

◆ 算法研究与应用 ◆

基于 SWIPT 的 D2D 通信辅助移动边缘计算任务卸载策略^{*}杜利俊¹, 李陶深^{1,2**}, 黄翊芯¹, 漆治君¹

(1. 广西大学计算机与电子信息学院, 广西南宁 530004; 2. 南宁学院, 中国-东盟综合交通国际联合实验室, 广西南宁 530200)

摘要:为解决 5G 移动通信系统中移动用户计算能力不足、能量消耗多、无线资源缺乏等问题, 本文构建一种基于无线携能通信(Simultaneous Wireless Information and Power Transfer, SWIPT)的多用户设备间(Device to Device, D2D)通信辅助移动边缘计算(Mobile Edge Computation, MEC)系统模型, 提出一种 D2D-MEC 联合卸载策略。该策略以系统中请求用户总能耗最小化为目标, 采用二进制卸载模式和功率分流模式对请求用户进行任务卸载和能量收集。针对能耗最小化问题为非线性混合整数规划问题, 根据整数变量和实数变量将原问题解耦为功率分配和计算任务卸载两个独立子问题, 并分别采用 Dinkelbach 方法和匈牙利算法求出两个子问题的最优解。仿真实验结果表明, 本文所提策略优于传统的 D2D 卸载策略和 MEC 卸载策略, 有效降低了请求用户的总能耗, 提高了任务执行效率。

关键词:无线携能通信; 移动边缘计算; 设备间通信; 任务卸载; 功率分配

中图分类号: TP393 文献标识码: A 文章编号: 1005-9164(2023)04-0754-10

DOI: 10.13656/j.cnki.gxkx.20230928.015

在传统的蜂窝网络中, 随着计算密集型移动应用的激增, 用户的服务请求随之增加。为了有效满足移动互联网、物联网高速发展所需的高回传带宽、低能耗的要求, 业界提出了移动边缘计算(Mobile Edge Computation, MEC)技术^[1]。MEC 技术的特点是将远端云的网络资源下沉至网络边缘, 使用户可以近距离地使用边缘节点的网络资源。在一些应用场景(如增强/虚拟现实、动态内容交付、物联网、车联网等)

中, 人们在蜂窝网络边缘部署 MEC 服务器, 将移动设备中的计算任务卸载到 MEC 服务器上进行存储、计算, 从而解决移动设备在资源存储、计算性能以及能效等方面的不足。任务卸载作为 MEC 中关键技术之一, 可以将资源受限的移动设备部分或者全部任务交给云计算环境处理, 从而降低能耗, 提高任务执行效率。

目前, MEC 任务卸载重点研究 3 种卸载决策: 本

收稿日期: 2022-03-08

修回日期: 2022-04-21

* 国家自然科学基金项目(61762010)和广西科技计划项目(桂科 AD20297125)资助。

【第一作者简介】

杜利俊(1997-), 女, 在读硕士研究生, 主要从事无线携能通信、移动边缘计算研究。

【**通信作者】

李陶深(1957-), 男, 教授, 博士研究生导师, 主要从事无线网络、云计算与大数据、网络计算与安全等研究, E-mail: tshli@gxu.edu.cn。

【引用本文】

杜利俊, 李陶深, 黄翊芯, 等. 基于 SWIPT 的 D2D 通信辅助移动边缘计算任务卸载策略[J]. 广西科学, 2023, 30(4): 754-763.

DU L J, LI T S, HUANG Y X, et al. Task Offloading Strategy of D2D Communication Assisted MEC Based on SWIPT [J]. Guangxi Sciences, 2023, 30(4): 754-763.

地执行、完全卸载和部分卸载。其中,本地执行是移动设备在本地执行任务,需要消耗设备本身资源和能量;完全卸载即二进制卸载则是将任务全部卸载到资源充足的服务器或移动设备上;部分卸载是按照一定的策略将一部分任务卸载到服务器执行,一部分则在本地执行。已有研究提出以降低能量消耗为目标的卸载策略^[2-5]。You 等^[2,3]考虑了由多个用户组成的 MEC 卸载系统的资源分配问题,并进一步研究了基于时分多址(Time Division Multiple Access, TDMA)和正交频分多址(Orthogonal Frequency Division Multiple Access, OFDMA)的多用户 MEC 卸载系统的资源分配问题,采用优先级与阈值相比较的方法决定卸载策略。Ali 等^[4]基于深度学习的智能决策算法,提出了一种新颖的、基于高效节能深度学习的卸载方案。薛建彬等^[5]针对一对多的广播系统任务分配问题,设计了一种基于拉格朗日乘子法的任务分配优化算法,对用户本身和不同参数的接入点进行合理的任务分配,以联合优化任务分配和执行时延,实现系统开销最小化。

MEC 任务卸载使得任务执行性能增强,但是当多个移动用户设备同一时刻卸载任务到服务器时,MEC 服务器上可能会发生资源争用的问题;同时,当移动设备远离基站时,设备自身资源不足,任务也不能卸载到基站。设备间(Device to Device, D2D)通信技术被认为是应对上述问题有效的技术。D2D 通信技术是指通信网络中邻近设备之间直接交换信息的技术,可以降低通信系统中核心网络的数据压力,提高频谱效率和吞吐量。近年来,D2D 通信技术和 MEC 技术的结合应用在提高蜂窝网络频谱效率和能源效率方面得到了广泛的研究。Chai 等^[6]研究了一个蜂窝-D2D-MEC 系统任务执行代价最小化问题,提出一个联合任务管理体系来实现高效的信息交互和任务管理。考虑利用邻近策略来支持高速移动计算和高速数据通信,He 等^[7]结合 MEC 和 D2D 通信技术来提高蜂窝网络的计算能力,最大限度地提高支持设备任务执行的数量。Kai 等^[8]研究了 D2D 通信辅助 MEC 网络中计算联合资源管理问题,通过 D2D 通信显著增加了 MEC 网络中执行的任务数,且大大降低了每个执行任务的能耗。上述研究专注于移动设备的能效,并未考虑利用无线携能通信(Simultaneous Wireless Information and Power Transfer, SWIPT)技术同时给移动设备传输信息和提供能量的特点,从而在满足移动设备所需能源供应的同时提

高实时计算能力。

由于移动设备电池容量有限,在没有电源的情况下移动设备无法长时间工作。传统的移动设备由普通的电池供电,在使用中需要频繁更换电池或充电,使得这些设备在无线通信过程中常常发生任务处理中断,一方面给设备通信带来不必要的麻烦,另一方面会产生昂贵的电池更换成本,给环境带来巨大的污染。基于射频(Radio Frequency, RF)信号的 SWIPT 技术可以对 RF 信号进行能量收集(Energy Harvesting, EH),具有更高的可控性和稳定性^[9]。基于 SWIPT 技术的 EH 设备有两种接收机模式,即时间切换(Time Switching, TS)和功率分流(Power Splitting, PS)^[10]。TS 模式下的 EH 和信息解码(Information Decoding, ID)处于不同时隙;PS 模式则将接收到的信号分成不同功率信号,同时进行 EH 和 ID。这两种接收模式使 SWIPT 技术能够应用于各种 MEC 网络,克服网络中移动用户的电池性能瓶颈。目前已有将 SWIPT 技术应用到 MEC 系统中的研究,Wen 等^[11]提出了一个基于多输入多输出(Multiple Input Multiple Output, MIMO)全双工中继的 SWIPT-MEC 系统,在保证移动用户能量充足的情况下降低系统总能耗。与之不同的是,Chen 等^[12]提出的框架中没有中继节点,将资源分配作为一个联合的非线性优化问题,以期达到最小化系统总能耗的目标。Fu 等^[13,14]针对车联网和卫星物联网两个场景,模拟单个部署有 MEC 服务器的基站与多个用户之间的无线通信过程,其中车联网场景的研究目标是 minimized 设备的能耗,而卫星物联网场景则是最大化终端的上行传输速率。上述研究主要考虑将任务卸载到 MEC 服务器,然而,随着应用场景的不断丰富,在最大可容忍时延约束下 MEC 服务器的计算资源将无法

满足大量的计算需求。

目前将 MEC 技术和 SWIPT 技术结合,可在一定范围内实现通信网络系统的增益,解决通信系统中移动用户计算能力不足以及无线资源缺乏问题,提高系统的计算能力和续航能力。考虑到大量计算密集型任务卸载到 MEC 服务器可能会发生资源争用,可以借助 D2D 通信将任务卸载给邻近设备,从而缓解 MEC 服务器的计算压力。本文构建一个基于 SWIPT 的多用户 D2D 通信辅助的 MEC 系统模型,采用 SWIPT 技术将基站的 RF 信号传输给移动用户,移动用户均配置 PS 接收机对接收的 RF 信号进行信息解码和能量收集;同时,提出一种 D2D-MEC

联合任务卸载策略,该策略采用二进制卸载模式,移动用户可以选择本地执行、MEC 卸载和 D2D 卸载 3 种策略中的任意 1 种,拟通过联合优化功率分配和任务卸载问题,实现请求用户总能耗最小化的目标。

1 系统模型

本文考虑如图 1 所示的一个基于 SWIPT 的多用户 D2D 通信辅助 MEC 系统。该系统由单个部署了 MEC 服务器的基站(Base Station, BS)、 N 个需要执行计算任务的请求用户(Request Users, RUs)以及 M 个空闲的服务用户(Service Users, SUs)组成。移动用户都是单天线,基站是多天线。每个请求用户 RU 都有一个计算密集型任务,该任务可以表示为 $\zeta_i \triangleq (I_i, D_i, T_i^{\max})$,其中 I_i, D_i, T_i^{\max} 分别表示 RU i 输入数据的大小、完成任务所需的计算资源量以及最大可容忍时延。

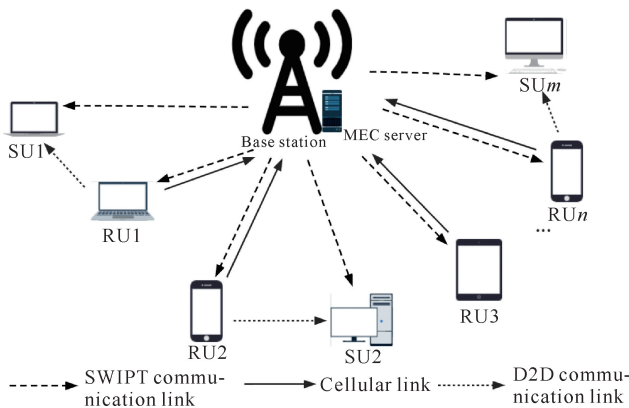


图 1 基于 SWIPT 的多用户 D2D 通信辅助的 MEC 系统

Fig. 1 Multi-user D2D communication assisted MEC system based on SWIPT

系统中每个请求用户 $RU_i (i = 1, 2, \dots, N)$ 既可以与基站建立蜂窝链路,也可以与一个且仅一个邻近的服务用户 $SU_j (j = 1, 2, \dots, M)$ 建立 D2D 通信链路。RUs 既可以通过蜂窝链路将计算任务卸载到 MEC 服务器,也可以通过 D2D 通信链路将计算任务卸载到某些性能相对较高且空闲的 SUs 上, SUs 为邻近的请求用户提供 D2D 卸载服务。总的来说, RU_i 可以在 3 种模式下执行任务,即本地执行、MEC 卸载、D2D 卸载,不能接收其他用户的任务进行求解。为了保证执行任务的安全性,假设 RU_i 的任务不可拆分,每个任务只能选择一种计算模式执行。由于 MEC 服务器的计算资源丰富,任务优先选择 MEC 卸载模式,只有当 MEC 服务器在最大可容忍时延内执行的任务达到上限时,再考虑选择 D2D 卸载模式;当所有服务用户都被占用时,最后考虑在本

地执行。

BS 配有多输入多输出天线、强大的计算芯片和充电设备^[15]。在该系统中,为了最大限度地扩大蜂窝网络待执行计算密集型任务的设备范围,它需要在最大可容忍时延下处理 RUs 上传的所有计算任务,然后通过子信道返回计算结果和能量。

1.1 能量收集模型

为了保证系统中每个移动用户(Mobile Users, MUs)不会因能量不足而造成任务执行中断,假设每个 MU 都配备了一个 PS 接收机,PS 接收机将基站传输的射频信号按照一定的比率分成两个不同的功率流,接收信号的 ρ 部分用于能量收集, $(1 - \rho)$ 部分用于信息解码^[16],功率分流架构如图 2 所示。为了保证终端的功率不被消耗,本文采用下行最大功率向终端传输能量。

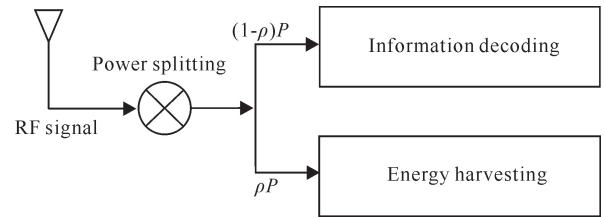


图 2 SWIPT 中的功率分流架构

Fig. 2 Power splitting architecture in SWIPT

$MU_k (k = 1, 2, \dots, N + M)$ 接收到的最大能量^[6]可以表示为

$$E_k^{\text{eh}} = \rho \beta_k (P_k^{\text{eh}} h_k^{\text{eh}} T^{\text{tolerant}} + \sigma^2), \quad (1)$$

式(1)中, β_k 表示 MU_k 的能量转换率, P_k^{eh} 表示基站到 MU_k 之间的下行最大传输功率, h_k^{eh} 则表示基站与 MU_k 之间用于能量传输的信道增益, T^{tolerant} 表示下行最大可容忍时延, σ^2 表示高斯白噪声功率。由于终端需要不断地接收能量,因此假设信道增益在一段时间内保持不变^[16],且这些信道是固定的。

1.2 任务执行模型

为了降低请求用户总能耗, RU_i 可以采取 3 种任务执行模式中的任意 1 种,任务执行模型如图 3 所示。

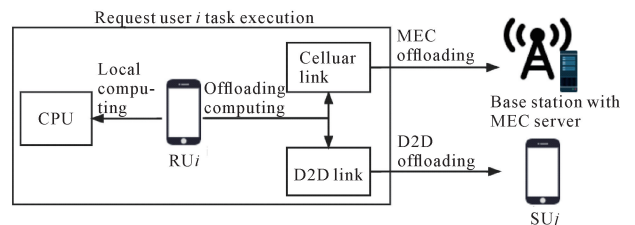


图 3 任务执行模型

Fig. 3 Task execution model

1.2.1 本地执行

T_i^{local} 表示 RU i 本地执行任务的时延^[17], 用公式表达为

$$T_i^{\text{local}} = \frac{D_i}{F_i}, \quad (2)$$

式(2)中, D_i 表示完成任务所需的计算资源量, F_i 表示 RU i 的计算能力 ($1 \leq i \leq N$)。本文中, RU i 、SU j 和 MEC 服务器的计算能力被定义为设备或服务器的 CPU 运行频率。

RU i 本地执行任务消耗的能量为^[13]

$$E_i^{\text{local}} = \kappa D_i F_i^2, \quad (3)$$

式(3)中, κ 表示与设备的 CPU 性能相关的有效开关电容。

1.2.2 MEC 卸载执行

在 MEC 卸载模式下执行任务, 完成任务的总时间包括任务上传时延、MEC 服务器中任务执行时延以及基站将计算结果回传给 RU i 的时延。

T_i^{mec} 表示 MEC 卸载模式下 RU i 的任务执行时延, 其计算公式为

$$T_i^{\text{mec}} = T_i^{\text{mec},t} + T_i^{\text{mec},e}, \quad (4)$$

式(4)中, $T_i^{\text{mec},t}$ 表示 RU i 将任务卸载到 MEC 服务器的传输时延; $T_i^{\text{mec},e}$ 表示任务在 MEC 服务器中的执行时延。由于 MEC 服务器上处理的业务输出数据量通常远小于输入数据量, 所以忽略结果回传时间^[11]。

$T_i^{\text{mec},t}$ 的计算公式如下:

$$T_i^{\text{mec},t} = \frac{I_i}{R_i^{\text{mec}}}, \quad (5)$$

式(5)中, I_i 表示 RU i 输入数据的大小; R_i^{mec} 表示 RU i 向 MEC 服务器传输任务时的可完成数据速率, 可以表示为

$$R_i^{\text{mec}} = W^b \log_2 \left(1 + \frac{p_i^{\text{mec}} h_i}{\sigma^2} \right), \quad (6)$$

式(6)中, W^b 表示 BS 的信道带宽, p_i^{mec} 表示 RU i 到 MEC 服务器的上行传输功率, h_i 表示 RU i 到基站的信道增益。

式(4)中的 $T_i^{\text{mec},e}$ 可以表示为

$$T_i^{\text{mec},e} = \frac{D_i}{\theta_i F^{\text{mec}}}, \quad (7)$$

式(7)中, $\theta_i \in [0, 1]$, 表示任务卸载到 MEC 服务器时分配给 RU i 的计算能力系数, F^{mec} 为 MEC 服务器的计算能力。

RU i 在 MEC 卸载模式下的传输能耗可以用上

行传输功率与输入数据的大小的乘积与数据传输速率之比表示:

$$E_i^{\text{mec}} = \frac{p_i^{\text{mec}} I_i}{R_i^{\text{mec}}}. \quad (8)$$

本文假设 RU i 自身由于信号处理而产生的能量消耗可以忽略, 这里只研究 RU i 将其任务传输到 MEC 服务器而产生的能量消耗。

1.2.3 D2D 卸载执行

当 RU i 通过 D2D 通信链路将任务传输给 SU j , D2D 卸载执行任务的总时延 $T_{i,j}^{\text{d2d}}$ 为

$$T_{i,j}^{\text{d2d}} = T_{i,j}^{\text{d2d},t} + T_{i,j}^{\text{d2d},e}, \quad (9)$$

式(9)中, $T_{i,j}^{\text{d2d},t}$ 表示 RU i 将任务卸载到 SU j 所需的传输时延, $T_{i,j}^{\text{d2d},e}$ 表示任务在 SU j 处的执行时延。这里 SU j 的计算能力高于 RU i 的计算能力, 有效地降低了任务完成总时延。 $T_{i,j}^{\text{d2d},t}$ 、 $T_{i,j}^{\text{d2d},e}$ 的计算公式分别为

$$T_{i,j}^{\text{d2d},t} = \frac{I_i}{R_{i,j}^{\text{d2d}}}, \quad (10)$$

$$T_{i,j}^{\text{d2d},e} = \frac{D_i}{F_j^{\text{d2d}}}, \quad (11)$$

式(11)中, F_j^{d2d} 为 SU j 的计算能力; 式(10)中, $R_{i,j}^{\text{d2d}}$ 为 RU i 向 SU j 传输任务时的数据传输速率, 可以表示为

$$R_{i,j}^{\text{d2d}} = W^d \log_2 \left(1 + \frac{p_i^{\text{d2d}} h_{i,j}}{\sigma^2} \right), \quad (12)$$

式(12)中, W^d 和 $h_{i,j}$ 分别表示 RU i 与 SU j 之间的信道带宽和信道增益, p_i^{d2d} 表示 RU i 传输任务到 SU j 时的传输功率。

RU i 上传任务到 SU j 的传输能耗为

$$E_{i,j}^{\text{d2d},t} = p_i^{\text{d2d}} T_{i,j}^{\text{d2d},t}, \quad (13)$$

SU j 执行 RU i 上传任务所消耗的能量为

$$E_{i,j}^{\text{d2d},e} = \kappa D_i (F_j^{\text{d2d}})^2. \quad (14)$$

2 问题描述与优化求解

在上述系统模型的基础上, 本文提出一种具有一系列必要约束条件的联合优化功率分配和任务卸载的请求用户总能耗最小化问题。

2.1 优化约束

为了解决联合优化功率分配和任务卸载问题, 使得请求用户总能耗最小, 需要考虑一系列优化约束条件。

本文中, 每个 RU i 只能选择本地执行、MEC 卸载、D2D 卸载 3 种任务卸载模式中的 1 种, 用 $\lambda_i^{\text{local}} \in$

$\{0,1\}$ 表示 RU_i 的本地执行变量, 即当 RU_i 在本地执行任务时, $\lambda_i^{\text{local}} = 1$, 否则 $\lambda_i^{\text{local}} = 0$; $\lambda_i^{\text{mec}} \in \{0,1\}$ 表示 RU_i 的 MEC 卸载变量, 即当 RU_i 卸载其任务到 MEC 服务器时, $\lambda_i^{\text{mec}} = 1$, 否则 $\lambda_i^{\text{mec}} = 0$; $\lambda_{i,j}^{\text{d2d}} \in \{0,1\}$ 表示 RU_i 的 D2D 卸载变量, 即当 RU_i 卸载其任务到 SU_j 时, $\lambda_{i,j}^{\text{d2d}} = 1$, 否则 $\lambda_{i,j}^{\text{d2d}} = 0$ 。本文将任务卸载约束表示为

$$C1: \lambda_i^{\text{local}} + \lambda_i^{\text{mec}} + \sum_{j=1}^M \lambda_{i,j}^{\text{d2d}} = 1, \lambda_i^{\text{local}}, \lambda_i^{\text{mec}}, \lambda_{i,j}^{\text{d2d}} \in \{0,1\}。 \quad (15)$$

受传输和计算资源的限制, 本文假设一个请求用户只能向一个服务用户发送 D2D 卸载服务, 即

$$C2: \sum_{j=1}^M \lambda_{i,j}^{\text{d2d}} \leq 1。 \quad (16)$$

为了保证 RU_s 在执行任务卸载时维持高效的数据传输速率防止数据传输阻塞, 假设单个 RU 的数据传输速率应不大于最大数据传输速率约束 R_i^{max} , 即当 $\lambda_i^{\text{local}} = 0$ 时,

$$C3: \lambda_i^{\text{mec}} R_i^{\text{mec}} + \sum_{j=1}^M \lambda_{i,j}^{\text{d2d}} R_{i,j}^{\text{d2d}} \leq R_i^{\text{max}}。 \quad (17)$$

在本文中, 假设 R_i^{max} 是一个给定常数, 由用户任务的特性决定。每个请求用户卸载任务到 MEC 服务器以及服务用户的传输功率都受其最大上行功率的限制, 即

$$C4: p_i^{\text{mec}} \leq p_{\text{max}}^{\text{mec}}, \quad (18)$$

$$C5: p_i^{\text{d2d}} \leq p_{\text{max}}^{\text{d2d}}。 \quad (19)$$

式(18)中 $p_{\text{max}}^{\text{mec}}$ 表示 RU_i 卸载任务到 MEC 服务器的最大上行传输功率, 式(19)中 $p_{\text{max}}^{\text{d2d}}$ 表示 RU_i 卸载任务到 SU_j 的最大上行传输功率。

为了提高用户的续航能力, 所有移动用户通过 SWIPT 技术收集的能量必须大于其自身消耗的能量^[18], 具体约束表示为

$$C6: \lambda_i^{\text{local}} E_i^{\text{local}} + \lambda_i^{\text{mec}} E_i^{\text{mec}} + \lambda_{i,j}^{\text{d2d}} E_{i,j}^{\text{d2d},t} \leq E_i^{\text{eh}}, \quad (20)$$

$$C7: E_{i,j}^{\text{d2d},e} \leq E_j^{\text{eh}}, 1 \leq j \leq M。 \quad (21)$$

式(20)中 E_i^{eh} 表示 RU_i 收集的能量, 式(21)中 E_j^{eh} 表示 SU_j 收集的能量。

在最大可容忍时延内所有 RU_s 同时进行计算, 系统总时延不超过 T^{max} , 即

$$C8: 0 \leq \sum_{i=1}^N (\lambda_i^{\text{local}} T_i^{\text{local}} + \lambda_i^{\text{mec}} T_i^{\text{mec}} + \sum_{j=1}^M \lambda_{i,j}^{\text{d2d}} T_{i,j}^{\text{d2d}}) \leq T^{\text{max}}。 \quad (22)$$

2.2 问题描述

本文的目标是在最大可容忍时延内, 确保全部请

求用户的计算任务能够成功计算的情况下, 最小化请求用户总能耗。每个 RU 的能耗可能包括本地执行能耗、MEC 卸载传输能耗、D2D 卸载传输能耗。为了求解该数学问题, 通过数学表达式得到目标函数 E^{total} 为

$$E^{\text{total}} = \sum_{i=1}^N (\lambda_i^{\text{local}} E_i^{\text{local}} + \lambda_i^{\text{mec}} E_i^{\text{mec}} + \sum_{j=1}^M \lambda_{i,j}^{\text{d2d}} E_{i,j}^{\text{d2d},t})。 \quad (23)$$

根据上述目标函数和优化约束条件, 基于请求用户能耗最小化的联合任务卸载和功率分配优化问题可以表述为

$$P1 \quad \min_{\lambda_i^{\text{local}}, \lambda_i^{\text{mec}}, \lambda_{i,j}^{\text{d2d}}, p_i^{\text{mec}}, p_i^{\text{d2d}}} E^{\text{total}}, \text{ s. t. } C1 \sim C8。 \quad (24)$$

问题 P1 中包含了整数变量 $\lambda_i^{\text{local}}, \lambda_i^{\text{mec}}, \lambda_{i,j}^{\text{d2d}}$ 和实数变量 $p_i^{\text{mec}}, p_i^{\text{d2d}}$, 是一个非线性混合整数规划问题, 不便于直接求解。

2.3 问题求解

针对优化问题 P1, 本文求解的思路如下: 首先根据整数变量和实数变量将原问题解耦为两个子问题, 即功率分配子问题和任务卸载子问题, 然后再分别求解。

2.3.1 求解功率分配子问题

由式(23)可知, E^{total} 由 3 部分组成, 分别对应 3 种不同的计算卸载模式。因为本文研究的是请求用户的总能耗, 所以 E^{total} 不包括任务通过 D2D 卸载到 SU_j 上执行所消耗的能量。由于 $E_i^{\text{mec}}, E_{i,j}^{\text{d2d},t}$ 分别与变量 $p_i^{\text{mec}}, p_i^{\text{d2d}}$ 有关, 所以下面分别考虑 MEC 卸载模式和 D2D 卸载模式, 求解能耗最小化问题。

在 MEC 卸载模式下, $\lambda_i^{\text{local}} = 0, \lambda_i^{\text{mec}} = 1, \lambda_{i,j}^{\text{d2d}} = 0$, RU_s 的总能耗最小化问题可以表示为

$$P2.1a \quad \min_{p_i^{\text{mec}}} E_i^{\text{mec}}, \text{ s. t. } C3, C4, C6, C8。 \quad (25)$$

上述问题是一个非线性分数规划问题, 是非凸的, 同样很难直接获得最优解。本文利用典型的 Dinkelbach 方法^[17] 对非线性分数规划问题进行求解, 提出最优的功率分配方案。为了方便后文分析, 记为

$$f(p^{\text{mec}}) = \sum_{i=1}^N p_i^{\text{mec}} I_i, \quad (26)$$

$$g(p^{\text{mec}}) = \sum_{i=1}^N W^b \log_2 \left(1 + \frac{p_i^{\text{mec}} h_i}{\sigma^2} \right)。 \quad (27)$$

令 $F(p^{\text{mec}}, \mu) = f(p^{\text{mec}}) - \mu g(p^{\text{mec}})$, 其中 μ 为

一个正的参数因子, p^{mec} 为总传输功率。基于优化问题 P2. 1a, 形成一个新的优化问题:

$$\text{P2. 1a}^* \quad \min_{p^{\text{mec}}} F(p^{\text{mec}}, \mu), \text{ s. t. } \quad \text{C3, C4, C6, C8.} \quad (28)$$

根据 Ghazi 等^[19] 给定的定理, 通过实现 $F(\mu) = 0$, 可以证明优化目标 P2. 1a 可以等价于 P2. 1a*。 $f(p^{\text{mec}}), g(p^{\text{mec}})$ 是连续函数和实值函数, 且 $g(p^{\text{mec}})$ 通常大于 0; $p^{\text{mec}} \in S$, 其中 S 是优化问题 P2. 1a 和 P2. 1a* 中参数 p^{mec} 的定义域。下面给出使用 Dinkelbach 方法的具体过程。

基于 Dinkelbach 方法求解功率分配子问题的算法步骤如下。

步骤 1: 取 $p_0^{\text{mec}} \in S$, 计算 $\mu_1 = \min f(p_0^{\text{mec}})/g(p_0^{\text{mec}})$ 和令 $i = 0$ 。

步骤 2: 决定

$$p_i^{\text{mec}} = \operatorname{argmin}_{p_i^{\text{mec}} \in S} \{ \min \{ f(p^{\text{mec}}) - \mu_i g(p^{\text{mec}}) \} \}。$$

步骤 3: 如果 $F(\mu_i) = 0$, 则 p_i^{mec} 就是在值 μ_i 下问题(25)的最优解决方案, 算法停止; 否则转到步骤 4。

步骤 4: 计算 $\mu_i = \min f(p_i^{\text{mec}})/g(p_i^{\text{mec}})$; 计算 $i = i + 1$, 转到步骤 2。

上述算法的关键是在步骤 2 中获得功率分配 p_i^{mec} , 因此问题可以转化为如何在给定 μ 下找到子问题 P2. 1a* 的最优方案。运用障碍函数法^[20] 来解决上述问题, 其中目标子问题可以转化为一组不带约束的最小化问题。子问题 P2. 1a* 的障碍函数为

$$\varphi(p^{\text{mec}}) = - \sum_{i=1}^N \text{lb}(R_i^{\text{max}} - R_i^{\text{mec}}) - \text{lb}(p_{\text{max}}^{\text{mec}} - \sum_{i=1}^N p_i^{\text{mec}}) - \sum_{i=1}^N \text{lb}(E_i^{\text{eh}} - E_i^{\text{mec}}) - \sum_{i=1}^N \text{lb}(T_i^{\text{max}} - T_i^{\text{mec}}), \quad (29)$$

其中 lb 表示以 2 为底的对数。

子问题 P2. 1a* 的最优解决方案能够通过解决如下非约束的最小值问题来估计:

$$\min \Psi_v(p^{\text{mec}}) = -vF(p^{\text{mec}}, \mu) + \varphi(p^{\text{mec}}), \quad (30)$$

其中, $v > 0$, 决定近似值的准确程度。通过求解式(30)得到最优的 $p_i^{\text{mec},*}$ 。

在 D2D 卸载模式下, $\lambda_i^{\text{local}} = 0, \lambda_i^{\text{mec}} = 0, \lambda_{i,j}^{\text{d2d}} = 1$, RUs 的能耗最小化问题可以表示为

$$\text{P2. 1b} \quad \min_{p_i^{\text{d2d}}} E_i^{\text{d2d},t}, \text{ s. t. } \quad \text{C3, C5, C7, C8.} \quad (31)$$

同样地, 优化问题 P2. 1b 可以用解决优化问题

P2. 1a 的方法求解, 最后得到 $p_i^{\text{d2d},*}$ 。

2.3.2 求解任务卸载子问题

根据上述求解功率分配子问题得到的 $p_i^{\text{mec},*}$ 和 $p_i^{\text{d2d},*}$, 将原问题 P1 转化为一个仅与 $\lambda_i^{\text{local}}, \lambda_i^{\text{mec}}, \lambda_{i,j}^{\text{d2d}}$ 有关的子问题, 即

$$\text{P2. 2} \quad \min_{\lambda_i^{\text{local}}, \lambda_i^{\text{mec}}, \lambda_{i,j}^{\text{d2d}}} E^{\text{total}}, \text{ s. t. } \quad \text{C1, C2, C3, C6, C8.} \quad (32)$$

鉴于任务的不可拆分性, 任务只能选择 3 种卸载模式中的 1 种。为了减少 RUs 的能耗, 优先考虑卸载计算。注意这里一个服务用户只能与一个请求用户建立 D2D 通信链接进行卸载计算。式(32)中的优化问题就可以变成二部图中的一对一匹配问题, 可以用匈牙利算法^[21] 求解。

为了解决问题 P2. 2, 本文将优化问题 P2. 1(包含 P2. 1a 和 P2. 1b 两个子问题)映射为一个完全加权二部图 $G^0 = (V1, V2, E, W)$, $V1$ 和 $V2$ 表示顶点集合, 其中 $V1$ 表示由还未确定卸载模式的剩余 RUs 组成的集合, $V2$ 表示由各种任务卸载模式组成的集合。 $E = \{e(v1, v2)\}$ 表示 $V1$ 中的一个顶点 $v1$ 到 $V2$ 中的一个顶点 $v2$ 之间边的集合, $W = \{w(v1, v2)\}$ 表示边 $e(v1, v2) \in E$ 的权重, 即给定不同卸载模式下 RUs 能量消耗的最小值。

基于匈牙利算法求解任务卸载子问题的算法步骤如下。

步骤 1: 找到一个初始可行顶点标记为 $l(v1)$ 。

步骤 2: 对于给定的 $l(v1)$, 从 G^0 中确定 G_t^0 , 并选择最大匹配 H 。

步骤 3: 如果 H 是最优匹配, 则求解式(32)中的优化问题; 否则, 在 $G_{t_0}^0$ 中选择 H 未被分配的标签。令 $P = V1, T = \emptyset, \psi$ 表示空集。

步骤 4: $N_{G_t^0}(S) = T$ 表示点集。如果 $N_{G_t^0}(P) \neq T$, 返回步骤 3; 否则计算 $\Delta = \min_{v1, v2} (l(v1) + l(v2) + l(v2) - w(v1, v2))$, $v1 \in P, v2 \in V2 - T$ 并定义

$$l'(v1) = \begin{cases} l(v1) - \Delta, & v1 \in P \\ l(v1) + \Delta, & v1 \in T. \\ l(v2), & \text{其他} \end{cases} \quad (33)$$

步骤 5: 用 $l'(v1)$ 替换 $l(v1)$, 返回步骤 2。

反复执行上述过程, 可以得到最优的计算卸载策略, 即 $\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*}$ 。

能耗最小化任务卸载算法的具体过程描述如下:

算法 1 能耗最小化任务卸载算法

输入: W^b, W^d, σ^2

输出: $\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*}, p_i^{\text{mec},*}, p_i^{\text{d2d},*}$,

$E^{\text{total},*}$

1. 初始化: $\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*} \in \{0,1\}, R_i^{\text{max}} > 0, p_{\text{max}}^{\text{mec}} > 0, p_{\text{max}}^{\text{d2d}} > 0$;
2. if ($\lambda_i^{\text{local}} = 0 \& \& \lambda_i^{\text{mec}} = 1 \& \& \lambda_{i,j}^{\text{d2d}} = 0$) then
3. 将 P1 转化为 P2. 1a;
4. 根据 Dinkelbach 方法, 将 P2. 1a 转化为 P2. 1a* ;
5. 然后根据障碍函数法求得 RU i 到 MEC 服务器的上行传输功率 $p_i^{\text{mec},*}$;
6. else if ($\lambda_i^{\text{local}} = 0 \& \& \lambda_i^{\text{mec}} = 0 \& \& \lambda_{i,j}^{\text{d2d}} = 1$) then
7. 将 P1 转化为 P2. 1b;
8. 同样根据障碍函数法求得 RU i 到 SU j 的上行传输功率 $p_i^{\text{d2d},*}$;
9. end if
10. 将 $p_i^{\text{mec},*}, p_i^{\text{d2d},*}$ 代入 P1, 得到 P2. 2;
11. 根据匈牙利算法求得 $\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*}$;
12. 将 $\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*}$ 代入式(23), 求得 $E^{\text{total},*}$;
13. 输出最优解 ($\lambda_i^{\text{local},*}, \lambda_i^{\text{mec},*}, \lambda_{i,j}^{\text{d2d},*}, p_i^{\text{mec},*}, p_i^{\text{d2d},*}$) 和最优目标解 $E^{\text{total},*}$ 。

3 仿真实验与结果分析

仿真实验在 R2018b 版本的 MATLAB 上实现。本文采用蒙特卡罗实验方法对所提策略进行性能评估, 假设信道系数独立同分布, 且服从瑞利分布。模拟区域的大小为 $200 \text{ m} \times 200 \text{ m}$, 假设移动用户均匀分布在模拟区域内。为方便实验对比分析, 仿真实验参数设置^[6]如下: 最大数据传输速率 $R^{\text{max}} = 30 \text{ kbps}$, MEC 最大上行功率 $p_{\text{max}}^{\text{mec}} = 5 \text{ W}$, D2D 最大上行功率 $p_{\text{max}}^{\text{d2d}} = 1 \text{ W}$, 基站到请求用户 i 的下行功率 $p_i^{\text{ch}} = 10 \text{ W}$, 能量转换率 $\beta = 0.7$, 功率分流比率 $\rho = 0.5$, 能耗系数 $\kappa = 10^{-28}$, 完成 RU i 任务的计算资源量 = 5 Gigacycles, RU i 的计算能力 $F_i = 1 \text{ Gigacycles/s}$, MEC 服务器的计算能力 $F^{\text{mec}} = 30 \text{ Gigacycles/s}$, SU j 的计算能力 $F_j^{\text{d2d}} = 1.2 \text{ Gigacycles/s}$, SU i 输入数据大小 $I_i = 1 \text{ Mbits}$, 最大可容忍时延 $T^{\text{max}} = 0.1 \text{ s}$ 。

设置 3 个基准策略与本文所提策略进行对比实验, 即本地执行、MEC 卸载策略和 D2D 卸载策略 (图中简称 MEC 卸载和 D2D 卸载)。MEC 卸载策略表示任务执行仅依靠本地执行和 MEC 卸载模式执行; D2D 卸载策略表示任务执行仅依靠本地执行和 D2D

卸载模式执行; 而本文所提策略则是本地执行、MEC 卸载、D2D 卸载 3 种模式联合执行。分别给出不同的请求用户数量、噪声功率、BS 的信道带宽等, 观察本文策略以及 3 个基准策略 RU s 总能耗的变化情况, 分析本文所提出的任务卸载策略的性能及有效性。

第一个实验是观察不同的请求用户数量 N 对 RU s 总能耗的影响, 实验结果如图 4 所示。在该仿真中, 考虑服务用户数量 M 分别为 4、6、8 的情况。从图 4 可以看出, 随着 RU s 数量 N 的增多, 所有卸载策略 RU s 的总能耗也随之增加。当 $N \leq 10$ 时, 本文提出的策略与 MEC 卸载策略中 RU s 的总能耗保持不变, 且耗能很少, 这是因为本文假设 MEC 服务器只能同一时刻执行 10 个移动用户的任务, 当 RU s 不大于 10 时, 这两个策略都是将任务卸载到 MEC 服务器上执行。当 $N > 10$ 时, 本文策略由于可以将任务卸载到邻近的服务用户 SU s 执行, 所以 RU s 的总能耗增长缓慢, 且总能耗均低于其他 2 个基准策略。针对不同的 SU s 数量 M , 从本文策略与 D2D 卸载策略 RU s 的总能耗变化趋势来看, SU s 的数量越多, RU s 的总能耗越低, 当 RU s 的数量高于 SU s 时, 总能耗按照相同的斜率增长。总的来说, 将任务全部卸载到 MEC 服务器的策略的执行优于全部进行 D2D 卸载的策略, 而本文策略的性能优于 MEC 卸载策略和 D2D 卸载策略。

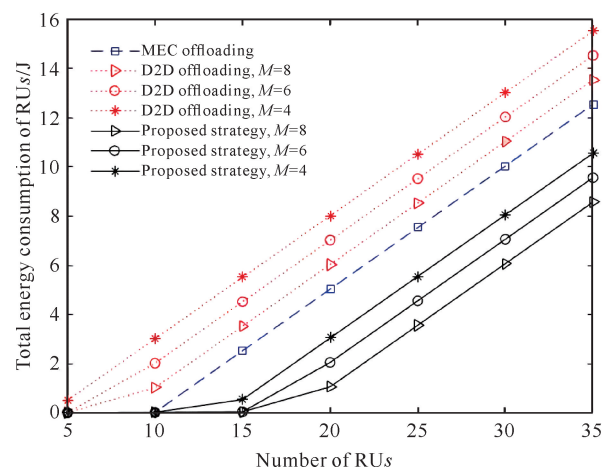


图 4 不同 SU s 数量 M 下 RU s 的数量 N 与 RU s 总能耗的关系

Fig. 4 Relationship between the quantity N of RU and the total energy consumption of RU under different SU s quantity M

图 5 和图 6 分别给出了在不同的噪声功率 σ^2 和基站信道带宽 W^b 下输入数据大小对 RU s 总能耗的影响结果。实验设置 RU $s = 15$, SU $s = 5$, σ^2 设置

-75 dBm 和 -85 dBm 2 种情况。从图 5 可以看出, RU_s 的总能耗随输入数据大小的增大而缓慢增长。在相同的输入数据大小下 σ^2 越小, RU_s 的总能耗相对较小, 这是因为噪声功率 σ^2 会影响数据传输速率, σ^2 越小数据传输速率越大, 数据传输速率直接影响传输能耗的大小。从图 6 可以看出, 针对不同的 W^b , 在相同的输入数据大小下 BS 的带宽 W^b 越大, RU_s 的总能耗越小, 这是因为 W^b 与任务卸载到 MEC 服务器的数据传输速率 R_i^{mec} 呈正相关, 当 RU_s 传输任务到 MEC 服务器时, 数据传输速率 R_i^{mec} 越大, RU_s 到 MEC 服务器的传输能耗越小。总的来说, W^b 对请求用户总能耗有一定的影响, 但是影响不大, 随着输入数据量增大, W^b 对请求用户总能耗的影响在小幅度增大。

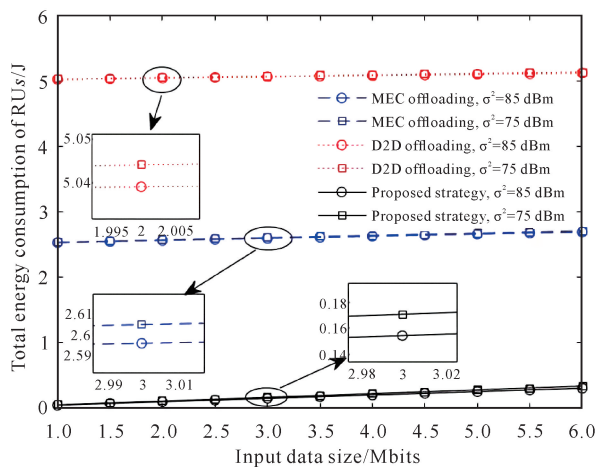


图 5 不同 σ^2 下输入数据大小与 RU_s 总能耗的关系

Fig. 5 Relationship between input data size and the total of RU_s under different σ^2

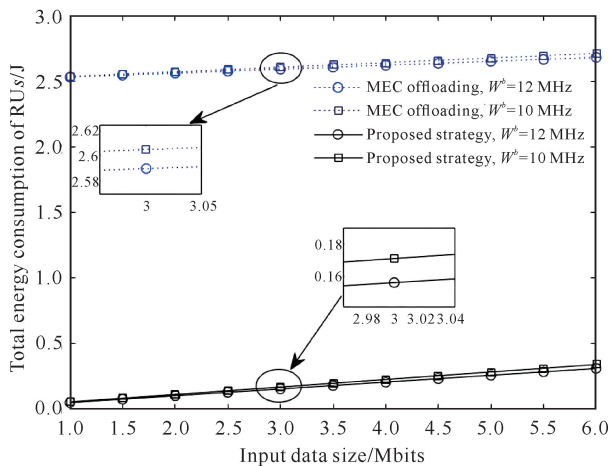


图 6 不同 W^b 下输入数据大小与 RU_s 总能耗的关系

Fig. 6 Relationship between input data size and the total energy consumption of RU_s under different W^b

图 7 给出了本地执行、MEC 卸载、D2D 卸载和本文所提策略等 4 种策略在不同 RU_s 所需的 CPU 周期数下 RU_s 的总能耗, 该实验中设置 $RU_s = 20$, $SU_s = 5$ 。从图 7 可以看出, 随着请求用户 RU_s 的 CPU 周期数的增大, 消耗的总能量也随之增大。本文所提策略的 RU_s 总能耗明显低于其他 3 个基准策略, 这表明本文所提策略能够较好地提高系统性能。这是因为本文所提策略优先选择 MEC 卸载, 当 MEC 服务器中资源竞争加剧时, 又考虑 D2D 卸载, 可合理利用附近空闲的设备。从图 7 还可以看出, MEC 卸载和 D2D 卸载的 RU_s 总能耗要低于本地执行的能量消耗, 这得益于 MEC 服务器和服务用户的高效资源利用。MEC 卸载策略比 D2D 卸载策略消耗的能量更加少, 主要是 MEC 服务器为移动设备任务的执行提供更多可用的资源和能量, 而 D2D 卸载虽然在一定程度上能够减少本地资源的使用, 但是资源还是不如 MEC 服务器充足。

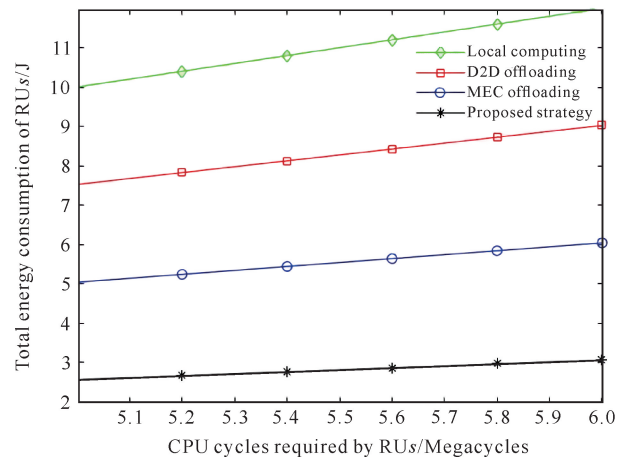


图 7 不同 RU_s 所需的 CPU 周期数下的 RU_s 总能耗

Fig. 7 Total energy consumption of RU_s under different CPU cycles required by RU_s

图 8 给出了在不同的 D2D 链路带宽 W^d 下随着 BS 的带宽 W^b 增大对 RU_s 总能耗的影响结果。如图 8 所示, 随着 W^b 的增大, 请求用户消耗的总能量随之降低; 随着 D2D 链路带宽 W^d 的增大, 请求用户消耗的总能量随之减小, 是因为 W^d 与 D2D 数据传输速率 R^{d2d} 有关, 且呈正比关系。当请求用户传输任务到服务用户时, 数据传输速率越大, 能量消耗越小。从实验结果来看, 本文所提策略整体优于 MEC 卸载策略, 这是由于本文所提策略比 MEC 卸载策略多了一种 D2D 卸载方式, 请求用户可以优先考虑将任务卸载到邻近的空闲设备。

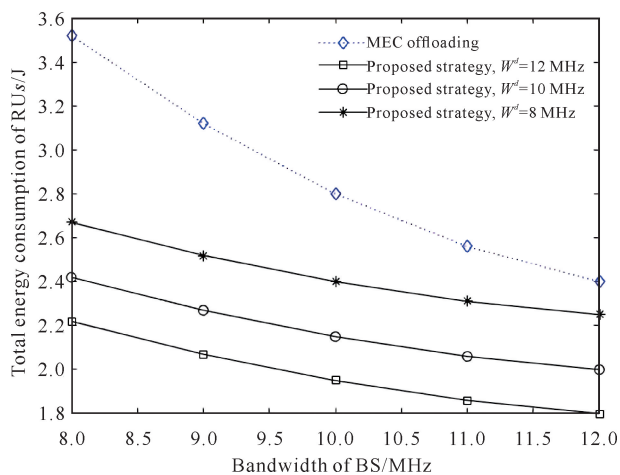


图8 不同 W^d 下 BS 的带宽与 RUs 总能耗的关系

Fig. 8 Relationship between bandwidth of BS and total energy consumption of RUs under different W^d

综合以上实验结果可知,针对由 SWIPT 技术供电后移动用户功率分配和任务卸载问题,本文所提策略是在 MEC 卸载策略的基础上联合 D2D 卸载策略,使得 MEC 资源争用问题得以解决,也避免了本地计算资源不足的问题,保证了移动用户的任务执行过程不会产生中断,避免对计算结果造成影响。因此,本文策略优于其他几种对比策略。

4 结论

本文在保证通信服务质量的条件下,构建了一个基于 SWIPT 的多用户 D2D 通信辅助的 MEC 系统模型,并基于该模型分别建立能量收集和任务执行两个数学模型;考虑任务卸载和功率分配对目标函数和约束条件的影响,建立了一个非线性混合整数规划问题模型,提出了一种基于 SWIPT 的多用户 D2D 通信辅助 MEC 任务卸载策略。该策略以请求用户总能耗最小为目标,通过数学变化将原非线性混合整数规划问题转换为任务卸载和功率分配两个子问题:首先,利用 Dinkelbach 方法将目标函数转换为非分式规划问题,并用障碍函数法将目标子问题转化为一组不带约束的最小化问题来求解功率分配子问题;然后,基于匈牙利算法将任务卸载子问题转化为二部图中的一对一匹配问题进行求解。仿真实验结果表明,在相同的 SWIPT 技术为移动设备供电的情况下,本文所提策略在最小化系统请求用户总能耗方面具有良好的性能。

参考文献

[1] 谢人超,廉晓飞,贾庆民,等. 移动边缘计算卸载技术综

述[J]. 通信学报,2018,39(11):138-155.

- [2] YOU C, HUANG K. Multiuser resource allocation for mobile-edge computation offloading [C]//2016 IEEE Global Communications Conference (GLOBECOM). Washington DC, USA: IEEE, 2016: 1-6.
- [3] YOU C, HUANG K, CHAE H, et al. Energy-efficient resource allocation for mobile-edge computation offloading [J]. IEEE Transactions on Wireless Communications, 2017, 16(3): 1397-1411.
- [4] ALI Z, JIAO L, BAKER T, et al. A deep learning approach for energy efficient computational offloading in mobile edge computing [J]. IEEE Access, 2019, 7: 149623-149633.
- [5] 薛建彬,安亚宁. 基于边缘计算的新型任务卸载与资源分配策略[J]. 计算机工程与科学, 2020, 42(6): 959-965.
- [6] CHAI R, LIN J, CHEN M, et al. Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems [J]. IEEE Systems Journal, 2019, 13(4): 4110-4121.
- [7] HE Y, REN J, YU G, et al. Joint computation offloading and resource allocation in D2D enabled MEC networks [C]//ICC 2019-2019 IEEE International Conference on Communications (ICC). Piscataway, NJ, USA: IEEE, 2019: 1-6.
- [8] KAI Y, WANG J, ZHU H. Energy minimization for D2D-assisted mobile edge computing networks [C]//ICC 2019-2019 IEEE International Conference on Communications (ICC). Piscataway, NJ, USA: IEEE, 2019: 1-6.
- [9] MAO Y, ZHANG J, LETAIEG K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605.
- [10] SREELAKSHMY K R, JACOB L. Simultaneous wireless information and power transfer in heterogeneous cellular networks with underlay D2D communication [J]. Wireless Networks, 2020, 26(5): 3315-3330.
- [11] WEN Z, YANG K, LIU X, et al. Joint offloading and computing design in wireless powered mobile-edge computing systems with full-duplex relaying [J]. IEEE Access, 2018, 6: 72786-72795.
- [12] CHEN F, FU J, WANG Z, et al. Joint communication and computation resource optimization in FD-MEC cellular networks [J]. IEEE Access, 2019, 7: 168444-168454.
- [13] FU J, HUA J, WEN J, et al. Optimization of energy consumption in the MEC-assisted multi-user FD-

- SWIPT system [J]. IEEE Access, 2020, 8: 21345 - 21354.
- [14] FU J, HUA J, WEN J, et al. Optimization of achievable rate in the multiuser satellite IoT system with SWIPT and MEC [J]. IEEE Transactions on Industrial Informatics, 2021, 17(3): 2072-2080.
- [15] 马惠荣, 陈旭, 周知, 等. 绿色能源驱动的移动边缘计算动态任务卸载[J]. 计算机研究与发展, 2020, 57(9): 1823-1838.
- [16] 李陶深, 施安妮, 王哲, 等. 基于 SWIPT 的吞吐量最优化 NOMA 全双工中继选择策略[J]. 通信学报, 2021, 42(5): 87-97.
- [17] ZHOU N, HU J, HOU J. Research on energy efficiency of NOMA-SWIPT cooperative relay network using GS-DinkelBach algorithm [J]. Sensors, 2021, 21(17): 5720.
- [18] 李陶深, 宁倩丽, 王哲. SWIPT-NOMA 机会协作系统的优化方案[J]. 通信学报, 2020, 41(8): 141-154.
- [19] GHAZI A, ROUBI A. Optimality conditions and DC-Dinkelbach-type algorithm for generalized fractional programs with ratios of difference of convex functions [J]. Optimization Letters, 2021, 15(7): 2351-2375.
- [20] SAMADI P, MOHSENIAN-RAD A H, SCHÖBER R, et al. Optimal real-time pricing algorithm based on utility maximization for smart grid [C]//2010 First IEEE International Conference on Smart Grid Communications. Piscataway, NJ, USA: IEEE, 2010: 415-420.
- [21] KHAN A A, ADVE R S, YU W. Optimizing downlink resource allocation in multiuser MIMO networks via fractional programming and the hungarian algorithm [J]. IEEE Transactions on Wireless Communications, 2020, 19(8): 5162-5175.

Task Offloading Strategy of D2D Communication Assisted MEC Based on SWIPT

DU Lijun¹, LI Taoshen^{1,2* *} , HUANG Yixin¹, QI Zhijun¹

(1. School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China; 2. China - ASEAN International Joint Laboratory of Integrate Transport, Nanning University, Nanning, Guangxi, 530200, China)

Abstract: In order to solve the problems of insufficient calculate capability, high energy consumption and scarce wireless resources of mobile users in 5G mobile communication system, a multi-user Device to Device (D2D) communications assisted Mobile Edge Computation (MEC) system model based on Simultaneous Wireless Information and Power Transfer (SWIPT) is constructed, and a D2D-MEC joint offloading strategy is proposed. The strategy aims to minimize the total energy consumption of the requesting users in the system, and uses binary offloading mode and power splitting mode to offload tasks and collect energy for the requesting users. For the problem of energy consumption minimization is a nonlinear mixed integer programming problem, the original problem is decoupled into two independent sub-problems: power allocation and computing task offloading according to integer variables and real variables, and then the optimal solutions of the two sub-problems are obtained by Dinkelbach method and Hungarian algorithm. The simulation results show that the proposed strategy is superior to the traditional D2D offloading strategy and MEC offloading strategy, which effectively reduces the total energy consumption of the requesting users and improves the efficiency of task execution.

Key words: simultaneous wireless information and power transfer; mobile edge computing; D2D communications; task offloading; power allocation

责任编辑: 米慧芝