资源失配时低代价的数据与计算密集型服务重配

周长兵[1)] 李小翠[1),2),3)] 王煜炜[4)] 王亚沙[2),3),5)]

[1)](中国地质大学(北京)信息工程学院 北京 100083)

[2)](北京大学计算机学院 北京 100871)

[3)](北京大学(天津滨海)新一代信息技术研究院 天津 300380)

[4)](中国科学院计算技术研究所 北京 100190)

[5)](北京大学软件工程国家工程研究中心 北京 100871)

摘 要 随着边缘计算的广泛应用,近年来在网络边缘侧激增了一些延迟敏感的用户请求,这些应用对边缘网络中物联网设备提供的资源提出了较高的服务质量(Quality of Service,QoS)需求,例如严格的地理空间约束、时延/能量及其他资源约束.物联网设备提供的功能通常被封装为运行在边缘节点上的服务,用户请求可以通过组合数据和/或计算密集型的物联网服务来实现.考虑到物联网设备的资源稀缺性以及用户请求的在线持续部署和潜在长期执行特征,边缘服务运行期间对物联网设备资源的占用和释放导致边缘网络中资源动态变化.由于物联网设备的资源通常难以得到有效补充,且消耗差异可能较大,有些设备可能会过载,导致在当前时间点适配的物联网服务,在随后时间点可能难以适配用户请求,并导致QoS降级.针对边缘网络高负载时新请求持续部署导致特定强约束难以满足的挑战,本文开展资源失配时低代价的服务重配研究,提出了一种资源高效的服务重配方法,旨在通过服务迁移技术重调度物联网设备所提供的服务,以满足更多具有一定QoS约束的用户请求.基于上海电信基站数据集进行了大量实验,实例验证本文方法的有效性.实验结果表明,本文所提方法在满足用户服务请求时延约束、降低物联网设备能量消耗、提高边缘网络资源利用效益等方面表现均优于对比技术.

关键词 数据与计算密集型服务;资源利用效益;服务重配;服务迁移;边缘网络

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2024.02035

Data and Computation-Intensive Service Reconfiguration with Low Cost of Imbalanced Resources in Edge Networks

ZHOU Zhang-Bing[1)] LI Xiao-Cui[1),2),3)] WANG Yu-Wei[4)] WANG Ya-Sha[2),3),5)]

[1)](*School of Information Engineering, China University of Geosciences (Beijing), Beijing* 100083)

[2)](*School Computer Science, Peking University, Beijing* 100871)

[3)](*Peking University (Tianjin Binhai) New Generation Information Technology Research Institute, Tianjin* 300380)

[4)](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[5)](*National Research & Engineering Center of Software Engineering, Peking University, Beijing* 100871)

Abstract The advent of edge computing has revolutionized the landscape of Internet of Things (IoT) applications, enabling a plethora of services to operate at the network periphery. Among these are augmented reality, online interactive gaming, and real-time video processing, all of which are characterized by their sensitivity to latency. The quality of service (QoS) demands for these applications are stringent, necessitating precise control over geospatial positioning, re-

收稿日期:2023-06-08;在线发布日期:2024-06-11.本课题得到国家自然科学基金(42050103,62372420)资助.**周长兵**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为服务计算、边缘智能、物联网.E-mail:zbzhou@cugb.edu.cn.**李小翠**,博士,助理研究员,主要研究领域为服务计算、边缘计算、异常检测、大模型垂域泛化与高效复用.**王煜炜**,博士,高级工程师,硕士生导师,主要研究领域为边缘智能、联邦学习、无人系统网络协同、网络功能虚拟化.**王亚沙**,博士,教授,博士生导师,教育部长江学者,中国计算机学会(CCF)高级会员,主要研究领域为深度学习、智慧医疗、大模型垂域泛化与高效复用.

sponse times, energy efficiency, and other resource-intensive constraints. IoT devices, which are integral to the functioning of these applications, encapsulate a variety of functionalities through IoT services. The fulfillment of user requests often hinges on the seamless composition of these data and computation-intensive services. However, the resource limitations inherent to IoT devices pose a significant challenge. The dynamic allocation and deallocation of resources during the runtime of IoT services lead to fluctuations in the availability of these resources within edge networks. Given the difficulty in replenishing IoT device resources and the potential for substantial variability in their consumption, the risk of device overload is a genuine concern. This can result in a decline in the ability of IoT services to consistently meet user requests, thereby degrading the QoS for both current and future requests. This paper introduces a novel low-cost service reconfiguration strategy designed to mitigate the imbalance of resources in edge computing environments. By leveraging service migration technologies, we propose a resource-efficient reconfiguration method capable of accommodating a greater number of forthcoming user requests, all while adhering to specified QoS constraints. We formulate the service reconfiguration problem as markov multi-phases decisions, which are addressed by Double Q-network with the replay buffer to enhance Reinforcement Learning (denoted DQRL). This algorithm considers the delay of IoT applications and resource utilization of IoT devices comprehensively, to optimize service reconfiguration in edge networks. The proposed strategy is underpinned by an intelligent decision-making framework that optimizes the allocation of resources to IoT services, thereby enhancing the resilience and adaptability of edge networks to fluctuating demands. To validate the efficacy of our approach, we conducted extensive experiments using a dataset from Shanghai telecom base stations, which provided a realistic and complex environment for testing. The experimental results show that our approach performs better than baseline techniques in terms of satisfying delay constraints of IoT applications, decreasing energy consumption, and improving the resource utilization efficiency of IoT devices. This performance not only contributes to the sustainability of IoT ecosystems but also enhances the overall resource utilization efficiency of IoT devices.

Keywords data and computation-intensive service; resource utilization efficiency; service reconfiguration; service migration; edge networks

1 引 言

随着边缘计算和微服务体系架构的快速发展与广泛应用[1-3],物联网设备被部署在网络边缘,彼此互联协同并提供兼容的功能以实现用户服务请求.物联网设备的计算资源,包括功能及支撑这些功能所需的算力、内存和网络带宽等,被封装为物联网服务[4-6],通过组合功能可兼容和时空等约束可满足的物联网服务以实现用户请求[7].但在边缘计算中,用户请求任务的特性是多样的,有的任务需要处理大量的数据,有的则需要强大的计算能力[8-9].这种异构性对组合服务的适配和优化策略选取产生影响,尤其是考虑到数据和计算密集型服务之间的计算与通信所产生的延迟和能量.这些都是影响用户服务请求性能的关键因素.

通常,物联网设备中计算资源的状态会随着用户服务请求任务的占用和释放而动态变化.考虑到物联网设备的资源稀缺与占用不均衡等特性,当需要处理的任务过多、物联网设备上任务负载过重时,针对即将到来的用户请求所需的物联网服务,候选物联网设备可能难以有足够的剩余资源来托管某些服务的更多实例.随着新的特定强约束用户请求持续部署,物联网设备存在高负载隐患[10],导致处于运行态服务的(Quality of Service,QoS)可能受到负面影响,即在当前时间点适配的物联网服务,在随后时间点可能难以适配用户服务需求,导致 QoS 降级、用户体验变差及网络生命周期缩短等问题.考虑到物联网服务运行时物联网设备上资源的动态变化特征以及新请求的特定强约束及其预测不准确性,

边缘设备上所提供的服务与新/旧请求的服务部署需求可能存在资源失配,导致服务“错位”从而降低请求的可满足性和网络总体性能.为保障用户服务请求的 QoS,提高物联网设备的资源利用效率,开展服务柔性重配置研究,旨在均衡边缘网络设备运行时的资源负载,从而整体提升用户服务请求的 QoS,是提升边缘网络中用户服务请求的健壮性和吞吐量的重要保障.

针对边缘网络高负载时新请求持续部署导致特定强约束难以满足的研究挑战,服务重配作为一个行之有效的解决方案,已引起了广泛的关注.现有研究工作着重考虑静态规约边缘设备资源的分配模型、服务执行成本效益及网络负载均衡等因素,构建最优策略,实现特定约束下服务配置方案.总体来看,这些研究对于边缘网络下物联网设备运行态性能指标考虑不够全面,重配策略技术方案存在的研究难点与挑战之处总结如下:

(1)组合服务动态自适应重配考虑不足.

组合服务的动态自适应重配是边缘计算中的一个关键难题,要求算法不仅要高效利用有限资源以满足多样化的用户需求,还要能够适应即将到来的任务需求,以确保系统的响应能力和资源利用最优化.现有技术大多集中于原子服务的重配[11-12],难以适用于组合服务的重配场景.尤其是在处理服务请求任务间的依赖关系和跨服务的资源约束时.一些研究工作提出重新配置服务功能链[13-15],优化用户服务请求的 QoS.然而,他们仅考虑预部署的用户服务请求,依据任务间的依赖约束,静态分配网络资源.针对网络运行态资源动态变化,自适应服务重配不是他们的研究重点.物联网设备资源占用的增加在一定程度上可能会导致运行时部分请求的 QoS 下降,一些研究工作提出将正在运行的物联网服务重配到本地、边缘或云层[16-17],使网络能够容纳更多请求,从而提高网络的应用吞吐量.这些工作仅仅针对原子任务的物联网服务实现重配,忽略了边缘网络下用户请求任务是由多个原子任务组合而成,难以适用于边缘计算下组合服务的重新配置.因此,提出一种低代价的服务重配策略,既能保障正在运行的用户服务请求的指定约束,又能尽可能多地满足即将到来的用户服务请求,是一个重要的研究挑战.

(2)边缘资源动态性且利用率低.

物联网设备资源的动态变化特征,以及新请求的特定强约束及其预测不准确性,使得边缘设备上的服务与新/旧请求的服务部署需求可能出现资源失配.边缘资源的动态性要求算法能够灵活应对资源的实时变化,即实时监控资源状态和用户请求模式,实现资源预留和再分配策略,保障当前任务的 QoS,也为即将到来的任务提供了资源保障.现有研究工作大多聚焦优化分配物联网设备的可用资源,以满足当前用户服务请求[18-20].然而,这些工作难以保证即将到来的任务有足够的资源支撑其运行.事实上,基于严格的空间和时间约束,有些任务可能必须配置到某个可能没有足够剩余资源的物联网设备上.考虑到已经部署运行在物联网设备上的任务可能具有相对宽松的约束,这些任务也可以配置在邻近的具有充足剩余资源的设备上,同时他们的 QoS 仍然可以满足.这些释放的资源,可以支撑即将到达的用户请求.因此,资源失配时错位服务柔性重构策略有望支撑更多即将到来的用户服务请求,从而显著提高边缘网络的资源利用效率.如何构建错位服务柔性重配的代价最优策略,提升用户请求的健壮性并提高边缘网络的吞吐量,是另一项研究挑战.

为解决上述难点与挑战,本文提出一种资源高效的服务重配方法(rEsource-Efficient service reConfig-uration,E^2rC),旨在重调度边缘网络中物联网设备的资源,以满足更多具有一定 QoS 约束的用户请求.本文贡献总结如下:

(1)构建基于 QoS 感知的错位服务在线探测算法,检测潜在过载的物联网设备,并判别错位服务最小集,增强了资源变动网络中服务重构的灵活性,从而更好地满足用户请求的 QoS 约束.

(2)提出基于依赖约束的组合服务调度算法,以保证跨服务依赖指定约束满足下,将错位服务重配至具有足够剩余资源的临近物联网设备上,有助于避免服务之间的冲突或竞争,从而避免系统性能下降,并提高服务重配的效率.

(3)提出错位服务柔性重配的代价最优策略.构建服务重配代价模型,依据网络资源动态变化特征,将在线服务重构问题建模为马尔可夫多阶段决策过程,提出一种融合策略回放与双层网络相结合的提升强化学习算法(Double Q-network with the replay buffer to enhance Reinforcement Learning,DQRL).该算法持续地与边缘网络环境进行交互,实现收益最优的在线服务重配策略.

(4)将本章节所提的 E^2rC 算法与现有相关研究工作进行对比实验,在不同相关参数设置下评估

性能指标,实验结果表明 E^2rC 在满足用户服务请求的时延约束,降低物联网设备能耗,提高物联网设备的资源利用效率等方面均优于对比技术.

本文后续组织如下:第 2 节阐述与本研究相关的前沿工作进展;第 3 节详细介绍了数据与计算密集型服务请求的网络模型,为后续的问题描述和方案策略提供了基础;第 4 节基于第 3 节的网络模型,提出了一种低代价的数据与计算密集型服务重配方法,旨在解决核心的研究问题;第 5 节通过一系列实验,验证了所提算法的性能,并对结果进行了深入的分析和讨论;第 6 节总结了本文的主要研究成果和贡献,并对未来的研究方向提供了建议.

2 相关工作

边缘计算架构支持物联网设备在网络边缘侧托管数据与计算密集型服务[21],以服务组合的模式构建"边缘服务适配"架构[22],实现边缘网络上资源效能的最大化.设备资源具有泛在异构和动态变化的特性,具有相同功能的边缘服务在时间、空间、能量有效性等非功能属性方面可能存在显著差异性[23].此外,应用请求子任务(对应于服务)可能具有数据密集型、计算密集型或者两者皆有之的特性[24-25].为避免较大规模数据长距离网络传输及单设备上较大强度计算需求,导致传输带宽成本过高、应用响应时延过长、部分边缘设备资源消耗过大致使网络系统整体性能下降等问题[26],Sun 等人[27]构建能量感知的边缘服务适配方法,解决部分边缘设备资源消耗过大致使网络系统整体性能下降等问题,延长网络生命周期.针对 I/O 密集型任务特点,Coleman 等人[28]提出数据密集型服务适配算法,有效提升应用平均执行时间,在一定程度上维持网络负载均衡.Lin 等人[29]通过监控和分析 CPU 密集型应用的本质特征,构建 CPU 密集型应用的识别和分类模型,提高应用处理效率和资源利用率.考虑到用户请求存在不确定性,某些边缘服务可能未被合适的边缘设备托管,或其非功能属性难以满足请求的特定约束,为解决边缘网络下用户请求的某些服务尚未被指定地理区域内的任何设备所托管的问题,Li 等人[30]通过将数据与计算密集型应用请求规约为一个多目标多约束优化问题,提出基于 NSGA-Ⅱ的服务适配算法,为迁移驱动的服务协同适配提供支撑.针对运行时服务与设备资源匹配问题,Botangen 等人[31]提出动态替换服务组合中的某些服务,为边缘应用请求分配最合适的服务,保障服务适配管理的可靠性.

边缘计算架构下,用户服务请求被分配至各个边缘服务器[32-34],而不是集中部署在云端.考虑到物联网设备资源(如 CPU、内存、能源等)有限,如何有效地进行服务调度是一个重要挑战[35-38].相关研究尝试从不同角度解决该问题.Hao 等人[39]考虑物联网设备资源使用效率,提出一种启发式调度算法来分配用户请求任务,并将物联网设备划分为簇,确保同一簇内的物联网设备具有最小的通信距离,从而节省了物联网设备之间的通信开销.Tran-Dang 等人[11]提出一种边缘服务调度算法,优化雾节点的计算资源,以满足延迟敏感型用户服务请求.随着物联网设备在网络边缘的大规模部署,催生了大量需要及时分析以支持延迟敏感型应用的感知数据,Xu 等人[40]提出将数据封装为特定服务,在网络边缘侧进行调度,并提出一种面向数据流的服务调度算法以降低延迟敏感型应用的调度延迟.Kaur 等人[41]提出一种最小化数据传输延迟的调度算法,以满足用户请求的实时性需求.Gu 等人[42]将物联网设备的能量封装为资源,提出从具有剩余绿色能量的边缘节点向负载过重的边缘节点调度能量,使得这些节点能够持续执行用户请求.该方法旨在最小化调度能耗和延迟.考虑到用户请求可假设为依据某种分布随机到达,Yang 等人[43]提出一种高效的强化学习算法,以在满足延迟和可靠性约束的同时最大化请求的数量.综上所述,现有研究工作致力于解决服务调度性能问题(包括能耗、延迟),为本文探索数据与计算密集型服务重配优化方法提供了支撑.

服务重配旨在优化边缘网络中物联网设备计算资源的分配,以满足具有 QoS 约束的新用户服务请求.Li 等人[13]考虑了用户移动性对服务重配的影响,提出一种服务功能链重配算法,以降低用户请求的时延.然而,该工作未探索多个正在运行的用户请求的服务重配问题.Donassolo 等人[16]提出一种将正在运行的物联网服务重新部署到本地、边缘或云层的方法,以减少物联网设备的资源消耗.近期一些研究提出优化物联网设备剩余资源的分配方法,以满足当前物联网应用任务的需求.An 等人[18]基于李雅普诺夫优化算法,提出一种动态协调和分配物联网设备上计算资源的方法.然而,该研究难以保障即将到来任务能够得到有效的资源分配,尤其是当边缘网络(部分)过载时.为解决这个问题,Battula 等人[19]提出一种资源可用性评估模型,量化用户请

求的 QoS,为实现动态物联网服务重配提供支撑. Zhao 等人[44]提出一种基于累积鲁棒性度量来实现运行态服务定量监测. 该方法激发本文检测正在运行的多个用户请求任务之间依赖约束的满意度变化. 考虑到即将到来用户请求的不确定性,某些任务可能难以得到满足,候选物联网设备的计算资源可能被当前正在运行的请求所占用. 这些物联网设备可能没有足够剩余资源再支持即将到来的请求,而且严格的延迟约束可能会加剧这种困境. 因此,在支持即将到来用户请求方面,现有研究难以解决组合服务的重配问题. 综上,现有研究激发本文探索错位服务重配问题,考虑两个因素:(1)用户请求之间任务依赖性约束,以解决组合服务重配的联动问题,(2)物联网设备资源利用效率优化,以支持更多即将到来的用户请求.

基于容器的服务迁移方法为解决时空约束下特定区域内服务重配问题提供了支撑[45-47]. 当用户移动至某些尚未提供特定类型物联网设备的区域时,通过容器技术将该类型物联网服务部署至该区域内已有其他类型物联网设备上,以实现服务的无缝切换. Ma 等人[48]提出一种容器分层技术以减少由于冗余文件传输引起的系统同步延迟的服务迁移算法. 该研究为支持无缝服务迁移提供了技术基础,确保用户可以在不同网络区域内得到服务. Liu 等人[49]提出一种多智能体优化算法,用于解决移动用户请求的服务迁移问题. 算法生成一个近似最优分配策略,选择物联网设备托管待迁移服务,同时最小化服务迁移开销. 这些研究主要关注单个服务的迁移,针对多个服务的联合迁移探索较少. Li 等人[50]提出一种改进的混合遗传进化算法,最小化服务迁移延迟,并平衡网络负载,以解决服务功能链的迁移问题. Fu 等人[51]提出一种面向动态资源的微服务迁移算法. 当承载某些微服务的设备资源过载时,该算法将某些微服务迁移到具有足够资源的临近物联网设备上,以平衡网络负载并保证用户请求的 QoS 约束. 该研究促使本文考虑生成低代价服务重配策略时的设备资源利用效益. 综上,现有研究着重研究单个服务或具有依赖关系的多个服务的迁移,激励本文探索基于服务迁移的数据与计算密集型服务重配优化方法.

综上所述,现有研究着重关注动态集成边缘资源优化,以提升服务请求的响应时间及能耗等效能指标. 实际上,数据与计算密集型服务重配需要考虑的关键因素包括网络资源利用效益、服务之间的数据传输带宽成本等. 现有研究对于边缘设备运行态性能指标考虑不全面,因此,本文开展资源失配时错位服务柔性重配研究,包括面向资源失配时错位服务的效益评估模型,柔性重配策略的生成,以提升边缘网络的吞吐量,为面向边缘计算框架下行业应用落地提供理论基础和技术支撑.

3　网络模型

表 1 归纳了本文服务重配研究中所涉及的各种参数符号说明.

表 1　系统模型相关参数说明

参数名称	参数说明
ac_l	第 l 个用户服务请求
Ac^t	当前时隙下(t)边缘网络中正在运行的用户服务请求
Ac^t_m	因资源失配而需重配置的错位服务集合
N_l	第 l 个用户服务请求中子任务的数量
$s_{l,n}$	第 l 个用户服务请求中第 n 个子任所对应执行的服务
K	边缘网络中物联网设备的数量
v_k	第 k 个物联网设备
P^A_k	物联网设备 v_k 处于激活状态下的功率
P^I_k	物联网设备 v_k 处于空间状态下的功率
P^T_k	物联网设备 v_k 的传输功率
$dt_{n,n'}$	物联网服务 $s_{l,n}$ 和物联网服务 $s_{l,n'}$ 之间的数据传输数量
D	边缘网络中物联网设备的资源类型数量
c^d_k	物联网设备 v_k 的第 d 维可用的剩余资源容量 $d \in D$
$w^d_{l,n}$	物联网服务 $s_{l,n}$ 执行任务所需的第 d 维资源的占用量 $d \in D$
$x_{l,n}$	物联网服务 $s_{l,n}$ 的迁移策略
$m_{l,n}$	物联网服务 $s_{l,n}$ 迁移文件量大小
$\boldsymbol{X}$	算法为 Ac^t_m 所生成的迁移配置策略
$\zeta^k_{l,n}$	物联网服务 $s_{l,n}$ 是否配置在第 k 个物联网设备
$\zeta_l(t)$	第 l 个用户服务请求的服务配置策略
$\boldsymbol{\zeta}(t)$	Ac^t_m 中所有用户服务请求 ac_l 的重配策略
$T_l(\zeta_l(t))$	第 l 个用户服务请求的服务重配响应延迟开销
$E_l(\zeta_l(t))$	第 l 个用户服务请求的服务重配能量消耗开销
$Z_r(\boldsymbol{\zeta}(t))$	评估错位服务 Ac^t_m 重配策略 $\boldsymbol{\zeta}(t)$ 的网络资源利用效益
$Z_s(\boldsymbol{\zeta}(t))$	评估错位服务 Ac^t_m 重配策略 $\boldsymbol{\zeta}(t)$ 的服务重配运行效益

定义 1. 物联网设备. 物联网设备定义为一个多元组 $v=(id,spt,f,rstg,rbnd,eng,S,N_c^{Max})$,其中:$id$ 为 v 的唯一标识符;spt 表示 v 的地理位置及空间信息;f 为 v 的计算能力(即,每秒 CPU 周期);$rstg$ 和 $rbnd$ 分别表示物联网设备剩余存储容量和带宽;eng 表示 v 的剩余能量;S 表示配置在 v 上的一系列服务集合;N_c^{Max} 表示 v 可以同时实例化的容

器的最大数量.

物联网设备的功能被封装为物联网服务,这些服务以容器的形式独立部署.一个容器可以托管多个物联网服务,但在容器运行期间仅有一个物联网服务可被激活,设备资源被占用以支持该容器运行[52].这些设备在计算、存储、通信和剩余能量等方面受到限制.这表明,为了保证所有服务可以顺利地执行任务,在同一个时间段内只能激活少量的容器,允许同时实例化的容器的数量应该受到限制,不能超过该设备所能承载的最大容器数.

物联网服务具有特定的名称和功能描述文本,这两类属性以短文本的形式进行表述.物联网设备与物联网服务间遵循多对多关系,即某种类型的物联网服务可在一个或多个物联网设备上部署;同样,一个物联网设备也可承载一种或多种类型的物联网服务.物联网服务的定义如下:

定义 2.物联网服务.物联网服务定义为一个多元组 $s=(id,nm,dsc,g,w,cr,stg,bnd,D)$,其中:$id$ 为 s 的唯一标识符;nm 为 s 的名称;dsc 为 s 的功能短文本描述;g 用来标识表示 s 的类型,即 $s.g=0$ 表明此物联网服务是数据密集型,反之 $s.g=1$ 表示为计算密集型服务;w 表示 s 的负载,即此物联网服务迁移过程中传输的文件量;cr 表示运行 s 所需物联网设备分配的 CPU 计算量;stg 和 bnd 分别表示物联网服务当前占用的存储容量和带宽;D 表示配置了 s 的物联网设备集合.

物联网服务可以是(1)数据密集型服务(data-intensive service,ds),主要负责处理一定规模数据量的输入和输出操作(例如视频流数据提取任务),或者是(2)计算密集型服务(computation-intensive service,cs),主要以占用 CPU 资源的计算量执行(例如图像处理任务).物联网服务依据容器可移植性,在多个物联网设备间按需迁移,以实现网络中资源的调度优化并满足用户请求.

定义 3.用户服务请求.用户服务请求可表示为一个有向无环图(Directed Acyclic Graph,DAG),定义为一个三元组 $ac=(Tsk,CnT,CsT)$,其中:Tsk 表示 ac 中包含的任务集合,即,$ac.Tsk=\{s_1,s_2,\cdots,s_i,s_{|Tsk|}\mid 0<i\leqslant|Tsk|\}$,$s_i$ 由一组功能相似的物联网服务执行,用于执行 s_i 的服务集合表示为服务类;CnT 表示 ac 的任务间控制调用执行关系;CsT 表示 ac 指定的 QoS 约束关系(如时空、时延约束、能耗约束等).

用户请求包含一系列服务类,通过地理邻近的物联网设备间相互协作,按照用户指定的服务约束关系,依次调用物联网设备上部署的物联网服务并进行组合,以满足用户服务需求.边缘计算下,用户请求的服务组合定义如下:

定义 4.数据与计算密集型服务组合.数据与计算密集型服务组合可定义为一个三元组 $cp=(S,E,D)$,其中:S 表示 cp 中包含的服务集合;E 表示 cp 中服务间调用执行关系;D 表示网络中托管 $cp.S$ 的物联网设备集合.

3.1 边缘网络用户服务请求重配模型

边缘网络表示为一个网络图 $G=<V,M>$,其中 V 表示边缘网络中所有物联网设备的集合,$V=\{v_0,v_1,v_2,\cdots,v_k,\cdots,v_{K-1}\}$.$M$ 表示边缘网络中所有物联网设备间通信链接集合,表示为 $\{<v_i,v_j>\mid i\neq j\}$.每个物联网设备 v_k 被表示为一个 D 维向量,$v_k.C=(c_k^1,c_k^2,\cdots,c_k^D)$,其中每个维度 c_k^d 代表一种资源容量大小,如该设备的 CPU、内存、能量等.边缘网络中,所有用户都可利用物联网设备接入网络并发起特定约束的服务请求.在某一时刻(t)边缘网络中所有用户的请求集合表示为 $Ac^t=\{ac_0,ac_1,ac_2,\cdots,ac_l,\cdots,ac_{L-1}\}$,其中 $0\leqslant l\leqslant L$,$L=|Ac^t|$.某个用户服务请求 ac_l,在定义 3 用户服务请求表示基础上,将服务请求解耦成为具有一定关联依赖关系的服务集合,表示为 $ac_l=\{s_{l,0},s_{l,1},s_{l,2},\cdots,s_{l,n},\cdots,s_{N_{L-1}}\}$,其中 N_L 表示第 l 个请求 ac_l 所包含的子任务个数.物联网设备需要为服务请求 ac_l 中的每个物联网服务需要分配足够的计算和存储资源,则运行在物联网设备 v_k 上所需占用的各类资源容量可表示为 $s_{l,n}.W=(w_{l,n}^1,w_{l,n}^2,\cdots,w_{l,n}^D)$,其中 $W_{l,n}^d$ 表示物联网服务 $s_{l,n}$ 所需的第 d 类资源量.本文将物联网设备资源过载而引发的资源失配时需要进行重配的服务,称之为“错位”服务.网络中在线探测识别得到的错位服务集表示为 Ac_m^t,在此集合中的用户服务请求是需要被重配到其他具有足够剩余资源的物联网设备,以此既能满足正在运行的用户服务请求的指定约束,又能尽可能多地实现即将到来的用户服务请求.

根据边缘网络资源情况,为 Ac_m^t 动态自适应调整所托管的物联网设备,其柔性重配策略表示为 $\boldsymbol{\zeta}(t)=\{\boldsymbol{\zeta}_l(t)\mid ac_l\in Ac_m^t,0\leqslant l\leqslant L_m,L_m=|Ac_m^t|\}$,其中 $\boldsymbol{\zeta}_l(t)$ 表示为第 l 个用户服务请求 ac_l 生成的在线重配置策略,可表示为一个 $N\times K$ 矩阵,如公式(1)所示:

$$\boldsymbol{\zeta}_l(t)=\begin{bmatrix}\zeta_{l,0}^{0} & \cdots & \zeta_{l,0}^{K-1}\\ \vdots & \ddots & \vdots\\ \zeta_{l,N_l-1}^{0} & \cdots & \zeta_{l,N_l-1}^{K-1}\end{bmatrix} \tag{1}$$

其中,$\zeta_{l,n}^{k}\in\{0,1\}$ 表示针对第 l 个用户服务请求 ac_l 中的第 n 个物联网服务 $s_{l,n}$ 是否配置在物联网设备 v_k . 当 $\zeta_{l,n}^{k}=1$ 表示物联网服务 $s_{l,n}$ 配置在第 k 个物联网设 v_k. $\zeta_{l,n}^{k}=0$ 表示物联网服务 $s_{l,n}$ 未配置在物联网设备 v_k . 在边缘网络中,所属用户服务请求中的任务在运行时,仅仅只能分配到一个物联网设备上执行,不能由多个物联网设备联合执行一个原子任务,即物联网设备与运行任务是一对多的所属关系,被规约为公式(2)所示:

$$\sum_{k=0}^{K-1}\sum_{l=0}^{L_m-1}\sum_{n=0}^{N_l-1}\zeta_{l,n}^{k}=\sum_{l=0}^{L_m-1}N_l \tag{2}$$

3.2　数据与计算密集型服务重配响应延迟性能模型

基于运行时物联网服务对托管该服务的设备上特定资源的具体需求,分析边缘设备运行时的资源动态性,将在线计算和网络资源的运行时效能评估引入到网络资源调整与服务柔性重配过程中,生成最优的服务重配策略,使得边缘网络性能(网络的整体负载、服务响应延迟、设备能量消耗)达到最优,提高边缘网络对用户请求的吞吐量. 错位服务 Ac_m^t 重配开销主要包括服务迁移、计算和通信等代价[53],本节提出 E²rC 算法所涵盖的以资源规约为导向的服务重配运行性能模型,包括服务响应延迟模型、能量消耗模型和资源利用效率模型.

采用二进制变量 $x_{l,n}\in\{0,1\}$ 表示物联网服务 $s_{l,n}$ 重配策略. $x_{l,n}=0$ 表示服务 $s_{l,n}$ 本地设备执行,不进行服务迁移重配. $x_{l,n}=1$ 表示服务 $s_{l,n}$ 从本地设备迁出,重配在其他资源充足的临近设备上.

(1)服务重配计算延迟度量

对于某个物联网服务 $s_{l,n}$ 的重配策略 $\zeta_{l,n}^{k}$,其计算延迟开销主要包括(1) $x_{l,n}=0$:服务 $s_{l,n}$ 由本地物联网设备 v_i 执行的计算延迟;(2) $x_{l,n}=1$:服务 $s_{l,n}$ 由物联网设备 v_i 迁移至邻近物联网设备 v_j 的迁移传输延迟和由 v_j 执行的计算延迟. 服务重配计算延迟度量如公式(3)所示:

$$T_e(s_{l,n})=\begin{cases}\dfrac{w_{l,n}^{1}}{c_i^{1}}, & (x_{l,n}=0);\\ \dfrac{m_{l,n}}{r_{i,j}}+\dfrac{w_{l,n}^{1}}{c_j^{1}}, & (x_{l,n}=1).\end{cases} \tag{3}$$

物联网设备 $\boldsymbol{v}_i$ 资源是一个 D 维向量 $\boldsymbol{v}_i$. $C=(c_i^{1},c_i^{2},\cdots,c_i^{D})$,每个维度 c_i^{d} 代表一种资源容量大小,如该设备的 CPU 和内存、存储、带宽、能量等. 在公式(3)中 $d=1$, c_i^{1} 或者 c_j^{1} 表示设备的 CPU 处理计算能力. $w_{l,n}^{1}$ 表示服务 $s_{l,n}$ 所占用物联网设备 v_i 或者 v_j 的 CPU 资源量. $m_{l,n}$ 表示重配服务 $s_{l,n}$ 时发生所需迁移的文件量. $r_{i,j}$ 表示在物联网设备 v_i 和 v_j 间传输速率,计算方式如下所示:

$$r_{ij}=B\times\log_2\left(1+\frac{P_i^{T}g_{ij}}{\theta^{2}}\right) \tag{4}$$

其中, B 表示物联网设备 v_i 和 v_j 间通信的信道带宽; P_i^{T} 表示发送方物联网设备 d_i 的传输功率; θ^{2} 表示为信道噪声功率; g_{ij} 表示物联网设备 d_i 和 d_j 间的信道增益,表示为 $g_{ij}=d_{ij}^{-\alpha}$,其中 d_{ij} 表示为物联网设备 d_i 和 d_j 间通信距离, α 表示通信耗损因子,通常设置为常数 4.

因此,第 l 个用户服务请求 ac_l 生成的在线重配策略 $\boldsymbol{\zeta}_l(t)$ 代价之计算延迟可度量为公式(5):

$$T_e(\boldsymbol{\zeta}_l(t))=\sum\nolimits_{(s_{l,n}\in ac_l)}T_e(s_{l,n}) \tag{5}$$

(2)服务重配通信延迟度量

对于一个用户服务请求 ac_l ,当具有数据先后执行顺序依赖约束的两个物联网服务 $s_{l,n}$ 和 $s_{l,n'}$ 配置运行在不同的物联网设备 v_i 上和 v_j ,存在通信延迟. 令 $\beta_{l,n,n'}^{i,j}\in\{0,1\}$ 表示 $s_{l,n}$ 和 $s_{l,n'}$ 间是否存在数据传输通信,表征为公式(6):

$$\beta_{l,n,n'}^{i,j}=\begin{cases}1, (P_{l,n'}\neq 0\cap s_{l,n}\in P_{l,n'});\\ 0, \text{otherwise}.\end{cases} \tag{6}$$

其中, $P_{l,n'}$ 表示物联网服务 $s_{l,n'}\in ac_l$ 的前序服务的集合. $P_{l,n'}\neq 0\cap s_{l,n}\in P_{l,n'}$ 表示物联网服务 $s_{l,n}$ 和 $s_{l,n'}$ 之间存在通信延迟.

因此,物联网服务 $s_{l,n}$ 和 $s_{l,n'}$ 之间通信延迟计算如公式(7)所示:

$$T_c(s_{l,n},s_{l,n'})=\beta_{l,n,n'}^{i,j}\times\frac{dt_{n,n'}}{r_{i,j}} \tag{7}$$

总的来说,通过将一些相邻的子任务对应的物联网服务迁移到同一物联网设备上,可以优化其通信延迟成本. 第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\boldsymbol{\zeta}_l(t)$ 代价之通信延迟可度量为公式(8):

$$T_c(\boldsymbol{\zeta}_l(t))=\sum\nolimits_{(s_{l,n},s_{l,n'}\in ac_l)}T_c(s_{l,n},s_{l,n'}) \tag{8}$$

综上公式(5)至公式(8),边缘网络中关于第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\boldsymbol{\zeta}_l(t)$ 的响应延迟性能度量评估如公式(9)所示:

$$T_l(\boldsymbol{\zeta}_l(t))=T_e(\boldsymbol{\zeta}_l(t))+T_c(\boldsymbol{\zeta}_l(t)) \tag{9}$$

3.3　数据与计算密集型服务重配能量消耗性能模型

(1)服务重配计算能耗度量

对于某个物联网服务 $s_{l,n}$ 的重配策略 $\zeta_{l,n}^{k}$,其

计算能耗开销主要包括(1) $x_{l,n}=0$:物联网服务 $s_{l,n}$ 由本地物联网设备 v_i 执行的计算能耗;(2) $x_{l,n}=1$:物联网服务 $s_{l,n}$ 由物联网设备 v_i 迁移至物联网设备 v_j 的迁移传输延迟和由 v_j 执行的计算能耗.如公式(10)所示:

$$E_e(s_{l,n})=\begin{cases}P_i^A\dfrac{w_{l,n}^1}{c_i^1}, & x_{l,n}=0;\\ (P_i^I+P_i^T)\times\dfrac{m_{l,n}}{r_{i,j}}+P_j^A\times\dfrac{w_{l,n}^1}{c_j^1}, & x_{l,n}=1.\end{cases}\tag{10}$$

其中,c_i^1、c_i^1、$w_{l,n}^1$、$m_{l,n}$、$r_{i,j}$、P_i^T 表示含义如公式(3)和(4)所述.P_i^I 和 P_i^A 分别表示物联网设备 v_i 在空闲状态、激活状态下的计算功率,P_j^A 表示物联网设备 v_j 在激活状态下的计算功率.

因此,v_i 用于执行第 l 个用户服务请求 ac_l 的任务的计算能耗,其度量方式如公式(11)所示:

$$E_e(v_i)=\sum\nolimits_{(s_{l,n}\in ac_l)}E_e(s_{l,n})\tag{11}$$

(2)服务重配通信能耗度量

对于一个用户服务请求 ac_l,当具有数据先后执行顺序依赖的两个物联网服务 $s_{l,n}$、$s_{l,n'}$ 配置运行在不同的物联网设备 v_i、v_j 上时,存在通信能耗.参照公式(6)和公式(7),物联网服务 $s_{l,n}$ 和 $s_{l,n'}$ 之间通信能耗计算如公式(12)所示:

$$E_c(s_{l,n},s_{l,n'})=P_i^T\times T_c(s_{l,n},s_{l,n'})\tag{12}$$

v_i 用于执行第 l 个用户服务请求 ac_l 的任务所发生的通信能耗,其度量方式如公式(13)所示:

$$E_c(v_i)=\sum\nolimits_{(s_{l,n},s_{l,n'}\in ac_l)}E_c(s_{l,n},s_{l,n'})\tag{13}$$

综上公式(11)至公式(13),边缘网络中关于第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\zeta_l(t)$ 的物联网设备的能量消耗性能度量评估如公式(14)所示:

$$E_l(\zeta_l(t))=\sum\nolimits_{i=0}^{I_l-1}(E_e(v_i)+E_c(v_i))\tag{14}$$

其中,$I_l(I_l\leqslant K)$表示用于执行第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\zeta_l(t)$ 所涉及的物联网设备的数量.

4 数据与计算密集型服务重配

本文问题域和约束条件是面向资源受限的边缘计算环境的.资源有限性、在线用户请求激增以及响应及时性等因素,是本文在设计组合服务重配策略时必须考虑的重要约束.本节提出了一种资源高效的服务在线柔性重配算法(rEsource-Efficient service reConfiguration,E^2rC),聚焦组合服务和满足即将到来的任务时延要求,重配边缘网络中物联网设备提供的资源,以保证正在运行的用户请求,同时尽可能满足即将到来的请求.本节分为五个部分,综合考虑网络资源负载特性,在线探测错位服务并判定其最小集;建立错位服务的依赖约束调度模型;构建低代价的服务在线重配策略;构建基于系统收益的服务在线重配策略评估算法,以此提升网络资源利用效益和服务重配运行效益.

4.1 问题模型构建

与传统模型相比,本文所提边缘网络用户服务请求重配模型,具有以下创新点:

考虑物联网设备资源的动态性:我们的模型考虑了物联网设备资源的动态变化,这是传统模型中常被忽略的一个重要因素.如公式(15)所示.3.1 重配模型为问题提供了一个明确的数学描述.

定量描述服务的运行效益敏感性:针对边缘计算的特性,我们的模型能够更加精确地描述服务的延迟及能耗需求.如公式(18)所示.3.2 和 3.3 性能模型为问题提供了一个明确的数学描述.

(1)网络资源利用效益

评估错位服务 Ac_m^t 重配策略 $\zeta(t)$ 的网络资源利用效益 $Z_r(\zeta(t))$ 的计算公式如下所示:

$$Z_r(\zeta(t))=\frac{1}{K_s}\sum\nolimits_{k=0}^{K_s-1}(cbR(v_k))\tag{15}$$

其中,$K_s(K_s\leqslant K)$表示边缘网络中用于执行 Ac_m^t 的物联网设备数量.$Z_r(\zeta(t))$ 用来评估物联网设备的资源利用效益.该值越大,表示边缘网络中物联网设备的资源越均衡.在本文中,错位服务 Ac_m^t 重配策略 $\zeta(t)$ 具有一定的可扩展性,当且仅当每个物联网设备 v_k 都有足够的剩余资源.$cbR(v_k)$ 表示物联网设备 v_k 的资源利用效率,计算公式如下:

$$cbR(v_k)=\frac{rm(v_k.C)-cd(v_k.C)}{rm(v_k.C)}\tag{16}$$

其中,$rm(v_k.C)$ 表示物联网设备 v_k 的剩余资源容量;$cd(v_k.C)$ 表示 v_k 用于支持物联网服务运行所消耗的资源容量.v_k 可选为承载配置物联网服务,其 $cbR(v_k)$ 应该是所有候选物联网设备中最大的.

错位服务在线柔性重配配置策略要满足物联网设备资源约束,避免设备过度负载,即对于物联网设备 v_k,满足约束如公式(17)所示.

$$\sum_{l=0}^{L_m-1}\sum_{n=0}^{N_l-1}\zeta_{l,n}^{k}cd(s_{l,n}.W)<rm(v_k.C) \tag{17}$$

其中，$cd(s_{l,n}.W)$ 表示物联网服务 $s_{l,n}$ 运行所消耗的资源容量.由公式(2)可知,配置给 v_k 的物联网服务所占用的计算资源总量不能超过 v_k 的剩余资源.

(2)服务重配运行效益

服务重配运行效益是指用户请求重配策略 $\boldsymbol{\zeta}(t)$ 使得用户服务请求 $ac_l\in Ac_m^t$ 具有较低的服务响应延迟和能量消耗.评估错位服务 Ac_m^t 重配策略 $\boldsymbol{\zeta}(t)$ 的服务重配运行效益 $Z_s(\boldsymbol{\zeta}(t))$ 如公式(18)所示：

$$Z_s(\boldsymbol{\zeta}(t))=\sum_{l=0}^{L_m-1}\omega_l\times U_s(\zeta_l(t)) \tag{18}$$

约束条件如公式(19)所示：

$$T_l(\boldsymbol{\zeta}_l(t))\leqslant T_l^{Max} \tag{19}$$

其中，ω_l 表示位于 Ac_m^t 中对应的用户服务请求 ac_l 的权重,计算参见公式(20)，$U_s(\boldsymbol{\zeta}_l(t))$ 表示针对 ac_l 所生成的服务重配策略 $\boldsymbol{\zeta}_l(t)$ 的效益,计算参见公式(21).公式(19)约束了服务重配后的响应延迟不能超过用户所能允许的最大延迟. $T_l(\boldsymbol{\zeta}_l(t))$ 计算参见公式(9)；T_l^{Max} 表示由用户服务请求 ac_l 指定的最大响应延迟.

$$\omega_l=\begin{cases}\dfrac{T_{L_m}^{Max}-T_l^{Max}}{T_{L_m}^{Max}-T_{L_m}^{Min}}, & T_{L_m}^{Max}-T_{L_m}^{Min}\neq 0;\\ 1, & T_{L_m}^{Max}-T_{L_m}^{Min}=0.\end{cases} \tag{20}$$

其中，$T_{L_m}^{Max}$ 和 $T_{L_m}^{Min}$ 分别表示在错位服务集合 Ac_m^t 中,用户服务请求指定响应延迟 T_l^{Max} 的最大值与最小值.

结合公式(9)和公式(14),针对用户服务请求 ac_l 所生成的服务重配策略 $\boldsymbol{\zeta}_l(t)$ 的效益 $U_s(\boldsymbol{\zeta}_l(t))$ 计算度量如公式(21)所示：

$$U_s(\boldsymbol{\zeta}_l(t))=\beta_t N(T_l)+\beta_e N(E_l) \tag{21}$$

其中，$0\leqslant\beta_t,\beta_e\leqslant 1$ 分别表示服务重配响应延迟 $T_l(\boldsymbol{\zeta}_l(t))$ 和能量消耗 $E_l(\boldsymbol{\zeta}_l(t))$ 的权重数值.本文针对延迟和能耗进行归一化处理,使其具有相同量纲，$N(T_l)$ 和 $N(T_l)$ 分别表示归一化的延迟和能耗,其计算方式如公式(22)和公式(23)所示：

$$N(T_l)=\begin{cases}\dfrac{T^{Max}(\boldsymbol{\zeta}(t))-T_l(\zeta_l(t))}{T^{Max}(\boldsymbol{\zeta}(t))-T^{Min}(\boldsymbol{\zeta}(t))}, & T^{Max}(\boldsymbol{\zeta}(t))-T^{Min}(\boldsymbol{\zeta}(t))\neq 0;\\ 1, & T^{Max}(\boldsymbol{\zeta}(t))-T^{Min}(\boldsymbol{\zeta}(t))=0.\end{cases} \tag{22}$$

其中，$T^{Max}(\boldsymbol{\zeta}(t))$ 和 $T^{Min}(\boldsymbol{\zeta}(t))$ 分别表示 Ac_m^t 中，所有用户服务请求的服务重配响应延迟的最大值和最小值. $T_l(\boldsymbol{\zeta}_l(t))$ 表示第 l 个用户服务请求 ac_l 服务重配策略 $\boldsymbol{\zeta}_l(t)$ 所产生的响应延迟.

$$N(E_l)=\begin{cases}\dfrac{E^{Max}(\boldsymbol{\zeta}(t))-E_l(\zeta_l(t))}{E^{Max}(\boldsymbol{\zeta}(t))-E^{Min}(\boldsymbol{\zeta}(t))}, & E^{Max}(\boldsymbol{\zeta}(t))-E^{Min}(\boldsymbol{\zeta}(t))\neq 0;\\ 1, & E^{Max}(\boldsymbol{\zeta}(t))-E^{Min}(\boldsymbol{\zeta}(t))=0.\end{cases} \tag{23}$$

其中，$E^{Max}(\boldsymbol{\zeta}(t))$ 和 $E^{Min}(\boldsymbol{\zeta}(t))$ 分别表示 Ac_m^t 中，所有用户服务请求的服务重配能量消耗的最大值和最小值. $E_l(\boldsymbol{\zeta}_l(t))$ 表示第 l 个用户服务请求 ac_l 的服务重配策略 $\boldsymbol{\zeta}_l(t)$ 所产生的能量消耗.

(3)服务重配问题定义

随着网络中用户请求的服务适配组合在线持续部署并长时执行,物联网设备的资源耗损可能存在较大差异,并可能存在过载等隐患,导致在当前时间点适配的物联网服务,在后续时间点可能难以适配用户服务需求,导致服务质量降级、用户体验变差及网络生命周期缩短等问题.因此,本文开展资源失配时错位服务重配研究,保障组合服务的健壮性和用户体验,提升边缘网络的吞吐量.针对以上两个研究目标,将服务在线柔性配置优化转化为一个多目标多约束问题,构建最大化服务重配运行效益 $Z_s(\boldsymbol{\zeta}(t))$ 和最大化网络资源利用效益 $Z_r(\boldsymbol{\zeta}(t))$ 这两个子目标,如下所示：

$$OP1:\mathrm{Max}Z_s(\boldsymbol{\zeta}(t))=\mathrm{Max}(\sum_{l=0}^{L_m}\omega_l\times U_s(\boldsymbol{\zeta}_l(t))) \tag{24}$$

$$OP2:\mathrm{Max}Z_r(\boldsymbol{\zeta}(t))=\mathrm{Max}(\frac{1}{K_s}\sum_{k=0}^{K_s-1}(cbR(v_k))) \tag{25}$$

需要满足的多约束条件如下：$s.t.$ 公式(2)、(17)、(19)等,

$$Dst(v_i,v_j)\leqslant Cov(v_i),\ \forall i,j\in\{0,\cdots,K-1\} \tag{26}$$

公式(26)约束了在执行物联网服务重配时,可供迁移选择的候选物联网设备所在的空间位置应该在源物联网设备的通信空间范围内,这些候选设备表示为 $Cnd_{ngh}^{i}=\{v_j\mid Dst(v_i,v_j)\leqslant Cov(v_i)\}$，其中 $Dst(v_i,v_j)$ 表示物联网设备 v_i 和 v_j 之间的空间欧氏距离，$Cov(v_i)$ 表示物联网设备 v_i 的覆盖半径.公式(2)约束了配置执行关系,即所属用户服务请求中的任务只能由一个物联网设备执行；公式(17)约束了配置在物联网设备上的服务所占用的资

源容量不能超过该设备的剩余资源容量;公式(19)约束了服务重配后的响应延迟不能超过用户所能允许的最大延迟.

边缘网络中的用户服务请求可分解为一系列具有依赖关系的服务集合,将这些服务重配至满足资源约束的物联网设备.算法需要满足(1)实时与边缘网络环境交互,获取资源动态变化;(2)实现多阶段自适应在线决策,以保证满足多个用户服务请求约束调配.因此,本文基于马尔可夫决策的在线强化学习算法,提出一种融合策略回放与双层网络相结合提升强化学习算法(DQRL),设置系统状态、行为及回报收益函数,生成低代价的错位服务在线柔性重配策略.基于公式(15)和公式(18),低代价的回报收益优化目标函数构建如下:

$$R(\boldsymbol{\zeta}(t))=\rho_s Z_s(\boldsymbol{\zeta}(t))+\rho_r Z_r(\boldsymbol{\zeta}(t)) \quad (27)$$

其中,$0\leqslant\rho_s,\rho_r\leqslant 1$ 分别表示服务重配运行效益和网络资源利用效益的权重数值,是基于系统性能优化目标确定的. $Z_s(\boldsymbol{\zeta}(t))$ 和 $Z_r(\boldsymbol{\zeta}(t))$ 分别表示服务重配策略的服务运行效益和网络资源利用效益.

因此,原问题 $OP1$ 和 $OP2$ 可转化为针对服务重配策略 $\boldsymbol{\zeta}(t)$ 的低代价优化问题,如公式(28)所示:

$$\mathrm{Max}R(\boldsymbol{\zeta}(t)) \quad (28)$$

本文引入强化学习,设计并提出基于 DQRL 的错位服务在线柔性重配算法(E2rC)算法,旨在高效解决上述服务重配问题.算法 3 是本文的主体算法,而算法 1、2、4 作为算法 3 数据处理调用部分,算法之间互相递进支撑,其之间的数据依赖及逻辑联系:算法 1 作为算法 2 的输入,主要负责对网络设备资源进行处理和约束;而算法 2 则作为算法 3 的输入,主要处理用户请求执行相关的约束条件.这两个算法共同为算法 3 提供必要的支持和保障,确保其生成的配置策略能够严格遵守各项规则约束.同时,算法 4 负责计算算法 3 中强化学习生成的重配策略的收益函数(reward),更准确地衡量算法 3 的性能,从而确保配置策略在优化过程中能够取得预期的效果.

4.2 基于 QoS 感知的错位服务在线探测算法

算法 1 构建了基于 QoS 感知的错位服务在线探测算法,检测过载物联网设备,识别服务重配最小集.首先根据用户请求的 QoS 约束,为新到达任务分配资源,得到候选服务组合执行可行解集(第 1 行).依据服务组合完成时间升序排列(第 2 行),从候选可行解集中依次检查可行解集,是否存在不迁移任何现有运行服务情况下,实现新任务加载(第 3~10 行).用户请求将通过第一个候选解集调度进入网络,检查设备的资源约束是否能够满足.若违反其约束,则标志物联网设备过载.算法 1 将依次检查其他候选解集,执行同样的约束检查.在不违反公式(17)约束的前提下,若用户任务请求可以由服务组合候选集中的某一个可行解调度执行,则算法结束(第 11~13 行).否则,若在不迁移任何现有服务的情况下,没有候选解集可以实现新任务加载执行,则选择迁移网络中某些物联网服务,以此既能满足正在运行的用户服务请求的指定约束,又能尽可能多地实现即将到来的用户服务请求,实现网络资源动态优化调配(第 14~18 行).最后,算法 1 在线探测识别得到错位服务集 Ac_m^t ,此集合中的用户服务请求是需要被重新配置到其他具有足够剩余资源的物联网设备上(第 19 行).

算法 1(1)确保所有即将到达的服务请求都会被考虑;(2)对于每一个即将到达的服务请求,算法通过检查是否可以满足约束条件来判断是否需要重配,这保证了服务的 QoS 要求不会因为资源失配而受到影响;(3)通过标记和检查的方式,算法确保了只有违反约束的服务才会被重配,避免了不必要的服务重配.算法 1 的时间复杂度是 $O(m\times mxp)$,其中 m 是候选服务组合的数量,p 是单个服务组合中的服务数量.

算法 1. 基于 QoS 感知的错位服务在线探测算法

输入:

当前时刻(t)运行的用户服务请求 Ac^t ;

即将($t+1$)到达的用户服务请求 Ac_n^{t+1} .

输出:

因资源失配而需重配的错位服务集合 Ac_m^t .

```
1.  CnS ← 系统为 Ac_n^{t+1} 生成的一系列满足其指定约束的
          候选服务组合集
2.  DSC ← sort ( CnS )
3.  for ∀ dsc_i ∈ DSC do
4.      for  ∀ v_k ∈ dsc_i do
5.          if  公式(17)约束被违反 then
6.              v_k.flag ← 1
7.              break
8.          end if
9.      end for
10. end for
11. if  v_k.flag == 0 then
12.     return Φ
```

13. else
14. for $\forall ac_l^{t+1} \in Ac_n^{t+1}$ do
15. if ac_l^{t+1} 可以被调度通过重配当前运行的用户服务请求 $ac_l \in Ac^t$ then
16. $Ac_m^t \leftarrow Ac_m^t \cup \{ac_l\}$
17. end if
18. end for
19. return Ac_m^t
20. end if

4.3 基于依赖约束的组合服务调度算法

算法 2. 基于依赖约束的组合服务调度算法

输入:

待重配的错位服务集合 Ac_m^t.

输出:

满足用户请求约束的多阶段决策调度集 SH.

1. $Q \leftarrow \{s_{l,n} \mid 0 \leqslant l \leqslant L_m - 1, 0 \leqslant n \leqslant |ac_l| - 1, ac_l \in Ac_m^t\}$
2. $H \leftarrow Q$, $P \leftarrow \Phi$, $d \leftarrow 0$
3. while $P \neq Q$ do
4. for $\forall s_{l,n} \in H$ do
5. if $pre(s_{l,n}) = \Phi$ then
6. $P \leftarrow P \cup \{s_{l,n}\}$
7. $H \leftarrow H - \{s_{l,n}\}$
8. $sh_d \leftarrow sh_d \cup \{s_{l,n}\}$
9. end if
10. end for
11. $SH \leftarrow SH \cup \{sh_d\}$
12. $d \leftarrow d + 1$
13. end while
14. return SH

基于算法 1 生成的错位服务集合 Ac_m^t,算法 2 提出一种基于依赖约束的组合服务调度算法,针对 Ac_m^t,建立优选调度模型,是一种规则式算法,旨在保证多个服务迁移时仍然满足用户请求的先后执行依赖约束. 算法流程包括:初始化基准集合 $Q = \{s_{l,n} \mid 0 \leqslant l \leqslant L_m - 1, 0 \leqslant n \leqslant |ac_l| - 1, ac_l \in Ac_m^t\}$(第 1 行). 遍历集合 $\forall s_{l,n} \in H$,查找所有 $pre(s_{l,n})$ 为空集的服务,将其加入调度集合,执行 $P \leftarrow P \cup \{s_{l,n}\}$, $H \leftarrow H - \{s_{l,n}\}$, $sh_d \leftarrow sh_d \cup \{s_{l,n}\}$(第 4—10 行). 迭代执行遍历集合 P,当满足终止条件,算法结束(第 3—13 行). 获取多阶段服务在线重配调度决策集,$SH = \{sh_d \mid 0 \leqslant d \leqslant |SH| - 1\}$(第 14 行),其中 sh_d 表示第 d 次调度所涉及的物联网服务集合.

算法 2 旨在调度一组服务,使其满足特定的依赖约束:(1)初始化正确性. 算法开始时,所有服务 $s_{l,n}$ 都被放入集合 Q 中,确保了所有服务都被考虑;(2)循环不变性. 在循环的每一步,算法都保证了集合 P 中的服务是没有先行依赖或者其依赖已经被添加到集合 P;(3)终止条件. 算法在 P 等于 Q 时停止,这时所有的服务都已经被考虑,确保了所有服务都可以被调度.

4.4 基于 DQRL 的服务配置优化算法

DQRL 算法的适配性:本文所研究的服务重配问题涉及到大量的物联网设备和服务,这导致了状态空间和动作空间都非常大. 此外,服务的需求和设备的资源都是动态变化的. DQRL 算法可以有效地处理这种动态性和复杂性,为每个时间步提供最优的服务配置策略.

论文所提基于 DQRL 的错位服务在线柔性重配算法旨在寻找全局最优解,即在整个边缘网络范围内,通过服务迁移实现资源的优化配置,以在边缘网络高负载时有效地适应资源动态变化,从而最大限度地满足用户请求的 QoS 约束.

错位服务多阶段决策调度 $SH = \{sh_d \mid 0 \leqslant d \leqslant |SH| - 1\}$ 转换为马尔可夫多阶段决策,辅助下一阶段服务配置决策预测信息,生成低代价的服务重配策略. 由于边物联网设备数量众多,服务重配策略状态空间会急速增加,强化学习算法在准确计算每个策略的系统收益时会产生巨大的计算量. 深度学习作为一种能够学习对象特征且运算速度快的方法,将其引入强化学习算法. 本节提出一种融合策略回放与双层网络相结合提升强化学习算法(DQRL),基于网络状态、动作和回报函数等不断更新,对边缘网络中用户服务请求进行自适应动态服务重配,具体步骤详见算法 3.

算法 3. 基于 DQRL 的错位服务在线柔性重配算法(E^2rC)

输入:

当前时刻(t)运行的用户服务请求 Ac^t;

在线即将($t+1$)到达的用户服务请求 Ac_n^{t+1};

执行 Ac^t 所涉及的物联网设备数量 K_s;

行为网络 B-Q 的权重训练参数 θ;

目标网络 T-Q 的权重训练参数 θ';

DQN 的策略回放池 $-D$;

DQN 与边缘网络交互的最大次数 Eps;

每一次交互中,DQN 训练步长 T.

输出:

错位服务在线柔性重配策略 $\boldsymbol{\zeta}(t) = \{\zeta_l(t)\}$.

1. $Ac_m^t \leftarrow$ 调用算法 1(Ac^t, Ac_n^{t+1})
2. $SH \leftarrow$ 调用算法 2(Ac_m^t)

3. for $\forall sh_d \in SH$ do
4. 初始 B-Q 和 T-Q,设置 $\theta' = \theta$
5. for each $eps = 1,2,\cdots,Eps$ do
6. 初始 DQN 的策略回放池 D
7. for step $t = 1,2,\cdots,T$ do
8. $A_t \leftarrow argmax_{A_t} Q(S_t, A_t;\theta)$
9. $R_t \leftarrow$ 调用算法 4($A_t.\boldsymbol{X}$, $A_t.\boldsymbol{\zeta}(t)$, K_s, sh_d)
10. 向策略回放池 D 存放 DRL 与边缘网络的一次交互样本 (S_t, A_t, R_t, S_{t+1})
11. 抽取小批量样本 M^τ 训练 B-Q 网络参数 θ
12. for $(S_\tau, A_\tau, R_\tau, S_{\tau+1}) \in M^\tau$ do
13. if 取到第 $\tau+1$ 个样本 then
14. $R_\tau^{tgt} \leftarrow R_\tau$
15. else
16. $R_\tau^{tgt} \leftarrow R_\tau + \gamma max_{A_{\tau+1}} Q(S_{\tau+1}, A_{\tau+1};\theta')$
17. end if
18. $L(\theta) \leftarrow E[(R_\tau^{tgt} - Q(S_\tau, A_\tau;\theta))^2]$
19. end for
20. $\theta \leftarrow argmin_\theta Q(L(\theta); M^\tau)$
21. $\theta' \leftarrow \theta$ 每执行 C 次
22. end for
23. end for
24. $\boldsymbol{\zeta}(t) \leftarrow \boldsymbol{\zeta}(t) \cup \{A_T\}$
25. end for
26. return $\boldsymbol{\zeta}(t)$

算法 3 描述了边缘网络资源失配时错位服务在线柔性重配方法,低代价的服务重配生成策略.首先,调用算法 1 获得边缘网络中错位服务的识别结果 Ac_m^t(第 1 行),输入算法 2 获得错位服务多阶段决策调度集 SH(第 2 行).算法 2 遍历执行 SH 中的调度 sh_d,执行深度强化学习(DQN),与边缘网络交互并获得网络状态空间集 S,对网络状态空间集给定网络动作空间集 A,网络状态发生改变时,得到一个新的状态空间,且得到执行此动作集后的系统回报值 R,继续不断与网络交互执行,学习知识使得决策更加适应网络环境(第 3～25 行).在算法 3 中,DQN 与边缘网络环境的每一次交互 eps 包含 T 次训练步长(第 7～22 行).在一个训练步长 t 内,所有的状态－动作对都由 Q-Network 计算并存储,表示为 $Q(S_t, A_t;\theta)$,即在状态 S_t 下,采取动作 A_t 后,未来决策阶段将得到的回报 R_t 值之和.其中,θ 表示网络的权重参数. S_t 和 A_t 详细表述如下:

(1)状态空间 S_t

第 d 次服务调度集 sh_d 中每一个物联网服务,根据当前边缘网络中物联网设备 V 的资源变动情况,为服务选择所配置的物联网设备.根据配置结果,计算获得服务运行效益 $Z_s(\boldsymbol{\zeta}(t))$ 和网络资源利用效益 $Z_r(\boldsymbol{\zeta}(t))$.状态空间 S_t 表征如公式(29)所示:

$$S_t = \{sh_d, V, Z_s(\boldsymbol{\zeta}(t)), Z_r(\boldsymbol{\zeta}(t))\} \tag{29}$$

(2)动作空间 A_t

算法根据当前状态 S_t,在动作空间 A_t 中选择当前决策最优动作 $argmax_{A_t} Q(S_t, A_t;\theta)$(第 8 行).根据迁移决策 $\boldsymbol{X}$,获取用户服务请求 Ac_m^t 的重配策略 $\boldsymbol{\zeta}(t)$.动作空间 A_t 表征如公式(30)所示:

$$A_t = \{\boldsymbol{X}, \boldsymbol{\zeta}(t)\} \tag{30}$$

(3)回报函数 R_t

在状态 S_t 下采取动作 A_t 所获得的回报收益 R_t,调用算法 4 获取(第 9 行). R_t 参见公式(27).回报函数与成本代价函数相关,目的是通过权衡服务重配运行效益和网络资源利用效益来确定执行服务重配的最佳策略.

基于当前训练步长 t,算法 3 选择 $Q(S_t, A_t;\theta)$ 最大值下对应的动作 A_t.因此,状态空间发生变化,进入下一个状态即 S_{t+1}.算法对系统的重配策略进行学习,根据回报函数,使得系统对于重配策略朝着最大化长期平均回报的方向学习. $Q(S_t, A_t;\theta)$ 更新规则如下:

$$Q(S_t, A_t;\theta) = (1-\alpha)Q(S_t, A_t;\theta) + \alpha[R_t + \gamma max_{A_{t+1}} Q(S_{t+1}, A_{t+1};\theta)] \tag{31}$$

其中,$\alpha \in [0,1]$ 表示 DQN 算法的学习率参数,$\gamma \in [0,1]$ 表示 DQN 算法的折扣因子,作为当前步下训练的服务重构策略的即时回报与长期回报之间的权衡.

神经网络训练样本独立同分布,而强化学习中样本数据关联且非静态.引入深度学习深度用非线性逼近器去代表动作的 Q 值时,由于策略样本数据之间有较强关联性,常会不稳定甚至不收敛.算法 2 采用融合策略回放(Replay Buffer)和双网络,即行为网络(Behavior Q-network,B-Q)、目标网络(Target Q-network,T-Q)相结合的机制,提升强化学习算法,以此消除连续样本间相关性,加快最优服务重构策略的收敛性.具体而言,算法 2 将状态转变作为与网络环境的一次交互,记为 (S_t, A_t, R_t, S_{t+1}) 并存入策略回放池 $-D$(第 10 行).抽取小批量样本 M^τ 对数据进行随机化,消除观测序列中的相关性,训练 B-Q 网络最小化损失函数(第 11～19 行).损失函数定义为目标值与预测值的均方差,通过更新权重 θ 使其最小化,如公式(32)所示:

$$L(\theta)=E[(R_\tau^{tgt}-Q(S_\tau,A_\tau;\theta))^2] \quad (32)$$

其中,$Q(S_\tau,A_\tau;\theta)$ 表示预测值由 B-Q 网络计算;R_τ^{tgt} 表示目标值,由 T-Q 网络计算,具体计算方式如下:

$$R_\tau^{tgt}=R_\tau+\gamma max_{A_{\tau+1}}Q(S_{\tau+1},A_{\tau+1};\theta') \quad (33)$$

如公式(31)所示,Q 值是依据当前时刻的回报和下一时刻的预测值进行更新,存在一些隐患,即数据样本差异可能造成一定的波动,每一轮迭代都可能产生一些波动,这些波动会立刻反映到下一迭代计算中,从而很难得到一个平稳的网络训练模型.因此,算法 3 引入 T-Q 网络,并周期性(每隔 C 步)拷贝 B-Q 网络的参数 θ,更新 T-Q 网络的参数值 θ'(第 20～21 行),减轻模型的波动性.具体描述为:在训练开始时,两个模型使用完全相同的参数 $\theta'=\theta$.在训练过程中,B-Q 网络负责与环境交互,得到交互样本.在学习过程中,使用 T-Q 网络计算目标值,该值在一段时间内将被固定;B-Q 网络计算预测值,并实时更新参数 θ.通过两部分数值的解耦、周期更新以此减轻模型波动性.

在第 d 次调度中,算法 3 与边缘网络交互,迭代上述过程(第 3～23 行),训练流程如图 1 所示.当 SH 中的所有调度 sh_d 执行完成时,算法 3 终止(第 3～25 行).关于错位服务 Ac_m^t,算法 3 返回生成的低代价的服务重构策略 $\boldsymbol{\zeta}(t)$(第 26 行).

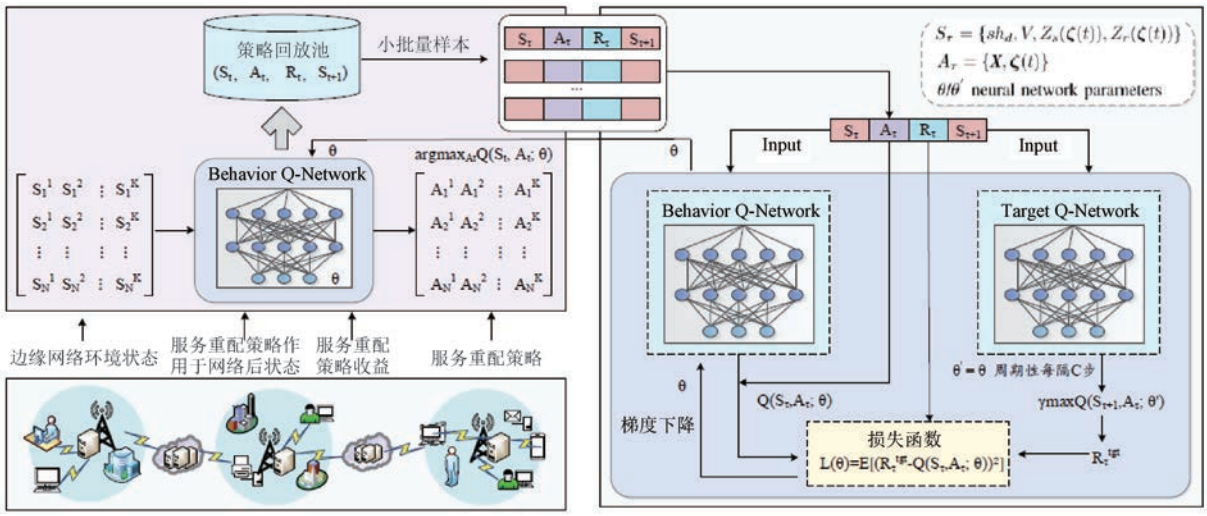

图 1　基于 B-Q 和 T-Q 双网络交互训练的服务重配策略

算法 3 旨在通过不断与环境交互的过程中,学习如何为用户服务请求找到最佳的重配策略,以最大化累积奖励.(1)模型正确初始化:行为网络 B-Q 和目标网络 T-Q 正确初始化并在迭代中更新.(2)策略回放池有效使用:验证策略回放池 D 的使用,它存储过去的经验,以防止快速的遗忘并提高学习稳定性.(3)决策行为正确选择:算法可以在每个决策点选择最优行为,这是通过 $Q(S_\tau,A_\tau;\theta)$ 函数的最大化实现的.(4)目标和损失函数正确构建:目标值 R_τ^{tgt} 和损失函数 $L(\theta)$ 正确反映预测值和目标值之间的差异.(5)网络权重正确更新:算法中权重更新步骤能够持续改善策略.对于算法 3 而言,主要受到 DQN 结构和在每次训练迭代中所采样的批量大小影响[54],时间复杂度是 $O(W)$,其中 W 是神经网络参数量.

4.5　基于系统收益的服务配置策略评估算法

算法 4 给出了第 d 次调度中服务重构策略效益的评估过程.服务重配代价成本主要包括:服务重配响应延迟、能耗和资源利用率,以计算得到的服务重配代价成本,构建服务重构效益评估模型,计算每一次服务调度 sh_d 的系统收益.首先,算法 4 根据输入参数 sh_d,获取属于 sh_d 的物联网服务对应的用户服务请求,记为 Ac_d^t(第 1～7 行).算法 4 获取边缘网络中关于第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\zeta_l(t)$ 的响应延迟和能量消耗性能度量值(第 9～18 行).最后,算法 4 生成了第 d 次调度中服务重构策略效益回报 R_t(第 20～23 行).

算法 4 旨在评估服务重配策略的系统收益,通过计算重配服务的计算延迟、能耗以及通信延迟和能耗、网络资源利用率等来实现这一点,并最终整合这些度量来得出系统收益.(1)收益和成本的正确计算:算法准确计算了每项服务重配的延迟和能耗(计算和通信),以及整体服务重配策略的收益.(2)符合目标函数:算法的输出需要满足定义的优化目标,即服务重配策略的系统收益最大化.(3)依赖关系保

证:算法正确处理服务之间的依赖关系,以保证服务重配不违反这些约束.

算法 4. 基于系统收益的服务柔性重配策略评估算法

输入:

错位服务集 Ac_m^t 中第 d 次调度中所参与重配的物联网服务数量 sh_d.

第 d 次调度中所参与重配的物联网服务所生成的迁移配置决策 $\boldsymbol{X}$.

第 d 次调度中所参与重配的物联网服务所生成的在线重配策略 $\boldsymbol{\zeta}(t)$.

执行 Ac^t 所涉及的物联网设备数量 K_s.

输出:

第 d 次调度在线重配策略的系统收益 R_t.

1. for $\forall s_{l,n} \in sh_d$ do
2. for $l = 0$ to $L_m - 1$ do
3. if $(s_{l,n} \in ac_l) \bigcap (ac_l \notin Ac_d^t)$ then
4. $Ac_d^t \leftarrow Ac_d^t \bigcup \{ac_l\}$
5. end if
6. end for
7. end for
8. for $\forall ac_l \in Ac_d^t$ do
9. for $n = 0$ to $N_l - 1$ do
10. if $s_{l,n} \in sh_d$ then
11. $T_e(s_{l,n}) \leftarrow$ 物联网服务 $s_{l,n}$ 重配的计算延迟度量由公式(3)计算获得
12. $E_e(s_{l,n}) \leftarrow$ 物联网服务 $s_{l,n}$ 重配的计算能耗度量由公式(10)计算获得
13. $T_c(s_{l,n}, s_{l,n'}) \leftarrow$ 物联网服务 $s_{l,n}$ 重配的通信延迟度量由公式(7)计算获得
14. $E_c(s_{l,n}, s_{l,n'}) \leftarrow$ 物联网服务 $s_{l,n}$ 重配的通信能耗度量由公式(12)计算获得
15. end if
16. end for
17. $T_l(\boldsymbol{\zeta}_l(t)) \leftarrow$ 第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\zeta_l(t)$ 的响应延迟性能度量,由公式(9)计算获得
18. $E_l(\boldsymbol{\zeta}_l(t)) \leftarrow$ 第 l 个用户服务请求 ac_l 生成的在线重配置策略 $\boldsymbol{\zeta}_l(t)$ 的能量消耗性能度量,由公式(14)计算获得
19. end for
20. $Z_r(\boldsymbol{\zeta}(t)) \leftarrow$ 网络资源利用效益,由公式(15)计算获得
21. $Z_s(\boldsymbol{\zeta}(t)) \leftarrow$ 服务重配运行效益,由公式(16)计算获得
22. $R_t \leftarrow$ 服务在线重配策略的系统收益,由公式(27)计算获得
23. return R_t

5 实验分析

本节通过构建实验仿真原型,模拟网络中不断接入的用户服务请求,并采用第 3 节提出的优化的服务适配组合算法将其部署在网络中的多个物联网设备上,实验从服务重配响应延迟、能量能耗和网络资源利用率三个方面与现有的技术进行对比,验证本文所提 E^2rC 算法的有效性和优势.

5.1 实验描述和参数设置

实验运行环境依托部署在配备 Intel(R) Core (TM) i7—10700F CPU @ 2.90 GHz 2.90 GHz 64 位 Windows10 操作系统,基于 x64 处理器.上海电信基站数据集包含 3233 个基站地理位置信息(如图 2 所示).该数据集收集来自上海基站连续 6 个月(2014 年 6 月 1 至 2014 年 11 月 30 日)用户的互联网接入日志,详细记录每个用户接入基站的开始和结束时间.每个基站的工作负载规定为用户的开始时间和结束时间计算出的总请求时间.该数据集提供的基站访问负载量,符合本文所提用户请求持续长时间部署所带来的设备资源存在过载等隐患而进行的错位服务重配研究的需求.实验选取连续 15 天(2014 年 6 月 1 日至 2014 年 6 月 15 日)基站的工作负载数据集作为输入,刻画提供用户服务请求的时间和空间分布信息,其中选取 6 个基站的负载信息如表 2 所示.

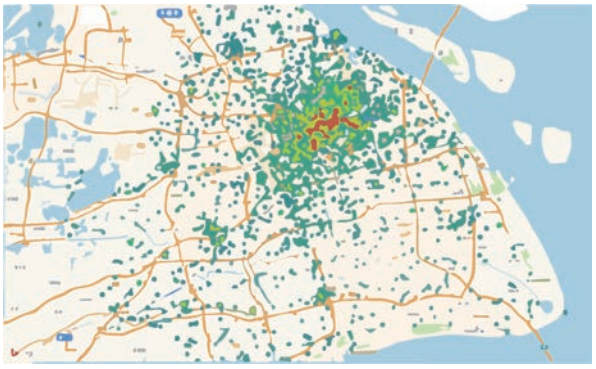

图 2 上海电信 3233 个基站的地理位置分布

表 2 上海电信 15 天数据集内的其中 5 个基站的接入负载统计

基站 ID	经度	纬度	用户服务请求 ID	基站负载/min
7	121.3825	31.2377	5	5556
193	121.1548	31.0154	367	454070
753	121.2564	31.4534	202	253360
1005	121.5077	31.3971	24	32982

在本文实验设置中,物联网设备的计算性能设置为[0.2,1.0] GHz,传输功率 P_i^T 设置为 0.2 W,物联网设备间通信的信道噪声功率 θ^2 设置为−174 dBm/Hz,通信带宽 B 范围设置为 10 KHZ.实验构建了具有一个输入层、两个隐藏层和一个输出层的全连接神经网络.两个隐藏层分别设置 100 个和 60 个隐藏神经元.算法的学习率和折扣因子分别设置为 0.1 和 0.9.其他实验参数设置如表 3 所示.

表 3　实验参数设置

参数名称	参数值
物联网设备的计算性能 $v_i.c_i^1$	[0.2, 1.0] GHz
物联网设备的传输功率 P_i^T	0.2 W
信道噪声功率 θ^2	−174 dBm/Hz
信道带宽 B	10 KHZ
硬件设施芯片架构参数 κ	10^{-26}
学习率 α	0.1
折扣因子 γ	0.9
ρ_s	0.5
ρ_r	0.5

5.2　实验对比方法

传统的强化学习算法采用 Q-table 存储状态−动作对,适合于状态−动作规模较小的情况,难以适用于边缘网络规模较大情况.本文所提 E²rC 算法采用 Q-Network 用于搜索策略解空间.在实验对比方法上,本文选择了在服务配置[36-38]相关研究中,从网络的服务响应延迟、能量消耗及资源利用率等性能指标方面,所常用的公认的动态规划对比算法,选取(1)与传统的强化学习策略对比,说明 E²rC 算法更加适合于边缘网络中服务重配问题的求解;(2)与基于 PSO 的资源分配算法对比,说明 E²rC 算法更加适合边缘网络下资源动态变化时资源调配;(3)与基于贪婪的服务放置算法对比,说明 E²rC 算法更加适合边缘网络下提升用户服务请求的吞吐量.所选对比方法具体说明如下:

(1)基于 Q-table 强化学习优化服务重配算法(TQL):运用 TQL 解决边缘网络资源失配时错位服务在线柔性重配问题.该算法采用传统的强化学习算法,而没有应用 Q-Network 学习最优策略.

(2)基于 PSO 资源分配算法(FTO):该算法关注物联网设备负载,以资源约束进行用户服务请求的在线任务分配,以优化服务响应延迟为目标,采用粒子群优化算法求解,生成一个近似最优服务调度策略.该算法未考虑在线到达请求的持续部署对于现有运行任务的 QoS 性能影响.

(3)基于贪婪服务放置算法(ISEP):该算法基于网络中未来用户服务请求,从云端提前拉取服务预配置至物联网设备上,以此减少用户服务请求的响应延迟.采用贪婪策略寻找剩余资源最多的物联网设备进行服务放置,生成一个近似最优服务放置策略.该算法未考虑物联网设备资源的占用、释放等动态变化对网络吞吐量的影响.

5.3　实验结果与分析

本文主要从边缘网络中用户服务请求数量、用户服务请求规模、物联网设备数量、物联网设备服务覆盖范围等四个不同参数设置下,从系统服务响应延迟、能量消耗和资源利用效益等三个性能指标,验证算法的有效性.所选实验参数设置如下:

(1)用户服务请求数量:数值设置范围从 3 到 20,增量为 5,以评估 E²rC 算法在应对突发请求激增方面,对网络资源调配、吞吐量的有效性.默认值设置为 5,用于其他参数算法性能参数设置实验.

(2)用户服务请求规模:该值范围从 3 到 10,增量为 2,指定用户服务请求中子任务数量,用于分析 E²rC 算法在网络资源自适应方面的性能.默认值设置为 6.

(3)物联网设备数量:该值范围从 100 到 800,增量为 200,以分析 E²rC 算法在物联网设备增多下对于生成低代价的服务重配策略的性能影响.默认值设置为 400.

(4)物联网设备服务覆盖范围:以物联网设备的通信半径表示.设置范围从 50 到 200 不等,增量为 50.以分析 E²rC 算法随着物联网设备通信服务范围的增大,对于服务重配策略下网络资源利用效益的性能评估,默认值被设置为 50.

从以下三个方面评估算法性能影响:

(1)评估边缘网络中不同用户服务请求数量对算法性能影响

边缘网络中用户服务请求数量影响物联网设备的负载,直接影响到请求处理效率.图 3 分别展示了不同用户服务请求数量设置下,E²rC、TQL、FTO、ISEP 算法性能对比评估,包括服务响应延迟、能量消耗和资源利用效益.其中,用户服务请求规模设置为 6,部署的物联网设备数量设置为 400,物联网设备的服务覆盖范围设置为 50.

服务响应延迟:图 3(a)展示了在网络中不同的用户服务请求数量设置下,E²rC 算法与 TQL、FTO 和 ISEP 算法在服务响应延迟性能方面的对比评估.从图中可以看出,随着网络中用户服务请求数量的增加,E²rC 算法与对比算法所带来的物联网设备的能量消耗都有所增加,这是由于用户服务请求持

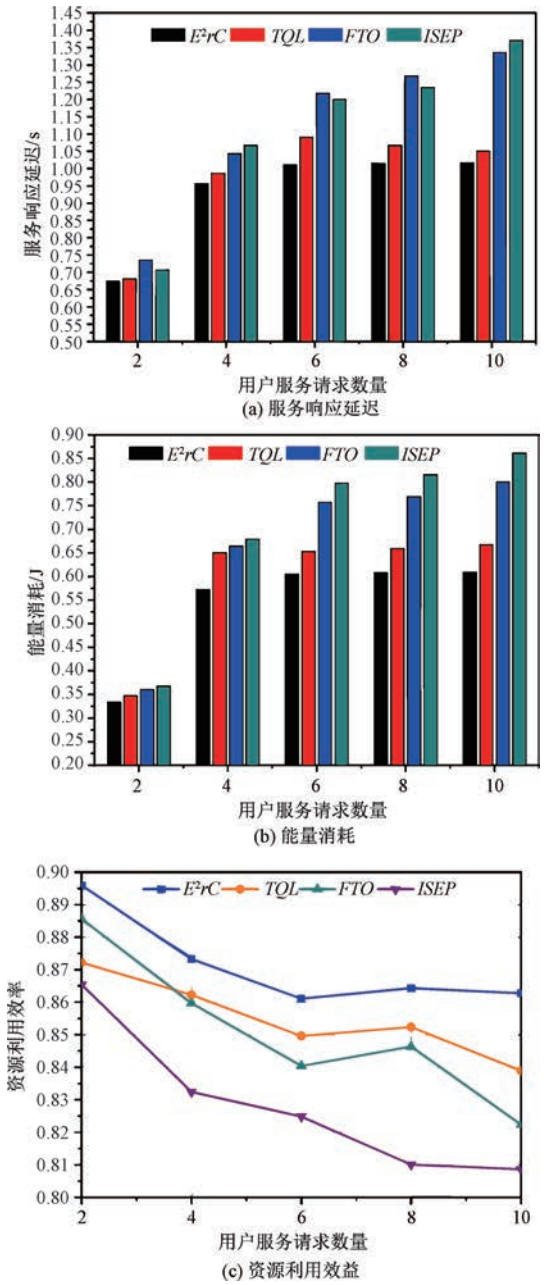

图 3 不同用户服务请求设置下 E²rC、TQL、FTO 和 ISEP 四种算法性能对比

续部署使得网络中某些物联网设备负载有所增加，在不同设备之间发生服务迁移进行文件数据的传输，从而提高了服务重配所带来的响应延迟. 此外，在用户服务请求数量相同设置下，所提 E²rC 算法在服务响应延迟性能方面优于 TQL、FTO 和 ISEP 算法. 事实上，对于 TQL 算法，在寻找服务重配策略时，以 Q-table 存储每一对 $Q(S_t,A_t)$. TQL 算法从 Q-table 中选择 $Q(S_t,A_t)$ 数值最大的动作用于指导算法迭代. 当网络中部署的请求数量增加时，TQL 的状态空间和动作空间呈现出高维化，导致 Q-table 的尺寸增大. 因此，Q-table 在时间效率上可能不适合获得最优的重新配置策略生成. E²rC 算法中，使用 Q-network 有效存储查找 $Q(S_t,A_t)$，解决了状态和动作空间高维遍历时间消耗问题. 对于 FTO 和 ISEP 算法，这些技术主要强调部署在线到达的用户服务请求的资源分配，而忽略了这些在线到达的请求有可能导致网络中某些物联网设备资源占用的增加，从而影响正在运行的用户请求的服务响应延迟. 如算法 1 和算法 2 所示，E²rC 算法捕获网络中物联网设备资源约束的满足度变化，识别网络中错位服务，同时以某种符合请求约束的方式进行服务多阶段决策调度. 总的来说，所提 E²rC 算法相较于对比算法，更适应于解决边缘网络下资源失配时错位服务重配问题，提升服务响应速度.

物联网设备能量消耗：图 3(b)展示了不同用户服务请求数量设置下，E²rC 算法与 TQL、FTO 和 ISEP 算法在物联网设备能量消耗性能方面的对比评估. 从图中可以看出，随着网络中用户服务请求数量的增加，E²rC 算法与对比算法的能量消耗都有所增加，这是由于用户服务请求持续部署使得网络中某些物联网设备负载有所增加，在不同设备之间发生服务迁移进行文件数据的传输，从而使得服务重配所带来的物联网设备能量消耗增加. 另外，在用户服务请求数量相同设置下，所提 E²rC 算法在物联网设备的能量消耗性能方面优于 TQL、FTO 和 ISEP 算法. E²rC 在生成低代价的服务重配策略时，更关注网络资源动态变化，将物联网设备负载纳入算法决策，使得物联网设备在能量消耗方面更加平衡，也更有利于延长整个边缘网络的生命周期. TQL、FTO 和 ISEP 则更侧重于优化服务响应延迟，而忽略了优化物联网设备资源.

网络资源利用效益：图 3(c)展示了在网络中不同的用户服务请求数量设置下，E²rC 算法与 TQL、FTO 和 ISEP 算法在网络资源利用效益性能方面的对比评估. 可以看出，随着网络中用户服务请求数量的增加，E²rC 算法所带来的资源利用效益分别比 TQL 算法高约 2%，比 FTO 算法高约 3%，比 ISEP 算法高约 5%. 如算法 3 所述，E²rC 考虑了在线到达请求的持续部署对于现有运行任务的 QoS 性能影响，关注物联网设备资源的占用、释放等动态变化，从而提高网络吞吐量. 然而 TQL、FTO 和 ISEP 在进行服务分配放置时未考虑调整现有运行任务的资源分配实现网络资源利用效益的提升. 我们的方法在服务重配过程中，考虑了物联网设备的资源状况，避免了

某些设备过载,以此最大程度地满足用户请求的 QoS 约束,并提高了整个边缘网络的资源利用效益.这种资源感知性是我们方法的一个重要特点,使得用户请求能够更好地适应边缘网络中资源动态变化.

(2)评估边缘网络中用户服务请求任务规模对算法性能影响

图 4 分别展示了在参数用户服务请求任务规模设置下,E^2rC、TQL、FTO、ISEP 算法性能对比评估,包括服务响应延迟、能量消耗和资源利用效益.其中,用户服务请求数量设置为 5,部署的物联网设备数量设置为 400,物联网设备的服务覆盖范围设置为 50.

服务响应延迟:图 4(a)展示了网络中用户服务请求任务规模不同数量设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在服务响应延迟性能方面的对比评估.从图中可以看出,所提 E^2rC 算法在服务响应延迟性能方面优于 TQL、FTO 和 ISEP 算法.注意到,用户服务请求任务之间的先后执行依赖约束,当某些物联网设备的资源约束被违反时,一些错位服务重配执行,会导致出现级联服务发生重配.因此,随着用户服务请求规模的增加,服务重配策略的访问延迟可能会增加.如算法 3 所述,E^2rC 联合物联网设备资源的动态变化,算法采用多阶段决策规约满足组合服务的依赖约束,设置 B-Q 和 T-Q 两个网络,将未来时刻服务重配收益加入当前决策,生成低代价的服务重配策略.而对比算法 TQL、FTO 和 ISEP 在生成服务分配策略时,仅考虑了在线到达的用户服务请求的资源分配,并不适用于网络中运行的组合服务的动态自适应重配置.

物联网设备能量消耗:图 4(b)展示了网络中用户服务请求任务规模不同数量设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在物联网设备能量消耗性能方面的对比评估.在用户服务请求规模相同设置下,与 TQL、FTO、ISEP 算法相比,E^2rC 算法所带来的物联网设备的能量消耗相对较小,正如图 4(b)所提到的,E^2rC 算法的服务重配策略考虑了未来后续级联服务的重配策略指导当前决策调度阶段的服务重配策略生成.此外,随着用户服务请求规模的增加,E^2rC 算法所带来的物联网设备的能量消耗在逐渐增加.主要在于 E^2rC 算法考虑了级联服务的重新配置,服务规模数量的增加可能会导致由不同物联网设备托管的更多服务被重新配置,因此,随着用户服务请求规模的增加,传输迁移后服务数据包会消耗更多的物联网设备能量.

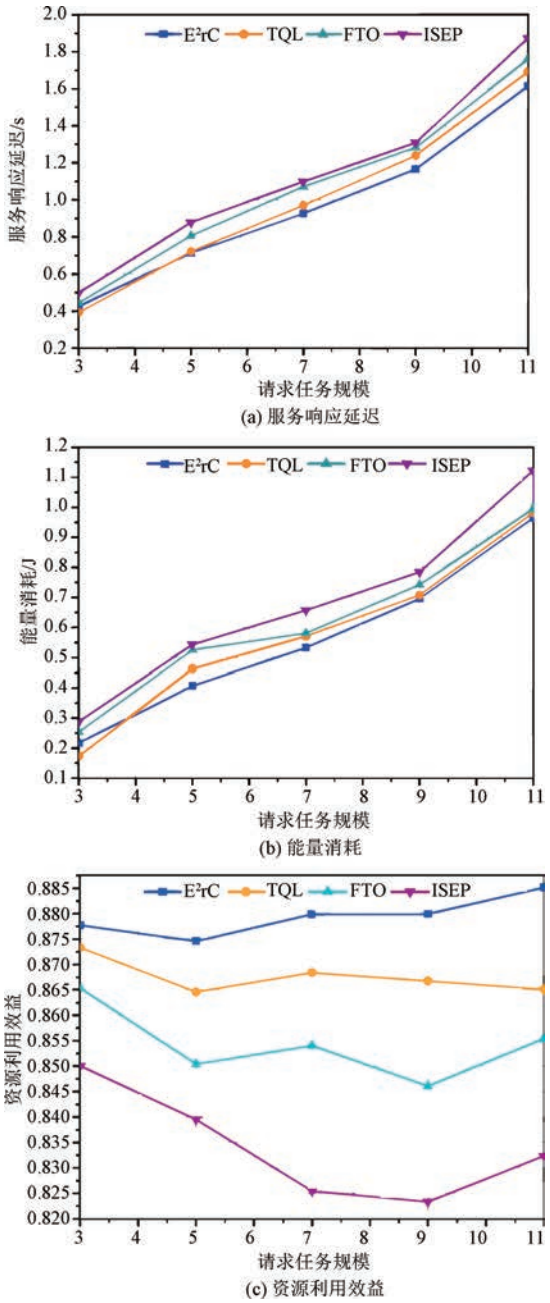

图 4　用户服务请求任务规模设置下 E^2rC、TQL、FTO 和 ISEP 四种算法性能对比

网络资源利用效益:图 4(c)展示了网络中用户服务请求任务规模不同数量设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在网络资源利用效益性能方面的对比评估.从图中可以看出,随着用户服务请求规模的增加,与 TQL、FTO、ISEP 算法相比,E^2rC 算法在提高网络资源利用效益方面表现较好.事实上,这些对比算法的重点在于优化服务响应延迟,而忽略了用户服务请求持续部署,算法在应对网络突发激增请求时物联网设备的资源动态变化,从而导致对比算法所生成策略的资源利用效益相对较低.

正如算法 2 所示，E^2rC 算法加入了基于依赖约束的组合服务调度算法来捕获由物联网设备共同支持的多个正在运行的用户服务请求的子任务之间资源约束的满意度变化，保障了这些重新配置的服务能够托管在具有足够剩余资源的设备.

(3)评估边缘网络中物联网设备数量对算法性能影响

边缘网络中物联网设备数量决定了提供用户服务请求的可支持能力，从而影响到请求处理效率. 图 5 分别展示了在不同的物联网设备数量设置下，E^2rC、TQL、FTO、ISEP 算法性能对比评估，包括服务响应延迟、能量消耗和资源利用效益. 其中，用户服务请求数量设置为 5，用户服务请求规模设置为 6，物联网设备的服务覆盖范围设置为 50.

服务响应延迟：图 5(a)展示了网络中不同的物联网设备数量设置下，E^2rC 算法与 TQL、FTO 和 ISEP 算法在服务响应延迟性能方面的对比评估. 可以看出，随着物联网设备数量的增加，与 TQL、FTO、ISEP 算法相比，E^2rC 算法的服务访问延迟略有减少. 主要在于，物联网设备数量的增加表明，具有充足剩余资源的可供用于承载服务的物联网设备随之增加，E^2rC 算法通过服务重配策略以进一步促进多个具有前后依赖执行关系的物联网服务.

在相同的物联网设备上运行，从而减少用户服务请求的子任务间通信延迟开销，优化了服务请求处理的响应延迟. 对于 TQL、FTO 和 ISEP 算法，主要关注用户服务请求的静态分配，当网络中物联网设备增加，用户服务请求可以通过本地物联网设备得到满足，但是忽略了对于物联网设备资源的优化. 图 5(a)表明本文所提 E^2rC 算法更加适合于边缘网络下持续部署的用户服务请求.

物联网设备能量消耗：图 5(b)展示了网络中不同的物联网设备数量设置下，E^2rC 算法与 TQL、FTO 和 ISEP 算法在物联网设备能量消耗性能方面的对比评估. 可以看出，随着物联网设备数量的增加，E^2rC 算法获得的服务重配策略在平衡边缘网络中物联网设备的能耗方面表现优于 TQL、FTO 和 ISEP 算法. 主要在于，ISEP 算法采取贪婪策略，选择了具有剩余资源最大的物联网设备来托管服务，但忽略了边缘网络中物联网设备之间的传输能耗. FTO 算法在优化用户服务请求的访问延迟时，并未考虑物联网设备的能耗. 图 5(b)所示，TQL 算法也可以实现更低的能耗，但在处理服务重配策略大规模状态和动作空间时，本文所提 E^2rC 算法采用的

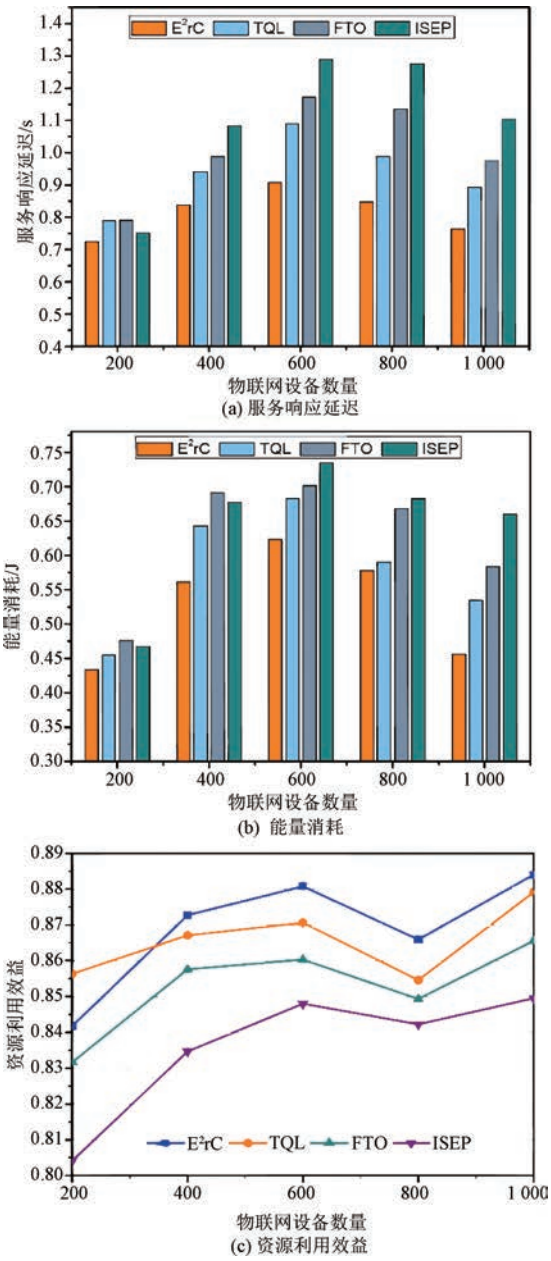

图 5 物联网设备数量设置下 E^2rC、TQL、FTO 和 ISEP 四种算法性能对比

策略回放池和两层 Q-network 比 TQL 效率更好. 总的来说，在边缘网络内可以针对空间位置需求，适当部署物联网设备数量，可以进一步平衡网络消耗，延长网络生命周期.

网络资源利用效益：图 5(c)展示了网络中不同的物联网设备数量设置下，E^2rC 算法与 TQL、FTO 和 ISEP 算法在网络资源利用效益性能方面的对比评估. 可以看出，随着物联网设备数量的增加，与 TQL、FTO 和 ISEP 算法相比，E^2rC 算法的网络资源利用效益性能较好于对比算法. 事实上，当边缘网络中有更多的物联网设备和足够的资源时，可能会

有更多的物联网设备作为候选设备来共同托管迁移的服务.如 3.1 节所述,本文所提 E^2rC 算法在生成低代价的服务重配策略时综合考虑了边缘网络的资源平衡,而网络资源负载平衡却不是 TQL、FTO 和 ISEP 算法的重点.总的来说,E^2rC 算法可以更好地平衡边缘网络中现有运行、在线持续部署的用户请求之间对于网络资源的竞争需求.

(4)评估边缘网络中物联网设备服务覆盖范围对算法性能影响

图 6 分别展示了不同设备服务覆盖范围设置下,E^2rC、TQL、FTO、ISEP 算法性能对比评估,包括服务响应延迟、能量消耗和资源利用效益.其中,用户服务请求数量设置为 5,用户服务请求规模设置为 6,部署的物联网设备数量设置为 400.

服务响应延迟:图 6(a)展示了网络中不同的物联网设备服务覆盖范围设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在服务响应延迟性能方面的对比评估.可以看出,随着物联网设备服务覆盖范围的增加,采用 E^2rC 算法的用户服务请求的响应延迟小于 TQL、FTO 和 ISEP 算法.当检测到某些物联网设备过载时,一些物联网服务需要迁移到一些位于过载设备覆盖范围内的物联网设备上,如公式(26)所述.当物联网设备的覆盖范围扩大时,可用于托管这些迁移服务的物联网设备的数量也会增加,从而可以减少 3.2 节中所述的用户服务请求子任务间数据传输的通信延迟.

物联网设备能量消耗:图 6(b)展示了网络中不同的物联网设备服务覆盖范围设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在物联网设备能量消耗性能对比评估.可以看出,随着物联网设备覆盖范围的增加,采用 E^2rC 算法响应用户服务请求所带来的物联网设备的能量消耗小于 TQL、FTO 和 ISEP 算法.如 3.3 节所述,E^2rC 算法考虑了服务迁移成本,将其集成到服务重配运行效益的生成策略.其他比较算法未考虑服务迁移过程中消耗的物联网设备的能量消耗,在边缘网络物联网资源稀缺且非均下,这些能量消耗会对网络的生命周期产生影响.图 6(b)表明可以通过设置适当的物联网设备覆盖范围,更好地优化物联网设备能耗.

网络资源利用效益:图 6(c)展示了网络中不同的物联网设备服务覆盖范围设置下,E^2rC 算法与 TQL、FTO 和 ISEP 算法在网络资源利用效益性能方面的对比评估.可以看出,E^2rC 算法在物联网设备的资源利用效率方面的表现较好于 TQL、FTO

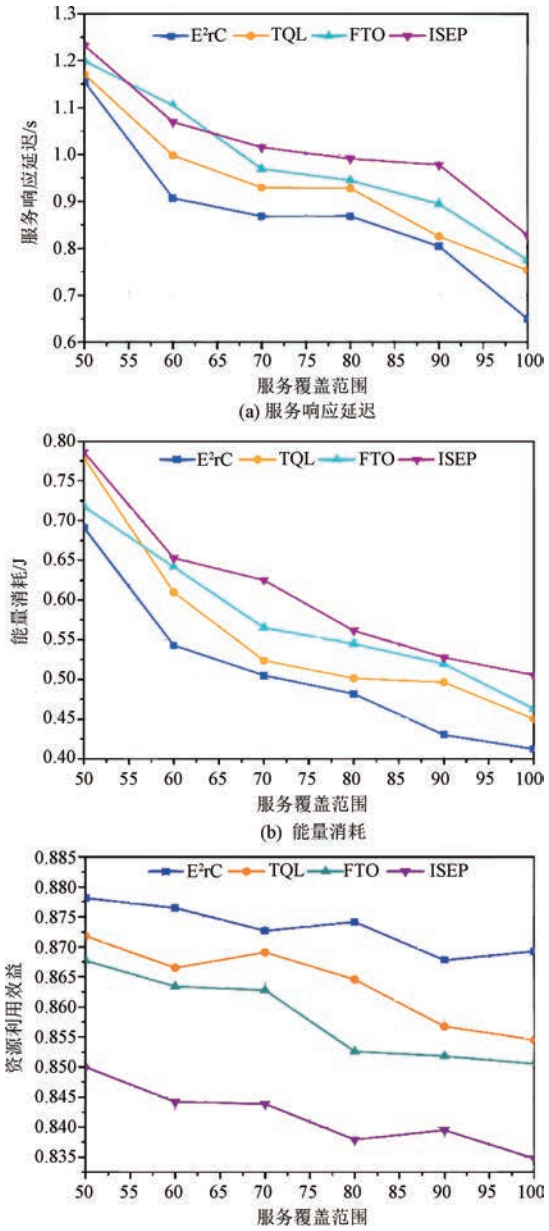

图 6　物联网设备服务覆盖范围设置下 E^2rC、TQL、FTO 和 ISEP 四种算法性能对比

和 ISEP 算法.随着物联网设备覆盖范围的增加,E^2rC 算法的资源利用率逐渐趋于稳定,并收敛到一个稳定的值.实际上,物联网设备的覆盖范围扩大到有利于可供托管运行迁移服务的候选物联网设备数量的增加,进一步平衡过载设备的资源,提高网络资源利用率.图 6(c)表明 E^2rC 算法在应对网络中用户服务请求在计算资源激增需求方面,生成的低代价服务重配策略更加有效.

综上实验分析,本文从算法的设计原理、策略目标、优化效率、实际应用潜力等角度定性总结了各个算法特性,如表 4 所示.

表 4 算法特性分析

特性/算法	E^2rC	TQL	FTO	ISEP
设计原理	基于深度神经网络的在线强化学习	基于马尔可夫决策的在线强化学习	采用粒子群优化算法求解	基于贪婪策略算法求解
策略目标	低代价服务重配,高资源利用率	优化网络资源负载	优化服务响应延迟	优化服务响应延迟
优化效率	高,综合了多个服务和资源指标	中,受限于算法状态空间大小	低,未考虑全局优化	低,未考虑全局优化
应用潜力	高,适用于资源受限且需求动态的场景	中,适用于资源受限且需求变动不大的场景	一般,适用于用户需求非动态变化的场景	低,适用于资源充足且较为固定的场景

6 讨 论

E^2rC 算法为边缘计算和物联网领域的资源优化和服务配置提供了一种有效的解决方案,不仅在学术理论上具有创新性,更在工业实践中展现出其实质性的应用价值.通过智能化地迁移服务,实现资源的动态平衡与高效利用,为工业界的智能化转型提供有力技术支撑.

6.1 E^2rC 在边缘计算中的泛化性与多因素考量

为增强算法在实际应用场景的泛化性,本文在设计算法时全面考虑了多种可能的网络变化因素.具体而言,本文考虑的网络变化因素包括但不限于以下几个方面:

(1)资源类别的多样性:网络中可能存在的资源类型丰富多样,如计算资源、存储资源、通信带宽等.本文算法根据不同资源类别的特性进行智能调度和分配,以适应网络资源的动态变化.

(2)资源剩余量的实时变化:网络资源剩余量会随着时间和服务请求的变化而实时波动.本文算法实时感知设备资源情况,并根据资源的实时剩余量进行动态的服务重配和资源调度,以确保服务请求的稳定性和高效性.

(3)服务请求的波动:服务请求数量、类型等可能会随着时间和用户行为的变化而波动.本文算法能够适配网络中随时到达的用户请求进行资源分配并提高整体服务质量.

(4)网络负载的变化:网络负载会直接影响服务的响应时间和资源利用率.本文算法感知网络负载的变化,并通过合理的服务调度和资源分配来平衡网络负载,提高系统的吞吐量和稳定性.

6.2 DRL 在边缘计算应用:理论探索与案例分析

深度强化学习(DRL)以其深度学习和强化学习融合的独特优势,在边缘计算领域展现出广阔的应用潜力.本文所提出的 E^2rC 算法正是基于这一框架设计的,旨在解决边缘计算环境中服务请求和资源分配的动态优化问题.

首先,DRL 的引入为 E^2rC 算法赋予了在不确定环境中学习和决策的能力.在边缘计算实际应用中,服务请求、网络状态和设备性能等因素都在实时变化,这种不确定性使得资源分配和服务调度变得尤为复杂.然而,DRL 通过与环境进行持续交互,不断学习并优化决策策略,使得算法能够逐渐适应这种不确定性,实现高效的资源分配和服务调度.

其次,DRL 强大之处在于其能够处理高维状态和动作空间[55].在边缘计算环境中,大量的设备和服务相互交互、协作,构成了一个高度复杂的系统.传统的优化方法往往难以处理这种高维状态空间,而 DRL 能够通过深度神经网络的强大表示能力,将高维状态空间映射到低维的决策空间,从而实现高效的优化和决策.这使得 E^2rC 算法在面对复杂的边缘计算环境时,能够更加精准地制定资源分配和服务调度策略.

本文实验主要基于数据集进行模拟,DRL 的通用性和灵活性使得 E^2rC 算法能够适应多种边缘计算场景.DRL 在边缘计算的实际应用案例已经逐渐涌现[55-57],例如,在智能家居领域,DRL 算法可以根据用户的习惯和偏好,智能地调度家电设备的运行,实现资源高效利用;在工业自动化领域,DRL 算法可以优化生产线的调度和资源配置,提高生产效率和质量.这些实际应用案例不仅验证了 DRL 在边缘计算领域的有效性,也为 E^2rC 算法在实际系统大规模应用的实际效果方面提供了实验支撑.

综上所述,无论是智能家居、智能城市还是工业自动化等领域,这些场景中,设备资源有限且分布不均,同时需要应对大量具有严格实时性要求的用户请求.本文所提算法可以针对具体场景进行定制化设计,实现更加精准和高效的资源分配和服务重配策略,满足了实际应用需求对于服务可靠性及低延迟的严苛要求,能够在更多实际场景中得到应用与验证,进一步推动边缘计算和物联网技术在产业界的深度融合与发展.

7 总　结

本文提出了一种资源高效的服务柔性重配算法(E^2rC),旨在均衡边缘网络设备运行时的资源负载,对某些过载的物联网设备上运行的某些服务进行重新配置,使其迁移到具有充足剩余资源的临近物联网设备上,同时这些服务的 QoS 仍然可满足.这些释放的资源可用来支撑即将在线到达的用户请求,从而达到整体提升用户服务请求 QoS 的目标,提升边缘网络中服务请求的健壮性,提高边缘网络针对服务请求的吞吐量.E^2rC 算法步骤如下,综合考虑网络资源负载特性,在线探测错位服务并判定其最小集,构建一种基于 QoS 感知的错位服务在线探测算法,检测过载物联网设备,识别服务重配最小集.基于错位服务识别结果,提出一种依赖约束的组合服务调度算法,以满足组合服务依赖约束,将服务重配到具有足够剩余资源的临近物联网设备上.最后,研究错位服务在线柔性重配的代价最优策略,构建服务重配代价模型,依据网络资源动态变化,基于依赖约束的组合服务多阶段调度建模为马尔可夫多阶段决策,提出一种融合策略回放与双层网络相结合提升强化学习算法.最后,基于上海电信基站数据集,在不同实验参数变量设置下,验证算法的有效性.实验结果表明,所提 E^2rC 算法相比较于对比算法,在满足用户服务请求的时延约束下,降低了物联网设备能量消耗,提高了边缘网络资源利用效益.为更好地适应大规模应用需求,未来研究会关注于将目前的工作扩展到边缘端算法的分布式训练,形成多 Agent 协作,以充分利用边缘设备协作能力,提升算法训练效率和性能.

参　考　文　献

[1] Behravesh R, Harutyunyan D, Coronado E, et al. Time-sensitive mobile user association and SFC placement in MEC-Enabled 5G networks. IEEE Transactions on Network and Service Management, 2021, 18(3): 3006-3020

[2] Mezni H, Benslimane D, Bellatreche L. Context-aware service recommendation based on knowledge graph embedding. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(11): 5225-5238

[3] Li Zhiyong, Wang Qi, Chen Yifan, et al. A survey of task offloading in vehicular edge computing environments. Chinese Journal of Computers, 2021, 44(5): 963-982
(李智勇, 王琦, 陈一凡等. 车辆边缘计算环境下任务卸载研究综述. 计算机学报, 2021, 44(5): 963-982)

[4] Zhou X, Peng X, Xie T, et al. Delta debugging microservice systems with parallel optimization. IEEE Transactions on Services Computing, 2022, 15(1): 16-29

[5] Tang Q, Xie R, Yu F R, et al. Distributed task scheduling in serverless edge computing networks for the internet of things: A learning approach. IEEE Internet of Things Journal, 2022, 9(20): 19634-19648

[6] Lin C, Mahmoudi N, Fan C, et al. Fine-grained performance and cost modeling and optimization for FaaS applications. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(1): 180-194

[7] Wang Z, Lv T, Chang Z. Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing. Computer Networks, 2022, 205: 108732

[8] Li X, Zhou Z, Shi Z, et al. Energy-efficient anomaly detection with primary and secondary attributes in edge-cloud collaboration networks. IEEE Internet of Things Journal, 2021, 8(15): 12176-12188

[9] Tang J, Wu S, Wei L, et al. Energy-efficient sensory data collection based on spatiotemporal correlation in IoT networks. International Journal of Crowd Science, 2022, 6(1): 34-43

[10] Kuang Zhufang, Chen Qinglin, Li Linfeng, et al. A multi-user edge computing task offloading scheduling and resource allocation algorithm based on deep reinforcement learning. Chinese Journal of Computers, 2022, 45(4):812-824
(邝祝芳, 陈清林, 李林峰等. 基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法. 计算机学报, 2022, 45(4):812-824)

[11] Tran-dang H, Kim D S. FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(10): 2491-2508

[12] Chen Y, Sun Y, Wang C, et al. Dynamic task allocation and service migration in edge-cloud IoT system based on deep reinforcement learning. IEEE Internet of Things Journal, 2022, 9(18): 16742-16757

[13] Li B, Cheng B, Chen J. An efficient algorithm for service function chains reconfiguration in mobile edge cloud networks//Proceedings of the 2021 IEEE International Conference on Web Services (ICWS). 2021: 426-435

[14] Liang Y, Ge J, Zhang S, et al. Interaction-oriented service entity placement in edge computing. IEEE Transactions on Mobile Computing, 2021, 20(3): 1064-1075

[15] Wang S, Guo Y, Zhang N, et al. Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach. IEEE Transactions on Mobile Computing, 2021, 20(3): 939-951

[16] Donassolo B, Legrand A, Mertikopoulos P, et al. Online reconfiguration of IoT applications in the fog: The Informa-

tion-coordination trade-off. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(5): 1156-1172

[17] Goudarzi M, Wu H, Palaniswami M, et al. An application placement technique for concurrent IoT applications in edge and fog computing environments. IEEE Transactions on Mobile Computing, 2021, 20(4): 1298-1311

[18] An X, Fan R, Hu H, et al. Joint task offloading and resource allocation for IoT edge computing with sequential task dependency. IEEE Internet of Things Journal, 2022, 9(17): 16546-16561

[19] Battula S K, O'reilly M M, GARG S, et al. A generic stochastic model for resource availability in fog computing environments. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(4): 960-974

[20] Gao H, Ma W, He S, et al. Time-segmented multi-level reconfiguration in distribution network: A novel cloud-edge collaboration framework. IEEE Transactions on Smart Grid, 2022, 13(4): 3319-3322

[21] Li Hui, Li Xiuhua, Xiong Qingyu, et al. Edge computing assisting industrial internet: Architecture, applications, and challenges. Computer Science, 2021, 48(1): 1-10
(李辉, 李秀华, 熊庆宇等. 边缘计算助力工业互联网:架构、应用与挑战. 计算机科学, 2021, 48(1): 1-10)

[22] Zhang S, Wang C, Jin Y, et al. Adaptive configuration selection and bandwidth allocation for edge-based video analytics. IEEE/ACM Transactions on Networking, 2022, 30(1): 285-298

[23] Sun M, Zhou Z, Wang J, et al. Energy-efficient IoT service composition for concurrent timed applications. Future Generation Computer Systems, 2019, 100: 1017-1030

[24] Du X, Tang S, Lu Z, et al. A novel data placement strategy for data-sharing scientific workflows in heterogeneous edge-cloud computing environments//Proceedings of the IEEE International Conference on Web Services (ICWS). 2020: 498-507

[25] Chadha M, Jindal A, Gerndt M. Architecture-specific performance optimization of compute-intensive faaS functions//Proceedings of the 2021 IEEE 14th International Conference on Cloud Computing (CLOUD). 2021: 478-483

[26] Farhadi V, Mehmeti F, He T, et al. Service placement and request scheduling for data-intensive applications in edge clouds. IEEE/ACM Transactions on Networking, 2021, 29(2): 779-792

[27] Sun M, Zhou Z, Xue X, et al. Adaptive configuration of service-based smart sensors in edge networks. IEEE Transactions on Industrial Informatics, 2022, 18(4): 2674-2683

[28] Elshazly H, Ejarque J, Badia R M. Storage-heterogeneity aware task-based programming models to optimize I/O intensive applications. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 3589-3599

[29] Jin P, Hao X, Wang X, et al. Energy-efficient task scheduling for CPU-intensive streaming jobs on hadoop. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(6): 1298-1311

[30] Li X, Zhou Z, Zhao Z, et al. Data & computation-intensive service re-scheduling in edge networks//Proceedings of the IEEE International Conference on Web Services (ICWS). 2021: 389-396

[31] Botangen K A, Yu J, Han Y, et al. Quantifying the adaptability of workflow-based service compositions. Future Generation Computer Systems, 2020, 102: 95-111

[32] Zhu T, Li J, Cai Z, et al. Computation scheduling for wireless powered mobile edge computing networks//Proceedings of the IEEE Conference on Computer Communications (INFOCOM). 2020: 596-605

[33] Long Long, Liu Zichen, Lu Zaiwang, et al. Joint optimization strategy for service caching and resource allocation in mobile edge networks. Journal of Communications, 2023, 44(1): 64-74
(龙隆, 刘子辰, 陆在旺等. 移动边缘网络下服务缓存与资源分配联合优化策略. 通信学报, 2023, 44(1): 64-74)

[34] Chen C L, Brinton C G, Aggarwal V. Latency minimization for mobile edge computing networks. IEEE Transactions on Mobile Computing, 2023, 22(4): 2233-2247

[35] Zhang Feifei, Ge Jidong, Li Zhongjin, et al. Collaborative computation offloading and dynamic task scheduling in edge computing. Journal of Software, 2023,34(1): 1-20
(张斐斐, 葛季栋, 李忠金等. 边缘计算中协作计算卸载与动态任务调度. 软件学报, 2023,34(1): 1-20)

[36] Afrasiabi S N, Ebrahimzadeh A, Mouradian C, et al. Reinforcement learning-based optimization framework for application component migration in NFV cloud-fog environments. IEEE Transactions on Network and Service Management, 2022, 20(2): 1866-1883

[37] Li Yun, Gao Qian, Yao Zhixiu, et al. Joint optimization method for intelligent service orchestration and network-computing resource allocation in mobile edge computing. Journal of Communications, 2023, 44(7): 51-63
(李云, 高倩, 姚枝秀等. 移动边缘计算中智能服务编排和算网资源分配联合优化方法. 通信学报, 2023, 44(7): 51-63)

[38] Liu Yao, He Yueyuan, Zhou Hongjing, et al. Partial computation offloading method based on joint resource allocation in mobile edge computing. Chinese Journal of Internet of Things, 2023, 7(1): 140-148
(刘耀, 何岳园, 周红静等. 移动边缘计算中基于资源联合分配的部分计算卸载方法. 物联网学报, 2023, 7(1): 140-148)

[39] Hao Y, Cao J, Wang Q, et al. Energy-aware scheduling in edge computing with a clustering method. Future Generation Computer Systems, 2021, 117: 259-272

[40] Xu K, Lv L, Li T, et al. Minimizing tardiness for data-intensive applications in heterogeneous systems: A matching theory perspective. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(1): 144-158

[41] Kaur A, Auluck N, Rana O. Real-time scheduling on hierar-

chical heterogeneous fog networks. IEEE Transactions on Services Computing, 2022, 16(2): 1358-1372

[42] Gu L, Cai J, Zeng D, et al. Energy efficient task allocation and energy scheduling in green energy powered edge computing. Future Generation Computer Systems, 2019, 95: 89-99

[43] Yang L, Jia J, Lin H, et al. Reliable dynamic service chain scheduling in 5G networks. IEEE Transactions on Mobile Computing, 2022, 22(8):4898-4911

[44] Zhao D, Zhou Z, Cai Z, et al. ASTL: Accumulative STL with a novel robustness metric for IoT service monitoring. IEEE Transactions on Mobile Computing, 2023, 22(10): 5751-5768

[45] Qu K, Zhuang W, Ye Q, et al. Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks. IEEE Transactions on Communications, 2020, 68(4): 2394-2408

[46] Labriji I, Meneghello F, Cecchinato D, et al. Mobility aware and dynamic migration of MEC services for the internet of vehicles. IEEE Transactions on Network and Service Management, 2021, 18(1): 570-584

[47] Zhang Q, Liu F, Zeng C. Online adaptive interference-aware VNF deployment and migration for 5G network slice. IEEE/ACM Transactions on Networking, 2021, 29(5): 2115-2128

[48] Ma L, Yi S, Carter N, et al. Efficient live migration of edge services leveraging container layered storage. IEEE Transactions on Mobile Computing, 2019, 18(9): 2020-2033

[49] Liu C, Tang F, Hu Y, et al. Distributed task migration optimization in MEC by extending multi-agent deep reinforcement learning approach. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(7): 1603-1614

[50] Li B, Cheng B, Liu X, et al. Joint resource optimization and delay-aware virtual network function migration in data center networks. IEEE Transactions on Network and Service Management, 2021, 18(3): 2960-2974

[51] Fu K, Zhang W, Chen Q, et al. Adaptive resource efficient microservice deployment in cloud-edge continuum. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(8): 1825-1840

[52] Zhu J, Yang R, Sun X, et al. QoS-aware co-scheduling for distributed long-running applications on shared clusters. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4818-4834

[53] Tang Z, Zhou X, Zhang F, et al. Migration modeling and learning algorithms for containers in fog computing. IEEE Transactions on Services Computing, 2019, 12(5): 712-725

[54] Chang J, Wang J, Li B, et al. Attention-based deep reinforcement learning for edge user allocation. IEEE Transactions on Network and Service Management, 2023, 54(3): 590-604

[55] Fan W, Yang F, Wang P, et al. DRL-based service function chain edge-to-edge and edge-to-cloud joint offloading in edge-cloud network. IEEE Transactions on Network and Service Management, 2023, 20(4): 4478-4493

[56] Afrin M, Jin J, Rahman A, et al. Dynamic task allocation for robotic edge system resilience using deep reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024, 54(3): 1438-1450

[57] Zhang Y, Hu J, Min G, et al. Joint charging scheduling and computation offloading in EV-assisted edge computing: A safe DRL approach. IEEE Transactions on Mobile Computing, 2024. DOI: 10.1109/TMC.2024.3355868

ZHOU Zhang-Bing, Ph. D., professor. His research interests include service computing, edge intelligence and the Internet of Things.

LI Xiao-Cui, Ph. D., assistant researcher. Her research interests include service computing, edge computing, anomaly detection, vertical generalization of large models and efficient reuse.

WANG Yu-Wei, Ph. D., senior engineer. His research interests include edge intelligence, federated learning, unmanned systems network collaboration, network function virtualization.

WANG Ya-Sha, Ph. D., professor, Changjiang Scholar. His research interests include deep learning, smart healthcare, vertical generalization and efficient reuse of large models.

Background

Service reconfiguration, as a promising research topic, has attracted much attention recently. Intuitively, it aims to reconfigure resources provided by IoT devices in edge networks, in order to accommodate more forthcoming applications with certain QoS constraints. Techniques have been developed to optimize the performance of IoT applications. However, they are inadequate with following challenges: (1) Inadequate for the reconfiguration of composition services. Current techniques focus mostly on the reconfiguration of atomic IoT services, but are inadequate for the reconfiguration of composition ones. They are inadequate for the reconfiguration of composite services. A reconfiguration strategy for satisfying on-running IoT applications as promised, while fulfilling forthcoming IoT applications as much as possible, is a challenge to be investigated. (2) Inefficiency of network resource utilization to support forthcoming IoT

applications. Most techniques optimally allocate available resources of IoT devices to satisfy tasks of current IoT applications, but they may hardly guarantee the configuration of an incoming task. Due to strict spatial and temporal constraints, this task may have to be configured to a certain IoT device which may have no enough remaining resources. This happens especially when edge networks are (partially) overloaded. Consequently, a service reconfiguration strategy is promising to accommodate more forthcoming IoT applications, and thus, to improve the resource utilization efficiency of edge networks significantly.

Service reconfiguration is of great significance to facilitate delay-sensitive IoT applications for the sustainable resource utilization of edge networks. This paper proposes E^2rC mechanism, to optimize the configuration of computational resources provided by IoT devices for satisfying complex requirements prescribed by IoT applications. This service reconfiguration is formulated as markov multi-phases decisions, which are solved through our enhanced Deep Reinforcement Learning (DRL) approach with a two-layer Q-network. Extensive experiments have been conducted, and evaluation results show that E^2rC is more efficient than the state-of-art counterparts in satisfying the delay constraints of IoT applications, while reducing the energy consumption and improving the resource utilization efficiency of IoT devices in edge networks.

This work is supported by the National Science Foundation of China (Nos. 42050103, 62372420).