

## 基于改进NSGA-III的D2D协同MEC多目标优化研究

王志鸿, 王高才, 赵启飞

### 引用本文

王志鸿, 王高才, 赵启飞. 基于改进NSGA-III的D2D协同MEC多目标优化研究[J]. 计算机科学, 2024, 51(3): 280-288.

WANG Zhihong, WANG Gaocai, ZHAO Qifei. [Multi-objective Optimization of D2D Collaborative MEC Based on Improved NSGA-III](#) [J]. Computer Science, 2024, 51(3): 280-288.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

**Similar articles recommended (Please use Firefox or IE to view the article)**

#### [多无人机辅助MEC环境中基于Wardrop路由博弈的计算卸载](#)

Computation Offloading with Wardrop Routing Game in Multi-UAV-aided MEC Environment

计算机科学, 2024, 51(3): 309-316. <https://doi.org/10.11896/jsjcx.221100242>

#### [基于在线学习稀疏特征的大规模多目标进化算法](#)

Large-scale Multi-objective Evolutionary Algorithm Based on Online Learning of Sparse Features

计算机科学, 2024, 51(3): 56-62. <https://doi.org/10.11896/jsjcx.230100004>

#### [面向缓存的动态协作任务迁移技术研究](#)

Study on Cache-oriented Dynamic Collaborative Task Migration Technology

计算机科学, 2024, 51(2): 300-310. <https://doi.org/10.11896/jsjcx.230600128>

#### [面向高速行驶车辆的在线任务卸载决策算法](#)

Online Task Offloading Decision Algorithm for High-speed Vehicles

计算机科学, 2024, 51(2): 286-292. <https://doi.org/10.11896/jsjcx.221200069>

#### [应急通信场景下基于JTORATPAIA的NOMA-MEC系统研究](#)

Study on NOMA-MEC System Based on JTORATPAIA in Emergency Communication Scenarios

计算机科学, 2023, 50(11A): 221000240-8. <https://doi.org/10.11896/jsjcx.221000240>

# 基于改进 NSGA-III 的 D2D 协同 MEC 多目标优化研究

王志鸿<sup>1</sup> 王高才<sup>1</sup> 赵启飞<sup>2</sup>

1 广西大学计算机与电子信息学院 南宁 530004

2 广西大学电气工程学院 南宁 530004

(wangzhgx@foxmail.com)

**摘要** 在当前的移动边缘计算(Mobile Edge Computing, MEC)模型中,由于任务是直接上传到 MEC 服务器执行,存在边缘服务器的计算压力大、空闲移动设备上的资源未得到充分利用等问题。使用边缘网络中的空闲设备进行协同计算,能够实现用户闲置资源的合理利用,增强 MEC 的计算能力。因此,提出了一种利用终端直通(Device-to-Device, D2D)进行协同计算的部分卸载 MEC 模型(D2D Collaborative MEC for Partial Offloading, DCM-PO)。在该模型中,除本地计算和 MEC 服务器计算外,还能将部分任务上传到空闲 D2D 设备进行辅助计算。首先,以最小化边缘网络的时延、能耗和费用为目标建立多目标优化问题。然后,在多染色体混合编码、自适应交叉率和变异率等方面对基于参考点的非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm III, NSGA-III)进行改进,使之适合 DCM-PO 模型中的多目标优化问题求解。最后,仿真结果表明,相比基准 MEC 模型,DCM-PO 模型在多项性能指标上有明显优势。

**关键词:** 移动边缘计算; D2D; 任务卸载; 多目标优化; NSGA-III

**中图分类号** TP301

## Multi-objective Optimization of D2D Collaborative MEC Based on Improved NSGA-III

WANG Zhihong<sup>1</sup>, WANG Gaocai<sup>1</sup> and ZHAO Qifei<sup>2</sup>

1 School of Computer and Electronic Information, Guangxi University, Nanning 530004, China

2 School of Electrical Engineering, Guangxi University, Nanning 530004, China

**Abstract** In the current mobile edge computing(MEC), since tasks are directly uploaded to the MEC server for execution, there are problems such as high computing pressure on the edge server and insufficient utilization of resources on idle mobile devices. Using idle devices in the edge network for collaborative computing can realize rational utilization of user's idle resources and enhance the computing capacity of MEC. Therefore, a device-to-device(D2D) collaborative MEC for partial offloading(DCM-PO) is proposed. In this model, in addition to local computing and MEC server computing, part of the tasks can be uploaded to idle D2D devices for auxiliary computing. First, a multi-objective optimization problem is established to minimize the delay, energy consumption and cost of the edge network. Then, the non-dominated sorting genetic algorithm III(NSGA-III) is improved in the aspects of multi-chromosome mixed coding, adaptive crossover rate and mutation rate, so that it is suitable for solving the multi-objective optimization problem in the DCM-PO. Finally, simulation results show that, compared with the baseline MEC, the DCM-PO has advantages in several performance indicators.

**Keywords** Mobile edge computing, Device-to-Device, Task offloading, Multi-objective optimization, NSGA-III

## 1 引言

为了满足新型智能应用对计算资源的需求,移动边缘计算模式(Mobile Edge Computing, MEC)被提出,其核心思想是将具备高速计算能力的边缘服务器部署到网络边缘,如蜂窝基站和网络汇聚点等<sup>[1]</sup>。移动设备通过将任务卸载到边缘服务器执行,能够有效降低任务的本地执行开销。终端直通(Device-to-Device, D2D)是 5G 通信的关键技术之一,在 D2D

通信模型中,用户设备具有独立的收发功能。在信号覆盖范围内,数据可以直接从一个终端传递到另一个终端,为用户进行端到端的资源共享提供支持<sup>[2]</sup>。

在经典 MEC 的基础上,将部分任务上传到空闲 D2D 设备进行辅助计算,能够提高 MEC 的整体性能。首先,相比 MEC 服务器,任务设备与 D2D 设备的距离通常更近,因此能够减少任务迁移产生的能耗。其次,通过对计算任务进行划分,使得本地部分、D2D 部分和 MEC 部分并行执行,能够

到稿日期:2022-11-29 返修日期:2023-01-11

基金项目:国家自然科学基金(62062007)

This work was supported by the National Natural Science Foundation of China(62062007).

通信作者:王高才(wangcgx@163.com)

提高任务的执行效率,缩短服务时延。最后,在加入 D2D 协同计算之后,对 MEC 系统而言,增加了计算结点的数量,扩展了 MEC 的计算容量。对用户而言,任务设备能够利用定价相对更低的 D2D 设备进行辅助计算,降低了计算成本。提供计算能力的 D2D 设备也能够从辅助计算中获取收益,实现了双赢。以下是适合将 D2D 协同计算引入 MEC 的典型应用场景。

(1)低功耗物联网。将计算密集型任务迁移至资源更丰富的 D2D 设备执行,能够有效延长物联网设备的持续工作时间。例如,使用智能手表分析过去 24h 的心电图(约 8 万~14 万次心跳数据),能够为心脏疾病的诊断和治疗提供参考,但是需要消耗大量能量<sup>[3]</sup>。在 D2D 协同的 MEC 中,智能手表首先利用 Wi-Fi 等无线技术,与周围其他性能较强的智能设备(智能手机、平板电脑等)建立 D2D 连接。然后通过数据分解,对心电图分析任务进行划分,如最初的 2 万次心跳数据在本地分析,接下来的 5 万次在 D2D 设备分析,剩余的 7 万次在 MEC 服务器分析,最后汇总各部分的分析结果。

(2)自动驾驶。D2D 协同的 MEC 能够有效扩展车辆的计算能力,为基于深度学习的自动驾驶提供支持。首先,车辆利用无线技术与司机、乘客或者路边用户的智能设备建立 D2D 连接。然后,通过功能分解,对寻路、避障和预警等任务进行划分,如用于寻路的数据在本地进行分析,用于避障的数据上传到 D2D 设备进行分析,用于预警的数据上传到 MEC 服务器进行分析。最后,汇总各部分分析结果,做出合理的机械动作。在此期间,如果与车辆配对的智能设备仍然无法承担所有计算任务,可以进一步对任务进行划分,如正面避障保留在本地进行分析,左右避障则间接上传到 MEC 服务器进行分析。由于车辆与其他用户设备的距离通常很近,因此,D2D 协同计算能够实现极低的传输时延。

本文的主要工作及创新之处如下:

(1)在经典的 MEC 模型中,用户设备之间是相互独立的,因此提出了一种 D2D 协同的部分卸载 MEC 模型(D2D Collaborative MEC for Partial Offloading,DCM-PO),空闲设备可以通过 D2D 链路为任务设备提供数据中转和计算服务。

(2)经典的基于参考点的非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm III,NSGA-III)是单染色体、固定交叉率和变异率的,因此提出了多染色体和余弦递减交叉率、反余弦递增变异率的改进策略。

## 2 相关研究工作

在经典的移动边缘计算研究中,任务设备通常与 MEC 服务器建立多对一的联系,然后利用决策引擎将待执行任务分配到本地或者 MEC 服务器执行,最后取回计算结果。这类研究的关注重点通常是如何划分任务和分配资源。例如,文献[4]在集成 MEC 的 IoT 网络中,对无线电资源和计算资源进行分配以提高能源效率。文献[5]为了解决复杂任务的迁移问题,基于任务间的依赖关系构建了子任务划分模型,在时延约束条件下利用遗传算法使得能耗最小化。文献[6]将无线功率传输引入 MEC,通过优化能量传输波束、卸载任务量和 CPU 频率,使得模型能耗在时延约束条件下最小化。

文献[7]首先将能耗和时延标准化,然后提出了一种迭代式算法求解任务分配和功率分配问题,在时延和能耗之间取得了平衡。

由于经典的移动边缘计算模型具有中心化的特点,网络的计算能力过于依赖边缘服务器,随着计算负载逐渐增大,MEC 服务器将成为网络的性能瓶颈,这就使得经典的移动边缘计算模型不适用于大规模网络<sup>[8]</sup>。利用广泛分布的边缘设备进行辅助计算,能够有效解决该问题。文献[9]提出了一种任务混合卸载的雾计算模型,将最小化任务执行成本问题转化为三层图上的最小权值问题。文献[10]构建了用户设备辅助的 MEC 模型,通过将子任务分配给基站和相邻用户,减少了时延和能量消耗。文献[11]提出了雾计算设备协作的 5G 蜂窝网络,在将计算卸载问题表述为随机优化问题后使用李雅普诺夫优化方法求解。文献[12]在由用户结点、辅助结点和接入点组成的 MEC 模型中,提出了实现联合通信和计算的四槽协议。文献[13]研究了用户设备协助下的 MEC 模型,通过将计算卸载问题定义为一个潜在博弈模型,在时延约束条件下降低了整体能耗。

移动边缘计算模型的主要性能指标有时延、能耗和费用等。在最小化时延方面,文献[14]在一个多用户多服务器 MEC 模型中,将最小化时延问题分解为任务卸载问题和传输功率分配问题,并使用匹配理论和启发式方法交替求解。文献[15]提出了一种结合云计算和移动边缘计算的三层网络结构,并使用 Kruskal 算法计算最小生成树以减少通信延迟。文献[16]研究了以认知无人机作为中继结点的 MEC 模型,通过联合优化用户的传输功率,制定了以最小化用户处理时延为目标的优化问题。最小化时延的移动边缘计算模型适用于处理时延敏感型任务,如人机交互术、自动驾驶等。在实际中,还存在大量能耗敏感型应用,如物联网应用、可穿戴设备应用等,这促进了对最小化能耗的移动边缘计算模型的研究。文献[4-6,10,12-13]都是基于最小化能耗的考虑,其中文献[10]使用基于背包问题的预分配算法,利用变量替换技术进行凸优化从而获得最优传输功率。此外,文献[17]研究了一种分层的 MEC 模型,通过优化移动设备的传输功率,在延迟约束条件下最小化移动设备的能耗。文献[18]通过探索设备的功率损耗特性,设计了按需分配的资源分配模型,使得能耗最小化。

最小化时延和最小化能耗都有各自适用的应用范围,实际上,平衡时延和能耗的需求比单一最小化时延或者最小化能耗具有更重要的意义。用户终端的电池容量是有限的,对于电池容量小、不具备充电环境的设备,如布置在野外的物联网设备、植入在体内的医疗设备等,需要平衡时延和能耗。文献[7,9-11,19]都是基于平衡时延和能耗的考虑。其中,文献[7]在将时延和能耗归一化之后,使用权重因子来控制时延和能耗的平衡。文献[11]利用变量替换技术进行凸优化,然后使用交替优化卸载决策和传输功率的方法取得最优解。文献[19]在多用户多 MEC 服务器的物联网环境中,考虑 IoT 系统的能源效率和服务延迟之间的权衡,并提出了一种基于李雅普诺夫优化的在线卸载和资源分配算法用于生成任务卸载方案。

针对移动边缘计算模型的费用问题,文献[20]通过将用户的计算成本定义为时延与费用之和,建立了边缘云与用户之间的博弈模型,在最大化边缘云收益的同时最小化用户成本,这种方案的缺陷在于时延和费用的量纲是不同的。文献[21]在车联网系统中,允许停泊车辆的车主通过共享空闲的计算能力获取收益,并使用一种动态定价策略使得成本最小化。文献[22]在资源受限的移动边缘计算模型中考虑资源分配和计费问题,并使用一种基于拍卖的分配机制来获取最优解。文献[23]提出了一种基于策略梯度的强化学习算法用于设置计价策略,使得 MEC 服务器的收益最大化。

综上所述,在目前的移动边缘计算研究中,存在如下问题:

- (1) 位于网络边缘的 MEC 服务器直接承载用户的计算任务,MEC 服务器有可能成为移动边缘计算模型的瓶颈。
- (2) 网络边缘存在大量空闲设备,这部分设备的资源可以被进一步利用,以提高移动边缘计算模型的性能。
- (3) 根据用户需求,需要进一步考虑对移动边缘计算模型的多个性能指标进行同时优化。

### 3 系统模型

#### 3.1 网络模型

图 1 给出了所提 DCM-PO 模型的物理组件,其由任务设备、MEC 服务器和 D2D 设备组成。任务设备产生计算任务,具有一定计算能力。通常情况下,任务全部在本地执行会带来过大的计算开销。MEC 服务器部署在蜂窝基站附近,通过高速链路与中心云连接,为网络边缘用户提供计算资源。D2D 设备是边缘网络中具有计算资源或通信资源的空闲设备。

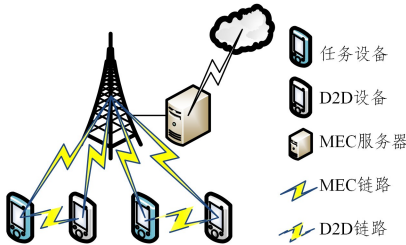


图 1 DCM-PO 物理组件

Fig. 1 Physical components of DCM-PO

图 2 给出了 DCM-PO 的任务卸载模型。首先,任务设备的应用层产生计算任务后,由本地决策引擎将任务划分为 MEC 计算、D2D 辅助计算和本地计算 3 部分,并分别传输到 MEC 服务器、合适的 D2D 设备和本地 CPU。然后,D2D 设备可以将任务进一步划分为本地计算和间接上传 MEC 服务器两部分,利用自身较强的计算能力和通信能力完成辅助计算。最后,在费用问题上,MEC 服务器的流量费率和 CPU 时间费率由运营商决定,D2D 设备的流量费率和 CPU 时间费率由其所有者决定。

本文的核心工作是建立决策引擎,使得移动边缘网络的总时延、能耗和费用最小化。研究工作建立在如下假设的前提下:

- (1) 任务可划分性。任务可以被划分为若干个能够

独立执行的子任务。

- (2) D2D 可行性。空闲设备愿意消耗自身的资源进行协同计算。

- (3) 连接稳定性。在完成计算任务之前,已经建立的 D2D 连接不会因为超出通信距离而断开。

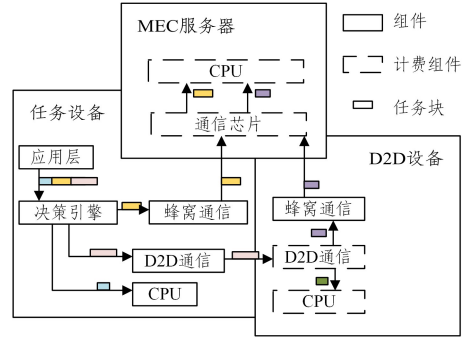


图 2 任务卸载模型

Fig. 2 Task offloading model

以上假设是具有普遍性的。首先,利用数据分解和功能分解等方法能够有效发现任务中的可并行部分,各个并行部分彼此无关,能够独立执行。其次,由于 D2D 本身就是一种资源共享技术,而且 DCM-PO 还允许用户从辅助计算中获取收益,大多数用户都将倾向于开放自身设备的资源。最后,目前用于 D2D 通信的无线通信技术已经相对成熟,例如, Wi-Fi 直连 (Wi-Fi Direct) 可以在约 200 m 的通信距离内提供最高 250 Mbps 的传输速率; LTE 直连 (LTE Direct) 能够在约 500 m 范围内提供最高 13.5 Mbps 的传输速率<sup>[24]</sup>。综合考虑用户对传输速率的要求以及用户的移动速度,选择合适的无线通信方式,可以近似认为 D2D 连接不会在任务卸载过程中发生中断。

基于以上假设,首先建立 DCM-PO 的时延模型,设集合  $\mathcal{K} = \{1, 2, \dots, K\}$  代表任务设备,集合  $\mathcal{J} = \{1, 2, \dots, I\}$  代表 D2D 设备,则模型中的移动设备总数可以表示为  $N = K + I$ 。设备  $k \in \mathcal{K}$  的一个任务由三元组  $\mathcal{T}_k = \{D_k, C_k, T_k^{\max}\}$  表示。其中  $D_k$  是以比特为单位的任务数据量;  $C_k$  是处理该任务的 1 比特数据需要花费的平均 CPU 周期数,即任务复杂度;  $T_k^{\max}$  是最大容忍时延。在产生计算任务后,设备  $k$  将任务数据  $D_k$  分为  $D_{k,l}$ ,  $D_{k,i}$  和  $D_{k,e}$  这 3 部分,分别由本地、D2D 设备和 MEC 服务器处理,即:

$$D_k = D_{k,l} + D_{k,i} + D_{k,e} \quad (1)$$

设分配给本地计算的 CPU 频率为  $f_{k,l}$ ,则本地计算时延为:

$$\tau_{k,l}^c = \frac{D_{k,l} C_k}{f_{k,l}} \quad (2)$$

直接利用 MEC 服务器进行辅助计算需要经过数据上传、MEC 服务器处理和结果回传,由于计算结果的体积通常很小,因此回传时延可以忽略不计。MEC 服务器辅助计算产生的总时延近似于上传时延和计算时延之和,即:

$$\tau_{k,e} = \tau_{k,e}^c + \tau_{k,e}^t \quad (3)$$

其中,

$$\tau_{k,e}^t = D_{k,e} / B_{k,e} \log_2 \left( 1 + \frac{p_{k,e} h_{k,e}}{\sigma^2} \right) \quad (4)$$

$$\tau_{k,e}^c = \frac{D_{k,e} C_k}{f_{k,e}} \quad (5)$$

其中,  $B_{k,e}$  是任务设备与 MEC 服务器的连接带宽,  $p_{k,e}$  是任务设备对 MEC 服务器的传输功率,  $h_{k,e}$  是信道增益,  $\sigma^2$  是环境噪声功率,  $f_{k,e}$  是 MEC 服务器分配给任务设备的 CPU 频率。

类似地, 任务上传到 D2D 设备的传输时延为:

$$\tau_{k,i}^d = D_{k,i} / B_{k,i} \log_2 \left( 1 + \frac{p_{k,i} h_{k,i}}{\sigma^2} \right) \quad (6)$$

任务同时上传到 D2D 设备和 MEC 服务器时, 不能超出最大发射功率的限制, 即:

$$p_{k,i} + p_{k,e} \leq p_k^{\max} \quad (7)$$

D2D 设备处理的任务  $D_{k,i}$  被分成两部分, 即:

$$D_{k,i} = D_{k,i,l} + D_{k,i,e} \quad (8)$$

其中,  $D_{k,i,l}$  部分由 D2D 设备使用本地 CPU 进行本地计算, 产生的计算时延为:

$$\tau_{k,i,l}^d = \frac{D_{k,i,l} C_k}{f_{k,i,l}} \quad (9)$$

与式(3)类似,  $D_{k,i,e}$  部分由 D2D 设备间接上传到 MEC 服务器, 产生的总时延近似为上传时延和计算时延之和, 即:

$$\tau_{k,i,e}^d = \tau_{k,i,e}^u + \tau_{k,i,e}^c \quad (10)$$

其中,

$$\tau_{k,i,e}^u = D_{k,i,e} / B_{k,i,e} \log_2 \left( 1 + \frac{p_{k,i,e} h_{k,i,e}}{\sigma^2} \right) \quad (11)$$

$$\tau_{k,i,e}^c = \frac{D_{k,i,e} C_k}{f_{k,i,e}} \quad (12)$$

其中,  $B_{k,i,e}$  是被选中的 D2D 设备与 MEC 服务器的连接带宽,  $p_{k,i,e}$  是 D2D 设备对 MEC 服务器的传输功率,  $h_{k,i,e}$  是信道增益,  $\sigma^2$  是环境噪声功率,  $f_{k,i,e}$  是 MEC 服务器分配给 D2D 设备的 CPU 频率。

由于各任务块能够并行执行, 则时延模型可以用如图 3 所示的 AOE 网表示, 其关键路径即为任务设备  $k$  利用 D2D 设备  $i$  执行一次任务卸载产生的总时延, 即:

$$\begin{aligned} T_{k,i,e} &= \max\{\tau_{k,l}^d, \tau_{k,e}^c + \tau_{k,e}^u, \tau_{k,i}^d + \tau_{k,i,l}^d, \tau_{k,i}^d + \tau_{k,i,e}^u + \tau_{k,i,e}^c\} \\ &= \max\{\tau_{k,l}^d, \tau_{k,e}^c, \tau_{k,i}^d + \tau_{k,i,l}^d, \tau_{k,i}^d + \tau_{k,i,e}^c\} \end{aligned} \quad (13)$$

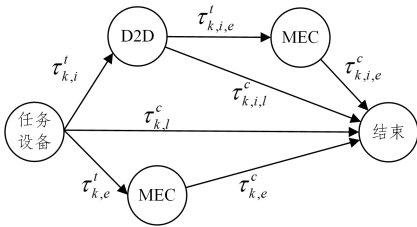


图 3 时延模型

Fig. 3 Delay model

为了满足时延要求, 需要有

$$T_{k,i,e} \leq T_k^{\max} \quad (14)$$

下面建立 DCM-PO 的能耗模型。任务设备的直接能耗等于本地执行能耗、上传 D2D 能耗和上传 MEC 服务器的能耗总和, 即

$$E_{k,i,e}^{\text{loc}} = \delta_k D_{k,i} C_k f_{k,i,l}^2 + p_{k,i} \tau_{k,i}^d + p_{k,e} \tau_{k,e}^c \quad (15)$$

其中,  $\delta_k$  是与 CPU 架构有关的能耗系数<sup>[25]</sup>。

由于数据接收过程是被动地从空间中接收电磁波的

过程, 其消耗的能量远小于发送数据所消耗的能量, 因此忽略 D2D 设备的接收能耗。则 D2D 设备的直接能耗近似于辅助计算和间接上传 MEC 服务器的能耗之和, 即:

$$E_{k,i,e}^{\text{aux}} = \delta_e D_{k,i} C_k f_{k,i,l}^2 + p_{k,i,e} \tau_{k,i,e}^c \quad (16)$$

由 MEC 服务器产生的计算能耗为:

$$E_{k,i,e}^{\text{mec}} = \delta_e D_{k,e} C_k f_{k,e}^2 + \delta_e D_{k,i,e} C_k f_{k,i,e}^2 \quad (17)$$

综上所述, 总能耗为:

$$E_{k,i,e} = E_{k,i,e}^{\text{loc}} + E_{k,i,e}^{\text{aux}} + E_{k,i,e}^{\text{mec}} \quad (18)$$

下面建立 DCM-PO 的费用模型。在 DCM-PO 中, MEC 服务器为不同用户设置不同的流量费率  $\lambda$  和 CPU 时间费率  $\mu$ , 根据用户上传的数据量和使用的 CPU 时间对用户计费。在 D2D 设备上, 不同的 D2D 设备为不同的任务设备设置不同的流量费率  $\alpha_{k,i}$  和 CPU 时间费率  $\beta_{k,i}$ , 对与自身建立 D2D 连接的任务设备收取流量费用和计算费用。

按照以上定义, 任务设备将任务直接上传到 MEC 服务器执行产生的费用为:

$$S_{k,i,e}^{\text{mec}} = \lambda_k D_{k,e} + \mu_k \tau_{k,i,e}^c \quad (19)$$

任务由 D2D 设备间接上传到 MEC 服务器执行产生的费用为:

$$S_{k,i,e}^{\text{mec}} = \lambda_i D_{k,i,e} + \mu_i \tau_{k,i,e}^c \quad (20)$$

D2D 设备的辅助计算费用为:

$$S_{k,i}^{\text{aux}} = \alpha_{k,i} D_{k,i} + \beta_{k,i} \tau_{k,i,l}^d \quad (21)$$

综上所述, 任务设备产生的总费用为:

$$S_{k,i,e} = S_{k,i,e}^{\text{mec}} + S_{k,i,e}^{\text{mec}} + S_{k,i}^{\text{aux}} \quad (22)$$

### 3.2 问题定义

对 D2D 协同下的移动边缘计算模型进行多目标优化, 优化变量包括任务的划分情况  $\mathbf{D} = (D_{k,l}, D_{k,i}, D_{k,e}, D_{k,i,l}, D_{k,i,e})$ 、移动设备的上传功率  $\mathbf{p} = (p_{k,i}, p_{k,e}, p_{k,i,e})$ 、移动边缘网络的 CPU 频率  $\mathbf{f} = (f_{k,l}, f_{k,i,l}, f_{k,e}, f_{k,i,e})$  和 D2D 辅助设备的配对规则  $\mathbf{u} = (u_{0,0}, u_{0,1}, \dots, u_{K,l})$ 。以最小化边缘网络的时延、能耗和任务设备的费用为目标建立多目标优化问题, 该优化问题可以表示为:

$$\begin{aligned} \mathcal{P}1: \min & \left\{ \sum_{k \in \mathcal{X}} \sum_{i \in \mathcal{J}} u_{k,i} T_{k,i,e}, \sum_{k \in \mathcal{X}} \sum_{i \in \mathcal{J}} u_{k,i} E_{k,i,e}, \right. \\ & \left. \sum_{k \in \mathcal{X}} \sum_{i \in \mathcal{J}} u_{k,i} S_{k,i,e} \right\} \\ \text{s. t. } & \text{C1: } 0 \leq D \leq D_k, \forall D \in \{D_{k,l}, D_{k,i}, D_{k,e}\} \\ & \text{C2: } 0 \leq D \leq D_{k,i}, \forall D \in \{D_{k,i,l}, D_{k,i,e}\} \\ & \text{C3: } 0 \leq p_{k,i,e} \leq p_k^{\max}, \forall i \in \mathcal{J} \\ & \text{C4: } 0 \leq f_{k,l} \leq f_k^{\max}, \forall k \in \mathcal{X} \\ & \text{C5: } 0 \leq f_{k,i,l} \leq f_{k,i}^{\max}, \forall i \in \mathcal{J} \\ & \text{C6: } 0 \leq \sum_{k \in \mathcal{X}} f_{k,e} + \sum_{k \in \mathcal{X}} \sum_{i \in \mathcal{J}} u_{k,i} f_{k,i,e} \leq f_e^{\max} \\ & \text{C7: } u_{k,i} \in \{0, 1\} \\ & \text{C8: } \sum_{i \in \mathcal{J}} u_{k,i} \leq 1, \forall k \in \mathcal{X} \\ & \text{C9: } \sum_{k \in \mathcal{X}} u_{k,i} \leq 1, \forall i \in \mathcal{J} \end{aligned} \quad (23)$$

$$(1), (7), (8), (14)$$

其中, C1 和式(1)是对任务设备直接划分任务的约束, C2 和式(8)是对 D2D 设备间接划分任务的约束。式(7)限制了任务设备的发射功率, C3 限制了 D2D 设备的上传功率, C4, C5

和 C6 分别限制了任务设备、D2D 设备和 MEC 服务器的 CPU 频率。式(14)是对最大容忍时延的约束。C7 的取值为:

$$u_{k,i} = \begin{cases} 0, & k \text{ 与 } i \text{ 不建立 D2D 连接} \\ 1, & k \text{ 与 } i \text{ 建立 D2D 连接} \end{cases} \quad (24)$$

为了避免滥用资源, C8 限制任务设备  $k$  最多只能选择一个 D2D 设备建立连接。为了避免资源充分的 D2D 设备成为热点, C9 限制一个空闲设备  $i$  同时最多只能服务于一个设备  $k$ 。

求解优化问题  $\mathcal{P}1$  存在困难。首先, 由于优化目标之间存在耦合, 在对某个目标进行优化时, 有可能会损害其他目标。例如在式(2)中, 可以通过减小  $D_{k,l}$  或者增大  $f_{k,l}$  来降低本地计算时延, 而根据式(1), 减小  $D_{k,l}$  会导致  $D_{k,i}$  或  $D_{k,e}$  增大。式(19)和式(21)则展示了  $D_{k,i}$  或  $D_{k,e}$  的增大会导致费用的增加。式(15)展示了  $f_{k,l}$  的增大会导致本地计算能耗的增加。类似的对立关系还体现在式(6)和式(15)中的  $p_{k,i}$ 、式(9)和式(16)中的  $f_{k,i,l}$  等。其次, 由于优化目标的非凸性,  $\mathcal{P}1$  存在多个局部极小值点, 因此, 避免陷入局部最优是面临的另一项挑战。例如, 考虑  $T_{k,i,e} = \max\{\tau_{k,i,l}^c, \tau_{k,e}, \tau_{k,i}^c + \tau_{k,i,l}^c, \tau_{k,i}^c + \tau_{k,i,e}\}$ , 当且仅当各分段均为凸函数时,  $T_{k,i,e}$  才是凸函数, 而本地计算时延  $\tau_{k,i,l}^c(D_{k,l}, f_{k,l}) = D_{k,l}C_k/f_{k,l}$  的 Hessian 矩阵

$$\begin{bmatrix} 0 & -\frac{C_k}{f_{k,l}^2} \\ -\frac{C_k}{f_{k,l}^2} & \frac{2C_k D_{k,l}}{f_{k,l}^3} \end{bmatrix} \quad (25)$$

非半正定,  $\tau_{k,i,l}^c$  是非凸函数。同理, 本地计算能耗  $E_{loc}^c(D_{k,l}, f_{k,l}) = \delta_k D_{k,l} C_k f_{k,l}^2$  也是非凸函数。此外, 计算费用  $S_{k,e}^{MC}(D_{k,e}, f_{k,e}) = \lambda_k D_{k,e} + \mu_k \frac{D_{k,e} C_k}{f_{k,e}}$  是仿射函数与非凸函数之和, 其合成结果仍是非凸函数。

由于问题的复杂性, 将多目标优化转换为单目标优化的传统算法难以取得较好的效果<sup>[26]</sup>。目前, 基于帕累托最优解(非支配解)的多目标进化算法已经成为多目标优化的主流算法<sup>[27-29]</sup>, 其经典代表有: 快速非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm II, NSGA-II)<sup>[30]</sup>、基于分解的多目标进化算法(Multi-objective Evolutionary Algorithm Based on Decomposition, MOEA/D)<sup>[31]</sup>和基于参考点的非支配排序遗传算法(Non-dominated Sorting Genetic Algorithm III, NSGA-III)<sup>[32]</sup>等。NSGA-II 使用精英保留、快速非支配排序和拥挤度选择算子, 具有运行速度快、收敛性好等优点, 但是拥挤度选择在三维及以上目标空间中的作用不明显, 会导致解的多样性下降。MOEA/D 通过聚合函数将多目标优化问题分解为若干个单目标优化问题, 然后根据邻域关系完成对子问题的协同进化, 其计算复杂度低、搜索能力强, 但邻域的无限替换容易造成种群的多样性缺失<sup>[33]</sup>, 聚合函数的确定也面临挑战。

相比之下, NSGA-III 直接在空间中搜索帕累托最优解, 不存在转换参数和信息丢失等问题, 搜索过程简单直观。其次, 遗传算法的固有特性使得 NSGA-III 的适应性广, 连续变量和离散变量的混合输入不会对算法的性能造成太大影响。最后, NSGA-III 使用空间中均匀分布的参考点引导选择非

支配解, 能够有效保证在高维目标空间中非支配解的广泛分布性和多样性。

综上所述, NSGA-III 在求解优化问题  $\mathcal{P}1$  上存在优势, 因此考虑对其进行适应性改进, 使其适合求解多目标优化问题  $\mathcal{P}1$ 。

## 4 改进 NSGA-III 求解多目标优化问题

### 4.1 NSGA-III 算法改进

对 NSGA-III 的改进主要有两处: 多染色体混合编码、自适应交叉率和变异率。

#### (1) 多染色体混合编码

多染色体编码, 即把逻辑意义不同的待优化变量编码到不同的染色体上; 混合编码, 即连续变量和离散变量共存。在使用多染色体混合编码之后, 逻辑意义不同的染色体之间是相对独立的, 变量类型不同的染色体之间也是相对独立的, 其交叉和变异等操作可以独立进行。

#### (2) 自适应交叉率和变异率

基于文献[34]提出的改进粒子群算法中的余弦递减惯性权重因子, 对交叉率进行动态自适应改进, 其表达式为:

$$P_c = P_c^{\min} + \frac{(P_c^{\max} - P_c^{\min})}{2} \times \left[ 1 + \cos\left(\frac{(g-1)\pi}{g_{\max}-1}\right) \right] \quad (26)$$

其中,  $P_c^{\min}$  是交叉率的最小值,  $P_c^{\max}$  是交叉率的最大值,  $g$  是当前进化代数,  $g_{\max}$  是最大进化代数。取最小交叉率为 0.7, 最大交叉率为 0.9, 最大进化代数为 1000, 改进后的交叉率曲线如图 4 所示。

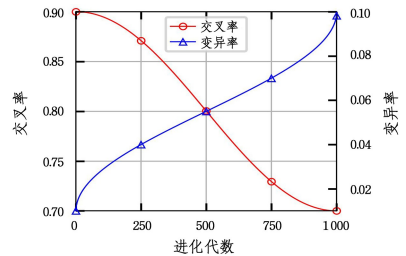


图 4 自适应交叉率和变异率

Fig. 4 Adaptive crossover and mutation rates

基于文献[35]提出的反余弦非线性学习因子改进变异率曲线, 改进后的变异率表达式为:

$$P_m = P_m^{\max} + (P_m^{\min} - P_m^{\max}) \times \left[ 1 - \frac{\arccos\left(\frac{-2g}{g_{\max}} + 1\right)}{\pi} \right] \quad (27)$$

其中,  $P_m^{\min}$  是变异率的最小值,  $P_m^{\max}$  是变异率的最大值,  $g$  是当前进化代数,  $g_{\max}$  是最大进化代数。取最小变异率为 0.01, 最大变异率为 0.1, 最大进化代数为 1000, 改进后的变异率曲线如图 4 所示。

### 4.2 多目标优化问题求解

#### (1) 编码算子

图 5 给出了所使用的多染色体混合编码方案: 个体的染色体数量为  $K+1$  条, 其中, 使用  $K$  条实数编码的染色体对  $K$  个任务设备的连续变量进行编码, 使用 1 条排列编码的染色体对 D2D 设备配对规则进行编码。

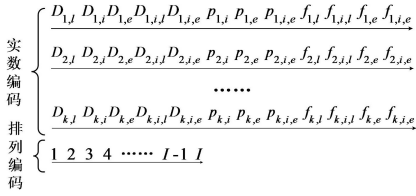


图 5 多染色体混合编码

Fig. 5 Multi chromosome mixed encoding

一个任务设备共有 12 个待优化的实数变量,故在一条实数编码的染色体上有 12 个基因。D2D 配对规则矩阵  $u_{k,i}$  的原始形式包含了  $K \times I$  个 0-1 变量。为了降低编码复杂度,首先将任务设备和 D2D 设备编号,然后将  $u_{k,i}$  编码为共有  $K$  个基因的  $u_k$ ,其取值为:

$$u_k = \begin{cases} -1, & k \text{ 不建立 D2D 连接} \\ i, & k \text{ 与 } i \text{ 建立 D2D 连接} \end{cases} \quad (28)$$

其中,  $k \in [1, K], i \in [1, I]$ 。式(28)的含义是索引为  $k$  的任务设备,与索引为  $i$  的 D2D 设备配对,例如在图 5 中,索引为 1 的任务设备与索引为 1 的 D2D 设备配对,则  $u_1 = 1$ 。在将任务设备和 D2D 设备编号后,只需要保证  $u_k$  中除了元素 -1 允许重复外,其余元素不允许重复,即可保证任务设备和 D2D 设备的一对一关系,从而将空间复杂度从  $O(K \times I)$  降低到了  $O(K)$ 。

### (2) 选择算子

沿用 NSGA-III 基于参考点的非支配排序选择算子。首先,在目标空间生成均匀分布的参考点。其次,计算目标函数的函数值,合并父子两代,并对所有个体进行非支配排序,得到非支配解的分层结构。然后,把位于临界层之前的个体全部放入下一代,并根据参考点,在临界层选择部分个体进入下一代,直到达到种群规模。

### (3) 交叉和变异算子

设  $RI_{g,m,k}$  为第  $g$  代中第  $m$  个个体的第  $k$  条实数编码的染色体,则  $RI_{g,m,k,i}$  表示任务设备  $k$  在该个体上第  $i$  个实数变量的取值,其中  $i \in [1, 12]$  表示从任务分配到功率分配再到 CPU 频率分配的共 12 个实数变量。采用染色体随机配对和模拟二进制交叉的交叉方式,从群体中再随机选择一个个体  $RI_{g,n}$ ,其中  $m \neq n$ ,则  $k$  的实数变量在经过交叉后为:

$$\begin{cases} RI_{g+1,m,k,i} = \frac{1}{2} [(1+\beta_i)RI_{g,m,k,i} + (1-\beta_i)RI_{g,n,k,i}] \\ RI_{g+1,n,k,i} = \frac{1}{2} [(1-\beta_i)RI_{g,m,k,i} + (1+\beta_i)RI_{g,n,k,i}] \end{cases} \quad (29)$$

其中,  $\beta_i$  是分布因子,其表达式为:

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_i+1}}, & u_i \leq 0.5 \\ \left(\frac{1}{2-2u_i}\right)^{\frac{1}{\eta_i+1}}, & u_i > 0.5 \end{cases} \quad (30)$$

其中,  $u_i$  是在  $[0, 1)$  上均匀分布的随机数,  $\eta_i$  是分布指数。

采用多项式变异对交叉后的实数变量进行变异,则在第  $g+1$  代,任务设备  $k$  在第  $m$  个个体上的第  $i$  个实数变量变异为:

$$RI'_{g+1,m,k,i} = RI_{g+1,m,k,i} + \delta_i \quad (31)$$

其中,  $\delta_i$  是扰动因子,其表达式为:

$$\delta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_i+1}} - 1, & u_i < 0.5 \\ 1 - [2(1-u_i)]^{\frac{1}{\eta_i+1}}, & u_i \geq 0.5 \end{cases} \quad (32)$$

其中,  $u_i$  是在  $[0, 1)$  上均匀分布的随机数,  $\eta_i$  是分布指数。

设  $DT_{g,m}$  为第  $g$  代中第  $m$  个个体的使用排列编码的染色体,则  $DT_{g,m,k}$  表示任务设备  $k$  在该个体上选择产生配对的 D2D 设备编号,其中  $k \in [1, K], DT_{g,m,k} \in [1, I]$ 。采用染色体随机配对和部分匹配的交叉方式,从群体中再随机选择一个个体  $DT_{g,n}$ ,其中  $m \neq n$ ,则 D2D 配对规则的交叉方式为:

$$DT_{g+1,m,k} = \begin{cases} DT_{g,m,k}, & k \notin [p, q] \\ DT_{g,n,k}, & k \in [p, q] \end{cases} \quad (33)$$

$$DT_{g+1,n,k} = \begin{cases} DT_{g,n,k}, & k \notin [p, q] \\ DT_{g,m,k}, & k \in [p, q] \end{cases}$$

其中,  $p \leq q$  是随机生成的交叉起始位点和结束位点。由于一个 D2D 设备同时只能与一个任务设备发生配对,如果在交叉之后,一个 D2D 设备发生了重复配对,即:

$$DT_{g+1,m,k} = DT_{g+1,m,j}, k \in [p, q], j \notin [p, q], \quad (34)$$

$$DT_{g+1,m,k} \neq -1$$

则使用如下公式修复配对规则:

$$DT_{g+1,m,j} = DT_{g+1,n,k} \quad (35)$$

如果在修复后仍然存在重复配对,则重复执行上述过程,直到冲突被消除。

D2D 配对规则的变异方式采用逆转变异,如式(36)所示。在随机确定变异片段后,将变异片段中的 D2D 设备序列逆转。

$$DT'_{g+1,m}[p, q] = \text{reverse}(DT_{g+1,m}[p, q]) \quad (36)$$

通过交叉、变异和选择算子的不断迭代,设得到的非支配解集  $S = \{(T^*, E^*, S^*)_n\}$ ,其中  $n$  是非支配解的个数。如果使用基于权值的策略从非支配解集中获取最优解,则最优解  $o$  表示为:

$$o = \arg \min(\vartheta T^* + \varphi E^* + \rho S^*) \quad (37)$$

其中,  $\vartheta + \varphi + \rho = 1$  为权重因子。例如,用户群体的电量充足时可以使用较小的  $\varphi$ ,对时延敏感的用户群体可以使用较大的  $\vartheta$  等。

综上所述,基于 NSGA-III 的改进算法求解多目标优化问题如算法 1 所示。

### 算法 1 改进 NSGA-III 算法求解多目标优化问题

输入:  $g_{\max}, P_{c,\min}, P_{c,\max}, P_{m,\min}, P_{m,\max}, \vartheta, \varphi, \rho$ , 边缘网络环境  
输出: 最优任务卸载策略

1. 将待优化变量混合编码到多条染色体上
2. 初始化群体,初始化进化代数  $g=0$
3. WHILE  $g < g_{\max}$ :
4.  $g = g + 1$
5. 计算群体适应度,使用 NSGA-III 的选择策略选择下一代
6. 根据式(26)调整交叉率
7. 根据式(29)一式(30)和式(33)一式(35)分别对实数编码的染色体和排列编码的染色体执行交叉
8. 根据式(27)调整变异率

9. 根据式(31)、式(32)和式(36)分别对实数编码的染色体和排列编码的染色体执行变异
10. END WHILE
11. 根据式(37)输出最优任务卸载策略,结束

## 5 仿真结果与分析

在实际网络中,各种异构的移动设备拥有不同的计算能力和通信能力,移动设备的芯片制作工艺也各不相同,故大部分仿真参数均在一定范围内随机生成。根据文献[36],在距离上,任务设备与 D2D 设备的距离为 $[1, 20]$  m,移动设备与基站的距离为 $[100, 150]$  m。在计算任务上,任务数为 $[1, 10]$  个,每个任务的数据量为 $[0.1, 0.5]$  Mbit,复杂度服从均值为1000、方差为250的正态分布,最大容忍时延为 $[1, 2]$  s。在信道质量上,移动设备之间的带宽为 $[1, 20]$  MHz,移动设备与基站之间的带宽为 $[1, 5]$  MHz,环境噪声功率为 $10^{-7}$  W。信道增益为 $10^{-3} d^{-3} \phi$ ,其中 $\phi$ 是服从标准正态分布的随机变量, $d$ 是设备间的距离。其余仿真参数设置如表1所列。

表1 参数设置

Table 1 Parameter settings

参数	取值
移动设备最大频率/GHz	$[0.4, 1]$
移动设备最大功率/mW	$[100, 200]$
MEC 服务器最大频率/GHz	12
能耗系数	$[1, 10] \times 10^{-27}$
MEC 流量费率	$[0.03, 0.05]$
MEC 计算费率	$[0.03, 0.05]$
D2D 流量费率	$[0.01, 0.02]$
D2D 计算费率	$[0.01, 0.02]$

为了验证 DCM-PO 模型的有效性,取权重因子 $\vartheta = \varphi = \rho = 1/3$ ,将 DCM-PO 任务卸载方案与其他3种任务卸载方案进行比较。

(1)全 MEC 方案(MEC Only, MECO):任务全部卸载到 MEC 服务器执行。

(2)时延-能耗权衡的部分卸载方案(Delay-Energy Balanced Partial Offloading, DEB-PO):文献[7]提出的部分卸载方案,任务被分成本地部分和 MEC 部分,分别在本地执行和 MEC 服务器执行。

(3)中继选择与资源分配的联合优化方案(Joint Optimization Algorithm of Relay Selection and Resource Allocation, JOSR):文献[36]提出的以 AP 作为中继节点的任务卸载方案,AP 能够对任务进行二次上传,但不具备计算能力。

图6给出了在移动边缘网络中,任务设备的平均任务量对边缘网络总能耗的影响。在任务量较少时,4种方案的能耗差距较小,随着任务量逐渐增多,DEB-PO 和 MECO 的能耗增长速度快于 DCM-PO 和 JOSR 的能耗增长速度。任务量通过影响上传能耗和计算能耗从而影响总能耗。当任务量增加时,上传能耗在总能耗中所占的比例逐渐增大,因此直接上传任务的 DEB-PO 和 MECO 受任务量的影响较大,其次是利用 AP 对任务进行辅助上传的 JOSR。DCM-PO 除了能够利用 D2D 设备进行辅助上传外,还能利用 D2D 设备的计算能力完成一部分任务的计算,减少了上传的任务量,因此能耗增幅较为平缓。

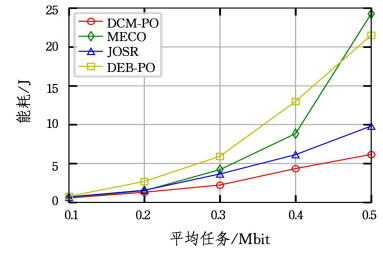


图6 平均任务量与能量消耗

Fig. 6 Average task amount versus energy consumption

图7给出了任务的平均复杂度对边缘网络总时延的影响。4种方案的时延都随着任务复杂度的增加而增加,在复杂度较低时,MECO 的时延最低,DCM-PO 的时延最高,DEB-PO 和 JOSR 的时延介于二者之间;在复杂度较高时,DCM-PO 的时延最低,MECO 的时延最高。任务复杂度主要通过影响计算时延从而影响总时延。在任务复杂度较低时,MECO 方案能够为单个设备提供更高的 CPU 频率,因此时延最短。随着任务复杂度逐渐增加,MEC 服务器能为单个设备提供的 CPU 频率也会逐渐降低,而 DCM-PO 通过联合本地、D2D 设备和 MEC 服务器协同计算,能够保持 CPU 频率的相对稳定供应,因此在任务复杂度较高时,DCM-PO 的时延低于 MECO。

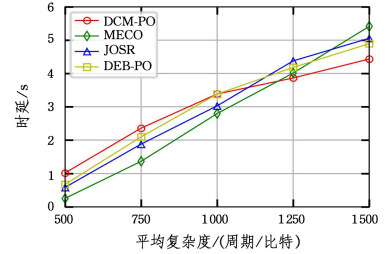


图7 平均任务复杂度与时延

Fig. 7 Average task complexity versus delay

图8给出了边缘网络中任务设备的数量对边缘网络总费用的影响。其中,MECO 的费用最高,其次是 JOSR 和 DEB-PO,DCM-PO 的费用低于其他3种方案。在 MECO 中,任务必须全部上传到 MEC 服务器,因此费用最高。JOSR 和 DEB-PO 允许部分任务在本地执行,减少了上传数据量,因此费用有所降低。JOSR 利用 AP 作为中转,能够把更多任务上传到 MEC 服务器,所以费用略高于 DEB-PO。DCM-PO 能够利用 D2D 设备的计算能力,减少了对 MEC 服务器的依赖,因此费用最低。

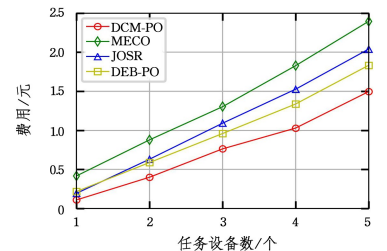


图8 任务设备数与费用

Fig. 8 Number of task devices versus cost

图9给出了边缘网络的吞吐量随着 D2D 配对数增加时

的变化趋势。其中,DCM-PO 的吞吐量明显优于另外 3 种方案,其次是 JOSR 和 DEB-PO,最后是 MECO。4 种任务卸载方案的最大差别在于卸载粒度,MECO 只允许将任务上传到 MEC 服务器执行,其粒度最粗;DEB-PO 允许把任务划分成两部分,分别在本地和 MEC 服务器执行,其粒度更细;而 DCM-PO 可以被视为是在 DEB-PO 的基础上扩展了 D2D 协同功能,其粒度最细;JOSR 允许任务设备使用 AP 作为中继,其粒度介于 DCM-PO 和 DEB-PO 之间。更细粒度的任务划分允许对资源进行更精确的调度和分配,因此具有最细粒度划分的 DCM-PO 在吞吐量上有明显优势。

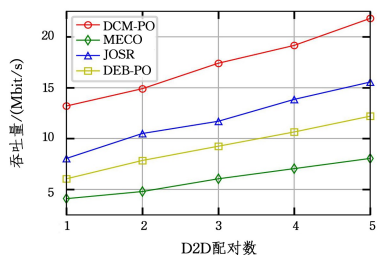


图 9 D2D 配对数与吞吐量

Fig. 9 Number of D2D pairs versus throughput

为了验证权重策略在选择非支配解上的有效性,设置  $\theta + \varphi = 1$  和  $\rho = 0$ ,通过改变  $\theta$  的取值观察 DCM-PO 的时延和能耗变化趋势。图 10 给出了当  $\theta$  的取值逐渐增大时 DCM-PO 的时延和能耗变化趋势。其中,DCM-PO 的时延随着  $\theta$  的增大而减少,能耗随着  $\theta$  的增大而增加。 $\theta$  是时延的权重因子,当  $\theta$  增加时,时延在代价函数中所占的比例也逐渐增加,为了使代价函数尽可能小,时延也必须尽可能小,因此呈现出时延减小的趋势。时延和能耗的变化幅度是不相同的,原因在于时延和能耗是不同维度的性能指标。

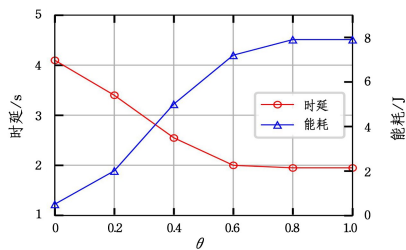


图 10  $\theta$  对时延和能耗的影响

Fig. 10 Influence of  $\theta$  on delay and energy consumption

综上所述,相比基准模型,DCM-PO 模型在任务量增加时能够保持较低的能耗,在处理复杂任务时能够保持较低的时延。随着网络规模的扩大,DCM-PO 能够在降低任务设备费用的同时提高边缘网络的吞吐量。通过权值策略,DCM-PO 能够有效选择非支配解,输出合理的任务卸载方案。

**结束语** 本文提出了一种 D2D 协同下的移动边缘计算模型,通过允许将任务同时卸载到邻近的 D2D 设备和 MEC 服务器的方式来提高用户体验。首先,建立 DCM-PO 的通信模型、计算模型和费用模型,并以最小化边缘网络的能耗、时延和费用为目标建立多目标优化问题。然后,对基于参考点的支配排序遗传算法进行改进,使用多染色体混合编码、余弦递减交叉率和反余弦递增变异率的 NSGA-III 改进算法求解

多目标优化问题。最后,仿真实验验证了模型的有效性。下一步的研究工作包括考虑连接中断问题和使用多跳 D2D 链路进行协同计算等。

### 参考文献

- [1] ABBAS N,ZHANG Y,TAHERKORDI A,et al. Mobile edge computing:a survey[J]. IEEE Internet of Things Journal,2018, 5(1):450-465.
- [2] BANGERTER B,TALWAR S,AREFI R,et al. Networks and devices for the 5G era[J]. IEEE Communications Magazine, 2017,52(2):90-96.
- [3] BUMGARNER J M,LAMBERT C T,HUSSEIN A A,et al. Smartwatch algorithm for automated detection of atrial fibrillation[J]. Journal of the American College of Cardiology,2018, 71(21):2381-2388.
- [4] LIU B,LIU C,PENG M. Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks[J]. IEEE Journal on Selected Areas in Communications,2021,39(4):1015-1027.
- [5] WANG Y C,ZHU H,HEI X H,et al. An energy saving based on task migration for mobile edge computing[J]. Journal on Wireless Communications and Networking,2019(1):133.
- [6] WANG F,XU J,WANG X,et al. Joint offloading and computing optimization in wireless powered mobile edge computing systems[J]. IEEE Transactions on Wireless Communications, 2018,17(3):1784-1797.
- [7] LIU S M,YU Y,GUO L,et al. Adaptive delay-energy balanced partial offloading strategy in mobile edge computing networks [J/OL]. https://doi.org/10.1016/j.dcan.2022.05.029.
- [8] HE Y,REN J,YU G,et al. D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks[J]. IEEE Transactions on Wireless Communications, 2019,18(3):1750-1763.
- [9] CHEN X,ZHANG J S. When D2D meets cloud:hybrid mobile task offloadings in fog computing[C]//2017 IEEE International Conference on Communications. NJ:IEEE,2017:1-6.
- [10] WANG H P,LIN Z P,LV T J. Energy and delay minimization of partial computing offloading for D2D-assisted MEC systems [C]//2021 IEEE Wireless Communications and Networking Conference. NJ:IEEE,2021:1-6.
- [11] JIA Q M,XIE R C,TANG Q Q,et al. Energy-efficient computation offloading in 5G cellular networks with edge computing and D2D communications[J]. IET Communications,2019,13(8): 1122-1130.
- [12] CAO X W,WANG F,XU J,et al. Joint computation and communication cooperation for mobile edge computing[J]. IEEE Internet of Things Journal,2018,6(3):4188-4200.
- [13] WANG C,QIN J H,YANG X X,et al. Energy-efficient offloading policy in D2D underlay communication integrated with MEC service[C]//Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications. NY:ACM,2019:159-164.
- [14] XIAO S R,LIU C B,LI K L,et al. System delay optimization for

- mobile edge computing[J]. *Future Generation Computer Systems*, 2020, 109: 17-28.
- [15] LI G S, WANG J P, WU J H, et al. Data processing delay optimization in mobile edge computing[J]. *Wireless Communications and Mobile Computing*, 2018(5): 1-9.
- [16] SABUJ S R, ASIEDU D K P, LEE K J, et al. Delay optimization in mobile edge computing; cognitive UAV-assisted eMBB and mMTC services[J]. *IEEE Transactions on Cognitive Communications and Networking*, 2022, 8(2): 1019-1033.
- [17] HABER E E, NGUYEN T M, EBRAHIMI D, et al. Computational cost and energy efficient task offloading in hierarchical edge clouds[C]//2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications. NJ: IEEE, 2018: 1-6.
- [18] LIU P C, CHAUDHRY S R, HUANG T, et al. Multi-factorial energy aware resource management in edge networks[J]. *IEEE Transactions on Green Communications and Networking*, 2019, 3(1): 45-56.
- [19] HU H, SONG W, WANG Q, et al. Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network[J]. *IEEE Internet of Things Journal*, 2022, 9(17): 15942-15956.
- [20] LIU M, LIU Y. Price-based distributed offloading for mobile edge computing with computation capacity constraints[J]. *IEEE Wireless Communications Letters*, 2018, 7(3): 420-423.
- [21] HAN D, CHEN W, FANG Y. A dynamic pricing strategy for vehicle assisted mobile edge computing systems[J]. *IEEE Wireless Communications Letters*, 2019, 8(2): 420-423.
- [22] BAHREINI T, BADRI H, GROSU D. Mechanisms for resource allocation and pricing in mobile edge computing systems[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2022, 33(3): 667-682.
- [23] CHEN S, LI L, CHEN Z, et al. Dynamic pricing for smart mobile edge computing; a reinforcement learning approach[J]. *IEEE Wireless Communications Letters*, 2021, 10(4): 700-704.
- [24] KAR U N, SANYAL D K. An overview of device-to-device communication in cellular networks[J]. *Information & Communications Technology Express*, 2018, 4(4): 203-208.
- [25] CHAI R, LIN J L, CHEN M L, 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.
- [26] FENG Q, LI Q, QUAN W, et al. Overview of multiobjective particle swarm optimization algorithm[J]. *Chinese Journal of Engineering*, 2021, 43(6): 745-753.
- [27] LIU J C, LI F, WANG H H, et al. Survey on evolutionary many-objective optimization algorithms[J]. *Control and Decision*, 2018, 33(5): 879-887.
- [28] LIU Z Y, WANG Y J, SUN F L, et al. Ensemble-assisted multi-objective optimization algorithm combining feature perturbation and allocation strategy[J]. *Computer Engineering*, 2022, 48(6): 115-123.
- [29] PAN X T, WANG L P, ZHANG M H. Sparse multi-objective feature selection algorithm based on target vector guiding strategy[J]. *Journal of Chinese Computer Systems*, 2023, 44(10): 2212-2220.
- [30] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [31] ZHANG Q, LI H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [32] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [33] GENG H T, HAN W M, ZHOU S S, et al. MOEA/D algorithm based on new neighborhood updating strategy[J]. *Computer Science*, 2019, 46(5): 191-197.
- [34] CHEN G C, YU J S. Enhanced particle swarm optimization algorithm and its application in soft sensing[J]. *Control and Decision*, 2005(4): 377-381.
- [35] BAI X B, JI X M, HU G. Improved particle swarm optimization method for needle roller bearings under multiple working conditions[J]. *Journal of Computer Aided Design and Graphics*, 2014, 26(10): 1900-1908.
- [36] LI Y, XU G, YANG K, et al. Energy efficient relay selection and resource allocation in D2D-enabled mobile edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(12): 15800-15813.



**WANG Zhihong**, born in 1999, post-graduate. His main research interests include computer network and mobile edge computing.



**WANG Gaocai**, born in 1976, Ph.D., professor, doctoral supervisor. His main research interests include computer network, performance evaluation and network security.

(责任编辑:何杨)