

NOMA-MEC系统中基于改进遗传算法的协作式计算卸载与资源管理

周天清* 胡海琴 曾新亮
(华东交通大学信息工程学院 南昌 330013)

摘要: 为平衡网络负载与充分利用网络资源, 针对超密集异构的多用户和多任务边缘计算网络, 在用户时延约束下, 该文构造了协作式计算任务卸载与无线资源管理的联合优化问题以最小化系统能耗。问题建模时, 为应对基站超密集部署导致的严重干扰问题, 该文采用了频带划分机制, 并引入了非正交多址技术(NOMA)以提升上行频谱利用率。鉴于该目标优化问题具备非线性混合整数的形式, 根据多样性引导变异的自适应遗传算法(AGADGM), 设计出了协作式计算卸载与资源分配算法。仿真结果表明, 在严格满足时延约束条件下, 该算法能获得比其他算法更低的系统能耗。

关键词: 超密集异构网; 边缘计算; 协作式任务卸载; 频谱划分; 非正交多址技术; 自适应遗传算法

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2022)09-3014-10

DOI: 10.11999/JEIT220306

Cooperative Computation Offloading and Resource Management Based on Improved Genetic Algorithm in NOMA-MEC Systems

ZHOU Tianqing HU Haiqin ZENG Xinliang

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: To balance the network loads and utilize fully the network resources, joint cooperative computation offloading and wireless resource management is considered for ultra-dense heterogeneous edge computing networks with multiple users and multiple tasks, which minimizes the system energy consumption under the constraints of users' delay. During the problem modeling, a frequency spectrum partitioning mechanism is introduced to tackle serious network interference caused by ultra-dense deployment of base stations, and Non-Orthogonal Multiple Access (NOMA) technology is introduced to improve the uplink frequency spectrum efficiency. Considering that the optimization problem is a nonlinear mixed-integer form, according to Adaptive Genetic Algorithm with Diversity-Guided Mutation (AGADGM), an effective algorithm used for cooperative computation offloading and resource allocation is designed. The simulation results show that proposed algorithm could achieve lower system energy consumption than other existing algorithms under strict constraints of users' delay.

Key words: Ultra-Dense Heterogeneous Networks (UDHN); Edge computing; Cooperative computation offloading; Frequency spectrum partitioning; Non-Orthogonal Multiple Access (NOMA); Adaptive Genetic Algorithm (AGA)

1 引言

随着机器学习和人工智能等科学技术的兴起, 人脸识别、智能家居、自动驾驶和医疗保健等应用

应运而生。这些新型应用在实时处理时占用了大量的计算和存储资源, 对存储空间、计算资源和电池容量受限的智能设备提出了巨大的挑战^[1-3]。为应对该挑战, 移动边缘计算(Mobile Edge Computing, MEC)允许资源受限的智能设备将计算密集型任务卸载至边缘网络, 让具备计算和存储能力的边缘节点处理任务。

超密集异构网络(Ultra-Dense Heterogeneous Network, UDHN)与MEC的结合进一步缩短了用户与计算中心的距离, 且UDHN通过在宏小区内部署大量小基站(Small Base Station, SBS), 极大地提

收稿日期: 2022-03-22; 改回日期: 2022-08-08; 网络出版: 2022-08-12

*通信作者: 周天清 zhoutian930@163.com

基金项目: 国家自然科学基金(61861017, 61861018, 61961020, 62171119), 国家重点研究开发计划(2020YFB1807201)

Foundation Items: The National Natural Science Foundation of China (61861017, 61861018, 61961020, 62171119), The National Key Research and Development Program of China (2020YFB1807201)

升了服务覆盖率和资源利用率,降低了智能设备至基站的传输时延。但是,密集部署的小基站会导致大量的能耗^[4,5]。此外,在UDHN中,每个基站覆盖下的用户密度较大,区内与区间均存在严重干扰^[6]。

合适的干扰消除和频带资源分配方案有利于降低MEC系统时延和能耗,从而有效提升系统性能。在单MEC服务器和多用户场景下,文献[7]提出了基于正交频分多址接入的MEC系统。具体而言,它将系统频带划分为多个正交子信道以消除用户间的干扰,并最大化了系统计算效率。在多小区和多用户的MEC场景中,文献[8]最小化时延和能耗的加权。其中,小区内的用户通过正交信道进行信号传输,但小区间的用户可能相互干扰。在多宏小区、多小小区和多用户的UDHN场景下,文献[9]研究了系统总能耗最小优化问题。它为消除UDHN中层间和层内干扰,提出了一种频谱资源分配方案。但是,随着用户规模的增大,频谱资源分配会越来越紧张。

为进一步提升频谱利用率、实现高速率传输及广覆盖,已有多数研究将非正交多址接入(Non-Orthogonal Multiple Access, NOMA)和MEC结合起来。文献[10]研究了时分多址接入(Time Division Multiple Access, TDMA)和NOMA下的部分卸载与二元卸载问题,并尝试最大化系统计算效率。文献[11]考虑了NOMA-MEC在车联网领域内的应用。为最大限度保证用户效益,它利用基于深度学习网络的博弈算法来为用户选择最优的卸载策略。文献[12]构建了UDHN下的NOMA-MEC系统,并研究了多SBS、多用户的资源分配问题以最小化用户能耗和任务时延。

上述研究都有对干扰进行消除和对频带资源进行分配,或者引入NOMA技术进一步提升频带利用率。然而,其中大部分研究考虑每个用户仅有一个任务需要处理的情况,单个宏小区场景与非协作场景;很少研究考虑UDHN场景下多用户多任务协作式计算卸载问题。不同于上述研究,本文构建了UDHN场景下NOMA-MEC系统,其中每个用户均存在多个计算任务,且宏基站(Macro Base Station, MBS)与小基站间存在协作式卸载。针对该系统,在MEC计算资源和用户时延约束下,联合优化计算资源分配、用户任务卸载和用户发射功率以最小化系统总能耗。本文具体工作如下:

(1)在UDHN中融入频带划分机制与NOMA技术。为解决UDHN中存在的层内与层间干扰,引入了频谱划分机制。在此基础上,为进一步提升上行频谱效益,又引入NOMA技术。

(2)在NOMA-MEC系统中,考虑多蜂窝与多用户。同时,假定每个用户有多个不可拆分的计算任务,且小基站可与宏基站协作完成任务。其次,根据计算任务所需计算资源占比,分配用户计算资源。这种计算资源分配方式既能很好地满足用户计算需求,又能简化优化问题,便于算法实际应用。

(3)鉴于所建模问题具备非线性整数形式,利用多样性引导变异的自适应遗传算法(Adaptive Genetic Algorithm with Diversity-Guided Mutation, AGADGM)设计出了协作式计算卸载与资源分配算法。不同于一些组合算法^[9,13],本文仅考虑单纯遗传算法(Genetic Algorithm, GA),仅涉及单层循环。显然,此种考虑可以削减算法的运行时间,且更适合超密集网络场景。最后,在仿真中,引入全本地算法、最强信道增益关联算法和传统遗传算法进行对比,调查了用户发射功率、用户密度与频谱划分因子对系统性能的影响。

2 系统模型

2.1 网络模型

本文考虑了如图1(a)所示的UDHN场景下的NOMA-MEC系统。图中每个正6边形代表一个宏小区,每个宏小区内部署了1个MBS、多个SBS与多个用户终端。具体而言,在超密集异构MEC系统中,部署了 \tilde{N} 个MBS, \tilde{N} 个SBS和 K 个用户终端,其中SBS的索引集记为 $\tilde{\mathcal{N}} = \{1, 2, \dots, \tilde{N}\}$, MBS的索引集记为 $\tilde{\mathcal{N}} = \{\tilde{N} + 1, \tilde{N} + 2, \dots, \tilde{N} + M\}$, 则所有基站的索引集为 $\mathcal{N} = \tilde{\mathcal{N}} \cup \tilde{\mathcal{N}}$; 用户终端集合表示为 $\mathcal{K} = \{1, 2, \dots, K\}$ 。此外,假定每个宏小区内所有SBS构成一个SBS簇,且所有基站均配置了MEC服务器;假定每个用户终端有多个计算任务,其任务可以在本地执行,也可卸载至基站MEC服务器执行;假定用户与基站之间的通信采用无线链路, MBS与SBS之间的通信采用有线回程链路(速率为 r_0)。

2.2 资源模型

为消除UDHN中层内干扰与层间干扰,根据文献[9,14],本文亦对频谱进行分割,如图1(b)所示。在UDHN中,为消除MBS与SBS间的层间干扰,将频带 F (带宽为 W)划分为子带 F_1 和子带 F_2 ,其中子带 F_1 (带宽为 μW)分配给MBS,子带 F_2 (带宽为 $(1 - \mu)W$)分配给SBS, $0 \leq \mu \leq 1$ 为频谱划分因子;为消除相邻MBS间的干扰和SBS簇间的干扰,子带 F_1 被分割为带宽为 $\mu W/3$ 的3个子频带 F_{11} , F_{12} 与 F_{13} ,且 F_2 被分割为带宽为 $(1 - \mu)W/3$ 的3个子频带 F_{21} , F_{22} 与 F_{23} 。为消除簇内SBS间的干扰,子频带

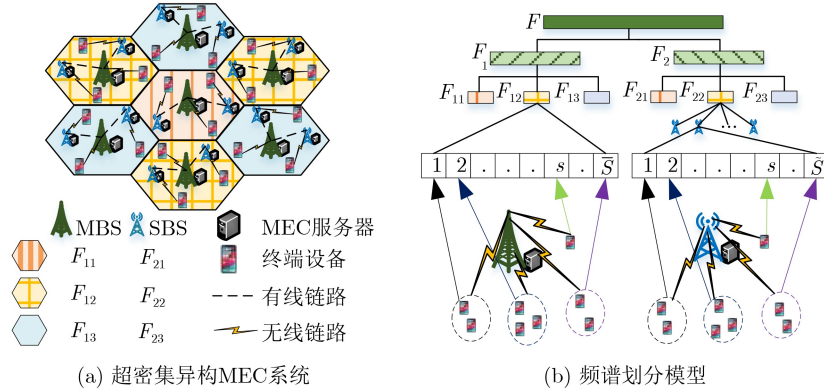


图1 系统模型

F_{21} , F_{22} 与 F_{23} 被再次均分为多个更小的子频带, 它们的带宽为 $(1-\mu)W/3\theta$, 其中 θ 为一个SBS簇内SBS的数目。

区别于文献[9]中MBS和SBS将频带资源均分给其所关联的所有用户, 本文在上行链路中引入NOMA技术以解码传输信号, 并借此提升频谱利用效率。如图1(b)所示, 在NOMA技术下, 将任意MBS和SBS的频带资源划分为多个NOMA子信道, 允许多个用户同时使用某个基站的不同子信道。假定每个子信道带宽为 w , 则任意MBS的子信道个数为 $\bar{S} = \mu W/3w$, 其子信道索引集记为 $\bar{S} = \{1, 2, \dots, \bar{S}\}$ 。同理, 任意SBS的子信道个数为 $\tilde{S} = (1-\mu)W/3wU$, 其子信道索引集记为 $\tilde{S} = \{1, 2, \dots, \tilde{S}\}$ 。

2.3 通信模型

在上述资源模型中, 本文考虑了上行NOMA技术。该技术按信道增益递减顺序进行信号解码, 这与下行NOMA的解码顺序相反[15]。假设有多个用户接入基站 n 的子信道 s , \mathcal{K}_{ns} 为关联至基站 n 的子信道 s 的用户集合。类似于文献[16], 对于任意基站的子信道, 将用户至该子信道的信道增益按升序排列, o_{nsk} 表示用户 k 在基站 n 的子信道 s 上的排序序号。当用户 j 和用户 k 同时接入基站 n 的子信道 s , 且用户 j 的信道增益 h_{nsj} 低于用户 k 的信道增益 h_{nsk} 时, $o_{nsj} < o_{nsk}$ 。那么, 解码用户 k 的信号时, 用户 j 的信号将被视为噪声。

根据上行NOMA的特性[15,16], 用户关联基站子信道时, 其上传速率为以下两种情况:

(1)当用户终端 k 选择关联MBS n 的子信道 s 时, 用户 k 的上传速率 r_{nsk} 为

$$r_{nsk} = w \log_2(1 + p_k h_{nsk} / (\sum_{j \in \mathcal{K}_{ns}: o_{nsj} < o_{nsk}} p_j h_{nsj} + \sigma^2)), \quad \forall n \in \bar{\mathcal{N}}, \forall s \in \bar{\mathcal{S}} \quad (1)$$

其中, p_k 为用户 k 的发射功率, σ^2 为噪声功率。

(2)当用户终端 k 选择关联SBS n 的子信道 s 时, 用户 k 的上传速率 r_{nsk} 为

$$r_{nsk} = w \log_2(1 + p_k h_{nsk} / (\sum_{j \in \mathcal{K}_{ns}: o_{nsj} < o_{nsk}} p_j h_{nsj} + \sigma^2)), \quad \forall n \in \tilde{\mathcal{N}}, \forall s \in \tilde{\mathcal{S}} \quad (2)$$

2.4 计算模型

假设任意用户 k 有 M_k 个计算任务, 其任意任务记为 $\{d_k, c_k, T_k^{\max}\}$ 。 d_k 为用户 k 的任意计算任务输入数据的大小(bit), c_k 为用户 k 的任意计算任务执行1 bit数据时所需CPU周期数(cycles/bit), T_k^{\max} 为用户 k 的任意计算任务的截止时延(s)。在对用户任务进行处理时, 考虑到基站MEC服务器的计算资源有限, 根据文献[5,17], 本文亦采用了协作式任务卸载策略。

具体而言, 当用户 k 接入MBS或者SBS时, y_k 个任务卸载至MBS或者SBS的MEC服务器, 剩余 $M_k - y_k$ 个任务于本地计算。但是, 当用户接入SBS且该基站的MEC服务器的计算资源不足时, SBS需通过有线链路将 z_k 个任务卸载至MBS的MEC服务器上。因此, 具体时延和计算能耗模型分为如下3部分。

(1)本地计算。用户 k 的 $M_k - y_k$ 个任务在本地计算的时延 T_k^{LOC} 和能耗 E_k^{LOC} 分别为

$$T_k^{\text{LOC}} = (M_k - y_k) c_k d_k / F_k^{\text{UE}} \quad (3)$$

$$E_k^{\text{LOC}} = \alpha_k (F_k^{\text{UE}})^3 T_k^{\text{loc}} = \alpha_k c_k d_k (M_k - y_k) (F_k^{\text{UE}})^2 \quad (4)$$

其中, F_k^{UE} 为用户 k 的计算能力(cycles/s), α_k 为用户 k 的每个CPU周期消耗能量的系数。

(2)卸载至MBS端。当用户 k 关联至MBS n 的子信道 s 时, y_k 个任务被卸载至该基站的MEC服务器上。考虑到计算结果的数据量非常小, 其传输时延与能耗可以忽略不计, 本文仅考虑任务上传至MBS所产生的时延和能耗, 及在MEC服务器上执行时

产生的时延和能耗。于是类似于文献[5,17], 用户 k 的任务卸载至MBS并完成计算的总时延 T_k^{MBS} 与消耗的总能量 E_k^{MBS} 分别为

$$T_k^{\text{MBS}} = \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (1/r_{nsk} + c_k/F_{nk}) y_k d_k \quad (5)$$

$$E_k^{\text{MBS}} = \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (p_k y_k d_k / r_{nsk} + \bar{\varepsilon}_n y_k c_k d_k) \quad (6)$$

其中, 时延 T_k^{MBS} 包含任务卸载至MBS的上传时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} y_k d_k / r_{nsk}$ 与任务在MEC服务器上的计算时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} y_k c_k d_k / F_{nk}$; 能耗 E_k^{MBS} 包含任务卸载至MBS的上传能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} p_k y_k d_k / r_{nsk}$ 与任务在MEC服务器上的计算能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} \bar{\varepsilon}_n y_k c_k d_k$; x_{nsk} 为关联指示变量, $x_{nsk} = 1$ 表示用户 k 关联至基站 n 的子信道 s ; $\bar{\varepsilon}_n$ 为MBS n 每个CPU周期消耗能量的系数。 F_{nk} 为MBS n 按任务比例分配给用户 k 的计算资源(能力), 记为

$$F_{nk} = \left(\sum_{s \in \mathcal{S}} x_{nsk} y_k c_k d_k + \sum_{m \in \mathcal{N}_n} \sum_{s \in \mathcal{S}} x_{msk} z_k c_k d_k \right) F_n^{\text{MBS}} / \left(\sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} x_{nsk} y_k c_k d_k + \sum_{m \in \mathcal{N}_n} \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} x_{msk} z_k c_k d_k \right) \quad (7)$$

其中, F_n^{MBS} 为MBS n 的计算能力(cycles/s); $\tilde{\mathcal{N}}_n$ 为关联至MBS n 的SBS集合。

(3) 卸载至SBS端。当用户 k 关联至SBS n 的子信道 s 时, y_k 个任务被卸载至该基站的MEC服务器上, y_k 个任务中又有 z_k 个任务被SBS通过有线链路传输到MBS的MEC服务器上。在此过程中, 忽视结果回传时延与能耗。类似于文献[5,17], 用户 k 的总时延 T_k^{SBS} 可表示为

$$T_k^{\text{SBS}} = \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (y_k d_k / r_{nsk} + (y_k - z_k) c_k d_k / F_{nk} + z_k d_k / r_0 + z_k c_k d_k / F_{m_n k}) \quad (8)$$

其中, 时延 T_k^{SBS} 包含4部分, 即 y_k 个任务上传至SBS的时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} y_k d_k / r_{nsk}$, $y_k - z_k$ 个任务在SBS的MEC服务器上的计算时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (y_k - z_k) c_k d_k / F_{nk}$, z_k 个任务通过有线链路上传至MBS的时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} z_k d_k / r_0$ 与

z_k 个任务在MBS的MEC服务器上的计算时延 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} z_k c_k d_k / F_{m_n k}$; m_n 为SBS n 关联的MBS; $F_{m_n k}$ 为MBS分配给通过SBS间接关联至MBS的用户的计算资源; F_{nk} 为SBS n 按比例分配给用户 k 的计算资源(cycles/s), 可表示为

$$F_{nk} = (y_k - z_k) c_k d_k F_n^{\text{SBS}} / \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} x_{nsk} (y_k - z_k) c_k d_k \quad (9)$$

其中, F_n^{SBS} 为SBS n 的计算能力。

忽视回传能耗, 用户 k 关联至SBS时产生的总能耗^[5,17] E_k^{SBS} 可表示为

$$E_k^{\text{SBS}} = \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (y_k p_k d_k / r_{nsk} + (y_k - z_k) c_k d_k \bar{\varepsilon}_n + \xi z_k d_k / r_0 + z_k c_k d_k \bar{\varepsilon}_{m_n}) \quad (10)$$

其中, 能耗 E_k^{SBS} 包含4部分, 即 y_k 个任务上传至SBS的能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} y_k p_k d_k / r_{nsk}$, $y_k - z_k$ 个任务在SBS的MEC服务器上的计算能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} (y_k - z_k) c_k d_k \bar{\varepsilon}_n$, z_k 个任务通过有线链路上传至MBS的能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} \xi z_k d_k / r_0$ 与 z_k 个任务在MBS的MEC服务器上的计算能耗 $\sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{nsk} z_k c_k d_k \bar{\varepsilon}_{m_n}$; $\bar{\varepsilon}_n$ 为SBS n 每个CPU周期消耗能量的系数; ξ 为SBS与MBS之间有线链路的功耗; $\bar{\varepsilon}_{m_n}$ 为SBS n 关联的MBS的每个CPU周期消耗能量的系数。

2.5 问题描述

为符合实际应用, 假设用户任务的本地执行与卸载至基站MEC服务器上的执行同时发生, 即考虑任务的本地与远程并行执行。于是, 用户 k 的时延 T_k 和能耗 E_k 可分别表示为

$$T_k = \max\{T_k^{\text{LOC}}, T_k^{\text{MBS}} + T_k^{\text{SBS}}\} \quad (11)$$

$$E_k = E_k^{\text{LOC}} + E_k^{\text{MBS}} + E_k^{\text{SBS}} \quad (12)$$

定义 E 为完成系统所有用户计算任务的总能耗, 并记为

$$E = \sum_{k \in \mathcal{K}} E_k \quad (13)$$

鉴于UDHN中部署了大量SBS, 消耗了大量能耗, 本文尝试最优化任务完成总能耗, 它涉及用户端与基站端的能耗。具体而言, 在计算资源和时延约束下, 联合优化基站子信道选择 $X = \{x_{nsk}, \forall n \in \mathcal{N}, \forall s \in \mathcal{S}_n, \forall k \in \mathcal{K}\}$ 、任务卸载决策 $y = \{y_k, \forall k \in \mathcal{K}\}$ 和 $z = \{z_k, \forall k \in \mathcal{K}\}$ 及用户发射功率 $p = \{p_k, \forall k \in \mathcal{K}\}$ 以最小化系统总能耗 E 。其中, \mathcal{S}_n 为基站 n 的子信道索引集。优化问题可规划为

$$\begin{aligned}
\text{P1: } \min_{\mathbf{X}, \mathbf{y}, \mathbf{z}, \mathbf{p}} E(\mathbf{X}, \mathbf{y}, \mathbf{z}, \mathbf{p}) &= \sum_{k \in \mathcal{K}} E_k \\
\text{s.t. C1: } 0 &\leq T_k \leq T_k^{\max}, \forall k \in \mathcal{K} \\
\text{C2: } 0 &\leq p_k \leq p_k^{\max}, \forall k \in \mathcal{K} \\
\text{C3: } 0 &\leq \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}_n} x_{n s k} y_k \leq M_k, \forall k \in \mathcal{K} \\
\text{C4: } 0 &\leq \sum_{n \in \tilde{\mathcal{N}}} \sum_{s \in \tilde{\mathcal{S}}} x_{n s k} z_k \\
&\leq \sum_{n \in \mathcal{N}} \sum_{s \in \mathcal{S}} x_{n s k} y_k \leq M_k, \forall k \in \mathcal{K} \\
\text{C5: } \sum_{s \in \mathcal{S}_n} x_{n s k} &= 1, \forall n \in \mathcal{N}, \forall k \in \mathcal{K} \\
\text{C6: } y_k, z_k &\in \{0, 1, \dots, M_k\}, \forall k \in \mathcal{K}
\end{aligned} \quad (14)$$

其中, 约束C1与C2分别给出了用户 k 的任务完成时延 T_k 及发射功率 p_k 的上下界; C3表示用户 k 关联基站时, 卸载至基站的任务数不能超过该用户的任务数; C4表示SBS将任务卸载至与其关联的MBS的任务数不能超过用户卸载至SBS的任务数; C5表示任意用户最多接入基站的一个子信道; C6表示卸载至SBS与MBS的任务数均为整数, 其范围在0至该用户任务数之间。

3 基于改进遗传算法的协作式计算卸载与资源分配算法

不难发现, 问题P1为0-1非线性混合整数规划问题。为解决该问题, 可以采用文献[9]中的联合自适应遗传算法和自适应粒子群算法的双层搜索算法。但是, 考虑到其中粒子群算法是对遗传算法所获解局部的刷新, 对问题解的优化不存在较大的影响, 本文仅尝试利用其第1层搜索算法(多样性引导变异的自适应遗传算法)求解问题P1。最后, 根据该改进遗传算法, 本文设计出了有效的协作式计算卸载与资源分配算法。

多样性引导变异的自适应遗传算法(AGADGM)是对传统遗传算法(Traditional GA, TGA)的改进。在TGA算法中, 1个种群包含多个个体。这些个体是带有染色体特征的实体, 它们是优化问题的潜在解。待种群初始化后, 根据适者生存和优胜劣汰的原则, TGA用适应度函数评价种群个体适应度并筛选出最佳个体。具体而言, 在一些特定规则下, TGA执行选择、交叉、变异等操作, 经过逐代演化, 最终找到问题的近似最优解。相比于TGA, AGADGM在自适应交叉和变异运算之前进行了多样性引导变异, 克服了早熟收敛问题[9]。该算法重要内容及相关操作如下。

3.1 染色体

类似于文献[13], 本文考虑实数编码。对于问

题P1的任意一种解的优化参量 $\{\mathbf{X}, \mathbf{y}, \mathbf{z}, \mathbf{p}\}$ 可以编码成某个个体 $i \in \mathcal{I}$ 的染色体 $\{\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i\}$, $\mathcal{I} = \{1, 2, \dots, I\}$ 记为个体索引集合, I 为种群的规模, $\mathbf{L}_i = \{l_{ik}, k \in \mathcal{K}\}$, $\mathbf{Q}_i = \{q_{ik}, k \in \mathcal{K}\}$, $\mathbf{U}_i = \{u_{ik}, k \in \mathcal{K}\}$ 与 $\mathbf{V}_i = \{v_{ik}, k \in \mathcal{K}\}$ 为个体 i 的染色体片段。为实现对参量 \mathbf{X} 的编码, 假定每个基站的子信道构成一个虚拟基站, 并对其标记唯一索引。因为任意用户仅能关联至某个基站的某一个子信道, 因此用户关联结果可用虚拟基站索引表示。于是, 在任意个体 i 中, l_{ik} 取值为用户 k 所关联虚拟基站的索引。此外, q_{ik} 和 u_{ik} 分别为用户 k 卸载至基站的任务数和自SBS上传至MBS的任务数, 且 v_{ik} 为用户 k 的发射功率。值得注意的是, q_{ik} 不能超过用户 k 的任务数, u_{ik} 不能超过 q_{ik} 并满足 q_{ik} 的限制条件, 且用户 k 的发射功率需满足 $0 \leq v_{ik} \leq p_k^{\max}$ 。

3.2 种群初始化

遗传算法从初始化种群开始, 需要生成满足条件的系列个体。为满足问题P1的约束C2-C6, 按照以下规则初始化种群, 即

$$\left. \begin{aligned}
l_{ik}^0 &= \text{randi}(\mathcal{B}), q_{ik}^0 = \text{randi}(\{0, 1, \dots, M_k\}) \\
u_{ik}^0 &= \text{randi}(\{0, 1, \dots, q_{ik}^0\}), v_{ik}^0 = \text{rand}(p_k^{\max}) \\
i &\in \mathcal{I}, k \in \mathcal{K}
\end{aligned} \right\} (15)$$

其中, $\mathcal{B} = \{1, 2, \dots, B\}$ 为 B 个虚拟基站的索引集, 即SBS和MBS所有子信道的索引集合; $\text{randi}(\psi)$ 表示从集合 ψ 里面随机输出一个元素; $\text{rand}(a)$ 为随机生成一个0至 a 之间的数。

3.3 适应度函数

在遗传算法中, 适应度函数主要用于度量个体对于环境的适应程度, 适应度高的个体具有较高的概率把基因遗传给下一代。在问题P1中, 优化问题参量的最终目的是最小化系统总能耗。鉴于此, 将其负目标函数作为适应度函数。考虑到约束C1具备非线性混合整数的形式, 在遗传操作中难以满足。同时, 为防止个体落入某个不可行区域, 将约束项C1作为罚函数项引入适应度函数中。于是, 个体 i 的适应度函数 $G(\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i)$ 定义为

$$\begin{aligned}
G(\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i) &= -E(\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i) \\
&\quad - H(\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i)
\end{aligned} \quad (16)$$

其中, $H(\mathbf{L}_i, \mathbf{Q}_i, \mathbf{U}_i, \mathbf{V}_i) = \sum_{k \in \mathcal{K}} \eta_k \max(T_k - T_k^{\max}, 0)$ 为个体 i 的惩罚函数, η_k 为该个体内用户 k 的惩罚因子。

3.4 选择

为从当前种群中选择优良个体, 本文采用锦标赛方式进行选择。在此方式中, 高适应度值的个体被选择的概率较大。此外, 在迭代更新过程中, 每一代都保留历史最佳个体。当历史最佳个体没有选

入下一代时,用历史最佳个体替换当前种群中的最差个体,从而提高遗传算法性能。

3.5 交叉

在遗传算法中,交叉操作的作用是为优良个体提供更多的交叉机会^[18]。在本文中,为保持种群多样性与加快算法收敛速度,自适应交叉策略中的交叉概率伴随进化过程不断自适应调整,而非传统遗传算法那样固定交叉概率。具体而言,依次选择两个相邻父系个体 i 和 $\bar{i} = i + 1$,并依据自适应交叉概率 $pc_{i\bar{i}}$ 交换相应染色体片段,从而形成两个新的个体(即子代)。其中,自适应交叉概率 $pc_{i\bar{i}}$ 记为

$$pc_{i\bar{i}} = \left. \begin{aligned} & b_1(\bar{G}_{i\bar{i}} - G^{\min}) / (G^{\text{ave}} - G^{\min}) \\ & \bar{G}_{i\bar{i}} < G^{\text{ave}}; pc_{i\bar{i}} = b_2, \bar{G}_{i\bar{i}} \geq G^{\text{ave}} \end{aligned} \right\} \quad (17)$$

其中, $0 < b_1 \leq 1$, $0 < b_2 \leq 1$ 为参数; $\bar{G}_{i\bar{i}}$ 为个体 i 和 \bar{i} 中较小的适应度值; G^{ave} 和 G^{\min} 分别为种群的平均适应度值和最小适应度值。

3.6 变异

为增强遗传算法的局部搜索能力,同时保持种群多样性,变异操作依据变异概率改变染色体基因,从而形成新的个体。为进一步保障算法搜索的全面性和精确性,跳出局部最优,自适应变异策略被广为推崇。在该策略中,变异概率随着种群进化特征改变。具体而言,随机选取一个个体 i ,再随机选取染色体的一个基因片段,在问题P1的约束条件下,依据变异概率 pm_i 对个体 i 进行变异操作。其中,变异概率 pm_i ^[18]为

$$pm_i = \left. \begin{aligned} & b_3(G^{\max} - \tilde{G}_i) / (G^{\max} - G^{\text{ave}}) \\ & \tilde{G}_i \geq G^{\text{ave}}; pm_i = b_4, \tilde{G}_i < G^{\text{ave}} \end{aligned} \right\} \quad (18)$$

其中, $0 < b_3 \leq 1$, $0 < b_4 \leq 1$ 为参数; G^{ave} 和 G^{\max} 分别为种群的平均适应度值和最大适应度值; \tilde{G}_i 为个体 i 的适应度值。

依据上述变异概率,个体 i 内的用户 k 按如下规则进行变异操作,即

$$l_{ik} = \left. \begin{aligned} & \text{round}(a_1 B + (1 - a_1)l_{ik}), a_2 > 0.5 \\ & \text{round}(a_1 + (1 - a_1)l_{ik}), a_2 \leq 0.5 \end{aligned} \right\} \quad (19)$$

$$q_{ik} = \left. \begin{aligned} & \text{round}(a_1 M_k + (1 - a_1)q_{ik}), a_2 > 0.5 \\ & \text{round}((1 - a_1)q_{ik}), a_2 \leq 0.5 \end{aligned} \right\} \quad (20)$$

$$u_{ik} = \left. \begin{aligned} & \text{round}(a_1 q_{ik} + (1 - a_1)u_{ik}), a_2 > 0.5 \\ & \text{round}((1 - a_1)u_{ik}), a_2 \leq 0.5 \end{aligned} \right\} \quad (21)$$

$$v_{ik} = \left. \begin{aligned} & (1 - a_1)v_{ik} + a_1 p_k^{\max}, a_2 > 0.5 \\ & (1 - a_1)v_{ik}, a_2 \leq 0.5 \end{aligned} \right\} \quad (22)$$

其中, a_1 和 a_2 均为(0,1)区间内均匀分布的随机数。更大的 a_1 意味着更大的突变幅度,而 a_2 则控制搜索方向^[18]。

3.7 多样性评价

为克服早熟收敛问题,在自适应交叉和变异之前进行多样性评价,让其引导变异。对多维数值问题,其多样性评价^[9]定义为

$$\begin{aligned} \bar{d} &= dl + dq + du + dv \\ &= (IL_1)^{-1} \sum_{i \in \mathcal{I}} \sqrt{\sum_{k \in \mathcal{K}} (l_{ik} - l_k^{\text{ave}})^2} \\ &\quad + (IL_2)^{-1} \sum_{i \in \mathcal{I}} \sqrt{\sum_{k \in \mathcal{K}} (q_{ik} - q_k^{\text{ave}})^2} \\ &\quad + (IL_3)^{-1} \sum_{i \in \mathcal{I}} \sqrt{\sum_{k \in \mathcal{K}} (u_{ik} - u_k^{\text{ave}})^2} \\ &\quad + (IL_4)^{-1} \sum_{i \in \mathcal{I}} \sqrt{\sum_{k \in \mathcal{K}} (v_{ik} - v_k^{\text{ave}})^2} \end{aligned} \quad (23)$$

$$\begin{aligned} l_k^{\text{ave}} &= \sum_{i \in \mathcal{I}} l_{ik} / I, q_k^{\text{ave}} = \sum_{i \in \mathcal{I}} q_{ik} / I, \\ u_k^{\text{ave}} &= \sum_{i \in \mathcal{I}} u_{ik} / I, p_k^{\text{ave}} = \sum_{i \in \mathcal{I}} v_{ik} / I \end{aligned} \quad (24)$$

其中, L_1, L_2, L_3 和 L_4 分别为问题P1中 $\mathbf{X}, \mathbf{y}, \mathbf{z}$ 和 \mathbf{p} 的可行域内对角线的长度。

然后,种群所有个体按概率 pd 进行多样性引导的变异操作, pd 记为

$$\begin{aligned} pd &= b_5, \bar{d} < \bar{d}_1; pd = b_6, \bar{d}_1 \leq \bar{d} < \bar{d}_2; \\ pd &= b_7, \bar{d} \geq \bar{d}_2 \end{aligned} \quad (25)$$

其中, $0 < b_5 \leq 1$, $0 < b_6 \leq 1$ 与 $0 < b_7 \leq 1$ 为参数; $0 < \bar{d}_1 < 1$ 和 $0 < \bar{d}_2 < 1$ 为多样性阈值。

至此,基于改进遗传算法(AGADGM)的协作式计算卸载与资源分配算法可归纳如下:

步骤1 设置迭代次数 $t = 1$;对优化问题的优化参量 $\{\mathbf{X}, \mathbf{y}, \mathbf{z}, \mathbf{p}\}$ 进行编码;根据式(15)初始化种群。

步骤2 按照式(7)和式(9)求取计算资源;根据式(16)计算所有个体的个体适应度值,并找出当前最佳个体;如果当前最佳个体的适应度值高于历史最佳个体的适应度值,则以前者替换后者。

步骤3 重复如下操作;

步骤4 根据锦标赛选择法选择个体建立新种群;如果历史最佳个体没有进入下一代,则以历史最佳个体替换种群中的最差个体;根据式(25)中的概率,按式(19)一式(22)进行多样性引导变异;按照式(7)和式(9)求取计算资源,并根据式(16)计算所有个体的适应度值。

步骤5 任意两个相邻个体按照式(17)中的自适应交叉概率执行交叉操作;根据式(18)中的自适应变异概率,所有个体按式(19)一式(22)进行变异操作。

步骤6 按照式(7)和式(9)求取计算资源;根据式(16)计算所有个体的适应度值,并找到当前种群的最佳个体。如果当前最佳个体的适应度值高于历史最佳个体的适应度值,则以前者替换后者。

步骤7 按照 $t = t + 1$ 更新迭代次数;如果 $t \leq T$ (T 为最大迭代次数),则回到步骤3,反之则输出历史最佳个体的染色体片段,并终止循环。

4 系统仿真

4.1 参数设置

本文考虑一个蜂窝宏小区内随机部署35个SBS的场景,其中宏小区内用户终端数量小于等于SBS数量。此外, $a_k = 10^{-25}$, $r_0 = 1$ Gps, $\xi = 1$ mW^[5]; $\bar{\varepsilon}_n = \bar{\varepsilon}_n = \bar{\varepsilon}_{m_n} = 1$ W/GHz^[19]; $\eta_k = 10$; $b_1 = 0.8$, $b_2 = 0.8$, $b_3 = 0.3$, $b_4 = 0.3$, $b_5 = 0.6$, $b_6 = 0.03$, $b_7 = 10^{-5}$, $\bar{d}_1 = 0.01$, $\bar{d}_2 = 0.05$ ^[18]。其他参数设置如表1所示。

为凸显本文算法的有效性,仿真中引入其他计算卸载与资源分配算法,包括最强信道增益关联算法(Maximum Channel Gain Association algorithm, MCGA)^[9]、基于传统遗传算法的计算卸载与资源分配算法(Traditional Genetic Algorithm based Scheme, TGAS)^[18]及全本地计算算法(Local Computing Algorithm, LCA)^[5]作比较。在MCGA中,任意用户将计算任务全部卸载至最强信道增益的基站子信道;TGAS则固定交叉和变

异概率,且没有引入多样性保护变异;LCA让用户自己完成所有任务。

4.2 仿真结果分析

图2显示了用户最大发射功率对系统效能的影响。在保持 $\mu = 0.5$, $K = 35$ 及其他条件不变的情况下,设置任意用户 k 的最大发射功率 p_k^{\max} 为同一常数。从式(1)、式(2)、式(6)和式(10)可知, p_k^{\max} 的增加会导致上传能耗的增加。如图2(a)所示,总体上,TGAS, MCGA与本文算法的系统总能耗随着 p_k^{\max} 的增长而增长。但是,因为LCA中用户任务全部本地执行,故其系统总能耗不随该功率变化。在所有算法中,MCGA可以达到最小系统总能耗。究其原因,该模式下用户总是接入信道质量最好的基站子信道,拥有较其他算法更低的上传时延,并由此导致更低的能耗。此外,LCA以最大计算能力计算导致了最高系统总能耗。相比于TGAS,本文算法由于其更强的搜索能力而达到了更低的系统总能耗。图2(b)显示了不同的 p_k^{\max} 对系统支持率的影响。可以看出,TGAS, LCA与本文算法总能支持所有用户在规定时间内完成所有计算任务。然而,MCGA支持率却一直低于1,且几乎不随 p_k^{\max} 变化。其原因在于,MCGA忽视了时延约束,TGAS与本文算法则考虑在时延约束下优化系统总能耗,而LCA没有上传时延。

在保持 $\mu = 0.5$ 、用户最大发射功率为23 dBm及其他条件不变的情况下,图3显示了用户密度

表1 参数设置

参数	数值	参数	数值
系统带宽 W	20 MHz	用户计算能力 F_k^{UE}	1 GHz
子信道带宽 w	15 kHz	截止时延 T_k^{max}	5~10 s
噪声功率谱密度 σ^2	-174 dBm/Hz	用户最大发射功率 p_k^{max}	23 dBm
用户 k 的任务数 M_k	3~7 个	基站计算能力 $F^{\text{MBS}}, F^{\text{SBS}}$	20 GHz
单个任务数据大小 d_k	200~500 kB	种群大小 I	64
单个任务执行时所需的CPU周期数 c_k	50~100 cycles/bit		

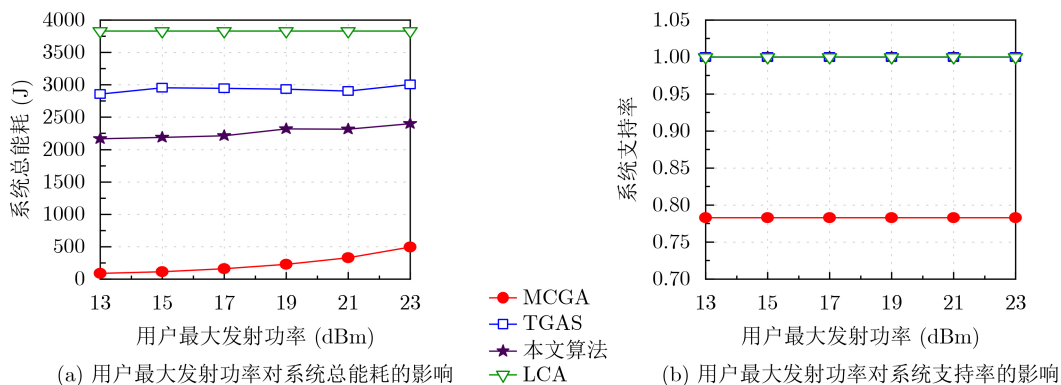


图2 用户最大发射功率 p_k^{\max} 对系统总能耗和系统支持率的影响

ρ (宏小区内用户个数)对系统效能的影响。根据式(13)可知, ρ 值的增大必定会导致系统总能耗的增大。如图3(a)所示, 4种算法的系统总能耗均随 ρ 的增长而增长。类似于图2(a), 图3(a)中MCGA系统总能耗一直低于其他算法, 本文算法获得了较前者更高的系统总能耗。类似于图2(b), 图3(b)中TGAS, LCA与本文算法总能耗支持所有用户在规定时间内完成所有计算任务。然而, MCGA支持率却一直低于1, 且总体上随着 ρ 先增加后下降。在低 ρ 域, 用户密度的增加导致靠近基站的用户增加, 系统支持率可能随 ρ 上升; 在高 ρ 域, 用户密度的增加导致可利用资源的下降, 系统支持率可能随 ρ 下降。

在保持 $K = 35$ 、用户最大发射功率为23 dBm及其他条件不变的情况下, 图4显示了频谱划分因子 μ 对系统效能的影响。由2.2节的资源模型可知, μ 越大, MBS的频带资源越多, SBS的频带资源则越少。如图4(a)所示, 总体上, TGAS, MCGA与本文算法的系统总能耗随着 μ 的增大而增加。其原因在于, 随着 μ 的增大, 部分选择SBS的用户因频带资源不足转而选择MBS。由式(6)可知, 距离MBS较远的用户越多, 传输时延及能耗越大, 系统总能耗也随之增加。但是, 因为LCA中用户任务全部本地执行, 故其系统总能耗不随 μ 变化。此外, 类似

于图2(a)、图4(a)中MCGA系统总能耗一直低于其他算法, 本文算法获得了较前者更高的系统总能耗。众所周知, MBS中MEC服务器计算资源有限。根据式(7)可知, 越多用户接入MBS, 用户分配到的计算资源越少, 计算时延就越长, 于是越来越多的用户不满足时延约束。因此, 图4(b)中MCGA的系统支持率随着 μ 的增大而减小。类似于图2(b)和图3(b), 图4(b)显示TGAS, LCA与本文算法总能耗支持所有用户在规定时间内完成所有计算任务。

图5显示了遗传算法中最佳个体适应度值的收敛情况, 其中 $K = 35$, $\mu = 0.5$ 及用户最大发射功率为23 dBm。容易发现, TGAS具有较本文算法更快的收敛速度, 但因前者陷入局部最优而导致获得较后者更差的适应度值。究其原因, 固定的交叉和变异概率容易使得TGAS陷于局部最优, 过早收敛。而本文算法引入了多样性引导变异、自适应交叉和变异概率等操作, 这些操作使得搜索更加细致的同时, 避免了过早收敛。

综合上述, 尽管MCGA能达到较其他算法更低的系统总能耗, 但其系统支持率一直较其他算法更低。虽然本文算法的系统总能耗高于MCGA, 但它都低于LCA和TGAS。此外, 本文算法的系统

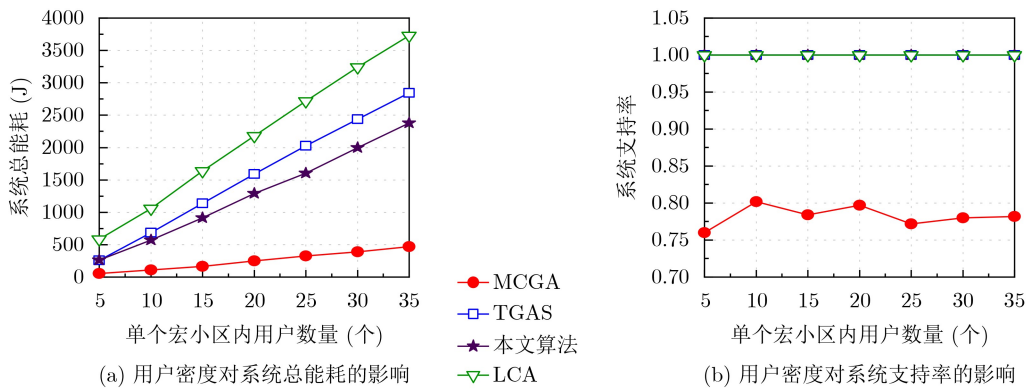


图3 用户密度 ρ 对系统总能耗和系统支持率的影响

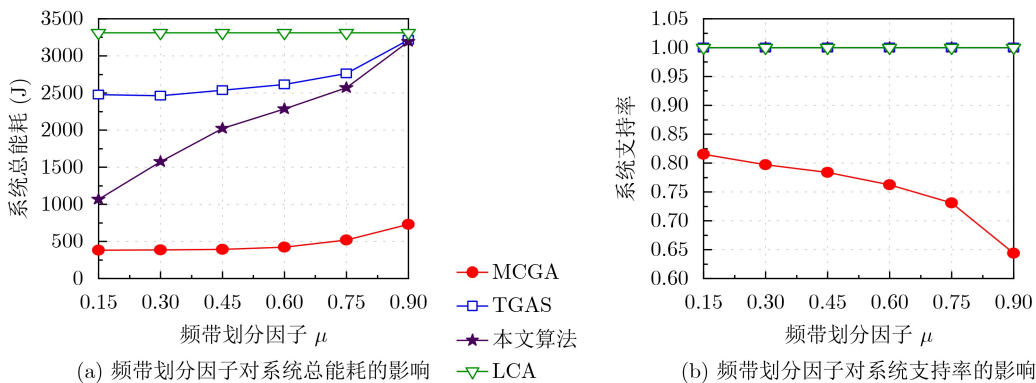


图4 频带划分因子 μ 对系统总能耗和系统支持率的影响

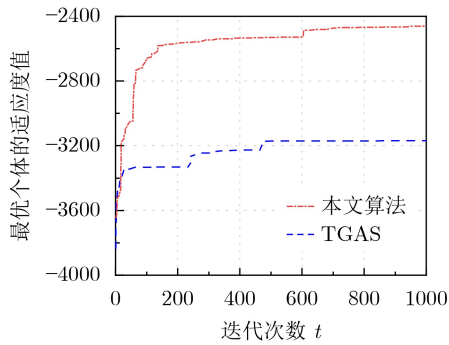


图5 最优个体的适应度值在遗传算法下的搜索情况

支持率始终为1,即本文算法总能支持所有用户在规定时间内完成所有计算任务。总体而言,同其他算法相比,本文提出的基于改进遗传算法(AGADGM)的协作式计算卸载与资源分配算法具有一定优势,且能在问题约束条件下较好地优化系统能耗。

5 结束语

本文构建了UDHN场景下的NOMA-MEC系统,并在该系统中研究了多用户多任务协作式卸载策略。为合理消除干扰和分配频谱资源,融入了频带划分机制与上行NOMA技术。然后,在计算资源占比分配策略与时延约束下,联合优化用户关联、卸载决策与用户发射功率以最小化系统总能耗。为求解所规划的问题,在改进遗传算法(AGADGM)的基础上,提出了协作式计算卸载与资源分配算法。仿真结果表明,同其他算法相比,本文所设计计算卸载与资源分配算法具有一定优势,且能在问题约束条件下较好地优化系统能耗。未来研究可涉及单用户多信道接入场景与大规模天线场景等。

参考文献

- [1] MAO Yuyi, YOU Changsheng, ZHANG Jun, *et al.* A survey on mobile edge computing: The communication perspective[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(4): 2322–2358. doi: [10.1109/COMST.2017.2745201](https://doi.org/10.1109/COMST.2017.2745201).
- [2] GUO Hongzhi, ZHANG Jie, LIU Jiajia, *et al.* Energy-aware computation offloading and transmit power allocation in ultradense IoT networks[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4317–4329. doi: [10.1109/JIOT.2018.2875535](https://doi.org/10.1109/JIOT.2018.2875535).
- [3] ZHAO Junhui, SUN Xiaoke, LI Qiuping, *et al.* Edge caching and computation management for real-time internet of vehicles: An online and distributed approach[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(4): 2183–2197. doi: [10.1109/TITS.2020.3012966](https://doi.org/10.1109/TITS.2020.3012966).
- [4] ZHOU Tianqing, JIANG Nan, LIU Zunxiong, *et al.* Joint cell activation and selection for green communications in ultra-dense heterogeneous networks[J]. *IEEE Access*, 2018, 6: 1894–1904. doi: [10.1109/ACCESS.2017.2780818](https://doi.org/10.1109/ACCESS.2017.2780818).
- [5] DAI Yueyue, XU Du, MAHARJAN S, *et al.* Joint computation offloading and user association in multi-task mobile edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(12): 12313–12325. doi: [10.1109/TVT.2018.2876804](https://doi.org/10.1109/TVT.2018.2876804).
- [6] 刘海燕. 面向5G的超密集网络中分布式无线资源管理的研究[D]. [博士学位], 北京交通大学, 2016.
LIU Haiyan. Research on distributed radio resource management in ultra dense networks for 5G communication systems[D]. [Ph. D. Dissertation]. Beijing Jiaotong University, 2016.
- [7] WU Yuhang, WANG Yuhao, ZHOU Fuhui, *et al.* Computation efficiency maximization in OFDMA-based mobile edge computing networks[J]. *IEEE Communications Letters*, 2020, 24(1): 159–163. doi: [10.1109/LCOMM.2019.2950013](https://doi.org/10.1109/LCOMM.2019.2950013).
- [8] DENG Maofei, TIAN Hui, and LYU Xinchen. Adaptive sequential offloading game for multi-cell mobile edge computing[C]. 2016 23rd International Conference on Telecommunications (ICT), Thessaloniki, Greece, 2016: 1–5. doi: [10.1109/ICT.2016.7500395](https://doi.org/10.1109/ICT.2016.7500395).
- [9] ZHOU Tianqing, QIN Dong, NIE Xuefang, *et al.* Energy-efficient computation offloading and resource management in ultradense heterogeneous networks[J]. *IEEE Transactions on Vehicular Technology*, 2021, 70(12): 13101–13114. doi: [10.1109/TVT.2021.3116955](https://doi.org/10.1109/TVT.2021.3116955).
- [10] ZHOU Fuhui and HU R Q. Computation efficiency maximization in wireless-powered mobile edge computing networks[J]. *IEEE Transactions on Wireless Communications*, 2020, 19(5): 3170–3184. doi: [10.1109/TWC.2020.2970920](https://doi.org/10.1109/TWC.2020.2970920).
- [11] 张海波, 刘香渝, 荆昆仑, 等. 车联网中基于NOMA-MEC的卸载策略研究[J]. 电子与信息学报, 2021, 43(4): 1072–1079. doi: [10.11999/JEIT200017](https://doi.org/10.11999/JEIT200017).
ZHANG Haibo, LIU Xiangyu, JING Kunlun, *et al.* Research on NOMA-MEC-based offloading strategy in internet of vehicles[J]. *Journal of Electronics & Information Technology*, 2021, 43(4): 1072–1079. doi: [10.11999/JEIT200017](https://doi.org/10.11999/JEIT200017).
- [12] CHENG Qianqian, LI Lixin, SUN Yan, *et al.* Efficient resource allocation for NOMA-MEC system in ultra-dense network: A mean field game approach[C]. 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 2020: 1–6. doi: [10.1109/ICCWorkshops49005.2020.9145070](https://doi.org/10.1109/ICCWorkshops49005.2020.9145070).
- [13] GUO Fengxian, ZHANG Heli, JI Hong, *et al.* An efficient computation offloading management scheme in the densely

- deployed small cell networks with mobile edge computing[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(6): 2651–2664. doi: [10.1109/TNET.2018.2873002](https://doi.org/10.1109/TNET.2018.2873002).
- [14] ZHOU Tianqing, ZHAO Junhui, QIN Dong, *et al.* Joint user association and time partitioning for load balancing in ultra-dense heterogeneous networks[J]. *Mobile Networks and Applications*, 2021, 26(2): 909–922. doi: [10.1007/s11036-019-01351-2](https://doi.org/10.1007/s11036-019-01351-2).
- [15] YANG Zheng, DING Zhiguo, FAN Pingzhi, *et al.* A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems[J]. *IEEE Transactions on Wireless Communications*, 2016, 15(11): 7244–7257. doi: [10.1109/TWC.2016.2599521](https://doi.org/10.1109/TWC.2016.2599521).
- [16] PHAM Q V, NGUYEN H T, HAN Zhu, *et al.* Coalitional games for computation offloading in NOMA-enabled multi-access edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(2): 1982–1993. doi: [10.1109/TVT.2019.2956224](https://doi.org/10.1109/TVT.2019.2956224).
- [17] ZHOU Tianqing, YUE Yali, QIN Dong, *et al.* Joint device association, resource allocation and computation offloading in ultra-dense multi-device and multi-task IoT networks[J]. *IEEE Internet of Things Journal*, To be published. doi: [10.1109/JIOT.2022.3161670](https://doi.org/10.1109/JIOT.2022.3161670).
- [18] LI Meiyi, CAI Zixing, and SUN Guoyun. An adaptive genetic algorithm with diversity-guided mutation and its global convergence property[J]. *Journal of Central South University of Technology*, 2004, 11(3): 323–327. doi: [10.1007/s11771-004-0066-6](https://doi.org/10.1007/s11771-004-0066-6).
- [19] ZHANG Ke, MAO Yuming, LENG Supeng, *et al.* Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks[J]. *IEEE Access*, 2016, 4: 5896–5907. doi: [10.1109/ACCESS.2016.2597169](https://doi.org/10.1109/ACCESS.2016.2597169).
- 周天清: 男, 副教授, 研究方向为超密集组网、移动边缘计算。
胡海琴: 女, 硕士生, 研究方向为超密集组网、移动边缘计算。
曾新亮: 男, 硕士生, 研究方向为超密集组网、移动边缘计算。

责任编辑: 马秀强