移动边缘计算中基于资源联合分配的部分计算卸载方法

刘耀[1,2]，何岳园[3]，周红静[3]，李超良[3]，李闯[2,3]
（1. 湖南工商大学数字传媒与人文学院，湖南 长沙 410205；2. 长沙人工智能社会实验室，湖南 长沙 410205；
3. 湖南工商大学计算机学院，湖南 长沙 410205）

摘 要：为了满足用户计算密集型任务的需求，解决移动终端计算资源和能量有限的问题，针对正交频分多址的多用户移动边缘计算系统，以任务时延为主要优化目标，设置任务时限、设备能量、通信资源等约束条件，提出了一个结合通信和边缘服务器计算资源分配的部分卸载方法。该方法在满足用户最低时延的条件下设置初始卸载比和分配通信资源，然后根据服务器计算能力来分配剩余计算资源，最后根据资源分配情况优化卸载比。仿真结果表明，该方法能够减少任务计算的时延和移动终端的能量消耗。

关键词：移动边缘计算；计算卸载；资源分配
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.2096−3750.2023.00313

Partial computation offloading method based on joint resource allocation for mobile edge computing

LIU Yao[1,2], HE Yueyuan[3], ZHOU Hongjing[3], LI Chaoliang[3], LI Chuang[2,3]
1. School of Digital Media and Humanities, Hunan University of Technology and Business, Changsha 410205, China
2. Changsha Social Laboratory of Artificial Intelligence, Changsha 410205, China
3. School of Computer Science, Hunan University of Technology and Business, Changsha 410205, China

Abstract: In order to meet the requirements of users for computing-intensive tasks and solve the problems of limited computing resources and energy of mobile terminals, a partial offloading method was proposed for multi-user mobile edge computing system with orthogonal frequency division multiple access by setting constraints such as task delay, device energy and communication resources, aiming at optimizing task delay. The initial offloading ratio was set and the communication resource was allocated under the condition of satisfying the user's minimum delay. And then the remaining computing resource was allocated according to the server's computing capability. Finally, the offload ratio was optimized according to the resource allocation. Simulation results show that this method can reduce the delay of task computing and the energy consumption of mobile terminals.

Key words: mobile edge computing, computation offloading, resource allocation

---

收稿日期：2022−08−08；**修回日期**：2022−11−21
通信作者：周红静，hjzhou@hutb.edu.cn
基金项目：国家自然科学基金资助项目（No.62002115）；湖南省教育厅科学研究项目（No.20C0538，No.21A0376）；湖南省自然科学基金资助项目（No.2020JJ4250）；湖南省社会科学成果评审委员会课题（No.XSP21YBZ113）；青年教师创新驱动计划项目（No.2020QD08）
Foundation Items: The National Natural Science Foundation of China (No.62002115), The Scientific Research Project of the Department of Education of Hunan Province (No.20C0538, No.21A0376), The Natural Science Foundation of Hunan Province (No.2020JJ4250), The Project of Hunan Social Science Achievement Appraisal Committee (No.XSP21YBZ113), The Project of Innovation Driven Program for Young Teachers (No.2020QD08)

0　引言

随着 5G 网络和物联网（IoT, internet of things）技术的迅猛发展，交互式在线游戏、人脸识别、虚拟现实、增强现实等计算密集型应用大量涌现，巨大的流量给主干网络带来了严重的负担[1]。移动边缘计算（MEC, mobile edge computing）[2-4]为避免将应用数据传输到远程的云端，将计算、存储等服务的资源部署到靠近用户的网络边缘，及时响应用户请求，给用户带来更好的体验。移动终端的计算和存储资源有限，不能够保障这些应用的高性能计算需求。计算卸载[5-7]通过卸载部分或全部计算任务到边缘服务器而不是云服务器来缩短处理时延，是解决移动终端计算能力不足、延长使用时间的有效手段。如何在移动终端和边缘服务器之间实现协同计算是当前研究的热点问题。

针对任务之间的不同时延敏感性，文献[8]提出了一种基于反应器的深度强化学习模型来自适应地卸载任务，控制卸载速率同时平衡了截止日期受限和时延敏感任务之间的卸载比。文献[9]提出将任务划分为子任务，把卸载问题转化为子任务和服务器资源两个集合之间的映射问题，并提出了一种基于堆栈的缓存机制来保障服务器资源分配的公平性。文献[10]在综合考虑任务时延和设备能耗的卸载决策模型的基础上，提出了基于移动用户信誉值的计算资源博弈分配模型合理分配计算资源。文献[11]针对异构环境中任务对时延和能耗要求的差异，采用分布式博弈和李雅普诺夫优化理论，提出了一种分布式多样化异构任务卸载算法。文献[12]联合优化多个边缘服务器之间以及边缘服务器与云服务器之间的计算和通信资源，通过二次约束二次规划得到了一个近似解来保证任务的时延。文献[13]将计算卸载转换为未知先验统计知识下时延约束的长期随机优化问题，全面分析了最优性差距、最坏情况下的时延和系统参数的影响。

将计算卸载和各类资源联合优化来提升 MEC 系统的总体性能得到了大量的关注[14-23]。文献[14]针对大计算量任务场景构建了分层边缘云计算架构，以终端设备能耗和任务执行时延为优化目标，提出了基于拉格朗日乘子法的集中式计算卸载方法和边缘移动节点的资源最优分配机制。文献[15]针对多用户、多任务和多网关多个边缘服务器的场景，提出了一种基于双深度 Q 学习的方法分配系统的通信、存储和计算资源并最小化平均能耗。文献[16]基于 Stackelberg 博弈模型分析了移动终端如何自主使用运营商的通信和计算资源，提出了一种分散近似计算卸载决策算法，从而实现移动终端任务完成时间的最小化。针对任务执行之前不完全了解计算需求的情况，文献[17]提出了一个联合优化卸载决策和计算与通信资源分配的方法，以最小化平均成本和成本变化的加权和。文献[18]提出了一种基于深度学习的联合卸载决策和资源分配算法，综合考虑了优化卸载决策、本地 CPU、带宽和外部 CPU 占用率来降低任务的时延和设备的能耗。文献[19]针对超密集网络场景，结合卸载策略、信道分配以及功率分配提出了一个系统开销最小化模型。文献[20]在多层协作边缘计算环境中，考虑用户设备服务需求和电量状态，提出了一个比例公平任务卸载和资源分配框架。文献[21]基于多背包问题提出一种 K 均值聚类和遗传算法结合的计算卸载方法来解决当要执行的任务的计算总量超过 MEC 系统的总计算容量时任务溢出的情况。

部分卸载[24]能够在有限带宽的情况下将部分任务上传到边缘服务器计算，有效并行利用移动终端和服务器的计算资源。文献[25]研究了具有二进制计算卸载策略的多用户无线边缘计算网络中的加权和计算速率最大化问题，并将问题转化为用户个体计算模式选择和系统传输时间分配的联合优化。文献[26]将无线电力移动边缘云引入部分卸载，以解决低功耗设备的电池和计算能力。文献[27]将任务卸载和资源分配形式化为一个开销最小化问题，使用拉格朗日乘子法导出计算、带宽分配问题的近似形式，提出了一种低计算复杂度的任务卸载策略。文献[28]考虑计算任务之间的依赖关系，提出了一种面向多用户细粒度的计算卸载调度方法优化能耗和时延。文献[29]在正交频分多址（OFDMA, orthogonal frequency division multiple access）的多用户 MEC 场景中，联合考虑部分卸载和系统资源分配，提出了混合整数非线性规划方法，使用多信道传输解决计算卸载总时延的最小化和节省能量的问题。文献[30]研究了 MEC 系统中移动终端的时延成本和能量成本加权和的优化问题，提出了一种基于深度强化学习的算法以获得最优调度，而无须事先了解任务到达、可采用的再生能

量以及信道条件。

综上，现有的资源分配和计算卸载联合优化的研究中，主要考虑的是完全卸载的场景，对部分卸载与资源联合优化的研究还较少，特别是当用户数量增加、计算量增大，而边缘服务器计算容量也有限的情况下，如何优化卸载的计算数据量。本文针对 OFDMA 的多用户 MEC 系统，在联合分配计算资源和网络通信资源的同时，通过优化移动终端任务的卸载比充分利用边缘服务器的计算资源，从而减少任务完成的时延并减轻移动终端的能耗。

1 系统模型

考虑一个单个小区的多用户 MEC 系统由一个基站和多个移动用户组成，每个用户使用一个移动终端，基站能够覆盖 N 个独立的用户，用户集合为 U。边缘服务器部署在基站，为用户提供有限的计算和存储服务。当用户无法满足其应用对时延和能耗的要求时，将任务部分或全部卸载到边缘服务器上运行。假设 MEC 系统已知信道增益、计算数据量、计算能耗等相关信息，并以此确定每个用户卸载的数据量以及分配的信道和计算资源，从而降低用户的计算时延和能耗。

1.1 本地计算模型

每个用户 k 都有一个时延受限的计算密集型任务需要完成，任务的最大时延为 $T_k^{\max}$，最大能耗为 $E_k^{\max}$，任务的数据大小为 S_k（单位：比特），在用户端计算每个比特所需的 CPU 周期为 C_k。当用户 k 的计算能力无法满足任务的时延或能耗的要求时，可以将任务全部或部分卸载到基站的边缘服务器上，卸载的比例为 $\lambda_k \in [0,1]$。因此，任务在本地运行的时间为 t_k^{local}，表示为

$$t_k^{\text{local}} = (1-\lambda_k)\frac{S_k C_k}{f_k^0} \tag{1}$$

其中，f_k^0 表示用户 k 的计算能力，单位为 CPU 周期/s。

用户 k 的每个 CPU 周期的能量消耗为 e_k^0。当部分任务留在本地执行时，所需要的能耗为 e_k^{local}，表示为

$$e_k^{\text{local}} = (1-\lambda_k) S_k C_k e_k^0 \tag{2}$$

1.2 数据通信模型

假设系统中的信道是 Rayleigh 衰减信道，采用 OFDMA 的方式传输数据，信道的集合 CH 共包含 M 个子信道，信道之间没有干扰，每个子信道在同一时刻只能给一个用户分配传输任务。每个用户能够分配多个子信道，$\boldsymbol{\delta}$ 表示子信道分配矩阵，$\delta_k^i \in \{0,1\}$ 表示用户 k 和子信道 i 之间的关系，$\delta_k^i = 1$ 表示子信道 i 分配给用户 k，$\delta_k^i = 0$ 表示子信道 i 没有分配给用户 k。g_k^i 表示用户 k 在信道 i 上的信道增益强度。$\boldsymbol{P}$ 表示功率分配矩阵。p_k^i 表示用户 k 在信道 i 上的传输功率，最大传输功率为 $P_k^{\max}$。因此，用户 k 在子信道 i 上的数据传输速率为 r_k^i，表示为

$$r_k^i = B^i \text{lb}\left(1 + \frac{p_k^i g_k^i}{N_0 B^i}\right) \tag{3}$$

其中，N_0 表示高斯白噪声的功率谱密度，B^i 表示每个子信道 i 的带宽。

用户 k 的最小数据传输速率为 $R_k^{\min}$，用户 k 被分配多个子信道传输数据时，总的数据传输速率 r_k 为

$$r_k = \sum_{i=1}^{M} \delta_k^i r_k^i \geqslant R_k^{\min} \tag{4}$$

假设用户 k 的卸载数据 S_k^{off} 按照信道上数据传输速率比值分配在各个子信道上。因此，用户 k 在其子信道 i 上卸载的数据量为 $S_k^{i,\text{off}}$，$S_k^{i,\text{off}} = \lambda_k S_k r_k^i / r_k$。由于多信道传输，用户 k 卸载时延 t_k^{off} 由最坏信道的传输时延决定，表示为

$$t_k^{\text{off}} = \max(t_k^{i,\text{off}}) = \max\left(\frac{S_k^{i,\text{off}}}{r_k^i}\right),\ i \in \{1,2,\cdots,M\} \tag{5}$$

其中，$t_k^{i,\text{off}}$ 表示用户 k 在子信道 i 上的传输时延。

用户 k 传输任务消耗的总能量 e_k^{off} 为

$$e_k^{\text{off}} = \sum_{i=1}^{M} \delta_k^i t_k^{i,\text{off}} p_k^i = \sum_{i=1}^{M} \frac{\delta_k^i S_k^{i,\text{off}} p_k^i}{r_k^i} \tag{6}$$

1.3 边缘服务器计算模型

部署在基站的边缘服务器具有有限的计算能力 f^{srv}，每个计算资源为 $f_{\text{unit}}^{\text{srv}}$，单位为 CPU 周期/s。在边缘服务器计算每个比特所需的 CPU 周期为 C^{srv}。服务器为用户 k 分配的计算资源 f_k^{srv}，所以用户 k 卸载的数据在边缘服务器运行的时间 t_k^{srv} 为

$$t_k^{\text{srv}} = \frac{S_k^{\text{off}} C^{\text{srv}}}{f_k^{\text{srv}}} = \frac{\lambda_k S_k C^{\text{srv}}}{f_k^{\text{srv}}} \tag{7}$$

1.4　问题表述

用户将一部分任务卸载到边缘服务器，并从边缘服务器下载计算结果，同时计算剩余任务。由于计算结果很小，计算结果下载时间不予考虑。任务执行的时间只考虑卸载任务上传时间 t_k^{off}、边缘服务器执行时间 t_k^{srv} 和剩余任务在本地的执行时间 t_k^{local}。因为本地计算和边缘服务器计算的并行性，完成任务总的计算时间为 t_k^{total}，表示为

$$t_k^{\text{total}} = \max\left(t_k^{\text{local}}, t_k^{\text{off}} + t_k^{\text{srv}}\right) \tag{8}$$

将 MEC 中时延和能耗限制下结合资源分配的多用户部分卸载当作一个优化问题考虑。系统优化目标是最小化任务完成总时延。问题优化目标表示为

$$\min_{\lambda,\delta,p} \sum_{k=1}^{N} \max\left(\frac{(1-\lambda_k)S_k C_k}{f_k^0}, \max\left(\frac{S_k^{i,\text{off}}}{r_k^i}\right) + \frac{\lambda_k S_k C^{\text{srv}}}{f_k^{\text{srv}}}\right) \tag{9}$$

$$\text{s.t. C1:}\ \ 0 \leqslant \lambda_k \leqslant 1,\ k \in \{1,2,\cdots,N\}$$

$$\text{C2:}\ \ e_k^{\text{local}} + e_k^{\text{off}} < E_k^{\max}$$

$$\text{C3:}\ \ \delta_k^i \in \{0,1\},\ \sum_{k=1}^{N} \delta_k^i = 1, \forall i \in \{1,2,\cdots,M\}$$

$$\text{C4:}\ \ \sum_{i=1}^{M} \delta_k^i \geqslant 1, \forall k \in \{1,2,\cdots,N\}$$

$$\text{C5:}\ \ \sum_{i=1}^{M} \delta_k^i r_k^i \geqslant R_k^{\min}, \forall k \in \{1,2,\cdots,N\}$$

$$\text{C6:}\ \ \sum_{i=1}^{M} \delta_k^i p_k^i \leqslant P_k^{\max},\ \forall k \in \{1,2,\cdots,N\}$$

$$\text{C7:}\ \ \sum_{k=1}^{N} f_k^{\text{srv}} \leqslant f^{\text{srv}}, \forall k \in \{1,2,\cdots,N\}$$

其中，C1 表示每个用户的卸载比范围；C2 表示用户在所给的卸载决策下，其能耗不能超过限定的范围；C3 和 C4 是用户和信道资源分配的约束，表示每个信道在同一时刻只为一个用户服务，一个用户至少分配一个子信道；C5 表示信道分配必须保障传输质量要求，用户的总传输速率不能低于最低阈值；C6 是对用户分配多个信道传输时功率的限制，总的传输功率不得高于用户的最大传输功率；C7 表示所有用户分配的计算资源不得超过服务器总的计算资源。

2　资源联合分配的部分计算卸载

系统的优化目标是最小化任务完成的总时延，从式(9)可以看出，该优化问题主要涉及卸载比和系统资源分配。因此，本文将该优化问题分解为卸载比设定和资源分配两个方面处理。首先，根据每个用户时延和能耗限制，给出一个初始卸载比，然后提出一种低复杂度的信道资源和服务器计算资源的联合分配策略，最后根据当前可用资源更新卸载比。

2.1　初始化卸载比

首先需要根据用户低时延的需求为用户设置初始卸载比。每个用户任务都有一个时延的限制，根据任务卸载部分和本地计算部分的时延都不得高于任务时延的限制，可以得到

$$\begin{cases} \dfrac{\lambda_k S_k}{R_k^{\min}} + \dfrac{\lambda_k S_k C^{\text{srv}}}{f_k^{\text{srv}}} \leqslant T_k^{\max} \\ \dfrac{(1-\lambda_k) S_k C_k}{f_k^0} \leqslant T_k^{\max} \end{cases} \tag{10}$$

从系统整体来看，单个用户卸载量越小，网络资源以及服务器资源可以为越多的用户提供服务。因此在确定初始卸载比时，应尽可能地选择小的卸载比。每个用户根据其对时延的限制求出的最小的卸载比为

$$\lambda_k = \min\left(1 - \frac{f_k^0 T_k^{\max}}{S_k C_k},\ \frac{f_k^{\text{srv}}\left(T_k^{\max} - \dfrac{S_k^{i,\text{off}}}{R_k^{\min}}\right)}{S_k C^{\text{srv}}}\right) \tag{11}$$

2.2　资源联合分配

1) 信道资源分配

信道资源的分配分为两轮。第一轮为每个用户分配一个子信道，每次根据式(12)选择子信道 i^* 分配给用户 k^*，使该用户数据传输速率最大化。由于每个用户只有一个子信道，其功率为最大功率。

$$\left(k^*, i^*\right) = \arg \max_{\substack{k\in\{1,2,\cdots,N\},\\ i\in\{1,2,\cdots,M\}}} \left(\frac{r_k^i}{\dfrac{1}{N}\sum_{k=1}^{N} r_k^i}\right) \tag{12}$$

第二轮分配剩余的子信道时，优先分配给具有最小 $\dfrac{r_k}{R_k^{\min}}$ 值的用户 k_0。为了使信道分配给用户带来更大的速率收益，根据式(12)为用户选择信道 i^*，在每次分配子信道的同时，也会更新功率分配，并

计算所有用户的总速率。本文为每个子信道携带相同数量数据的用户执行统一的功率分配方案。随后的迭代旨在提高所有用户的总速率，从而减少卸载时延。

2) 服务器计算资源分配

首先根据以下时延限制条件对计算资源进行初步的分配：一方面，卸载部分数据的卸载时延不得超过本地计算的时延，即 $t_k^{\text{off}} < t_k^{\text{local}}$；另一方面，整个任务的计算时延不得超过总的时延限制，即 $t_k^{\text{total}} < T_k^{\max}$。第二次计算资源分配是对剩余可用计算资源的分配，在此定义用户的计算速率差为目标函数 σ，即将一个计算资源分配给用户 k，计算分配这份计算资源前后的计算速率差值，表示为

$$\sigma = \frac{S_k}{t_k^{\text{local}} + t_k^{\text{off}} + t_k^{\text{srv+1}}} - \frac{S_k}{t_k^{\text{local}} + t_k^{\text{off}} + t_k^{\text{srv}}} \tag{13}$$

其中，$t_k^{\text{srv+1}}$ 是服务器为用户 k 分配一个新的计算资源后服务器的计算时间。对每个用户获得计算资源前后的计算速率差值进行比较，为用户添加计算资源能获得最大利益的用户分配一个计算资源，使得 $k^* = \arg\max(\sigma)$。

资源联合分配算法见算法 1。

算法 1 资源联合分配算法

输入：用户信息、信道信息、服务器信息

输出：信道分配矩阵、功率分配矩阵

初始化任务信息、用户信息、信道信息、边缘服务器计算资源

//信道资源分配

while $U \neq \phi$ do

根据式(12)选取信道 i^* 与用户 k^* 进行配对

计算用户的传输速率并设置最大发射功率:

$r_{k^*} = r_{k^*}^{i^*}$， $p(k^*, i^*) = \boldsymbol{P}_{k^*}^{\max}$

$\text{CH} = \text{CH} - \{ i^* \}$， $U = U - \{ k^* \}$

end while

while $\text{CH} \neq \phi$ do

选取需要分配信道资源的用户 $k_0, k_0 = \arg\max\left(r_k / R_k^{\min}\right)$

根据式(12)从剩余信道中为用户选取最优子信道， $\delta(k_0, i^*) = 1$

$\text{CH} = \text{CH} - \{ i^* \}$

if 一个用户被分配多个信道

每个信道平均分配功率，更新功率分配矩阵

更新用户传输速率 $r_{k_0} = r_{k_0}^{i^*} + r_{k^*}$

end if

end while

//服务器计算资源分配

while $U \neq \phi$ do

for 用户 k^* 不满足 $t_{k^*}^{\text{off}} < t_{k^*}^{\text{local}}$ or $t_{k^*}^{\text{total}} < T_{k^*}^{\max}$

$f^{\text{srv}} = f^{\text{srv}} - f_{\text{unit}}^{\text{srv}}$

end for

$U = U - \{ k^* \}$

end while

while $f^{\text{srv}} > 0$ do

分配剩余计算资源： $k_0 = \arg\max(\sigma)$，

$f_{k_0}^{\text{srv}} = f_{k^*}^{\text{srv}} + f_{\text{unit}}^{\text{srv}}$

$f^{\text{srv}} = f^{\text{srv}} - f_{\text{unit}}^{\text{srv}}$

end while

2.3 卸载比更新

在为每个用户完成资源的初始分配之后，系统确定边缘服务器的可用计算资源，并更新用户的卸载比，使得移动用户的计算任务更多地卸载到边缘服务器上执行，减少计算时延。在本地任务计算的时间和部分卸载计算时间达到平衡时，时延最小。由式(9)可以得到用户更新后的卸载比

$$\lambda_k = \frac{\left(\dfrac{S_k}{f_k^0} - \max\left(\dfrac{S_k^{i,\text{off}}}{r_k^i} \right) \right)}{S_k \left(\dfrac{1}{f_k^{\text{srv}}} + \dfrac{1}{f_k^0} \right)} \tag{14}$$

由于信道和计算资源是动态变化，因此，λ_k 小于 0，表示资源条件无法满足任务要求，卸载比取值为 0；λ_k 大于 1，表示资源足够满足任务执行完全卸载，卸载比取值为 1。

用户卸载比更新算法见算法 2。

算法 2 用户卸载比更新算法

输入：用户信息、信道信息、服务器信息

输出：用户卸载比

while $U \neq \phi$

if 计算资源条件无法满足最低时延需求

$\lambda_k = 0$

else if 资源条件满足用户执行完全卸载

$\lambda_k = 1$

else

通过式(14)对每个用户进行卸载比更新

end if

$U = U - \{k\}$

end while

3　仿真结果与分析

使用 Python 对提出的方法进行仿真，并与本地计算、完全卸载、二进制卸载[25]、启发式任务卸载与资源分配（HTR, heuristic task offloading and resource allocation）[27]和性能保证部分卸载（PGPO, performance guaranteed partial offloading）[29]进行比较。其中，本地计算是指所有计算任务都在移动终端上执行。完全卸载是指所有计算任务卸载到边缘服务器执行。二进制卸载是指通过比较本地计算速率和卸载计算速率来确定是选择本地计算模式还是卸载模式。HTR 是指最初所有任务都在本地执行，然后将任务在服务器获得的收益作为优先级，将收益大的任务优先卸载到服务器执行，如果服务器没有资源可分配时，任务就在本地执行。PGPO 是指根据当前用户设备、通信资源和服务器计算资源的限定条件计算出卸载比进行计算卸载。

MEC 系统仿真场景考虑一个基站和多个移动用户，基站部署边缘服务器。基站覆盖区域的半径为 500 m，用户随机分布在该区域中。仿真主要参数见表 1。

表 1　　仿真主要参数

参数	数值
服务器的计算能力/GHz	10
网络总带宽/MHz	20
正交子信道数量	64
每个用户的任务量/KB	[200, 1 000]
任务计算时延限制/ms	[50, 100]
用户终端计算能力/GHz	[0.1, 1]
计算单位比特所需的 CPU 周期数	[500, 1 500]
每个 CPU 周期的能耗范围/J	$[0.5, 2]\times10^{-10}$ J
最大发射功率/dBm	23
信道噪声功率/dB	−100

用户数量对总时延的影响如图 1 所示。由于本地计算所有任务都在用户终端上运行，所有任务的总时延随用户数量线性增加。完全卸载将任务全部卸载到服务器进行计算，由于服务器的计算能力比用户终端强，在用户数量较少时，时延较低；当用户数量增加时，分配给每个用户的计算资源减少，总的时延比用户数量较少的时候增加得快，但还是低于本地计算。在用户数量较少时，二进制卸载和 HTR 都采用完全卸载的方式。随着用户数量增加，二进制卸载部分用户选择本地计算方式，而 HTR 则根据卸载的收益信息来决定是否在本地计算，以此缓解网络和服务器的压力，降低用户的总时延，因此比二进制卸载的时延要少。PGPO 采用的是部分卸载，随着用户数量的增加，每个用户会根据网络资源的情况动态调整卸载的数量，而不像二进制卸载那样简单地选择完全卸载或本地计算，因此总的时延增加得少。本文方法采用的是部分卸载方式，在用户数量较多时，网络通信资源的竞争增强，服务器的计算压力变大，合理利用服务器的资源分配来优化任务卸载的数量，能最大化利用服务器的计算资源，从而减少任务完成的时延。

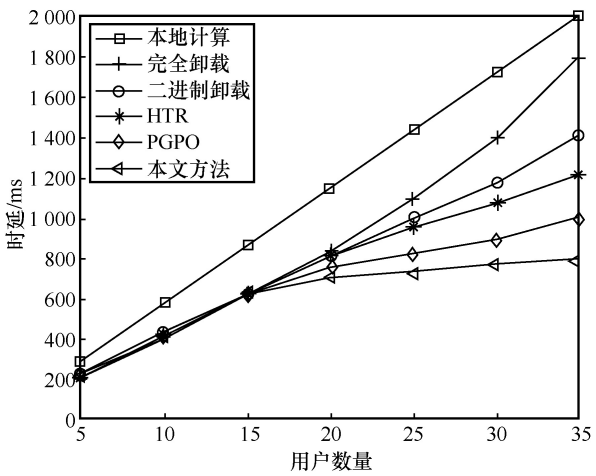

图 1　用户数量对总时延的影响

卸载比和用户数量的关系如图 2 所示。PGPO 方法的卸载比在 40%上下波动，由于不同用户所产生的卸载比不同，其平均值会产生一个波动。本文方法中，卸载比的最终确定受资源分配的影响，随着用户数量的不断增加，可分配的网络资源和计算资源会减少，导致卸载比降低，但明显高于 PGPO。

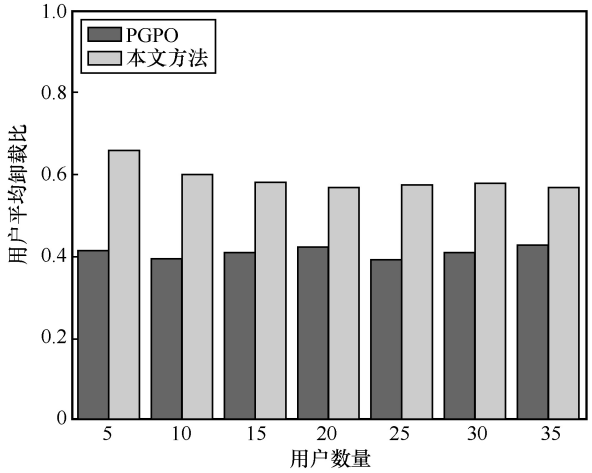

图 2　卸载比和用户数量的关系

在有 35 个用户时，服务器计算能力对用户平均时延的影响如图 3 所示。本地计算由于和服务器没有关系，所以平均时延保持不变。完全卸载方法的时延只受网络传输和服务器计算能力的影响，随着服务器计算能力的增加，用户的平均时延显著降低。在二进制卸载和 HTR 方式下，随着服务器计算能力的提高，系统的平均时延呈现缓慢下降的趋势。随着服务器计算能力越来越大，由于存在网络资源的限制，其时延逐渐向着完全卸载模式靠近。PGPO 和本文方法都是将部分任务传输到服务器，用户终端和服务器并行执行，由于本文方法考虑了服务器计算资源的分配，本文方法的平均时延较低。当服务器计算能力达到一定程度后，对用户平均时延的影响主要来自网络传输。

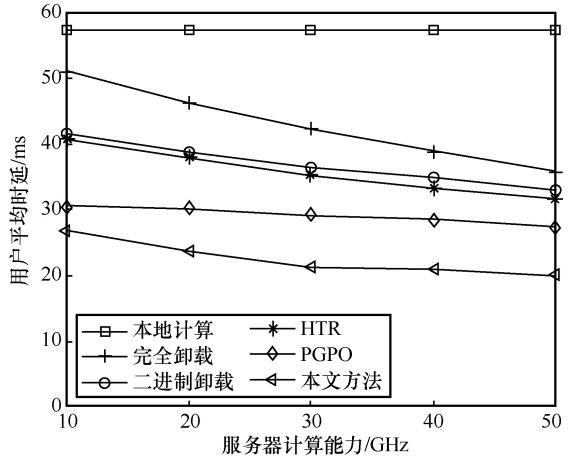

图 3　服务器计算能力对用户平均时延的影响

带宽对用户能耗的影响如图 4 所示。本地计算所有任务都在移动终端完成，与带宽无关，所以用户能耗最大并保持不变。完全卸载方法所产生的能耗是用户终端传输数据所产生的能耗，终端没有计算能耗。二进制卸载方法和 HTR 方法根据网络条件的差异来选择是否将计算任务卸载到边缘服务器，所以当带宽增加时，卸载数据越多，在移动终端进行计算的能耗就越少。PGPO 和本文方法都在传输信道分配和功率分配的基础上考虑卸载比，通信能耗相对于计算能耗要小。从图 2 可知，本文方法比 PGPO 卸载了更多的数据到服务器计算，因此本文方法在本地计算的数据比 PGPO 少，因此能耗比 PGPO 低。

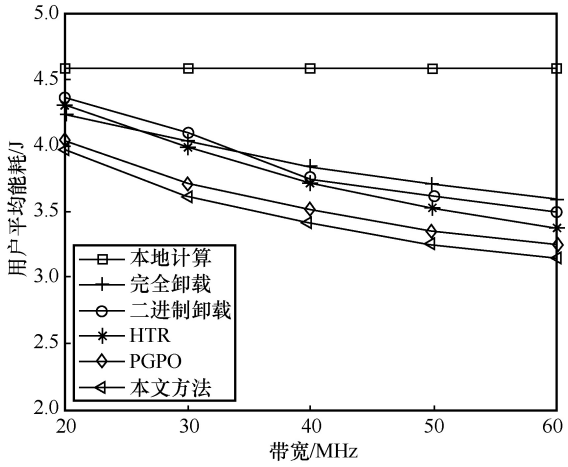

图 4　带宽对用户能耗的影响

4　结束语

本文在 OFDMA 的多用户 MEC 场景中，通过对用户设备计算资源、通信资源和边缘服务器计算资源的联合分配，提出了一种将用户计算密集型任务部分卸载到边缘服务器的方法。在满足用户时延和能耗的任务要求的情况下，每个用户尽可能地利用服务器的计算资源，卸载更多的任务到服务器端执行。实验结果表明，本文提出的部分卸载方法能够按照计算任务的需求有效地利用通信资源和服务器的计算资源来减少任务执行的总时延，同时降低用户设备的能量消耗。

参考文献：

[1] CISCO. CISCO annual internet report (2018–2023)[EB]. 2020.

[2] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

[3] 张厚浩, 李晗琳, 高林. 移动边缘计算中的分层资源部署与共享策略[J]. 物联网学报, 2021, 5(1): 11-18.

ZHANG H H, LI H L, GAO L. Hierarchical resource deployment and sharing strategy in mobile edge computing[J]. Chinese Journal on Internet of Things, 2021, 5(1): 11-18.

[4] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.

[5] NGUYEN Q H, DRESSLER F. A smartphone perspective on computation offloading—A survey[J]. Computer Communications, 2020(159): 133-154.

[6] LIN L, LIAO X F, JIN H, et al. Computation offloading toward edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1584-1607.

[7] JIANG C, CHENG X, GAO H, et al. Toward computation offloading in edge computing: a survey[J]. IEEE Access, 2019(7): 131543-131558.

[8] ZHANG T Y, CHIANG Y H, BORCEA C, et al. Learning-based offloading of tasks with diverse delay sensitivities for mobile edge computing[C]//Proceedings of 2019 IEEE Global Communications Conference. Piscataway: IEEE Press, 2019: 1-6.

[9] XIAO K L, GAO Z P, YAO C C, et al. Task offloading and resources allocation based on fairness in edge computing[C]//Proceedings of 2019 IEEE Wireless Communications and Networking Conference. Piscataway: IEEE Press, 2019: 1-6.

[10] 亓晋, 孙海蓉, 巩锟, 等. 移动边缘计算中基于信誉值的智能计算卸载模型研究[J]. 通信学报, 2020, 41(7): 141-151.
QI J, SUN H R, GONG K, et al. Research on intelligent computing offloading model based on reputation value in mobile edge computing[J]. Journal on Communications, 2020, 41(7): 141-151.

[11] 夏士超, 姚枝秀, 鲜永菊, 等. 移动边缘计算中分布式异构任务卸载算法[J]. 电子与信息学报, 2020, 42(12): 2891-2898.
XIA S C, YAO Z X, XIAN Y J, et al. A distributed heterogeneous task offloading methodology for mobile edge computing[J]. Journal of Electronics & Information Technology, 2020, 42(12): 2891-2898.

[12] MUKHERJEE M, KUMAR S, SHOJAFAR M, et al. Joint task offloading and resource allocation for delay-sensitive fog networks[C]//Proceedings of ICC 2019 - 2019 IEEE International Conference on Communications. Piscataway: IEEE Press, 2019: 1-7.

[13] YUE S, REN J, QIAON, et al. TODG: distributed task offloading with delay guarantees for edge computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(7): 1650-1665.

[14] 代美玲, 刘周斌, 郭少勇, 等. 基于终端能耗和系统时延最小化的边缘计算卸载及资源分配机制[J]. 电子与信息学报, 2019, 41(11): 2684-2690.
DAI M L, LIU Z B, GUO S Y, et al. A computation offloading and resource allocation mechanism based on minimizing devices energy consumption and system delay[J]. Journal of Electronics & Information Technology, 2019, 41(11): 2684-2690.

[15] 喻鹏, 张俊也, 李文璟, 等. 移动边缘网络中基于双深度 Q 学习的高能效资源分配方法[J]. 通信学报, 2020, 41(12): 148-161.
YU P, ZHANG J Y, LI W J, et al. Energy-efficient resource allocation method in mobile edge network based on double deep Q-learning[J]. Journal on Communications, 2020, 41(12): 148-161.

[16] JOŠILO S, DÁN G. Wireless and computing resource allocation for selfish computation offloading in edge computing[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 2467-2475.

[17] ESHRAGHI N, LIANG B. Joint offloading decision and resource allocation with uncertain task computing requirement[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1414-1422.

[18] ZHU X, CHEN S G, CHEN S L, et al. Energy and delay co-aware computation offloading with deep learning in fog computing networks[C]//Proceedings of 2019 IEEE 38th International Performance Computing and Communications Conference. Piscataway: IEEE Press, 2019: 1-6.

[19] GAO Y, ZHANG H R, YU F, et al. Joint computation offloading and resource allocation for mobile-edge computing assisted ultra-dense networks[J]. Journal of Communications and Information Networks, 2022, 7(1): 96-106.

[20] VU T T, HOANG D T, PHAN K T, et al. Energy-based proportional fairness for task offloading and resource allocation in edge computing[C]//Proceedings of ICC 2022 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2022: 1912-1917.

[21] TANG H J, WU H M, ZHAO Y B, et al. Joint computation offloading and resource allocation under task-overflowed situations in mobile-edge computing[J]. IEEE Transactions on Network and Service Management, 2022, 19(2): 1539-1553.

[22] WANG C, LIANG C, YU F R, 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.

[23] LI Q, ZHAO J, GONG Y. Cooperative computation offloading and resource allocation for mobile edge computing[C]//Proceedings of 2019 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2019: 1-6.

[24] BOZORGCHENANI A, TARCHI D, CORAZZA G E. Mobile edge computing partial offloading techniques for mobile urban scenarios[C]//Proceedings of 2018 IEEE Global Communications Conference. Piscataway: IEEE Press, 2018: 1-6.

[25] BI S Z, ZHANG Y J. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6): 4177-4190.

[26] MAHMOOD A, AHMED A, NAEEM M, et al. Partial offloading in energy harvested mobile edge computing: a direct search approach[J]. IEEE Access, 2020(8): 36757-36763.

[27] WANG Z L, DU H W, YE Q. HTR: a joint approach for task offloading and resource allocation in mobile edge computing[C]//Proceedings of ICC 2021 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2021: 1-6.

[28] 崔玉亚, 张德干, 张婷, 等. 一种面向移动边缘计算的多用户细粒度任务卸载调度方法[J]. 电子学报, 2021, 49(11): 2202-2207.
CUI Y Y, ZHANG D G, ZHANG T, et al. A multi-user fine-grained task offloading scheduling approach of mobile edge computing[J]. Acta Electronica Sinica, 2021, 49(11): 2202-2207.

[29] SALEEM U, LIU Y, JANGSHER S, et al. Performance guaranteed partial offloading for mobile edge computing[C]//Proceedings of 2018 IEEE Global Communications Conference. Piscataway: IEEE Press, 2018: 1-6.

[30] WANG L Y, ZHANG G L. Deep reinforcement learning based joint partial computation offloading and resource allocation in mobility-aware MEC system[J]. China Communications, 2022, 19(8): 85-99.

[作者简介]

刘耀（1976– ），男，博士，湖南工商大学副教授，主要研究方向为边缘计算、移动社会网络等。

何岳园（1994– ），男，湖南工商大学硕士生，主要研究方向为移动边缘计算等。

周红静（1976– ），女，湖南工商大学讲师，主要研究方向为移动边缘计算、机会网络、移动社会网络等。

李超良（1972– ），男，博士，湖南工商大学讲师，主要研究方向为物联网、隐私保护、区块链等。

李闯（1990– ），男，博士，湖南工商大学讲师，主要研究方向为物联网、高性能计算、并行计算等。