

基于改进量子遗传和 QoS 感知方法的车联网 云雾计算系统任务调度策略

张福琦^{1,2}, 姜会林^{1,3}, 刘富¹, 侯涛¹, 刘禹佳⁴, 关岳琦⁵, 沐星彤⁵

(1. 吉林大学通信工程学院, 吉林 长春 130012; 2. 奥克兰大学电气、计算机和软件工程系, 新西兰 奥克兰 1010;
3. 长春理工大学空间光电技术研究所, 吉林 长春 130022; 4. 长春工程学院电气与信息工程学院, 吉林 长春 130012;
5. 中国移动通信集团吉林有限公司, 吉林 长春 130000)

摘要: 针对车联网云雾计算系统中任务调度的并发拥塞、QoS 多样性与资源分配复杂问题, 提出了基于改进量子遗传与 QoS 感知方法的调度策略。通过量子编码与旋转优化调度方案, 引入 QoS 平衡参数和负载均衡罚项, 提升完工时间、能耗与调度灵活性。仿真实验表明, 所提策略完工时间最多缩短 69.0%, 并在多项性能指标上表现优异, 有效助力用户与运营商实现双赢, 具有良好的推广价值。

关键词: 车联网云雾计算系统; 任务调度; 个性化服务质量需求; 改进的量子遗传算法; 网络拥塞

中图分类号: TN914

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025057

Task scheduling strategy for cloud-fog computing system of IoV based on improved quantum genetics and QoS awareness method

ZHANG Fuqi^{1,2}, JIANG Huilin^{1,3}, LIU Fu¹, HOU Tao¹, LIU Yujia⁴, GUAN Yueqi⁵, MU Xingtong⁵

1. College of Communication Engineering, Jilin University, Changchun 130012, China

2. Department of Electrical, Computer and Software Engineering, University of Auckland, Auckland 1010, New Zealand

3. Institute of Space Optoelectronics Technology, Changchun University of Science and Technology, Changchun 130022, China

4. School of Electrical Engineering and Information Technology, Changchun Institute of Technology, Changchun 130012, China

5. Group Jilin Co., Ltd. China Mobile Communications, Changchun 130000, China

Abstract: An improved quantum genetic and QoS-aware scheduling strategy was proposed to address concurrent congestion, QoS diversity, and complex resource allocation in IoV cloud-fog computing systems. Using quantum encoding and rotation-based optimization, along with QoS balancing parameters and load balancing penalties, the proposed strategy improved makespan, energy efficiency, and scheduling flexibility. Simulation results show that the proposed strategy is up to 69.0% reduction in makespan and strong performance across multiple metrics, supporting practical applicability.

Keywords: IoV cloud-fog computing system, task scheduling, personalized QoS requirement, improved quantum genetic algorithm, network congestion

收稿日期: 2025-02-10; 修回日期: 2025-03-18

通信作者: 刘富, liufu@jlu.edu.cn

基金项目: 国家自然科学基金资助项目(No.51835006, No.52021003); 吉林省科技发展计划基金资助项目(No.20210101184JC); 科技创新 2025 重大专项基金资助项目(No.2022Z018); 吉林省教育厅科学研究基金资助项目(No.JJKH20230707KJ)

Foundation Items: The National Natural Science Foundation of China (No.51835006, No.52021003), Jilin Province Science and Technology Development Plan Project (No.20210101184JC), Science and Technology Innovation 2025 Major Special Project (No.2022Z018), Scientific Research Project of Education Department of Jilin Province (No.JJKH20230707KJ)

0 引言

物联网技术的应用和智慧交通的发展促进了车辆对车辆、车辆对基础设施、车辆对行人和车辆对网络之间的高效协同^[1]。车联网能够显著提高道路安全和交通效率^[2-3]，不仅包含紧急刹车、车道保持等基础应用，还能够实现视觉检测、增强现实等新兴业务^[4-7]。随着车联网业务的种类和数量的快速增加，业务的突发性导致了网络数据流的不确定性。网络资源分配不均匀现象可能引发网络拥塞，不仅影响业务的服务质量（QoS, quality of service）还会大幅提升网络运营成本^[8-9]。

为了提升车联网资源的分配能力，该领域研究人员正在广泛地研究和应用云雾计算技术。云雾计算系统的框架通常包含终端层、雾层和云层^[10]。其中，雾层部署在终端层和云层之间，主要包含具有计算和通信功能的雾节点。当终端层产生业务请求时，附近的雾节点能够利用计算能力迅速处理部分任务，并利用通信能力将其余的任务传输到云层。因此，由雾层和云层组成的云雾计算系统能够优化用户的使用时延，避免网络拥塞，并大幅降低车联网带宽的利用率和网络业务的运营成本。

然而，车联网云雾计算系统面临任务调度这一挑战。目前，如何合理地为业务分配计算和通信资源仍然是学术界的热门话题。对于用户，任务调度策略要优化完工时间。对于运营商，任务调度策略要同时优化云服务器能耗、运维成本和网络带宽利用率。同时，任务调度策略还要规避突发的网络拥塞现象。在协同优化上述目标后，任务调度策略才能帮助用户和车联网运营商达成双赢的局面。

目前，车联网云雾计算任务调度领域仍面临诸多挑战，如数据样本有限、实验环境受限、应用拓展性不足，以及与先进算法的比较不足等。为了更好地解决这些难题，本文提出了一种基于改进量子遗传和 QoS 感知方法的任务调度策略（QSIQ, QoS-aware scheduling based on improved quantum genetic algorithm）。本文围绕系统架构、任务调度方法、多维指标优化及实验验证等方面进行了创新，具体包括以下 4 个方面。

1) 依据车联网实际情况建立了云雾计算系统的架构模型。以雾代理服务器为主要研究对象和策

略执行单元，模拟系统的任务调度和数据传输过程，并设定了资源和任务的属性和方法。

2) 提出了一种 QSIQ 方法。通过新型量子编码对任务请求、网络拓扑和链路带宽利用率等关键环节数值化。同时，利用量子旋转机制提高了种群的多样性和调度效率。

3) 优化云雾计算系统的完工时间、能耗、算法开销和负载均衡度等多个性能指标。还通过调节平衡参数，满足不同任务的个性化 QoS 需求，增强了任务调度的灵活性和适应性。

4) 进行了多维度的性能评估实验，并对比了多种先进优化算法，包括基于生命选择的优化算法（LCO, life-choice-based optimization algorithm）^[11]、改进的生命选择优化算法（ILOA, improved version of the life-choice-based optimization algorithm）^[12]、麻雀搜索算法（SSA, sparrow search algorithm）^[13]、鲸鱼优化算法（WOA, whale optimization algorithm）^[14]、混合改进鲸鱼优化算法（HI-WOA, hybrid improved WOA）^[15]和变量邻域搜索算法（VNS, variable neighborhood search）^[16]。数值仿真结果证明了本文方法的有效性、先进性和可信度。

1 相关工作

目前，研究人员致力于解决云雾计算系统的任务调度问题，并取得了很多进展，以下是近 3 年的相关研究工作。

雾计算设备具备分散、动态和资源受限的特点。为了解决异构雾计算资源时延敏感问题，Azizi 等^[17]提出了优先级感知半贪婪算法和多程序优先级感知半贪婪算法，将物联网任务传输到雾节点。首先，以数学形式表达任务调度问题；接着，在满足物联网 QoS 要求的同时，降低雾节点总能耗；然后，评价任务成功百分比、总能耗、任务失败总时长和系统完工时间 4 项指标；最后，将该方法的性能与其他方法进行对比。实验结果表明，该方法降低了总能耗和完工时间，并能为雾服务运营商提供最大的利润。

为了解决物联网传感器不稳定和网关过载导致的网络链路故障，Batista 等^[18]提出了一种基于雾计算的物联网环境范式。在优化云服务器利用率的同时，提出了一种用于雾计算网关、云服务器和网络链路的负载均衡方法，利用可编程软件定义网络实

现了物联网平台的负载均衡。通过实验,评估了任务失败率、响应时间和活动时间等指标。实验结果表明,在大多数场景中,该方法的性能优于轮询和最少连接等传统方法。

为了解决大规模云计算技术中的负载均衡问题,Kanbar等^[19]提出了一种新型的区域感知动态调度模型。该模型包含基于任务的双神经网络、改进的飞蛾火焰优化、改进的演员-评论家、虚拟机状态感知和霍普克洛夫特-卡普5项算法。通过实验,评估了时延、带宽、完工时间、吞吐量、能耗、CPU利用率、内存利用率、QoS劣化率和算法开销等QoS指标,证明了该模型的良好性能。

针对无人机雾计算中的动态特性难题,Min等^[20]提出了一种基于动态规划的任务卸载模型。通过模拟无人机的移动模式、雾节点的计算能力、物联网设备的通信约束和时延要求等因素,确定了最优的任务分配策略。该方法首先设计了一种无人机通信的通用框架,接着开发了基于雾的无人机物联网计算模型,然后设计了无人机雾计算的动态卸载策略,最后开发了一种对应的仿真平台。实验结果表明,所提模型优化了时延、可用性和资源成本等指标。

为了解决雾计算中资源利用率低和计算效率低的问题,Baek等^[21]提出了一种基于随机博弈和演员-评论家算法的强化学习框架。将联合优化问题表述为随机博弈,并在该框架中训练价值函数和消息函数。实验结果表明,该框架的链路成本、链路和服务器利用率及任务失败率指标均优于对比算法。

为了解决雾集群中的业务调度和任务分配问题,Ibrahim等^[22]提出了一种基于多目标深度强化学习的雾计算智能调度算法。首先,部署了一个中央智能雾代理服务器;然后,使用基于深度Q网络的多目标深度强化学习算法,根据负载、距离和优先级选择目标节点;最后,使用基于多目标分解的进化算法和非支配排序遗传算法选择最佳计算节点。仿真结果表明,该算法的任务完成时间、完工时间、排队时延、传输时延、处理时延、计算时延、CPU负载和存储利用率等指标表现良好。

为了解决密集雾网络下的任务卸载能耗问题,Zhao等^[23]提出了一种时延约束的动态电压调节和

传输功率控制方法。首先,将研究目标分解为已知匹配的能耗优化和雾服务器匹配2个子问题;然后,为了解决已知匹配的能耗优化子问题,提出了一种新型的基于部分计算卸载的服务器端能源优化算法;最后,将雾服务器匹配子问题转化为加权二分图匹配问题,并提出了一种基于匈牙利算法的匹配问题能源优化算法。仿真实验表明,该方案的任务失败率和能耗指标均优于对比算法。

为了解决云服务器需求激增和资源中断导致的负载均衡问题,Khaleel^[24]通过一种新的权重分配算法分配虚拟机上的任务。首先,使用迭代关键路径搜索、基于层次的优先级分配和多项式时间最长路径3种算法将任务敏感度分级;然后,使用改进的飞蛾火焰优化算法调度任务,并有效地减少了调度时延;最后,使用岩蹄兔优化算法将任务分配给最优的云服务器。另外,算法还向云调度程序提供虚拟机状态的基本信息。实验结果表明,该算法的时延、带宽、完工时间、吞吐量、能耗、QoS劣化率和算法开销等指标表现良好。

为了满足物联网设备中大量计算任务的QoS需求,Abdulazeez等^[25]提出了一种基于模糊逻辑的任务调度方法,对协作雾节点进行负载均衡操作。首先,根据远程卸载选择、雾节点选择、雾节点间协作激励、任务需求和计算节点异构性等因素做出最优决策;然后,根据每个任务的需求确定目标节点,并利用模糊逻辑调度任务;最后,使用一种新型的深度Q学习确定最优雾节点,并对集群内的雾节点执行负载均衡操作。实验结果表明,该方法的时延、能耗、网络使用率、吞吐量和卸载率等指标均优于对比算法。

为了提高雾设备和边缘设备部署过程中的网络连接能力和覆盖范围,Singh等^[26]提出了一种雾设备和边缘设备内部通信的部署方法。首先,将设备部署过程建模为多目标问题;然后,为了保证雾设备和边缘设备的通信和覆盖,计算了多目标聚合函数;最后,提出了一种新型元启发Jaya算法,并优化了多目标函数,得到最佳的部署方案。实验结果表明,雾设备和边缘设备的覆盖范围均优于对比算法。

云雾计算系统相关工作内容总结如表1所示,总结了任务调度研究的应用环境、优化目标、主要优势和未来方向。

表1 云计算系统相关工作内容总结

参考文献	方法名称	应用环境	优化目标	主要优势	未来方向
文献[17]	PSG和PSG-M	雾计算	传输成功任务百分比、总能耗、传输失败任务总时长和系统完工时间	专注于研究异构雾网络中物联网任务的调度,尤其是在满足时延约束条件下,优化系统的总能耗	计划改进物联网任务调度算法,并评估该算法在不同真实世界数据集下的性能
文献[18]	基于雾计算的物联网范式	云雾计算	任务丢失率和响应时延和活动时延	提出了一种雾计算网关范式和一种云和网络链路的负载平衡方法,主要提高基于雾计算的物联网场景的性能	未来将继续评估不同的网络拓扑和网络配置,并改进网络的性能和软件定义网络的范式
文献[19]	RADISH	云计算	时延、带宽、完工时间、吞吐量、能耗、CPU和内存利用率、服务水平违规和算法开销	为了解决物联网多云雾环境中的动态任务调度和负载均衡问题,提出了RADISH模型,主要目的是减少数据中心的高时延、数据包丢失和能耗	未来将继续研究多云雾环境中的安全性问题
文献[20]	基于动态规划的新型任务卸载模型	雾计算	时延、可用性和资源成本	通过引入飞行雾计算的概念,满足了在物联网设备低时延和高吞吐量的需求。	—
文献[21]	FLoadNet	云雾计算	链路成本、链路和服务器利用率及任务丢弃率	研究了3层雾网络中的联合链路和负载均衡问题,满足任务需求的同时调度任务	未来将利用新技术降低强化学习框架中的时间复杂度,并将该算法应用在大规模雾计算场景中
文献[22]	MODRL	云雾计算	任务完成时间、完工时间、排队时延、传播时延、传输时延、处理时延、计算时延、CPU负载和存储利用	解决了雾计算中的任务分配与调度问题,通过综合考虑负载、优先级和距离,选择最优节点	未来将调整该算法,并增强较长排队时间下的任务重调度机制
文献[23]	EOPCO-S、EOPCO-M	雾计算	任务失败率和能耗	在多终端设备和多雾服务器网络中,以最小化全局能耗为目标,将部分任务卸载问题转化为最优部分任务卸载和最优匹配2个子问题,并分别求解	—
文献[24]	DAG	云计算	时延、带宽、完工时间、吞吐量、能耗、QoS违规和算法开销	采用5种连续的算法解决负载均衡问题,该方案的探索能力和收敛能力都很强	扩展算法的适用性,并将算法应用在其他科学领域
文献[25]	DQN	雾计算	时延、能耗、网络使用率、吞吐量和卸载率	通过一种新型卸载策略提高了雾计算环境中的应用性能,主要涉及雾架构不同层中的传感数据	未来还将专注于安全参数以外的任务卸载研究,并将关注受损雾节点和节点移动性等关键因素
文献[26]	Jaya Algorithm	雾计算和边缘计算	覆盖范围	提出的新型雾设备部署方法既能够减少计算和通信开销,又能实现广泛的网络连接和覆盖	未来将专注于轻量级模型的部署,并探索机器学习在效率和资源优化等领域的应用

目前,云雾计算系统的任务调度问题取得了一系列进展和突破。与上述方法不同的是,本文聚焦于车联网云雾计算系统,并重点研究了城市交通管理网络的任务调度问题。旨在解决大量任务并发导致的网络拥塞问题,还将优化任务调度过程中的完工时间、能耗、算法开销和负载均衡度等指标,以

满足用户和运营商的个性化QoS需求。

2 系统模型

本节描述了车联网云雾计算系统中的系统架构和计算、通信模型。其中,主要研究问题是云雾计算系统中的任务调度和流量控制过程,功能执行主体是云层和雾层之间的雾代理服务器。

2.1 系统架构

本文研究的架构模型由终端层、雾层和云层组成。当用户终端产生网络业务请求时，雾代理服务器会根据网络拓扑流量和服务器负载等因素调度任务的执行节点。为了在模型中更加贴近实

际网络，本文建模了多云雾节点系统，车联网云计算系统任务调度架构如图1所示，设定了26个节点组成的车联网云计算系统。其中，25号节点为交通管理中心，26号节点为车辆管理中心，4号、15号和23号节点为高速公路出入口，6号节点为

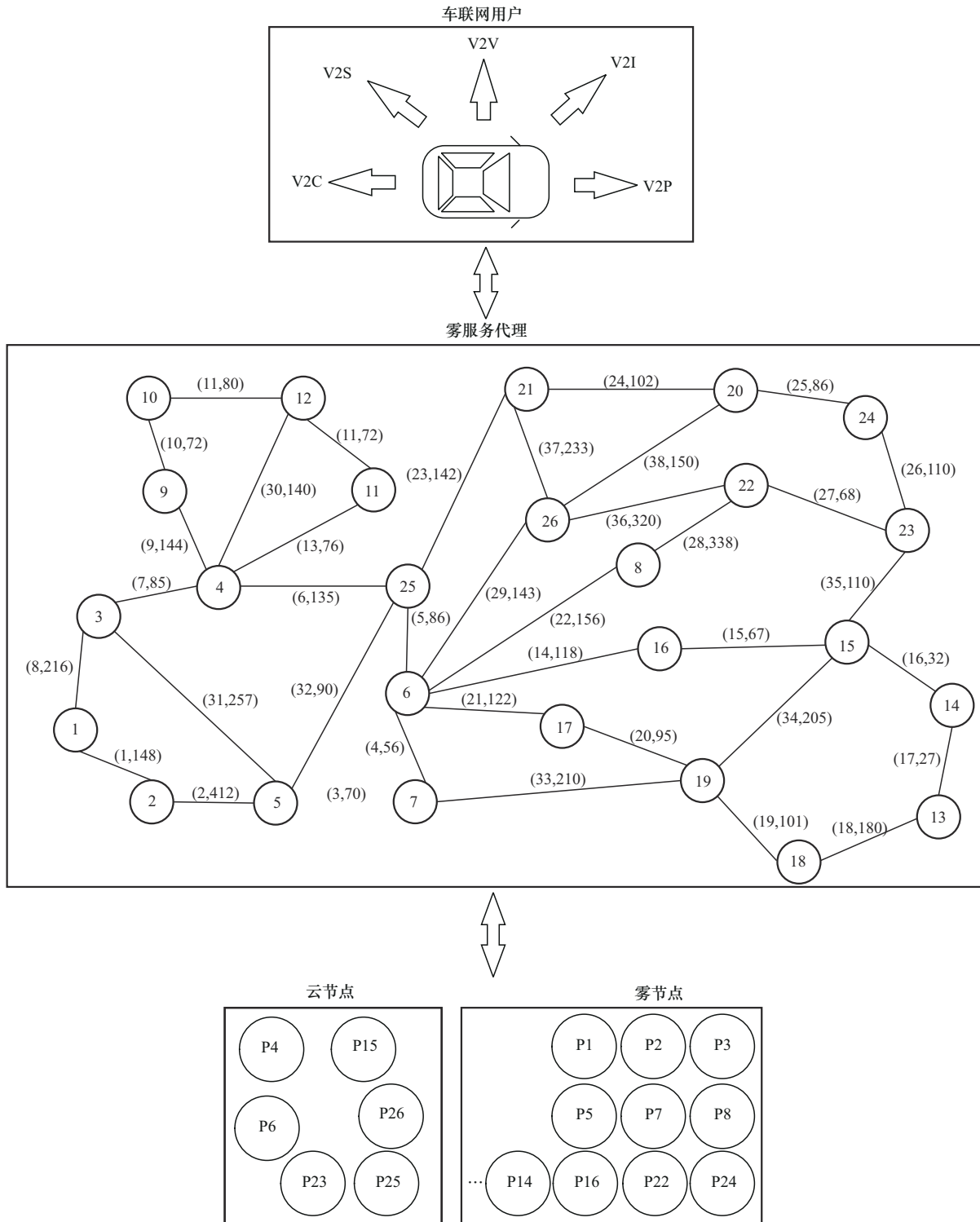


图1 车联网云计算系统任务调度架构

汇聚节点。在上述 6 个重要节点上, 分别部署一部云服务器; 其余为城市道路节点, 分别部署一部雾服务器。

车联网用户终端首先将任务请求发送给雾代理服务器, 雾代理服务器再将任务分配给云层或雾层。在本文的雾代理服务器模型中, 主要包含任务接收模块、云雾信息收集模块和任务调度模块 3 个部分。其中, 任务接收模块主要维护任务请求列表, 记录计算负载、通信负载、时间门限等参数, 并定期将信息传输到任务调度模块。云雾信息收集模块依托云雾路由拓扑图, 主要收集云节点和雾节点的状态。任务调度模块则根据当前任务的需求和车联网云雾计算系统的资源, 按照调度策略分配和传输待计算的任务。另外, 系统还会将处理后的任务信息返回给雾代理服务器, 而雾服务器也会将信息定向传输给提交任务请求的用户终端。

本文的雾代理服务器模型抽象了内部部署的调度策略和拓扑信息, 并将任务传输到云层或雾层。如果任务优先在云层计算, 雾代理服务器会调用任务调度模块, 将任务分配到目标云节点的服务器上处理。如果任务优先在雾层计算, 由于雾代理服务器和雾节点很近, 雾代理服务器能够更加迅速地将任务分配到雾节点上处理。

大量并发的任务会大幅增加任务调度的难度, 雾代理服务器的任务接收模块和任务调度模块会产生大量的数据流量。同时, 并发任务还会严重影响完工时间、能耗、算法开销和负载均衡度等指标, 使网络无法满足用户和运营商的个性化 QoS 需求。

2.2 问题公式化

在本文的车联网云雾计算系统模型中, 设定了 k 个独立计算和通信的任务, 记为 $T = \{T_1, T_2, \dots, T_k\}$ 。其中, T 为所有任务的集合。任务模型的通用属性包括输入和输出的数据量、处理任务所需的内存大小和计算任务时所需的 CPU 机器周期数。其中, 数据量和内存的单位为 Mbit, CPU 机器周期数的单位为百万指令。同时, 设定车联网云雾计算系统的每个节点记作 V , 节点总数记为 m 。云节点和雾节点分别记为 V_{cloud} 和 V_{fog} 。云雾节点的属性包括内存大小、网络带宽、存储容量和 CPU 的处理速率。其中, 网络带宽的单位是 Mbit/s, CPU 的处理速率为

每秒百万指令。其他属性的单位与任务属性的单位相同。

任务的预期完成时间 (ETC, expected time of completion) [27] 计算式为

$$\text{ETC}_{i,j} = \frac{\text{TL}_i}{S_j} \quad (1)$$

其中, TL_i 为处理任务 T_i 所需要的 CPU 计算周期数, $i \in \{1, 2, \dots, k\}$, S_j 为计算节点 CPU 的处理速率, $j \in \{1, 2, \dots, m\}$ 。由式(1)可知, $\text{ETC}_{i,j}$ 的单位为 s。

完工时间记为 $\text{MS}(T)$, 是所有任务均处理完成的总时间, 计算式为

$$\text{MS}(T) = \max_j \sum_{i=1}^k \text{ETC}_{i,j} \quad (2)$$

由式(2)可知, $\text{MS}(T)$ 的单位也为 s。

云雾节点 V_j 能耗的计算式为

$$E(V_j) = \text{ETC}_j v_j + (\text{MS}(T) - \text{ETC}_j) u_j \quad (3)$$

其中, v_j 为云雾节点的工作能耗系数; u_j 是常数, 表示节点的待机电耗系数; v_j 是 CPU 利用率 U_j 的函数, 计算式为

$$v_j = a_j U_j + u_j \quad (4)$$

其中, a_j 是一次项系数, 表示耗能随着 U_j 的上升而上升。另外, 每个节点的待机时间按 $\text{MS}(T) - \text{ETC}_j$ 计算, 即在云雾计算系统尚未处理完所有任务时, 已完成各自任务的云雾节点处于待机状态。

3 基于 QSIQ 方法的任务分配与流量控制策略

车联网云雾计算系统可以为用户提供网络、算力、通信和社交媒体等服务。为了降低完工时间、能耗、算法开销, 并均衡网络负载, 本文提出了一种基于 QSIQ 方法的任务分配与流量控制策略, 以满足不同的个性化 QoS 需求。QSIQ 方法作为一种混合元启发式算法, 由改进的量子遗传算法和 QoS 感知调度组成。其中, 改进的量子遗传算法能够根据任务请求、网络拓扑和链路带宽利用率进行量子编码, 并通过量子旋转丰富种群数量, 获得最优解后解码输出最优的流量控制策略。QoS 感知调度能够依托雾代理服务器, 检测过载云服务器, 并将重载任务卸载到雾服务器上。QSIQ 方法在效用函数中还增加了负载均衡罚项, 能够有效均衡链路带宽

利用率,有效解决网络拥塞问题。同时,本文方法的时间复杂度低,能够满足车联网云雾计算系统的部署要求。

3.1 效用函数与适应度函数的设计

经典的“效用”是经济学中最基本的概念之一。但在多目标优化框架下,效用函数常常难以平衡各项待优化的指标。本文改进了群体智能中的“效用函数”^[28],通过引入平衡参数,协同优化多项个性化QoS指标。同时,为了更好地兼顾车联网云雾计算系统的负载均衡度,适应复杂、变化的实际网络环境,本文通过在效用函数中增加负载均衡罚项,重新定义了一种新的“适应度函数”。适应度函数模拟了自然选择和进化过程,并通过评价每个个体的适应度水平,最终使任务分配与流量控制的效果更接近全局最优^[29]。

3.1.1 效用函数

任务分配与流量控制过程中的完工时间和总能耗对车联网云雾计算系统的性能起到决定性作用。为了协同优化两项指标,本文首先设计了效用函数,将多目标问题转化为单目标问题。云雾计算系统节点效用函数的计算式为

$$Uti_j = \lambda_i \times E(V_j) + (1 - \lambda_i) \times MS(T) \quad (5)$$

其中, λ_i 为任务 T_i 的平衡参数,根据任务的个性化QoS需求, λ_i 的取值范围最大可以为(0, 1),用来调整完工时间和总能耗的优化力度和偏好。在后文中,任务分配与流量控制的目标是使效用函数最小,并使完工时间和总能耗满足计算用户和运营商的个性化QoS需求。

3.1.2 适应度函数

因为云雾节点能耗系数的计算式是CPU利用率 U_j 的函数,所以在任务分配与流量控制的过程中,效用函数的优化目标可能导致任务被集中分配到某个能耗系数较小的云节点。这会使云雾计算系统中产生网络拥塞现象,不仅使完工时间指标劣化,还会由于大量云雾节点处于待机状态,增加车联网云雾计算系统的能耗。为了规避网络拥塞问题,本文在适应度函数中增加了负载均衡罚项,以弥补效用函数的不足。负载均衡罚项 ζ 和适应度函数 F 的计算式为

$$\zeta = \exp\left\{\left[\max U_j(V_{\text{cloud}}) - \min U_j(V_{\text{cloud}})\right]\gamma\right\} - 1 \quad (6)$$

$$F = \sum_{i=1}^m Uti_i + \zeta \quad (7)$$

其中, $\max U_j(V_{\text{cloud}})$ 和 $\min U_j(V_{\text{cloud}})$ 分别是当前所有云节点的最大CPU利用率和最小CPU利用率; $\gamma = \frac{k}{m}$ 代表负载均衡系数,当计算任务数量增多时,网络拥塞的风险随之增大。负载均衡系数的快速增加可以更严厉地惩罚效用函数,使任务分配与流量控制的效果达到全局最优。改进的自适应函数不仅能够提升任务调度效率、增强算法的泛化能力,还能避免负载不均引发的网络拥塞,并满足用户和运营商的个性化QoS需求。

3.2 QSIQ方法

基于经典遗传算法的调度方法在面对大量并发任务时,经常出现收敛速度慢和陷入局部最优的问题。因此,本文结合改进量子遗传算法与QoS感知调度,提出一种新型的QSIQ方法。首先,改进的量子编码将任务请求、网络拓扑和链路带宽利用率编码为量子比特。接着,使用量子旋转门进行量子测量和量子变异。然后,执行QoS感知调度,为雾代理服务器获取用户任务所需的CPU和内存需求,继续迭代改进的量子遗传,并获取全局最优解。最后,依据调度策略调度任务,分配计算和通信资源。

3.2.1 改进的量子遗传算法

传统的遗传算法存在编码过长、求解适应度函数次数过多等问题,通常在推导有效解时效率不高。而量子遗传算法可以通过调整概率幅值来提高收敛速度和精度,并通过引入量子门来实现染色体的变异和交叉,从而有效提高推导有效解时的效率。量子遗传是一种基于量子计算原理的遗传算法。它的编码方式与二进制编码、实数编码和树形编码等遗传算法编码方式不同,采用一种基于量子比特的编码方式,即用量子比特来存储和表达一个基因。量子比特可以处于0和1这2个本征态或2个状态的任意叠加,因此可以表达多个可能的解,增加了种群的多样性和搜索空间。

量子遗传算法利用量子比特来存储和表达每个基因,量子比特的状态表示为

$$|\phi\rangle \geq \alpha |0\rangle + \beta |1\rangle \quad (8)$$

其中, $|\alpha|^2$ 和 $|\beta|^2$ 分别表示量子比特处于“0”态和“1”态的概率,并满足 $|\alpha|^2 + |\beta|^2 = 1$ 。

由 q 个量子比特构成的量子染色体可以表示为

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_q & \cdots \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_q & \cdots \end{bmatrix} \quad (9)$$

其中, $|\alpha_q|^2 + |\beta_q|^2 = 1$, 该染色体可同时表达 2^q 个状态信息, 保证了种群的多样性。随着量子比特概率幅趋于 1 或 0, 染色体收敛到单一态, 种群多样性呈逐渐减小趋势, 算法逐步收敛至最终解状态。

在量子编码与译码环节, 本文为了减小算法的时间复杂度, 并获取全局最优, 提出了一种多效用量子编码译码算法。编码算法将任务请求、网络拓扑和链路带宽利用率数值化。首先, 对 t 时刻并发的全部 k 项任务进行路由规划, 由于车联网任务传输时延与路由总距离呈正相关^[30], 为了减小完工时间, 选择全部候选传输路径中最短的 3 条。接着, 对全部任务源宿节点对的 3 条路径按 10% 任务比例粒度遍历分配, 并保持 3 条路径的任务数据量总和为该任务传输流量的 100%。然后, 在全部 k 项任务比例分配中, 忽略距离最长路径占总流量 100% 的条目, 并将最长路径占总流量 90% 的两条合并为一项 (其余两条路径任务流量占比分别为 5%), 得到 64 种任务流量分配方案。最后, 全部任务流量分配的编号自减后按 6 位二进制数编码为量子比特, 保证量子比特编码在 “000000~111111” 之间。

在量子测量环节, 由染色体计算得到长度为 n 的二进制字符串, 数学公式表示为

$$\text{String} = \{x_1, x_2, \dots, x_i, \dots, x_n\} \quad (10)$$

其中, x_i 为二进制字符 0 或 1。具体测量过程为随机产生一个数 R , $R \in [0, 1]$, 若 R 大于 $|\alpha_i|^2$, 则 x_i 的数值取 1, 反之取 0。

改进的量子遗传算法使用量子旋转门执行量子变异环节。具体为一个 2×2 的矩阵算子, 算子的计算式为

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (11)$$

其物理意义代表复数的向量围绕原点逆时针旋转 θ 弧度, 量子门的旋转角为

$$\theta = g(x_q, \text{best}_q) \Delta \theta \quad (12)$$

其中, $g(x_q, \text{best}_q)$ 为旋转门方向, 取值空间为 $\{1, -1, 0\}$, 取值为 “1” 代表正向旋转, 取值为 “-1” 代表负向旋转, 取值为 “0” 代表不旋转; x_q 表示当前染色体的第 q 位; best_q 表示当前最优染色体的第 q 位; $\Delta \theta$ 为旋转角大小, 用以控制算法的收敛速度, 其数值太大可能导致算法运行结果发散或早熟, 从而得到局部最优解。改进量子遗传算法的旋转门转换示意如图 2 所示, 算法 1 为改进量子遗传算法的伪代码。

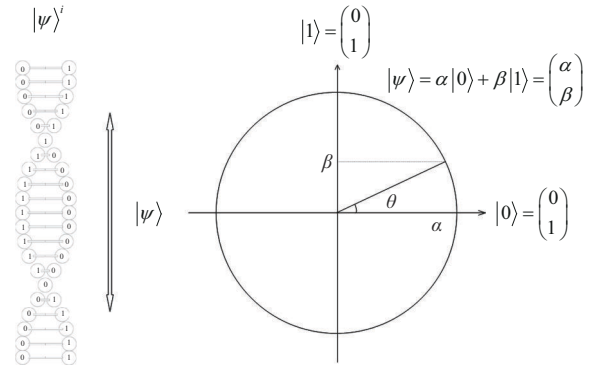


图 2 改进量子遗传算法的旋转门转换示意

算法 1 改进的量子遗传算法

输入 车联网云雾计算系统的节点位置、相对距离和计算能力, 网络链路的当前利用率和路由信息, 以及计算任务的数据量、源节点和目的节点等信息

输出 最优任务计算策略和对应的网络拓扑负载

- 1) $\text{gen} \leftarrow 0$
- 2) 对云雾计算系统的网络拓扑和任务信息进行初始化
- 3) for $i = 1$ to k
- 4) 为系统中节点的任务分配与流量控制策略量子编码为 $Q(\text{gen})$
- 5) 对 $Q(\text{gen})$ 进行量子测量
- 6) 计算效用函数 $F(\text{gen})$
- 7) 定义全局最优效用函数 F_g , 并将 F_g 初始化为 $F(\text{gen})$
- 8) end for
- 9) while $\text{gen} \leq \text{gen_max}$
- 10) $\text{gen} \leftarrow \text{gen} + 1$
- 11) 利用量子旋转门对 $Q(\text{gen})$ 进行量子变异
- 12) 对 $Q(\text{gen})$ 进行量子测量

- 13) 计算效用函数 $F(\text{gen})$
- 14) if $F(\text{gen}) < F_g$
- 15) $F_g \leftarrow F(\text{gen})$
- 16) $Q_g \leftarrow Q(\text{gen})$
- 17) end if
- 18) end while
- 19) 将 Q_g 解码为最优任务计算策略和对应的网络拓扑负载

3.2.2 QoS感知调度

当用户任务到达雾代理服务器时, 本文提出的调度策略首先执行算法2, 即过载检测阶段。具体而言, 调度策略为雾代理服务器获取用户任务所需的CPU和内存需求, 并检测全部云节点的CPU和内存走势。算法2每次能够返回一个过载云节点的信息。云节点 $V_{\text{cloud}i}$ 当前的CPU和内存利用率计算式为

$$U_{\text{cloud}i}^{\text{CPU}} = \sum_{i=1}^k T_{\text{cloud}i}^{\text{CPU}} \quad (13)$$

$$U_{\text{cloud}i}^{\text{MEM}} = \sum_{i=1}^k T_{\text{cloud}i}^{\text{MEM}} \quad (14)$$

其中, $\sum_{i=1}^k T_{\text{cloud}i}^{\text{CPU}}$ 和 $\sum_{i=1}^k T_{\text{cloud}i}^{\text{MEM}}$ 分别代表云节点 $V_{\text{cloud}i}$ 上的CPU和内存需求, 计算单位为百万指令和Mbit; $U_{\text{cloud}i}^{\text{CPU}}$ 和 $U_{\text{cloud}i}^{\text{MEM}}$ 分别代表云节点 $V_{\text{cloud}i}$ 当前的CPU和内存利用率。如果当前节点的CPU和内存利用率超过总体资源容量, 则该云节点将被识别为过载节点。随后, 雾代理服务器将收集当前信息, 并启用调度模式。接着, 本文提出的QoS感知调度将进行雾节点计算卸载选择阶段。雾代理服务器上事先配备了任务调度模块, 并利用算法1迭代计算用户任务分配后的适应度函数。从而确定将要卸载的用户任务和目标云节点, 大幅降低完工时间、能耗和算法开销, 并有效避免网络拥塞。算法3给出了从过载的云节点中选择计算任务和卸载到目标雾节点的基本过程。

最后, 本文提出的感知调度策略将执行资源调度环节。QSIQ方法将一直执行, 直到适应度函数最小, 并保证全部云节点的负载均在正常范围内。同时, 保持每次车联网云雾计算系统刷新时, 适应度函数不大于均比上一时隙的数值。本文QSIQ方法不仅提升了收敛速度、获取全局最优, 还能有效避免网络拥塞问题。同时, 为了提高模型在不同数据集和应用场景下的鲁棒性, 本文改

进了量子编码和解码过程, 使其在多样化的网络条件下仍能保持良好的性能。改进后的QSIQ方法具有较强的泛化能力, 并增强了本文模型的实用性和可靠性, 还能在复杂的网络环境中高效、准确地调度网络任务。

算法2 云节点过载检测算法

输入 计算任务的目标云计算节点, 虚拟机的当前负载和下一时刻负载预测, 虚拟机的负载阈值 T

输出 判断云节点虚拟机是否过载的标志位

- 1) IsCloudVMOverloaded \leftarrow False
- 2) 初始化检测过载的窗口时段 j_{max}
- 3) 将云计算节点 $V_{\text{cloud}i}$ 添加至过载检测列表
- 4) 调用 $V_{\text{cloud}i}$ 的负载历史信息 $U_{V_i}(0) - U_{V_i}(t)$
- 5) 预测下一时刻负载信息 $U_{V_i}(t+1)$
- 6) for $j = 1$ to j_{max} do
- 7) if $U_{V_i}(t-j) > T$
- 8) IsCloudVMOverloaded \leftarrow True
- 9) Break
- 10) end if
- 11) end for
- 12) if $U_{V_i}(t) > T$
- 13) IsCloudVMOverloaded \leftarrow True
- 14) end if
- 15) if $U_{V_i}(t+1) > T$
- 16) IsCloudVMOverloaded \leftarrow True
- 17) end if
- 18) 将云计算节点 $V_{\text{cloud}i}$ 移出过载检测列表
- 19) Return IsCloudVMOverloaded

算法3 雾节点计算卸载选择算法

输入 过载云节点虚拟机列表, 重载任务阈值 T

输出 卸载计算任务列表, 及目标雾服务器

- 1) OffloadingTask \leftarrow Null
- 2) 由算法2获取标志位 “IsCloudVMOverloaded”
- 3) TaskList \leftarrow OverloadedCloudVM.GetTaskList, 获取过载云节点任务列表
- 4) TaskRatio \leftarrow Ucpu / Umem, 计算任务的CPU利用率和内存利用率比例
- 5) TargetFogService \leftarrow Null
- 6) for Task in TaskList

- 7) if TaskRatio > T
- 8) OffloadingTask ← Tsak
- 9) 由算法 1 获取 TargetFogService
- 10) end if
- 11) end for
- 12) Return OffloadingTask, TargetFogService

综上所述，本文提出的 QSIQ 方法最小化了完工时间、能耗、算法开销，并优化了负载均衡度，能够满足用户和运营商的个性化 QoS 需求。

3.3 策略实施流程

本文提出的 QSIQ 方法涉及完工时间、能耗、算法开销和负载均衡度等多个协同优化目标。为了更加快速、准确地找到全局最优解，以下将介绍策略实施流程，并会分析策略的时间复杂度。

3.3.1 策略实现

策略的实施流程如图 3 所示，具体的实现步骤如算法 4 所示。

算法 4 策略流程实现算法

输入 车联网云雾计算系统的网络模型参数、用户任务参数以及 QSIQ 方法的初始化参数

输出 QSIQ 方法适应度函数的全局最优解，与最优解对应的任务分配与流量控制策略，以及此时的完工时间、能耗、算法开销和负载均衡度等指标

1) 初始化云雾计算系统节点的 CPU、链路带宽和内存等计算、通信和存储资源模型，雾代理服务器在网络拓扑中的位置信息，以及用户任务的计算量、数据量和个性化 QoS 需求。

2) 计算适应度函数，对云雾计算系统节点的任务分配与流量控制策略进行编码和译码，并初始化 QSIQ 方法的染色体样本。

3) 根据式(10)~式(12)执行改进量子遗传算法测量环节和变异环节，计算新染色体的适应度函数。

4) 判断计算结果是否小于当前全局最优值。如果是，则更新全局最优值；反之，则继续执行步骤 3)。

5) 判断 QSIQ 方法迭代次数是否达到预设值，如果已达到，则根据 QoS 感知调度返回任务分配与流量控制策略，以及此时的完工时间、能耗、算法开销和负载均衡度等评价指标。如果迭代次数未达到预设值，则跳转至步骤 3)，继续执行 QSIQ 方法。

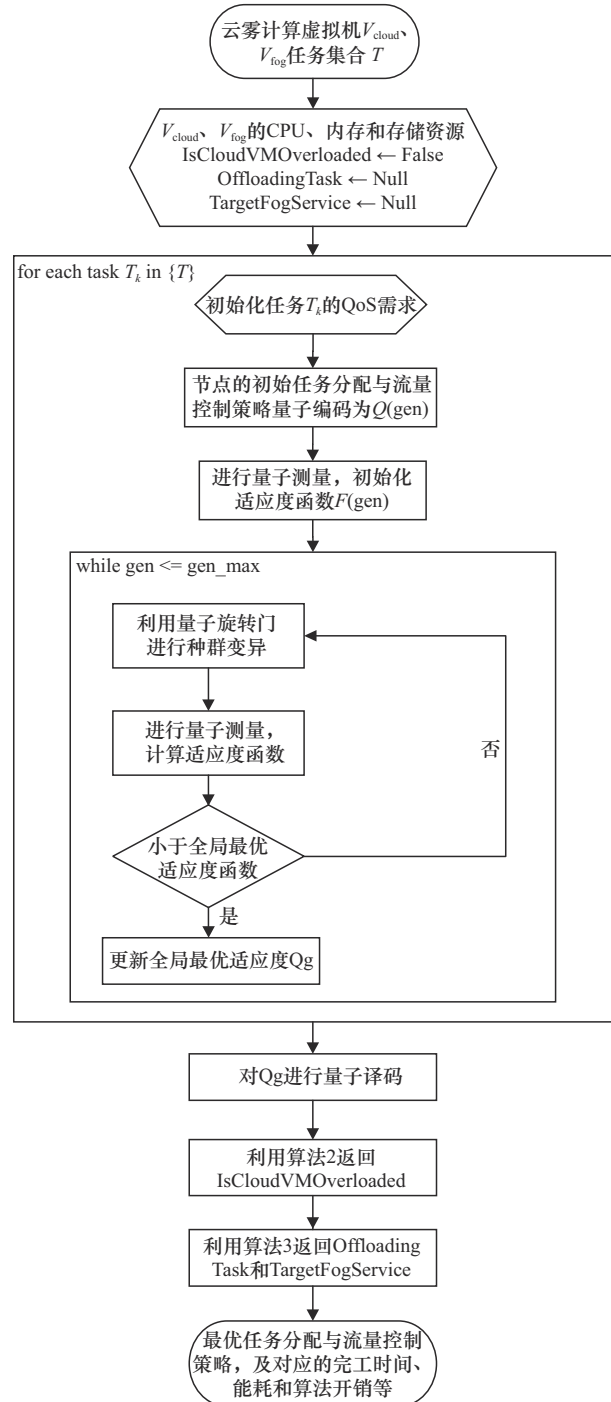


图 3 策略的实施流程

3.3.2 时间复杂度分析

因为本文策略需要协同优化完工时间、能耗、算法开销和负载均衡度等多个目标，所以必须评估策略的时间复杂度。该策略主要依赖 QSIQ 方法，并用于迭代计算多个效用函数和一个适应度函数。其中，计算单个效用函数的时间复杂度为 $O(mk)$ 。由于增加了负载均衡罚项，计算多个效用函数和适

应度函数的时间复杂度为 $O(m(k+1))$ 。在考虑最大迭代 n 次后,本文策略的时间复杂度计算式为

$$O(nm(k+1)) \quad (15)$$

4 数值仿真与实验结果

4.1 系统配置

本文利用MATLAB R2022a平台对提出的策略进行仿真验证,仿真环境为Intel(R) Core(TM) i7-12700H CPU @2.30 GHz, RAM为16.0 GB,并采用基于x64处理器的64位操作系统, GPU为NVIDIA GeForce RTX 3080 Ti Laptop,显存为16.0 GB。

将本文QSIQ方法与LCO、ILOA、SSA、WOA、HI-WOA和VNS这6种算法进行了详细的对比实验。所选的6种对比算法具有广泛的应用背景和研究基础。与本文QSIQ方法类似,对比算法均为当前流行的群体智能优化算法和搜索算法,涵盖了多种类型和优化策略。对比算法的多样性确保了对比结果的全面性和代表性,也有助于分析不同策略在复杂任务调度中的表现。与这6种算法的对比实验也反映了近年来在智能优化领域的研究趋势。随着车联网和云雾计算技术的发展,本文QSIQ方法和上述对比算法均能快速调度动态、变化的网络任务请求,并能有效提升QoS指标和系统性能。通过对比实验发现,LCO算法和ILOA算法适宜处理动态环境中的多目标优化问题,也擅长解决复杂约束条件的优化问题。SSA算法是一种新兴的群体智能算法,能有效处理大规模和复杂优化问题,也能通过简单而有效的搜索策略快速找到问题的局部最优解。而WOA和HI-WOA通过模拟鲸鱼的行为,在全局搜索能力上表现良好;HI-WOA还通过引入混合策略,进一步提高了搜索效率。VNS算法用于多个领域的组合优化问题,能在有限时间内完成任务调度。在实验中挖掘了本文方法和6种算法的对比维度,主要包括完工时间、能耗、算法开销、负载均衡度和统计检验。同时,实验还反映了所有算法的数据处理性能和任务调度效率。以上工作都有助于证明本文QSIQ方法的优越性能。最后,通过城市交通管理系统实例,以及多任务并发场景验证了本文方法和6种对比算法的性能,贴近真实的通信、网络和计算模型,并助于验证本文算法的实用性。

实验中建模了26个节点组成的云雾计算系统,

如图1所示,在6个重要节点上,分别部署一部云服务器虚拟机;其余为城市道路节点,分别部署一部雾服务器虚拟机。节点虚拟机计算能力如表2所示。

表2 节点虚拟机计算能力参数

编号	计算能力/GHz
1	2.7
2	2.5
3	3.0
4	3.5
5	3.0
6	3.5
7	2.6
8	2.7
9	3.0
10	2.9
11	2.5
12	3.1
13	2.8
14	2.7
15	3.5
16	2.4
17	2.9
18	2.7
19	3.0
20	2.5
21	2.8
22	2.7
23	3.4
24	2.9
25	3.5
26	3.5

同时,数值仿真中还设定了300个计算任务分批并发的调度场景。其中,每50个任务为一组,每个任务数据量大小为32~64 MB,同组数据并发至车联网云雾计算系统中。间隔1 s后,下一组任务并发至云雾计算系统。在5 s内,所有6组任务派发完成。雾代理服务根据本文策略进行任务调度和流量控制,将任务分配到对应的云层和雾层节点上。最后,由各个云雾节点虚拟机执行每项任务,并分别统计任务完工时间、能耗、算法开销和负载均衡度等指标,以此评价任务调度算法的性能。

4.2 完工时间对比实验

任务数量由 50 项增加至 300 项时的完工时间对比如图 4 所示。从图 4 可以看出，本文 QSIQ 方法的完工时间由 23.63 s 逐步增长至 227.50 s，LCO 算法的完工时间由 40.52 s 逐步增长至 248.19 s，ILCO 算法的完工时间由 38.62 s 逐步增长至 245.94 s，SSA 算法的完工时间由 84.63 s 逐步增长至 766.72 s，WOA 算法的完工时间由 55.26 s 逐步增长至 540.73 s，HI-WOA 算法的完工时间由 57.81 s 逐步增长至 418.33 s，VNS 算法的完工时间由 48.91 s 逐步增长至 734.52 s。因为 QSIQ 方法收敛速度快，并能将重载云节点上的任务调度到雾节点虚拟机上，所以完工时间表现最佳。另外，本文 QSIQ 方法还通过减少网络拥塞问题，满足用户的时延类个性化 QoS 要求。

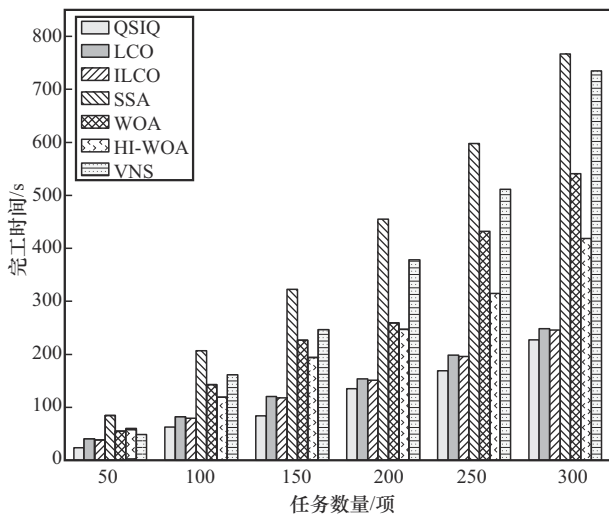


图 4 任务数量由 50 项增加至 300 项时的完工时间对比

4.3 能耗对比实验

任务数量由 50 项增加至 300 项时的能耗对比如图 5 所示。从图 5 可以看出，本文 QSIQ 方法的能耗指标优于其他 6 种对比算法。其中，QSIQ 方法的能耗由 0.02 kW·h 逐步增长至 0.25 kW·h，LCO 算法的能耗由 0.03 kW·h 逐步增长至 0.31 kW·h，ILCO 算法的能耗由 0.03 kW·h 逐步增长至 0.29 kW·h，SSA 算法的能耗由 0.04 kW·h 逐步增长至 0.31 kW·h，WOA 算法的能耗由 0.04 kW·h 逐步增长至 0.34 kW·h，HI-WOA 算法的能耗由 0.04 kW·h 逐步增长至 0.29 kW·h，VNS 算法的能耗由 0.046 kW·h 逐步增长至 0.32 kW·h。QSIQ 方法能够改变效用函数中的平衡参数，并实时监测云雾计算系统负载。当云雾计算系统重载时，雾代理服务器可以灵活地调节能耗优化的偏好。另

外，QSIQ 方法的全局搜索能力平衡了完工时间和能耗这两项不同的个性化 QoS 需求，能同时满足用户的时延类 QoS 需求和运营商的能耗类 QoS 需求。

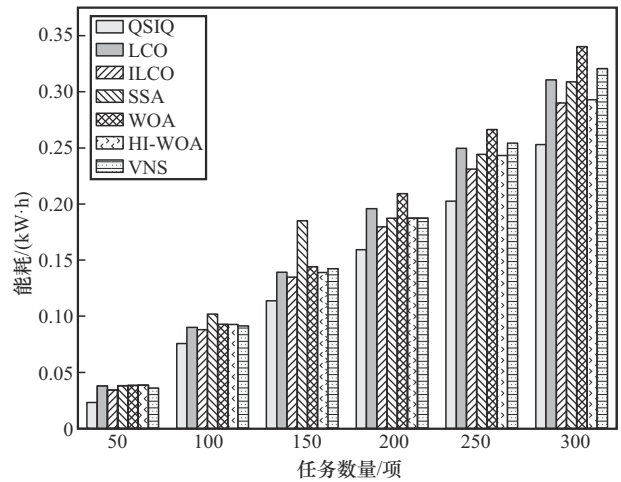


图 5 任务数量由 50 项增加至 300 项时的能耗对比

4.4 算法开销对比实验

任务数量由 50 项增加至 300 项时的算法开销对比如图 6 所示。从图 6 可以看出，本文 QSIQ 方法的算法开销与其他 6 种对比算法处于同一数量级，各项数值总体相近。当任务数量分别为 50 项和 100 项时，QSIQ 方法的开销分别为 401.06 元和 777.67 元，优于其他 6 种对比算法。这是由于任务数量较少时，QSIQ 方法输出的任务调度策略能够更有效地利用雾节点虚拟机的算力。在满足车联网云雾计算系统时延类和能耗类个性化 QoS 需求的同时，雾代理服务器将计算任务卸载到开销更低的雾服务器上计算。当任务数量增长至 250 项的过程中，QSIQ 方法增加了云节点虚拟机的使用时长，算法开销增长至 2 656.53 元。虽然 QSIQ 方法的开销多于其他 6 种算法，但 QSIQ 方法能够继续优化时延类和能耗类 QoS 指标。同时，QSIQ 方法通过计算适应度函数中的负载均衡罚项，有效解决了大量并发任务导致的网络拥塞问题。当任务数量增长至 300 项后，QSIQ 方法的开销增长至 2 986.88 元。该数值增长趋势趋于平稳，并最终位列 7 种算法中的第 4 名。这是由于随着时间的推移，QSIQ 方法已经将早期派发的任务调度到合适的节点。而新增任务对雾代理服务器调度策略的影响减小，QSIQ 方法产生的边际开销也随之递减。综上所述，本文 QSIQ 方法的开销与其他 6 种对比算法交替领先并且数值

相近, 均能满足300项任务逐步派发场景的QoS指标需求。

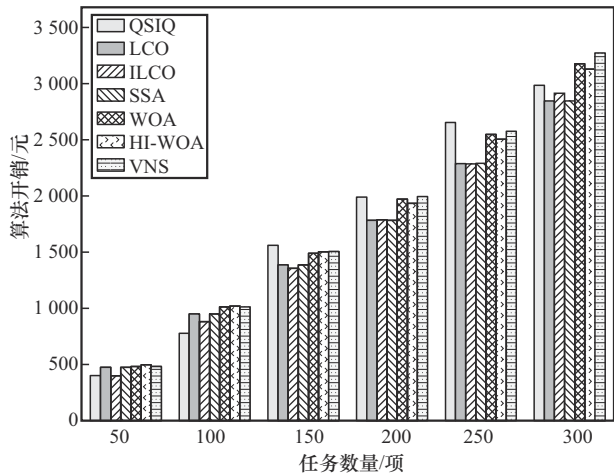


图6 任务数量由50项增加至300项时的算法开销对比

4.5 负载均衡效果对比实验

1) 负载均衡评价指标

在实验中, 用户终端首先在 t 时刻并发第一组计算任务, 而后的5 s内依次并发其他组计算任务。接着, 统计网络全部链路的带宽和实时流量。然后, 用除法计算链路的带宽利用率。最后, 计算不同算法链路带宽利用率的标准差和信息熵, 定量评价实施算法后的负载均衡效果。

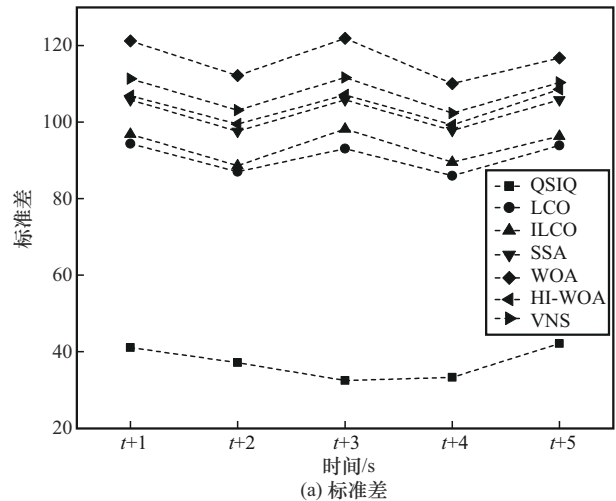
链路带宽利用率的标准差与信息熵随时间变化对比分别如图7(a)和图7(b)所示。在图7(a)中, 本文QSIQ方法链路利用率的标准差始终最小, 优于其他6种对比算法。当时间从 $t+1$ s进行到 $t+5$ s时, QSIQ的标准差由41.03波动到42.14, 其他6种对比算法的标准差始终处于85以上。大量车联网业务的源节点集中于云节点, 并发的计算任务导致部分与云节点连接的链路过载或重载。QSIQ方法执行了云节点过载检测算法, 能够有效地进行QoS感知任务调度。实验结果表明, 本文QSIQ方法能够很好地均衡网络链路中的负载。

在图7(b)中, 本文利用信息熵对比评价了7种算法的负载均衡效果。信息熵源于热力学中熵的概念, 计算式为

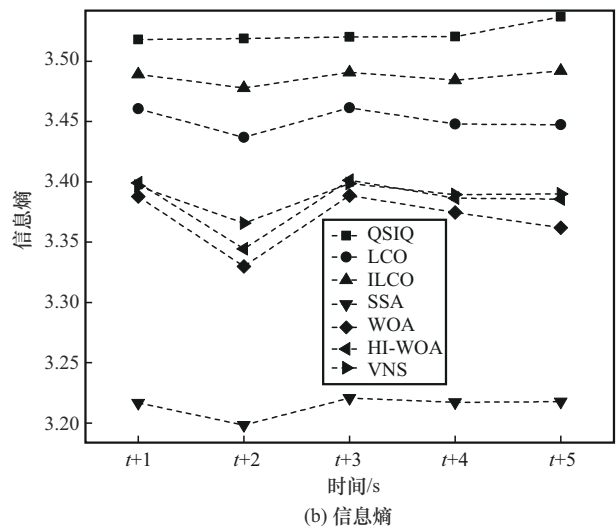
$$Entropy = \sum_{i=1}^e p_i \ln p_i \quad (16)$$

其中, $p_i = \frac{u_i}{\sum_i u_i}$ 是归一化的链路带宽利用率; $u_i = \frac{f_i}{b_i}$

为第 i 条链路的带宽利用率, 代表链路的业务繁忙程度; f_i 为第 i 条链路的总流量; b_i 为第 i 条链路的带宽; Entropy是评价算法负载均衡度的信息熵, 如果某种算法的信息熵越大, 则说明该算法所有链路的带宽利用率差别越小。实验结果表明, 本文QSIQ的信息熵由3.52波动到3.54, 始终优于其他6种对比算法。信息熵实验能够补充说明标准差实验的正确结果, 并证明QSIQ方法能够为运营商提供良好的负载均衡效果。



(a) 标准差



(b) 信息熵

图7 链路带宽利用率的标准差与信息熵随时间变化对比

2) 负载均衡效果

实施QSIQ方法前后部分链路的负载均衡效果分别如图8(a)和图8(b)所示, 其中, 圆圈代表网络节点, 数字代表节点编号。节点间的直线代表网络链路, 链路中间的数字分别代表链路编号和实际距离(单位为km)。链路附近的曲线密集程度代表链路的带宽利用率, 曲线越密集则带宽利用率越高。

其中，26号节点为车辆管理中心，是车联网中的重要节点，并配有云服务器虚拟机。在图8(a)中， t 时刻下26号节点并发大量计算任务，导致35号、25号、36号和27号链路过载，网络拥塞问题严重。实施QSIQ方法后，雾代理服务器改变了最短距离调度策略。同时，借助25号节点云服务器的帮助，重载的26号节点云服务器重新规划了部分计算任务的路由。在图8(b)中，链路间的曲线密集程度相近，过载链路的负载大大减轻。另外，QSIQ方法还有助于满足完工时间、能耗和算法开销等个性化QoS需求。综上所述，本文QSIQ方法能够有效调度云雾计算系统中的并发任务，并通过控制流量均衡不同链路的负载，有效解决了网络拥塞问题。

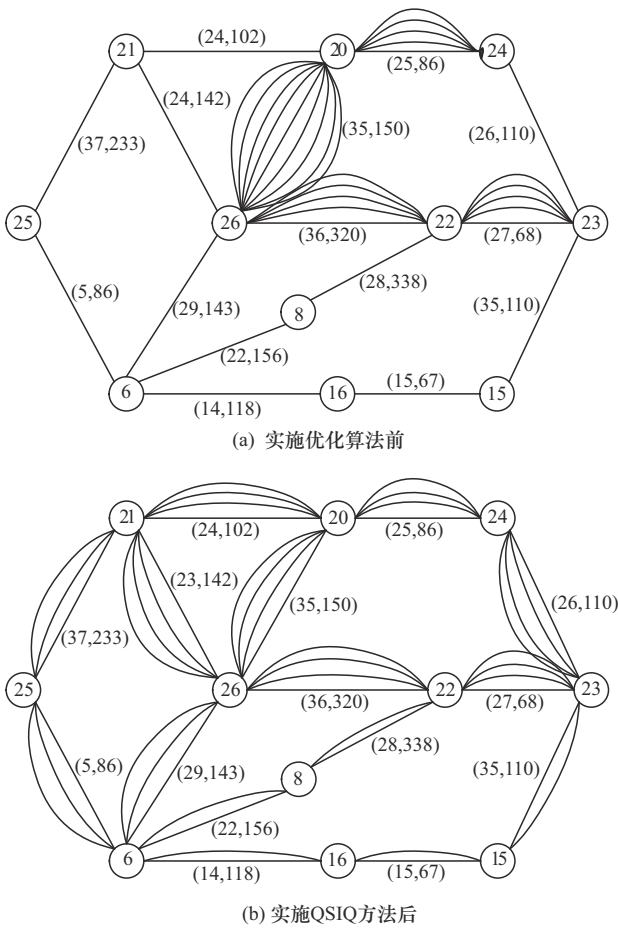


图8 部分链路的负载均衡效果示意

4.6 QSIQ方法收敛性实验

QSIQ方法的收敛性实验结果如图9所示。结果表明，当迭代次数达到约50次时，本文QSIQ方法收敛至全局最优信息熵值。当迭代次数接近30次时，QSIQ方法的信息熵数值仅与全局最优值相差

0.0166，偏差占比仅为0.47%。这是因为在经典的遗传算法和量子遗传算法中，交叉和变异操作引入的随机性过大，不利于输出结果的快速收敛。本文QSIQ方法改进了量子编码和解码的过程，具体而言，QSIQ方法预设了每个计算任务源宿节点间距离最短的3条路由，并分配了各个路由的流量比例。实验结果表明，计算任务的最小子任务占该任务数据量的10%。在调用QSIQ方法后，雾代理服务器在解码时查询编码表，以获取具体的任务调度策略。QSIQ方法还能预先部署在雾代理服务器中，使改进的量子编码解码操作能够大幅提高收敛速度。

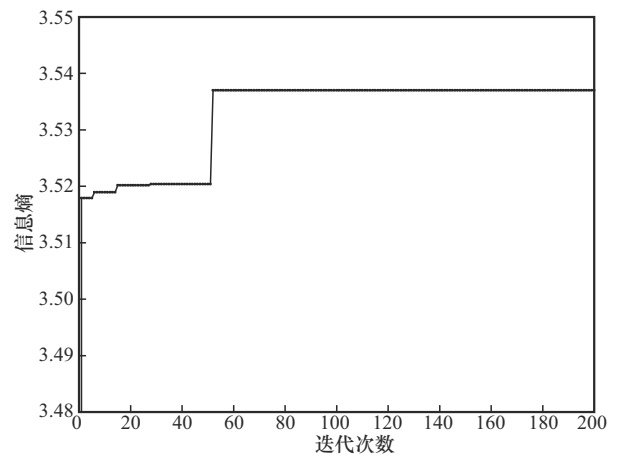


图9 QSIQ方法的收敛性实验结果

4.7 Mann-Whitney统计检验对比实验

Mann-Whitney统计检验的作用是评价两组数据的差异性。首先，对数据从低到高进行排序。接着，计算两组数据的秩和。如果秩和相似，则表示两组数据差异性不显著；反之，则表示两组数据的差异性非常显著。在Mann-Whitney统计检验中，还会计算U值、Z值和p值。其中，U值衡量两组样本中观测值排名差异的统计量，用于反映组间总体分布是否存在差异；Z值将U值标准化后的统计量用于正态分布近似检验，便于查表获得p值；p值表示观察到当前或更极端排名差异在原假设成立下出现的概率，用于判断差异是否具有统计学意义，当 $p < 0.05$ 时，统计检验认为数据间存在显著的差异性。

1) Mann-Whitney统计检验

在Mann-Whitney统计检验中，输入的是云节点和雾节点的任务调度时间。本文QSIQ方法和其他6种对比算法的U值、Z值和p值对比情况如表3所示。其中，总任务数量为300项。虽然QSIQ方

法的 U 值为 132, 略高于 LCO 和 ILCO 算法, 但 QSIQ 方法的 p 值仅为 0.008 1, 与其他 6 种算法相比数值最低。对比算法的 p 值最小为 0.039, 均高于 QSIQ 方法一个数量级。这说明在 Mann-Whitney 统计检验中, 本文 QSIQ 方法的云、雾节点任务调度时间存在显著差异, 明显改善了其他算法调度任务时的局限性。这是因为 QSIQ 方法在检测到云节点过载后, 能够充分利用雾节点的算力。QSIQ 方法还为雾代理服务器提供了全局流量控制策略, 解决了局部链路带宽利用率过高的网络拥塞问题。

表3 Mann-Whitney 统计检验的 U 值、 Z 值和 p 值对比情况

算法名称	U 值	Z 值	p 值
QSIQ	132	2.582	0.008 1
LCO	90.68	1.99	0.048
ILCO	110.36	1.532	0.045 3
SSA	632.71	1.157 5	1.149
WOA	586.97	1.286	1.312
HI-WOA	542.13	1.638	0.039
VNS	678.45	1.029	0.986

2) 事后分析

为了验证本文提出的 QSIQ 方法, 实验沿用 Mann-Whitney 检验进行数据的事后分析。图 10 展示了事后分析中的 p 值热力图。其中, 将本文 QSIQ 方法与其他 6 种对比算法两两进行分析。在纵横坐标交汇的正方形中, 标记了 2 种算法的事后分析结果。QSIQ 方法与其他 6 种对比算法均存在差异, 尤其对 HI-WOA ($p = 0.02$)、SSA ($p = 0.03$) 和 VNS ($p = 0.03$) 这 3 种算法存在显著性差异。

表 4 对比了 7 种算法事后分析中的 Z 值和 p 修正值。其中, Holm 和 Holland 修正值均属于多重比较中的 p 值校正策略。Holm 修正值是一种基于排序的逐步校正方法, 适用于显著性梯度明显的多组对比; Holland 修正值基于累积分布函数调整, 更适合分布控制较为严格的分析情境。本文同时引入两者, 增强了检验实验的可信度。QSIQ 方法在事后分析实验中的 Z 值最小, 并与统计检验值相近。同时, QSIQ 方法的 Holm 修正值和 Holland 修正值也最小, 同样优于其他 6 种对比算法。综上所述, 事后分析实验结果印证了 Mann-Whitney 检验统计分析的实验结果, 进一步证明了 QSIQ 方法的有效性

和先进性。与其他 6 种算法相比, 本文 QSIQ 方法最具统计学意义。

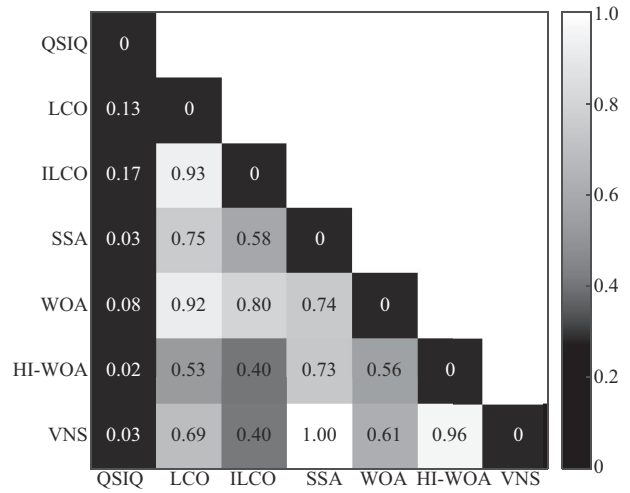


图 10 事后分析中的 p 值热力图

表 4 7 种算法事后分析中的 Z 值和 p 修正值对比情况

算法	Z 值	Holm 修正值	Holland 修正值
QSIQ	2.42	0.001	0.001
LCO	9.86	0.024	0.053
ILCO	9.53	0.051	0.052
SSA	18.65	0.011	0.016
WOA	15.54	0.015	0.018
HI-WOA	13.85	0.016	0.028
VNS	16.01	0.014	0.022

车联网雾计算系统是智慧交通系统的重要组成部分。首先, 车联网通过无线通信技术将车辆、基础设施、行人、雾节点和云服务器连接成一个网络, 实现了交通信息的实时共享。其次, 雾计算系统将计算能力、存储资源和网络资源分散到靠近数据源的雾层节点和设备, 显著提高了数据处理效率, 降低了任务处理时延和云服务器的负载。在智慧交通系统中, 车联网和雾计算系统的结合实现了车辆与基础设施之间的有效协作。云服务器和雾层的协同实现了快速访问信息、高效导航、实时交通信息更新和应急响应等功能。车联网雾计算系统不仅提升了交通系统的智慧水平, 还增强了行驶安全性、网络可持续性, 提高了城市发展经济效益。

传统的遗传算法作为一种基于自然选择和遗传学原理的优化算法, 广泛应用于复杂问题的求解, 但在处理高维问题时存在收敛速度慢、易陷入局部

最优解等缺陷。而本文 QSIQ 方法结合了量子遗传算法的特性,提高了收敛速度和搜索效率,并增加了种群多样性。在显著降低完工时间、能耗和算法开销的同时,能够提升雾节点的负载均衡度。在城市交通管理网络高峰时段任务调度的应用案例中, QSIQ 方法能够有效处理大量的并发任务,并能有效规避网络拥塞现象。具体而言, QSIQ 方法的完工时间较对比方法缩短了 69.0%,还在能耗、算法开销、负载均衡度、收敛性和统计检验等实验中表现出色。综上所述,本文 QSIQ 方法适用于高密度和动态变化的任务需求场景,并能均衡满足用户和网络运营商的个性化 QoS 需求,适宜于车联网云雾计算系统的任务调度领域。

5 结束语

在车联网云雾计算系统中,本文首先将用户和运营商的个性化 QoS 需求建模为多目标协同优化问题,提出了一种新型的 QSIQ 方法,并由雾代理服务器调度用户终端的网络任务请求。与多个知名群体智能方法进行对比实验,本文方法的完工时间、能耗、算法开销和负载均衡度等 QoS 指标表现出色。综上所述,本文方法能够更好地解决车联网云雾计算系统的任务调度问题。

本文方法虽然贴合实际场景,但实验的数据样本有限。未来计划将本文方法移植到更多的实际数据集中,并与 Q 学习等强化学习方法进行对比实验。还将继续改进本文方法,使之获得更好的应用效果和更高的推广价值。

参考文献:

- [1] RAJAPAKSHA S, KALUTARAGE H, AL-KADRI M O, et al. AI-based intrusion detection systems for in-vehicle networks: a survey[J]. *ACM Computing Surveys*, 2023, 55(11): 1-40.
- [2] 于刊,李东,张奇勋,等.车联网泛在感知、潜在通信、融合计算、内生安全综述:最新进展与未来方向[J]. *通信学报*, 2024, 45(11): 223-243.
- [3] 刘淼,林婉茹,王琴,等.车联网联邦学习的数据异质性问题及基于个性化的解决方法综述[J]. *通信学报*, 2024, 45(10): 207-224.
- [4] ZHANG R H, SUN C, VALIOLLAHIMEHRIZI R, et al. An uncertainty-aware, dual-tiered decision-making method for safe autonomous driving[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2025, 26(1): 691-702.
- [5] FENG Y Y, YU S Y, SHENG E C, et al. Distributed MPC of vehicle platoons considering longitudinal and lateral coupling[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 25(3): 2293-2310.
- [6] ZHENG K, MEI J, YANG H J, et al. Digital retina for IoV towards 6G: architecture, opportunities, and challenges[J]. *IEEE Network*, 2024, 38(2): 62-69.
- [7] XIE G C, XIONG Z H, ZHANG X Y, et al. GAI-IoV: bridging generative AI and vehicular networks for ubiquitous edge intelligence[J]. *IEEE Transactions on Wireless Communications*, 2024, 23(10): 12799-12814.
- [8] KHEZRI E, HASSANZADEH H, YAHYA R O, et al. Security challenges in Internet of vehicles (IoV) for ITS: a survey[J]. *Tsinghua Science and Technology*, 2025, 30(4): 1700-1723.
- [9] HAMDY W, KSOURI C, BULUT H, et al. Network slicing-based learning techniques for IoV in 5G and beyond networks[J]. *IEEE Communications Surveys & Tutorials*, 2024, 26(3): 1989-2047.
- [10] KUMAR N, SOOD S K, SAINI M. Internet of vehicles (IoV) based framework for electricity demand forecasting in V2G[J]. *Energy*, 2024, 297: 131199.
- [11] CHALABI N E, ATTIA A, ALNOWIBET K A, et al. A multi-objective Gaining-sharing knowledge-based optimization algorithm for solving engineering problems[J]. *Mathematics*, 2023, 11(14): 3092.
- [12] NGUYEN B M, NGUYEN T, VU Q H, et al. A novel nature-inspired algorithm for optimal task scheduling in fog-cloud blockchain system [J]. *IEEE Internet of Things Journal*, 2023, 11(2): 2043-2057.
- [13] SALIM A, KHEDR A M, OSAMY W. IoVSSA: efficient mobility-aware clustering algorithm in Internet of vehicles using sparrow search algorithm[J]. *IEEE Sensors Journal*, 2023, 23(4): 4239-4255.
- [14] ZHANG K P, LIU Y H, WANG X, et al. Enhancing IoT (Internet of Things) feature selection: a two-stage approach via an improved whale optimization algorithm[J]. *Expert Systems with Applications*, 2024, 256: 124936.
- [15] WANG L, REN Z, FAN S, et al. A novel hybrid information-based PF-WOA algorithm for gas source localization in 3D space[J]. *Robotica*, 2024, 42(11): 3748-3767.
- [16] XIAO Y G, LI Y H, LIU H, et al. Adaptive large neighborhood search algorithm with reinforcement search strategy for solving extended cooperative multi task assignment problem of UAVs[J]. *Information Sciences*, 2024, 679: 121068.
- [17] AZIZI S, SHOJAFAR M, ABAWAJY J, et al. Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: a semi-greedy approach[J]. *Journal of Network and Computer Applications*, 2022, 201: 103333.
- [18] BATISTA E, FIGUEIREDO G, PRAZERES C. Load balancing between fog and cloud in fog of things based platforms through software-defined networking[J]. *Journal of King Saud University - Computer and Information Sciences*, 2022, 34(9): 7111-7125.
- [19] KANBAR A B, FARAJ K. Region aware dynamic task scheduling and resource virtualization for load balancing in IoT-fog multi-cloud environment[J]. *Future Generation Computer Systems*, 2022, 137: 70-86.
- [20] MIN W, KHAKIMOV A, ATEYA A A, et al. Dynamic offloading in

flying fog computing: optimizing IoT network performance with mobile drones[J]. Drones, 2023, 7(10): 622.

- [21] BAEK J, KADDOUM G. FLoadNet: load balancing in fog networks with cooperative multiagent using actor - critic method[J]. IEEE Transactions on Network and Service Management, 2023, 20(1): 400-414.
- [22] IBRAHIM M A, ASKAR S. An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm[J]. IEEE Access, 2023, 11: 133607-133622.
- [23] ZHAO H, XU J H, LI P, et al. Energy minimization partial task offloading with joint dynamic voltage scaling and transmission power control in fog computing[J]. IEEE Internet of Things Journal, 2023, 11(6): 9740-9751.
- [24] KHALEEL M I. A dynamic weight - assignment load balancing approach for workflow scheduling in edge-cloud computing using ameliorated moth flame and rock hyrax optimization algorithms[J]. Future Generation Computer Systems, 2024, 155: 465-485.
- [25] ABDULAZEEZ D H, ASKAR S K. A novel offloading mechanism leveraging fuzzy logic and deep reinforcement learning to improve IoT application performance in a three-layer architecture within the fog-cloud environment[J]. IEEE Access, 2024, 12: 39936-39952.
- [26] SINGH S, SHAM E E, VIDYARTHI D P. Optimizing fog device deployment for maximal network connectivity and edge coverage using metaheuristic algorithm[J]. Future Generation Computer Systems, 2024, 157: 529-542.
- [27] KARAMI S, AZIZI S, AHMADIZAR F. A bi-objective workflow scheduling in virtualized fog-cloud computing using NSGA-II with semi-greedy initialization[J]. Applied Soft Computing, 2024, 151: 111142.
- [28] LI S Y, LIU H, LI W Z, et al. Non-convex optimization of resource allocation in fog computing using successive approximation[J]. Journal of Systems Science and Complexity, 2024, 37(2): 805-840.
- [29] OGUNDOYIN S O, KAMIL I A. Optimal fog node selection based on hybrid particle swarm optimization and firefly algorithm in dynamic fog computing services[J]. Engineering Applications of Artificial Intelligence, 2023, 121: 105998.
- [30] KOTSIU V, PAPADOPOULOS G Z, CHATZIMISIOS P, et al. LDSF: low-latency distributed scheduling function for industrial Internet of Things[J]. IEEE Internet of Things Journal, 2020, 7(9): 8688-8699.

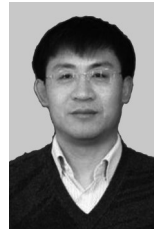
[作者简介]



张福琦 (1990-), 男, 吉林长春人, 吉林大学博士生, 主要研究方向为车载雾计算、强化学习、群体智能等。



姜会林 (1945-), 男, 辽宁沈阳人, 中国工程院院士, 长春理工大学教授, 主要研究方向为应用光学等。



刘富 (1968-), 男, 吉林长春人, 吉林大学教授, 主要研究方向为人工智能、模式识别等。



侯涛 (1980-), 女, 贵州贵阳人, 吉林大学教授, 主要研究方向为仿生弱信号感知芯片研发、精密检测设备研发等。



刘禹佳 (1991-), 女, 吉林长春人, 长春工程学院讲师, 主要研究方向为图像加密、FPGA设计、深度学习等。



关岳琦 (1994-), 男, 吉林长春人, 中国移动通信集团吉林有限公司无线网规划高级工程师, 主要研究方向为车联网规划、5G网络规划等。



沐星彤 (1995-), 女, 吉林长春人, 中国移动通信集团吉林有限公司客户响应工程师, 主要研究方向为政企业务技术解决方案支持等。