

# 移动边缘网络中基于 QoE 的网络媒体流卸载算法

王再见<sup>1,2</sup>, 程浩<sup>1,2</sup>

(1. 安徽师范大学物理与电子信息学院, 安徽 芜湖 241002;  
2. 安徽省智能机器人信息融合与控制工程研究中心, 安徽 芜湖 241002)

**摘要:** 针对移动边缘计算中新兴网络媒体流业务面临的高时延、高能耗、高带宽、低用户体验质量 (QoE) 等问题, 提出一种基于 QoE 反馈配置卸载 (QFCO) 算法。首先, 联合考虑预处理和优先级划分, 从而最大化网络资源利用率, 并为计算任务赋予不同的权重建立资源分配关系; 然后, 综合考虑截止时间、计算资源、功率和带宽等约束, 以任务时延、能耗和精确度加权和为优化目标建立 QoE 模型, 利用拉格朗日乘法求解。仿真结果表明, 相比深度增强学习在线卸载 (DROO) 算法, 所提算法可有效实现资源的整体优化配置, 更好地提升用户体验质量。

**关键词:** 移动边缘计算; 用户体验质量; 拉格朗日乘法; 网络媒体流; 计算卸载

**中图分类号:** TN393

**文献标志码:** A

**DOI:** 10.11959/j.issn.1000-436x.2024035

## Network media streaming offloading algorithm based on QoE in mobile edge network

WANG Zaijian<sup>1,2</sup>, CHENG Hao<sup>1,2</sup>

1. School of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China

2. Anhui Engineering Research Center on Information Fusion and Control of Intelligent Robot, Wuhu 241002, China

**Abstract:** Aiming at the problems of high-latency, high energy consumption, high bandwidth, and poor quality of experience (QoE) caused by emerging network media streaming business in mobile edge computing, a computing offloading algorithm based on QoE feedback configuration was proposed. Firstly, both preprocessing and priority were comprehensively considered to maximize network resource utilization. Meanwhile, different weights were assigned to the computation tasks for establishing a resource allocation relationship. Secondly, after comprehensively taking into account deadline, computing resource, power and bandwidth constraint, an QoE model was established where the optimization objective was the weighted sum of task delay, energy consumption and precision, and the method of Lagrange multipliers was utilized to solve the established model. Simulation results indicate that, compared with the deep reinforcement learning-based online offloading algorithm, the proposed algorithm can effectively optimize the resource allocation and better improve the QoE.

**Keywords:** mobile edge computing, quality of experience, Lagrange multiplier method, network media streaming, computing offloading

## 0 引言

随着 B5G/6G<sup>[1]</sup>技术的发展, 网络媒体流超高清

(UHD, ultra high-definition) 进程迅速推进。网络媒体流视频质量提高导致数据爆炸式增长, 给传统的传输方法带来巨大挑战<sup>[2]</sup>。面对这一挑战, 移动边

收稿日期: 2023-09-05; 修回日期: 2023-12-05

基金项目: 安徽省自然科学基金资助项目 (No.2008085MF222)

**Foundation Item:** The Natural Science Foundation of Anhui Province (No.2008085MF222)

缘计算 (MEC, mobile edge computing) 技术应运而生。MEC 是一项由欧洲电信标准组织 (ETSI, European Telecommunications Standards Institute) 主导定义并标准化的技术<sup>[3]</sup>, 通过任务卸载到靠近终端网络边缘的服务设备上提供就近服务, 从而减小带宽压力并降低传输时延。MEC 催生出许多新型网络媒体流场景, 如 4K/8K 超高清视频监控、虚拟现实 (VR, virtual reality)、自动驾驶<sup>[4-5]</sup>等, 但 MEC 有限的计算和通信资源成为新型网络媒体流场景中影响用户体验质量 (QoE, quality of experience) 的主要挑战<sup>[6]</sup>, 因此需要设计高效任务卸载和资源分配策略。

近年来, 在云边端协同的三层架构基础上, 网络媒体流场景下的计算卸载已有大量研究工作。终端负责感知数据采集, 边缘负责靠近终端局部数据通信计算, 云端汇集所有的边缘感知数据, 对计算要求大、通信要求低的任务进行计算、存储、分析<sup>[7]</sup>。文献[8]提出可以通过终端传感器上的低复杂度局部二值模式 (LBP, local binary pattern) 算法识别高质量的人脸图像, 而不是卸载到 MEC 服务器, 可以节省计算和带宽资源。文献[9]提出一个经济高效 Spatula 系统, 利用空间和时间的相关性, 忽略与当前位置不相关的帧, 大幅减少通信和计算成本。文献[10]设计一个 RESPIRE 系统过滤器来有效减少空间和时间冗余, 并通过压缩搜索空间来降低不必要的处理成本。文献[11]基于深度学习对象检测算法选择关键帧, 一些非关键帧被选择性丢弃, 有利于以较少带宽进行平滑网络传输。文献[8-11]对媒体流进行预处理, 主要针对时空相关性、检测关键帧和冗余传输几个方面研究, 缺乏考虑通过改变任务配置以适应网络环境, 进一步提高网络资源利用率。

文献[12]提出 VideoEdge 系统, 该系统可以平衡实时视频流的计算资源和准确性, 并通过传输有兴趣的配置来缩小搜索空间。文献[13]设计一种新的网络协议, 考虑视频源大小和计算模型 2 个关于配置的关键参数, 根据无线信道条件和计算负载改变增强现实 (AR, augmented reality) 应用的配置。文献[14]设计一种在线方式调整配置的算法, 该算法动态调整配置文件中的配置和相应的分析间隔, 以捕获多个视频流在边缘的内部变化, 实现高达 25% 的精确度改进。文献[15]提出一种关于李雅普诺夫排队拥塞的高效在线算法, 联合优化配置适应

和带宽分配。文献[12-15]主要基于网络带宽和计算资源来优化配置, 通过优化任务配置减少上传服务器数据来降低带宽压力, 没有将优化配置与目标函数建立联系, 根据计算目标函数的数值来反馈调节任务配置会取得更好的效果。

此外, 机器学习方法也大量用于解决卸载决策问题<sup>[16-18]</sup>。文献[16]提出一种基于深度 Q 网络 (DQN, deep Q network) 的框架聚合和任务卸载方法, 不同的服务数据被划分为具有相似的服务质量 (QoS, quality of service) 需求的队列, 该方法有效地降低了时延和能耗, 提高了吞吐量和总体利用率。文献[17]提出改进深度确定性策略梯度 (DDPG, deep deterministic policy gradient) 算法, 获得具有低计算复杂度的次优解决方案。文献[18]提出深度增强学习在线卸载 (DROO, deep reinforcement learning-based online offloading) 框架, 在快速衰落环境中实现任务低时延处理。文献[19-20]在多目标优化上运用免疫算法和改进的进化算法, 与深度强化学习算法相比, 有更小的复杂度和更快的收敛速度。文献[21]提出一种服务缓存与资源分配联合优化策略, 与其他策略相比时延降低 10% 左右。文献[22]介绍一种为多客户端联合 QoE 优化解决方案, 根据 QoE 公平性, 对不同优先级的用户进行差异化服务。文献[23]设计面向超低时延传输应急物联网的多切片网络架构, 有效兼顾资源利用率和切片差异化服务质量要求。文献[22-23]考虑了根据网络切片进行差异化服务, 灵活的切片可根据不同的优先级匹配任务, 但是未能充分根据业务特征预处理确定优先级、差异化分配网络频谱资源。

上述文献分别从边缘预处理、卸载配置和常用卸载方法多个角度对现有工作进行了介绍, 可以看到, 目前对于边缘计算卸载策略的研究较为成熟, 在网络媒体流场景下, 根据 QoS 指标 (例如时延、能耗) 建立成本函数优化计算卸载的文献较多, 而面向 5G/B5G 场景的新业务并根据 QoE 目标函数进一步反馈调节媒体流任务配置的研究仍然存在不足。针对该问题, 考虑在网络媒体流场景下结合中心云、边缘服务器和终端三层架构模型, 通过将卸载策略和信道质量结合, 以突出计算卸载策略和根据网络带宽更新配置之间的关系, 本文提出一种基于 QoE 的反馈配置卸载算法。本文主要研究工作如下。

1) 综合考虑计算卸载时延、能耗和精确度多目标优化, 建立 QoE 模型目标函数。根据任务多方面

特征计算权重，设计一个切片优化算法，通过优先级为任务匹配合适的资源。

2) 使用凸优化求解方法获取最佳资源分配策略，在给定的计算条件下求解拉格朗日方程，提出基于 QoE 的反馈配置卸载算法，得到最优的资源配置分配策略。

3) 仿真结果表明，与 DROO、DDPG、DQN、Greedy 等算法对比，本文算法能够提高资源利用率，更好地提升 QoE。

### 1 系统模型

边缘计算系统中基于 QoE 的网络媒体流业务卸载模型如图 1 所示。系统参数如表 1 所示。由于计算结束的任务数据很小，因此本文忽略从卸载服务器返回终端的下行过程。考虑一个 5G 单小区上行链路卸载过程，基于 QoE 的网络媒体流卸载模型由一个云服务器、一个 MEC 服务器和一个基站(BS, base station) 组成，边缘服务器上配有不同卷积层数的 CNN 模型，用来处理不同配置视频帧的切片。在边缘服务器下有  $N$  个终端， $\mathcal{N} = \{1, 2, \dots, n, \dots, N\}$  为终端集合， $M$  个中继边缘节点， $\mathcal{M} = \{1, 2, \dots, m, \dots, M\}$  为中继边缘节点集合。终端任务的集合为  $\mathcal{I} = \{1, 2, \dots, i, \dots, I\}$ ，终端  $n(n \in \mathcal{N})$  均有  $i(i \in \mathcal{I})$  个待处理的计算密集型任务，以  $\{D_{n,i}, C_{n,i}, T^{\max}, O_{n,i}\}$  来描述任务  $T_{n,i}$  的特征，其中， $D_{n,i}$  为数据大小， $C_{n,i}$  为计算复杂度， $T^{\max}$  为最大容忍时延， $O_{n,i}$  为识别目标数目。根据优先级排序，为视频帧分配频谱资源不同的切片。边缘服务器上

配有不同卷积层数的 CNN 模型，具有较少卷积层的 CNN 模型更快但不太准确；具有较多卷积层的 CNN 模型更慢但是准确度较高。假设本文中边缘服务器存储未处理任务的缓存足够大。

表 1 系统参数

参数	含义
$N$	终端数量
$M$	中继边缘节点数量
$D_{n,i}$	数据大小
$C_{n,i}$	计算复杂度
$T^{\max}$	最大容忍时延
$O_{n,i}$	识别目标数目
$h_{n,i}^j$	终端和边缘服务器之间的信道增益
$\delta^2$	噪声功率
$I_{n,i}^j$	信道上终端设备引起的信噪比
$B$	传输带宽
$P^T$	传输功率
$f_{n,i}^j$	分配给任务的计算资源
$r_{k,i}$	分辨率对应的精确度函数
$f_{k,i}$	帧率对应的精确度函数

#### 1.1 识别目标数目

虽然终端的计算和存储资源极其有限，然而，终端离媒体流源头最近，在低时延快速分析和视频流预处理方面具有不可替代的优势，因此本文在终端调用应用程序接口 (API, application programming interface) 计算视频帧的目标识别数目。

如图 2 所示，终端对实时媒体流  $S$  进行预处理。

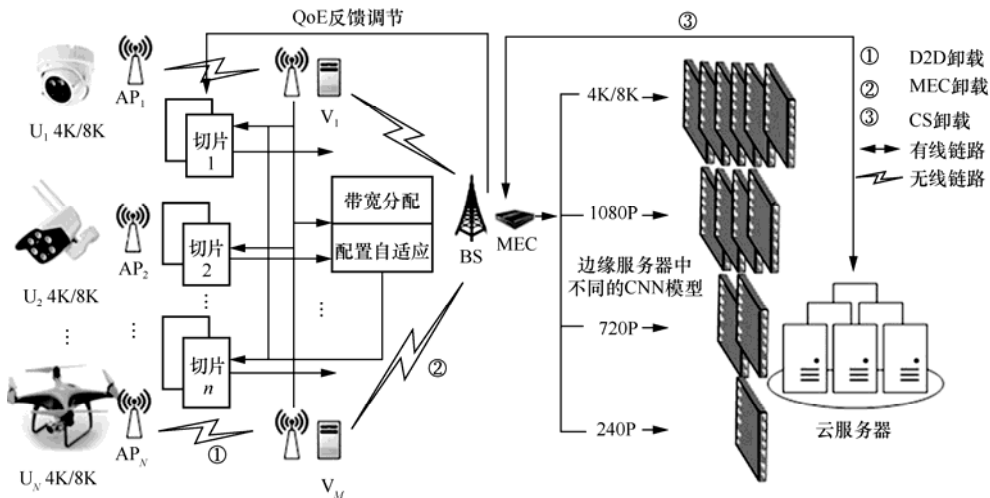


图 1 基于 QoE 的网络媒体流业务卸载模型

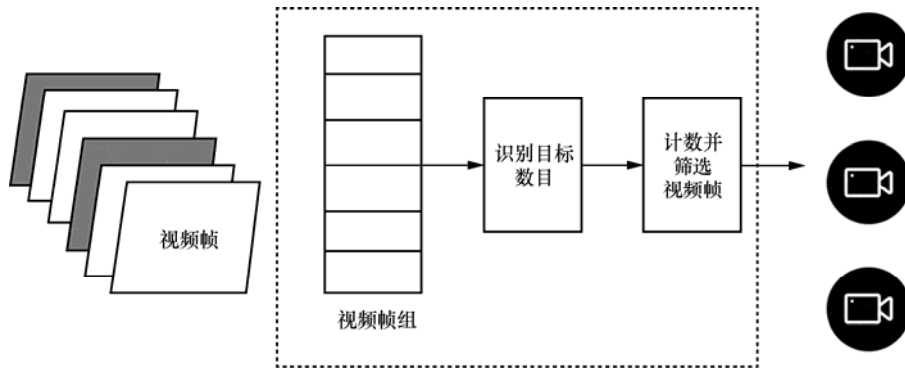


图 2 媒体流边缘预处理

首先，将所采集到的媒体流时长间隔设置为  $T$ ，每隔时长  $T$  形成一个视频帧组，并调用终端封装的 API 来识别视频帧中的目标数目，然后将每个任务所包含的目标数量  $O_i$  返回任务中，用于切片优先级算法的权重计算。

### 1.2 切片优先级算法

当网络带宽资源充足时，切片将媒体流卸载传输到边缘服务器的不同 CNN 模型上；当网络带宽资源不足时，4K/8K 媒体流传输是个棘手的问题。若对不同需求的任务采取相同的卸载策略，将会导致资源的不合理分配，从而影响终端体验。因此，任务卸载传输前，根据业务的属性对任务进行优先级划分，有利于资源的合理分配并有效解决网络带宽不足的问题。

对于不同需求的请求任务，其卸载传输过程所需的计算、通信资源都有所不同，所以需要一种能根据任务特征属性来划分优先级的方法。层次分析法 (AHP, analytic hierarchy process) 是一种根据任务特征来计算权重、划分优先级、解决多指标复杂决策问题的方法。本文选择任务识别目标数目、计算复杂度、最大容忍时延、数据大小作为评价因素，根据评价因素结果大小占比来计算每个任务的权重，权重越大的任务分配的计算、通信资源越多。

对于层次分析法的建模，大致可分为以下步骤。首先，同一层次元素关于上一层次元素的重要性进行两两比较，并构造判断矩阵。然后，由判断矩阵计算被比较元素对该准则的相对权重，并进行一致性检验。最后，填充权重矩阵，并根据权重矩阵计算得出归一化结果。本文的 4 个评价因素中，识别目标数目的重要程度最高，计算复杂度第二，

最大容忍时延第三，数据大小最低。将评价因素判断矩阵设置为  $A = (a_{xy})_{4 \times 4}$ ，目标层相对于准则层的判断矩阵设置为  $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_k = (a_{xy})_{z \times z}$ 。

$$a_{xy} = \begin{cases} \frac{1}{a_{yx}} = n, & x \neq y \\ 1, & x = y \end{cases} \quad (1)$$

其中， $x$  表示终端  $U_n$  的第  $x$  个特征评价因素。

计算出评价因素判断矩阵  $A$  权重向量矩阵  $\Omega = [\Omega^1, \Omega^2, \Omega^3, \Omega^4]$ ，其中权重向量元素为

$$\Omega^x = \frac{\sqrt[4]{\prod_{j=1}^4 a_{xy}}}{\sum_{i=1}^4 \sqrt[4]{\prod_{j=1}^4 a_{xy}}} \quad (2)$$

同理，求得目标层相对于准则层的判断矩阵  $\mathbf{Y}_n$

对应权重向量元素为

$$U_k^{in} = \frac{\sqrt[k]{\prod_{j=1}^k a_{xy}}}{\sum_{i=1}^k \sqrt[k]{\prod_{j=1}^k a_{xy}}} \quad (3)$$

其中， $k$  表示第  $k$  个终端。将式(3)中所有判断矩阵  $\mathbf{Y}_k$  对应权重向量元素组合起来填充矩阵，可得所有终端任务权重向量矩阵为

$$U = \begin{pmatrix} u_1^1 & \dots & u_1^4 \\ \vdots & \ddots & \vdots \\ u_k^1 & \dots & u_k^4 \end{pmatrix} \quad (4)$$

将矩阵  $U$  和  $\Omega$  相乘形成一个新矩阵，通过一

致性检验后, 得到权重向量  $\mathbf{W}$ , 其中每一个元素分别代表终端计算任务权重, 其计算式为

$$\mathbf{W} = \mathbf{U}\mathbf{\Omega} = \begin{pmatrix} \sum_{i=1}^4 U_1^i \mathbf{\Omega}^1 \\ \sum_{i=1}^4 U_2^i \mathbf{\Omega}^2 \\ \vdots \\ \sum_{i=1}^4 U_k^i \mathbf{\Omega}^i \end{pmatrix} \quad (5)$$

对权重  $\mathbf{W}$  进行归一化, 有

$$\tilde{\mathbf{W}} = \frac{w_i}{\sum_{k=1}^K w_k} \quad (6)$$

边缘服务器分配给任务  $i$  的频谱资源  $B_i$  为

$$B_i = \tilde{w} B \quad (7)$$

其中,  $B$  为总带宽, 频谱分配集合为  $B_{\text{ser}} = \{B_1, \dots, B_n, \dots, B_N\}$

$$\sum_{i \in \mathcal{I}} x_{i,n} B_i \leq B \quad (8)$$

**算法 1** 切片优先级算法

**输入** 终端数  $N$ , 用户终端  $U_n, n \in \mathcal{N}$ , 所有任务集  $M$ , 任务  $i$  的频谱资源  $B_i$ , 网络总带宽  $B$

**输出** 分配带宽  $B_{\text{ser}} = \{B_1, \dots, B_n, \dots, B_N\}$

**开始**

- 1) 根据式(2)计算判断矩阵  $U_k^i$  对应权重向量
- 2) 根据式(3)得到所有终端任务权重矩阵
- 3) 根据式(4)评价因素矩阵  $A$  权重向量
- 4) 根据式(5)终端计算任务权重
- 5) 根据式(6)对  $\mathbf{W}$  进行归一化
- 6) 根据式(7)计算分配给任务  $i$  的频谱资源  $B_i$

**结束**

### 1.3 任务卸载

每个任务既可以选择在本地执行, 也可以卸载到边缘节点、边缘服务器或云服务器, 因此引入卸载决策变量  $X = \{x_{n,i}^j\}$  表示任务卸载的情况

$$x_{n,i}^j = \begin{cases} 0, & \text{任务在本地执行} \\ 1, & \text{任务卸载执行} \end{cases} \quad (9)$$

#### 1.3.1 中继边缘节点卸载

当任务卸载到中继边缘节点时, 数据传输速率

(单位为 bit/s) 可表示为

$$r_{n,i}^v = B \text{lb} \left( 1 + \frac{P^v h_{n,i}^v}{\delta^2} \right) \quad (10)$$

其中,  $h_{n,i}^v$  是终端  $n$  与中继边缘节点  $v$  之间的信道增益,  $P^v$  是中继车传输功率,  $\delta^2$  是信道的噪声功率<sup>[24]</sup>。

卸载到中继边缘节点所需的传输时延和传输能耗分别为

$$T_{n,i}^v = \frac{D_{n,i}}{r_{n,i}^v} \quad (11)$$

$$E_{n,i}^v = P^v T_{n,i}^v = P^v \frac{D_{n,i}}{r_{n,i}^v} \quad (12)$$

若任务卸载到边缘服务器  $j$  上, 数据的传输速率为

$$r_{n,i}^j = B \text{lb} \left( 1 + \frac{P^M h_{n,i}^j}{\delta^2} \right) \quad (13)$$

其中,  $h_{n,i}^j$  为终端  $n$  与边缘服务器  $j$  之间的信道增益。

#### 1.3.2 MEC 卸载

若任务卸载到 MEC 服务器上, 则所需的传输时间和传输能耗分别为

$$T_{n,i}^j = \frac{D_{n,i}}{r_{n,i}^j} \quad (14)$$

$$E_{n,i}^j = P^M T_{n,i}^j = P^M \frac{D_{n,i}}{r_{n,i}^j} \quad (15)$$

其中,  $P^M$  是终端与边缘服务器之间的传输功率。

#### 1.3.3 云卸载

当任务卸载到云服务器时, 因为云服务器计算资源非常大, 所以本文忽略任务在云服务器上的计算时延。任务先通过无线链路从终端卸载到边缘服务器, 再从边缘服务器通过回程链路卸载到云服务器。假设从边缘服务器到云服务器的传输时间和能耗分别用  $T_c$  和  $E_c$  表示, 其中  $T_c$  和  $E_c$  为常数, 则边缘服务器到云服务器的速率被认为是常数, 且可通过实际测量获得<sup>[25]</sup>。从终端卸载到云服务器的时间和能耗分别为

$$T_{n,i}^C = T_{n,i}^j + T_c \quad (16)$$

$$E_{n,i}^C = E_{n,i}^j + E_c \quad (17)$$

### 1.4 任务处理

由于计算结束后的任务数据很小，因此忽略从服务器卸载返回终端的时间和能耗。假设所有终端随机均匀分布在基站旁，距离服从均匀分布  $L \sim \text{Unif}([l_{\min}, l_{\max}])$ <sup>[26]</sup>，边缘服务器  $j$  的计算资源定义为  $F_j$ ，记  $f_{n,i}^j$  为分配给任务  $T_{n,i}$  的计算资源，同一个计算实体上可以卸载多个任务，但所有任务计算资源总和需小于  $F_j$ ，即  $\sum_{n,i \in \mathcal{N}} x_{n,i}^j f_{n,i}^j \leq F_j$ 。

任务的计算时间和能耗可分别表示为

$$T_{n,i}^j(\text{co}) = \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} \quad (18)$$

$$E_{n,i}^j(\text{co}) = \delta_{n,i}^L (f_{n,i}^j)^2 c_{n,i} \quad (19)$$

其中， $\delta_{n,i}^L$  为终端设备  $n$  的有效电容系数，取决于 CPU 的芯片架构。

令  $\sum_{n=1}^N \sum_{i=1}^I T_{n,i}(\text{tr}) = T_{n,i}^V(\text{tr}) + T_{n,i}^j(\text{tr}) + T_{n,i}^C(\text{tr})$  表示

任务的传输总时延， $\sum_{n=1}^N \sum_{i=1}^I T_{n,i}(\text{co})$  表示计算总时延，同理  $\sum_{n=1}^N \sum_{i=1}^I E_{n,i}(\text{tr}) = E_{n,i}^V(\text{tr}) + E_{n,i}^j(\text{tr}) + E_{n,i}^C(\text{tr})$  表示

传输总能耗， $\sum_{n=1}^N \sum_{i=1}^I E_{n,i}(\text{co})$  表示计算总能耗。

总时延由传输和计算两部分组成，可表示为

$$T = \sum_{n=1}^N \sum_{i=1}^I (T_{n,i}(\text{tr}) + T_{n,i}(\text{co})) = \sum_{n=1}^N \sum_{i=1}^I (T_{n,i}^V(\text{tr}) + T_{n,i}^j(\text{tr}) + T_{n,i}^C(\text{tr}) + T_{n,i}^j(\text{co})) \quad (20)$$

同理，任务总能耗可表示为

$$E = \sum_{n=1}^N \sum_{i=1}^I (E_{n,i}(\text{tr}) + E_{n,i}(\text{co})) = \sum_{n=1}^N \sum_{i=1}^I (E_{n,i}^V(\text{tr}) + E_{n,i}^j(\text{tr}) + E_{n,i}^C(\text{tr}) + E_{n,i}^j(\text{co})) \quad (21)$$

### 1.5 精确度模型

因为视频精确度对终端体验起到了至关重要的作用，所以本文将精确度与时延、能耗这 3 个因素联合考虑到 QoE 模型中。

令  $\varepsilon_i^t(r)$  表示在时隙  $t$  用户  $n_i$  分辨率所对应的

精确度函数， $x_{n,i}^j(t)r_{n,i}$  表示在时隙  $t$  任务  $i$  的帧分辨率；帧率与精确度函数建模为凹函数  $\phi_i^j(f_{n,i}^j(t))$ ，

$f_{n,i}^j(t)$  表示时隙任务  $i$  的帧率。帧分辨率和帧率独立影响精确度，本文将时隙  $t$  中  $n_i$  精确度建模<sup>[15]</sup>为

$$A = \frac{1}{i} \sum_{i=1}^I \varepsilon_i^t \left( \sum_{i=0}^N x_{n,i}^j(t)r_{n,i} \right) \phi_i^j(f_{n,i}^j(t)) \quad (22)$$

然而，时延、能耗和精确度是需要平衡的因素，无法同时减少。因此本文将目标函数 QoE 定义为时延、能耗和精确度的加权和，最大化网络媒体流中 QoE，以寻求 3 个因素的平衡点。在整个媒体流卸载传输过程中，用 3 个方面的服务质量指标来共同构成 QoE 评估函数，第一部分是服务时延，通过式(20)计算卸载过程中任务的总时延，当  $T$  上升时，任务对低服务时延有更大的需求；第二部分是能耗，通过式(21)可计算卸载过程中任务的总能耗，但是当  $E$  过高时，用户会尽量将卸载到边缘服务器的任务减少，更多地安排在本地产地执行；第三部分是视频帧的精确度，精确度受到分辨率和帧率独立影响，用户的精确度  $A$  越高，卸载到服务器的任务越多。利用参数来均衡时延、能耗和精确度，将 QoE 表示为

$$\text{QoE} = \alpha T + \beta E + \eta A \quad (23)$$

其中， $\alpha$ 、 $\beta$ 、 $\eta$  分别是任务  $T_{n,i}$  的时延、能耗和精确度权重占比，且  $\alpha + \beta + \eta = 1$ 。

本文的目标是在有限的计算资源和带宽资源条件下，根据网络信道状态，来反馈调节资源配置，使目标函数 QoE 最大化，目标函数最大化问题为

$$\begin{aligned} \text{P1: } & \max_{x_{n,i}^j, f_{n,i}^j} \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}} \text{QoE} \\ \text{s.t. C1: } & T_{n,i}^j \leq T^{\max} \\ & \text{C2: } x_{n,i}^j \in \{0, 1\} \\ & \text{C3: } \sum_i x_{n,i}^j f_{n,i}^j \leq F_j \\ & \text{C4: } f_{n,i}^j \geq 0 \\ & \text{C5: } \sum_n \sum_i x_{n,i}^j B_{n,i} \leq B^{\max}, \forall i \in \mathcal{I}, \forall n \in \mathcal{N} \\ & \text{C6: } \sum_n \sum_i x_{n,i}^j p_{n,i}^j \leq P^M, \forall i \in \mathcal{I}, \forall n \in \mathcal{N} \\ & \text{C7: } \sum_{t=0}^T x_{n,i}^t, t \in \mathcal{T} = \{1, \dots, T\} \end{aligned} \quad (24)$$

其中，C1 保证每个任务的完成时间在  $T^{\max}$  范围内，

C2 保证每个任务只能卸载到中继车、边缘服务器或者云服务器中的一个, C3 保证任务的计算资源总和不能超过总的计算资源, C4 保证每个任务的计算资源大于 0, C5 是带宽约束, C6 是功率约束, C7 保证每个时隙中任务只能选择一个 CNN 模型。

## 2 基于 QoE 反馈配置卸载算法

### 2.1 问题分析

P1 是一个混合整数非线性规划问题<sup>[27]</sup>, 卸载变量  $x_{n,i}^j$  和资源分配变量  $f_{n,i}^j$  是解决问题的主要障碍, 为更好地解决问题 P1, 需要一种高效卸载方法, 根据信道状态信息优化调整配置。为解决此问题, 本文提出基于 QoE 反馈配置卸载算法, 将原 NP-hard 问题转换为确定性优化问题, 并将原问题分成资源分配和反馈配置, 分别求解资源分配  $f_{n,i}^j$  和卸载变量  $x_{n,i}^j$ 。

#### 2.1.1 资源分配问题

首先对卸载决策问题进行建模, 在计算卸载对被固定的情况下, P1 可转化为 P2 进行求解

$$\begin{aligned} \text{P2: } & \max_{x_{n,i}^j, f_{n,i}^j} \sum_{n \in N} \sum_{i \in I} \text{QoE} \\ \text{s.t. } & \text{P1 中的 C1, C3~C7} \end{aligned} \quad (25)$$

P2 中卸载对被固定, 忽略 C2 约束。通过固定卸载对减小动作空间, 降低复杂度。

根据附录 1 证明子问题 P2 为凸优化问题, 因此目标函数存在最优解, 根据 C3 约束可知, 该问题是在边缘服务器资源范围区间内求解目标函数极大值点, 而拉格朗日乘法是解决凸优化极大值问题的有效方法。基于 KKT (Karush-Kuhn-Tucker) 条件<sup>[28]</sup>资源分配的拉格朗日方程为

$$\begin{aligned} L(f, v, \mu, \gamma) = & \sum_{n \in N} \sum_{i \in I} \frac{(1 + P^T \varepsilon_{n,i}) D_{n,i}}{r_{n,i}^j} + \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} + \varepsilon_{i_n} p_j f_{n,i}^j + \\ & \frac{1}{i} \sum_{i=1}^I (\varepsilon_i^t x_{n,i}^j(t) r_{n,i}) \phi_k^t(f_{n,i}^j(t)) + \\ & \sum_{n \in N} \sum_{i \in I} v_{n,i}^j \left( \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} + \frac{D_{n,i}}{r_{n,i}^j} - T_{n,i}^{\max} \right) + \\ & \left( \sum_{n \in N} \sum_{i \in I} f_{n,i}^j - F_j \right) + \sum_{n \in N} \sum_{i \in I} \gamma_{n,i}^j (A_{i_n} - A_{\max}) \end{aligned} \quad (26)$$

其中,  $v_{n,i}^j$ 、 $\mu_{n,i}^j$  和  $\gamma_{n,i}^j$  为非负的拉格朗日乘子。

根据 KKT 条件, 可得

$$\frac{\partial L}{\partial f_{n,i}^j} = \varepsilon_{n,i} p_j + \mu_{n,i}^j - \frac{(1 + v_{n,i}^j) D_{n,i} C_{n,i}}{(f_{n,i}^j)^2} \quad (27)$$

$$v_{n,i}^j \left( \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} + \frac{D_{n,i}}{r_{n,i}^j} - T_{n,i}^{\max} \right) = 0 \quad (28)$$

$$\mu_{n,i}^j \left( \sum_{n \in N} f_{n,i}^j - F_j \right) = 0 \quad (29)$$

$$\gamma_{n,i}^j (A_{n,i} - A_j) = 0 \quad (30)$$

求解上述方程, 可得  $v_{i_n, j}$ 、 $\mu_{i_n, j}$  和  $\gamma_{i_n, j}$  的值, 将最优计算资源定义为

$$(f_{n,i}^j)^* = \left[ \sqrt{\frac{(1 + v_{n,i}^j) D_{n,i} C_{n,i}}{\varepsilon_{n,i} p_j + \mu_{n,i}^j + \gamma_{n,i}^j}} \right]^+ \quad (31)$$

其中,  $[a]^+ = \max\{0, a\}$ ,  $v_{n,i}^j$ 、 $\mu_{n,i}^j$ 、 $\gamma_{n,i}^j$  按照式(32)~式(34)更新

$$v_{n,i}^j(t+1) = \left[ v_{n,i}^j(t) - \Delta(t) \left( \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} + \frac{D_{n,i}}{r_{n,i}^j} - T_{n,i}^{\max} \right) \right]^+ \quad (32)$$

$$\mu_{n,i}^j(t+1) = \left[ \mu_{n,i}^j(t) - \Delta(t) \left( \sum_{n \in N} f_{n,i}^j - F_j \right) \right]^+ \quad (33)$$

$$\gamma_{n,i}^j(t+1) = \left[ \gamma_{n,i}^j(t) - \Delta(t) \left( \sum_{n \in N} A_{n,i} - A \right) \right]^+ \quad (34)$$

其中,  $\Delta(t)$  代表迭代的步长。

#### 2.1.2 反馈配置问题

根据目标函数在每个时隙内计算的 QoE 值, 来反馈调节图像压缩因子和计算模型。

根据目标函数在时隙  $t$  计算的 QoE 值, 来反馈调节时隙  $t+1$  图像压缩因子和计算模型。  $d_{n,i}^j$  和  $l_{n,i}^j$  分别代表图像压缩因子和 CNN 模型, 使用  $l_{n,i}^j$  的值表示计算模型, 给定一个  $l_{n,i}^j$  的值, 则对应一个 CNN 模型。  $d_{n,i}^j$  越大, 数据越小, 则 QoE 越小, 所以  $d_{n,i}^j$  与 QoE 为负相关关系;  $l_{n,i}^j$  越大, 分辨率越大, 则

QoE 越大, 所以  $l_{n,i}^j$  与 QoE 为正相关关系, 因此本文采用循环块坐标梯度投影<sup>[29]</sup>方法来求解反馈配置问题。根据该方法, 本文通过固定 3 个变量中的 2 个并推导出剩下的一个变量来求解。迭代此过程, 直到每个变量的值收敛。

$\nabla y(x)$  为与变量  $x$  对应的函数  $y$  的偏导数, 定义  $P_{\Omega}(x) = \arg \min_{y \in \Omega} \|x - y\|^2$  表示  $x$  在  $\Omega$  上的欧几里得投影。该解决方案的步骤可总结如下。

给定  $l_k$  和  $f_k$ , 更新  $d_k$

$$d_k^{(j+1)} = P_{\Omega_d} \left[ d_k^j - \Delta(t) \alpha_k \text{QoE}(d_k^j) \right]^+ \quad (35)$$

其中,  $\alpha_k > 0$  是一个恒定的步长,  $\Omega_d$  是受约束的有界域,  $[a]^+ = \max\{0, a\}$ 。

给定  $d_k$  和  $f_k$ , 更新  $l_k$

$$l_k^{(j+1)} = P_{\Omega_l} \left[ l_k^j - \Delta(t) \beta_k \text{QoE}(l_k^j) \right]^+ \quad (36)$$

其中,  $\beta_k > 0$  是一个恒定的步长,  $\Omega_l$  是受约束的有界域。

## 2.2 算法设计

基于 QoE 反馈配置卸载算法如算法 2 所示。

**算法 2** 基于 QoE 反馈配置卸载算法

**步骤 1** 初始化变量  $T_i^{\max}$ 、 $F_i^{\max}$  和拉格朗日乘子  $v_{n,i}^j$ 、 $\mu_{n,i}^j$  和  $\gamma_{n,i}^j$ ;

**步骤 2**  $l_{n,i}^j \leftarrow l_{\min}, \forall i \in \mathcal{I}$ ;

**步骤 3**  $x_{n,i}^j \leftarrow 1$ ;

**步骤 4** 通过式(31)计算  $(f_{n,i}^j)^*$ ;

**步骤 5** 通过式(35)和式(36)分别计算  $d_k^{j+1}$  和  $l_k^{j+1}$ ;

**步骤 6** 通过式(32)~式(34)更新拉格朗日乘子;

**步骤 7** 输出  $d_k^{j+1}, l_k^{j+1}$

## 3 仿真分析

### 3.1 实验设置

为了验证本文算法的有效性, 本节对 QFCO 算法、Greedy<sup>[8]</sup>算法、DQN<sup>[16]</sup>算法、DDPG<sup>[17]</sup>算法、DROO<sup>[18]</sup>算法的时延、能耗和分辨率随数据大小和服务器计算能力的变化情况进行对比, 仿真中分辨

率代表视频帧的分辨率, 其对应参数分别为 240P(320×240)、360P(480×360)、480P(640×480)、720P(1280×720)、1080P(1920×1080)、4K(3840×2160)。本文实验中制定的详细仿真参数如表 2 所示。

表 2 详细仿真参数

参数	数值
视频帧帧率/FPS	30
本地终端计算能力/GHz	[0.5,1.0]
服务器计算能力/GHz	[2,14]
通信带宽/MHz	5
任务数据大小/KB	[100,400]
传输功率/mW	100
信道增益	$10^{-6}$

### 3.2 仿真结果

为了实验方便起见, 实验参数中令视频帧的帧率为 30 FPS, 在帧率相同的情况下, 考虑精确度对 QoE 模型的影响, 即分辨率对 QoE 模型的影响, 不同度量组合方案得到的 QoE 模型性能比率如图 3 所示。由图 3 可知, 获得 QoE 比率最高值的是 3 个参数组合而成的算法, 这意味着同时具有 3 个参数的 QoE 模型性能优于 2 个参数的 QoE 模型, 考虑分辨率的 QoE 模型优于不考虑分辨率的 QoE 模型。当不考虑能耗、时延或分辨率时, QoE 比率几乎相同。原因是时延、能耗和分辨率在卸载过程中对 QoE 影响占比相差不大。然而, 当同时考虑时延和能耗时, QoE 模型的性能仅次于联合优化。结果表明, 3 个指标的组合比 2 个指标的组合取得更好的性能。在 500 时隙片段, 时延和分辨率组合的 QoE 比率折线在能耗和分辨率组合的 QoE 比率折线上方; 在 500 到 700 时隙片段之间, 时延和分辨率组合的 QoE 比率折线与能耗和分辨率组合的 QoE 比率折线的关系发生变化, 原因是具体的任务时延和能耗发生变化, 参考 QoE 计算式(23), 每个时刻每个任务的时延和能耗的数值大小都是随机的, 在 500 到 700 时隙片段之间的折线变化是任务的时延和能耗产生了变化, 导致 QoE 数值产生变化, 进一步引起图 3 中折线之间比率关系的变化。

图 4~图 6 分别展示了本文算法与 Greedy、DQN、DDPG 和 DROO 算法的时延、能耗和分辨率随数据大小的变化情况。

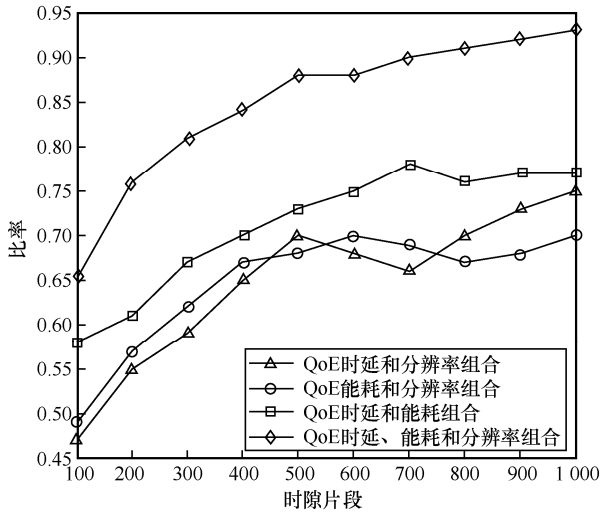


图 3 QoE 模型表现

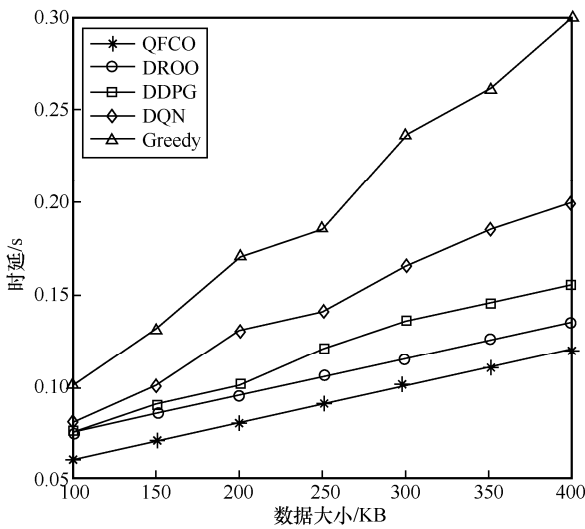


图 4 时延随数据大小的变化情况

由图 4 可知，随着数据大小的增大，时延也跟着增加。在上传速度不变的情况下，数据越大时延越大；并且，随着数据的增大，会给带宽带来上传压力，网络会变得更加拥塞，进一步加速时延的增加。从图 4 中可知，相比于 Greedy、DQN、DDPG 和 DROO 算法，QFCO 获得最优性能。当数据大小设置为数据取值范围平均值 250 KB 时，选取本文算法与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，时延缩短 5.56%；同时 QFCO 优于 DDPG，时延缩短 10.35%。综上所述，本文算法在时延方面取得了最优的性能。

由图 5 可知，随着数据大小的增大，能耗也跟着增大；在上传速度不变的情况下，数据越大能耗越大；并且，随着数据的增大，会给带宽带来上传压力，网络会变得更加拥塞，进一步加速能

耗的增加；相比其他算法，本文算法获得最优性能，优于 Greedy、DQN、DDPG 和 DROO。当数据大小设置为数据取值范围平均值 250 KB 时，选取本文算法与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，能耗降低 6.25%；同时 QFCO 优于 DDPG，能耗降低 13.75%。

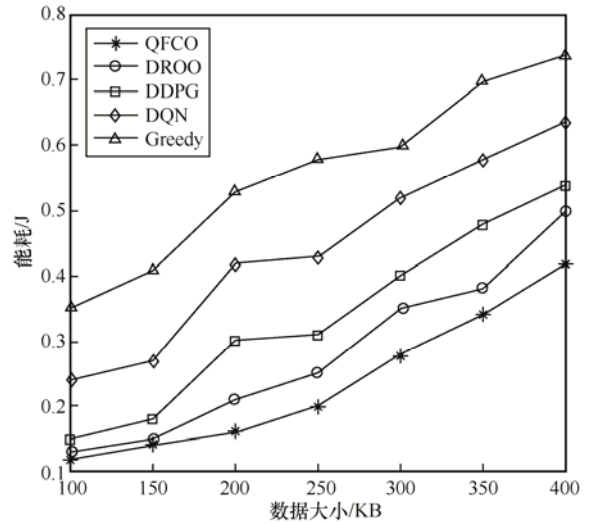


图 5 能耗随数据大小的变化情况

由图 6 可知，随着数据大小的增大，分辨率越来越小。因为在数据增大的情况下，边缘服务器的计算和带宽压力增大，其处理任务能力下降，分辨率随之下降。相比 Greedy、DQN、DDPG 和 DROO 算法，QFCO 算法获得最优性能。当数据大小设置为数据取值范围平均值 250 KB 时，选取本文算法分别与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，分辨率增加 14.45%；同时 QFCO 优于 DDPG，分辨率提高 24.54%。

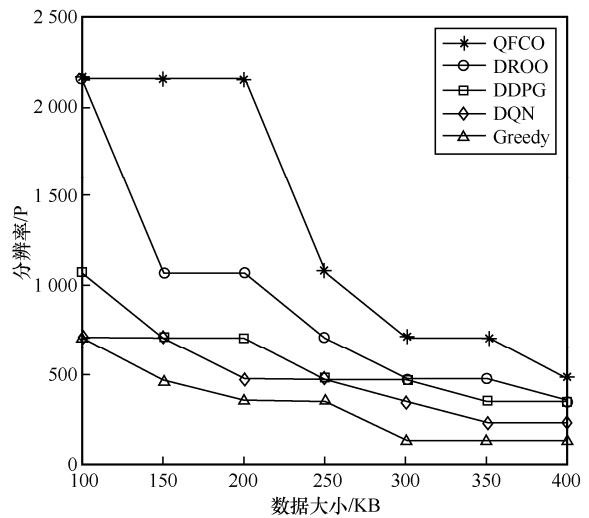


图 6 分辨率随数据大小的变化情况

图 7 是服务器计算能力对时延的影响。随着边缘服务器计算能力的增加，时延降低，边缘服务器计算能力增加，更多的数据可以被边缘服务器处理，从而将更多数据上传到边缘服务器。边缘服务器的计算与通信资源远大于终端，取得的执行时延要低于本地处理，同时边缘服务器靠近终端，传输时延较低，所以执行时延和传输时延之和有了明显的下降。从图 7 中可知，当服务器计算能力为 8 GHz 时，选取本文算法分别与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，时延缩短 8.17%；同时 QFCO 优于 DDPG，时延缩短 15.52%。

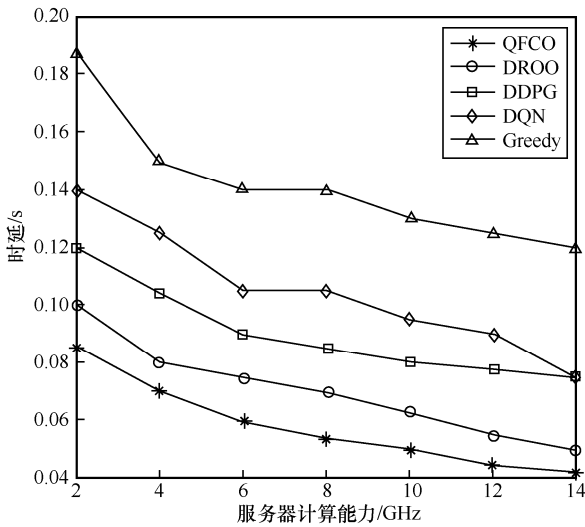


图 7 服务器计算能力对服务时延的影响

图 8 展示了服务器计算能力对能耗的影响。从图 8 中可以看出，随着服务器计算能力的增大，能耗不断上升。边缘服务器计算能力增大，服务器可以处理更多的数据，其处理任务能力上升会导致更多的数据涌向边缘服务器，能耗随之上升。当服务器计算能力为 2 GHz 时，虽然服务器计算能力较低，但是本文算法合理运用资源配置，减少冗余传输，所以系统能耗处于一个比较低的数值。由于更多的数据从本地上传到边缘服务器，能耗也随之上升。从图 8 中可知，相比其他算法 QFCO 取得更好的效果，当服务器计算能力为 8 GHz 时，选取本文算法分别与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，能耗降低 7.78%；同时 QFCO 优于 DDPG，能耗降低 14.45%。

例如当服务器计算能力为 2~4 GHz 时，Greedy、DQN、DDPG、DROO 算法的分辨率保持稳定状态，原因是对于上述算法，当前服务器计算能力的提升太小，不足以提高视频帧的分辨率，当服务器计算能力进一步提高时，视频帧的分辨率随之提升。由于计算能力的不断增大，其处理任务能力上升，分辨率随之上升，并且在分辨率方面有显著提升。当服务器计算能力为 2 GHz 时，虽然服务器计算能力较低，但是本文算法合理运用资源配置，减少冗余传输，所以在计算资源有限时，依然取得较高的分辨率。相比其他算法，QFCO 取得更好的效果，当服务器计算能力为 8 GHz 时，选取本文算法分别与 DROO 和 DDPG 比较，QFCO 与 DROO 相比，分辨率增加 43.26%；同时 QFCO 优于 DDPG，分辨率提高 57.63%。

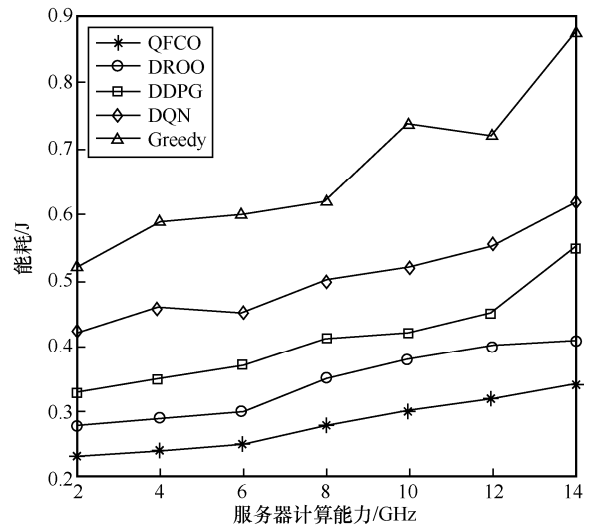


图 8 服务器计算能力对能耗的影响

图 9 展示了服务器计算能力对分辨率的影响。从图 9 中可以看出，随着服务器计算能力的增大，分辨率不断上升。图 9 中部分曲线保持水平状态。

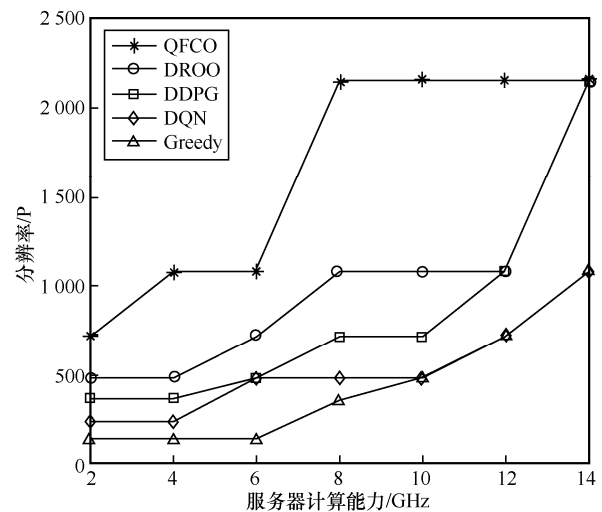


图 9 服务器计算能力对分辨率的影响

## 4 结束语

本文对移动边缘计算中基于 QoE 网络媒体流卸载方法进行研究。通过联合考虑预处理和资源配置问题，建立最小化网络冗余传输和最大化资源配置、资源利用率卸载模型。综合考虑时延、能耗和精确度，建立 QoE 模型目标函数。根据业务特征确定任务优先级，通过优先级分配给任务更合适的资源。将原始非凸问题解耦为 2 个子问题进行研究。通过应用 KKT 条件求解拉格朗日方程，得到最优计算资源分配的闭式表达式。此外，本文提出一种基于 QoE 的反馈配置卸载算法，对任务卸载进行合理的资源配置。仿真结果表明，本文算法可有效降低时延和能耗，提高 QoE。

## 附录 1 目标函数凹凸性证明

QoE 优化问题式(23)的目标函数如式(37)所示。为了便于计算，令  $y = \alpha T + \beta E + \eta A$ ，并对其求二阶偏导。

$$y = T_{n,i}^V(\text{tr}) + T_{n,i}^j(\text{tr}) + T_{n,i}^C(\text{tr}) + T_{n,i}^j + E_{n,i}^V(\text{tr}) + E_{n,i}^j(\text{tr}) + E_{n,i}^C(\text{tr}) + \left( \varepsilon_i^j x_{n,i}^j(t) r_{n,i} \right) \phi_k^j(f_{n,i}^j(t)) = \frac{D_{n,i}}{r_{n,i}^V} + \frac{2D_{n,i}}{r_{n,i}^j} + T_c + \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} + P^V \frac{D_{n,i}}{r_{n,i}^V} + P^M \frac{2D_{n,i}}{r_{n,i}^j} + E_c + \delta_{n,i}^L \left( f_{n,i}^j \right)^2 c_{n,i} + \left( \varepsilon_i^j x_{n,i}^j(t) r_{n,i} \right) \phi_k^j(f_{n,i}^j(t)) \quad (37)$$

$$y' = \left( \frac{D_{n,i} C_{n,i}}{f_{n,i}^j} \right)' = \frac{-D_{n,i} C_{n,i}}{\left( f_{n,i}^j \right)^2} \quad (38)$$

$$y'' = \left( \frac{-D_{n,i} C_{n,i}}{\left( f_{n,i}^j \right)^2} \right)' = \frac{2D_{n,i} C_{n,i}}{\left( f_{n,i}^j \right)^3} > 0 \quad (39)$$

由式(39)可得，问题 P2 为凸优化问题。证毕。

## 参考文献：

- [1] XU W, YANG Z H, NG D W K, et al. Edge learning for 5G networks with distributed signal processing: semantic communication, edge computing, and wireless sensing[J]. IEEE Journal of Selected Topics in Signal Processing, 2023, 17(1): 9-39.
- [2] KHAN M A, BACCOUR E, CHKIRBENE Z, et al. A survey on mobile edge computing for video streaming: opportunities and challenges[J]. IEEE Access, 2022, 10: 120514-120550.
- [3] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing—a key technology towards 5G[R]. ETSI White Paper, 2015.
- [4] DU J B, YU F R, LU G Y, et al. MEC-assisted immersive VR video

- streaming over terahertz wireless networks: a deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(10): 9517-9529.
- [5] HOU Y Z, WANG C R, ZHU M, et al. Joint allocation of wireless resource and computing capability in MEC-enabled vehicular network[J]. China Communications, 2021, 18(6): 64-76.
- [6] NGUYEN T H, PARK L. A survey on deep reinforcement learning-driven task offloading in aerial access networks[C]//Proceedings of 13th International Conference on Information and Communication Technology Convergence (ICTC). Piscataway: IEEE Press, 2022: 822-827.
- [7] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [8] HU H J, SHAN H G, WANG C K, et al. Video surveillance on mobile edge networks—a reinforcement-learning-based approach[J]. IEEE Internet of Things Journal, 2020, 7(6): 4746-4760.
- [9] JAIN S, ZHANG X, ZHOU Y H, et al. Spatula: efficient cross-camera video analytics on large camera networks[C]//Proceedings of IEEE/ACM Symposium on Edge Computing (SEC). Piscataway: IEEE Press, 2020: 110-124.
- [10] DAI X X, YANG P, ZHANG X Y, et al. RESPIRE: reducing spatial-temporal redundancy for efficient edge-based industrial video analytics[J]. IEEE Transactions on Industrial Informatics, 2022, 18(12): 9324-9334.
- [11] DOU W C, ZHAO X, YIN X C, et al. Edge computing-enabled deep learning for real-time video optimization in IIoT[J]. IEEE Transactions on Industrial Informatics, 2021, 17(4): 2842-2851.
- [12] HUNG C C, ANANTHANARAYANAN G, BODIK P, et al. VideoEdge: processing camera streams using hierarchical clusters[C]//Proceedings of IEEE/ACM Symposium on Edge Computing (SEC). Piscataway: IEEE Press, 2018: 115-131.
- [13] LIU Q, HAN T. DARE: dynamic adaptive mobile augmented reality with edge computing[C]//Proceedings of IEEE 26th International Conference on Network Protocols (ICNP). Piscataway: IEEE Press, 2018: 1-11.
- [14] WU K, JIN Y B, MIAO W W, et al. Soudain: online adaptive profile configuration for real-time video analytics[C]//Proceedings of IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). Piscataway: IEEE Press, 2021: 1-10.
- [15] ZHANG S, WANG C, JIN Y B, et al. Adaptive configuration selection and bandwidth allocation for edge-based video analytics[J]. IEEE/ACM Transactions on Networking, 2022, 30(2): 285-298.
- [16] YUAN X M, ZHANG Z D, FENG C J, et al. A DQN-based frame aggregation and task offloading approach for edge-enabled IoMT[J]. IEEE Transactions on Network Science and Engineering, 2023, 10(3): 1339-1351.
- [17] QIAO G H, LENG S P, MAHARJAN S, et al. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2020, 7(1): 247-257.
- [18] HUANG L, BI S Z, ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. IEEE Transactions on Mobile Computing, 2020, 19(11): 2581-2593.
- [19] 朱思峰, 蔡江昊, 柴争义, 等. 车联网云边协同计算场景下的多目

- 标优化卸载决策[J]. 通信学报, 2022, 43(6): 223-234.
- ZHU S F, CAI J H, CHAI Z Y, et al. Multi-objective optimal offloading decision for cloud-edge collaborative computing scenario in Internet of vehicles[J]. Journal on Communications, 2022, 43(6): 223-234.
- [20] XU X L, WU Q, QI L Y, et al. Trust-aware service offloading for video surveillance in edge computing enabled Internet of vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(3): 1787-1796.
- [21] 龙隆, 刘子辰, 陆在旺, 等. 移动边缘网络下服务缓存与资源分配联合优化策略[J]. 通信学报, 2023, 44(1): 64-74.
- LONG L, LIU Z C, LU Z W, et al. Joint optimization strategy of service cache and resource allocation in mobile edge network[J]. Journal on Communications, 2023, 44(1): 64-74.
- [22] MA X T, LI Q, JIANG Y, et al. Learning-based joint QoE optimization for adaptive video streaming based on smart edge[J]. IEEE Transactions on Network and Service Management, 2022, 19(2): 1789-1806.
- [23] 孙国林, 欧睿杰, 刘贵松. 基于深度强化学习的应急物联网切片资源预留算法[J]. 通信学报, 2020, 41(9): 8-20.
- SUN G L, OU R J, LIU G S. Deep reinforcement learning-based resource reservation algorithm for emergency Internet-of-things slice[J]. Journal on Communications, 2020, 41(9): 8-20.
- [24] WANG C M, YU F R, LIANG C C, et al. Joint computation offloading and interference management in wireless cellular networks with mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2017, 66(8): 7432-7445.
- [25] ZHANG G L, ZHANG S, ZHANG W Q, et al. Joint service caching, computation offloading and resource allocation in mobile edge computing systems[J]. IEEE Transactions on Wireless Communications, 2021, 20(8): 5288-5300.
- [26] 邝祝芳, 陈清林, 李林峰, 等. 基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法[J]. 计算机学报, 2022, 45(4): 812-824.
- KUANG Z F, CHEN Q L, LI L F, et al. Multi-user edge computing task offloading scheduling and resource allocation based on deep reinforcement learning[J]. Chinese Journal of Computers, 2022, 45(4): 812-824.
- [27] YANG B, CAO X L, BASSEY J, et al. Computation offloading in multi-access edge computing: a multi-task learning approach[J]. IEEE Transactions on Mobile Computing, 2021, 20(9): 2745-2762.
- [28] GOODFELLOW I, BENGIO Y, COURVILLE A, et al. Deep learning[M]. Cambridge: MIT Press, 2016.
- [29] KANG Z Y, YOU C S, ZHANG R. 3D placement for multi-UAV relaying: an iterative Gibbs-sampling and block coordinate descent optimization approach[J]. IEEE Transactions on Communications, 2021, 69(3): 2047-2062.

#### [作者简介]



王再见 (1980- ), 男, 安徽定远人, 博士, 安徽师范大学教授、博士生导师, 主要研究方向为面向 5G/B5G 的移动多媒体通信、大数据技术、视频分析、人工智能等。



程浩 (1997- ), 男, 安徽宣城人, 安徽师范大学硕士生, 主要研究方向为边缘计算、无线多媒体通信。