# MEC 多服务器启发式联合任务卸载和资源分配策略

路 亚

(重庆电子工程职业学院人工智能与大数据学院 重庆 401331)

**摘 要** 针对移动边缘计算(Mobile Edge Computing,MEC)网络中的多服务器联合任务卸载和资源分配问题,提出一种 MEC 多服务器启发式联合任务卸载和资源分配策略。将原始优化问题表述为一个混合整数非线性规划,并分解成具有固定任务卸载决策的资源分配(Resource Allocation,RA)优化问题和任务卸载(Task Offloading,TO)最优值函数问题。将 RA 问题进一步解耦为上行链路功率分配和计算资源分配两个独立子问题,并分别采用拟凸和凸优化技术加以解决。提出一种启发式算法解决 TO 问题,可以在多项式时间内实现目标问题的次优解。仿真结果表明,与其他优化策略相比,该策略能够很好地实现最优解,显著提高系统的平均卸载效率。

**关键词** 移动边缘计算 分布式部署 计算卸载 资源分配

**中图分类号** TP393 **文献标志码** A DOI: 10.3969/j.issn.1000-386x.2020.10.013

## HEURISTIC JOINT TASK OFFLOADING AND RESOURCE ALLOCATION STRATEGY FOR MEC MULTI-SERVER

Lu Ya

(*College of Artificial Intelligence and Big Data, Chongqing College of Electronic Engineering, Chongqing 401331, China*)

**Abstract** Aiming at the problem of multi-server joint task unloading and resource allocation in mobile edge computing networks, this paper proposes a heuristic joint task offloading and resource allocation strategy for EMC multi-server. The original optimization problem was expressed as a mixed integer nonlinear programming, and then it was decomposed into resource allocation (RA) optimization problem and task offloading (TO) optimal value function with fixed task offloading decision. The RA problem was further decoupled into two independent sub-problems of uplink power allocation and computational resource allocation, and solved by quasi-convex and convex optimization techniques respectively. A heuristic algorithm was proposed to solve the TO problem, which could achieve the suboptimal solution of the target problem in polynomial time. The simulation results show that compared with other optimization strategies, this strategy can achieve the optimal solution well and significantly improve the average unloading efficiency of the system.

**Keywords** Mobile edge computing Distributed deployment Task offloading Resource allocation

## 0 引 言

随着移动应用程序和物联网的快速增长,对云基础设施和无线接入网络(如超低延迟、用户体验连续性和高可靠性)的要求也越来越高[1]。为了将电信、IT 和云计算结合起来,直接从网络边缘提供云服务,移动边缘计算概念应运而生[2]。与传统云计算系统不同,MEC 服务器为网络运营商所有,并直接在蜂窝基站或本地无线接入点通过通用计算平台实施,这使得 MEC 可以在靠近最终用户的地方执行应用程序,从而大大减少端到端的延迟,减轻回程网络的负担[3-4]。

由于 MEC 的出现,资源受限的移动设备可以将计算任务卸载到 MEC 服务器上,从而使其支持各种新的服务和应用,如增强现实、物联网、自动驾驶和图像处理[5]。由于设备与 MEC 服务器之间在上行链路无线信道中的通信需要,任务卸载在延迟和能耗方面会产生额外的开销。此外,在一个拥有大量卸载用户的系

收稿日期:2019-05-23。重庆市重点产业共性关键技术创新专项重点研发项目(cstc2017zdcy-zdyfx0017);重庆市教委科学技术研究项目(自然科学类)(KJ132207)。路亚,副教授,主研领域:云计算,网络与信息安全。

统中，MEC 服务器上有限的计算资源严重影响了任务执行延迟。因此，卸载决策和执行资源分配成为实现高效计算卸载的关键问题[6]。

文献[7]提出一种完全卸载优化算法，采用将边缘设备中的任务全部卸载到 MEC 服务器上的策略，减少了任务执行的时间和能耗。文献[8]提出一种部分卸载优化算法，采取将部分满足条件的计算任务卸载到云服务器，剩余任务在边缘设备执行的方式实现能耗和时间的优化。文献[9]虽然考虑了多用户系统中的联合任务卸载和资源优化问题，但是这种方法只能应用在具有单个 MEC 服务器的系统上。文献[10]在移动边缘计算环境中提出一种多重资源计算卸载能耗优化模型，在计算卸载时综合考虑边缘设备、边缘服务器与云数据中心的负载情况，利用粒子群任务调度算法，降低移动终端能耗。文献[11]提出建立一个设备(D2D)框架，其中网络边缘的大量设备利用网络辅助 D2D 协作进行计算和通信资源共享，但是只考虑了任务分配问题。

与上述方法不同，本文在多 MEC 服务器辅助网络中设计一个整体的解决方案，对联合任务卸载和资源分配进行优化，从而最大限度地提高用户的卸载收益。首先把每个用户的卸载效用建模为任务完成时间和设备能耗改进的加权和；然后将联合任务卸载和资源分配(Joint Task Offloading and Resource Allocation，JTORA)问题作为一个混合整数非线性规划(Mixed Integer Non-linear Program，MINLP)，采用 Tammer 分解方法将高复杂度的原始问题转化为等效的主问题和一组复杂度较低的子问题；最后利用本文提出的低复杂度启发式算法，以次优解的方式解决 JTORA 问题，实现共同优化任务卸载决策和用户上行链路传输功率的目标。

## 1 移动边缘计算

国际标准组织 ETSI 提出的移动边缘计算(MEC)是基于 5G 演进架构的一项新技术，为减少网络延迟，确保高效的网络运营和服务交付，并提高用户体验，通过借助边缘计算中心在无线接入网(Radio Access Network，RAN)内提供 IT 服务环境、云计算和存储功能。

### 1.1 MEC 基本架构

移动边缘计算的关键要素是集成在 RAN 元素上的移动边缘计算 IT 应用服务器。MEC 的基本框架如图 1 所示，该框架从一个比较宏观的层次出发，对 MEC 架构下不同的功能实体进行了网络层、移动边缘主机层和移动边缘系统层的划分。其中：MEC 主机层包含主机(ME host)和相应的主机层管理实体(ME host-level management entity)，ME 主机又可以进一步划分为 ME 平台(ME platform)、ME 应用(ME application)和虚拟化基础设施(virtualization infrastructure)；网络层主要包含 3GPP 蜂窝网络、本地网络和外部网络等相关的外部实体，该层主要表示 MEC 工作系统与局域网、蜂窝移动网或者外部网络的接入情况；最上层是 ME 系统层的管理实体，负责承载应用程序及 MEC 系统的全局掌控。

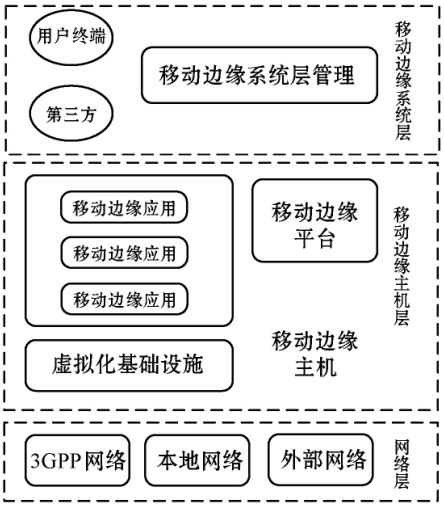



图 1 MEC 架构示意图

### 1.2 MEC 部署方案

由于具备计算和存储功能的 MEC 服务器为边缘计算提供服务，因此 MEC 服务器在网络中的部署位置是至关重要的。4G 网络中边缘服务器部署存在多种方案：小基站云(Small Cell Cloud，SCC)部署方案、移动微型云(Mobile Micro Cloud，MMC)部署方案、快速移动私人云(Fast Move Personal Cloud，FMPC)部署方案和漫游云(Follow Me Cloud，FMC)部署方案。各个部署方案的优缺点对比如表 1 所示。MEC 服务器部署方案的选择是根据物理部署约束、可扩展性以及性能指标的综合考虑。图 2 给出了 FMC 部署方案示意图，该方案主要是通过在分布式的数据中心位置部署服务器来提供边缘服务。

表 1 边缘服务器部署方案对比

| 方案 | 服务器部署位置 | 控制实体 | 控制实体部署位置 | 优点 | 缺点 |
|---|---|---|---|---|---|
| SCC | SCeNBs | SCM | L-SCM 和 R-SCM | 靠近网络，较低时延 | 安装成本高，存在安全问题 |
| MMC | eNodes | 无 | — | 较低时延，连续性强 | 增大信令开销，存在安全问题 |
| FMPC | RAN 侧 | MC | SDN 中 | 较低时延，信令开销小 | 存在安全问题 |
| FMC | CN 侧 | FMCC | CN 后 | 安全问题得以解决 | 较高时延，占用核心网资源 |

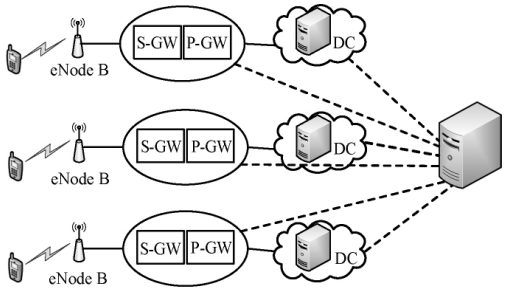



图 2　FMC 部署方案示意图

5G 网络在根本架构上不同于 4G 网络，所以为了让 MEC 更好地融合 5G 网络，需要根据需求重新设计 MEC 部署方案，如图 3 所示。MEC 处在接入网与核心网融合的部分，通过 NEF 接入 5G 网络。该方案根据平台应用相关信息，通过 5G 控制面的应用功能(AF)直接或者间接地将 MEC 用户面应用传递给策略控制功能模块(PCF)，控制会话管理功能模块(SMF)进而影响用户面功能模块(UPF)的选择。

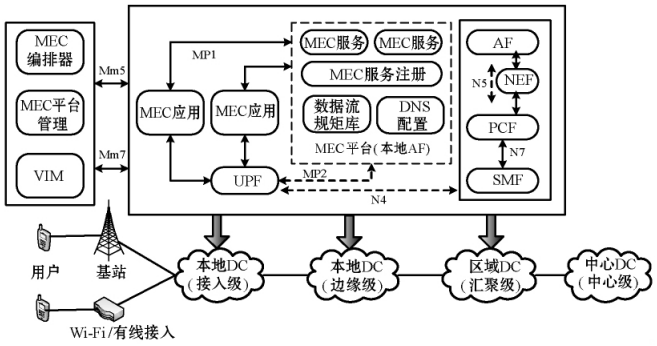



图 3　5G MEC 融合架构示意图

## 2　多服务器 MEC 系统的 JTORA 建模及启发式整体优化方案

多服务器 MEC 系统是指在每个基站都配备一个 MEC 服务器，为资源受限的移动用户(如智能手机、平板电脑和可穿戴设备)提供计算卸载服务的平台。每个 MEC 服务器可以是物理服务器，或者是网络运营商提供的具有中等计算能力的虚拟机，通过相应的 BS 提供的无线通道与移动设备连接通信。每个移动用户都可以通过附近的 BS 将计算任务卸载到 MEC 服务器。

本文主要有两点创新。一是将联合任务卸载和资源分配问题表述为一个混合整数非线性规划问题，通过对任务卸载决策、用户上行链路传输功率和计算资源分配的共同优化来实现卸载效用的最大化；二是提出一种低复杂度启发式算法来解决任务卸载最优值问题，可以在多项式时间内达到次优解。

### 2.1　多服务器 MEC 建模

本文将移动系统中的一组用户和 MEC 服务器分别表示为 $u=\{1,2,\cdots,U\}$ 和 $s=\{1,2,\cdots,S\}$。多服务器 MEC 系统如图 4 所示。

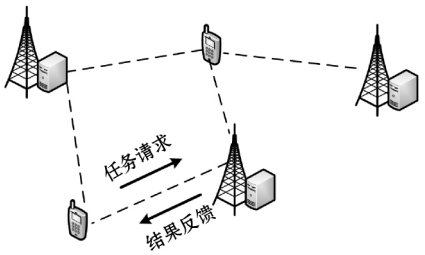



图 4　多服务器 MEC 系统

#### 2.1.1　本地计算任务

假设单个用户 $u\in U$ 每次只有一个计算任务，即 $T_u$，它是原子的、不可分的。每个 $T_u$ 由两个参数 $d_u$、$c_u$ 组成，其中：$d_u$ 表示将程序执行(包括系统设置、程序代码和输入参数等)从本地设备传输到 MEC 服务器所需的请求数据量；$c_u$ 表示完成任务的计算量。每个任务可以在用户设备上执行本地卸载，或者卸载到 MEC 服务器。如果将计算任务卸载到 MEC 服务器，移动用户能够节省执行任务的能量，但是将任务请求发送到上行链路需要消耗额外的时间和能量。

令 $f_u^l>0$ 表示用户 $u$ 在 CPU 方面的本地计算能力。如果用户 $u$ 在本地执行任务，则任务完成时间为 $t_u^l=\dfrac{c_u}{f_u^l}$ s。为了计算用户设备在本地执行任务时的能耗，采用计算周期的能耗模型 $\varepsilon=kf^2$ 表示。其中，$k$ 是依赖芯片结构的能量系数，$f$ 是 CPU 频率。因此，用户 $u$ 在本地执行任务 $T_u$ 时的能耗 $E_u^l$ 计算为：

$$E_u^l=k\left(f_u^l\right)^2c_u \tag{1}$$

#### 2.1.2　任务上载

若用户 $u$ 选择将任务 $T_u$ 卸载到 MEC 服务器，则产生的延迟包括：在上行链路上将请求传输到 MEC 服务器的时间 $t_{\mathrm{up}}^u$，在 MEC 服务器上执行任务的时间 $t_{\mathrm{exe}}^u$ 以及在下行链路上将结果从 MEC 服务器返回用户的时间 $t_{\mathrm{down}}^u$。由于结果的大小通常比请求小得多，而且下行数据速率比上行数据速率高得多，本文在计算中省略结果传输的延迟 $t_{\mathrm{down}}^u$。

本文采用正交频分多址(Orthogonal Frequency Division Multiple Access，OFDMA)技术作为上行中的接入方案，其中工作频段 $B$ 被划分为 $N$ 个等子频段 $W=B/N$。为了确保与同一个基站相关联的用户之间的上行链路传输的正交性，每个用户被分配到一个子频段。因此，每个 $BS$ 可以同时服务最多 $N$ 个用户。设 $n=\{1,2,\cdots,N\}$ 是每个 BS 的可用子带的集合。定义任务卸载变量 $x_{u,s}^j$，其中：$u\in U$，$s\in S$，$j\in N$。$x_{u,s}^j=1$ 表示用户 $u$ 的任务 $T_u$ 被卸载到子波段为 $j$ 的 BS 上，否

则 $x^j_{u,s}=0$。将包含所有任务卸载变量的真实集合 $G$ 定义为 $G=\{x^j_{u,s}\mid u\in U,s\in S,j\in N\}$，任务卸载策略 $\chi$ 表示为 $\chi=\{x^j_{u,s}\in G\mid x^j_{u,s}=1\}$。由于每个任务可以在本地执行，也可卸载到至多一个 MEC 服务器上，因此一个可行的卸载策略必须满足如下限制:

$$\sum_{s\in S}\sum_{j\in N}x^j_{u,s}\leqslant 1\quad \forall u\in U \tag{2}$$

同时，$U_s=\{u\in U\mid \sum_{j\in N}x^j_{u,s}=1\}$ 表示将任务卸载到服务器的用户集合，$U_{\text{off}}=\cup_{s\in S}U_s$ 表示所有执行卸载任务的用户集合。

此外，每个用户和基站都有一个用于上行链路传输的天线。令 $h^j_{u,s}$ 表示子带 $j$ 上用户 $u$ 和 BS 之间的上行链路信道增益，它捕获路径损耗、阴影和天线增益的影响。$P=\{p_u\mid 0<p_u\leqslant P_u,u\in U_{\text{off}}\}$ 表示用户的传输功率，其中 $p_u$ 代表将任务请求 $d_u$ 上传到 BS 时用户 $u$ 的传输功率。当 $\forall u\notin U_{\text{off}}$ 时，$p_u=0$。由于传输到同一基站的用户使用不同的子频段，因此上行小区内干扰得到很好的缓解，但这些用户仍然受到小区间干扰的影响。在这种情况下，子带 $j$ 上用户 $u$ 到 BS 的信号与干扰加噪声比(Signal to Interference plus Noise Ratio，SINR) 如下:

$$\gamma^j_{u,s}=\frac{p_u h^j_{u,s}}{I^j_{u,s}+\sigma^2}\quad \forall u\in U,s\in S,j\in N \tag{3}$$

式中: $\sigma^2$ 是背景噪声方差; $I^j_{u,s}=\sum_{r\in S\setminus\{s\}}\sum_{k\in U_r}x^j_{k,r}p_k h^j_{k,s}$ 表示子带 $j$ 上与其他 BS 相关联的所有用户在小区内的累积干扰。由于每个用户仅在单个子带上传输数据，所以用户 $u$ 向 BS 发送数据时的速率为:

$$R_{u,s}(\chi,P)=W\log_2(1+\gamma_{u,s}) \tag{4}$$

式中: $\gamma_{u,s}=\sum_{j\in N}\gamma^j_{u,s}$。那么，用户 $u$ 在上行发送任务请求 $d_u$ 时的传输时间为:

$$t^u_{\text{up}}=\sum_{s\in S}\frac{x_{u,s}d_u}{R_{u,s}(\chi,P)}\quad \forall u\in U \tag{5}$$

式中: $x_{u,s}=\sum_{j\in N}x^j_{u,s}$。

### 2.1.3 MEC 计算资源

每个 BS 上的 MEC 服务器能够同时向多个用户提供计算卸载服务。每个 MEC 服务器提供给关联用户共享的计算资源由计算速率 $f_s$ 进行量化。服务器从用户接收到卸载的任务后，代表用户执行该任务，完成后返回输出结果给用户。将计算资源分配策略定义为 $F=\{f_{u,s}\mid u\in U,s\in S\}$，其中 $f_{u,s}>0$ 是 BS 分配给卸载任务 $T_u$ 的计算资源量。因此，当 $\forall u\notin U_s$ 时，$f_{u,s}=0$。此外，一个可行的计算资源分配策略必须满足计算资源约束，表示为:

$$\sum_{u\in U}f_{u,s}\leqslant f_s\quad \forall s\in S \tag{6}$$

给定计算资源分配 $\{f_{u,s},s\in S\}$，任务 $T_u$ 在 MEC 服务器上的执行时间为:

$$t^u_{\text{exe}}=\sum_{s\in S}\frac{x_{u,s}c_u}{f_{u,s}}\quad \forall u\in U \tag{7}$$

### 2.1.4 用户卸载实用程序

考虑到卸载策略 $\chi$、传输功率 $p_u$ 和计算资源分配 $f_{u,s}$，用户 $u$ 在卸载任务时经历的总延迟如下:

$$t_u=t^u_{\text{up}}+t^u_{\text{exe}}=\sum_{s\in S}x_{u,s}\left(\frac{d_u}{R_{u,s}(\chi,P)}+\frac{c_u}{f_{u,s}}\right) \tag{8}$$

用户 $u$ 上传请求所消耗的能量 $E_u=\frac{p_u t^u_{u,p}}{\xi_u}$，其中 $\xi_u$ 是用户 $u$ 的功放效率。一般假设 $\xi_u=1$。因此，用户 $u$ 的上行链路能耗简化为:

$$E_u=p_u t^u_{up}=p_u d_u\sum_{s\in S}\frac{x_{u,s}d_u}{R_{u,s}(\chi,P)} \tag{9}$$

在移动云计算系统中，用户的 QoE 主要表现为任务完成时间和能耗。在所考虑的场景中，任务完成时间和能耗的相对改善分别以 $\frac{t^l_u-t_u}{t^l_u}$ 和 $\frac{E^l_u-E_u}{E^l_u}$ 为特征。因此，用户 $u$ 的卸载实用程序定义为:

$$J_u=\left(\beta^t_u\frac{t^l_u-t_u}{t^l_u}+\beta^e_u\frac{E^l_u-E_u}{E^l_u}\right)\sum_{s\in S}x_{u,s} \tag{10}$$

式中: $\beta^t_u,\beta^e_u\in[0,1]$ 分别指用户对任务完成时间和能耗的权重，$\beta^t_u+\beta^e_u=1$。实践中，移动用户可以通过不同的节电方式设置 $\beta^e_u$ 的值。如果将太多的任务卸载到 MEC 服务器，可能因为 MEC 服务器上的带宽和计算资源有限而导致过度延迟，用户的 QoE 会降低。因此，如果 $J_u\leqslant 0$，用户 $u$ 不应该将其任务卸载到 MEC 服务器。

## 2.2 联合任务卸载与资源分配启发式整体优化方案

### 2.2.1 JTORA 问题及分解

对于给定的卸载决策 $\chi$、上行链路功率分配 $p$ 和计算资源分配 $f$，将系统实用程序定义为所有用户卸载实用程序的加权和:

$$J(\chi,p,f)=\sum_{u\in U}\lambda_u J_u \tag{11}$$

式中: $\lambda_u$ 是权重系数，可以根据用户类型和计算任务的关键性来设置。将联合任务卸载和资源分配问题转化成系统效用最大化问题，即:

$$\max_{\chi,p,f}J(\chi,p,f) \tag{12}$$

$$\text{s.t.}\quad x_{u,s}^{j}\in\{0,1\}\quad \sum_{s\in S}\sum_{j\in N}x_{u,s}^{j}\leqslant 1$$
$$\sum_{u\in U}x_{u,s}^{j}\leqslant 1\quad 0<p_u\leqslant P_u$$
$$f_{u,s}>0\quad \sum_{u\in U}f_{u,s}\leqslant f_s$$

式中: 第一和第二约束条件表示每个任务可以在本地执行或者卸载到子带 $j$ 上的至多一个服务器; 第三约束条件表示每个 BS 在子带 $j$ 上至多服务一个用户; 第四约束条件表示用户的传输功率预算; 最后两个约束条件表示每个 MEC 服务器必须为与其相关联的每个用户分配相应的计算资源,并且分配给所有相关用户的总计算资源不得超过服务器的计算能力。

式(12)中的 JTORA 问题属于混合整数非线性规划问题。考虑到问题中的变量与用户数量、MEC 服务器数量和子带数量呈线性关系,因此,本文通过设计一个低复杂度、次优解决方案,实现联合任务卸载与资源分配的优化。

通过临时固定二元变量$\{x_{u,s}\}$,式(12)可以被分解为目标和约束条件分离的多个子问题,本文采用 Tammer 分解方法将高复杂度的原始问题转化为等效的主问题和一组复杂度较低的子问题。因此,式(12)中的约束条件可以分解成关于 $\chi$ 的任务卸载(Task Offloading,TO)和关于 $p$、$f$ 的资源分配(Resource Allocation,RA):

$$\max_{x} J^{*}(\chi) \tag{13}$$
$$\text{s.t.}\quad x_{u,s}^{j}\in\{0,1\}\quad \sum_{s\in S}\sum_{j\in N}x_{u,s}^{j}\leqslant 1$$
$$\sum_{u\in U}x_{u,s}^{j}\leqslant 1$$
$$J^{*}(\chi)=\max_{p,f}J(\chi,p,f) \tag{14}$$
$$\text{s.t.}\quad 0<p_u\leqslant P_u\quad f_{u,s}>0$$
$$\sum_{u\in U}f_{u,s}\leqslant f_s$$

式(13)中的问题等同于 TO 问题; 式(14)中的问题等同于 RA 优化问题。

### 2.2.2　JTORA 启发式解决方案

联合任务卸载和资源分配的求解问题等效为分别求解任务卸载和资源分配的优化问题,首先解决式(14)中的 RA 问题,然后使用其解决方案来推导式(13)中 TO 问题的解决方案。

给定满足约束条件的可行任务卸载决策 $\chi$,式(14)可以改写为:

$$J(\chi,p,f)=\sum_{s\in S}\sum_{u\in U_s}\lambda_u(\beta_u^t+\beta_u^e)-V(\chi,p,f) \tag{15}$$

$$V(\chi,p,f)=\sum_{s\in S}\sum_{u\in U_s}\frac{\phi_u+\varphi_u p_u}{\log_2(1+\gamma_{u,s})}+\sum_{s\in S}\sum_{u\in U_s}\frac{\eta_u}{f_{u,s}} \tag{16}$$

式中: $\phi_u=\dfrac{\lambda_u\beta_u^t d_u}{t_u^l W}$; $\varphi_u=\dfrac{\lambda_u\beta_u^e d_u}{E_u^l W}$; $\eta_u=\lambda_u\beta_u^t f_u^l$。

对于特定的卸载决策 $\chi$,式(15)右侧第一项是恒定的,而 $V(\chi,p,f)$ 可以看作是所有卸载用户的总卸载开销,因此,式(14)可以设计为最小化总卸载开销的问题:

$$\min_{p,f}V(\chi,p,f) \tag{17}$$
$$\text{s.t.}\quad 0<p_u\leqslant P_u\quad f_{u,s}>0$$
$$\sum_{u\in U}f_{u,s}\leqslant f_s$$

由于功率分配 $p_u$ 和计算资源分配 $f_{u,s}$ 的目标和约束可以彼此分配,因此式(17)可以分解为两个独立的问题: 上行链路功率分配(UPA)和计算资源分配(CRA)。

上行链路功率分配问题可以表示为:

$$\min_{p}\sum_{s\in S}\sum_{u\in U_s}\frac{\phi_u+\varphi_u p_u}{\log_2(1+\gamma_{u,s})} \tag{18}$$
$$\text{s.t.}\quad 0<p_u\leqslant P_u$$

式中: $\gamma_{u,s}$ 是用户 $u$ 对应的上行链路的 SINR。由于式(18)是非凸的,求解非常困难,本文通过寻求 $I_{u,s}^{j}$ 的近似确定 $\gamma_{u,s}$,进而将式(18)分解成可以解决的子问题。

假设每个 BS 独立计算其上行链路功率分配,那么 $I_{u,s}^{j}$ 上界可估计为:

$$\hat{I}_{u,s}^{j}=\sum_{w\in S\setminus\{s\}}\sum_{k\in U_w}x_{k,s}^{j}P_k h_{k,s}^{j} \tag{19}$$

进而得到了用户 $u$ 上传到子带 $j$ 上 BS 的上行链路信噪比的近似值:

$$\hat{\gamma}_{u,s}^{j}=\frac{p_u h_{u,s}^{j}}{\hat{I}_{u,s}^{j}+\sigma^2} \tag{20}$$

通过式(20)可以将式(18)中目标函数和对应的用户发射功率约束条件实现分离,因此,式(18)中的目标函数可以近似为:

$$\min\sum_{u\in U_s}\Gamma_s(p_u) \tag{21}$$
$$\text{s.t.}\quad 0<p_u\leqslant P_u$$

式中: $\Gamma_s(p_u)=\dfrac{\phi_u+\varphi_u p_u}{\log_2(1+\vartheta_{u,s}p_u)}$; $\vartheta_{u,s}=\dfrac{\sum_{j\in N}h_{u,s}^{j}}{\sum_{w\in S\setminus\{s\}}\sum_{k\in U_w}x_{k,s}^{j}P_k h_{k,s}^{j}+\sigma^2}$。

式(21)可以采用拟凸性优化技术来解决。

计算资源分配问题可以表示为:

$$\min_{f}\sum_{s\in S}\sum_{u\in U_s}\frac{\eta_u}{f_{u,s}} \tag{22}$$
$$\text{s.t.}\quad \sum_{u\in U}f_{u,s}\leqslant f_s\quad f_{u,s}>0$$

由于目标函数的 Hessian 矩阵是正定的,而且约束条件是凸的,因此,式(22)问题是一个凸优化问题,可

以采用 Karush-Kuhn-Tucker 进行求解。

根据上面的讨论，对于给定的任务卸载决策$\chi$，根据式(14) －式(16)得到计算资源分配的解决方案:

$$J^*(\chi) = \sum_{s\in S}\sum_{u\in U_s}\lambda_u(\beta_u^t+\beta_u^e) - \Gamma(x,p^*) - \Lambda(x,f^*) \tag{23}$$

$$\Lambda(\chi,f^*) = \sum_{s\in S}\frac{1}{f_s}\Big(\sum_{u\in U}\sqrt{\eta_u}\Big)^2 \tag{24}$$

根据式(23)、式(24)，式(13)中的 TO 问题可以改写为:

$$\max_{x}\ \sum_{s\in S}\sum_{u\in U_s}\lambda_u(\beta_u^t+\beta_u^e) - \Gamma(\chi,p^*) - \Lambda(\chi,f^*) \tag{25}$$

$$\text{s.t.}\quad x_{u,s}^j\in\{0,1\}\qquad \sum_{s\in S}\sum_{j\in N}x_{u,s}^j\leqslant 1$$

$$\sum_{u\in U}x_{u,s}^j\leqslant 1$$

考虑到 TO 问题的组合性质，在多项式时间内求解最优解是一个非常具有挑战性的问题。解决式(25)的一个简单方法是对所有可能的任务卸载决策使用穷举搜索法。然而，由于候选任务卸载决策的总数是$2^n$，$n=S\times U\times N$，因此穷举搜索方法显然是不切实际的。

**算法 1　删除和交换操作**

remove($\chi$,$x_{u,s}^j$)

1. 设置$\chi\leftarrow\chi\backslash\{x_{u,s}^j\}$
2. 输出$\chi$

exchange($\chi$,$x_{u,s}^i$)

3. for $w\in S$,$i\in N$ do
4. $\quad\chi\leftarrow\chi\backslash\{x_{u,w}^i\}$
5. end for
6. for $v\in U$ do
7. $\quad\chi\leftarrow\chi\backslash\{x_{v,s}^j\}$
8. end for
9. 设置$\chi\leftarrow\chi\cup\{x_{u,s}^j\}$
10. 输出$\chi$

为了克服上述缺点，本文提出一种低复杂度的启发式算法，可以在多项式时间内找到式(19)的次优解。算法从一个空的集合$\chi=\varnothing$开始，如果它提高了设定值$J^*(x)$，就重复执行一个本地操作—删除或交换操作，在算法 1 中给出。当处理两个拟阵约束时，交换操作主要是添加一个当前集合的外部元素，并删除集合中至多 2 个元素从而符合约束。总之，提出的启发式任务卸载调度算法在算法 2 中给出。

**算法 2　启发式任务卸载调度**

1. 初始化: $\chi=\varnothing$
2. 寻找 $x_{k,w}^i=\arg\max_{j\in N,s\in S,u\in U}J^*(\{x_{u,s}^i\})$
3. 设置$\chi\leftarrow\{x_{k,w}^i\}$
4. if 存在 $x_{u,s}^i\in\chi$，满足

$$J^*(\text{remove}(\chi,x_{u,s}^i)) > \Big(1+\frac{1}{p(n,\varepsilon)}\Big)J^*(\chi)$$，那么

5. $\quad$设置$\chi\leftarrow$remove($\chi$,$x_{u,s}^i$)
6. $\quad$返回第 4 步
7. else if 存在 $x_{u,s}^i\in G\backslash\chi$，满足

$$J^*(\text{exchange}(\chi,x_{u,s}^i)) > \Big(1+\frac{1}{p(n,\varepsilon)}\Big)J^*(\chi)$$，那么

8. $\quad$设置$\chi\leftarrow$exchange($\chi$,$x_{u,s}^i$)
9. $\quad$返回第 4 步
10. end if
11. 输出$\chi$

# 3 实　验

## 3.1 实验设置

为了评估提出的启发式联合任务卸载调度和资源分配策略(hJTORA)的性能，本文通过仿真实验进行测试。所有实验均在配置为 CPU Intel Core i7-4700MQ-2.4 GHz@6 GB RAM 的机器上执行，通过 MATLAB R2017B 进行仿真。考虑一个由多个六角形子区构成的蜂窝网络系统，每个小区的中心都配有一个 BS，相邻基站相距 1 km。假设用户和 BS 分别使用单个天线进行上行链路传输和接收。本文的测试中用户的最大发射功率为$P_u=20$ dBm，系统带宽设置为$B=20$ MHz，背景噪声方差假设为$\sigma^2=-100$ dBm。上行链路信道增益是使用距离相关的路径损耗模型生成的，该模型的计算公式为:

$$L=140.7+36.7\lg d_{[\text{km}]} \tag{26}$$

式中: 对数正常阴影衰落的标准偏差为 8 dB。

在计算资源方面，假设每个 MEC 服务器和用户的 CPU 性能分别为$f_s=20$ GHz 和$f_u^l=1$ GHz。能量系数设置为$\kappa=5\times10^{-27}$。在计算任务方面，采用机场安全与监控的人脸检测与识别应用，选择任务请求大小为$d_u=420$ KB，其他参数为$\beta_u^t=0.2$，$\beta_u^e=0.8$，$\lambda_u=1$。此外，用户在网络覆盖区域内随机放置、分布均匀，子带数$N$设为每个小区的用户数。

为了验证提出算法的优越性，本文将与其他优秀算法如穷举搜索法、GOJRA(Greedy Offloading and Joint Resource Allocation)、IOJRA(Independent Offloading and Joint Resource Allocation)和 DORA(Distributed Offloading and Resource Allocation)等进行对比。

## 3.2 实验结果分析

### 3.2.1 不同算法平均系统效能对比

首先，将本文算法的结果与穷举法得到的最优解

以及其他三种算法进行了比较。由于穷举法搜索所有可能的卸载调度决策,因此对于大量变量,它的运行时间非常长。因此,测试时选择在一个小的网络: 设置用户数 $U=6$,单元数 $S=7$,每个单元有 $N=2$ 个子带。分别设置 $c_u=1\ 000$、$1\ 500$ 和 $2\ 000$ 兆周期时,随机生成 500 个阴影衰落,得到不同算法 95% 置信区间的平均系统效能结果如图 5 所示。可以看出,hJTORA 算法与最优穷举算法的性能非常接近,但明显优于其他算法,充分体现算法 2 得到的 hJTORA 解的次优性。同时,还可以发现,所有方案的性能都随着任务工作量的增加而提高。在所有情况下,hJTORA 的平均系统效能都在穷尽算法的 0.6% 以内,而与 DORA、GOJRA 和 IOJRA 方案相比,其平均增益分别为 33%、43% 和 96%。

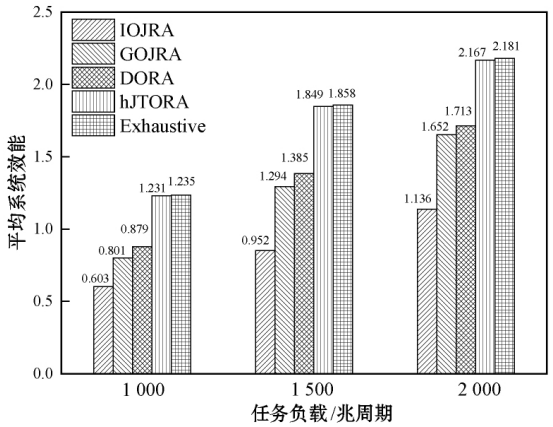
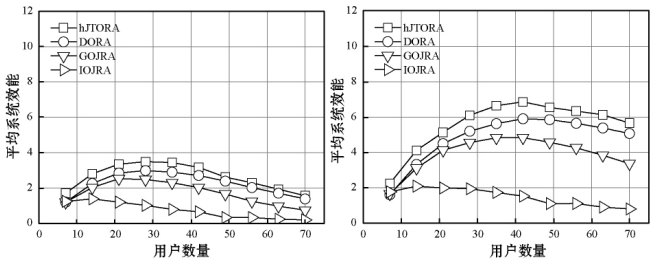


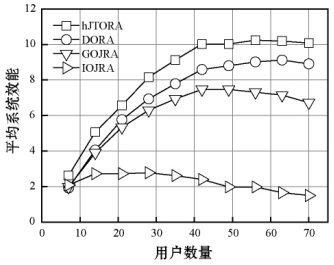

图 5　不同算法的平均系统效能比较

### 3.2.2　用户数量对平均系统效能的影响

图 6 给出了不同数量的用户在卸载任务时的平均系统效能。测试中每个单元的用户数从 1 到 10 不等,并在三个具有不同任务工作负载的场景中进行比较。而且子带数 $N$ 等于每个单元的用户数,因此当系统中有更多的用户时,为每个用户分配的带宽会减少。可以看出,hJTORA 算法性能表现最好,而且当任务的工作量增加时,方案的性能也会显著提高。当用户数量较少时,系统效能随着用户数量的增加而增加; 但是,当用户数量超过某些阈值时,系统效能开始降低。这是由于过多用户为卸载任务而竞争无线和计算资源,使得 MEC 服务器上发送任务和执行任务的开销升高,导致卸载效能降低。




(a) $c_u=1\ 000$ 兆周期　　(b) $c_u=1\ 500$ 兆周期

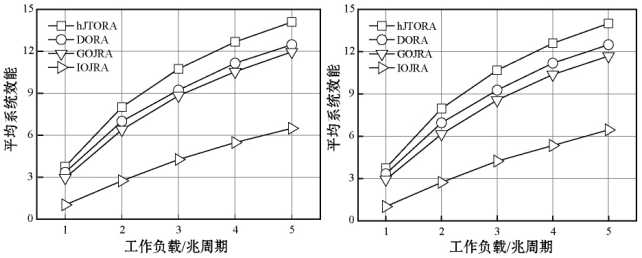



(c) $c_u=2\ 000$ 兆周期

图 6　不同用户数量时的平均系统效能比较

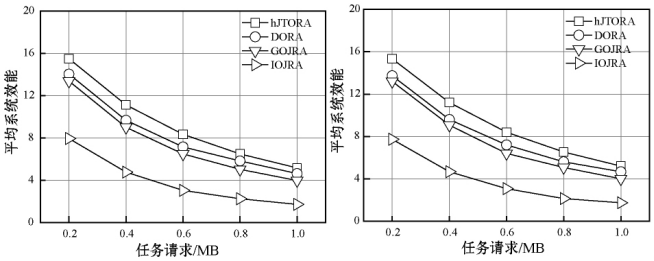
### 3.2.3　任务配置文件对平均系统效能的影响

本文中的任务配置文件是指请求数据量和工作负载两个方面。根据请求数据量 $d_u$ 和工作负载 $c_u$ 来评估系统效能。考虑两种 MEC 服务器的配置: (1) 同构服务器—所有服务器的 CPU 速度是 20 GHz; (2) 异构服务器—服务器的 CPU 速度随机从{10,20,30} GHz 中选择。四个算法在不同 $c_u$、$d_u$ 值时的平均系统效能如图 7、图 8 所示。可以看出,所有方案的平均系统效能都随任务工作量的增加而增加,随任务请求的增加而减少。这表示,与那些具有大请求量和低工作负载的任务相比,具有小请求量和高工作负载的任务从卸载中获益更多。同构服务器设置中所有方案的性能与异类服务器设置中所有方案的性能之间的差异是微乎其微的。除此之外,在所有方案中,hJTORA 方案的性能是最优的。




(a) 同构服务器　　(b) 异构服务器

图 7　不同工作负载下的平均系统效能比较($U=28$,$d_u=420$)




(a) 同构服务器　　(b) 异构服务器

图 8　不同任务请求的平均系统效能比较($U=28$,$c_u=3\ 000$)

## 4　结　语

本文目标是在多 MEC 服务器辅助网络中设计一

个整体的解决方案，对联合任务卸载和资源分配进行优化，从而最大限度地提高用户的卸载收益。首先，为每个用户的卸载效用建模，将 JTORA 问题转化为 MINLP 问题。然后，采用 Tammer 分解方法将高复杂度的原始问题转化为等效的主问题和一组复杂度较低的子问题。最后，利用本文提出的低复杂度启发式算法，以次优解的方式解决 JTORA 问题，实现共同优化任务卸载决策、用户上行链路传输功率的目标。仿真结果表明，本文提出的优化策略的平均系统效能明显优于其他方案，提出的启发式算法能够很好地实现最优解，显著提高系统的平均卸载效率。

## 参 考 文 献


[1] 于博文，蒲凌君，谢玉婷，等. 移动边缘计算任务卸载和基站关联协同决策问题研究[J]. 计算机研究与发展，2018，55(3)：537－550.

[2] Li S，Zhai D，Du P，et al. Energy-efficient task offloading，load balancing，and resource allocation in mobile edge computing enabled IoT networks[J]. Science China Information Sciences，2019，62(2)：29307－29309.

[3] Tran T X，Hajisami A，Pandey P，et al. Collaborative mobile edge computing in 5G networks：new paradigms，scenarios，and challenges[J]. IEEE Communications Magazine，2017，55(4)：54－61.

[4] Roman R，Lopez J，Mambo M. Mobile edge computing，fog et al.：a survey and analysis of security threats and challenges[J]. Future Generation Computer Systems，2018，78：680－698.

[5] Tran T X，Hosseini M P，Pompili D. Mobile edge computing：Recent efforts and five key research directions[J]. IEEE COMSOC MMTC Communications-Frontiers，2017，12(4)：29－33.

[6] Chen M，Hao Y. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications，2018，36(3)：587－597.

[7] Mao Y，Zhang J，Letaief 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.

[8] You C，Huang K. Multiuser resource allocation for mobile-edge computation offloading[C]//2016 IEEE Global Communications Conference(GLOBECOM)，2016.

[9] 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.

[10] 徐佳，李学俊，丁瑞苗，等. 移动边缘计算中能耗优化的多重资源计算卸载策略[J]. 计算机集成制造系统，2019，25(4)：954－961.

[11] Chen X，Pu L，Gao L，et al. Exploiting massive d2d collaboration for energy-efficient mobile edge computing[J]. IEEE Wireless Communications，2017，24(4)：64－71.


(上接第 70 页)


[8] Salinas D，Flunkert V，Gasthaus J. DeepAR：probabilistic forecasting with autoregressive recurrent networks[J]. International Journal of Forecasting，2019，36(3)：1181－1191.

[9] Vaswani A，Shazeer N，Parmar N，et al. Attention is all you need[C]//The Thirtieth Conference on Neural Information Processing Systems，2017.

[10] 徐超，项薇，季孟忠，等. 基于 ARIMA 与自适应过滤法的组合预测模型研究[J]. 计算机应用与软件，2018，35(11)：296－300，320.

[11] 沈旭东. 基于深度学习的时间序列算法综述[J]. 计算机应用技术，2019(1)：71－76.

[12] 吴双双. 基于深度神经网络的时间序列预测技术研究[D]. 南京：南京理工大学，2017.

[13] 权钲杰. 基于深度神经网络集成和信息论学习的时间序列预测[D]. 杭州：浙江大学，2019.

[14] 刘峰，瞿俊. 基于聚类分析和神经网络的时间序列预测方法[J]. 微电子学与计算机，2006，23(9)：85－87.

[15] 王慧健，刘峥，李云，等. 基于神经网络语言模型的时间序列趋势预测方法[J]. 计算机工程，2019(7)：13－19.

[16] 李洁，林永峰. 基于多时间尺度 RNN 的时序数据预测[J]. 计算机应用与软件，2018，35(7)：33－37，62.

[17] 蒋倩仪. 基于时间序列预测的股票交易决策建议系统[J]. 计算机应用与软件，2017，34(4)：75－81，104.

[18] Alexandrov A，Benidis K，Bohlkeschneider M，et al. GluonTS：probabilistic time series models in python[EB]. arXiv：1906.05264，2019.

[19] Li S，Jin X，Xuan Y，et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting[C]//The Thirty-second Conference on Neural Information Processing Systems，2019.

[20] Ziegel E R，Box G E P，Jenkins G M. Time series analysis，forecasting，and control[J]. Journal of Time，1976，31(2)：238－242.

[21] Oord A V D，Dieleman S，Zen H，et al. WaveNet：a generative model for raw audio[EB]. arXiv：1609.03499，2016.

[22] Gneiting T，Raftery A E. Strictly proper scoring rules，prediction，and estimation[J]. Journal of the American Statistical Association，2007，102(477)：359－378.