研究与开发

基于拉格朗日的计算迁移能耗优化策略

乐光学[1,2]，朱友康[2]，刘建生[2]，戴亚盛[2]，游真旭[2]，徐浩[2]
（1. 嘉兴学院数理与信息工程学院，浙江 嘉兴 314001;
2. 江西理工大学理学院，江西 赣州 341000）

摘　要：随着移动网络技术的发展和智能终端的普及应用，移动边缘计算已成为云计算的一个重要应用。计算迁移策略已成为移动边缘计算服务的关键问题之一。以移动终端总的计算时间和移动终端能耗最小化为目标，将移动终端的计算迁移资源划分问题建模为一个凸优化问题，运用拉格朗日乘子法进行求解，提出基于阈值的迁移优化策略模型。仿真实验表明，本迁移优化策略模型能有效平衡本地计算和迁移计算之间的关系，为移动边缘计算中执行计算密集型应用提供保障。

关键词：边缘计算；计算迁移；能耗；计算时间；拉格朗日；凸优化
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−0801.2018295

Optimizing strategy of computing off loading energy consumption based on Lagrangian method

YUE Guangxue[1,2], ZHU Youkang[2], LIU Jiansheng[2], DAI Yasheng[2], YOU Zhenxu[2], XU Hao[2]
1. College of Mathematical Information and Engineering, Jiaxing University, Jiaxing 314001, China
2. College of Science, Jiangxi University of Science and Technology, Ganzhou 341000, China

Abstract: With the development of mobile network technology and the popularization and application of intelligent terminals, mobile edge computing has become an important application of cloud computing. Computing offloading strategy has become one of the key issues in mobile edge computing services. Targeting the total computing time of the mobile terminal and minimizing the energy consumption of the mobile terminal, the computation offloading resource allocation problem of the mobile terminal was modeled as a convex optimization problem, solved by the Lagrange multiplier method, and a threshold-based offloading was proposed. Simulation experiments show that the proposed offloading optimization strategy model can effectively balance the relationship between local computing and offloading, and provide guarantee for executing computation-intensive applications in mobile edge computing.

Key words: edge computing, computation offloading, energy consumption, computing time, Lagrangian, convex optimization

收稿日期：2018−06−06；修回日期：2018−12−10
基金项目：国家自然科学基金资助项目（No.61462036，No.61572014，No.61702224）；浙江省自然科学基金资助项目（No.LY16F020028，No.LQ15F010008，No.LY15F020040）
Foundation Items: The National Natural Science Foundation of China (No.61462036, No.61572014, No.61702224), The National Natural Science Foundation of Zhejiang Province of China (No.LY16F020028, No.LQ15F010008, No.LY15F020040)

1　引言

近年来，智能手机、平板电脑等移动终端正在成为学习、娱乐、社交、新闻更新和商业交流的重要工具，计算密集型应用程序（如图像处理、移动游戏、移动医疗、移动学习等）大量出现，用户对移动设备性能的要求越来越高。然而，由于移动终端的资源（计算能力、电池能源、存储容量）的限制[1]，这种渐高的期望与终端有限资源之间的落差影响着用户对移动设备的体验质量，降低了用户的满意度。

在物联网和 5G 通信的驱动下，移动计算领域出现了重要转变，由传统的集中式云计算向分布式的移动边缘计算转变。但延迟问题始终是制约云计算发展的一个关键问题，针对该问题，研究者提出了将云服务转移到与用户物理位置邻近的地方，以充分利用移动网络的边缘，构建边缘计算范式。

在传统的云计算中，用户通过访问 Internet 使用云服务。在边缘计算中，计算和存储资源分布于用户的近距离网络拓扑范围内[1]。云计算完全以集中的方式部署，服务器通常集中地放置在一个或几个位置，边缘计算完全以分布式的方式部署。与云计算相比，边缘计算可以明显降低延迟和抖动[2]，但是边缘计算只能够提供有限的计算和存储资源。云计算和边缘计算的关键技术比较见表 1。

表 1　云计算与边缘计算的关键技术比较

评价指标	云计算	边缘计算
部署方式	集中式	分布式
与用户的距离	远	近
延迟	高	低
抖动	大	小
计算能力	强	有限
存储能力	强	有限

在云计算的发展热潮之后，许多云服务，如移动医疗、移动学习、移动游戏和移动管理等都可以在移动设备上直接使用。边缘计算将终端用户的计算任务迁移到附近其他空闲的移动终端上，通过在边缘网络上实现计算和存储来降低网络延迟，达到节省终端能耗、网络传输费用以及计算时间等目的。移动边缘计算作为一种新兴技术，它是在移动用户的近距离范围内提供计算和存储等的网络服务，边缘云服务器部署于用户的物理近距离范围内，而网络运营商只需要负责转发和过滤数据分组。

随着互联网技术的发展，特别是智能移动终端的更新升级，智能手机拥有越来越强大的功能。要让移动设备变得更小、更轻、电池寿命更长，这意味着他们的计算能力将会受到限制。但是，随着用户对智能移动终端的期望越来越高，这就要求终端对计算和数据的操作能力达到更高的水平，而实现更高水平的计算和数据操作能力则是以高能耗为代价的，如何协调这一矛盾是目前智能移动终端发展的技术瓶颈。

针对移动终端资源受限这一课题，移动边缘计算迁移（mobile edge computing offloading，MECO）近年在计算机科学领域被广泛研究，计算迁移技术的发展，为解决终端资源受限这一问题引入了新的方法。计算迁移可以实现将计算体迁移到其他资源丰富的终端上运行[3]、跨终端任务同步[4]、移动设备资源共享[5]等应用目标。Satyanarayanan[6]提出的 Cyber Foraging 思想是计算迁移的最早起源，提出将资源受限的移动终端上的计算、存储等任务交给终端设备附近计算、存储能力更强的服务器来执行，以节省移动终端的计算能量，提升终端的性能。Liu 等[7]运用排队理论对雾计算系统计算迁移过程中的能耗、延迟进行了研究，通过找到每个终端的最优迁移概率和传输功率来最小化能耗和延迟，但是与移动边缘计算迁移相比，雾计算的延迟相对较高。

Gabriel 等[8]提出了 CloudAware，它的主要设计目标是实现与附近节点的临时和短时间交互，CloudAware 的设计可以通过并行化加速计算、迁移计算节省能量或带宽，并且还支持多种移动应用场景的任务迁移。Sherif 等[9]提出的 Replisom 系统，利用压缩采样构造算法，将边缘云中的任务存储到相应的虚拟机中，当多个物联网设备将任务复制到附近的边缘云时，Replisom 体系结构可以降低任务迁移期间的延迟和成本。Karim 等[10]设计出的 Femtoclouds 系统，利用客户附近空闲的移动设备，为本地移动终端提供计算服务，减少了传统的将计算迁移到云数据中心过程中产生的网络延迟。Michael 等[11]提出了 ME-VoLTE（mobile edge computing enabled voice over LTE），目的是减少视频通话中移动设备的电池消耗，并为迁移策略的选择提供通信协议。Noriyuki 等[12]提出了 EAB（edge accelerated Web browsing）移动边缘计算原型，以加速 Web 应用请求的执行。Swaroop 等[13]针对无线网络的高时延，提出了基于 5G 技术和移动边缘计算的上下文感知协同实时应用架构，边缘服务器部署于每个边缘节点上，利用了近距离服务、上下文感知计算等 5G 技术的特性来实现节点间的协作。Gao 等[14]研究了一个基于战区环境的计算迁移架构，该架构将部分应用程序迁移到本地节点附近的其他移动节点执行，以减少本地节点的计算时间和能源消耗，节点间的迁移方案依赖于节点的计算能力、相邻节点的能量级别以及节点之间未来可能发生的交互。表 2 比较了近年边缘计算的几个研究成果。

从表 2 可以看出，学者们对边缘计算迁移系统架构的研究，主要目标集中在增强移动终端计算能力、降低能耗、缩短延迟、提高网络资源利用效率等方面，采用的主要方法是优化迁移策略、优化传输效率等，特别是迁移策略的优化，是当前边缘计算研究的主要方向和重点领域。

移动边缘计算迁移的节能研究需要移动边缘计算和无线通信技术的联合设计。Zhang 等[15]研究了一个单用户 MECO 系统，通过比较本地计算（带有可变 CPU 周期）和优化后的迁移计算（可变传输速率）的能量消耗，得出了最优迁移策略。You[16]和 Mao[17]研究了自适应的计算迁移方法，并以无线能量传输和采集为目标，设计出了基于无缝集成移动云计算和微波功率传输（microwave power transmission，MPT）两种技术的解决方案。Xiang 等[18]研究了单用户的 MECO 系统，将动态迁移与自适应的 LTE/Wi-Fi 链接选择进行集成，并提出了一种可扩展的近似动态规划（approximate dynamic programming，ADP）算法。

参考文献[19-21]针对不同类型的多用户系统研究了 MECO 的资源划分。Stefania[19]研究了一个多用户的 MECO 系统，为了减少在迁移延迟约束下的终端能量消耗，提出将无线和计算资源进行联合分配。Zhao 等[20]研究了在中心云和边缘云共存的情况下，迁移到不同云的最优用户调度问

表 2　近年边缘计算成果比较

研究成果	发表年份	实现方法	设计目标
CloudAware[8]	2016 年	优化迁移策略	加速计算、节能
Replisom[9]	2016 年	克隆虚拟机、压缩	减少响应时间
Femtoclouds[10]	2015 年	优化迁移策略	增强计算能力
ME-VoLTE[11]	2015 年	优化传输	降低能耗
EAB[12]	2015 年	优化传输	加速资源访问
上下文感知协作实时应用[13]	2015 年	上下文感知	在实时场景中降低延迟
端到端计算迁移[14]	2014 年	优化迁移策略	提高整个网络资源利用效率

题，提出了一个基于任务负载的阈值迁移策略。Chen 等[21]研究了多用户 MECO 的分布式迁移，利用博弈论来实现能量和延迟的最小化。

Chen 等[22]用 Lyapunov 技术开发了一种边缘基站之间的在线计算迁移框架，以最大化边缘计算系统的性能，但仅仅涉及了边缘基站之间的协作计算，并没有优化终端的能耗。Zhang 等[23]研究了用户终端移动模式未知情况下的能耗优化，提出了基于卷积神经网络的深度 Q 网络的强化学习算法，从用户过去的数据状态中学习，实现能耗优化，缺点是没有对计算时间进行约束。Zhang 等[24]研究了多流数据迁移问题，一个终端有多个应用程序需要迁移，将多流数据迁移问题构建为限时空离散时间马尔可夫决策过程，通过基于动态规划的算法建立最优策略，但是该策略计算复杂度高，会在一定程度上损害性能。Muhammad 等[25]提出了一种基于激励的博弈论数据下载框架，实现了纳什均衡，但是该框架过分关注于移动网络运营商的服务质量（QoS），而忽视了终端用户的能耗考虑。

Donghyeok 等[26]提出了一种将部分视频流量卸载到 Wi-Fi 网络来缓解蜂窝网络拥塞的软件定义网络（SDN）架构，通过有效且公平地共享有限的蜂窝网络资源来提高所有用户的视频质量，提出了追求用户之间的全局系统效用和服务质量公平性的资源分配算法，该架构能够有效降低延迟，但是对于终端能耗并不能有效优化。Reza 等[27]探讨了迁移计算中的用户隐私安全问题，建议使用智能分区或动态迁移，应默认采用“本地优先”方法，在处理敏感数据时避免使用网络，但是缺乏对富应用程序复杂性的考虑。Liu 等[28]提出了一种权衡迁移延迟和可靠性的框架，设计基于启发式搜索、重构线性化技术和半定松弛 3 种算法来实现延迟和可靠性的权衡，缺点是没有对能耗进行优化。Lyu 等[29]研究了移动边缘计算迁移面临的可扩展性问题，提出了一个轻量级的请求和准入框架来解决可伸缩性问题，设计选择性迁移方案以最小化设备的能量消耗。相关迁移策略的优缺点见表 3。

综上所述，学者们关于 MECO 资源划分的研究主要集中在算法设计上，大多数都是仅考虑了

表 3　相关迁移策略的优缺点

参考文献	发表年份	创新点	存在不足
[15]	2013 年	比较本地计算和迁移计算的能量消耗，得出最优迁移策略	仅通过对比迁移前后的能量消耗无法得到最优的迁移策略，仅研究了单用户情况
[16]	2016 年	以无线能量传输为目标，设计了基于无缝集成移动云计算的解决方案	仅研究了低复杂度设备（如传感器），对于高计算复杂度的智能终端并不适用
[17]	2016 年	以无线能量采集为目标，设计出了基于微波功率传输的解决方案	仅考虑 CPU 频率和发射功率的优化，忽略了计算任务
[18]	2014 年	将动态迁移与自适应的 LTE/Wi-Fi 链接选择进行集成，提出了一种可扩展的近似动态规划算法	注重于单用户的算法设计，缺乏全局最优考量
[19]	2015 年	将无线和计算资源进行联合优化	缺乏严密的计算延迟考量
[20]	2015 年	研究了迁移到不同云的最优用户调度问题，提出了一个基于任务负载的阈值迁移策略	仅关注于延迟问题，没有涉及研究终端能耗
[21]	2015 年	研究了多用户分布式计算迁移，运用博弈论来实现能量和延迟的最小化	分布式的计算迁移算法难以达到全局最优
[24]	2018 年	基于马尔可夫决策和动态规划算法建立最优策略	计算复杂度高
[27]	2018 年	使用智能分区和动态迁移研究了迁移计算中的用户隐私安全问题	缺乏对富应用程序复杂性的考虑
[28]	2018 年	设计基于启发式搜索、重构线性化技术和半定松弛 3 种算法来实现延迟和可靠性的权衡	缺乏对能耗的优化

计算延迟或终端能耗，一些研究的计算复杂度偏高，没有达到全局最优，并不能有效解决移动终端资源受限的问题，特别是能耗问题，因此当前研究中资源划分能耗优化的问题就凸显出来。

本文针对移动终端资源受限的问题，将移动终端计算迁移的资源划分问题建模为一个凸优化问题，采用拉格朗日乘子法进行求解。建立一个面向LTE（long term evolution）应用的、基于时分多址（time division multiple access，TDMA）的多用户MECO系统，研究了多个用户将任务迁移到一个边缘云基站的任务迁移模型。在系统的多用户计算迁移能耗凸优化问题中，以计算时间最优为约束条件、以最小化终端本地能耗为目标来设计迁移方案，最终确定一个最优的终端资源划分策略。

2 系统模型

系统模型为构建在蜂窝网络（LTE）环境下的基于时分多址的MECO系统，将迁移时间 T 分割成互不重叠的时段（帧），再将帧分割成互不重叠的时隙（信道），每个时隙与一个用户具有一一对应的关系，系统依据时隙区分来自不同地址的用户信号，从而完成多址连接的多用户单一边缘云的计算迁移，该系统模型由 K 个索引为 1,2,⋯,k 的终端用户和一个作为边缘云网关的基站构成。系统模型如图1所示。

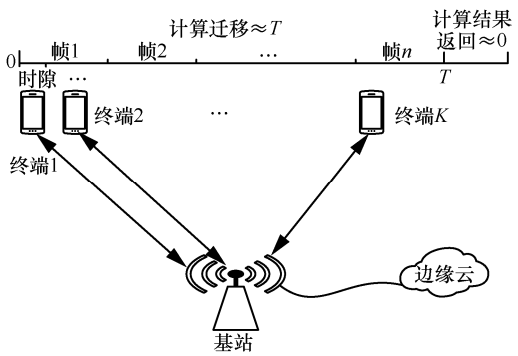


图1 系统模型

如图1所示，当发生迁移计算时，每个时隙包括两个连续的阶段：

（1）计算迁移或本地计算；

（2）云计算以及将边缘云的计算结果下载到移动终端。

由于边缘云服务器计算能力强，数据在服务器上的计算时间几乎可以忽略，并且计算结果相对较小，结果返回的时间也可以忽略。基于以上原因，与第一阶段相比，第二阶段的持续时间可以忽略不计，因此在资源划分时则不予考虑。

在任意时隙内，基站基于TDMA对用户的一个任务子集进行完整/部分迁移调度。基于基站的任务调度相比于博弈论的分布式任务调度方式有一个显著的优点，即任务信息集中到基站，基站可以根据实时网络通信状态调整迁移策略，从而避免分布式调度方式产生的额外系统开销。部分迁移或不迁移的用户分别使用本地CPU计算一部分或全部输入数据。

假定基站对所有终端用户的信道增益、本地计算能量和输入数据大小完全掌握。使用这些信息，基站选择迁移用户，确定迁移的数据大小。假设信道在每个时隙内保持不变。

3 迁移模型

You 等[30]对多用户计算迁移资源划分进行了研究，基于能耗最优构建迁移计算系统模型如式(1)，该文研究表明其模型能有效地满足用户低能耗的要求。

$$\min_{\{l_k,t_k\}} \sum_{k=1}^{K} \beta_k \left[\frac{tk}{h_k^2} f\left(\frac{l_k}{t_k}\right) + (R_k - l_k) C_k P_k \right]$$

$$\text{s.t.} \begin{cases} \sum_{k=1}^{K} t_k \leqslant T \\ t_k \geqslant 0, \forall k \end{cases} \tag{1}$$

但该模型没考虑计算时间约束，当系统执行计算迁移时，会出现执行迁移算法比本地计算方法时间开销大的情况，不能满足系统的计算时间最优条件。

针对这一问题，本文以系统总计算时间最优

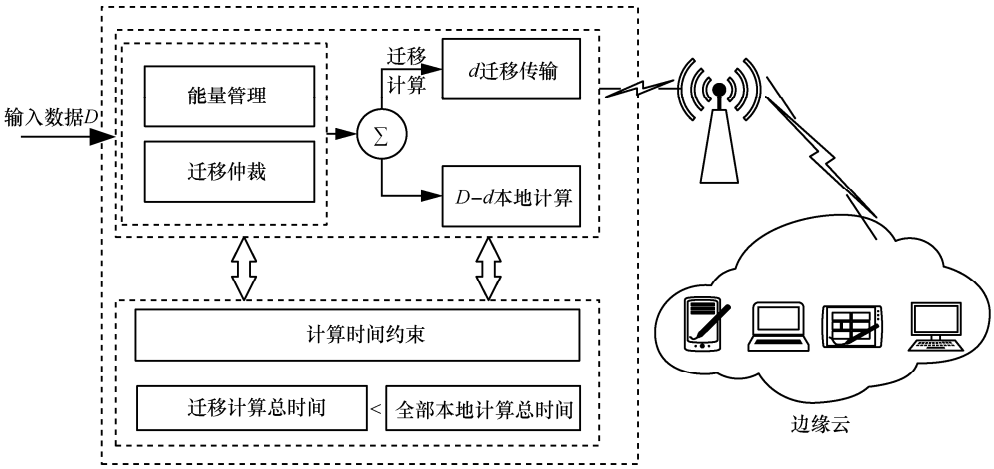


图 2　多用户边缘计算迁移系统

作为约束条件，构建计算迁移模型，如图 2 所示。其核心思想为在多用户计算迁移系统中以能耗优化为目标，以计算时间动态划分为约束，均衡系统计算能力，规避多用户迁移系统中某个用户为了节省资源而盲目迁移的问题，以保证系统总计算时间最低。

假定整个多用户迁移系统中的每个终端都相互联系，由终端 k 来计算时隙内的 D_k 比特输入数据，其中 d_k 比特被迁移，(D_k-d_k)比特留在本地计算。

用 H 表示信道增益(无线信道对传输距离归一化后的衰落度)，σ 是复高斯白噪声的方差，p_k 表示移动终端 k 的传输功率，系统信道带宽为 B。根据香农定理，构建终端 k 可达到的速率模型 r_k 为：

$$r_k = B\mathrm{lb}\left(1+\frac{p_k H^2}{\sigma}\right), \forall k \qquad (2)$$

4　计算模型

计算模型涉及的符号及其含义，见表 4。

对于计算模型，每个终端 k 有一个输入数据为 D_k 比特的计算任务需要完成。D_k 比特数据分组包括系统设置参数、程序代码和输入参数，其中 d_k 比特执行计算迁移，(D_k-d_k)比特留在终端本地计算。Wang 等[31]研究了边缘计算迁移资源联合优化问题，构建能耗、时间和缓存联合优化模型如式（3），其中将 MEC 服务器计算时间分为传输时间、执行时间，没有考虑迁移时本地计算部分执行时间。

表 4　符号及其含义

符号	含义
K	系统中终端用户总数量
k	第 k 个终端用户
T	迁移总时隙
D_k	终端 k 某个任务的数据量
d_k	终端 k 任务迁移的数据量
C_c	边缘云上计算 1 bit 数据所需的 CPU 周期数
C_k	终端 k 计算 1 bit 数据所需的 CPU 周期数
H	信道增益
σ	高斯白噪声的方差
p_k	终端 k 的传输功率
B	系统信道带宽
r_k	数据传输速率
$f_k^{(l)}$	终端 k 的计算能力（即每秒的 CPU 周期数）
$f_k^{(e)}$	边缘云的计算能力（即 MEC 服务器每秒的周期数）
t_k	分配给终端 k 的迁移时隙
a_k	迁移决策约束变量
e_k	终端 k 本地计算每个 CPU 周期的能耗
w_k	终端 k 对应的拉格朗日乘子

$$\underset{a,s,c,h}{\mathrm{Max}} \sum_{n\in N}\sum_{k_n\in K_n} a_{k_n} u(s_{k_n}\Psi_{k_n}+c_{k_n}\phi_{k_n})+h_{k_n}\Lambda_{k_n} \qquad (3)$$

用C_k表示在第k个终端计算1 bit输入数据所需的CPU周期数，用C_c表示在边缘云上计算1 bit输入数据所需的CPU周期数，因此C_kD_k和$C_k(D_k-d_k)$分别表示在终端k上计算D_k和(D_k-d_k)比特数据所需的CPU总周期数，C_cd_k表示在边缘云上计算d_k比特数据所需的CPU总周期数。

（1）本地计算

对于本地计算的方法，任务的所有输入数据D_k都在每个移动设备上本地执行。用$f_k^{(l)}$表示为终端k的计算能力（即每秒的CPU周期数），并且允许不同的终端可以具有不同的计算能力。计算延迟约束条件为：

$$C_k(D_k-d_k)\leqslant f_k^{(l)}T \tag{4}$$

由终端k完成本地执行D_k比特输入数据的计算时间$T_k^{(l)}$表示为：

$$T_k^{(l)}=\frac{C_kD_k}{f_k^{(l)}} \tag{5}$$

（2）MEC服务器计算

对于MEC服务器计算的方法，终端k通过LTE接入，将d_k比特数据迁移到MEC服务器上执行。迁移任务过程中，终端k将计算数据发送到MEC服务器时，需要一定的传输时间。t_k表示系统分配给终端k用于迁移的时隙长度，当发生计算迁移时（t_k>0），最优传输速率可以固定为$r_k=\frac{d_k}{t_k}$，因为这是在时间限制下最节能的传输策略。因此，d_k比特数据的传输时间成本可表示为：

$$T_{k,\text{off}}^{(e)}=\frac{d_k}{r_k} \tag{6}$$

本地执行的(D_k-d_k)比特数据的执行时间为：

$$T_k^{(l_{D_k-d_k})}=\frac{C_k(D_k-d_k)}{f_k^{(l)}} \tag{7}$$

由于计算结果的大小一般远小于计算的输入数据的大小，所以计算结果从MEC服务器返回到终端的传输时间开销可以忽略不计。

根据式（6）、式（7），执行任务迁移的总的执行时间可表示为：

$$T_k^{(e)}=T_{k,\text{off}}^{(e)}+T_k^{(l_{D_k-d_k})} \tag{8}$$

为了保证整个迁移系统不会因为通信情况等的变化，而导致迁移后的总计算时间大于完全本地计算的时间，在迁移决策时，将完全本地计算和迁移计算时间之差作为决策约束条件：

$$\begin{aligned}a_k&=T_k^{(l)}-T_k^{(e)}\\&=\frac{C_kD_k}{f_k^{(l)}}-\left(\frac{d_k}{r_k}+\frac{C_k(D_k-d_k)}{f_k^{(l)}}\right)\\&=\left(\frac{C_k}{f_k^{(l)}}-\frac{1}{r_k}\right)d_k\end{aligned} \tag{9}$$

其中，a_k表示迁移决策时间约束变量。当$T_k^{(l)}-T_k^{(e)}>0$，$a_k=1$，计算迁移对终端k的计算时间的节省是有利的，此时可以进行迁移；而当$T_k^{(l)}-T_k^{(e)}\leqslant 0$，$a_k=0$，即计算迁移并没有节省终端的总计算时间，则完全由本地执行计算，不进行迁移计算。

5 能耗模型

5.1 本地计算能耗

假设每个用户的CPU频率是固定的，不同的用户具有不同的CPU频率。用C_k表示在第k个终端计算1 bit输入数据所需的CPU周期数，e_k表示用户本地计算的每个CPU周期的能量消耗。那么乘积C_ke_k就是每比特的计算所需能量。

由式（4）可以得出：

$$d_k\geqslant D_k-\frac{f_k^{(l)}T}{c_k} \tag{10}$$

令$m_k=D_k-\frac{f_k^{(l)}T}{c_k}$，定义函数$(m)^+=\max\{m,0\}$。因此，在计算延迟约束下，移动终端上被迁移的数据量就可以最小化为$d_k\geqslant m_k^+$，那么在移动终端k上本地计算的总能量消耗$E_{\text{loc},k}$表示为：

$$E_{\text{loc},k}=(D_k-d_k)C_ke_k \tag{11}$$

5.2　计算迁移能耗

如上所述，将分配给终端 k 用于迁移的时隙长度表示为 t_k，$t_k \geqslant 0$，其中 $t_k = 0$ 对应于没有迁移的情况。对于迁移的情况（$t_k > 0$），传输速率固定为 $r_k = \frac{d_k}{t_k}$，因为这是在时间限制下最节能的传输策略。

由式（2）通信模型可得：$p_k = \frac{\sigma\left(2^{\frac{r_k}{B}} - 1\right)}{H^2}$

因此，移动终端 k 的迁移能量消耗为：

$$E_{\text{off},k} = p_k t_k = \frac{t_k \sigma\left(2^{\frac{r_k}{B}} - 1\right)}{H^2} \tag{12}$$

如果 $d_k = 0$ 或 $t_k = 0$，$E_{\text{off},k} = 0$。

多用户 MECO 的资源划分问题可以构建为一个凸优化问题，目标是最小化移动终端能耗的加权总和：$\sum_{k=1}^{K}(E_{\text{off},k} + E_{\text{loc},k})$。在计算延迟和云计算能力等约束下，资源划分模型可以构建为：

$$\min_{\{d_k, t_k\}} \sum_{k=1}^{K}\left[\frac{t_k \sigma\left(2^{\frac{r_k}{B}} - 1\right)}{H_2} + (D_k - d_k)C_k e_k\right] \tag{13}$$

约束条件为：

$$\begin{cases} a_k > 0 \\ \sum_{k=1}^{K} t_k \leqslant T, t_k \geqslant 0 \\ d_k \leqslant D_k, \forall k \end{cases} \tag{14}$$

式（13）问题的凸性证明：定义一个函数 $g(r_k) = \sigma(2^{\frac{r_k}{B}} - 1)$，由于函数 $g(r_k)$ 是凸函数，它的透视函数也是凸函数，即 $y = \frac{t_k \sigma(2^{\frac{r_k}{B}} - 1)}{H^2}$ 是凸函数。因此，目标函数是一组凸函数的和。由于直线也是凸函数，因此式（13）的约束条件是一组线性凸约束。

5.3　拉格朗日乘子法求解

用拉格朗日方法求解式（13）问题，目的是确定需要迁移的数据量 d_k 的大小，用于最优资源划分。式（13）问题的拉格朗日函数为：

$$L = \sum_{k=1}^{K}\left[\frac{t_k g(r_k)}{H^2} + (D_k - d_k)C_k e_k\right] - \omega\left(\frac{C_k}{f_k^{(l)}} - \frac{1}{r_k}\right)d_k \tag{15}$$

其中，ω 是与迁移时间约束相关的拉格朗日乘子。根据 KKT 条件：

$$\frac{\partial L}{d_k} = \frac{g'(r_k)}{H^2} - C_k e_k - \omega\left(\frac{C_k}{f_k^{(l)}} - \frac{1}{r_k}\right) = 0 \tag{16}$$

$$\frac{\partial L}{t_k} = \frac{g(r_k) - r_k g'(r_k)}{H^2} = 0 \tag{17}$$

$$\omega\left(\frac{C_k}{f_k^{(l)}} - \frac{1}{r_k}\right)d_k = 0 \tag{18}$$

由式（16）~式（18）解得：

$$\frac{d_k}{t_k} = B\text{lb}\frac{BH^2 C_k e_k}{\sigma \ln 2} \tag{19}$$

令 $\Gamma_k = \frac{BH^2 C_k e_k}{\sigma \ln 2}$，由对数函数的性质可知，$\Gamma_k > 1$ 时，式（19）才有意义，终端才会进行迁移；否则，该终端不会迁移数据到边缘服务器上进行计算。令 $Q = \left(\frac{C_k r_k}{f_k^{(l)}} - \frac{C_c r_k}{f_k^{(e)}} - 1\right)$，则有：

$$\omega_k(C_k, e_k, H) = \frac{\sigma(\Gamma_{k-1}) - \sigma(\ln 2)(\text{lb}\Gamma_k)\Gamma_k}{H^2 Q} \tag{20}$$

称 ω_k 为迁移优先级函数，移动终端迁移的数据量随着迁移优先级的增加而增加。

证明：优先级函数 ω_k 是根据本地计算能耗、本地计算 CPU 周期以及信道增益的相应变量产生移动终端 k 的迁移优先级值 $\omega_k(C_k, e_k, H)$。对于 $\Gamma_k > 1$ 时，$\omega_k(C_k, e_k, H)$ 是关于 C_k, e_k, H 的单调递增函数，可以通过推导 ω_k 相对于每个参数的一阶导数来证明。

综上所述，关于式（13）的最优资源划分策略可以为：

$$d_k \begin{cases} =0, & \omega < \omega^* \\ =D_k, & \omega > \omega^* \end{cases} \tag{21}$$

$$t_k = \frac{d_k}{B\mathrm{lb}\Gamma} \tag{22}$$

ω^*表示拉格朗日乘子的最优值，并且，分时约束是有效的$\sum_{k=1}^{K} t_k = T$。该划分策略揭示了以节能、低时延为目标的最优集中式资源划分策略在迁移时是一个基于阈值的结构。即可以表述为，由于$\omega_k = \omega^*$的情况实际上很少发生，所以最优策略对每个终端做出二进制迁移决策。具体而言，如果相应的迁移优先级超过给定阈值，即$\omega_k > \omega^*$，则移动终端将其任务的全部数据D_k迁移到边缘云；如果相应的迁移优先级小于给定阈值，即$\omega_k < \omega^*$，移动终端只能在计算时延限制下迁移最小量的数据$d_k = m_k^+ = 0$。

传统的求解凸优化的方法是梯度下降法，但是梯度下降法的计算复杂度较高。为求解式(13)，将每一个终端对应的ω_k做上标签并排序，根据迁移时隙$\sum_{k=1}^{K} t_k \leqslant T$约束，只需要对$\omega_k$进行一维搜索，求解最优值$\omega^*$，可以显著降低计算复杂度，称之为最优资源划分算法（optimal resource partitioning algorithm，ORPA）。为了便于搜索，下面给出ω^*的范围，当至少有一个终端发生迁移计算时，最优拉格朗日乘子ω^*满足：

$$\omega_{\min} \leqslant \omega^* \leqslant \omega_{\max} \tag{23}$$

在做出满足低能耗约束的迁移决策后，为了满足系统的低时延和降低系统总的计算时间，ORPA 所得出的迁移结果还需要满足式(8)约束，即 ORPA 中的步骤 3。

- **步骤 1** 初始化

将每个用户对应的ω_k值按从小到大（$\omega_0 \sim \omega_n$，即$\omega_{\min} = \omega_0, \omega_{\max} = \omega_n$）进行排序，令$\omega_m = \omega_{\max}$，根据$\sum_{k=`}^{K} t_k \leqslant T$，有$T_m = \sum_{k=1}^{K} t_{k,m}$,其中$\{t_{k,m}\}$是$\omega_m$的情况所分配的时隙。

- **步骤 2** 搜索最优ω^*

当$T_m \neq T$且$a_k > 0$时，按如下步骤更新$\{\omega_m\}$:

（1）若$T_m < T$，令$\omega_m = \omega_{n-1}$；

（2）若$T_m = T$，则$\omega^* = \omega_m$，最优策略就可以确定；

（3）若$T_m > T$，停止搜索，$\omega^* = \omega_{m+1}$，最优策略就可以确定。

- **步骤 3** 迁移决策时间约束变量

对于所有$\omega \geqslant \omega^*$的用户：

（1）若a_k>0，该用户迁移；

（2）若$a_k \leqslant 0$，拒绝该用户迁移。

6 仿真与性能评估

为验证本文所提出 ORPA 的有效性，对仿真模型做如下约定，详细仿真参数设置见表 5。

MECO 系统包括具有相等公平权重的 K(20, 40, 60, 80, 100)个移动终端，时隙长度为 T=20 ms，复高斯白噪声的噪声方差$\sigma = 10^{-9}$ W，信道增益 H=180，系统信道带宽 B=10 MHz。

对于每个终端用户，允许不同的移动设备有不同的随机变量，移动终端 k 的传输功率 p_k=100 mW，每个周期的本地计算能耗 e_k 的范围为(0,20×10^{-11})J/转，终端 CPU 的计算能力 $f_k^{(l)}$（即每秒的 CPU 周期）从{1,2,…,10}GHz}集合中均匀选择。

对于计算任务，数据大小 D_k 和每比特所需的 CPU 周期数 C_k 遵循均匀分布，其中 $D_k \in$ [100,500]KB，$C_k \in$[500,1 500]KB 个周期/比特。

边缘云上计算每比特数据所需的CPU周期数 C_c=200 个周期/比特，边缘云分配给终端 k 的 MEC 服务器的计算能力（即 MEC 服务器每秒的 CPU 周期数）$f_k^{(e)}$=100 GHz。

为了便于性能比较，考虑基线相等的资源分配策略，其为具有$\Gamma \geqslant 1$的移动终端分配相应的迁移时隙，并且基于此，优化迁移的数据大小。

表 5　仿真参数

参数	数值
系统中终端用户总数量(K)	(20,40,60,80,100)
迁移总时隙(T)	20 ms
终端 k 某个任务的数据量(D_k)	100~500 KB
边缘云上计算 1 比特数据所需的 CPU 周期数(C_e)	200 个周期/比特
终端 k 计算 1 比特数据所需的 CPU 周期数(C_k)	500~1 500 个周期/比特
高斯白噪声的方差(σ)	10^{-9} W
系统信道带宽(B)	10 MHz
终端 k 的传输功率 p_k	100 mW
终端 k 的计算能力(即每秒的 CPU 周期数)($f_k^{(l)}$)	1~10 GHz
边缘云的计算能力(即 MEC 服务器每秒的周期数)($f_k^{(e)}$)	100 GHz
终端 k 本地计算每个 CPU 周期的能耗(e_k)	0~2×10^{-10} J/转

6.1　系统固定时隙与迁移用户数量关系仿真

将系统时隙长度设置为 20 ms，以系统总用户数量为 20、40、60、80、100 分别计算执行迁移算法和本地计算的用户数量，为了提高仿真结果的准确性，每组计算分别重复执行 1 000 次，取平均值。实验结果如图 3、图 4 所示。

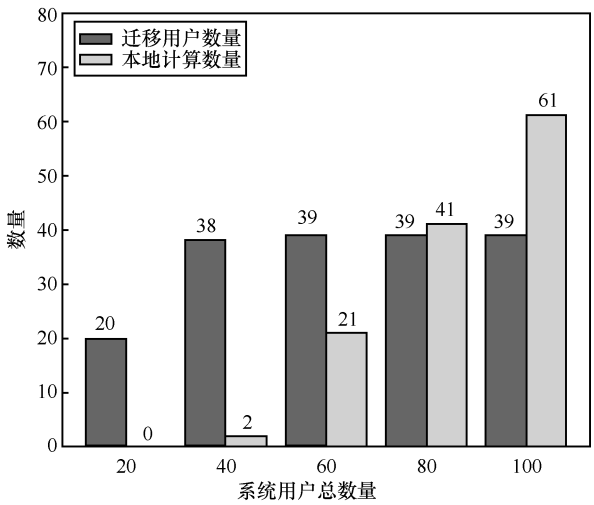


图 3　系统用户总数量与迁移用户数量对比

可以看出：当系统中只有 20 个终端用户时，所有用户的任务都会进行迁移，迁移率为 100%；当系统用户总数量升至 40 个时，有 38 个用户的任务进行了迁移，迁移率为 96%；当系统用户升至 60、80、100 个时，系统中迁移任务用户的都为 39 个，迁移率分别为 65%、48%、39%。当系统迁移时隙长度设置为 20 ms 时，系统能容纳的最优用户数量约为 39 个，若用户数量超过该范围，会出现部分用户的任务留在终端本地计算，不进行迁移计算。

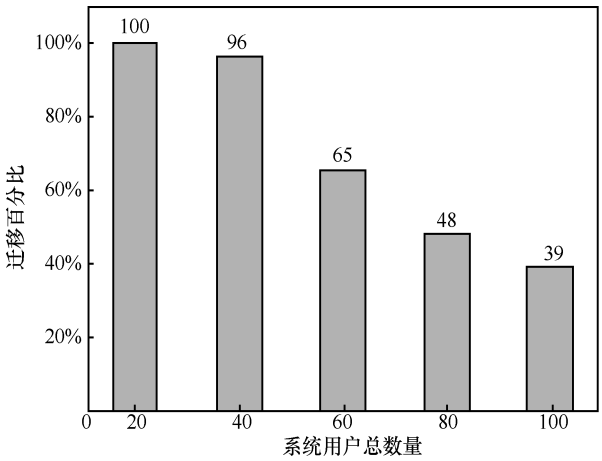


图 4　迁移用户数量占系统总用户数量百分比

6.2　系统固定时隙计算能耗仿真

将系统时隙长度设置为 20 ms，以系统总用户数量为 10、20、30、40、50、60 分别计算执行迁移算法和本地计算所产生的能耗，为了提高仿真结果的准确性，每组计算分别重复执行 1 000 次，取平均值。实验结果如图 5、图 6 所示。

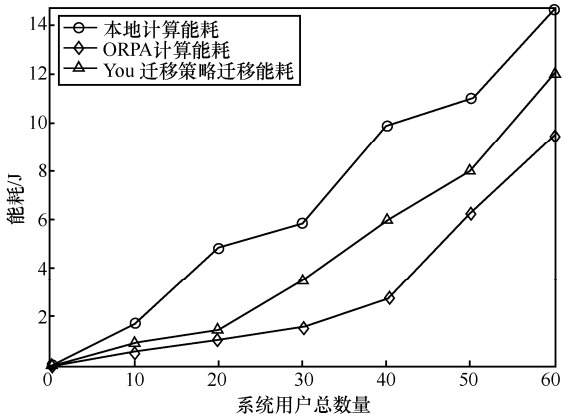


图 5　不同计算策略能耗对比

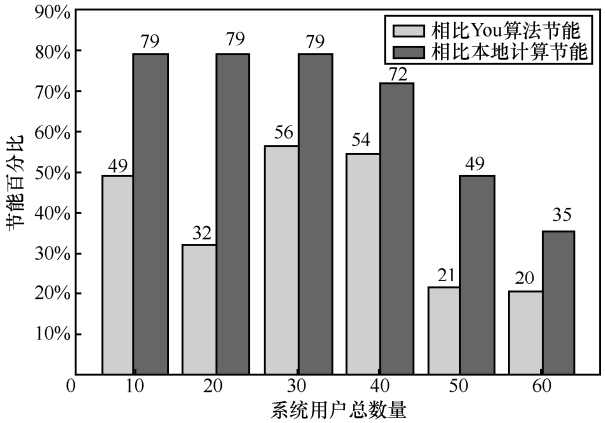


图 6　执行 ORPA 算法相比于 You、本地计算节能的百分比

可以看出：当系统用户小于 40 个时，节能效果尤为明显，节能百分比约为 79%左右；当系统用户总数量超过 40 个时，执行迁移算法的节能效果会呈现明显下降的趋势；当系统用户总数量为 50 个时，节能百分比降至 49%；当系统用户总数量为 60 个时，节能百分比降至 35%；本文提出的 ORPA 所产生的能耗明显小于 You 所提出的策略。并且当用户数量为 30~40 个时，相比于 You 所提出的策略效果更显著。

第 6.2 节的实验结果刚好与第 6.1 节实验得出的结果相吻合，即当系统时隙设置为 20 ms 时，系统能容纳的最优用户数量约为 39 个，执行迁移算法所产生的能耗始终小于完全本地计算产生的能耗，若用户数量超过该范围，会出现部分用户的任务留在终端本地计算，从而能耗也会随着本地计算用户数量的增加而增加。

6.3　系统固定时隙计算时间仿真

将系统时隙长度设置为 20 ms，以系统总用户数量为 10、20、30、40、50、60 分别计算执行迁移算法和本地计算所需的计算时间，为了提高仿真结果的准确性，每组计算分别重复执行 1 000 次，取平均值。实验结果如图 7 所示。

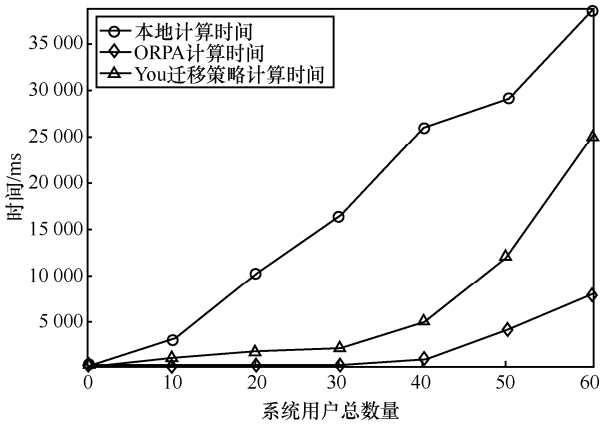


图 7　执行不同迁移策略计算时间对比

可以看出：当系统用户总数量低于 40 个时，执行迁移算法所需的计算时间很小，最高为 500 ms，远远低于本地计算方法所需的时间；当系统用户总数量超过 40 个时，执行迁移算法所需的计算时间上升速度加快。本文提出的 ORPA 所需的计算时间明显小于 You 所提出的策略。

当系统时隙长度设置为 20 ms 时，执行迁移算法所需的计算时间远低于完全本地计算所需的计算时间，这一结果也再次验证第 6.1 节、第 6.2 节实验的仿真结果，系统最优的用户数量约为 39 个。

6.4　时隙长度与系统最优用户数量关系仿真

通过上文的仿真可以看出，在其他条件不变的情况下，时隙长度对系统可容纳的用户数量具有重要影响，因此本实验探讨了不同的时隙长度下系统的最优用户数量。将系统时隙长度分别设置为 10 ms、20 ms、30 ms、40 ms、50 ms、60 ms，计算执行迁移算法的系统最优用户数量，为了提高仿真结果的准确性，每组计算重复执行 1 000 次，取平均值。实验结果如图 8 所示。

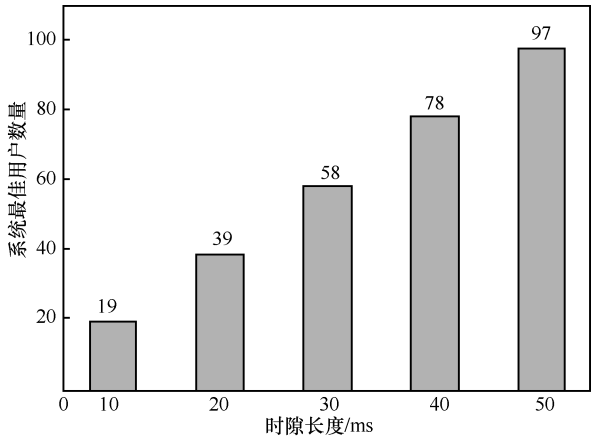


图 8　不同时隙长度与系统最优用户数量

可以看出：时隙长度为 20 ms 时，系统的最优用户数量为 39 个；在相同约束条件下，时隙长度的设置对整个迁移系统所能容纳的最优用户数量具有重要的影响。

迁移系统所能容纳的最优用户数量与时隙长度成正比关系，当时隙长度分别设置为 10 ms、20 ms、30 ms、40 ms、50 ms、60 ms 时，迁移系统所能容纳的最优用户数量分别为 19、39、58、78、97。

6.5　不同算法性能对比

本实验的目的是将本文提出的 ORPA 算法与梯度下降法求解凸优化问题进行性能对比。将系统时隙长度设置为 20 ms，以系统总用户数量为 10、20、30、40、50、60 分别计算 ORPA 算法和梯度下降法求解本文凸优化问题的计算时间。实验结果如图 9 所示。

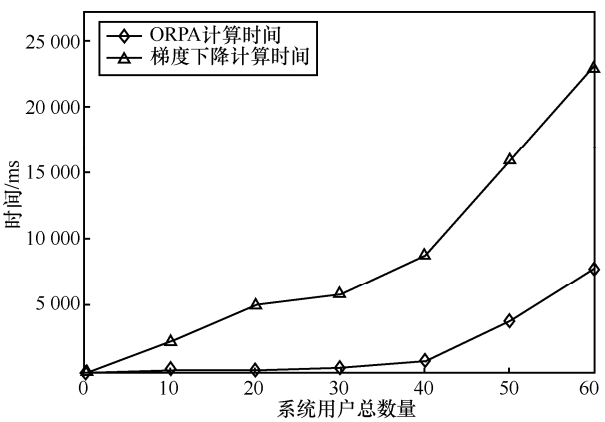


图 9　不同算法性能对比

从图 9 中可以看出，本文提出的 ORPA 算法在计算时间上比梯度下降法求解凸优化问题的速度快，这表明 ORPA 算法在计算复杂度方面具有明显优势。

7　结束语

本文针对移动终端资源受限的问题，将移动终端的计算迁移资源划分问题建模为一个凸优化问题，采用拉格朗日乘子法进行求解。提出了一个面向 LTE 应用的、基于 TDMA 的多用户 MECO 系统，研究了多个用户将任务迁移到一个边缘云基站的系统模型。系统的多用户计算迁移能耗优化被定制为一个凸优化问题，以计算时间最优为约束条件、以最小化移动终端能耗为目标来优化迁移方案，最终确定一个最优的终端资源迁移策略。

通过仿真实验，验证了本文所提出模型的计算迁移策略能够显著降低移动终端任务执行能耗，从而有效地达到节省移动终端能耗的目的。并且在计算延迟约束下，能够显著加快任务的执行时间。在此基础上，本文还得出了不同时隙条件下，系统所能容纳的最优用户数量，以便于按照实际应用场景的需求来设置时隙，满足不同用户种群需求。

参考文献：

[1] PAVEL M, ZDENEK B. Mobile edge computing-a survey on architecture and computation offloading[J]. IEEE Communications Surveys and Tutorials, 2017, PP(99): 1.

[2] 张文丽, 郭兵, 沈艳, 等. 智能移动终端计算迁移研究[J]. 计算机学报, 2016, 39(5): 1021-1038.
ZHANG W L, GUO B, SHEN Y, et al. Mobile offloading on intelligent mobile terminal[J]. Chinese Journal of Computers, 2016, 39(5): 1021-1038.

[3] 关沫. 复杂网络中的计算迁移问题[D]. 沈阳: 东北大学, 2005.
GUAN M. Computing migration in complex networks[D]. Shenyang: Northeastern University, 2005.

[4] 徐羽琼, 谌宗佳, 潘纲, 等. Task Shadow-V: 基于虚拟化的跨移动设备用户任务迁移[J]. 软件学报, 2011, 22(2): 129-136.

XU Y Q, SHEN Z J, PAN G, et al. Task Shadow-V: user task migration across mobile devices based on virtualization[J]. Journal of Software, 2011, 22(2): 129-136.

[5] HUERTA C G, LEE D. A virtual cloud computing provider for mobile devices[C]//ACM Workshop on Mobile Cloud Computing and Serevice: Social Networks and Beyond, June 15, 2010, San Francisco, California, USA. New York: ACM Press, 2010: 1-5.

[6] SATYANARAYANAN M. Pervasive computing: vision and challenges[J]. IEEE Personal Communications, 2001, 8(4): 10-17.

[7] LIU L Q, CHANG Z, GUO X J, et al. No cloud: exploring network disconnection through on-device data analysis[J]. IEEE Pervasive Computing, 2018, 17(1): 64-74.

[8] GABRIEL O, BADE D, WINFRIED L. Computing at the mobile edge: designing elastic android applications for computation offloading[C]//Wireless and Mobile Networking Conferrence, Oct 5-7, 2015, Munich, Germany. Piscataway: IEEE Press, 2015: 112-119.

[9] SHERIF A, BECHIR H, MOHSEN G, et al. Replisom: disciplined tiny memory replication for massive IoT devices in lte edge cloud[J]. Internet of Things Journal, 2016, 3(3): 327-338.

[10] KARIM H,MOSTAFA A, KHALED A H, et al. Fem to clouds: leveraging mobile devices to provide cloud service at the edge[C]// Cloud Computing (CLOUD) 2015 IEEE 8th International Conference, June 27-July 2, 2015, New York, NY, USA. Piscataway: IEEE Press, 2015: 9-16.

[11] MICHAEL T B, SEBASTIAN F, THOMAS S, et al. Me-VoLTE: network functions for energy-efficient video trans coding at the mobile edge[C]//(ICIN)2015 18th International Conference/ Intelligence in Next Generation Networks, Feb 17-19, 2015, Paris, France. Piscataway: IEEE Press, 2015: 38-44.

[12] NORIYUKI T, HIROYUKI T, RYUTARO K. Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated Web browsing[C]//2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile Cloud), March 30-April 3, 2015, San Francisco, CA, USA. Washington DC: IEEE Computer Society, 2015: 233-234.

[13] SWAROOP N, APOSTOLOS K, MOHAMED I, et al. Enabling real-time context-aware collaboration through 5G and mobile edge computing[C]//2015 12th International Conference on Information Technology New Generations (ITNG), April 13-15, 2015, Las Vegas, NV, USA. Piscataway: IEEE Press, 2015: 601–605.

[14] GAO W. Opportunistic peer-to-peer mobile cloud computing at the tactical edge[C]//Military Communications Conference (MILCOM), Oct 6-8, 2014, Baltimore, MD, USA. Washington DC: IEEE Computer Society, 2014: 1614-1620.

[15] ZHANG W, WEN Y. Energy-optimal mobile cloud computing under stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2013, 12(9): 4569-4581.

[16] YOU C, HUANG K, CHAE H C. Energy efficient mobile cloud computing powered by wireless energy transfer (extended version)[J]. IEEE Journal on Select Areas in Communications, 2016, 34(5): 1757-1771.

[17] MAO Y, ZHANG J, LETAIEF K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Select Areas in Communications, 2016, 34(12): 3590-3605.

[18] XIANG X, LIN CL, CHEN X. Energy-efficient link selection and transmission scheduling in mobile cloud computing[J]. IEEE Wireless Communications Letters, 2014, 3(2): 153-156.

[19] STEFANIA S, GESUALDO S, SERGIO B. Joint optimization of radio and computational resources for multicell mobile-edge computing[J]. IEEE Transactions on Signal and Information Processing over Networks, 2014, 1(2): 89-103.

[20] ZHAO T, ZHOU S, GUO X. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing[C]//IEEE Globecom Workshops, Dec 6-10, 2015, San Diego, CA, USA. Piscataway: IEEE Press, 2015: 1-6.

[21] CHEN X, JIAO L, LI W. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2015, 24(5): 2795-2808.

[22] CHEN L X, XU J, ZHOU S. Computation peer offloading in mobile edge computing with energy budgets[C]//2017 IEEE Global Communications Conference, Dec 4-8, 2017, Singapore. Piscataway: IEEE Press, 2017: 1-6.

[23] ZHANG C, LIU Z, GU B, et al. A deep reinforcement learning based approach for cost-and energy-aware multi-flow mobile data offloading[J]. IEICE Transactions on Communications, 2018.

[24] ZHANG C, GU B, LIU Z, et al. Cost-and energy-aware multi-flow mobile data offloading using Markov decision process[J]. IEICE Transactions on Communications, 2018.

[25] MUHAMMAD A, SHAFI U K, RASHID A, et al. Game- theoretic solutions for data offloading in next generation networks[J]. Symmetry Open Access Journal, 2018, 10(8).

[26] DONGHYEOK H, GI S P, SONG H J. Mobile data offloading system for video streaming services over SDN-enabled wireless networks[C]//The 9th ACM Multimedia Systems Conference, June 12-15, 2018, Amsterdam, Nederland. New York: ACM Press, 2018: 174-185.

[27] REZA R, TIMOTHY J P, RONALD P, et al. No cloud: exploring network disconnection through on-device data analysis[J]. IEEE Pervasive Computing, 2018, 17(1): 64-74.

[28] LIU J H, ZHANG Q. Offloading schemes in mobile edge computing for ultra-reliable low latency communication[J]. IEEE Access, 2018, PP(99): 1.

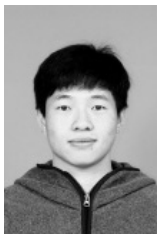
[29] LYU X C, TIAN H, JIANG L, et al. Selective offloading in mobile edge computing for the green internet of things[J]. IEEE Network, 2018, 32(1): 54-60.

[30] YOU C S, HUANG K B. Multiuser resource allocation for mobile-edge computation offloading[C]//2016 IEEE Global Communications Conference, Dec 4-8, 2016, Washington, DC, USA. Piscataway: IEEE Press, 2016: 1-6.

[31] WANG C M, LIANG C C, RICHARD Y F, et al. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2017, 16(8): 4924-4938.

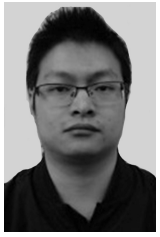
[作者简介]

乐光学（1963−），男，嘉兴学院数理与信息化工程学院教授，主要研究方向为多云融合与协同服务、无线 mesh 网络与移动云计算、混成与嵌入式系统。



朱友康（1992−），男，江西理工大学理学院硕士生，主要研究方向为边缘计算和计算迁移。

刘建生（1959−），男，江西理工大学理学院副教授，主要研究方向为深度学习。



戴亚盛（1993−），男，江西理工大学理学院硕士生，主要研究方向为 mesh 网络协同服务。

游真旭（1993−），女，江西理工大学理学院硕士生，主要研究方向为数据挖掘和推荐算法。

徐浩（1993−），男，江西理工大学理学院硕士生，主要研究方向为数据挖掘。