基于次模优化的边云协同多用户计算任务迁移方法

梁冰[1,2,3]，纪雯[1,3,4]

（1. 中国科学院计算技术研究所，北京 100190；2. 中国科学院大学计算机科学与技术学院，北京 100190；
3. 移动计算与新型终端北京市重点实验室，北京 100190；4. 鹏城实验室，广东 深圳 518055）

摘 要：为了提升多用户计算任务卸载时的系统效用，提出了一种基于边云联合计算的多用户任务卸载方案。该方案在提升系统效用的同时，考虑了边云资源的协同优化问题。针对计算任务卸载模式的选择及边云资源分配的问题，设计了一种基于次模理论的贪心算法并充分利用了云端以及边缘端的计算和通信资源。仿真结果表明，所提方案能够有效降低计算任务执行的时延和能耗，且当多用户卸载计算任务时，所提方案在资源受限的条件下仍然能够保持稳定的系统性能。

关键词：云计算；边缘计算；多用户计算卸载；次模优化；边云联合计算

中图分类号：TN92

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2020205

Multiuser computation offloading for edge-cloud collaboration using submodular optimization

LIANG Bing[1,2,3], JI Wen[1,3,4]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
2. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China
3. Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing 100190, China
4. Peng Cheng Laboratory, Shenzhen 518055, China

Abstract: A computation offloading scheme based on edge-cloud computing was proposed to improve the system utility of multiuser computation offloading. This scheme improved the system utility while considering the optimization of edge-cloud resources. In order to tackle the problems of computation offloading mode selection and edge-cloud resource allocation, a greedy algorithm based on submodular theory was developed by fully exploiting the computing and communication resources of cloud and edge. The simulation results demonstrate that the proposed scheme effectively reduces the delay and energy consumption of computing tasks. Additionally, when computing tasks are offloaded to edge and cloud from devices, the proposed scheme still maintains stable system utilities under ultra-limited resources.

Key words: cloud computing, edge computing, multiuser computation offloading, submodular optimization, edge-cloud computing

1 引言

随着人工智能和物联网移动应用的快速发展，自然语言处理、增强现实、人脸识别、行为分析等高计算能力和通信资源需求的服务被广泛地应用在各种移动终端设备中。由于移动终端设备受到电

收稿日期：2020−07−16；**修回日期**：2020−09−03

通信作者：纪雯，jiwen@ict.ac.cn

基金项目：国家重点研发计划基金资助项目（No.2017YFB1400100）；国家自然科学基金资助项目（No.62072440）；北京市自然科学基金资助项目（No.4202072）

Foundation Items: The National Key Research and Development Program of China (No.2017YFB1400100), The National Natural Science Foundation of China (No.62072440), The Beijing Natural Science Foundation (No.4202072)

池电量、计算能力和存储空间等限制，通常难以满足上述应用的超低时延和低能耗的需求。因此，高资源需求的应用服务与资源受限的移动设备之间的鸿沟给当前及未来物联网移动应用的发展带来了极大挑战[1-3]。

针对上述挑战，移动云计算（MCC, mobile cloud computing）的提出为这些应用需求提供了新的解决思路[4-5]。MCC 利用无线网络将移动终端的应用任务卸载到计算能力更强的云端执行，可以解决终端计算能力不足的问题，并降低终端能耗。然而，云服务器往往位于远离用户终端位置的核心网，因此终端与云服务器之间的数据交换过程需要花费较多的传输时间和能耗。并且，针对许多特定的应用，例如语音识别、智能环境控制等，较长的时延会损害用户的使用体验并影响相关应用的性能表现。

为了解决上述移动云计算的缺点，欧洲电信标准化协会（ETSI, European Telecommunications Standards Institute）提出了一种新型的计算和存储服务技术——移动边缘计算（MEC, mobile edge computing）[6-7]。MEC 作为 5G 的一项核心技术，通过将服务器部署在离用户更近的边缘位置，例如附近的网关和基站侧，能更有效地解决用户对特定应用服务的需求，例如高计算能力、存储服务、高可靠性、移动性支持和低时延等[8-10]。利用 MEC 网络将地理分散的用户的计算任务从移动终端设备迁移到资源丰富的 MEC 服务器，从而加快任务的执行[11-12]。然而，受限于移动边缘计算网络的通信能力以及 MEC 服务器的计算资源和存储容量的限制，卸载过多的计算任务到边缘侧会给计算资源有限的 MEC 服务器及网络传输带来沉重的负担，进而降低边缘计算服务的用户体验[13-14]。对于上述云计算和边缘计算所存在的缺点，结合各自优势的基于边云联合计算的方式为当前的应用需求提供了新的解决思路。然而，如何平衡云端和边缘端的计算任务的负载，提高边云联合计算的服务质量，成了边云联合计算待解决的关键问题之一[15-16]。

本文提出了一种基于边云联合计算下的多用户任务卸载方案，该方案联合考虑了用户任务卸载选择、边缘端和云端的通信及计算资源分配的问题，进而最大化系统的效用。本文的主要贡献如下。

1) 基于用户 QoE（quality of experience）的效用函数，将用户任务卸载选择以及边缘端和云端的通信、计算资源分配的问题表示为一个混合整数非线性规划（MINLP, mixed integer nonlinear programming）问题。本文以最大化系统效用为目标，对用户的任务卸载决策、发射功率、边缘节点的计算资源分配以及回程链路的通信资源分配等问题进行了联合优化。

2) 本文将原始的系统效用最大化问题分解成 2 个子问题，分别为固定用户的卸载决策后的资源分配问题以及优化资源分配问题后对应的最优值函数下的任务卸载决策问题。对于资源分配问题，本文进一步将该问题分解为用户上传发射功率分配问题、边缘端计算资源分配问题和核心网传输带宽分配问题，并利用拟凸和凸优化技术对上述 3 个问题进行了求解。

3) 通过对用户系统效用函数的分析，本文从卸载决策的角度证明了系统效用函数是一个次模函数，进而基于次模理论设计了一种贪心的卸载策略算法用来求解用户任务的卸载决策问题。仿真实验结果表明，与其他卸载方案相比，本文提出的基于边云联合的计算方案下的用户卸载方法能够有效降低用户任务执行时延和能耗，并且在资源受限的条件下仍然能够保持稳定良好的系统效用值。

2 相关工作

如今，国内外学者开展了大量针对移动云计算和移动边缘计算的任务卸载问题的研究。文献[17]研究了动态环境下的多用户计算卸载问题。考虑到多个物联网设备同时通过无线信道卸载计算任务时的信道干扰问题，该研究将用户的计算卸载决策问题表述为一种进化博弈的模型，并设计了一种基于强化学习的进化博弈算法用来求解用户的卸载决策。文献[18]研究了基于车辆边缘计算网络中的计算卸载问题。考虑到车辆需要在动态网络环境下实时确定其任务卸载策略，该研究提出了一种多用户非合作计算卸载博弈，以调整车辆边缘计算网络中每辆车的任务卸载概率，并同时考虑了车辆与边缘计算接入点之间的距离。进一步地，该研究基于计算卸载博弈模型构造了分布式最佳响应算法，以最大化每种车辆的效用。文献[19]构建了适用于移动和普适计算场景的三层架构下的用户卸载问题，提出了一种分布式的均衡计算算法用来确定用户的计算任务卸载的决策。文献[20]研究了动态环境下移动云计算的多用户计算卸载问题，考虑到移动用户在将计算任务卸载到移动云上的自利性和自

私性，将动态环境下移动用户的卸载决策过程描述为随机博弈问题，最后求解用户的卸载决策。文献[21]提出了移动边缘计算和云计算的联合卸载优化问题，并设计了一种基于博弈论的卸载调度和负载均衡方案，但该研究仅对分层框架下的用户卸载决策进行了建模与优化，并未优化边、云的计算以及通信资源的分配问题。上述相关研究对用户任务卸载的决策问题给出了一些求解的方法，但是这些研究主要集中在用户的卸载决策问题上，忽略了卸载过程中系统有限的通信与计算资源分配的问题。针对上述研究存在的问题，文献[4]研究了基于移动边缘云计算的无线信道干扰环境下的多信道多用户的计算卸载问题，提出了一种分布式的用户计算卸载算法并联合考虑了边缘云的计算资源分配问题。文献[22]以移动边缘计算系统下的用户移动设备能耗最小化为目标，提出了一种计算卸载、子载波分配和计算资源分配的联合优化策略，并设计了一个有界改进分支定界算法来寻找全局最优解。文献[23]提出了一种联合用户计算任务部分卸载和资源分配的方案，联合考虑所需能耗、部分卸载和资源分配约束条件，最大程度地减少所有设备任务执行产生的时延之和。文献[24]提出了一种由半定松弛、交替优化和连续调节组成的三步算法，联合优化了用户任务的卸载决策以及计算和通信资源的分配，以最大限度地降低所有用户执行任务产生的能耗和时延。文献[25]考虑了移动边缘计算用户计算卸载的花费和时延问题，该研究以最小化移动设备系统花费和时延为目标，提出了一种多目标的计算卸载与资源分配的算法，以联合优化用户的卸载决策以及边缘端的资源分配。文献[26]面向工业物联网场景下的雾节点计算任务，以雾节点能耗最小化为目标，提出了一种节能的计算卸载方案，并综合考虑了雾节点能耗、本地计算、传输状态和等待状态的能耗。针对该能耗最小化问题，该研究提出了一种加速梯度算法，可以快速找到最优的任务卸载比，从而提高了传统方法的收敛速度。文献[27]考虑了基于移动边缘计算的节能卸载框架并联合优化无线通信资源和计算资源，以最小化用户设备的能耗为目标，设计了基于基尼系数的贪婪启发式算法以降低用户的能耗。文献[28]研究了 5G 异构网络的 MEC 节能计算卸载机制，以最小化系统能耗为目标，联合优化任务卸载和无线资源分配问题，并结合 5G 异构网络的多址特性，提出了一种节能计算卸载（EECO, energy-efficient computation offloading）算法，有效提升了系统能耗的效率。文献[29-30]研究了边缘端或近端云的联合用户卸载决策与资源优化的问题，并分别提出了一种启发式的卸载决策算法提升系统的效用。但上述研究仅考虑用户任务只向边缘端或云端进行任务卸载，并未考虑在边云联合计算的环境下用户如何进行卸载的策略以及资源的优化问题。

综上所述，已有的相关研究主要包含以下 2 个方面：1) 在不考虑通信及计算资源分配的情况下，优化云或边缘计算的用户卸载决策；2) 针对云或边缘计算的用户任务卸载，联合优化用户的卸载决策以及资源分配的问题。然而，在实际应用场景中，当选择任务卸载的用户数量增多时，考虑到边缘节点的计算能力以及远端云传输带宽的受限性，仅面向云端或边缘端的任务卸载将带来任务卸载的高时延性问题[1]。当多用户选择将任务卸载到云端时，由于远端的云需要支撑大量用户的接入需求，因此还需要考虑传输过程中核心网有限的带宽资源分配问题。针对上述问题，本文研究了基于边云联合计算下的任务卸载方法，联合优化了该方法下的用户计算任务卸载决策、边缘端计算和通信资源及远端传输过程中核心网有限的带宽资源分配。并且，本文提出的方法还能够解决仅考虑边缘计算或云计算情况下的用户计算卸载问题，因此具有更广泛的适用性。

3　系统模型

如图 1 所示，本文给出了基于边云联合计算的任务卸载框架的系统模型。系统模型由一个宏 eNode B（MeNB）和其所覆盖小区内的多个终端设备用户组成。该宏 eNode B 配备了边缘计算服务器并通过核心网与远端的云服务器相连。

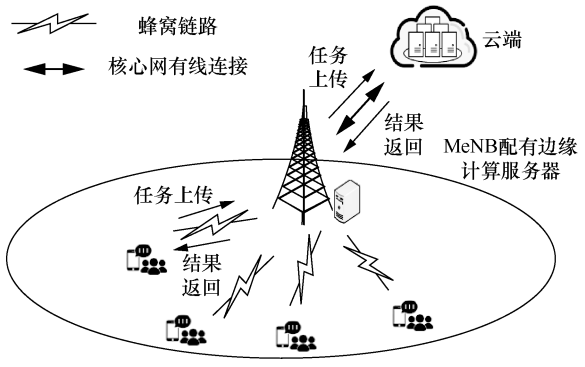

图 1　边云联合计算的多用户任务卸载框架

本文定义 MeNB 覆盖下的用户数量的集合为 $\mathcal{U}=\{1,2,\cdots,U\}$，$u\in\mathcal{U}$ 表示该用户集合中的某一个用户。设用户 u 需要执行的计算任务为 $\mathcal{T}_u=\{d_u,c_u\}$，且该任务不可再分割。其中 d_u 为该任务执行需要传输的数据量（例如系统的设置、参数的设置以及程序代码等），c_u 为该任务执行所需要的计算资源（例如完成该任务所需要的 CPU cycle 总数）。每个用户可以选择的任务执行模式如下：1) 本地计算，即用本地的终端设备处理任务；2) 边缘计算，即通过蜂窝网络将任务卸载到 MeNB 后，在边缘计算服务器处理任务；3) 云端计算，首先通过蜂窝网络将任务卸载到 MeNB 后，再通过核心网传至远端的云服务器处理。此外，本文定义了任务卸载决策变量 $x_{u,j}=\{0,1\}$。其中 $x_{u,j}$=1 表示用户 u 选择模式 j 进行计算，$x_{u,j}$=0 表示用户 u 选择其他模式进行计算。j=0,1,2 表示所选择的任务执行模式，其中 j=0 表示本地计算，j=1 表示边缘端计算，j=2 表示云端计算。接下来，本文将分别给出本地计算、边缘端计算及云端计算。

3.1 本地计算

设 f_u^l 为用户本地设备的计算能力，t_u^l 为用户在本地设备完成该任务所需的时间，因此，可以得出用户在本地执行时的计算时间为

$$t_u^l=\frac{c_u}{f_u^l} \tag{1}$$

根据文献[20,28]，用户执行任务 $\mathcal{T}_u$ 的能耗可表示为

$$E_u^l=P_u^l t_u^l \tag{2}$$

其中，E_u^l 为用户在本地计算的能耗；P_u^l 为用户的本地设备执行任务时的功率[28-29]，该系数值取决于用户的芯片结构以及本地设备的 CPU 频率，并可通过实验测定[31-32]。

3.2 边缘端计算

当用户选择通过将任务卸载到边缘端及云端时，完成任务的总时间包括：1) 用户将计算任务上传至 MeNB 所需的时间 $t_u^{\mathrm{up}(e)}$；2) 用户任务在 MEC 的执行时间 $t_u^{\mathrm{exe}(e)}$；3) 将任务完成的结果从 MEC 传输到用户设备的时间。如果该任务卸载到云端执行，则完成该任务的总时间除了包含将任务上传至 MeNB 所需的时间 $t_u^{\mathrm{up}(e)}$ 外，还包含：1) 用户将计算任务从 MeNB 上传至云端所需的时间 $t_u^{\mathrm{up}(ec)}$；2) 用户任务在云端的执行时间 $t_u^{\mathrm{exe}(c)}$；3) 将任务完成的结果从云端传输到 MeNB 的时间。由于通常情况下，任务完成的输出结果的大小远远小于任务的输入大小，并且考虑到传输的下行速度远大于上行速度，因此本文忽略任务从云端传输到 MEC 的时间以及 MEC 传输到用户设备的时间[29,33-34]。

本文考虑上传时用户网络为多用户正交频分多址接入（OFDMA, orthogonal frequency division multiple access）系统，该系统中的每个信道都是正交的，因此可以忽略小区内的干扰。定义 B 为系统的无线链路的上行带宽，则每个用户可用的上行带宽 $W=\frac{B}{N}$，其中 N 为小区内的用户数量。由此，得到用户 u 的任务 $\mathcal{T}_u$ 的上行传输速率为

$$R_u(p_u)=W\mathrm{lb}\left(1+\frac{p_u h_u}{{\sigma_0}^2}\right) \tag{3}$$

其中，p_u 表示用户 u 上传任务的输入为 d_u 时的发射功率，p_u 为正数且不超过允许的最大值 P_u，即 $0<p_u\leqslant P_u$；h_u 表示用户 u 与基站间的蜂窝上行信道增益；${\sigma_0}^2$ 表示传输背景噪声功率。根据式(3)，用户 u 卸载任务 $\mathcal{T}_u$ 到 MeNB 的上行传输时间 $t_u^{\mathrm{up}(e)}$ 为

$$t_u^{\mathrm{up}(e)}=\frac{d_u}{R_u} \tag{4}$$

接下来，给出用户 u 的任务 $\mathcal{T}_u$ 在 MEC 的执行时间。本文设 MEC 服务器的计算资源上限为 f^e，其表示边缘服务器可用的 CPU cycle 总数。所有请求将任务卸载到 MEC 服务器进行计算的用户共同分享 MEC 服务器的计算资源。定义 MEC 服务器分配给卸载到边缘的用户 u 的计算资源大小为 f_u^e，且 $f_u^e>0$。由于 MEC 服务器计算资源有限，分配给所有将任务卸载到边缘端的用户的计算资源总和不能超过 MEC 服务器的计算资源的上限，因此 f_u^e 满足约束条件，即

$$\sum_{u\in\mathcal{U}_e} f_u^e\leqslant f^e \tag{5}$$

其中，$\mathcal{U}_e=\{u\in\mathcal{U}\mid\sum_{u\in\mathcal{U}}x_{u,1}=1\}$ 为选择将任务卸载到边缘端执行的用户集合。根据 MEC 服务器分配的计算资源大小 f_u^e，能够得出任务 $\mathcal{T}_u$ 在 MEC 服务器的计算耗时 $t_u^{\mathrm{exe}(e)}$ 为

$$t_u^{\mathrm{exe}(e)}=\frac{c_u}{f_u^e} \tag{6}$$

根据式(4)和式(6)，可以得出当给定用户发射功率 p_u 时，用户选择边缘计算模式执行任务卸载的总时延为

$$t_u^e = t_u^{\mathrm{up}(e)} + t_u^{\mathrm{exe}(e)} \tag{7}$$

用户通过边缘计算模式产生的能耗 E_u^e 为

$$E_u^e = p_u \frac{d_u}{R_u} \tag{8}$$

3.3 云端计算

当用户选择云端计算模式执行任务卸载时，假设云端给卸载任务 $\mathcal{T}_u$ 分配的计算资源大小为 f_u^c。尽管云端具有非常丰富的计算资源，但由于需求远程云端计算的任务请求数量非常庞大，因此云端会为每个用户分配固定且受限的计算资源。本文设 f_u^c 为固定大小，且等于云端能为用户分配的计算资源的最大值。因此，类似于式(6)，可以得出用户任务在云端的计算时间 $t_u^{\mathrm{exe}(c)}$ 为

$$t_u^{\mathrm{exe}(c)} = \frac{c_u}{f_u^c} \tag{9}$$

考虑到在云端执行用户任务需要通过核心网传至远端的云服务器，因此可以得出选择云端执行模式时任务卸载的总上传时延为

$$t_u^{\mathrm{up}(c)} = t_u^{\mathrm{up}(e)} + t_u^{\mathrm{up}(ec)} = \frac{d_u}{R_u} + \frac{d_u}{R_u^c} \tag{10}$$

其中，$t_u^{\mathrm{up}(e)}$ 为任务从用户设备端卸载到 MeNB 的时间，$t_u^{\mathrm{up}(ec)}$ 为任务从 MeNB 传输到云端的时间，R_u^c 为核心网分配给用户 u 的传输速率大小。考虑到核心网的总传输带宽有限，因此 R_u^c 满足如式(11)所示的约束条件。

$$\sum_{u\in\mathcal{U}_c} R_u^c \leqslant R^c \tag{11}$$

其中，$\mathcal{U}_c = \{u \in \mathcal{U} \mid \sum_{u\in\mathcal{U}} x_{u,2} = 1\}$ 为选择将任务卸载到云端执行的用户集合，R^c 为核心网总传输带宽。根据式(9)和式(10)，可以得出用户选择云端计算模式执行任务卸载的总时延为

$$t_u^c = t_u^{\mathrm{up}(c)} + t_u^{\mathrm{exe}(c)} \tag{12}$$

由于用户仅在将任务上传至 MeNB 时有能源消耗，因此用户通过云端计算模式产生的能耗为

$$E_u^c = E_u^e = p_u \frac{d_u}{R_u} \tag{13}$$

3.4 基于边缘端−云端联合计算的系统效用最大化问题

在边缘端−云端联合计算框架下，用户的 QoE 主要由完成任务所产生的时延和能耗体现。基于 3.1～3.3 节各模式下计算卸载模型及用户偏好，本文定义用户 u 的卸载效用函数为[29-30]

$$V_u = x_{u,1}\left(\beta_u^t \frac{t_u^l - t_u^e}{t_u^l} + \beta_u^e \frac{E_u^l - E_u^e}{E_u^l}\right) + x_{u,2}\left(\beta_u^t \frac{t_u^l - t_u^c}{t_u^l} + \beta_u^e \frac{E_u^l - E_u^c}{E_u^l}\right) \tag{14}$$

其中，β_u^t 和 β_u^e 分别代表用户对完成任务产生的时延以及能耗的偏好权重，且 $\beta_u^e, \beta_u^t \in [0,1]$，$\beta_u^t + \beta_u^e = 1$，$\forall u \in \mathcal{U}$。例如，当用户 u 的设备电池所能使用的时长较短时，用户会更偏好于提升 β_u^e 的值，以牺牲时延为代价从而节省电量。基于上述用户 u 的卸载效用函数，本文定义系统效用函数为 $V=\sum_{u=1}^{U} V_u$ [29-30]。上述系统效用函数模型涉及通信资源、边缘服务器计算资源以及云端传输资源的分配，其不仅考虑到用户的效用并且关注资源提供者的资源分配问题。因此，本文将基于边云联合计算的系统效用最大化问题表示为

$$\begin{aligned}
&\max_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{p},\boldsymbol{R}} \ V = \sum_{u=1}^{U} V_u \\
&\text{s.t.} \quad \text{C1: } x_{u,j} = \{0,1\}, u \in \mathcal{U}, j \in \{1,2\} \\
&\qquad \text{C2: } 0 < p_u \leqslant P_u, \forall u \in \mathcal{U} \\
&\qquad \text{C3: } \sum_{u\in\mathcal{U}_e} f_u^e \leqslant f^e \\
&\qquad \text{C4: } f_u^e > 0, \forall u \in \mathcal{U}_e \\
&\qquad \text{C5: } \sum_{u\in\mathcal{U}_c} R_u^c \leqslant R^c \\
&\qquad \text{C6: } R_u^c > 0, \forall u \in \mathcal{U}_c
\end{aligned} \tag{15}$$

在上述系统效用最大化问题中，卸载决策 $\boldsymbol{x}$ 与通信资源和计算资源的优化相结合。由于卸载决策 $\boldsymbol{x}$ 是 0-1 整型向量且 $\boldsymbol{f}$、$\boldsymbol{p}$、$\boldsymbol{R}$ 是连续型向量，因此式(15)优化问题是一个 MINLP 问题[35]。考虑到优化问题的表达结构，当给定卸载决策 $\boldsymbol{x}$ 的取值时，可以将复杂度较高的原始优化问题分解为具有较低复杂度的主问题和一系列的子问题[36]。因此，式(15)所示问题可以转化为

$$\begin{aligned}
&\max_{\boldsymbol{x}} \max_{\boldsymbol{R},\boldsymbol{f},\boldsymbol{p}} \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c} V_u \\
&\text{s.t. C1~C6}
\end{aligned} \tag{16}$$

由于卸载决策的限制条件 C1 与资源分配策略的限制条件 C2～C6 是可分离的，因此式(16)所示的优化问题可以分解为主问题和子问题，分别如式(17)和式(18)所示。

$$\max_{\boldsymbol{x}} V^* \quad \text{s.t. C1} \tag{17}$$

$$V^* = \max_{\boldsymbol{R},\boldsymbol{f},\boldsymbol{p}} \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c} V_u \quad \text{s.t.} \quad \text{C2~C6} \tag{18}$$

将式(15)的优化问题分解为式(17)和式(18)的优化问题并不会改变其最优解[36]，接下来，本文将分别给出式(17)和式(18)优化问题的求解方法，并最终求解出式(15)问题。

3.5 联合优化边云资源方法

根据式(14)的形式，当给定卸载决策 $\boldsymbol{x}$ 时，式(18)优化问题可以转化为

$$\max_{\boldsymbol{R},\boldsymbol{f},\boldsymbol{p}} V(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}) = \max_{\boldsymbol{R},\boldsymbol{f},\boldsymbol{p}} \sum_{u\in U_e\cup U_c} (\beta_u^t+\beta_u^e) - I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}) \tag{19}$$

其中，$I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R})$ 为

$$I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}) = \sum_{u\in\mathcal{U}_e}\left(\beta_u^t\frac{t_u^e}{t_u^l}+\beta_u^e\frac{E_u^e}{E_u^l}\right)+\sum_{u\in\mathcal{U}_c}\left(\beta_u^t\frac{t_u^c}{t_u^l}+\beta_u^e\frac{E_u^c}{E_u^l}\right) \tag{20}$$

在给定卸载决策 $\boldsymbol{x}$ 时，$\sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c}(\beta_u^t+\beta_u^e)$ 为常数，因此可以将式(19)转化为 $I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R})$ 最小化问题，即

$$\min_{\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}} I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}) \quad \text{s.t.} \quad \text{C2~C6} \tag{21}$$

根据式(1)～式(13)，可以得出

$$I(\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{R}) = \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c}\frac{\phi_u+\psi_u p_u}{\mathrm{lb}(1+p_u\gamma_u)}+\sum_{u\in\mathcal{U}_e}\frac{\beta_u^t f_u^l}{f_u^e}+\sum_{u\in\mathcal{U}_c}\frac{\beta_u^t f_u^l}{f_u^c}+\sum_{u\in\mathcal{U}_c}\frac{\beta_u^t d_u f_u^l}{c_u R_u^c} \tag{22}$$

其中，$\phi_u=\dfrac{\beta_u^t d_u}{t_u^l W}$，$\psi_u=\dfrac{\beta_u^e d_u}{E_u^l W}$，$\gamma_u=\dfrac{h_u}{\sigma_0}$。

根据式(22)的形式可以发现，当给定卸载策略 $\boldsymbol{x}$ 时，式(22)等号右边的第三项为常数。对于式(22)的上传发射功率 p_u、边缘计算资源 f_u^e 以及核心网传输带宽 R_u^c 的分配，其目标函数和约束条件可以彼此解耦。因此，可以将式(21)的优化问题转化为求解以下 3 个独立的优化问题：1) 上传发射功率分配问题；2) 边缘端计算资源分配问题；3) 核心网传输带宽分配问题。

3.5.1 上传发射功率分配问题

上传发射功率分配优化问题的表达形式为

$$\min_{\boldsymbol{p}} \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c} \Lambda(p_u) \quad \text{s.t.} \quad \text{C2: } 0<p_u\leqslant P_u, \forall u\in\mathcal{U} \tag{23}$$

其中，有

$$\Lambda(p_u)=\frac{\phi_u+\psi_u p_u}{\mathrm{lb}(1+p_u\gamma_u)} \tag{24}$$

引理 1 $\Lambda(p_u)$ 在定义域 C2 内是严格拟凸的。

证明 详见附录 1。

对于上述拟凸问题，通常可以采用二分法对其进行求解[37]。由于拟凸函数在一阶导数的递减点处达到局部最优，且严格拟凸函数的任何局部最优都是全局最优[38]。基于引理 1，可以确定上传发射功率的最优解 p_u^* 要么位于约束条件的边界，如 $p_u^*=P_u$，要么其最优解 p_u^* 满足条件 $\Lambda'(p_u^*)=0$。当 $\Lambda'(p_u)=0$ 时，根据式(38)可知，$\Theta(p_u)=\psi_u\mathrm{lb}(1+p_u\gamma_u)-\dfrac{\gamma_u(\phi_u+\psi_u p_u)}{(1+p_u\gamma_u)\ln 2}=0$。考虑到 $\Theta(p_u)$ 的一阶导数 $\Theta'(p_u)=\dfrac{\gamma_u^2(\phi_u+\psi_u p_u)}{(1+p_u\gamma_u)^2\ln 2}>0$，且 $\Theta(0)=-\dfrac{\gamma_u\phi_u}{\ln 2}<0$，所以 $\Theta(p_u)$ 是单调递增函数并且在起始点 $p_u=0$ 处函数值为负数。因此，依据文献[29-30]，本文设计了一种低复杂度的二分法，通过在每次迭代中计算 $\Theta(p_u)$ 进而求解最优解 p_u^*。

算法 1 二分法的用户发射功率分配算法

输入 用户最大发射功率上限 P_u

输出 用户发射功率 p_u^*

1) 计算 $\Theta(P_u)$
2) if $\Theta(P_u)\leqslant 0$ then
3) $\quad p_u^*=P_u$
4) else
5) $\quad \varepsilon>0$，初始化 $p_b=0$ 以及 $p_t=P_u$
6) $\quad$ 循环
7) $\quad\quad p_u^*=\dfrac{p_b+p_t}{2}$
8) $\quad\quad$ if $\Theta(p_u^*)\leqslant 0$ then
9) $\quad\quad\quad p_b=p_u^*$

10)　else
11)　　$p_t = p_u^*$
12)　end if
13)　until $p_t - p_b \leqslant \varepsilon$
14)　$p_u^* = \dfrac{p_b + p_t}{2}$
15)　end if

3.5.2　*边缘端计算资源分配问题*

边缘端计算资源分配优化问题的表达形式为

$$\begin{aligned} &\min_{\boldsymbol{f}} \ \Omega(f_u^e) \\ &\text{s.t. C3: } \sum_{u\in\mathcal{U}_e} f_u^e \leqslant f^e \\ &\quad\ \text{C4: } f_u^e > 0, \forall u \in \mathcal{U}_e \end{aligned} \tag{25}$$

其中，有

$$\Omega(f_u^e) = \sum_{u\in\mathcal{U}_e} \frac{\beta_u^t f_u^l}{f_u^e} \tag{26}$$

引理 2　当给定卸载决策向量 $\boldsymbol{x}$ 时，式(25)的优化问题是凸优化问题，且最优的资源分配 f_u^{e*} 和最优的目标函数值 $\Omega(f_u^{e*})$ 为

$$f_u^{e*} = \frac{f^e \sqrt{\beta_u^t f_u^l}}{\sum_{u\in\mathcal{U}_e} \sqrt{\beta_u^t f_u^l}}, \forall u \in \mathcal{U}_e \tag{27}$$

$$\Omega(f_u^{e*}) = \frac{\left(\sum_{u\in\mathcal{U}_e} \sqrt{\beta_u^t f_u^l}\right)^2}{f^e} \tag{28}$$

证明　详见附录 2。

3.5.3　*核心网传输带宽分配问题*

核心网传输带宽分配优化问题的表达形式为

$$\begin{aligned} &\min_{\boldsymbol{R}} \ \Phi(R_u^c) \\ &\text{s.t. C5: } \sum_{u\in\mathcal{U}_c} R_u^c \leqslant R^c \\ &\quad\ \text{C6: } R_u^c > 0, \forall u \in \mathcal{U}_c \end{aligned} \tag{29}$$

其中，有

$$\Phi(R_u^c) = \sum_{u\in\mathcal{U}_c} \frac{\beta_u^t d_u f_u^l}{c_u R_u^c} \tag{30}$$

引理 3　当给定卸载决策向量 $\boldsymbol{x}$ 时，式(29)的优化问题是凸优化问题，且最优的资源分配 R_u^{c*} 和最优的目标函数值 $\Phi(R_u^{c*})$ 为

$$R_u^{c*} = \frac{R^c \sqrt{\beta_u^t f_u^l \dfrac{d_u}{c_u}}}{\sum_{u\in\mathcal{U}_c} \sqrt{\beta_u^t f_u^l \dfrac{d_u}{c_u}}}, \forall u \in \mathcal{U}_c \tag{31}$$

$$\Phi(R_u^{c*}) = \frac{\left(\sum_{u\in\mathcal{U}_c} \sqrt{\beta_u^t f_u^l \dfrac{d_u}{c_u}}\right)^2}{R^c} \tag{32}$$

证明　证明过程与引理 2 的思路相同。

3.6　联合资源分配的任务卸载策略算法

在 3.5 节中，当给定卸载策略 $\boldsymbol{x}$ 时，能够求得上传发射功率 p_u、边缘计算资源 f_u^e 以及核心网传输带宽 R_u^c 的分配问题的最优解。根据式(17)～式(32)，可以得出

$$V^* = \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c} (\beta_u^t + \beta_u^e) - \Lambda(p_u^*) - \Omega(f_u^{e*}) - \sum_{u\in\mathcal{U}_c} \frac{\beta_u^t f_u^l}{f_u^c} - \Phi(R_u^{c*}) \tag{33}$$

将式(33)代入式(17)，式(17)的系统效用最大化的问题可以表示为

$$\begin{aligned} \max_{\boldsymbol{x}} \ & \sum_{u\in\mathcal{U}_e\cup\mathcal{U}_c} (\beta_u^t + \beta_u^e) - \Lambda(p_u^*) - \Omega(f_u^{e*}) - \\ & \sum_{u\in\mathcal{U}_c} \frac{\beta_u^t f_u^l}{f_u^c} - \Phi(R_u^{c*}) \\ \text{s.t.} \ & \text{C1: } x_{u,j} = \{0,1\}, u \in \mathcal{U}, j \in \{1,2\} \end{aligned} \tag{34}$$

定理 1　系统效用函数 V 为次模函数

证明　详见附录 3。

根据定理 1，上述系统效用最大化问题式(34)能够被证明是 NP-hard 问题[39-40]。针对该问题，本文提出了一种基于次模理论的贪心卸载策略算法，求解出问题式(34)的近似解[41-42]。

算法 2　基于次模理论的贪心卸载策略算法

输入　每个用户的发射功率 p_u^*，本地计算设备参数 P_u^l、f_u^l，用户的计算任务参数 d_u、c_u，云端为用户所分配的计算资源 f_u^c，核心网总传输带宽 R^c 以及 MEC 服务器计算资源 f^e

输出　卸载决策集 X^e 和 X^c

1) 初始化：卸载集 $X^e = \varnothing; X^c = \varnothing$
2) 循环
3)　对所有用户 $i \in \mathcal{U}$
4)　计算 $\Delta_i V(X^e \cup X^c)$；
5)　$i \leftarrow \arg\max \Delta_i V(X^e \cup X^c)$
6)　　if $(\Delta V_i \mid x_{i,1} = 1) > (\Delta V_i \mid x_{i,2} = 1)$
7)　　　set $X^e = X^e \cup \{i\}$ and $\mathcal{U} = \mathcal{U} \setminus \{i\}$
8)　　else
9)　　　set $X^c \cup \{i\}$ and $\mathcal{U} = \mathcal{U} \setminus \{i\}$

10)　end if

11) until $\Delta_i V(X^e \cup X^c) < 0$ or $\mathcal{U} = \varnothing$

3.7　算法时间复杂度分析

对于算法 1 二分法的用户发射功率分配算法，当 $\Omega(P_u) \geqslant 0$ 时，p_u^* 计算需要 $\left\lceil \mathrm{lb}\frac{P_u}{\varepsilon} \right\rceil$ 次的迭代收敛，因此可以得到用户发射功率分配算法的时间复杂度为 $O\left\lceil \mathrm{lb}\frac{P_u}{\varepsilon} \right\rceil$，其中 ε 为收敛的阈值。

对于算法 2 贪心卸载策略算法，所有用户的效用函数的计算需要 $O(n)$ 次迭代完成。在上述步骤的每一次计算迭代中，找到最大的 $\Delta_i V(X^e \cup X^c)$ 且保证 $\Delta_i V(X^e \cup X^c) > 0$ 以及 $U \neq \varnothing$ 情况下的时间复杂度为 $O(n)$。因此，算法 2 的时间复杂度为 $O(n^2)$。

4　仿真结果

本节通过仿真实验评估所提出的边云联合计算方案下的优化资源分配和多用户任务卸载决策算法的系统效用。具体仿真环境如下：假设 U 个用户随机均匀地分布在 200 m×200 m 的小区中，基站部署在小区的中心位置，N 为小区内覆盖用户的数量。用户计算任务输入数据大小 d_u 在 100～1 000 KB 随机均匀分布，所对应的完成任务需要的计算资源 c_u（CPU cycle 总数）在[0.2,1] Gcycle 随机均匀分布。考虑到用户本地设备计算能力的异构性，f_u^l 在{0.5 GHz, 0.8 GHz, 1.0 GHz}集合内等概率取值[20]。参考之前研究的用户设备功率参数值选择[4,30]，并结合当前最新相关的实测设备功率情况，本文选择的用户设备计算能力所对应的 P_u^l={0.5 W, 0.75 W, 0.9 W}[32]。设用户最大传输发射功率 P_u=100 mW，MeNB 到云端的上行总传输速率 R^c=100 Mbit/s[43]。其余相关仿真参数设置如表 1 所示。

表 1　　**仿真参数设置**

参数名称	取值
系统带宽 B/MHz	20
路径损失模型	128.1+37.5lg10(d)
背景噪声 σ_0^2 /dBm	−100
对数正态阴影衰落标准差/dB	10
边缘计算资源 f^e /GHz	20
云端分配给用户的计算资源 f_u^c /GHz	5

在同等的参数设置下，本节选择将本文提出的基于边云联合计算方案下的用户卸载策略的系统效用表现分别与以下方案进行对比。

1) 本地计算：所有的用户全部采用本地计算的方式来完成任务。

2) 基于边缘计算的联合资源优化的全卸载策略：所有的用户全部将任务卸载到边缘端执行，如文献[44-45]，并采用 3.4 节的优化资源分配方案。

3) 基于云端计算的联合资源优化的全卸载策略：所有的用户全部将任务卸载到云端执行，如文献[44-45]，并采用 3.4 节的优化资源分配方案。

不失一般性地，仿真结果均为各仿真实验重复 1 000 次并取平均的结果。

随着用户数量的变化，各方案的系统效用值的变化如图 2 所示。从图 2 可以看出，随着用户数量的增加，对比其他方案，本文提出的方案（以下简称本文方案）的系统效用值有很大的提升。当用户数量较少时，除本地计算方案外，其他各方案的系统效用值随用户数量的增加而增加，且本文方案系统效用值要高于其他 2 种方案。当用户总数超过某一阈值时，伴随着用户数量的增加，选择全卸载的边缘计算和云计算方案的系统效用值会逐渐下降，且当用户的数量超过一定阈值时，系统效用值低于本地计算方案。这是因为当卸载用户数量过多时，受限的上行通信资源和边缘计算资源不能为每个需求用户提供更丰富的资源需求，进而导致更高的计算时间和传输时延。用户通过全卸载的方式发送和执行任务将导致所有的用户对有限的资源进行竞争。当卸载用户过多时，边缘计算方案下的边缘节点分配给用户的计算资源会低于本地的计算资源，而云计算方案下用户较低的上行通信资源会导致任务卸载的传输时延过高，进而导致以上 2 种方案的系统效用降低甚至低于本地计算的效用。而随着用户增加，应用本文方案下的用户卸载策略选择算法仍能维持较高且稳定的系统效用值。这是因为该方案能够合理地为用户规划任务卸载的模式选择，从而保证有限的计算和通信资源的最大化利用。

为了评估本文方案针对应用任务时的系统效用性能，本文选择人脸识别这一特定的应用任务：该计算任务输入数据大小 d_u=420 KB，完成该任务需要的计算资源 c_u=1 000 Mcycle[30]，相关实验结果如图 3 所示。

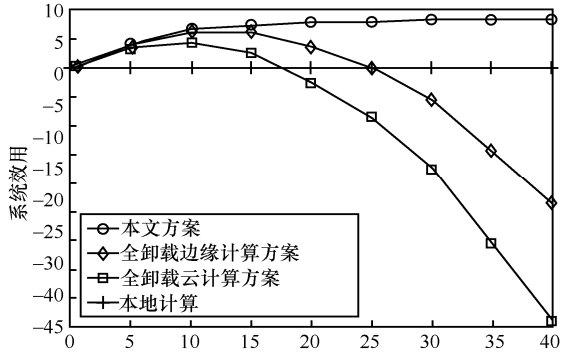

图 2　不同用户总数条件下各方案的系统效用

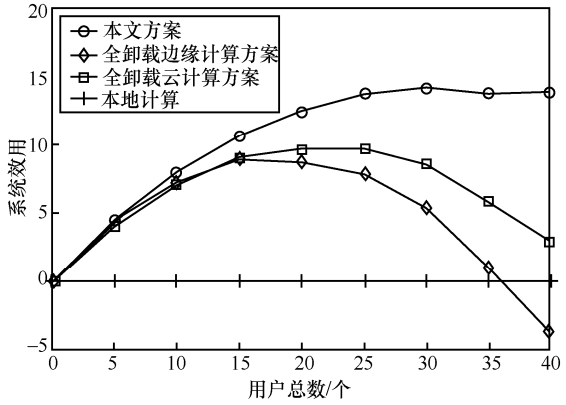

图 3　特定任务下的不同用户总数条件下各方案的系统效用

从图 3 可以看出，当针对人脸识别这一特定的任务时，本文方案的系统效用值始终高于全卸载的边缘计算方案和全卸载云计算方案。并且随着用户总数逐渐增多，全卸载边缘计算方案和全卸载云计算方案的系统效用值会逐渐下降，而本文方案依旧能够保持稳定较高的系统效用。本文方案通过联合边缘端、云端以及本地端的所有可用的计算和通信资源，在用户数量增加时依然保持系统效用平稳占优。

基于人脸识别这一特定的应用任务，在不同用户总数的条件下，本文方案中用户对边缘端、云端以及本地计算模式的任务卸载的决策选择的总数如图 4 所示。从图 4 可以看出，当用户数量较少时，用户更倾向于将任务卸载到边缘端和云端执行。随着用户数量的增加，选择将任务卸载到边缘端和云端的用户数量逐渐趋于平稳，用户主要由边缘端和云端计算模式向本地计算模式迁移。这是因为，随着用户数量的增加，受到上行通信资源和边缘计算资源的限制，边缘节点的计算资源和上行的通信资源逐渐不能满足用户的计算和通信需求，进而更多用户愿意选择计算资源稳定且没有传输时延的本地计算模式。

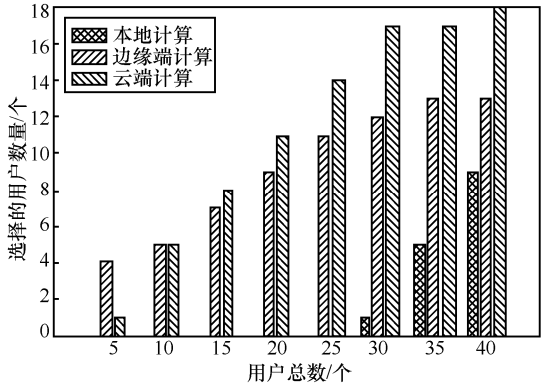

图 4　特定任务下的不同用户总数条件下各模式参与用户数

接下来，本文分析了在人脸识别这一特定的应用任务情况下，随着用户对时间偏好权重的变化，即 β_u^t 从 0.1 变化到 0.9 的过程中，所有用户的平均时间消耗情况，结果如图 5 所示。从图 5 可以看出，随着用户对时间偏好权重的增加，任务卸载的平均时间消耗逐渐降低。从不同用户数量下的任务平均时间消耗曲线能够看出，当系统中有更多的用户时，任务的平均时间消耗更多。这是由于当系统中用户数量增加时，将有更多的用户争夺有限的通信和计算资源，因此每个用户所获得的计算和通信资源会降低，进而导致通过任务卸载方式完成任务的平均时耗增加。

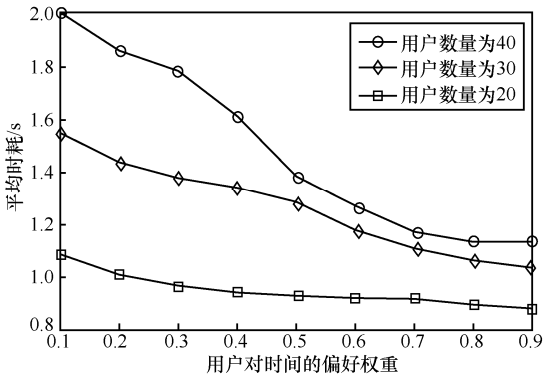

图 5　不同用户对时间偏好权重下的任务卸载平均时耗对比

5　结束语

本文提出了一种基于边云联合计算的多用户任务卸载方案。本文提出的方案通过联合考虑边缘计算以及云计算的相关优势特性，解决了多用户异构任务的卸载选择问题并实现了边缘端和云端资源的合理分配和充分利用。通过证明系统效用函数是次模函数，本文利用次模理论设计了一种基于边云联合计算的贪心的用户卸载策略选择算法，并联

合优化了边云联合计算框架下计算和通信资源的分配。仿真结果表明，在面对受限的通信资源和计算资源的条件下，且当卸载计算的用户数量增加时，本文提出的方案仍能保持稳定良好的系统效用。

附录 1 $\Lambda(p_u)$严格拟凸性证明

$\Lambda(p_u)$ 的计算式为

$$\Lambda(p_u)=\frac{\phi_u+\psi_u p_u}{\mathrm{lb}(1+p_u\gamma_u)} \tag{35}$$

式(35)在实数域内是二阶可微的。接下来，验证严格拟凸函数的二阶条件，即 p_0 点满足一阶导数 $\Lambda'(p_0)=0$ 且二阶导数 $\Lambda''(p_0)>0$ [37]。

首先， $\Lambda(p_u)$ 的一阶导数为

$$\Lambda'(p_u)=\frac{\psi_u\mathrm{lb}(1+p_u\gamma_u)-\dfrac{\gamma_u(\phi_u+\psi_u p_u)}{(1+p_u\gamma_u)\ln 2}}{\mathrm{lb}^2(1+p_u\gamma_u)} \tag{36}$$

$\Lambda(p_u)$ 的二阶导数为

$$\Lambda''(p_u)=\frac{\gamma_u\left\{[\gamma_u(\phi_u+\psi_u p_u)-2\psi_u(1+p_u\gamma_u)]\mathrm{lb}(1+p_u\gamma_u)+\dfrac{2\gamma_u(\phi_u+\psi_u p_u)}{\ln 2}\right\}}{(1+p_u\gamma_u)^2\ln 2\,\mathrm{lb}^3(1+p_u\gamma_u)} \tag{37}$$

当满足 $\Lambda'(p_0)=0$ 时，有

$$\Theta(p_0)=\psi_u\mathrm{lb}(1+p_0\gamma_u)-\frac{\gamma_u(\phi_u+\psi_u p_0)}{(1+p_0\gamma_u)\ln 2}=0 \tag{38}$$

将 p_0 代入式(37)，得到

$$\Lambda''(p_0)=\frac{{\gamma_u}^3(\phi_u+\psi_u p_0)^2}{(1+p_0\gamma_u)^3\psi_u\ln^2 2\,\mathrm{lb}^3(1+p_0\gamma_u)} \tag{39}$$

对 $\forall p_u\in(0,P_u]$，式(39)的分子分母是严格正的，即 $\Lambda''(p_0)>0$。因此 $\Lambda(p_u)$ 在定义域 $(0,P_u]$ 内是严格拟凸的。

证毕。

附录 2 引理 2 证明

对于目标问题式(25)，其关于 f_u^e 的二阶偏导为

$$\frac{\partial^2\Omega(f_u^e)}{\partial {f_u^e}^2}=\frac{2\beta_u^t f_u^l}{{f_u^e}^3}>0 \tag{40}$$

$$\frac{\partial^2\Omega(f_u^e)}{\partial f_u^e\,\partial f_v^e}=0,\forall u\neq v \tag{41}$$

因此，目标函数海森矩阵 $\boldsymbol{H}$ 是对称正定矩阵，根据参考文献[37]，可以得出式(25)是凸优化问题，且可以应用 KKT 条件进行求解。

式(25)的拉格朗日函数为

$$\mathcal{L}(\Omega,\lambda)=\sum_{u\in\mathcal{U}_e}\frac{\beta_u^t f_u^l}{f_u^e}+\lambda\left(\sum_{u\in\mathcal{U}_e}f_u^e-f^e\right) \tag{42}$$

对拉格朗日函数 $\mathcal{L}(\Omega,\lambda)$ 求关于 f_u^e 的偏导得到

$$\frac{\partial\mathcal{L}(\Omega,\lambda)}{\partial f_u^e}=\lambda-\frac{\beta_u^t f_u^l}{{f_u^e}^2} \tag{43}$$

当 $f_u^e={f_u^e}^*$ 时，满足 $\dfrac{\partial\mathcal{L}(\Omega,\lambda)}{\partial f_u^e}=0$。因此

$${f_u^e}^*=\sqrt{\frac{\beta_u^t f_u^l}{\lambda^*}} \tag{44}$$

其中， $\lambda^*>0$。

根据 KKT 条件，可以得到

$$\sum_{u\in\mathcal{U}_e}{f_u^e}^*=f^e \tag{45}$$

将式(44)代入式(45)，可以得到

$$\lambda^*=\left(\frac{\sum\limits_{u\in\mathcal{U}_e}\sqrt{\beta_u^t f_u^l}}{f^e}\right)^2 \tag{46}$$

将式(46)代入式(44)，可以得到 ${f_u^e}^*$ 的解析解为

$${f_u^e}^*=\frac{f^e\sqrt{\beta_u^t f_u^l}}{\sum\limits_{u\in\mathcal{U}_e}\sqrt{\beta_u^t f_u^l}},\forall u\in\mathcal{U}_e \tag{47}$$

因此，式(25)的最优的目标函数值 $\Omega({f_u^e}^*)$ 为

$$\Omega({f_u^e}^*)=\frac{\left(\sum\limits_{u\in\mathcal{U}_e}\sqrt{\beta_u^t f_u^l}\right)^2}{f^e} \tag{48}$$

证毕。

附录 3 效用函数为次模函数证明

针对系统效用函数 V，当增加元素 i 到用户卸载选择集合 $U=\mathcal{U}_e\cup\mathcal{U}_c$ 中时，即其边际效用值 $\Delta_i V(U)$ 可以表示为

$$\begin{aligned}\Delta_i V(U)&=\sum_{u\in\mathbf{U}\cup\{i\}}V_u-\sum_{u\in\mathbf{U}}V_u\\&=(\beta_i^t+\beta_i^e)-\Delta(i)-\Delta(i\,|\,U)\end{aligned} \tag{49}$$

其中，有

$$\Delta(i)=\begin{cases}\Delta(i)^e=\dfrac{\phi_i+\psi_i p_i}{\mathrm{lb}(1+p_i\gamma_i)}+\dfrac{\beta_i^t f_i^l}{f^e},x_{i,1}=1\\[2ex]\Delta(i)^c=\dfrac{\phi_i+\psi_i p_i}{\mathrm{lb}(1+p_i\gamma_i)}+\dfrac{\beta_i^t f_i^l}{f_i^c}+\dfrac{\beta_i^t f_i^l\dfrac{d_i}{c_i}}{R^c},x_{i,2}=1\end{cases} \tag{50}$$

$$\Delta(i\,|\,U)=\begin{cases}\Delta(i\,|\,U)^{e}=\dfrac{2\sqrt{\beta_i^t f_i^l}\sum\limits_{u\in\mathcal{U}_e}\sqrt{\beta_u^t f_u^l}}{f^{e}}, x_{i,1}=1\\ \Delta(i\,|\,U)^{c}=\dfrac{2\sqrt{\beta_i^t f_i^l\dfrac{d_i}{c_i}}\sum\limits_{u\in\mathcal{U}_c}\sqrt{\beta_u^t f_u^l\dfrac{d_u}{c_u}}}{R^{c}}, x_{i,2}=1\end{cases} \tag{51}$$

根据式(50)和式(51)，对于 $\Delta(i)$，当用户选择将任务卸载到边缘或云端计算时，即 $x_{i,1}=1$ 或 $x_{i,2}=1$ 的情况下，$\Delta(i)$ 是常数，且不随用户卸载集合 U 的变化而变化。对于 $\Delta(i\,|\,U)$，当用户选择边缘计算时，即 $x_{i,1}=1, x_{i,2}=0$。此时 $\Delta(i\,|\,U)$ 随着用户卸载集 $\mathcal{U}_e$ 的增长而增长，随用户卸载集 $\mathcal{U}_c$ 的增长保持不变。而当用户选择云端计算时，即 $x_{i,1}=0, x_{i,2}=1$，此时 $\Delta(i\,|\,U)$ 随着用户卸载集 $\mathcal{U}_c$ 的增长而增长，随用户卸载集 $\mathcal{U}_e$ 的增长保持不变。所以，随着用户卸载集 $U=\mathcal{U}_e\cup\mathcal{U}_c$ 的增长，$\Delta(i\,|\,U)$ 是单调非递减函数。根据式(49)可以得出，对于卸载集合 U，$\Delta_i V(U)$ 是单调非递增函数。因此，对于卸载集 U、W，当 $W\subseteq U$，且 $i\notin U$，$\Delta_i V(W)\geqslant\Delta_i V(U)$，即 $V(W\cup\{i\})-V(W)\geqslant V(U\cup\{i\})-V(U)$。因此，效用函数 V 是次模的[39-41]。

证毕。

参考文献：

[1] SHI W, CAO J, ZHANG Q, et al. Edge computing: vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[2] Cisco. Cisco annual internet report (2018–2023) white paper[R]. (2020-03-09)[2020-07-16].

[3] PULIAFITO C, MINGOZZI E, LONGO F, et al. Fog computing for the Internet of things: a survey[J]. ACM Transactions on Internet Technology, 2019, 19(2): 1-41.

[4] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2016, 24(5): 2795-2808.

[5] CHEN W, WANG D, LI K. Multi-user multi-task computation offloading in green mobile edge cloud computing[J]. IEEE Transactions on Services Computing, 2018, 12(5): 726-738.

[6] PATEL M, NAUGHTON B, CHAN C, et al. Mobile-edge computing introductory technical white paper[R]. Mobile-Edge Computing (MEC) Industry Initiative, (2018-09-14)[2020-07-16].

[7] ETSI. ETSI first meeting of new standardization group on mobile-edge-computing[R]. (2014)[2020-07-16].

[8] PLACHY J, BECVAR Z, STRINATI E C. Dynamic resource allocation exploiting mobility prediction in mobile edge computing[C]//2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Piscataway: IEEE Press, 2016: 1-6.

[9] JI W, LIANG B, WANG Y, et al. Crowd V-IoE: visual Internet of everything architecture in AI-driven fog computing[J]. IEEE Wireless Communications, 2020, 27(2): 51-57.

[10] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.

[11] JI W, DUAN L Y, HUANG X, et al. Astute video transmission for geographically dispersed devices in visual IoT systems[J]. IEEE Transactions on Mobile Computing, DOI:10.1109/TMC.2020.3009745, 2020.

[12] 董思岐, 李海龙, 屈毓锛, 等. 移动边缘计算中的计算卸载策略研究综述[J]. 计算机科学, 2019, 46(11): 32-40.
DONG S Q, LI H L, QU Y B, et al. Survey of research on computation unloading strategy in mobile edge computing[J]. Computer Science, 2019, 46(11): 32-40.

[13] 吴大鹏, 吕吉, 李职杜, 等. 移动性感知的边缘服务迁移策略[J]. 通信学报, 2020, 41(4): 1-13.
WU D P, LYU J, LI Z D, et al. Mobility aware edge service migration strategy[J]. Journal on Communications, 2020, 41(4): 1-13.

[14] ZHANG J, HU X, NING Z, et al. Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching[J]. IEEE Internet of Things Journal, 2018, 6(3): 4283-4294.

[15] 满君丰, 赵龙乾, 彭成, 等. 云边协同计算架构下大规模工厂接入的任务调度方法[J]. 计算机集成制造系统, (2020-05-06) [2020-07-16].
MAN J F, ZHAO L Q, PENG C, et al. Task scheduling method for large-scale factory access in cloud and edge collaborative computing architecture[J]. Computer Integrated Manufacturing Systems, (2020-05-06) [2020-07-16].

[16] 汤闻达. 支持云雾端应用集成的资源调度策略及其优化技术[D]. 南京: 南京大学, 2019.
TANG W D. Resource scheduling strategies and optimization techniques supporting cloud-fog-thing integration[D]. Nanjing: Nanjing University, 2019.

[17] CUI Y, ZHANG D, ZHANG T, et al. Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices[J]. AEU-International Journal of Electronics and Communications, 2020: 153134.

[18] WANG Y, LANG P, TIAN D, et al. A game-based computation offloading method in vehicular multi-access edge computing networks[J]. IEEE Internet of Things Journal, 2020, 7(6): 4987-4996.

[19] CARDELLINI V, PERSONÉ V D N, DI VALERIO V, et al. A game-theoretic approach to computation offloading in mobile cloud computing[J]. Mathematical Programming, 2016, 157(2): 421-449.

[20] ZHENG J, CAI Y, WU Y, et al. Dynamic computation offloading for mobile cloud computing: a stochastic game-theoretic approach[J]. IEEE Transactions on Mobile Computing, 2018, 18(4): 771-786.

[21] 赵临东, 庄文芹, 陈建新, 等. 异构蜂窝网络中分层任务卸载: 建模与优化[J]. 通信学报, 2020, 41(4): 34-44.
ZHAO L D, ZHUANG W Q, CHEN J X, et al. Hierarchical task offloading in heterogeneous cellular network: modeling and optimization[J]. Journal on Communications, 2020, 41(4): 34-44.

[22] YANG X, YU X, HUANG H, et al. Energy efficiency based joint computation offloading and resource allocation in multi-access MEC systems [J]. IEEE Access, 2019, 7: 117054-117062.

[23] SALEEM U, LIU Y, JANGSHER S, et al. Latency minimization for D2D-enabled partial computation offloading in mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4472-4486.

[24] CHEN M H, LIANG B, DONG M. Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point[C]//IEEE INFOCOM 2017-IEEE Conference

on Computer Communications. Piscataway: IEEE Press, 2017: 1-9.
[25] LONG L, LIU Z, ZHOU Y, et al. Delay optimized computation offloading and resource allocation for mobile edge computing[C]//2019 IEEE 90th Vehicular Technology Conference. Piscataway: IEEE Press, 2019: 1-5.
[26] CHEN S, ZHENG Y, WANG K, et al. Delay guaranteed energy-efficient computation offloading for industrial IoT in fog computing[C]//2019 IEEE International Conference on Communications. Piscataway: IEEE Press, 2019: 1-6.
[27] ZHAO P, TIAN H, QIN C, et al. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing[J]. IEEE Access, 2017, 5: 11255-11268.
[28] ZHANG K, MAO Y, LENG S, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks[J]. IEEE Access, 2016, 4: 5896-5907.
[29] LYU X, TIAN H, SENGUL C, et al. Multiuser joint task offloading and resource optimization in proximate clouds[J]. IEEE Transactions on Vehicular Technology, 2016, 66(4): 3435-3447.
[30] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2018, 68(1): 856-868.
[31] ZHANG J, XIA W, YAN F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing[J]. IEEE Access, 2018, 6: 19324-19337.
[32] GENG Y, YANG Y, CAO G. Energy-efficient computation offloading for multicore-based mobile devices[C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2018: 46-54.
[33] CHEN X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(4): 974-983.
[34] GUO S, XIAO B, YANG Y, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. Piscataway: IEEE Press, 2016: 1-9.
[35] POCHET Y, WOLSEY L A. Production planning by mixed integer programming[M]. Berlin: Springer Science & Business Media, 2006.
[36] TAMMER K. The application of parametric optimization and imbedding to the foundation and realization of a generalized primal decomposition approach[J]. Mathematical research, 1987, 35: 376-386.
[37] BOYD S, BOYD S P, VANDENBERGHE L. Convex optimization[M]. Cambridge: Cambridge University Press, 2004.
[38] BEREANU B. Quasi-convexity, strictly quasi-convexity and pseudo-convexity of composite objective functions[J]. Revue Francaise D Automatique Informatique Recherche Operationnelle, 2009, 6(R1): 15-26.
[39] FUJISHIGE S. Submodular functions and optimization[M]. Amsterdam: Elsevier, 2005.
[40] FEIGE U, MIRROKNI V S, VONDRÁK J. Maximizing non-monotone submodular functions[J]. SIAM Journal on Computing, 2011, 40(4): 1133-1153.
[41] JI W, ZHU W. Profit maximization for sponsored data in wireless video transmission systems[J]. IEEE Transactions on Mobile Computing, 2020, 19(8): 1928-1942.
[42] KHULLER S, MOSS A, NAOR J S. The budgeted maximum coverage problem[J]. Information Processing Letters, 1999, 70(1): 39-45.
[43] ZHANG J, HU X, NING Z, et al. Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching[J]. IEEE Internet of Things Journal, 2018, 6(3): 4283-4294.
[44] SARDELLITTI S, SCUTARI G, BARBAROSSA S. Distributed joint optimization of radio and computational resources for mobile cloud computing[C]//2014 IEEE 3rd International Conference on Cloud Networking. Piscataway: IEEE Press, 2014: 211-216.
[45] SARDELLITTI S, SCUTARI G, BARBAROSSA S. Joint optimization of radio and computational resources for multicell mobile-edge computing[J]. IEEE Transactions on Signal and Information Processing over Networks, 2015, 1(2): 89-103.

[作者简介]

梁冰（1992− ），男，河北保定人，中国科学院计算技术研究所博士生，主要研究方向为多媒体通信与网络、视频传输、边缘计算、机器学习等。

纪雯（1976− ），女，陕西西安人，博士，中国科学院计算技术研究所研究员、博士生导师，主要研究方向为多媒体通信与网络，包括视频传输和编码、优化理论和信息论、边缘计算、多媒体经济学和智能计算等。