2021 年 10 月 第 10 期
Vol. 42 No. 10 2021

物联网系统中边缘计算卸载决策问题: 建模、求解与分类

屠友鹏,陈海明,严林杰

(宁波大学 信息科学与工程学院,浙江 宁波 315211)
E-mail: chenhaiming@nbu.edu.cn

摘 要: 随着物联网的飞速发展,连接到互联网的终端设备数量不断增加,终端设备在处理计算密集型任务时可能面临着能力不足的问题,而将任务卸载到云平台上的方法难以满足延迟敏感型任务的需求. 因此在网络边缘处将计算量大的任务合理分配给计算资源充足的边缘服务器进行计算处理,再把计算完成的结果返回到终端,能有效的解决此类问题. 本文首先介绍了边缘计算和计算卸载的基本概念和度量指标,其次围绕计算卸载问题的建模方法、模型求解对目前提出的卸载决策问题进行阐述,并从最小化时延、最小化能耗、最小化系统成本这 3 个方面对比了现有卸载策略的优缺点. 最后提出了未来边缘计算中的卸载决策问题的研究挑战.

关 键 词: 边缘计算; 计算卸载; 卸载决策; 建模方法; 时延; 能耗
中图分类号: TP393 **文献标识码**: A **文 章 编 号**: 1000-1220(2021)10-2145-08

Offloading Decision Problems for Edge Computing in IoT Systems: Modeling, Solution and Classification

TU You-peng, CHEN Hai-ming, YAN Lin-jie
(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: With the rapid development of IoT, the number of end devices connected to the Internet is increasing, and the end devices may face the shortage problem of capacity in handling computation-intensive tasks, and the method of offloading tasks to the cloud platform can hardly meet the demand of latency-sensitive tasks. Therefore, reasonably allocating computationally intensive tasks at the edge of the network to edge servers with sufficient computing resources for computation processing, and then returning the completed computation results to the end devices, can effectively solve such problems. This paper first introduces the basic concepts and metrics of edge computing and computational offloading, followed by elaborating on the currently proposed offloading decision problem with respect to the modeling method and model solution of the computational offloading problem, and comparing the advantages and disadvantages of existing offloading strategies in three aspects, which are minimizing delay, minimizing energy consumption, and minimizing system cost. Finally, future research challenges for the offloading decision problem in edge computing are presented.

Key words: edge computing; computational offloading; offloading decisions; modeling methods; latency; energy consumption

1 引 言

随着物联网(IoT)技术不断发展,越来越多的终端设备连接到互联网中,给人们生活带来便捷的服务[1,2]. 根据思科年度互联网 2018-2023 白皮书报告,物联网已经成为一个普遍的系统,在这个系统中,人、物和数据与互联网彼此连接. 在全球范围内,M2M(Machine-To-Machine)连接将增长 2.4 倍,从 2018 年的 61 亿增长到 2023 年的 147 亿[3]. 终端设备的激增,必然带来数据的爆炸式产生,尽管当前移动设备的计算能力和存储容量在不断提升,但是在面对新型的密集型任务(如虚拟现实、自然语言处理、交互式游戏等)时,这些终端设备是无法胜任的,因此它们需要与具有更强计算和存储能力的设备相结合,以协作方式执行任务. 针对这些密集型任务,云计算作为一种集中式的计算模型被提出[4-6],将本地任务卸载到云中心进行处理,云中心强大的计算存储能力很好的解决了终端的资源匮乏问题[7]. 但由于云服务器位于核心网络层,地理位置上远离终端和用户,当任务卸载到云服务器时,任务数据须流经整个接入网和核心网,通过多个基站等网络设备进行传输. 计算结果同样也是经过长距离传输以后才能返回给终端设备,带来了额外的传输延时,同时也容易导致数据泄露等安全问题. 因此,这种以云为中心的架构解决方案难以满足低延迟、高可靠性和高安全性的应用需求[8]. 边缘计算(Edge Computing, EC)在这样的背景下应运而生[9,10].

2 边缘计算

2.1 边缘计算概述

图 1 展示了当前边缘计算网络的典型架构,在垂直方向上云-边-端 3 层协作计算,通过无线网和核心网络进行数据

收稿日期: 2021-03-22 收修改稿日期: 2021-04-25 基金项目: 浙江省自然科学基金项目(LY18F020011)资助; 宁波市自然科学基金项目(2018A610154)资助. 作者简介: **屠友鹏**,男,1992 年生,硕士研究生,CCF 会员,研究方向为边缘计算及任务的计算卸载; **陈海明**(通讯作者),男,1981 年生,博士,副教授,CCF 会员,研究方向为云边端协同计算; **严林杰**,男,1996 年生,硕士研究生,CCF 会员,研究方向为边缘计算及任务的计算卸载.

的传递,同一层中设备之间进行分布式水平协作计算.

云计算可以按需提供计算能力和服务资源,而不考虑计算的数据量、任务的容忍时延以及网络连接的质量.尽管边缘

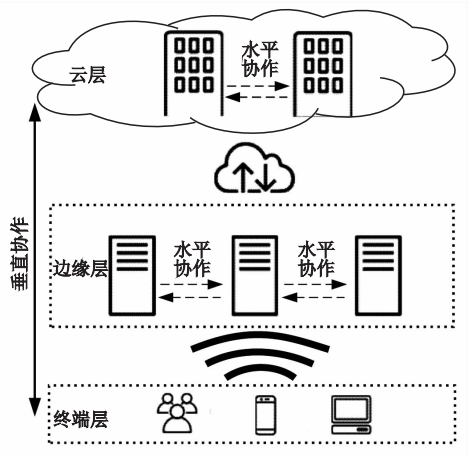

图 1 边缘计算网络架构图

Fig. 1 Basic architecture diagram of edge computing network

计算提供的服务资源有限,但相对于远程云计算,其可以使用分布在数据源和云中心的路径上具有计算、存储功能的设备来实现数据的预处理,并将预处理后的数据上传到云中心或把计算结果返回给移动终端设备.边缘计算相对于云计算而言,有以下优势:

实时性:通过边缘计算,将降低网络传输带宽负载和传输延迟.在一些需要实时反馈的应用场景中,如自动驾驶、智能制造、视频监控等位置感知领域,满足业务实时需求.

低能耗、低成本:在边缘计算中,减少了核心网络传输的数据量,降低了传输成本和带宽压力,降低了本地设备的能耗,提高了计算效率.

安全和保护隐私:边缘计算允许在数据源处理时过滤一些数据,减少了数据传输的总量,避免数据的转发和传输,也降低了数据被盗和隐私泄露的可能性,提高数据的安全性.表 1 给出了边缘计算和云计算的区别[11-13].

表 1 边缘计算和云计算的区别

Table 1 Difference between edge computing and cloud computing

对比属性	云计算	边缘计算
计算位置	远程网络中心	靠近用户
计算模型	集中式	分布式
计算能力	强大	中等
存储能力	强大	中等
相对执行延迟	较高	较低
设备间通信	带宽和网络限制	实时交互
处理任务安全性	较低	较高
开发维护成本	较高	较低

边缘计算技术和云计算相互补充,共同解决大数据时代的计算问题,满足智能物联系统在敏捷链接、实时业务、数据优化、应用智能、安全和隐私方面的关键需求,可应用于工业生产和服务消费等各个领域,已经成为当今的研究热点[14,15].

2.2 计算卸载简介

计算卸载作为边缘计算中最主要的研究内容之一,其通常被描述为在资源和计算能力有限的终端设备上,无法胜任资源密集型和延迟敏感型的任务,终端设备将其计算任务全部或部分交付给边缘服务器或者云计算中心进行任务计算,从而增强移动终端设备的任务处理能力,提高用户体验质量[16,17].计算卸载技术弥补了终端设备在计算能力、存储资源以及能效等方面的不足.当然边缘服务器的计算资源有限,在高负载时期无法满足大量的卸载请求时,也可以通过联合云中心进行 3 层卸载计算[18].此外,由于边缘协同应用的出现,多个用户在相同邻近位置时,任务卸载计算的结果可以被重复利用[19,20].

在计算卸载技术中,卸载决策需要考虑网络链路质量、终端设备性能、边缘服务器性能等因素的影响.具体问题包括:新生成的任务是否可以卸载;是部分卸载还是全部卸载;卸载到何处来执行计算.对于“是部分卸载还是全部卸载”的问题,根据任务是否可切分,决策出任务是全部都卸载至边缘服务器中执行,还是将部分任务卸载,剩下的留在本地执行.对于“卸载到何处来执行任务”的问题,需要考虑任务的性质,同一个应用的各个任务之间的关系可以是并行的也可以是连续串行的,需要根据计算任务之间的先后顺序关系,才能决定卸载策略.

3 卸载决策问题的建模

在本节中,介绍了卸载决策问题的建模方法和常用的度量指标.

3.1 决策变量

不同任务的应用程序可能有不同的性能需求,如表 2 所示为目前计算卸载的研究工作中常见的决策变量和度量指标.

表 2 计算卸载的决策变量与度量指标

Table 2 Decision variables and metrics that used in modeling the problem of computing offloading

决策变量	度量指标	文献
能耗	CPU 频率	[21-23]
	通信信道	[24-28]
	电池使用	[29,30]
时延	处理时延	[19,31,32]
	通信时延	[22,26,33,34]
	排队时延	[35,36]
服务质量	用户体验	[17,37]
	响应时间	[38,39]
	可靠性	[40,41]

3.1.1 能耗

能耗对于边缘计算架构的每一层来说都很重要,大多数终端设备性能在设计上受到电池的限制[42].一般而言,设备的功耗近似线性依赖于它的 CPU 负载,考虑到功耗和能耗之间的差异,能耗还取决于任务执行消耗的时间量,就系统整体能耗而言,将任务卸载到计算速率更快的设备是有益的.

3.1.2 时延

如前所述,边缘计算可被视为云计算在网络边缘的拓展,旨在为终端设备中具有低延迟要求的任务提供更低的时延,最小化时延的优化目标是边缘计算背后的主要驱动力[43,44].在任务卸载过程中,时延包括任务在移动终端和边缘服务器的处理时延,任务从终端卸载到边缘服务器的通信时延,有些

研究者还考虑了任务在边缘服务器等待处理的排队时延.

3.1.3　服务质量(QoS)

在物联网环境中,设备的移动性、异构性、不稳定性等特性给提供可靠的 QoS 带来挑战,在计算卸载中需要合理分配有限的通信和计算资源,最大化利用网络资源,保证用户体验不会受到很大影响[38]. QoS 的衡量标准包括异构环境下服务的可靠性、公平性以及快速响应的能力,Li 等人[39]提出一个统计计算模型和一个统计传输模型来量化统计服务质量与任务卸载策略之间的相关性.

3.2　决策建模

3.2.1　数学规划模型

数学规划模型是在一定约束下合理分配有限资源,从而达到目标期望的数学模型. 在计算卸载决策模型中一般分为整数规划(Integer programming,IP)模型和混合整数规划(Mimed integer programming,MIP)模型,前者要求规划中的决策变量全部限制为整数,当然还有一些研究者将该决策问题建模为非线性问题.

在物联网边缘计算中,任务生成过程是高度动态的,很难获得精确的统计信息[36],Gao 等人[25]研究了动态卸载和流量预测资源分配问题,通过将问题公式化为一个 IP 问题,目标是最小化系统中所有任务的平均功耗. Guo 等人[29]研究了超密集边缘计算中任务卸载、计算频率缩放和发射功率分配的联合优化问题,并给出了系统模型和数学公式,从而最小化移动设备的能量消耗及延长其电池寿命. Rui 等人[21]基于智能可穿戴设备通信网络的边缘卸载系统,在卸载之前,提出了一种多目标计算排序分段算法来划分要卸载的部分任务. 当作出卸载决定时,以最低的传输成本为目标建立相关模型,提出动态卸载策略. 当任务在本地处理时,优化 CPU 时钟频率,当任务被卸载时,自适应地分配传输功率. Mukherjee 等人[23]考虑边缘计算节点的异构性,在能耗和延迟约束的目标下,将最优的任务卸载问题转化为 MIP 问题. Cui 等人[35]使用排队论和优化算法对本地执行过程、传输过程和边缘服务器执行过程进行建模,将该问题建模为一个带约束的多目标优化问题. Vu 等人[45]为了找到移动用户的最优任务卸载方案,将联合任务卸载和资源分配问题转化为混合整数非线性规划(MINLP)问题. 与之类似,文献[46]提出了一种近似算法,应用分枝定界法迭代获得最优解. Hu 等人[27]考虑任务卸载中的功率分配问题,提出了一个基于次梯度的非合作博弈模型,并将卸载请求和能耗等资源的调度问题建模为一个混合整数非线性规划问题,求解最大化系统效益.

以文献[47]中建立的一个多任务多服务器的 MIP 任务卸载模型为例,首先网络模型定义如下: 边缘服务器的集合是 M, $M=\{EC_i \mid i=1,\cdots,M\}$,其中对于每一个边缘服务器,$Q_{MAX}$ 是其最大的可用计算资源,R_{MAX} 是其最大的可用存储资源,网络中要执行的所有任务的集合为 $T=\{T_j \mid j=1,\cdots,N\}$,对于每一个任务(考虑任务不可划分),当任务所需的计算资源大于其本地的最大可用资源时,就可以将任务卸载到边缘服务器上进行计算. 每个任务的卸载操作可以描述为 $T_j=\{A_j, Q_c^j, X^j, T_{MAX}^j\}$,$A_j$ 是系统首先做出的任务卸载决策,定义为 $A_j \in \{0,1\}$,0 表示任务在本地执行,1 表示任务被卸载到边缘服务器,Q_c^j 是执行所需的资源,X^j 是传输任务中涉及的数据量可以用来计算传输通信时延,T_{MAX}^j 是任务的最大容许延迟. 目标是设计一个高效的边端协作任务卸载方案,以确保在满足任务需求的同时降低总体能耗和延迟. 对于每一个任务而言,无论任务是本地执行还是边缘服务器执行将总能耗统一设置为 E_{total}^j,包括通信能耗、传输能耗、任务执行能耗. 并将总延迟设置为 D_{total}^j,包括通信延迟、传输延迟,任务执行延迟. 最小化成本任务卸载决策的联合优化多目标函数公式(1)如下:

$$\begin{aligned} \text{Min } S = & \min_{\forall T_j, \forall A_i} \left[\lambda E_{total}^j + (1-\lambda) D_{total}^j\right] \forall i \in M, \forall j \in T \\ \text{s.t.} \quad & C_1: D_{total}^j < T_{MAX}^j \\ & C_2: Q_c^j \leqslant Q_{MAX} \\ & C_3: A_j \in \{0,1\} \end{aligned} \tag{1}$$

λ 是用于实现延迟和能量消耗之间的动态折衷的权重参数,C_1-C_3 是式(1)的约束条件,其中 C_1 指出任务的执行时间必须在其容许延迟内,C_2 指出任务执行所需的资源不能大于服务器的最大可用资源,C_3 表示这项任务是由移动设备或边缘服务器来执行.

3.2.2　马尔可夫决策过程和强化学习模型

马尔可夫决策过程(Markov decision process,MDP)通常是指用状态转移概率描述连续不确定决策过程的数学模型,强化学习(Reinforcement Learning,RL)问题是序列决策问题,是智能体 Agent 通过与环境的交互不断学习,按照一定策略作出一系列决策以完成给定任务,使得奖励最大化得出最优策略. RL 通常有 4 个关键要素,包括行动空间 Action、状态空间 State、奖励 Reward、环境 Environment,研究者们大都将延迟和能耗的加权和作为优化目标,利用 MDP 和 RL 对多目标任务卸载的问题进行建模[48-50]. MDP 可以用四元组模型(　,　,P,R)表示,分别是状态集、动作集、状态转移概率和奖励. 在决策目标时,通常选择合适的动作使得长期获得的奖励最大化.

文献[51]将每个物联网用户视为一个智能体,网络中的其他一切组成环境. 从图 2 中可见,每个物联网用户 u_i 在时隙 k 从状态空间 S 中观察状态 s^k,然后,从动作空间 A 采取动作 a^k,通过选择不同策略来决定是否卸载任务. 然后环境改变观察到新的状态 s^{k+1},并获得奖励 R^k,新状态再次选择动作,不断重复直到所有任务完成.

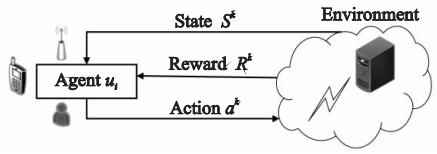

图 2　强化学习计算卸载模型

Fig. 2　Reinforcement learning computational offloading model

1) 行动空间 Action

在时隙 k,每个智能体的动作 a^k 通常表述为一个二进制决策,用 0 表示该任务在本地执行,用 1 表示该任务卸载到边缘服务器上执行. 动作一开始是智能体随机探索的,最后随着与环境的交互逐渐确定下来.

2) 状态空间 State

状态是智能体在时隙 k 中对环境的探测信息,可使用信道增益 g^k、待处理任务队列 t^k 及剩余计算能力 r^k 等来表示,

即 $s^k=\{g^k,t^k,r^k\}$.

3) 奖励 Reward

系统成本表述为模型中的能耗和延迟的加权和. 为了在计算卸载和资源分配方面做出正确的决策,采用负奖励来最小化系统成本. 因此,每个智能体在时隙 k 计算卸载策略的奖励函数由公式(2)给出定义:

$$R^k=\begin{cases}C_d^k & \text{if local}\\ C_e^k & \text{if edge},1-p\\ \delta & \text{if edge},p\end{cases} \tag{2}$$

其中 C_d^k 表示在本地执行的成本,C_e^k 表示在边缘服务器上执行的成本,p 是在边缘计算模式下任务传输的失败率(考虑到任务传输失败的可能性),δ 是任务执行失败时的惩罚函数. 此外,状态转移可以建模为 MDP,其中状态转移概率和成本仅取决于环境和获得的策略,转移概率 $P=(s^{k+1},R^k\mid s^k,a^k)$ 被定义为执行动作 a^k 从状态 s^k 到状态 s^{k+1} 所获得的奖励 R^k 的概率. $V(s,\pi)$ 用来评价状态的好坏,表示智能体与环境交互结束后获得的累积奖励,具体定义由公式(3)给出:

$$V(s,\pi)=\mathbf{E}_\pi\left[\sum_{k=1}^{K}\gamma^k R^k\right] \tag{3}$$

其中 s 为状态,π 为策略,$\gamma\in[0,1]$ 为折扣因子.

4) 环境 Environment

智能体观测到一些环境的信息,如当前可使用的信道数量、边缘服务器剩余的计算能力和存储能力. 智能体需要不断探索并与环境交互,使 $V(s,\pi)$ 最小化. 这意味着对于任何给定的状态 s,可以通过公式(4)获得最优策略:

$$\pi^*=\arg\min_{\pi}V(s,\pi)\quad\forall s\in S \tag{4}$$

文献[52]中针对任务的可部分卸载进行建模,将移动设备的任务卸载问题建模为具有时隙为 τ 的动态马尔可夫决策过程. 令 $=\{f^e,B,T\}$ 表示边缘计算系统的状态集,其中 f^e 表示边缘服务器的计算能力,B 表示基站剩余无线通信资源大小,T 表示任务请求信息. $=\{a_i(\eta),a_i(b),a_i(f^e)\}$ 表示边缘计算系统的动作集,其中 $a_i(\eta)\in[0,1]$ 表示任务的卸载率,$a_i(b)$ 表示任务传输所需要的带宽资源,$a_i(f^e)\in[0,1]$ 表示边缘服务器给任务提供的计算资源比例. 集合 $\boldsymbol{A}_\tau^s$ 是决策空间 的子集,表示在状态 $\boldsymbol{S}_\tau\in$ 下的可选动作集合. 在决策动作 $\boldsymbol{A}_\tau$ 下状态从 S_τ 转移为 $S_{\tau+1}$ 的转移概率表示为 $P(S_{\tau+1}\mid S_\tau,A_\tau)$. 该动作由策略 $\pi:S\to A$ 给出,该策略是通过资源分配算法获得的. 在当前状态 S_τ 执行决策 A_τ 之后,得到的奖励 $R_\tau(S_\tau,A_\tau)$ 是从环境中获取的. 基于动态马尔科夫决策过程建模的优化目标函数表示为公式(5):

$$\begin{aligned}&\max_{A_\tau}\mathbf{E}\left[\sum_{\tau=0}^{\infty}R_\tau(\boldsymbol{S}_\tau,\boldsymbol{A}_\tau)\right]\\ &\text{s.t. } C_1:\ \forall\tau,\forall a_i(\eta)\in[0,1]\\ &C_2:\ \forall\tau,\sum_{i=1}^{n}a_i(f^e)\leqslant 1\\ &C_3:\ \forall\tau,\sum_{i}^{n}a_i(b)\leqslant B\end{aligned} \tag{5}$$

其中,C_1 意味着任务卸载率应≤1. C_2 表示 n 个移动设备请求的任务分配的计算资源总和应小于边缘服务器 e 的总计算资源. 此外,C_3 表示分配给 n 个移动设备的无线通信带宽资源之和应小于基站的总带宽 B.

Li 等人[53]针对异构边缘服务器的计算卸载问题,提出了一种深度强化学习(DRL)算法. 该算法考虑了多任务、边缘子网的异构性和边缘设备的可移动性,探索网络环境并生成计算卸载决策,采用 Actor-Critic 算法和深度确定性策略梯度优化计算卸载决策,使任务延迟最小化. 与此类似,Yan 等人[22]提出了一个基于 Actor-Critic 网络结构的深度强化学习框架,Actor 网络利用深度神经网络(DNN)来学习从输入状态(无线信道增益和边缘中央处理器频率)到每个任务的二进制卸载策略的最佳映射. 同时,Critic 网络在给定卸载决策的情况下,快速评估由 Actor 网络输出的决策的成本,然后选择当前最佳卸载动作,并将状态-动作对(state-action)放入经验池存储器中作为训练数据集,以持续改进 DNN. Alfakih 等人[24]基于 RL 以最小化系统成本为目标对边缘服务器中的资源管理问题进行建模,并提出了一个基于信息学习的状态-动作-奖励-状态-动作(RL-SARSA)算法来解决该问题. Lei 等人[54]提出了一种在物联网边缘计算系统中的联合计算卸载和多用户调度算法,该算法在随机流量到达的情况下,最小化延迟与功耗的长期加权和. 将动态优化问题转化为连续时间马尔可夫决策过程(CTMDP)模型,利用半梯度下降法和时间差分算法线性逼近值函数,导出了求解 CTMDP 模型的简单算法. Zhan 等人[31]针对车联网计算卸载调度问题,在任务延迟和能耗之间进行权衡,通过设计 MDP 对其建模,基于策略优化算法实现 DRL,利用网络结构共享参数结合卷积神经网络逼近策略和价值函数,对状态和奖励进行了一系列调整,以处理庞大的状态空间,进一步提高卸载效率.

4 卸载决策问题的求解

边缘计算系统中设备的异构性及其动态特性导致了系统模型高度复杂,通过设计得到最优的解决方案几乎是不可能,所以只能使用适当的方法来对决策模型进行求解,以此来确定最佳的解决方案. 卸载决策模型通常根据其涉及云-边-端 3 层模型中的哪些层进行分类. 例如,Jindal 等人[55,56]只涉及到云层,此时只是纯云层模型的优化求解问题[57],Al-Turjman 等人[58]仅涉及到终端设备层的优化. 因此,能算的上边缘卸载决策模型至少涉及 2 层,一般这些可被分为以下 3 类:

1) 涉及云层和边缘层的卸载决策模型. 可以根据资源容量和任务延迟限制来优化云和边缘资源的总体能耗. 与分布式云计算有些相似之处,而存在的差异是边缘资源的数量比分布式云的数量高几个数量级.

2) 涉及边缘层和终端层的卸载决策模型. 具有边缘资源和终端设备的协作是典型的边缘计算问题,并且由于终端设备资源的有限,模型求解方法在这种情况下起着至关重要的作用. 少数研究者考虑的是单个边缘资源与它服务的多个终端设备,大多数研究者考虑多个边缘资源以及它们服务的多个终端设备[59],后者可以平衡多个边缘资源进行协作,从而达到更好的卸载目标,但会导致更复杂的优化求解问题.

3) 云-边-端 3 层同时考虑的卸载决策模型,层与层之间的错综复杂和设备的动态特性,导致这种卸载决策模型与涉及所有资源的决策变量求解的计算复杂性更大. 许多模型可

能只将问题抽象为一个关注云、边缘资源和终端设备的单一优化问题. 在每层中,针对数据、代码、任务或这些组合来进行模型的求解.

以上 3 类卸载决策问题可采用以下方法求解: 1) 将卸载问题表述成凸优化问题,对于凸优化问题来说,局部最优解就是全局最优解. 有时卸载决策无法归结为凸优化问题,就需要使用松弛算法,将非凸限制条件松弛为凸限制条件,降低问题的求解难度; 2) 使用启发式算法,基于直观或经验构造的算法,在可接受的代价约束下给出待解决组合优化问题的一个可行解,以有限的计算量解决难题,找到潜在的次优解; 3) 使用动态的分布式卸载算法,利用分布式博弈机制并结合李雅普诺夫优化理论,设计一种资源的动态报价机制,并对不同业务类型差异化控制和计算资源的弹性按需分配,以求解任务卸载及资源分配问题; 4) 建立强化学习模型,当状态数量有限时,使用 on-policy 策略的 SARSA 算法或者 off-policy 策略的 Q-learning 算法进行求解,当面对数不清的状态环境时,使用基于深度神经网络的 DQN 求解.

表 3 通过区分卸载决策模型涉及到的层次、以及文献中所提的求解优化方法和实验所得到的效果,对当前一些学者的研究进行对比.

表 3　计算卸载决策模型的求解

Table 3　Calculation of the solution of the offloading decision model

文献	涉及层	建模方法	求解优化方法	效果
[46]	云边	混合整数规划	应用分枝定界法迭代获得最优解	降低总成本 34%
[47]		整数规划	基于 SDN 技术的编排数据服务机制	降低延迟 73%、能耗 10 %
[10]	边端	混合整数规划	一种启发式的传输时延最小化算法	证明方案有效性
[22]		强化学习模型	利用 DNN 来学习输入状态	算法性能提升 99.5%
[24]		强化学习模型	基于强化学习 SARSA 算法优化	性能优于 RL-QL
[54]		MDP 建模	时间差分算法以及半梯度下降法	显著提升性能
[60]		整数规划	基于决策后在线学习确定性算法	优于传统方案
[61]		混合整数规划	基于 Lyapunov 优化算法	性能优于基线算法
[18]	云边端	其他模型	基于博弈论的分布式卸载方法	优于传统方案
[62]		整数规划	基于模拟退火的候鸟优化算法	高于同类算法

5　卸载决策方法的分类

根据任务类型和卸载需求,一般将卸载决策分为最小化时延、最小化能耗、最小化系统成本 3 类. 最小化时延卸载策略就是完成任务执行计算的时间要求最低; 最小化能耗卸载策略是在任务执行的过程中,在满足一定时延的要求下,使得整个系统的能耗最小; 最小化系统成本一般是权衡时延和能耗,试图在能耗和延迟之间找到一个平衡点,以满足不同物联网应用的用户需求.

5.1　最小化时延卸载决策方法

Shu 等人[63] 研究了低功耗物联网系统边缘计算中的细粒度任务卸载问题. 通过考虑物联网任务的调度、边缘服务器上的异构资源、多址接入边缘网络中的无线干扰,提出了一种轻量而有效的多用户边缘系统卸载方案,将最合适的物联网任务、子任务卸载到边缘服务器,从而使执行时间最小化. 文献[63]为单用户边缘服务器提出了最早完成时间卸载算法(EFO),以决定需要卸载哪些子任务来获得更多的性能增益. 然后进一步将 EFO 算法扩展到具有异构服务器的多用户系统,以协调多用户之间的通信和计算资源竞争. 此外,为了提高卸载决策的效率,设计了一种能够达到纳什均衡的分布式计算卸载算法,并由模拟实验结果表明,所提出的卸载算法能降低 81.6% 端到端任务的执行时间,有效提高边缘服务器的资源利用率.

Guo 等人[19] 考虑多个移动用户将重复的计算任务卸载到网络边缘服务器,并考虑这些服务器之间共享计算结果的可能,设计最优的细粒度协作卸载策略,增强数据缓存以最小化移动终端的整体任务执行延迟. 为了降低部署问题的复杂性,将任务定义为一个多标签分类问题,并使用了深度监督学习方法来实现快速卸载决策,同时最小化计算量和卸载开销. Li 等人[53] 提出的算法也考虑了多任务、边缘子网的异构性和边缘设备的可移动性,通过与网络环境交互并生成计算卸载决策,使任务延迟最小化.

Xu 等人[10] 基于深度学习的边缘服务卸载框架建立在集中式单元-分布式单元架构上,通过估计源-目的边缘节点距离和启发式搜索目的虚拟机,提出一种启发式的传输时延最小化算法. Wu 等人[64] 考虑信息泄露的问题,制定了保密规定、计算卸载和无线电资源分配(包括时间和功率分配)的联合优化,提出了一个有效的算法来寻找设备的最佳选择,最大限度地减少系统完成计算任务的整体延迟.

5.2　最小化能耗卸载决策方法

允许移动设备将延迟受限的计算任务卸载到网络边缘服务器,任务到每个边缘基站的传输功率和卸载传输数据量的消耗被定义为优化最小化移动设备的总能量消耗[65].

Vu 等人[45] 提出了一种使边缘节点能够共享计算资源和无线电资源的边缘计算网络体系结构,以最小化移动用户的总能耗,同时满足任务的延迟要求. Sun 等人[66] 考虑多个边缘服务器且每个服务器资源有限的场景,通过为每个用户设备选择最佳的服务器卸载,从而最大化卸载任务的数量,最小化所有用户设备和边缘服务器的能耗. Yousafzai 等人[30] 分析了平台相关的本地应用对边缘网络中计算卸载的影响,并提出了一个基于轻量级进程迁移的计算卸载框架,实验测试结果表明,该框架节省了 44% 的执行时间和 84% 的能耗.

Guo 等人[29] 考虑具有多台边缘服务器的超密集边缘计算场景,以及计算任务在移动终端设备上的随机生成和任务在边缘服务器上的随机到达,对任务卸载、计算频率缩放和发射功率分配进行了联合优化,以最小化移动设备的能量消耗.

Zhao 等人[67]发现在高资源消耗和高通信成本之间很难实现平衡,提出了一种局部计算卸载方法,通过联合优化任务的卸载率、系统的 CPU 速度、可用信道的分配带宽和每个系统在每个时隙的传输功率,使系统和边缘服务器消耗的总能耗最小. Yu 等人[68]以最小化所有移动设备完成多个任务的能耗为目标,提出了一种多设备多服务器系统联合计算传输功率和计算速度分配的卸载算法,以满足应用的实时性. Hui 等人[69]提出了一种基于实时强化学习的卸载方案,通过检查 3 种系统状态(系统利用率、动态时差和信道增益) ,做出行动决策(降低 CPU 频率和任务卸载) ,并通过来自环境的反馈进一步学习,在周期结束时,周期中消耗的能量返回给调度器,以指示选择最小能耗的策略.

5.3 最小化系统成本卸载决策方法

Yang 等人[26]提出了一种面向海上通信网络的边缘计算框架,实现两阶段联合优化卸载算法,在有限能量和延迟敏感约束下优化计算和通信资源分配. 在第 1 阶段,海事用户考虑他们的需求和环境,决定是否卸载计算. 在第 2 阶段,考虑延迟和能量消耗的动态权衡,通过信道分配和功率分配,优化与中心云服务器相协调的卸载策略. Yuan 等人[62]提出了一个云边端计算系统,该系统包括终端层、边缘层和云层. 在此基础上,设计了一种利润最大化的协同计算卸载和资源分配算法,最小化系统成本,并保证任务的响应时间得到满足. Guo 等人[70]采用纳什均衡分布式计算卸载算法,其中计算卸载决策依赖于一定的网络信道参数. 数值结果表明,当应用程序选择有效的信道传输数据,将任务卸载到边缘服务器时,该算法在最小化系统成本限制时具有优越的性能.

考虑到潜在的移动设备在具有多个无线边缘服务器的网络空间中不均匀分布,允许边缘服务器将任务进一步卸载到附近的边缘服务器. 缓解回程网络和骨干网的拥塞,将任务路由与卸载优化、卸载决策、传输功率控制和计算服务器选择结合在一起. Thai 等人[46]建立了以最小化系统计算工作量和通信容量为目标的成本优化问题,提出了一种提供垂直和水平卸载的通用边缘计算架构. 实验结果表明,与只支持垂直卸载的传统设计相比,这种云边缘计算架构可以显著降低系统总成本约 34%. Yan 等人[22]联合确定每个任务的卸载决策和资源分配,共同优化卸载决策和本地中央处理器的频率,从而最小化移动设备的能量、时间成本. Chen 等人[36]提出了一种动态计算卸载算法实现卸载成本和性能之间的权衡,该算法将优化问题分解为一系列子问题,并以在线和分布式方式并行求解这些子问题,来进一步减少延迟和提高服务质量.

Huang 等人[47]基于软件定义网络(SDN) 技术,提出了一种基于服务编排的云-边计算协同任务卸载方案,一方面将收集到的元数据由边缘服务器作为高质量的服务进行编排,降低了将资源上传到云所带来的网络负载,另一方面数据处理尽可能在边缘层完成,实现了负载均衡,也降低了数据泄露的风险. 实验结果表明,与基于随机决策的任务卸载方案和基于最大缓存的任务卸载方案相比,方案的延迟降低了约 73.82%-74.34%,能耗降低了 10.71%-13.73%.

表 4 计算卸载决策分类

Table 4 Computational offloading decision classification

任务类型	卸载需求	参考文献	建模方法	求解方法	优势	不足
时间敏感型	最小化时延	[10]	混合整数规划	启发式搜索	卸载过程中任务时延最小	仅考虑了延迟最小,忽略了移动端的能量消耗
		[19]	整数规划	博弈论		
		[53]	强化学习模型	策略梯度		
		[71]	其它模型	分布式算法		
		[72]	混合整数规划	迭代算法		
		[73]	整数规划	Lyapunov 算法		
能量消耗型	最小化能耗	[27]	混合整数规划	非合作博弈论	卸载过程中系统能耗最少	仅考虑能耗最少,忽略了任务所能忍受的时间限制
		[29]	其它模型	博弈论		
		[30]	其它模型	基于流程框架		
		[45]	混合整数规划	分支定界算法		
		[66]	混合整数规划	最大流问题		
		[67]	混合整数规划	离散化近似算法		
		[68]	整数规划	分布式算法		
成本需求型	最小化成本	[21]	整数规划	分段排序算法	卸载过程中总花费最小	适用范围有限,针对特定要求的任务不适用
		[22]	强化学习模型	DNN 算法		
		[23]	混合整数规划	成本最大流		
		[31]	MDP 建模	卷积神经网络		
		[47]	混合整数规划	SDN 技术编排		
		[51]	强化学习模型	深度神经网络		
		[62]	整数规划	启发式搜索		
		[74]	混合整数模型	多算法组合		
		[75]	混合整数模型	改进搜索算法		

表 4 从任务类型、卸载需求、建模方法、求解方法的角度,对以上提到的计算卸载决策研究的优势和不足进行了分类总结.

6 总 结

边缘计算正在成为在各种应用场景中确保降低延迟、减

少带宽使用和保护数据隐私的重要技术. 计算卸载决策涉及云-边-端协作环境中多设备之间的资源管理和分配. 本文对边缘计算中的计算卸载技术进行了深入的调查. 总结了计算卸载的在边缘计算中的重要作用. 综述了计算卸载的建模方法、模型求解方法以及计算卸载决策的分类. 本文中做出的工作不仅能帮助研究者快速了解计算卸载的概念,而且可以帮助工业实践者们掌握计算卸载的最新技术,以设计更好的计算卸载决策算法和资源管理系统.

References:

[1] Shi Wei-song, Sun Hui, Cao Jie, et al. Edge computing-an emerging computing model for the internet of everything era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.

[2] Miraz M, Ali M, Excell P, et al. Internet of Nano-things, things and everything: future growth trends [J]. Future Internet, 2018, 10(8): 68-81.

[3] Cisco. Cisco annual internet report(2018-2023) white paper [EB/OL]. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2020-03-09.

[4] Dale P J. Introduction to cloud computing [J]. Journal of Electronic Resources in Medical Libraries, 2011, 8(4): 449-458.

[5] Ashrafi T H, Hossain M A, Arefin S E, et al. IoT infrastructure: fog computing surpasses cloud computing [J]. Intelligent Communication and Computational Technologies, Springer, Singapore, 2018: 43-55, doi: 10.1007/978-981-10-5523-2_5.

[6] Zyane A, Bahiri M N, Ghammaz A. IoTScal-H: hybrid monitoring solution based on cloud computing for autonomic middleware-level scalability management within IoT systems and different SLA traffic requirements [J]. International Journal of Communication Systems, 2020, 33(14): 4495-4518.

[7] Huang Jun, Yin Ying, Yan Hui-fang, et al. Context-aware resource allocation for device-to-device communications in cloud-centric internet of things [J]. Journal of Chongqing University of Posts Telecommunications, 2015, 27(4): 484-492.

[8] Pan Jian-li, Mcelhannon J. Future edge cloud and edge computing for internet of things applications [J]. IEEE Internet of Things Journal, 2017, 5(1): 439-449.

[9] Cao Ke-yan, Liu Ye-fan, Meng Gong-jie, et al. An overview on edge computing research [J]. IEEE Access, 2020, 8: 85714-85728, doi: 10.1109/ACCESS.2020.2991734.

[10] Xu Xiao-long, Li Dao-ming, Dai Zhong-hui, et al. A heuristic offloading method for deep learning edge services in 5G networks [J]. IEEE Access, 2019, 7: 67734-67744, doi: 10.1109/ACCESS.2019.2918585.

[11] Taleb T, Samdanis K, Mada B, et al. On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration [J]. IEEE Communications Surveys, 2017, 19(3): 1657-1681.

[12] Dolui K, Datta S K. Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing [C]//2017 Global Internet of Things Summit(GIoTS), Geneva, 2017: 1-6.

[13] Yu Wei, Liang Fan, He Xiao-fei, et al. A survey on the edge computing for the internet of things [J]. IEEE Access, 2017, 6: 6900-6919, doi: 10.1109/ACCESS.2017.2778504.

[14] Ahmed E, Rehmani M H. Mobile edge computing: opportunities, solutions, and challenges [J]. Future Generation Computer Systems, 2016, 70(5): 59-63.

[15] Cheng Xiao-lan, Zhou Xin, Jiang Cong-feng, et al. Towards computation offloading in edge computing: a survey [J]. IEEE Access, 2019, 7: 131543-131558, doi: 10.1109/ACCESS.2019.2938660.

[16] Wang Shang-guang, Zhao Ya-li, Huang Lin, et al. QoS prediction for service recommendations in mobile edge computing [J]. Journal of Parallel, 2017, 127(5): 134-144.

[17] Noreikis M, Xiao Yu, Ylä-Jaäiski A. QoS-oriented capacity planning for edge computing [C]//IEEE International Conference on Communications, Paris, 2017: 1-6.

[18] Wu Bin-wei, Zeng Jie, Ge Lu, et al. Energy-latency aware offloading for hierarchical mobile edge computing [J]. IEEE Access, 2019, 7: 121982-121997, doi: 10.1109/ACCESS.2019.2938186.

[19] Guo Hong-zhi, Liu Jia-jia. Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks [J]. IEEE Transactions on Vehicular Technology, 2018, 67(5): 4514-4526.

[20] Abbas N, Yan Zhang, Taherkordi A, et al. Mobile edge computing: a survey [J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.

[21] Rui Lan-lan, Yang Ying-tai, Gao Zhi-peng, et al. Computation offloading in a mobile edge communication network: a joint transmission delay and energy consumption dynamic awareness mechanism [J]. IEEE Internet of Things Journal, 2019, 6(6): 10546-10559.

[22] Yan Jia, Bi Su-zhi, Huang Liang, et al. Deep reinforcement learning based offloading for mobile edge computing with general task graph [C]//IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020: 1-7.

[23] Mukherjee M, Kumar V, Kumar S, et al. Computation offloading strategy in heterogeneous fog computing with energy and delay constraints [C]//IEEE International Conference on Communications(ICC), 2020: 1-5.

[24] Alfakih T, Hassan M M, Gumaei A, et al. Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA [J]. IEEE Access, 2020, 8: 54074-54084, doi: 10.1109/ACCESS.2020.2981434.

[25] Gao Xin, Huang Xi, Bian Si-meng, et al. PORA: predictive offloading and resource allocation in dynamic fog computing systems [J]. IEEE Internet of Things Journal, 2020, 7(1): 72-87.

[26] Yang Ting-ting, Feng Hai-long, Gao Shan, et al. Two-stage offloading optimization for energy-latency tradeoff with mobile edge computing in maritime internet of things [J]. IEEE Internet of Things Journal, 2020, 7(7): 5954-5963.

[27] Hu Shi-hong, Li Guang-hui. Dynamic request scheduling optimization in mobile edge computing for IoT applications [J]. IEEE Internet of Things Journal, 2020, 7(2): 1426-1437.

[28] Zhang Qi, Gui Lin, Hou Fen, et al. Dynamic task offloading and resource allocation for mobile edge computing in dense cloud RAN [J]. IEEE Internet of Things Journal, 2020, 7(4): 3282-3299.

[29] Guo Hong-zhi, Zhang Jie, Liu Jia-jia, et al. Energy-efficient task offloading and transmit power allocation for ultra-dense edge computing [C]//IEEE Global Communications Conference(GLOBECOM), Abu Dhabi, 2019: 1-6.

[30] Yousafzai A, Yaqoob I, Imran M, et al. Process migration-based computational offloading framework for IoT-supported mobile edge/cloud computing [J]. IEEE Internet of Things Journal, 2020, 7(5): 4171-4182.

[31] Zhan Wen-han, Luo Chun-bo, Wang Jin, et al. Deep reinforcement learning-based offloading scheduling for vehicular edge computing [J]. IEEE Internet of Things Journal, 2020, 7(6): 5449-5465.

[32] Zhang Daniel, Ma Yue, Zheng Chao, et al. Cooperative-competitive task allocation in edge computing for delay-sensitive social sensing [C]//ACM/IEEE Symposium on Edge Computing, Seattle, 2018: 243-259.

[33] Wang Nan, Varghese B, Matthaiou M, et al. ENORM: a framework for edge node resource management [J]. IEEE Transactions on Services Computing, 2017, 13(6): 1086-1099.

[34] Cozzolino V, Ott J, Ding A Y, et al. ECCO: edge-cloud chaining and orchestration framework for road context assessment [C]//IEEE/ACM 5th International Conference on Internet-of-Things Design and Implementation(IoTDI), Sydney, Australia, 2020: 223-230.

[35] Cui Lai-zhong, Chong Xu, Shu Yang, et al. Joint optimization of energy consumption and latency in mobile edge computing for internet of things [J]. IEEE Internet of Things Journal, 2018, 6(3): 4791-4803.

[36] Chen Ying, Zhang Ning, Zhang Yong-chao, et al. Dynamic computation offloading in edge computing for internet of things [J]. IEEE Internet of Things Journal, 2019, 6(3): 4242-4251.

[37] Hong Zi-cong, Huang Hua-wei, Guo Song, et al. QoS-aware cooperative computation offloading for robot swarms in cloud robotics [J]. IEEE Transactions on Vehicular Technology, 2019, 68(4): 4027-4041.

[38] Song Yao-zhong, Yau S S, Yu Ruo-zhou, et al. An approach to QoS-based task distribution in edge computing networks for IoT applications [C]//IEEE International Conference on Edge Computing, Honolulu, 2017: 32-39.
[39] Li Qing, Wang Shang-guang, Zhou Ao, et al. QoS driven task offloading with statistical guarantee in mobile edge computing [J]. IEEE Transactions on Mobile Computing, 2020, PP(99): 1-1.
[40] Rausch T, Nastic S, Dustdar S. EMMA: distributed QoS-aware MQTT middleware for edge computing applications [C]//IEEE International Conference on Cloud Engineering(IC2E), Orlando, 2018: 191-197.
[41] Lai Phu, He Qiang, Cui Guang-ming, et al. QoE-aware user allocation in edge computing systems with dynamic QoS [J]. Future Generation Computer Systems, 2020, 112(11): 684-694.
[42] Zhang Ke, Mao Yu-ming, Leng Su-peng, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks [J]. IEEE Access, 2017, 4: 5896-5907, doi: 10.1109/ACCESS.2016.2597169.
[43] Beraldi R, Mtibaa A, Alnuweiri H. Cooperative load balancing scheme for edge computing resources [C]//Second International Conference on Fog & Mobile Edge Computing, Valencia, Spain, 2017: 94-100.
[44] Yuan Quan, Zhou Hai-bo, Li Jing-lin, et al. Toward efficient content delivery for automated driving services: an edge computing solution [J]. IEEE Network, 2018, 32(1): 80-86.
[45] Vu T T, Huynh N V, Hoang D T, et al. Offloading energy efficiency with delay constraint for cooperative mobile edge computing networks [C]//IEEE Global Communications Conference(GLOBECOM), Abu Dhabi, 2019.
[46] Thai M T, Lin Y D, Lai Y C, et al. Workload and capacity optimization for cloud-edge computing systems with vertical and horizontal offloading [J]. IEEE Transactions on Network, 2020, 17(1): 227-238.
[47] Huang Ming-feng, Liu Wei, Wang Tian, et al. A cloud-MEC collaborative task offloading scheme with service orchestration [J]. IEEE Internet of Things Journal, 2019, 7(7): 5792-5805.
[48] Yan Pei-zhi, Choudhury S. Optimizing mobile edge computing multi-level task offloading via deep reinforcement learning [C]//IEEE International Conference on Communications(ICC), 2020: 1-7.
[49] Zhao Lei, Wang Jia-dai, Liu Jia-jia, et al. Routing for crowd management in smart cities: a deep reinforcement learning perspective [J]. IEEE Communications Magazine, 2019, 57(4): 88-93.
[50] Huang Liang, Bi Su-zhi, Zhang Y J A. Deep reinforcement learning for online offloading in wireless powered mobile-edge computing networks [J]. IEEE Transactions on Mobile Computing, 2018, 19(11): 2581-2593.
[51] Liu Xiao-lan, Yu Jia-dong, Wang Jian, et al. Resource allocation with edge computing in IoT networks via machine learning [J]. IEEE Internet of Things Journal, 2020, 7(4): 3415-3426.
[52] Chen Si-yu, Wang Qi, Chen Jie-nan, et al. An intelligent task offloading algorithm(iTOA) for UAV network [C]//IEEE Globecom Workshops(GC Wkshps), 2019: 1-6.
[53] Li Yun-zhao, Qi Feng, Wang Zhi-li, et al. Distributed edge computing offloading algorithm based on deep reinforcement learning [J]. IEEE Access, 2020, 8: 85204-85215, doi: 10.1109/ACCESS.2020.2991773.
[54] Lei Lei, Xu Hui-juan, Xiong Xiong, et al. Joint computation offloading and multi-user scheduling using approximate dynamic programming in NB-IoT edge computing system [J]. IEEE Internet of Things Journal, 2019, 6(3): 5345-5362.
[55] Kaur A, Singh V P, Singh G S. The future of cloud computing: opportunities, challenges and research trends [C]//International conference on IoT in Social, Mobile, Analytics and Cloud, Palladam, India, 2018: 213-219.
[56] Jindal R, Kumar N, Nirwan H. MTFCT: a task offloading approach for fog computing and cloud computing [C]//10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020.
[57] Al-Dhuraibi Y, Paraiso F, Djarallah N, et al. Elasticity in cloud computing: state of the art and research challenges [J]. IEEE Transactions on Services Computing, 2018, 11(99): 430-447.
[58] Al-Turjman F. Energy-aware data delivery framework for safety-oriented mobile IoT [J]. IEEE Sensors Journal, 2018, 18(1): 470-478.
[59] Cao Xiao-feng, Tang Guo-ming, Guo De-ke, et al. Edge federation: towards an integrated service provisioning model [J]. IEEE/ACM Transactions on Networking, 2019, 28(3): 1116-1129.
[60] Sehati A, Ghaderi M. Energy-delay tradeoff for request bundling on smartphones [C]//IEEE Conference on Computer Communications, 2017: 1-9.
[61] Du Wei, Lei Tao, He Qiang, et al. Service capacity enhanced task offloading and resource allocation in multi-server edge computing environment [C]//IEEE International Conference on Web Services(ICWS), Milan, 2019: 83-90.
[62] Yuan Hai-tao, Zhou Meng-chu. Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems [J]. IEEE Transactions on Automation ence, 2020, PP(99): 1-11.
[63] Shu Chang, Zhao Zhi-wei, Han Yun-peng, et al. Multi-user offloading for edge computing networks: a dependency-aware and latency-optimal approach [J]. IEEE Internet of Things Journal, 2019, 7(3): 1678-1689.
[64] Wu Yuan, Shi Jia-jun, Ni Ke-jie, et al. Secrecy-based delay-aware computation offloading via mobile edge computing for internet of things [J]. IEEE Internet of Things Journal, 2019, 6(3): 4201-4213.
[65] Gu Qi, Wang Gong-pu, Liu Jing-xian, et al. Optimal offloading with Non-orthogonal multiple access in mobile edge computing [C]//IEEE Global Communications Conference(GLOBECOM), Abu Dhabi, 2019.
[66] Sun Hai-jian, Zhou Fu-hui, Hu R Q. Joint offloading and computation energy efficiency maximization in a mobile edge computing system [J]. IEEE Transactions on Vehicular Technology, 2019, 68(3): 3052-3056.
[67] Zhao Tian-chu, Zhou Sheng, Song Lin-qi, et al. Energy-optimal and delay-bounded computation offloading in mobile edge computing with heterogeneous clouds [J]. China Communications, 2020, 17(5): 191-210.
[68] Yu Hong-yan, Wang Qu-yuan, Guo Song-tao. Energy-efficient task offloading and resource scheduling for mobile edge computing [C]//IEEE International Conference on Networking, Architecture and Storage(NAS), Chongqing, China, 2018: 1-4.
[69] Hui Huang, Ye Qiang. Reinforcement learning based offloading for realtime applications in mobile edge computing [C]//IEEE International Conference on Communications(ICC), Dublin, 2020: 1-6.
[70] Guo Xue-ying, Singh R, Zhao Tian-chu, et al. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems [C]//IEEE International Conference on Communications, Kuala Lumpur, 2016: 1-7.
[71] Gong Xiao-wen. Delay-optimal distributed edge computing in wireless edge networks [C]//IEEE Conference on Computer Communications, Toronto, Canada, 2020: 2629-2638.
[72] Ma Xiao, Zhou Ao, Zhang Shan, et al. Cooperative service caching and workload scheduling in mobile edge computing [C]//IEEE Conference on Computer Communications, Toronto, 2020: 2076-2085.
[73] Jia Qing-min, Xie Ren-chao, Tang Qin-qin, et al. Energy-efficient computation offloading in 5G cellular networks with edge computingand D2D communications [J]. IET Communications, 2019, 13(8): 1122-1130.
[74] Liang Ze-zu, Liu Yuan, Lok T M, et al. Multiuser computation offloading and downloading for edge computing with virtualization [J]. IEEE Transactions on Wireless Communications, 2019, 18(9): 4298-4311.
[75] Wang Yan-ting, Sheng Min, Wang Xi-jun, et al. Mobile-edge computing: partial computation offloading using dynamic voltage scaling [J]. IEEE Transactions on Communications, 2016, 64(10): 4268-4282.