基于云服务器数据的多用户卸载

Jinho Choi

arXiv:2302.02498v1 [cs.IT] 5 Feb 2023

摘要

计算卸载对于计算能力有限的用户很有用，而移动边缘计算（MEC）可以帮助移动用户更高效地执行任务。本文研究用户在使用本地数据以及云存储服务器中的数据执行任务时的MEC，因为用户通常会将数据保存在云存储服务器中，或者任务需要使用公共数据集。本文在总传输时间约束下，构建了一个优化问题，以最小化用户和基站（BS）的总能耗，并通过将其转化为多个二元整数线性规划问题来求解。

索引词

卸载；服务器数据；二进制整数线性规划

I. 引言

随着智能手机等用户设备（UE）的存储和计算能力不断增强，越来越多的应用需要处理数据量更大、所需计算资源更多的任务。因此，可能有必要通过计算卸载来利用云服务器强大的计算能力 [1] [2]。在蜂窝系统中，可以采用移动边缘计算（MEC） [3] [4] [5] ，使集成在基站（BS）中的移动边缘计算服务器能够提供计算能力。由于一个小区内可能有多个用户设备或用户共存，因此可以考虑面向多个用户的联合计算卸载 [6] [7]。

在 [8] 中扩展了多用户移动边缘计算卸载，以满足异构网络中的不同用户需求；在由

作者现任职于澳大利亚吉朗 VIC 3220 迪肯大学信息技术学院（电子邮箱：jinho.choi@deakin.edu.au）。本研究由澳大利亚政府通过澳大利亚研究理事会的发现项目资助计划（DP200100391）提供支持。

小蜂窝基站和无线中继组成的异构网络中进行协作卸载。此外，与 [10], 类似，可以采用分布式方法，通过随机接入来执行卸载所需的数据上传，从而无需协调无线电资源分配。

大多数计算卸载方法都假设用户拥有执行任务所需的数据 [6] [8]。然而，除了用户的本地数据外，有些用户的任务可能还需要云服务器上的数据。例如，用户可能会使用云存储 [11] [12]。在这种情况下，如果用户不选择计算卸载，就需要接收存储在云存储中的数据；当所需的服务器数据量很大时，下载可能会产生高昂的成本（例如，用户执行人脸识别任务时，为查找匹配项而从服务器下载已知人脸数据库，可能会产生很高的通信成本）。因此，计算卸载更具吸引力，因为它不仅能让用户利用服务器的计算能力，还免去了下载数据集的需要；当服务器数据量较大时，下载这些数据集可能代价高昂。

本文研究用户需要服务器上的数据来执行任务时的移动边缘计算卸载。值得注意的是，如果无需使用服务器数据，该问题就归结为传统的移动边缘计算卸载。因此，本文的问题可视为一种推广。我们建立了一个优化问题，在上行链路和下行链路的总传输时间约束下，最小化计算和信号传输的总能耗[1]。随后，我们提出一种方法，将该问题转化为多个二元整数线性规划问题，从而无需进行穷举搜索即可找出最优卸载用户集合；穷举搜索的复杂度会随用户数量呈指数增长。

II. 系统模型

本节介绍无线多用户卸载的系统模型。假设有 K 个用户和一个基站。在全文中，我们假设计算和

[1]在传统的移动边缘计算卸载问题中，由于无需使用存储服务器中的数据，因此只考虑上行传输。

数据存储服务器通过高数据速率链路连接到基站。因此，假设服务器与基站可以互换。

A. 服务器端有数据的任务

假设用户 k 有一个计算任务，其大小为 L_k的本地数据存储在用户k处，大小为 B_k的服务器数据存储在服务器上。因此，聚合数据的大小为 $I_k = L_k + B_k$。稍微滥用一下符号，B_k、 L_k和 I_k 也分别称为用户 k处该任务的服务器数据、本地数据和聚合数据。完成该任务所需的中央处理器（CPU）周期数（使用 I_k ）记为 C_k。在大多数现有研究中，假设服务器没有数据。因此， $I_k = L_k$ ，每个用户都能在不与基站通信的情况下完成任务。然而，当 $B_k > 0$时，用户 k 必须与基站通信。

令 $a_k \in \{0, 1\}$ 表示计算卸载的选择。也就是说，如果 $a_k = 1$，用户 k选择计算卸载（该用户称为卸载用户）。否则（即 $a_k = 0$），用户 k 执行本地计算（该用户称为非卸载用户）。此外，令$\mathcal{A} = \{k\colon\ a_k = 1,\ \ k = 1、.....、\ K\}$和 一个 $= [a_1\ \cdots\ a_K]^{\mathrm{T}}$。对于用户 $k \in \mathcal{A}$，需要执行以下步骤:

1) 用户 k 通过上行传输将 L_k 发送至基站。

2) 基站使用其CPU（即服务器CPU），利用（L_k， B_k）执行任务，并获得任务的结果，该结果记为 Y_k （这也是结果的大小）。

3) 结果 Y_k通过下行传输发送给用户 k 。

另一方面，对于用户 $k \notin \mathcal{A}$，需要执行以下步骤:

1) 基站通过下行传输将服务器数据 B_k发送给用户 k).

2) 用户 k 使用其 CPU（即本地CPU）执行任务 (L_k， B_k)，并获得任务的结果。

请注意，卸载用户需要进行上行链路和下行链路传输，而非卸载用户则需要进行下行链路传输，如图 1 所示。

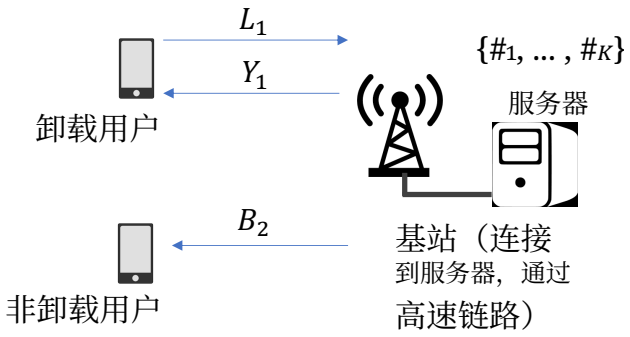

图1. 当用户需要使用云存储服务器上的数据集时计算卸载的示意图（此处，用户1是卸载用户，用户2是非卸载用户）。

B. 通信模型

本文假设上行链路和下行链路传输采用动态时分双工（TDD），并且时隙可以灵活划分。

对于上行传输，允许多个卸载用户同时传输[6]。令 h_k 表示用户 k 到基站的信道系数。若 $k \in \mathcal{A}$，则上行传输速率为

$$u_k(\mathbf{a}) = W \log_2 \left(1 + \frac{P_k \beta_k}{\sum_{i \in \mathcal{A} \setminus k} P_i \beta_i + N_0} \right), \tag{1}$$

其中，W 为系统带宽，P_k 为用户 k的发射功率，$\beta_k = |h_k|^2$，N_0 为基站处背景噪声的方差。

对于下行传输，基站采用时分多址（TDMA）向卸载用户发送数据。向用户 k 进行下行传输的数据速率为

$$v_k = W \log_2 \left(1 + \frac{\bar{P}_k \beta_k}{N_0} \right), \tag{2}$$

其中 $\bar{P}_k$ 是基站向用户 k的发射功率。在（2）中，信道功率增益 β_k假设与上行传输的信道功率增益相同，这是由于

时分双工。正如（2）所示，由于不存在干扰，每个用户只需执行单用户解码，而基站则需要对所有卸载用户执行多用户解码，正如（1）所示。

请注意，对于上行传输，也可以像 [7], 中那样使用 TDMA，并通过较高的发射功率实现较高的传输速率，P_k。另一方面，在考虑同时传输时，由于干扰，式（9）中的上行传输速率会受到限制，因此不必采用较高的发射功率，P_k。因此，对于发射功率有限的用户，同时传输可能更为理想。

由式（1）和（2），用户 k 的下行传输时间为

$$\bar{T}_k = \begin{cases} \frac{Y_k}{v_k}, & \text{if } k \in \mathcal{A} \\ \frac{B_k}{v_k}, & \text{if } k \notin \mathcal{A}. \end{cases} \tag{3}$$

对于卸载用户 $k \in \mathcal{A}$，其上行传输时间变为 $T_k = \frac{L_k}{u_k(\mathbf{a})}$。由于所有卸载用户同时传输，总上行传输时间为

$$T_{\text{up}} = \max_k T_k = \max_k \frac{L_k}{u_k(\mathbf{a})}. \tag{4}$$

如前所述，卸载用户的总传输时间包括上行传输时间和下行传输时间。另一方面，非卸载用户的总传输时间仅包括下行传输时间。

请注意，上传完成后，无需再进行传输。因此，部分卸载用户可能会提前结束传输，从而在上行传输的后续阶段减少干扰。因此，$u_k(\mathbf{a})$ 可能会更大，这意味着 T_{up} 式（4）可视为一个上界。

C. 能耗模型

设 g_0 为服务器每个 CPU周期的能耗，g_k 为用户 k 的能耗。因此，当用户 k 的任务在本地执行时，执行该任务的能耗为 $g_k C_k$。另一方面，卸载时，服务器的能耗为 $g_0 C_k$。此外，还

需要计入数据传输的能耗。用户 $k \in \mathcal{A}$ 的总能耗为

$$E_k = g_0 C_k + P_k \frac{L_k}{u_k(\mathbf{a})} + \bar{P}_k \frac{Y_k}{v_k}, \tag{5}$$

其中，第一项是在服务器执行计算所消耗的能量，第二项是上行传输所消耗的能量，最后一项是下行传输所消耗的能量。同理，可以求得用户 $k \notin \mathcal{A}$ 的总能耗。因此，我们有

$$E_k = \begin{cases} g_0 C_k + P_k \frac{L_k}{u_k(\mathbf{a})} + \bar{P}_k \frac{Y_k}{v_k}, & \text{if } k \in \mathcal{A} \\ g_k C_k + \bar{P}_k \frac{B_k}{v_k}, & \text{if } k \notin \mathcal{A}. \end{cases} \tag{6}$$

III. 信道反演功率控制下的总能耗最小化

在本小节中，我们考虑一个优化问题，旨在通过信道反演功率控制策略使总能耗最小化。

A. 信道反演功率控制

由于假设采用时分双工，得益于信道互易性，可以使用信号处理技术，根据基站发送的导频信号估计用户的上行信道。随后，上行传输采用如下信道反演功率控制策略:

$$P_k = \frac{P_{\text{BS}}}{\beta_k} \text{ and } \bar{P}_k = \frac{P_{\text{user}}}{\beta_k}, \tag{7}$$

其中， P_{BS} 和 P_{user} 分别表示基站和用户处的有效接收信号功率。由于最大发射功率有限，当 β_k 过小（由于深度¯衰落）时， P_k 或 P_k 可能超过最大发射功率，因此必须进行截断。不过，为方便起见，我们不考虑任何截断，并假设排除深度衰落用户（即，对于某些 $\epsilon > 0$， $\beta_k \leq \epsilon$ ）。

B. 在传输时间约束下最小化能量

除了信道反演功率控制策略外，我们还假设，为了支持所有 K 个用户，会分配一个长度为 τ的时隙。因此，所有 K 个用户的总传输

时间（包括上行链路和下行传输）受到限制，即 $\sum_{k=1}^{K} \bar{T}_k + T_{\text{up}} \leq \tau$。这里，我们假设服务器的计算时间足够短，因此在总传输时间约束中予以忽略。我们可以建立以下问题，在满足总传输时间约束的条件下，使总能耗最小：

$$\hat{\mathbf{a}} = \underset{\mathbf{a} \in \mathbb{B}^K}{\operatorname{argmin}} \sum_{k=1}^{K} E_k \quad \text{subject to (s.t.)} \sum_{k=1}^{K} \bar{T}_k + T_{\text{up}} \leq \tau, \tag{8}$$

其中 $\mathbb{B} = \{0, 1\}$。穷举搜索的复杂度为 2^K 量级。因此，对于较大的K，有必要找到一种能够以较低复杂度求解的方法。

注意，式（8）中的目标函数表示基站（服务器）以及 K 用户的总能耗。在 [7],中，目标函数仅表示用户消耗的总能量。另一方面，在 [6], 中，信号传输所消耗的能量未计入目标函数。由于我们考虑的是任务需要使用存储服务器上的数据集的情况，因此有必要将基站向非卸载用户发送数据集所产生的成本纳入考虑，目标函数应采用式（8）所示的总能耗。

C. 求解

本小节将说明，式（8）中的问题可以通过将其转化为多个二进制整数线性规划问题来求解。每个问题都可以采用标准方法求解，例如分支定界法 [13]。

为了求解（8），我们假设卸载用户的数量为 n（即 $|\mathcal{A}| = n$）。由于采用信道反演功率控制，根据（1），可以得到

$$u_k(\mathbf{a}) = u(n) = W \log_2 \left(1 + \frac{\gamma_{\text{BS}}}{(n-1)\gamma_{\text{BS}} + 1} \right), \forall k \tag{9}$$

其中，$\gamma_{\text{BS}} = \frac{P_{\text{BS}}}{N_0}$ 是基站的信噪比（SNR）（用于上行传输），$v_k = v = W\log_2(1+\gamma_{\text{user}})$，其中 $\gamma_{\text{user}} = \frac{P_{\text{user}}}{N_0}$是用户的信噪比（用于下行传输）。因此，对于 $n \in \{1, \ldots, K\}$，我们有

$$T(n) = T_{\text{up}} + \sum_{k=1}^{K} \bar{T}_k = T_0 + \frac{L_{\max}}{u(n)} + \sum_{k=1}^{K} \frac{Y_k - B_k}{v} a_k, \tag{10}$$

其中 $L_{\max} = \max_k L_k$ 且 $T_0 = \frac{\sum_{k=1}^{K} B_k}{v}$ 是在没有卸载用户时的总传输时间（即，$T(0) = T_0$）。类似地，根据（6），可以证明

$$E(n) = \sum_{k=1}^{K} E_k = E_0 + \sum_{k=1}^{K} e_k(n) a_k, \tag{11}$$

其中 $E_0 = \sum_{k=1}^{K} g_k C_k + \frac{P_{\text{user}} B_k}{\beta_k v}$ 和 $e_k(n) = (g_0 - g_k) C_k + \frac{P_{\text{BS}} L_k}{\beta_k u \, n()} + P_{\text{user}}(Y_k - B_k) \, \beta_k v$。然后，令 $d_k = \frac{Y_k - B_k}{v}$，对于给定的 n，该优化问题可转化为如下二元整数线性规划：

$$\hat{\mathbf{a}}(n) = \underset{\mathbf{a} \in \mathbb{B}^K}{\operatorname{argmin}} \, E_0 + \mathbf{e}^{\mathrm{T}} \mathbf{a} \quad \text{s.t.} \begin{cases} \mathbf{d}^{\mathrm{T}} \mathbf{a} \leq \tau - T_0(n) \\ \mathbf{1}^{\mathrm{T}} \mathbf{a} = n, \end{cases} \tag{12}$$

其中 $\mathbf{e} = [e_1(n) \ \cdots \ e_K(n)]^{\mathrm{T}}$，$\mathbf{d} = [d_1 \ \cdots \ d_K]^{\mathrm{T}}$，且 $\mathbf{1} = [1 \ \cdots \ 1]^{\mathrm{T}}$。第二个约束源于以下假设：$|\mathcal{A}| = n$ 或 $||\mathbf{a}||_1 = n$。

ˆ对于给定的 12，我们可以求解式（$n \in \{1, \ldots, K-1\}$。令 $E(n)$ 表示最小ˆ能量；若存在可行解，则该值有限。否则，$E(n) = \infty$。注意，若$n = 0$ 或 K，则无需求解式（12），因为 ˆ$\mathbf{a}(n)$ 分别为 $\mathbf{0}$ 或 $\mathbf{1}$。于是，式（8）的解为

$$\hat{\mathbf{a}} = \hat{\mathbf{a}}(n^*), \tag{13}$$

ˆ ˆ ˆ 其中 $E(n^*) \leq E(n)$，$\forall n \neq n^*$。请注意，若 $E(n^*) = \infty$，则不存在可行解。若无可行解，可以增加时隙长度 τ。

D. 平均传输时间

假设 L_k、B_k 和 C_k 均独立同分布。此外，用户 k 的任务输出大小 Y_k 由 $Y_k = f(L_k + B_k)$ 给出，其中 $f(\cdot)$ 是一个特定函数，即任务输出的大小由总输入大小 $L_k + B_k$ 决定。为简单起见，我们将 $f(x)$ 线性化如下：$f(x) \approx c_0 + c_1 x$，其中 $c_0 > 0$ 是独立于输入的输出长度，$c_1 > 0$ 是一个假设较小的常数。例如，假设任务是分类（其中 L_k 是测试图像，B_k 是用于比较的一组图像）。那么，输出就是一个标签，只需

少量比特即可表示。然后，根据（10），可以得出平均总传输时间为

$$\begin{aligned}\mathbb{E}[T(n)] &= \frac{K\bar{B}}{v} + \frac{\mathbb{E}[L_{\max}]}{u(n)} + n\frac{\mathbb{E}[f(L_k+B_k)]-\bar{B}}{v} \\ &\approx \frac{K\bar{B}}{v} + \frac{\mathbb{E}[L_{\max}]}{u(n)} + n\frac{c_0+c_1\bar{L}-(1-c_1)\bar{B}}{v} \\ &\approx \frac{K\bar{B}}{v} + n\theta, \end{aligned} \tag{14}$$

其中 $\theta = \frac{\mathbb{E}[L_{\max}]\ln 2}{W} + \frac{c_0+c_1\bar{L}-(1-c_1)\bar{B}}{v}$。最后一个近似是由于 $u(n) \approx W\log_2\left(1+\frac{1}{n}\right) \approx \frac{W}{n}\ln 2$。因此，若 $\theta > 0$，我们可以看到 $\mathbb{E}[T(n)]$ 随 n 增加，这意味着在给定 τ 的情况下，卸载用户数量往往较少，以满足约束。对于给定的 τ，我们还可以求出卸载用户数量的最大值，记为 $\bar{n}$；这有助于降低在 III-C小节中求解 ˆa 的复杂度，因为式 (12) 需要对 $n \in \{1,\dots,\bar{n}\}$ 求解。另一方面，当 θ 接近 0 或小于 0 时，总传输时间与卸载用户数量 n 变得或多或少相互独立。在这种情况下，给定 $\tau > \frac{K\bar{B}}{v}$ 后，可以通过决定卸载用户来最小化总能耗，同时减少总传输时间约束的影响。

IV. 仿真结果

在仿真中，我们假设 L_k、B_k 和 C_k 相互独立，而 $L_k \sim$ 服从 Exp(L)，$B_k \sim$ 服从 Exp(B)，且 $C_k \sim$ 服从 Exp(C)。此外，设 $f(x) = c_1 x$，其中 $c_1 = 0.1$。假设 L_k、B_k 和 C_k 的单位均以 W 归一化。因此，单位变为比特每赫兹。我们假设 g_0 已归一化（即 $g_0 = 1$），而 $g_k \sim$ 服从 Unif$[0,\bar{g}]$，其中 $\bar{g} = 10$。对于信道增益 β_k，我们假设其服从修正的瑞利衰落分布，使得 β_k 服从平移指数分布：$\beta_k \sim \zeta e^{-\zeta(\beta_k-\epsilon)}$，其中 $\beta_k \geq \epsilon = 0.05$ 和 $\zeta = \frac{1}{1-\epsilon}$。

在图2中，当 $K = 10$、$\gamma_{\text{BS}} = 3$、$\gamma_{\text{user}} = 6$、$\bar{C} = 1$、$\bar{L} = 2$ 和 $\tau = T_0\big|_{\bar{B}=10} = \frac{10K}{\log_2(1+\gamma_{\text{user}})} = 35.63$ 时，展示了服务器数据的平均大小 B 变化时的性能。总能耗（见左上子图）和平均卸载用户数（见

右上子图）会随 B 增加。这是因为下载服务器数据的成本会随 B 增加（这可能会导致更多用户进行卸载）。这一现象也能从卸载用户数量的经验分布中看出（见下方子图）。还可以看到，当 $B \leq L = 2$ 时，存在 2 个卸载用户的概率最高。可是，随着 B 增加，卸载用户数量超过 3 个的概率也会增加。

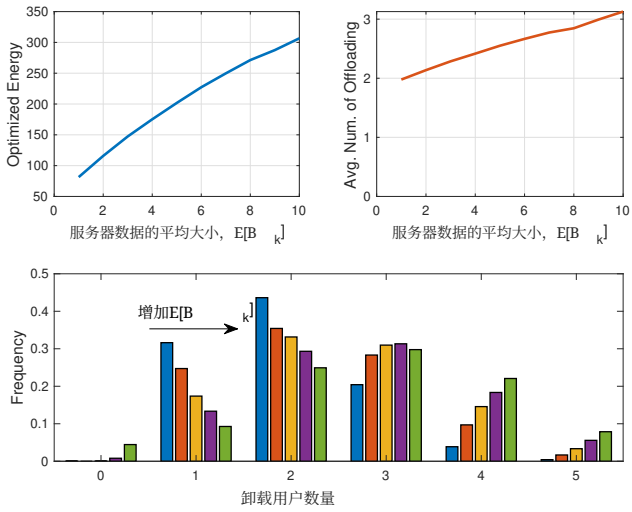

图2。不同服务器数据大小 $\bar{B}$下的性能，其中 $K = 10$、 $\gamma_{\mathrm{BS}} = 3$、 $\gamma_{\mathrm{user}} = 6$、 $\bar{C} = 1$、 $\bar{L} = 2$和 $\tau = 10K \log_2(1+\gamma_{\text{用户}}) = 35.63$。

图3 展示了不同总传输时间阈值下的性能， τ，其中 $K = 10$, $\gamma_{\mathrm{BS}} = 3$, $\gamma_{\mathrm{user}} = 6$, $C = 1$, $L = 2$, 以及 $B = 4$。随着 τ 增加（即约束放宽），总能耗增加（如左上子图所示），且卸载用户数量增多（如右上和下方子图所示）。

五、结论

我们研究了用户需要使用云存储中的数据时的计算卸载问题。我们建立了一个在总传输时间约束下最小化能耗的问题，并通过二元整数线性规划求得了该问题的解。

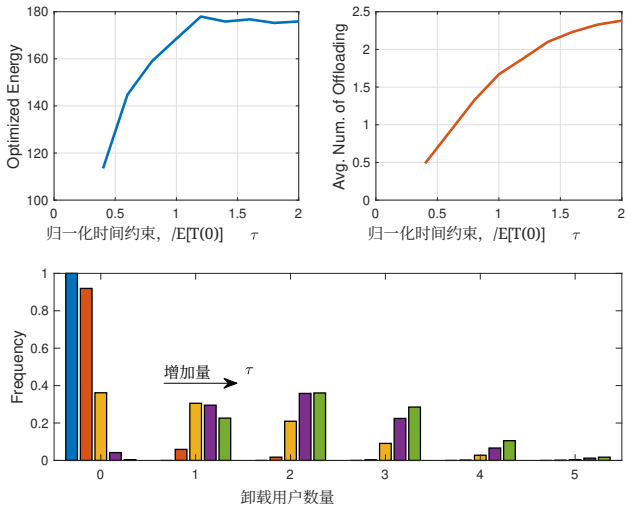

图3. 不同总传输时间阈值 τ 下的性能，其中 $K = 10$、 $\gamma_{\mathrm{BS}} = 3$、 $\gamma_{\mathrm{user}} = 6$、 $\bar{C} = 1$、 $\bar{L} = 2$ 和 $\bar{B} = 4$。

在后续研究中，我们将扩展这一方法，以考虑从云存储服务器检索数据的成本。

参考文献

[1] K. Yang、S. Ou 和 H.-H. Chen，“面向运行较重型移动互联网应用的资源受限移动设备的有效卸载服务”，*IEEE Communications Magazine*，第46卷，第1期，第56–63页，2008年。[2] K. Kumar 和 Y.-H. Lu，“面向移动用户的云计算：卸载计算能否节省能量？”，*Computer*，第43卷，第4期，第51–56页，2010年。[3] S. Barbarossa、S. Sardellitti 和 P. Di Lorenzo，“边计算边通信：基于5G异构网络的分布式移动云计算”，*IEEE Signal Processing Magazine*，第31卷，第6期，第45–55页，2014年。[4] P. Mach 和 Z. Becvar，“移动边缘计算：架构与计算卸载综述”，*IEEECommunications Surveys Tutorials*，第19卷，第3期，第1628–1656页，2017年。[5] N. Abbas、Y. Zhang、A. Taherkordi 和 T. Skeie，“移动边缘计算：综述”，*IEEE Internet of Things Journal*，第5卷，第1期，第450–465页，2018年。[6] X. Chen、L. Jiao、W. Li 和 X. Fu，“移动边缘云计算的高效多用户计算卸载”，*IEEE/ACM Transactions on Networking*，第24卷，第5期，第2795–2808页，2016年。[7] C. You、K. Huang、H. Chae 和 B.-H. Kim，“面向移动边缘计算卸载的能效资源分配”，*IEEE Transactions onWireless Communications*，第16卷，第3期，第1397–1411页，2017年。[8] P. Zhou、K. Shen、N. Kumar、Y. Zhang、M. M. Hassan 和 K. Hwang，“5G异构网络中面向移动边缘计算的通信高效卸载”，*IEEE Internet of Things Journal*，第8卷，第13期，第10237–10247页，2021年。[9] J. H. Anajemba、T. Yue、C. Iwendi、M. Alenezi 和 M. Mittal，“面向能效多接入边缘计算的最优协作卸载方案”，*IEEE Access*，第8卷，第53931–53941页，2020年。[10] J. Choi，“面向物联网应用设备的基于随机接入的多用户计算卸载”，*IEEE Internet of ThingsJournal*，第9卷，第21期，第22034–22043页，2022年。

[11] R. Buyya、C. S. Yeo、S. Venugopal、J. Broberg 和 I. Brandic，“云计算与新兴IT平台：作为第五种公共事业提供计算服务的愿景、炒作与现实”，*Future Generation Computer Systems*，第25卷，第6期，第599–616页，2009年。[12] J. Wu、L. Ping、X. Ge、Y. Wang 和 J. Fu，“云存储作为云计算的基础设施”，载于*2010*年智能计算与认知信息学国际会议，第380–383页，2010年。[13] C. Papadimitriou 和 K. Steiglitz，组合优化：算法与复杂度。Dover计算机科学丛书，Dover出版社，1998年。