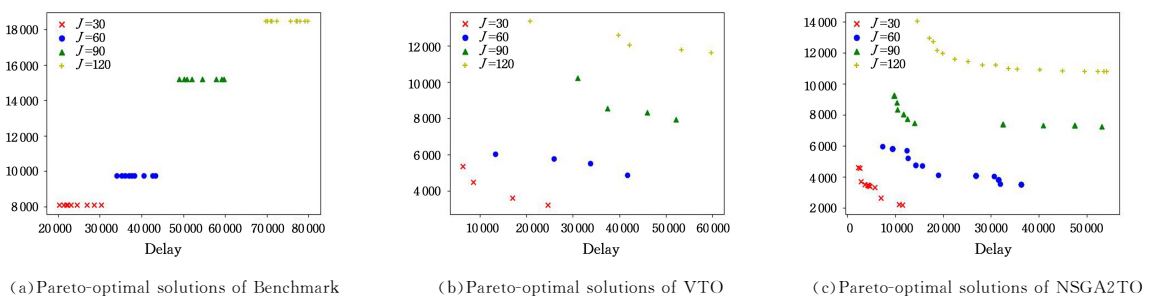


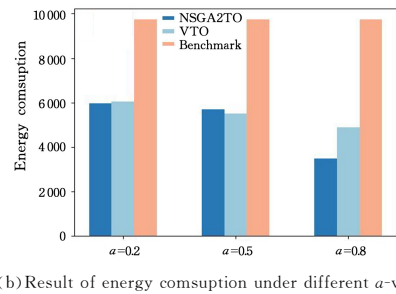
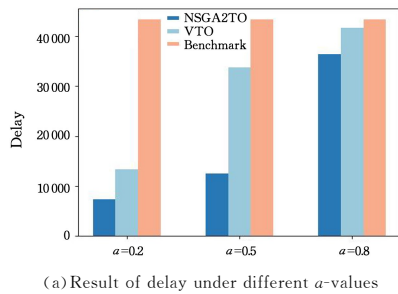
图 3 不同方案下的 pareto 最优解

Fig. 3 Pareto-optimal solutions in different schemes

5.2.2 最优解分析

为了满足不同性能需求, 本文根据式(16)的 P 模型改变 a 和 b 的值, 选择车辆数为 60 进行实验。延迟的实验结果如图 4(a) 所示, 能耗的结果如图 4(b) 所示。我们可以发现延迟

目标和能耗目标的运行方向相反, 这意味着一个目标的提高可能导致另一个目标的下降。但是不管更关注实时性能 (a 值较大), 或者更关注节能 (a 值较小), 或是综合考虑, 本文提出的 NSGA2TO 的性能均优于 VTO 和 Benchmark。

(a) Result of delay under different a -values(b) Result of energy consumption under different a -values图 4 不同 a 值下的最优解Fig. 4 Optimal solution at different a -values

结束语 本文通过综合考虑计算资源和通信资源的分配,对车辆边缘计算中的任务卸载问题进行了研究,旨在最小化总延迟和能耗。首先提出了车辆边缘计算的任务卸载问题,在此基础上考虑最小化总时延与能耗,将其表述为一个多目标优化问题。利用 pareto 最优的概念,提出了一种基于改进非支配排序遗传算法的任务卸载算法(NSGA2TO),用于获得 pareto 最优解,此外进行了大量仿真实验,验证了算法的性能,并分析了 Pareto 最优解所涉及的延迟与能耗之间的关系。

在今后的工作中,将考虑车辆的移动性的影响以及一辆车有多个任务的场景,继续对该方法进行改进和优化。

参考文献

- [1] HUANG Z, LOO B P Y. Urban traffic congestion in twelve large metropolitan cities: A thematic analysis of local news contents, 2009 – 2018 [J]. *International Journal of Sustainable Transportation*, 2023, 17(6): 592-614.
- [2] ANG L M, SENG K P, IJEMARUG K, et al. Deployment of IoV for smart cities: Applications, architecture, and challenges [J]. *IEEE Access*, 2018, 7: 6473-6492.
- [3] FAROOQU I, ISLAM M N, MUHAMMAD M K, et al. An Empirical Investigation of Performance Challenges Within Context-Aware Content Sharing for Vehicular Ad Hoc Networks [J]. *Trans. Emerging Telecommunications Technologies*, 2022, 33(10): e4157.
- [4] HEID B, HUTH C, KEMPFS, et al. Ready for inspection: The automotive aftermarket in 2030 [R]. McKinsey & Company, Tech. Rep., 2018.
- [5] RAJASEKHAR K, KUMAR R, KIRAN M. Next-Generation Technologies Empowered Future IoV [C] // 2022 IEEE 7th International Conference for Convergence in Technology (I2CT). IEEE, 2022: 1-5.
- [6] LU W, LEE W. Vehicular edge computing and networking: A survey [J]. *Mobile Networks and Applications*, 2000, 5(2): 101-102.
- [7] XU W, ZHOU H, SHEN X, et al. V2X Interworking via Vehicular Internet Access [M] // *Internet Access in Vehicular Networks*. Berlin: Springer, 2021: 57-82.
- [8] TANG J, LI X, JIN M, et al. A Mobility Aware Task Offloading Scheme For Vehicle Edge Computing [C] // 2021 13th In-

- ternational Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2021: 1-5.
- [9] FENG W, ZHANG N, LI S, et al. Latency minimization of reverse offloading in vehicular edge computing [J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(5): 5343-5357.
- [10] WEN Y H, ZHANG Q, YUAN H, et al. Multi-Stage PSO-Based Cost Minimization for Computation Offloading in Vehicular Edge Networks [C] // 2021 IEEE International Conference on Networking, Sensing and Control (ICNSC). IEEE, 2021, 1: 1-6.
- [11] DU J, SUN Y, ZHANG N, et al. Cost-effective task offloading in NOMA-enabled vehicular mobile edge computing [J]. *IEEE Systems Journal*, 2022, 17(1): 928-939.
- [12] LIANG D, MA L, LOU H, et al. An Adaptive Algorithm to Offload Task for User's QoE in Vehicular Edge System [C] // 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, 2023: 1263-1268.
- [13] ZHU L, ZHANG Z, LIU L, et al. Online Distributed Learning-Based Load-Aware Heterogeneous Vehicular Edge Computing [J]. *IEEE Sensors Journal*, 2023, 23(15): 17350-17356.
- [14] LU Y, AI B, ZHONG Z, et al. Energy-efficient task transfer in wireless computing power networks [J]. *IEEE Internet of Things Journal*, 2022, 10(11): 9353-9365.
- [15] CHEN X, DAI W, NI W, et al. Augmented Deep Reinforcement Learning for Online Energy Minimization of Wireless Powered Mobile Edge Computing [J]. *IEEE Transactions on Communications*, 2023, 71(5): 2698-2710.
- [16] ZHENG K, JIANG G, LIU X, et al. DRL-Based Offloading for Computation Delay Minimization in Wireless-Powered Multi-Access Edge Computing [J]. *IEEE Transactions on Communications*, 2023, 71(3): 1755-1770.
- [17] SHINDE S S, BOZORGCHENANI A, TARCHI D, et al. On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems [J]. *IEEE Transactions on Vehicular Technology*, 2021, 71(2): 2041-2057.
- [18] YADAV R, ZHANG W, KAIWARTYA O, et al. Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing [J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(12): 14198-14211.
- [19] TANG D, ZHANG X, TAO X. Delay-optimal temporal-spatial computation offloading schemes for vehicular edge computing

- systems[C] // 2019 IEEE Wireless Communications and Networking Conference(WCNC). IEEE,2019;1-6.
- [20] LIU Y, WANG S, HUANG J, et al. A computation offloading algorithm based on game theory for vehicular edge networks [C] // 2018 IEEE International Conference on Communications (ICC). IEEE,2018;1-6.
- [21] REN J, YU G, HE Y, et al. Collaborative cloud and edge computing for latency minimization[J]. IEEE Transactions on Vehicular Technology,2019,68(5):5031-5044.
- [22] PAN Z Y, CHEN J L, CHANG Y C. Low-latency computation offloading based on 5G Edge Computing Systems[C] // 2022 24th International Conference on Advanced Communication Technology(ICACTION). IEEE,2022;95-100.
- [23] ZHANG K, MAO Y, LENG S, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks[J]. IEEE Access,2016,4:5896-5907.
- [24] JANG Y, NA J, JEONG S, et al. Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation[C] // 2020 IEEE 91st Vehicular Technology Conference(VTC2020-Spring). IEEE,2020;1-5.
- [25] LV W, YANG P, ZHENG T, et al. Energy Consumption and QoS-Aware Co-Offloading for Vehicular Edge Computing[J]. IEEE Internet of Things Journal,2022,10(6):5214-5225.
- [26] KALYANMOY D. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Trans. on Evolutionary Computation,2002,6(2):182-197.
- [27] WU Q, XU X, ZHAO Q, et al. Tasks offloading for connected autonomous vehicles in edge computing[J]. Mobile Networks and Applications,2022,27(6):2295-2304.



LI Wenwang, born in 2000, postgraduate. His main research interests include edge computing and performance evaluation.



ZHOU Haohao, born in 1988, Ph.D, associate research fellow. His main research interests include edge computing and performance evaluation.