



研究与开发

基于移动边缘计算的多流自适应卸载方案

胡叠丽^{1,2,3}, 杨哲铭^{1,3}, 纪雯^{1,2}

(1. 中国科学院计算技术研究所, 北京 100190;

2. 鹏城实验室, 广东 深圳 518055;

3. 中国科学院大学计算机科学与技术学院, 北京 101408)

摘要: 海量视频流的传输和分析需要大量的带宽和计算资源, 这对当前基于移动边缘计算 (mobile edge computing, MEC) 的卸载方案提出了严峻挑战。对此, 提出了一种基于多流协同优化框架的自适应卸载方案。首先, 在满足长期MEC能量预算的约束条件下, 通过联合优化数据流选择决策、服务器卸载决策、带宽资源分配和计算资源分配来最小化视频任务的处理成本。然后, 基于李雅普诺夫优化方法, 将长期优化问题转化为每个时隙独立的确定性子问题, 并利用马尔可夫近似和KKT条件求解每个时隙的混合整数非线性规划问题。仿真结果表明, 所提方案在满足长期MEC能量约束的同时, 其成本性能显著优于已有基准研究方案。

关键词: 移动边缘计算; 多流自适应卸载; 视频处理; 李雅普诺夫优化

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2024233

Multi-stream adaptive offloading scheme based on mobile edge computing

HU Dieli^{1,2,3}, YANG Zheming^{1,3}, JI Wen^{1,2}

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

2. Peng Cheng Laboratory, Shenzhen 518055, China

3. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China

Abstract: The transmission and analysis of massive video streams require significant edge bandwidth and computational resources, posing severe challenges to the current multimedia frameworks based on mobile edge computing (MEC). To address this issue, an adaptive offloading scheme based on a multi-stream collaborative optimization framework was proposed. Firstly, under the constraint of the long-term MEC energy budget, the processing cost of video tasks was minimized by optimizing the data stream selection decisions, server offloading decisions, bandwidth resource allocation, and computing resource allocation. Then, based on the Lyapunov optimization method, the long-

收稿日期: 2024-07-30; 修回日期: 2024-10-14

基金项目: 国家重点研发计划项目 (No.2023YFB4502805); 国家自然科学基金资助项目 (No.62072440); 北京市自然科学基金资助项目 (No.L221004)

Foundation Items: The National Key Research & Development Program of China (No.2023YFB4502805), The National Natural Science Foundation of China (No.62072440), The Beijing Natural Science Foundation (No.L221004)



term optimization problem was transformed into independent deterministic subproblems for each time slot, and the mixed-integer nonlinear programming problems for each time slot were solved by Markov approximation and KKT conditions. Simulation results indicate that the proposed scheme not only meets the long-term MEC energy constraint, but also significantly outperforms existing benchmark schemes in terms of cost performance.

Key words: mobile edge computing, multi-stream adaptive offloading, video processing, Lyapunov optimization

0 引言

视频在移动网络流量中的占比不断增加。根据爱立信预测,到2028年,视频流量将占据全球移动数据流量的80%^[1]。随着视频应用程序的兴起,对更低时延、更大带宽^[2]和更高密度的计算需求也日益增加。在视频任务中,移动边缘计算(mobile edge computing, MEC)发挥着至关重要的作用,以提供低时延服务^[3-4]。然而,大规模高码率压缩视频流的传输和分析需要较高的带宽和充足的边缘计算资源^[5-6]。因此,鉴于计算、通信和存储资源的限制,当前多媒体框架无法满足大规模视频分析的需求^[7]。

为了解决上述问题,一些研究工作^[5-7]分别针对多媒体系统^[5-6]和智慧城市^[7],提出了一种新颖的多流框架,通过混合压缩视频流、特征流和语义流来替代传统的压缩视频流。在这些框架中,压缩视频流主要服务于人类视觉,特征流是从原始视频中实时提取的紧凑特征数据,其比特率显著低于压缩视频流,主要服务于机器视觉^[7],而语义流则提供语义和信息感知以促进任务处理^[5]。这样的框架能够同时支持各种应用,如目标识别和人工辅助交通违规监控,因为特征数据和视觉数据都是可用的^[6]。将紧凑的特征流卸载到边缘服务器时,其比特率远低于压缩视频流^[7],从而显著降低了带宽需求。此外,后续的高级数据分析操作是基于特征数据和语义数据实现的,而不是基于压缩视频流重新构建原始的视频像素,这种方法能够大大节省计算资源和缓存资源^[5]。虽然特征流具有上述优点,但在终端设备

上进行特征提取可能会导致更高的计算时延和能耗。因此,自适应卸载方案对于基于MEC的多流协同优化框架至关重要。该方案能够依据有限的计算能力和带宽资源,实时选择最适合的数据流进行卸载。

近年来,如何设计高性能的MEC系统引起了大量学者的关注。许多任务从资源受限的终端设备被卸载到附近的边缘服务器,以实现多种性能目标^[8-15]。其中,文献[8]聚焦于最大化视频分析准确性并最小化时延的问题,文献[9]致力于降低用户能耗,文献[10]研究了最大化回传数据量的问题,文献[11]旨在最小化任务平均完成时间与请求资源数量的长期加权和。文献[12]专注于最大化用户的体验质量(quality of experience, QoE)并最小化能耗的问题,文献[13]研究了最小化任务时延和能耗组合函数的问题,文献[14-15]则关注最小化时延的问题。通常情况下,MEC的计算和带宽资源有限。因此,MEC中的资源分配问题备受关注,以进一步提高系统性能。例如,文献[8]设计了一种针对多设备边缘视频分析系统的在线联合视频配置和带宽分配算法,目的是改善移动视频分析中的准确性一时延权衡。文献[9]提出了一个基于MEC的能量感知视频分析系统,通过联合优化应用配置和边缘服务器的带宽资源分配,在不降低用户对服务时延和检测准确性偏好的基础上,最小化用户每帧视频的处理能耗。文献[10]通过联合优化任务卸载路径和计算资源分配,在严格的时限要求内最大化总回传数据量。文献[11]优化了任务卸载中的计算资源分配问题,旨在最小化任务平均完成时

间和请求资源数量的长期加权。尽管针对 MEC 任务卸载的研究已有较多成果,但往往未考虑任务卸载中的计算资源分配问题^[8-9],或者忽略了带宽分配^[10-11],从而导致性能次优。显然,不同的计算资源和带宽资源分配策略通常会导致不同的时延和能耗。因此,对于 MEC 系统的整体性能而言,计算资源和带宽资源的联合分配显得尤为重要。在此基础上,部分研究进一步集成了计算资源与带宽资源分配,深入探讨任务卸载问题^[12-15]。文献[12]提出了一种基于深度强化学习的高效视频卸载方案,以在安全速率、计算能力和无线资源受限的情况下提升用户的 QoE 并降低能耗。文献[13]综合考虑了 MEC 的计算卸载与网络切片的资源调度,在有限资源条件下,致力于最小化任务执行时延和系统能耗。文献[14]提出了在异构边缘环境中联合任务卸载、接入点选择、带宽和计算资源分配的方法,以实现时延最小化。文献[15]研究了 MEC 系统中为自主无线设备分配无线和计算资源的问题,其中设备可自主决定是否利用 MEC 资源卸载计算任务,而边缘云运营商则负责为设备分配无线和计算资源,以最小化任务完成时间。

上述研究主要致力于在不同 MEC 场景下设计高效的任务卸载和资源分配方案。这些研究表明,MEC 支持的任务卸载和资源分配能够有效降低系统时延和能耗。然而,任务卸载也会引入额外的 MEC 能耗^[16]。现有研究大多集中在短期性能优化方面,而忽略了边缘节点长期能量约束的影响。这种忽略在实际应用中可能导致系统运行效率下降,甚至引发性能次优问题,特别是在能量受限的场景下。受上述多流框架研究及基于 MEC 的任务卸载和资源分配研究的启发,本文研究了长期 MEC 能量约束下的多流卸载和资源分配问题,提出了一种创新的多流自适应卸载方案,该方案在考虑长期 MEC 能量约束的基础上,根据实时信道状态和系统资源,实现了联合压缩

视频流、特征流和语义流的多流自适应卸载,并对计算资源和通信资源进行了联合优化。

本文考虑了一个由多个终端设备和边缘服务器组成的系统,其中终端设备可以将压缩视频流卸载到边缘服务器进行特征提取,或将特征流卸载到服务器进行更深入的视频分析。此外,卸载的数据流中须额外包含一个超低比特率的语义流,该流携带终端设备 ID、时间戳、位置和其他基于视频预处理的语义数据,以便于任务处理^[5-6]。考虑压缩视频流的卸载可能导致额外的 MEC 能耗,甚至可能违背长期 MEC 能量预算,本文研究了在长期 MEC 能量受限条件下的多流卸载和资源分配性能。为了在长期 MEC 能量约束下实现有效的多流自适应卸载和资源分配,有以下几个关键挑战需要解决。

(1) 与传统压缩流相比,多个数据流的卸载更加复杂。在决定选择卸载哪种数据流时,需要考虑任务处理时延、终端设备能耗以及终端设备和边缘服务器的异构性。

(2) 除了确定要选择的数据流类型,还需要确定卸载到哪个边缘服务器以及相应的带宽资源和计算资源分配。MEC 带宽资源和计算资源分配需要适应跨时隙的各种任务需求。当视频任务卸载到附近的边缘服务器时,带宽资源或计算资源不足会导致较长的传输或计算时延。然而,如果单个终端设备被分配过多的带宽资源或计算资源时,边缘服务器有限的带宽资源或计算资源可能不足以支持所有终端设备的数据传输或任务处理。最佳的资源分配策略可以平衡服务器之间的有限资源。

(3) 如果边缘服务器在当前时隙耗能太多,则将没有能量预算可用于后续时隙。这种时间耦合的 MEC 能量约束需要有关任务需求的完整未来信息,以保证卸载性能不受影响。

为了解决上述挑战,本文主要贡献如下。

(1) 对联联合压缩视频流、特征流和语义流的



多流协同优化框架进行了研究,旨在提出一种基于MEC的多流自适应卸载方案以高效处理异构终端设备的视频任务。

(2) 本文对多流卸载和资源分配优化问题进行了建模,并通过优化数据流选择决策、服务器卸载决策、计算资源分配和带宽资源分配,实现在长期MEC能量约束下,最小化任务处理时延与终端设备能耗加权平均的任务处理成本。

(3) 由于无线信道和系统资源的随机性,以及优化变量在不同时隙上的相互关联性,建立的问题涉及多阶段序列决策的随机优化,直接求解较为困难。为解决此问题,本文采用了李雅普诺夫优化方法,将其分解为每个时隙的确定性优化问题。每个时隙的子问题都利用马尔可夫近似和KKT条件来求解。最后,仿真结果验证了所提方案的性能优于所有其他基准方案。

1 系统模型和问题描述

考虑一个基于MEC的多流视频任务卸载系统,系统模型如图1所示。在该系统中,假设有

U 个终端设备,终端设备集合为 $\mathcal{U}=\{1,2,\dots,U\}$,定义 $u\in\mathcal{U}$ 为该终端设备集合中的某一个终端设备。边缘服务器集合为 $\mathcal{M}=\{1,2,\dots,M\}$,定义 $m\in\mathcal{M}$ 为该边缘服务器集合中的某一个边缘服务器。每个终端设备不断将视频任务卸载到边缘服务器,其选择是将压缩视频流和语义流一起卸载到一个边缘服务器进行特征提取,或者在本地进行特征提取,然后将提取的特征流和语义流一起卸载到一个边缘服务器。边缘服务器不仅具备特征提取能力,还承担了数据存储和预处理等功能。借助最先进的虚拟化技术和频分多址技术,边缘服务器能够为多个终端设备提供虚拟计算资源和带宽资源,以同时执行多项任务^[16]。最终,云服务器能够从边缘服务器收集特征数据,并进行进一步的视频分析。本文专注于终端设备和边缘服务器之间的多流卸载。云中的数据分析和采集时延在这里可以忽略不计,因为云服务器具有强大的计算能力,并且特征数据比原始视频少得多^[6]。

在所考虑系统中,从 U 个终端设备到 M 个边缘服务器的任务卸载是在时隙基础上执行的。

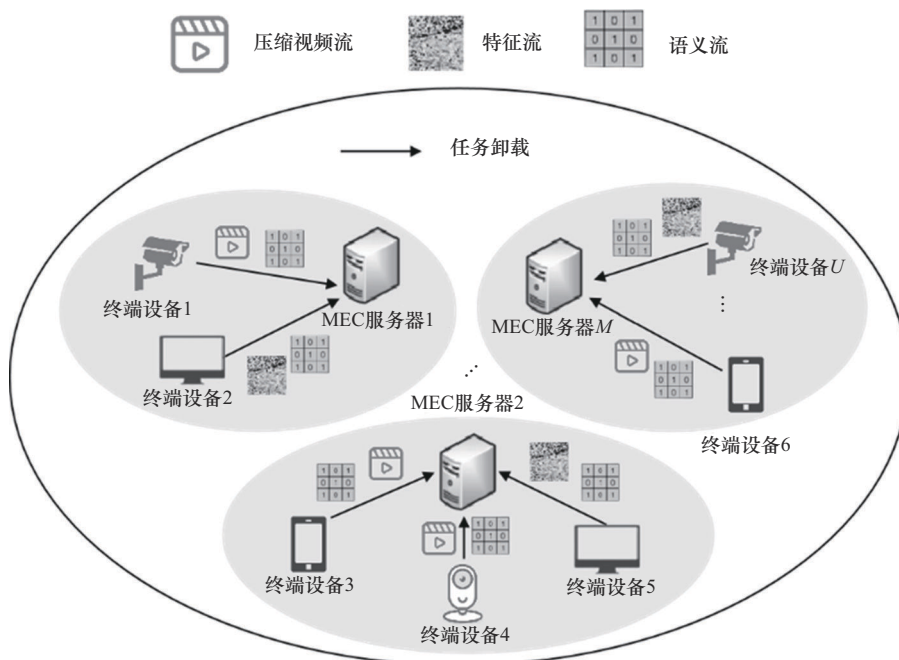


图1 系统模型

与文献[4]类似, 假设时间被离散地划分为 T 个等长时隙, 时隙集合为 $\mathcal{T} = \{1, 2, \dots, T\}$, 定义 $t \in \mathcal{T}$ 为该时隙集合中的某一个时隙索引。本文假设在每个时隙内, 信道状态信息及边缘服务器的计算资源和通信资源保持恒定, 而在不同时隙之间发生变化。时隙持续时间与视频任务卸载和资源分配决策更新的时间尺度相匹配。因此, 为了实现全局和长期的最佳资源配置状态, 在每个时隙的开始时刻根据当前系统资源和历史信息制定自适应卸载和资源分配策略。这样的策略能够确保最佳的资源利用。此外, 本文分别定义了终端设备的数据流选择决策 $x_u^t \in \{0, 1\}$ 和边缘服务器卸载决策 $y_{u,m}^t \in \{0, 1\}$ 。其中 $x_u^t = 1$ 表示终端设备选择卸载的是压缩视频流, $x_u^t = 0$ 表示终端设备选择卸载的是特征流。 $y_{u,m}^t = 1$ 表示在时隙 t 终端设备 u 卸载数据到边缘服务器 m , 否则 $y_{u,m}^t = 0$ 。

1.1 信道模型

假设终端设备和边缘服务器之间的连接基于正交频分多址接入技术, 则带宽资源的分配对应于资源单元的分配^[14]。设 $w_{u,m}^t$ 是边缘服务器 m 在时隙 t 分配给终端设备 u 的带宽资源。因此, 在第 t 个时隙, 从终端设备 u 到边缘服务器 m 的上行数据传输速率可以表示为:

$$R_{u,m}^t = w_{u,m}^t \log \left(1 + \frac{p_u^t |h_{u,m}^t|^2}{N_0^t} \right) \quad (1)$$

其中, p_u^t 是终端设备 u 在时隙 t 上传视频任务时的发射功率, N_0^t 是噪声功率, $h_{u,m}^t$ 是终端设备 u 与边缘服务器 m 的上行信道增益。与文献[17]中的假设一致, 本文不考虑小区间干扰和同小区干扰。这是因为本文的研究重点是系统级性能, 而非无线链路级性能。所提方案可被视为实际场景中性能的上限。

1.2 时延模型

无论终端设备生成的是压缩视频流还是特征

流, 都需要卸载到边缘服务器。当终端设备 u 将压缩视频卸载到边缘服务器 m 时, 边缘服务器必须对接收到的视频帧进行解码和特征提取。当终端设备 u 将特征流卸载到边缘服务器 m 时, 需要先在本地进行特征提取。因此, 视频任务每帧的处理时延是传输时延加上计算时延的函数。传输时延受数据大小和上行数据传输速率的共同影响。对于时隙 t 中终端设备 u 的分辨率为 r_u^t 的压缩视频帧, 一个视频帧的数据大小表示为 $s_u^t = \sigma r_u^t$, 其中 σ 是常数^[12], 代表每个像素包含的比特数。与传统压缩视频流相比, 特征流的比特率显著降低^[5]。在时隙 t 中, 第 u 个终端设备每帧特征数据的大小表示为 $s_u^{t,F} = \gamma s_u^{t,C}$, 其中 γ 也是常数, 代表特征数据相对于原始帧数据大小的压缩率^[6]。设 $s_u^{t,I}$ 为时隙 t 中终端设备 u 的语义信息数据的大小。因此, 可以得出在第 t 个时隙, 当终端设备 u 卸载数据流到边缘服务器 m 执行任务时, 每一帧的传输时延为:

$$l_u^{t, \text{tran}} = \frac{x_u^t s_u^{t,C} + (1 - x_u^t) s_u^{t,F} + s_u^{t,I}}{\sum_{m=1}^M y_{u,m}^t R_{u,m}^t} \quad (2)$$

定义第 u 个终端设备在时隙 t 进行特征提取时, 每比特视频任务所需的计算资源 (即每比特所需的 CPU/GPU 周期数) 为 c_u^t , $f_{u,0}^t$ 是终端设备 u 本地的计算能力 (CPU/GPU 周期频率), $f_{u,m}^t$ 是边缘服务器 m 分配给终端设备 u 的计算资源。由此, 可以得到在第 t 个时隙, 终端设备 u 的视频任务的每帧计算时延为:

$$l_u^{t, \text{comp}} = \begin{cases} \frac{s_u^{t,C} c_u^t}{f_{u,0}^t}, & x_u^t = 0 \\ \frac{s_u^{t,C} c_u^t}{\sum_{m=1}^M y_{u,m}^t f_{u,m}^t}, & x_u^t = 1 \end{cases} \quad (3)$$

因此, 终端设备 u 的视频任务在时隙 t 的每帧总处理时延可以表示为:



$$l_u^t = l_u^{t, \text{tran}} + l_u^{t, \text{comp}} \quad (4)$$

1.3 能耗模型

随着视频服务规模的扩大, 计算量和数据量的增加不可避免地导致能源消耗的上升。鉴于此, 本文着重关注终端设备和边缘服务器的能源效率, 特别是视频流的传输和处理对能耗的影响。终端设备的传输能耗可以表示为:

$$e_u^{t, \text{tran}} = p_u^t l_u^{t, \text{tran}} \quad (5)$$

根据文献[13], 定义 κ 为每个终端设备相对于芯片架构的能耗系数, 可得终端设备 u 在时隙 t 处理特征提取任务 (即 $x_u^t = 0$) 时, 每帧本地计算能耗为:

$$e_u^{t, \text{comp}} = \kappa s_u^{t, C} c_u^t (f_{u,0}^t)^2 \quad (6)$$

结合传输能耗和计算能耗的表达式, 可以得到终端设备 u 在时隙 t 处理视频任务的每帧本地总能耗为:

$$e_u^t = e_u^{t, \text{tran}} + (1 - x_u^t) e_u^{t, \text{comp}} \quad (7)$$

当终端设备 u 卸载压缩视频流到边缘服务器进行特征提取 (即 $x_u^t = 1$) 时, 该任务将由边缘服务器根据其分配的计算资源来执行。设 μ 为执行 1 个 CPU/GPU 周期的边缘服务器能耗^[16]。由此, 在第 t 个时隙, 边缘服务器 m 处理视频任务的计算总能耗为:

$$e_m^t = \sum_{u=1}^U \mu x_u^t y_{u,m}^t s_u^{t, C} c_u^t \quad (8)$$

M 个边缘服务器在第 t 个时隙处理视频任务的平均计算能耗为:

$$e^t = \frac{1}{M} \sum_{m=1}^M e_m^t \quad (9)$$

1.4 问题描述

处理实时视频任务不仅能耗高, 还对时延极为敏感。考虑移动终端设备充电通常不便, 电池寿命便成了终端用户的主要关注点。因此, 在设计自适应方案时, 目标是在满足长期 MEC 能量预算的约束条件下, 尽可能地降低任务处理时延

与终端设备能耗加权平均的任务处理成本。综合式 (4) 所示的任务处理时延和式 (7) 所示的终端设备本地能耗, 定义在第 t 个时隙的平均任务处理成本为:

$$Q^t = \frac{1}{U} \sum_{u=1}^U (\phi l_u^t + (1 - \phi) e_u^t) \quad (10)$$

其中, ϕ 和 $1 - \phi$ 分别为处理时延和终端设备能耗的权重, $\phi \in [0, 1]$ 。MEC 支持的任务卸载节省了终端设备的本地计算能耗, 但会产生额外的 MEC 能耗。因此, 目标是在长期 MEC 能量预算约束下最小化长期平均任务处理成本。定义 $\mathbf{x} = \{x_u^t | u \in \mathcal{U}, t \in \mathcal{T}\}$, $\mathbf{y} = \{y_{u,m}^t | u \in \mathcal{U}, m \in \mathcal{M}, t \in \mathcal{T}\}$, $\mathbf{w} = \{w_{u,m}^t | u \in \mathcal{U}, m \in \mathcal{M}, t \in \mathcal{T}\}$, 以及 $\mathbf{f} = \{f_{u,m}^t | u \in \mathcal{U}, m \in \mathcal{M}, t \in \mathcal{T}\}$ 。因此, 本文将多流卸载和资源分配优化问题表示为:

$$\begin{aligned} \text{P1: } & \min_{\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{f}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[Q^t] \\ \text{s.t. } & \text{C1: } w_{u,m}^t \geq 0, \forall u, m, t \\ & \text{C2: } \sum_{u=1}^U y_{u,m}^t w_{u,m}^t \leq W_m^{t, \max}, \forall m, t \\ & \text{C3: } f_{u,m}^t \geq 0, \forall u, m, t \\ & \text{C4: } 0 \leq \sum_{u=1}^U x_u^t y_{u,m}^t f_{u,m}^t \leq F_m^{t, \max}, \forall m, t \\ & \text{C5: } x_u^t \in \{0, 1\}, \forall u, t \\ & \text{C6: } \sum_{m=1}^M y_{u,m}^t = 1, \forall u, t \\ & \text{C7: } y_{u,m}^t \in \{0, 1\}, \forall u, m, t \\ & \text{C8: } \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[e^t] \leq E^{\max} \end{aligned} \quad (11)$$

约束 C1 和约束 C2 表示分配给所有关联终端设备的带宽资源必须为正数, 并且总和不能超过边缘服务器的带宽资源上限。约束 C3 和约束 C4 表示分配给所有关联终端设备的计算资源必须为正数, 并且总和不能超过边缘服务器的计算资源上限。约束 C5 限制数据流选择指示符为二进制

变量。约束 C6 和约束 C7 确保每个终端设备只能将视频任务卸载到一台边缘服务器。由于边缘节点的运行是长期的，约束 C8 要求长期时间范围内 MEC 消耗的平均能量不能超过长期能量预算 $E^{\max}$ 。由于 MEC 具有时间相关性，任务卸载和资源分配策略必须考虑跨时隙的耦合效应。特别是，如果在某一时隙 t 边缘服务器能耗过多，那么在接下来的时隙中，服务器的剩余可用能量将不可避免地减少，进而影响随后的任务卸载与资源分配的效率。这种能源消耗的时间依赖性是实现有效任务卸载和资源配置的一个主要障碍。

问题 P1 的难度在于 2 个方面。首先，由于各个时隙的系统信息（即信道状态信息、边缘服务器的计算和通信资源）是动态变化的，在时隙 t 准确预测未来信息是很困难的。而服务器能耗的时间相关性导致卸载决策跨时隙耦合，P1 是一个经典的随机优化问题，很难直接求解。其次，即使事先获得未来信息，求解 P1 也很困难，因为 P1 是混合整数非线性规划（mixed integer nonlinear programming, MINLP）问题。为了有效解决这一问题，需要设计一种在线算法，该算法可以在线调整卸载策略，并为终端设备分配通信资源和计算资源，而不需要长期的全局信息。

2 基于自适应卸载和资源分配算法的多流自适应卸载方案

P1 中的长期时间平均约束 C8 使得目标函数在不同时隙之间耦合，导致高维动态规划。为了解除长期时间平均约束，本文提出了一种低复杂度的自适应卸载和资源分配算法。该算法基于李雅普诺夫优化方法^[18]，将原始的随机优化问题转化为各个时隙的独立子问题，并利用马尔可夫近似^[19]和 KKT 条件^[20]对每个时隙的子问题进行求解。

2.1 李雅普诺夫优化框架

为了使用李雅普诺夫优化方法求解问题 P1，

首先，将虚拟队列 $a(t)$ 定义为在时隙 t 结束时的能量不足队列，用以衡量超出 MEC 能量预算的程度，并假设初始队列积压^[3, 21]为零，即 $a(0) = 0$ 。因此虚拟队列的状态变化为：

$$a(t+1) = \max\{a(t) + e^t - E^{\max}, 0\} \quad (12)$$

如式 (12) 所示，时隙 $t+1$ 的队列积压是时隙 t 的积压与当前时隙的 MEC 能量预算超出部分 $(e^t - E^{\max})$ 之和^[21-23]。当边缘服务器在第 t 个时隙耗能过多时，第 $t+1$ 个时隙的虚拟队列 $a(t+1)$ 将会增大。文献[18]中已经证明，保证队列 $a(t)$ 的稳定性等同于满足问题 P1 中的长期时间平均约束 C8。也就是说，如果虚拟队列稳定，即 $\lim_{T \rightarrow \infty} \frac{\mathbb{E}[a(T)]}{T} = 0$ ，实际 MEC 能耗将不会超出能量预算。根据李雅普诺夫理论，可将二次李雅普诺夫函数^[21-23]作为 MEC 能量预算超出程度的度量：

$$L(a(t)) = \frac{1}{2}(a(t))^2 \quad (13)$$

$L(a(t))$ 较小意味着在第 t 个时隙队列积压较小。为了验证长期时间平均约束 C8，引入李雅普诺夫漂移函数^[21-23] $\Delta(a(t))$ ：

$$\Delta(a(t)) = \mathbb{E}[L(a(t+1)) - L(a(t)) | a(t)] \quad (14)$$

式 (14) 表示 $a(t)$ 从时隙 t 到时隙 $t+1$ 变化的期望值。 $\Delta(a(t))$ 越小，说明虚拟队列稳定性越强。为了保证所有队列的稳定性，自适应卸载策略应最小化李雅普诺夫漂移。基于李雅普诺夫优化理论，通过对式 (14) 和问题 P1 的目标函数进行加权，进一步定义出李雅普诺夫漂移加惩罚函数^[21, 23]，即：

$$\Delta(a(t)) + V \cdot \mathbb{E}[Q^t | a(t)] \quad (15)$$

其中，非负权重参数 V 用于控制 Q^t 和边缘服务器能耗之间的折中。

为了进一步简化优化问题，引入了引理 1，引理 1 给出了李雅普诺夫漂移加惩罚函数的上限。



引理 1 在每个时隙中, 李雅普诺夫漂移加惩罚函数存在以下上限:

$$\Delta(a(t)) + V \cdot \mathbb{E}[Q^t | a(t)] \leq B + a(t) \mathbb{E}[e^t - E^{\max} | a(t)] + V \cdot \mathbb{E}[Q^t | a(t)] \quad (16)$$

其中, B 为常数, 不影响李雅普诺夫优化, 可以写为 $B = \frac{1}{2}(\max_{t \in \mathcal{T}} \{e^t\} - E^{\max})^2$.

证明 从式 (14) 可以得到:

$$\begin{aligned} \Delta(a(t)) &= \mathbb{E}[L(a(t+1)) - L(a(t)) | a(t)] = \\ &= \frac{1}{2} \mathbb{E}[a^2(t+1) - a^2(t) | a(t)] \leq \\ &= \frac{1}{2} \mathbb{E}[(a(t) + e^t - E^{\max})^2 - a^2(t) | a(t)] = \\ &= \frac{1}{2} (e^t - E^{\max})^2 + a(t) \mathbb{E}[e^t - E^{\max} | a(t)] \leq \\ &= B + a(t) \mathbb{E}[e^t - E^{\max} | a(t)] \end{aligned} \quad (17)$$

现在, 将原优化问题的目标函数 $V \cdot \mathbb{E}[Q^t | a(t)]$ 合并到式 (17) 的两边, 那么可以得到:

$$\Delta(a(t)) + V \cdot \mathbb{E}[Q^t | a(t)] \leq B + a(t) \mathbb{E}[e^t - E^{\max} | a(t)] + V \cdot \mathbb{E}[Q^t | a(t)] \quad (18)$$

证毕。

基于李雅普诺夫优化方法, 可以利用式 (16) 中漂移加惩罚函数的上界把优化问题 P1 转化为最小化每个时隙中漂移加惩罚函数的上限^[22], 将依赖未来信息的 P1 转化为当前时隙的实时优化问题 P2:

$$\begin{aligned} \text{P2: } \min_{x^t, y^t, w^t, f^t} \quad & a(t) \cdot e^t + V \cdot Q^t \\ \text{s.t.} \quad & \text{C1} \sim \text{C7} \end{aligned} \quad (19)$$

其中, x^t 、 y^t 、 w^t 、 f^t 分别表示时隙 t 所有终端设备的数据流选择、服务器卸载、带宽资源分配和计算资源分配决策。求解 P2 仅需要当前可用信息作为输入。通过考虑附加项 $a(t) \cdot e^t$, 系统以在线方式满足长期 MEC 能量约束。

2.2 当前时隙优化问题求解

问题 P2 是 MINLP 问题, 一旦确定了卸载决策 x^t 和 y^t , 优化问题 P2 可以转化为:

$$\begin{aligned} \text{P3: } \min_{w^t, f^t} \quad & Q^t \\ \text{s.t.} \quad & \text{C1} \sim \text{C4} \end{aligned} \quad (20)$$

对于问题 P3 中的边缘带宽资源 w^t 和边缘计算资源 f^t 的分配, 其目标函数和约束条件可以彼此解耦, 从而将问题 P3 转化为两个独立的优化子问题。这两个独立的子问题分别是: 边缘带宽资源分配和边缘计算资源分配。

(1) 边缘带宽资源分配

边缘带宽资源分配问题表示为:

$$\begin{aligned} \text{P4: } \min_{w^t} \quad & Q^t \\ \text{s.t.} \quad & \text{C1}, \text{C2} \end{aligned} \quad (21)$$

可以看出, 式 (21) 中目标函数的海森矩阵是对称正定矩阵, 根据文献[20], 可以得出 P4 是凸优化问题, 可以使用 KKT 条件^[20, 24]求解。因此有以下引理。

引理 2 问题 P4 的最优边缘带宽资源分配为:

$$w_{u,m}^{t,*} = \frac{W_m^{t,\max} \sqrt{\eta_u^t}}{\sum_{u \in \mathcal{U}_m^t} \sqrt{\eta_u^t}}, \quad \forall u \in \mathcal{U}_m^t, m \in \mathcal{M}, t \in \mathcal{T} \quad (22)$$

其中, $\mathcal{U}_m^t = \{u \in \mathcal{U} | y_{u,m}^t = 1\}$ 表示时隙 t 将视频任务卸载到边缘服务器 m 的一组终端设备集合, η_u^t 为:

$$\eta_u^t = \frac{y_{u,m}^t (\phi + (1-\phi) p_u^t) (x_u^t s_u^{t,C} + (1-x_u^t) s_u^{t,F} + s_u^{t,I})}{U \text{lb} \left(1 + \frac{p_u^t |h_{u,m}^t|^2}{N_0^t} \right)} \quad (23)$$

证明 式 (21) 的拉格朗日函数为:

$$\begin{aligned} \mathcal{L}(w, \mu) &= \sum_{u=1}^U \left(\frac{\eta_u^t}{w_{u,m}^t} \right) + \\ &= \sum_{m=1}^M \mu_m \left(\sum_{u=1}^U y_{u,m}^t w_{u,m}^t - W_m^{t,\max} \right) \end{aligned} \quad (24)$$

其中, $\boldsymbol{\mu} = [\mu_1, \dots, \mu_m, \dots, \mu_M]$ 是拉格朗日乘子向量。对拉格朗日函数 $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\mu})$ 求关于 $w_{u,m}^t$ 的偏导可以得到:

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, \boldsymbol{\mu})}{\partial w_{u,m}^t} = -\frac{\eta_u^t}{(w_{u,m}^t)^2} + \mu_m, \quad \forall u \in \mathcal{U}_m^t, m \in \mathcal{M}, t \in \mathcal{T} \quad (25)$$

当 $w_{u,m}^t = w_{u,m}^{t*}$ 时, 满足 $\frac{\partial \mathcal{L}(\boldsymbol{w}, \boldsymbol{\mu})}{\partial w_{u,m}^t} = 0$ 。因此可以得到:

$$w_{u,m}^{t*} = \sqrt{\frac{\eta_u^t}{\mu_m}}, \quad \forall u \in \mathcal{U}_m^t, m \in \mathcal{M}, t \in \mathcal{T} \quad (26)$$

其中, $\mu_m^* > 0$ 。

根据 KKT 条件, 可以得到:

$$\sum_{u \in \mathcal{U}_m^t} w_{u,m}^{t*} = W_m^{t, \max}, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \quad (27)$$

将式 (26) 代入式 (27), 可以得到:

$$\mu_m^* = \left(\frac{1}{W_m^{t, \max}} \sum_{u \in \mathcal{U}_m^t} \sqrt{\eta_u^t} \right)^2 \quad (28)$$

将式 (28) 代入式 (26), 可以得到:

$$w_{u,m}^{t*} = \frac{W_m^{t, \max} \sqrt{\eta_u^t}}{\sum_{u \in \mathcal{U}_m^t} \sqrt{\eta_u^t}} \quad (29)$$

证毕。

(2) 边缘计算资源分配

边缘计算资源分配问题表示为:

$$\begin{aligned} \text{P5: } \min_{\boldsymbol{f}^t} \quad & Q^t \\ \text{s.t.} \quad & \text{C3、C4} \end{aligned} \quad (30)$$

当给定卸载决策 $\boldsymbol{x}^t$ 和 $\boldsymbol{y}^t$ 时, 优化问题 P5 是凸优化问题, 和求解问题 P4 的思路相同, 可以应用 KKT 条件求解得到最优边缘计算资源分配为:

$$f_{u,m}^{t*} = \frac{F_m^{t, \max} \sqrt{x_u^t s_u^{t,C} c_u^t}}{\sum_{u \in \mathcal{U}_m^{t,C}} \sqrt{x_u^t s_u^{t,C} c_u^t}}, \quad \forall u \in \mathcal{U}_m^{t,C}, m \in \mathcal{M}, t \in \mathcal{T} \quad (31)$$

其中, $\mathcal{U}_m^{t,C} = \{u \in \mathcal{U} \mid x_u^t y_{u,m}^t = 1\}$ 表示时隙 t 将压缩视频流卸载到边缘服务器 m 的一组终端设备集合。

根据以上分析, 一旦确定了数据流选择决策 $\boldsymbol{x}^t$ 和服务器卸载决策 $\boldsymbol{y}^t$, 问题 P2 就很容易求解。由于 $\boldsymbol{x}^t$ 和 $\boldsymbol{y}^t$ 是离散优化变量, 无法在多项式时间内获得最优解, 本文基于马尔可夫近似算法^[19,25]获得数据流选择和服务器卸载决策的近似最优解。

首先利用首次适应 (first-fit) 算法^[26]将视频流分配给边缘服务器作为初始边缘服务器卸载决策 $\boldsymbol{y}_0^t$: 每个视频流的大小被视为 1, 时隙 t 第 m 个边缘服务器的容量被视为:

$$v_m^t = \frac{W_m^{t, \max} + \delta F_m^{t, \max}}{\sum_{m=1}^M (W_m^{t, \max} + \delta F_m^{t, \max})} U, \quad \forall m \in \mathcal{M}, t \in \mathcal{T} \quad (32)$$

其中, δ 是控制计算资源和带宽资源之间权衡的参数。这样可以确保 M 个边缘服务器可以容纳 U 个视频流。直观地说, 上行带宽大、计算能力强的边缘服务器应能容纳更多的视频流。然后, 为所有终端设备随机选取数据流作为初始数据流选择决策 $\boldsymbol{x}_0^t$ 。因此, 所有终端设备的初始卸载决策表示为 $\boldsymbol{d}_0^t = \{\boldsymbol{x}_0^t, \boldsymbol{y}_0^t\}$ 。接下来, 在每次迭代中, 随机选择一个终端设备 u 并更改其卸载决策 d_u^t 为 $\tilde{d}_u^t$ (例如, 终端设备 u 原本卸载压缩流到某一服务器, 更改为卸载特征流到任意服务器或者卸载压缩流到新的服务器), 同时保持其他终端设备的卸载决策不变, 得到新的卸载决策 $\tilde{\boldsymbol{d}}^t$ 。然后可以使用新的卸载决策来求解 P3 以获得新的带宽资源分配决策 $\tilde{\boldsymbol{w}}^t$ 和计算资源分配决策 $\tilde{\boldsymbol{f}}^t$ 。新的目标函数表示为 $\tilde{g}$, 并随后以概率 ρ 更新为新的自适应卸载和资源分配策略 $\{\tilde{\boldsymbol{d}}^t, \tilde{\boldsymbol{w}}^t, \tilde{\boldsymbol{f}}^t\}$, 同时以概率 $1-\rho$



保持不变, 其中, $\rho = \frac{1}{\left(1 + e^{\frac{(\tilde{g}-g)}{\tau}}\right)^{[25]}}$, g 为通过旧

的自适应卸载和资源分配策略 $\{d^t, w^t, f^t\}$ 得到的目标函数值, 参数 $\tau > 0$ 用于控制随机性。当新的自适应卸载和资源分配策略 $\{\tilde{d}^t, \tilde{w}^t, \tilde{f}^t\}$ 导致目标函数减小时, 卸载决策更有可能被更新, 因为概率 ρ 取决于目标函数差值 $(\tilde{g}-g)$ 。当迭代次数达到上限或 $|\tilde{g}-g| < 0.01$ 且迭代次数超过 15 次时, 算法停止迭代。

综上, 可得自适应卸载和资源分配算法如算法 1 所示。

算法 1 自适应卸载和资源分配算法

输入 $\phi, \omega, \kappa, E^{\max}, p_u^t, c_u^t, s_u^{t,C}, s_u^{t,F}, s_u^{t,I}, f_{u,0}^t, \mu, \forall u$

输出 d^t, w^t, f^t

(1) 初始化: $a(0) \leftarrow 0$

(2) 循环 while $t \leq T$

for $m = 1:1:M$

根据式 (32) 计算第 m 个边缘服务器的容量

利用 first-fit 算法^[26]将 U 个视频流分配给 M 个边缘服务器并得到初始卸载决策 d_0^t , 设置 $d^t \leftarrow d_0^t$

循环 do

随机选择一个设备 u 并更改其卸载决策 d_u^t 为 $\tilde{d}_u^t$, 得到新的卸载决策 $\tilde{d}^t \leftarrow \{d_1^t, d_2^t, \dots, \tilde{d}_u^t, \dots, d_U^t\}$

每个设备根据式 (22) 和式 (31) 求解得到新的带宽资源分配决策 $\tilde{w}^t$ 和计算资源分配决策 $\tilde{f}^t$

计算 $\rho = \frac{1}{\left(1 + e^{\frac{(\tilde{g}-g)}{\tau}}\right)}$

以概率 ρ 更新到新的自适应卸载和资源分配策略 $d^t \leftarrow \tilde{d}^t, w^t \leftarrow \tilde{w}^t, f^t \leftarrow \tilde{f}^t$

until 达到迭代停止条件 (迭代次数达到上限或 $|\tilde{g}-g| < 0.01$ 且迭代次数超过 15 次)

更新 $a(t+1) \leftarrow \max \{a(t) + e^t - E^{\max}, 0\}$

end while

算法 1 的计算复杂度主要来自问题式 (19)。

在每次更新迭代中, 系统会任意选择一个终端设备更新其卸载决策。所有可行决策的总数最多可以列举到 $2M$ 。假设需要迭代 I 次才能收敛, 则时间复杂度为 $O(2MI)$ 。因此, 算法 1 是低复杂度的高效算法。

3 仿真实验与结果分析

3.1 实验设置

本节通过计算机仿真比较了本文所提多流自适应卸载方案与 5 个基线方案的性能。模拟环境为一个面积为 $500 \text{ m} \times 500 \text{ m}$ 的正方形区域, 其中配置了 4 台边缘服务器, 终端设备数量为 20。为简单起见, 假设终端设备和边缘服务器均匀部署在这个正方形区域中。每个时隙长度为 1 s , 将时隙 t 中从终端设备 u 到边缘服务器 m 的上行信道增益表示为 $h_{u,m}^t = \sqrt{\beta_0 (d_{u,m})^{-\alpha}} g_{u,m}^t$ ^[27], 其中 $\beta_0 = -30 \text{ dBm}$ 是参考距离 1 m 处的平均信号功率衰减, $d_{u,m}$ 是终端设备 u 到边缘服务器 m 之间的距离, $\alpha = 3$ 表示路径损耗指数^[28], $g_{u,m}^t$ 表示小尺度衰落系数, 它是一个均值为零、单位方差的指数分布随机变量, $N_0 = 10^{-10} \text{ W}$ 。根据文献[5], 与压缩视频流相比, 特征流平均可以降低 70% 的传输比特率, 因此, 特征数据相对于原始帧数据大小的压缩率 γ 设置为 0.3。根据文献[29], 对 15 天数据集的语义数据进行分析, 可以认为语义数据大小均匀分布在 $[400, 720]$ 比特。视频的分辨率均匀分布在 $\{360\text{P}, 540\text{P}, 720\text{P}, 960\text{P}, 1080\text{P}\}$ ^[6]。根据文献[30], 视频任务的计算需求可以粗略地分为 3 类: 重型 ($10\,000$ 周期/比特)、中型 ($2\,500$ 周期/比特) 和轻型 (250 周期/比特)。边缘服务器具有多种可用的通信和计算资源, 边缘

服务器的通信资源和计算资源被建模为马尔可夫链^[31], 其中, $P_r(W_m^{t+1, \max} = z | W_m^{t, \max} = z') = P_{zz'}$, $\forall z, z' \in \{1, 2, 3, 4, 5\}$ MHz, $P_r(F_m^{t+1, \max} = z | F_m^{t, \max} = z') = P_{zz'}$, $\forall z, z' \in \{2, 3, 4, 5, 6\}$ GHz。其余参数设置为: $p_u^t = 20$ dBm, $f_{u,0}^t = [1, 2]$ GHz, $\kappa = 10^{-27}$ ^[16], $E^{\max} = 10$ J, $\mu = 8.2$ nJ^[16], $\phi = 0.5$ 。

在仿真中, 将本文所提多流自适应卸载方案与以下 5 种基准方案进行比较。

(1) 仅压缩流方案^[12]: 所有终端设备只卸载压缩视频流到边缘服务器, 不受长期 MEC 能量预算的约束。

(2) 仅特征流方案: 所有终端设备在本地进行特征提取并卸载特征流到边缘服务器。

(3) 带宽资源分配方案^[8]: 视频任务从终端设备卸载到边缘服务器时, 可以根据任务需求调整分配的带宽资源。

(4) 计算资源分配方案^[11]: 该方案认为边缘服务器的计算资源可以根据任务特性灵活配置, 以获得更好的系统性能。

(5) 随机卸载方案: 这是一种没有优化边缘服务器卸载决策的方案, 其中卸载决策是随机的。

3.2 实验结果与分析

不同非负权重参数 V 对平均任务处理成本和平均队列积压性能的影响如图 2 所示。随着参数 V 的增大, 平均任务处理成本逐渐减少并最终趋于稳定。相反, 平均队列积压性能与参数 V 呈正相关。这是因为在问题 P2 中, 系统在优化时赋予了平均任务处理成本更高的权重, 从而使得降低边缘服务器队列积压性能的要求变得相对宽松。当参数 V 趋近于无穷大时, 最优平均任务处理成本将以更大的平均队列积压性能为代价来实现。因此, 可以在算法中通过调整参数 V 至适当的值, 以实现平均任务处理成本和平均队列积压性能之间的平衡。

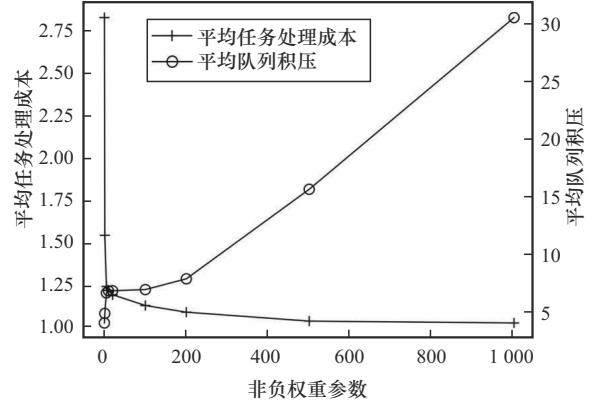


图2 不同非负权重参数 V 对平均任务处理成本和平均队列积压性能的影响

不同非负权重参数 V 对平均队列积压性能的影响如图 3 所示。所有平均队列积压均逐渐收敛到一定值, 这反映了所提算法能够满足长期 MEC 能量约束。理论证明, 如果虚拟队列稳定, 即 $\lim_{T \rightarrow \infty} \frac{E[a(T)]}{T} = 0$, 那么在实际的长期时间范围内, MEC 消耗的平均能量将不会超出能量预算。因此, 随着时隙的增加, 系统可以满足长期 MEC 能量约束。此外, 当参数 V 较大时, 问题 P2 更多关注降低任务处理成本, 导致平均队列积压的收敛速度变慢, 从而需要更多时隙来实现收敛。相反, 当参数 V 较小时, 问题 P2 侧重于减少边缘服务器的队列积压, 从而有助于快速稳定平均队列积压。

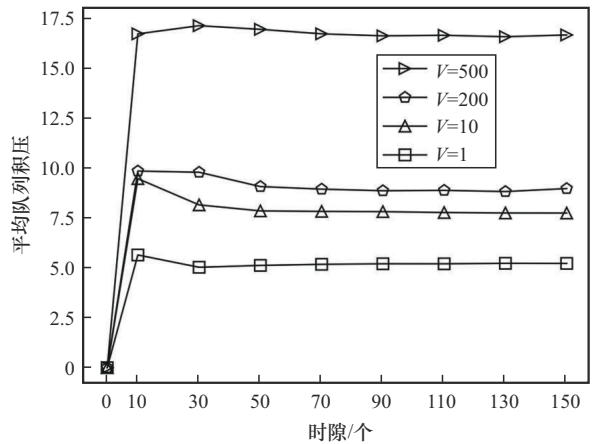


图3 不同非负权重参数 V 对平均队列积压性能的影响



MEC能量预算变化时不同方案的性能比较如图4所示。与5种基准方案相比,本文所提多流自适应卸载方案的平均任务处理成本最低。具体而言,与仅压缩流方案相比,本文方案最多能节省53.77%的成本。因此,采用本文提出的基于MEC的多流自适应卸载方案,可以显著降低任务处理成本。此外,在图4中发现增加能量预算可以降低平均任务处理成本,这是因为增加能量预算有助于缓解MEC能量不足的问题,提升边缘服务器计算能力,从而促进任务卸载,减少终端设备本地计算能耗。

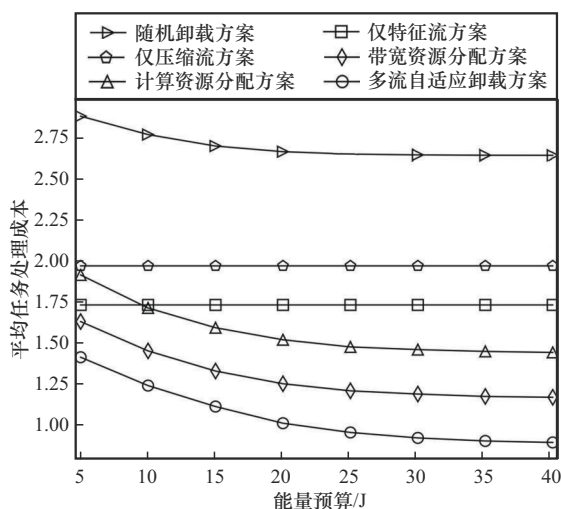


图4 MEC能量预算变化时不同方案的性能比较

特征流压缩率变化时不同方案的性能比较如图5所示。随着压缩率 γ 的增加,除仅压缩流方案外,所有方案的平均任务处理成本均有所上升。这是由于特征数据大小的增加导致了特征流传输时延和传输能耗的增加。此外,实验结果验证了本文所提多流自适应卸载方案在所有特征流压缩率区域范围内均优于5种基线方案。当 $\gamma=0.3$ 时,本文方案的成本相比计算资源分配方案和带宽资源分配方案,分别降低了27.49%和14.49%。值得注意的是,本文方案的任务处理成本明显低于仅压缩流方案,这证实了多流自适应卸载方案与传统压缩视频流方案相比的优越性。

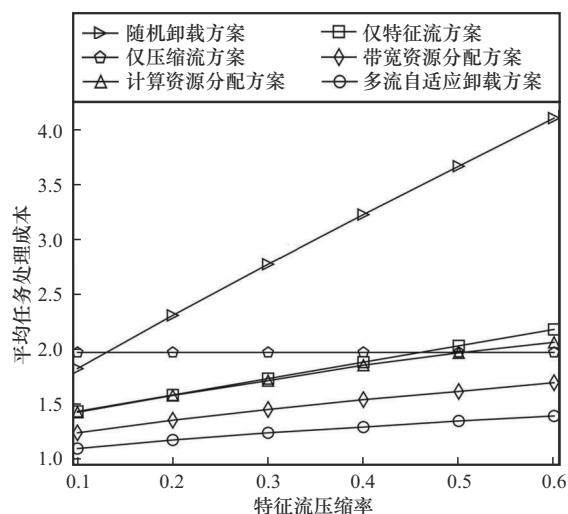


图5 特征流压缩率变化时不同方案的性能比较

终端设备数量变化时不同方案的性能比较如图6所示。从图6可以看出,当终端设备数量较少(如 $U \leq 15$)时,仅压缩流方案的任务成本相对较低,然而,随着终端设备数量的增加(如 $U > 15$),仅特征流方案的成本优势逐渐显现。这一现象可以归因于终端设备数量的增多导致边缘服务器的上行通信资源和计算资源变得紧张,并逐渐无法满足所有终端设备的通信和计算需求。在这种情况下,仅特征流方案依靠稳定的终端设备本地计算资源及较低的传输时延,使其成本效益更加突出。

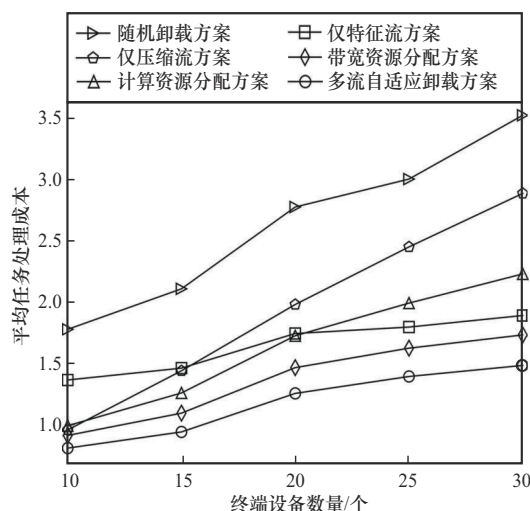


图6 终端设备数量变化时不同方案的性能比较

视频分辨率变化时不同方案的性能比较如图7所示。随着视频分辨率的增加,所有方案的平均任务处理成本均呈现增长趋势。本文方案确保即使在高分辨率条件下,也能维持较低的成本,从而增强系统整体性能和成本效率。此外,可以发现,在视频分辨率低于720P时,计算资源分配方案的成本低于仅特征流方案,而在视频分辨率高于720P时,成本高于仅特征流方案。这是因为随着分辨率的提高,视频任务对上行带宽和边缘服务器资源的需求增大。计算资源分配方案未优化上行带宽资源的分配,导致较长的传输时延,而仅特征流方案依靠终端设备本地稳定的计算资源和较短的传输时延,在高分辨率下的成本优势更为显著。

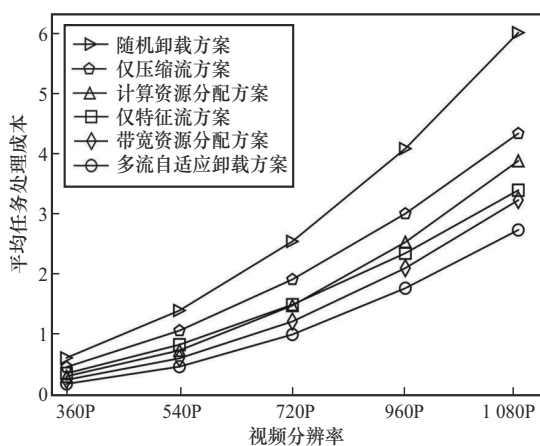


图7 视频分辨率变化时不同方案的性能比较

利用马尔可夫近似算法解决问题P2时,马尔可夫近似算法在不同带宽下的收敛性如图8所示。从图8中可以观察到,该算法总是能在60次迭代内收敛到接近最优解。图8还表明,在少于10次的迭代后,算法所得到的成本已降低约60%。鉴于马尔可夫近似算法已被证明为近似最优,这一结果不仅印证了其在求解效率上的优势,同时也强调了其在迅速接近最优解方面的高效性能。此外,可以发现,带宽越小,算法的收敛速度越慢。这是因为在带宽受限的情况下,系统可用资源

变得有限。每个终端设备获得的带宽减少,导致系统在资源分配上更加紧张。此时,任何小的优化变量的调整都会显著影响P2中的目标函数。此时,需要更加精确的自适应卸载和资源分配方案,从而需要更多的迭代次数来收敛。

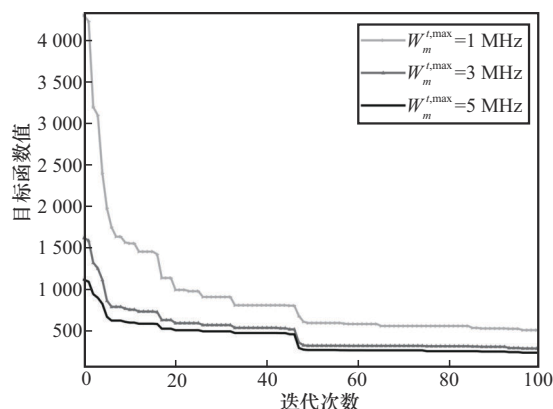


图8 马尔可夫近似算法在不同带宽下的收敛性

4 结束语

本文提出了一种基于MEC的多流自适应卸载方案,以支持联合压缩视频流、特征流和语义流的多流自适应卸载,同时最大限度地降低平均任务处理成本。针对长期MEC能量受限情况,将动态资源分配问题建模为一个随机优化问题。通过利用李雅普诺夫优化方法,将复杂的随机优化问题转化为一系列确定性的子问题进行求解,从而提出了一种低复杂度的自适应卸载和资源分配算法。仿真结果验证了所提方案和算法的优越性能。在未来的研究中,多流自适应卸载方案将进一步拓展,考虑更多影响系统性能的关键参数,例如终端设备的本地计算频率、发射功率和视频分辨率等。这些拓展将为实际应用场景提供更加有效和灵活的解决方案。

参考文献:

- [1] Ericsson. Ericsson mobility report 2022[R]. Stockholm: Ericsson, 2022.
- [2] Cisco. Cisco annual internet report (2018 - 2023) white paper[R].



- San Jose, CA: Cisco, 2020.
- [3] ZHANG S, WANG C, JIN Y, et al. Adaptive configuration selection and bandwidth allocation for edge-based video analytics[J]. *IEEE/ACM Transactions on Networking*, 2022, 30(1): 285-298.
- [4] YANG P, LYU F, WU W, et al. Edge coordinated query configuration for low-latency and accurate video analytics[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16(7): 4855-4864.
- [5] JI W, LIANG B, WANG Y, et al. Crowd V-IoE: visual internet of everything architecture in ai-driven fog computing[J]. *IEEE Wireless Communications*, 2020, 27(2): 51-57.
- [6] HU D L, JI W, WANG Z. Multi-stream adaptive offloading of joint compressed video streams, feature streams, and semantic streams in edge computing systems[C]//*Proceedings of the 2023 IEEE International Conference on Multimedia and Expo (ICME)*. Piscataway: IEEE Press, 2023: 996-1001.
- [7] GAO W, MA S W, DUAN L Y, et al. Digital retina: a way to make the city brain more efficient by visual coding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(11): 4147-4161.
- [8] WANG S Y, BI S Z, ZHANG Y-J A. Edge video analytics with adaptive information gathering: a deep reinforcement learning approach[J]. *IEEE Transactions on Wireless Communications*, 2023, 22(9): 5800-5813.
- [9] WANG H, XIE J. User preference based energy-aware mobile AR system with edge computing[C]//*Proceedings of the IEEE INFOCOM 2020-IEEE conference on computer communications*. Piscataway: IEEE Press, 2020: 1379-1388.
- [10] LIU Y, LI Y, NIU Y, et al. Joint optimization of path planning and resource allocation in mobile edge computing[J]. *IEEE Transactions on Mobile Computing*, 2020, 19(9): 2129-2144.
- [11] XIONG X, ZHENG K, LEI L, et al. Resource allocation based on deep reinforcement learning in IoT edge computing[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(6): 1133-1146.
- [12] ZHAO T T, HE L J, HUANG X Y, et al. DRL-based secure video offloading in EC-enabled IoT networks[J]. *IEEE Internet of Things Journal*, 2022, 9(19): 18710-18724.
- [13] 王晔, 王逸飞, 陈康, 等. 6G 网络任务卸载与细粒度切片资源调度联合优化算法[J]. *电信科学*, 2024, 40(5): 86-99.
- WANG Y, WANG Y F, CHEN K, et al. Joint optimization algorithm for 6G network task offloading and fine-grained slice resource scheduling[J]. *Telecommunications Science*, 2024, 40(5): 86-99.
- [14] LIU Y, MAO Y, LIU Z, et al. Joint task offloading and resource allocation in heterogeneous edge environments[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(6): 7318-7334.
- [15] JOŠILO S, DÁN G. Wireless and computing resource allocation for selfish computation offloading in edge computing[C]//*Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. Piscataway: IEEE Press, 2019: 2467-2475.
- [16] JIANG H B, DAI X X, XIAO Z, et al. Joint task offloading and resource allocation for energy-constrained mobile edge computing[J]. *IEEE Transactions on Mobile Computing*, 2023, 22(7): 4000-4015.
- [17] 吴柳青, 朱晓荣. 基于边-端协同的任务卸载资源分配联合优化算法[J]. *电信科学*, 2020, 36(3): 42-52.
- WU L Q, ZHU X R. Joint optimization algorithm for task offloading resource allocation based on edge-end collaboration[J]. *Telecommunications Science*, 2020, 36(3): 42-52.
- [18] NEELY M J. *Stochastic network optimization with application to communication and queueing systems*[M]. Switzerland: Springer, 2010.
- [19] CHEN M, LIEW S C, SHAO Z, et al. Markov approximation for combinatorial network optimization[J]. *IEEE Transactions on Information Theory*, 2013, 59(10): 6301-6327.
- [20] BOYD S, VANDENBERGHE L. *Convex optimization*[M]. Cambridge: Cambridge University Press, 2004.
- [21] NING Z, DONG P, WANG X, et al. Distributed and dynamic service placement in pervasive edge computing networks[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 32(6): 1277-1292.
- [22] LONG Y, ZHAO S, GONG S, et al. AoI-aware sensing scheduling and trajectory optimization for multi-UAV-assisted wireless backscatter networks[J]. *IEEE Transactions on Vehicular Technology*, 2024.
- [23] LIU Y, CHEN Q C, TANG X H, et al. On the buffer energy aware adaptive relaying in multiple relay network[J]. *IEEE Transactions on Wireless Communications*, 2017, 16(9): 6248-6263.
- [24] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(1): 856-868.
- [25] OUYANG T, ZHOU Z, CHEN X. Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing[J]. *IEEE Journal on Selected Areas in Communications*, 2018, 36(10): 2333-2345.
- [26] CORMEN T H, LEISERSON C E, RIVEST R L, et al. Intro-

- duction to algorithms[M]. 2nd ed. Cambridge, MA, USA: MIT Press, 2009.
- [27] HU D L, HUANG G F, TANG D, et al. Joint task offloading and computation in cooperative multicarrier relaying-based mobile-edge computing systems[J]. IEEE Internet of Things Journal, 2021, 8(14): 11487-11502.
- [28] GOLDSMITH A. Wireless communications[M]. Cambridge: Cambridge University Press, 2005.
- [29] WANG S G, GUO Y, ZHANG N, et al. Delay-aware microservice coordination in mobile edge computing: a reinforcement learning approach[J]. IEEE Transactions on Mobile Computing, 2021, 20(3): 939-951.
- [30] OUYANG T, CHEN X, ZHOU Z, et al. Adaptive user-managed service placement for mobile edge computing via contextual multiarmed bandit learning[J]. IEEE Transactions on Mobile Computing, 2023, 22(3): 1313-1326.
- [31] CHENG Z P, MIN M H, LIWANG M H, et al. Multiagent DDPG-based joint task partitioning and power control in fog computing networks[J]. IEEE Internet of Things Journal, 2022, 9(1): 104-116.

[作者简介]



胡叠丽 (1995–), 女, 中国科学院计算技术研究所、鹏城实验室和中国科学院大学计算机科学与技术学院博士生, 主要研究方向为边缘计算、多媒体系统、网络优化等。



杨哲铭 (1996–), 男, 中国科学院计算技术研究所和中国科学院大学计算机科学与技术学院博士生, 主要研究方向为多媒体系统、边缘智能、分布式推理、机器学习等。



纪雯 (1976–), 女, 博士, 中国科学院计算技术研究所研究员、博士生导师, 鹏城实验室兼职教授, 主要研究方向为多媒体系统、智能多媒体计算编码传输、多媒体VPU芯片设计等。