数字孪生边缘网络端到端时延优化的任务卸载与资源分配方法

李 松*①② 李 顺① 王博文① 孙彦景①②
①(中国矿业大学信息与控制工程学院 徐州 221116)
②(徐州市智能安全与应急协同工程研究中心 徐州 221116)

摘 要：针对移动边缘计算(MEC)场景中任务卸载、计算和结果反馈全过程时延优化问题，该文提出了一种数字孪生(DT)辅助的联合MEC任务卸载、设备关联与资源分配的端到端时延优化方法。首先，在数字孪生边缘网络(DITEN)框架下，为包含传感器、边缘服务器以及执行器构成的边缘计算网络建立了物理模型与数字孪生模型，以及全过程边缘网络任务模型并推导了任务端到端时延，进而建立了时延、能耗等约束下的端到端时延优化问题。其次，为解决所提出的混合整数非凸优化问题，将原问题分解为4个子问题，并提出了一种基于内部凸近似方法和匈牙利算法的交替优化算法。在DT辅助下联合优化了设备关联、卸载比例、发射功率、传输带宽以及DT估计处理速率。最后，仿真结果表明，与其他基准方案相比，所提联合优化方案显著降低了端到端时延。

关键词：移动边缘计算；数字孪生；任务卸载；资源分配；交替优化
中图分类号：TN929.5 **文献标识码**：A **文章编号**：1009-5896(2025)03-0633-12
DOI: 10.11999/JEIT240344

1 引言

随着无线通信与物联网的快速发展，自动驾驶和虚拟现实技术等时延敏感型和计算密集型应用大量出现。这类应用对时延有着更高要求，例如在自动驾驶领域，传感器需要不断感测环境信息，并对这些信息进行实时处理来保证驾驶安全。移动边缘计算(Mobile Edge Computing, MEC)利用边缘服务器强大的计算和存储能力能够实现更低的应用时延[1]。MEC通过任务卸载技术[2]，将资源受限设备的计算任务卸载到具有更强处理能力的边缘端，以满足任务要求。因此，为满足任务需求并充分发挥MEC的优势，设计MEC任务卸载和计算通信资源联合分配策略具有重要意义。

现有工作围绕MEC任务卸载和资源优化展开研究。文献[3]在支持MEC的车联网中提出一种基于优先级的任务调度算法，解决了具有依赖性任务的时延最小化问题。文献[4]利用深度强化学习方法优化多用户MEC系统中的任务时延，以获得更好的服务质量。文献[5]研究了多设备MEC系统中联合任务卸载和资源分配问题，并提出一种交替优化方法来减少受时延约束的设备能耗。文献[6]提出了一种针对上行链路多用户MEC的车联网资源分配算法，该算法能够在多维资源约束下最小化时延和能耗的加权和。然而，上述工作仅针对MEC任务卸载和资源分配展开研究，忽略了设备关联问题，而设备关联的优化设计不仅能够进一步降低时延，还能实现各个服务器的负载均衡。文献[7]通过联合优化设备关联、卸载比例以及计算通信资源来最小化任务时延，最大限度地减轻了服务器的计算负载。类似的，文献[8]也考虑了设备关联优化，并在云边端异构网络中得到了最优关联策略。但是，这些工作均忽略了本地计算结果的上传过程，而在分布式机器学习、跨区域人员跟踪等任务执行过程中，需将本地计算结果上传至服务器进行集中分析或存储，且本地计算结果数据量及传输时延不可忽略。文献[9]优化了本地计算结果上传过程，通过动态电压调节技术联合优化设备和MEC服务器的通信计算资源分配来实现设备能耗的最小化。虽然该工作

收稿日期：2024-04-29；改回日期：2025-02-12；网络出版：2025-02-20
*通信作者： 李松 lisong@cumt.edu.cn
基金项目：国家自然科学基金(62071472, 62101556)，江苏省自然科学基金(BK20210489)，中央高校基本科研业务费项目(2020ZDPYMS26)，江苏省研究生科研与实践创新计划项目(KYCX24_2763)，中国矿业大学研究生创新计划项目(2024WLJCRCZL133)，西安市网络融合通信重点实验室开放基金项目(2022NCC-N103)，海南省省属科研院所技术创新项目(KYYSGY2024-005)，工信部项目(CBG01N23-01-04)
Foundation Items: The National Natural Science Foundation of China (62071472, 62101556), The Natural Science Foundation of Jiangsu Province of China (BK20210489), The Fundamental Research Funds for the Central Universities (2020ZDPYMS26), The Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX24_2763), The Graduate Innovation Program of China University of Mining and Technology (2024WLJCRCZL133), Xi'an Key Laboratory of Network Convergence Communication (2022NCC-N103), The Project of Technical Innovation of Hainan Scientific Research Institutes (KYYSGY2024-005), The Ministry of Industry and Information Technology (CBG01N23-01-04)

考虑了本地结果的上传过程，但同其他工作一样，均基于设备状态与网络环境信息已知的理想假设，忽视了时变的信道条件以及不可预知的网络环境对卸载和资源分配策略产生的影响。

数字孪生(Digital Twin, DT)技术的发展，为解决上述问题提供了一种有效方案。DT通过在虚拟空间构建物理实体的数字镜像，实现了从物理世界到数字空间的映射[10]。DT模型建立后，便可借助数字域中丰富的历史和实时数据，对物理实体乃至整个系统进行模拟、预测和控制，在不浪费物理资源的情况下制定出最优卸载和资源分配策略[11]。鉴于DT的优势，文献[12,13]等将DT技术与边缘网络(Edge Network, EN)结合构建了数字孪生边缘网络(DIgital Twin Edge Network, DITEN)。在DITEN框架下，接入点(Access Point, AP)等边缘节点收集物理对象的实时信息，并基于这些信息建立和维护DT模型。通过这种方式，DITEN能够直接在数字域中设计和优化任务卸载与资源分配方案，不仅提高了网络决策效率，还能有效减少物理资源的浪费。而在没有DITEN时，为得到最优任务卸载和资源分配策略，需要边缘服务器和设备之间持续通信以获得实时信息。这不仅增加了通信成本，还可能因通信时延而影响决策的有效性。因此，DITEN的引入，降低了资源有限设备的计算负载，提升了决策生成效率，还能在最小化通信成本的同时获得最优资源分配策略。例如，文献[14]提出了一种DT结合智能反射面辅助的任务卸载方案，通过联合优化卸载决策、资源分配和反射面参数来最小化系统能耗。文献[15]在DITEN中考虑随机任务到达模型，提出一个随机计算卸载和资源分配问题，以最小化长期能量效率。文献[16]设计了一个无线DT模型，利用DT来解决用户与服务器之间不可靠的通信问题。文献[17]在DITEN框架下，利用DT估计边缘服务器状态并提供训练数据，进而使用深度强化学习的方法来降低卸载时延。

然而，上述工作均针对任务源设备和目标设备为同一设备的场景，即设备既是任务发起者，也是结果接收者，未考虑任务发起设备和结果接收设备为不同设备的情况。而在工业无人化作业场景中，传感器作为任务数据源，执行器接收计算结果对工业生产流程做出控制，任务执行全过程包括数据上传、任务计算、结果传输等过程。现有边缘计算任务卸载及资源优化研究侧重于任务计算环节，忽视了计算结果传输对整体性能的影响，尤其是端到端时延这一指标。端到端时延是指从用户发起任务请求，经过网络传输至边缘计算节点进行处理，再将计算结果返回至用户的总时延，是反映系统响应速度和实时性的重要指标。因此，有必要研究面向边缘网络端到端时延优化的任务卸载、设备关联及资源分配问题。

综上所述，本文在数字孪生边缘网络框架下，通过联合优化任务卸载、设备关联以及计算通信资源分配，来最小化全过程任务的端到端时延。其中，全过程任务时延由本地计算、本地结果上传、任务卸载、边缘计算以及结果下载5部分组成。由于优化问题的混合整数非凸特性，本文将原问题解耦为4个独立的子问题。然后利用内部凸近似方法(Inner Convex Approximation, ICA)和匈牙利算法(Hungarian Algorithm, HA)，提出了一种交替优化(Alternating Optimisation, AO)算法。本文主要贡献如下：

(1)首先，针对包括传感器、边缘服务器和执行器的MEC系统，建立物理域全过程任务模型、计算通信模型和能耗与时延模型，以及数字域物理设备和通信环境的DT模型，进而建立了数字孪生边缘网(DITEN)，在DITEN中根据设备和网络的历史及实时信息来模拟卸载决策与资源分配过程。

(2)其次，根据边缘网络全过程任务模型，推导了任务端到端时延。从最小化端到端时延的角度出发，在满足时延、能耗和功率等约束下，建立了任务执行过程中设备关联、任务卸载比例以及计算通信资源分配的联合优化问题。

(3)最后，由于优化问题是一个混合整数非凸非线性、多变量耦合的NP难问题。因此，将原问题分解为计算和通信资源优化、设备关联优化、卸载决策优化以及传输带宽优化4个子问题。然后基于内部凸近似方法(ICA)和匈牙利算法(HA)，提出了一种交替优化算法(AO)，即ICA-HA-AO算法。仿真结果表明，本文所提联合优化方案可以显著降低DT辅助MEC系统的端到端时延，性能优于基准方案。

本文的其余部分组织如下。第2节建立数字孪生边缘网络系统模型并提出优化问题。第3节提出基于内部凸近似方法和匈牙利算法的交替优化算法，以解决端到端时延最小化问题。第4节通过仿真实验证明了联合优化方案的有效性。最后，第5节对全文进行了总结。

2 系统模型

本文考虑一个两层数字孪生边缘网络(DITEN)，DITEN由物理层和DT层组成。物理层包括配备MEC服务器的AP、传感器以及执行器，三者协同

实现从数据上传、任务计算到结果下载的完整过程。传感器感知环境数据并形成计算任务，采取部分卸载的方式，借助本地和边缘服务器共同完成计算任务。AP利用MEC服务器的强大能力处理卸载任务，整合本地和边缘的计算结果，并将所有结果传输给执行器。执行器接收计算结果后执行相应操作命令。DT层由物理设备以及整个通信环境的数字孪生体共同构成，负责监测物理系统的运行状态，并通过实时交互来优化任务卸载和资源分配方案。系统模型如图1所示。

系统中K个传感器、M个AP和L个执行器分别用集合$k\in\mathcal{K}\triangleq\{1,2,\cdots,K\}$，$m\in\mathcal{M}\triangleq\{1,2,\cdots,M\}$和$l\in\mathcal{L}\triangleq\{1,2,\cdots,L\}$表示。$\pi=\{\pi_{km}\}_{\forall k,m}$表示传感器和AP的关联变量，当传感器$k$和AP m关联时$\pi_{km}=1$，否则$\pi_{km}=0$。此外，假设每一个AP最多只能服务N个传感器，每个传感器只能与一个AP关联。参考文献[18]，本文考虑智能分拣等工业场景，其中分拣设备配备的摄像头等感知设备被视作传感器，分拣设备本身即执行器，因此传感器与执行器之间的配对关系是预先确定的，并且根据特定应用可以灵活扩展为一对多或多对一形式。由于一对一配对，因此当传感器与AP关联之后，执行器与AP的关联也就随之确定了。假设相同索引位置的传感器与执行器配对，因此有$l=k,\forall l\in\mathcal{L},k\in\mathcal{K}$。

传感器k的计算任务由$J_k=\{I_k,C_k,\beta_k,\mu_k\}$，$k\in\mathcal{K}$表示，其中$I_k$表示传感器$k$的任务量(bit)，$C_k$表示执行任务所需的CPU周期数(cycle)，$\beta_k$和$\mu_k$分别表示传感器$k$的本地计算结果占本地任务的比例以及边缘计算结果占卸载任务的比例。$\alpha=\{\alpha_k\},0\le\alpha_k\le1$表示传感器$k$的卸载因子，传感器将$\alpha_kI_k$大小的数据卸载到AP处执行，剩余$(1-\alpha_k)I_k$数据在本地执行。

DT模型依靠边缘服务器强大的计算和存储能力建立和维护。传感器-执行器对(k-l)的DT模型为

$$\mathrm{DT}_{k-l}=\left\{S_k,S_l,f_k,\hat{f}_k\right\}\tag{1}$$

其中S_k表示DT监测传感器k的数字域状态信息，例如本地资源(剩余能量、发射功率等)、信道信息(信噪比、带宽、干扰信息等)等。S_l表示执行器l的数字域状态信息，包括电池电量、活动范围等。f_k表示传感器k的DT估计处理速率，由于DT无法精确复制其物理实体，所以要设置偏差来弥补物理域和数字域之间的差距[7]，$\hat{f}_k$表示实际速率和DT估计速率的偏差。

AP m的DT模型为

$$\mathrm{DT}_m^i=\left(S_m^{\mathrm{AP}},f_{km}^{\mathrm{AP}},\hat{f}_{km}^{\mathrm{AP}}\right)\tag{2}$$

其中S_m^{AP}表示AP m的数字域状态信息，包括关联状态、覆盖范围等。f_{km}^{AP}表示AP m的DT估计处理速率。类似地，为弥补DT与其物理实体的差距，用$\hat{f}_{km}^{\mathrm{AP}}$表示实际速率和DT估计速率的偏差。

2.1　全过程任务模型

在物理系统中，传感器收集环境数据以生成计算任务，采用部分卸载的方式，将任务进行分割，通过本地和边缘共同完成。本地计算结果由传感器上传至AP后，AP整合本地计算结果和边缘计算结果，并将两部分结果发送到相应执行器，由执行器执行后续操作。全过程任务模型由图2所示的5部分组成：

(1)本地计算：传感器根据卸载因子大小，在本地执行部分计算任务。

(2)本地结果上传：本地计算完成后，传感器将本地计算结果上传至AP进行进一步分析或存储。

(3)任务卸载：传感器将部分任务卸载到AP处，利用边缘服务器辅助完成任务计算。

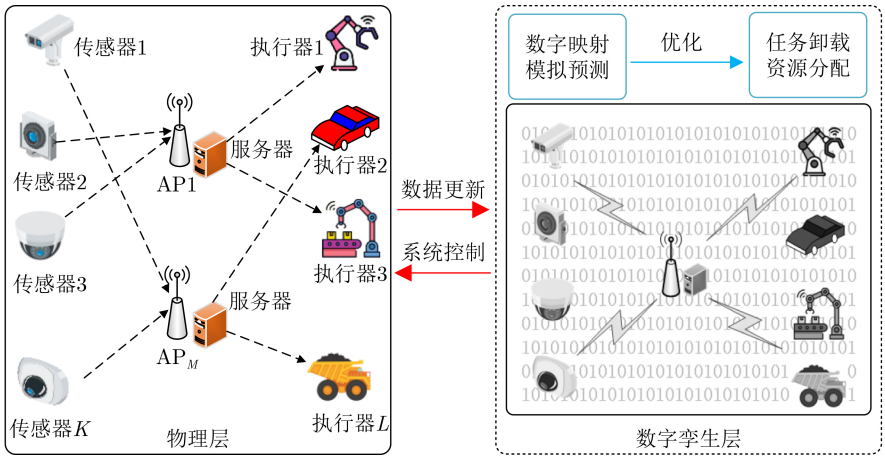

图 1 数字孪生边缘网络(DITEN)系统模型

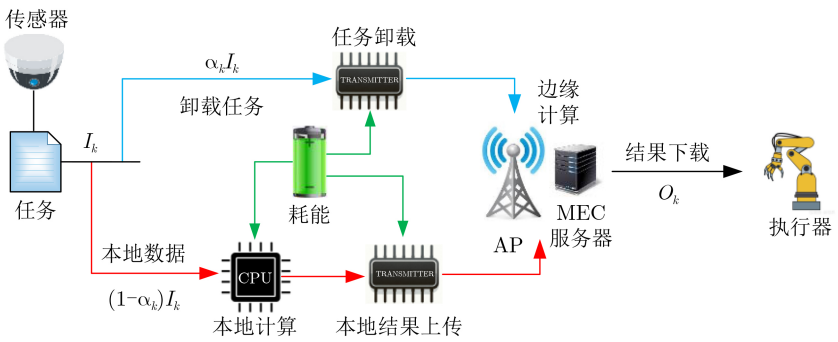

图 2 全过程任务模型

(4)边缘计算：AP利用配备的MEC服务器执行卸载任务。

(5)结果下载：AP整合本地计算结果和边缘计算结果后将所有结果发送至相应执行器。

2.2 通信模型

2.2.1 上行传输

(1)任务卸载。传感器和AP通过无线链路连接，$\bar{b}_k \in [0,B]$表示传感器k卸载任务时分配的带宽。$h_{km}=\beta_0(d_{km}/d_0)^{-3}$表示传感器$k$和AP m之间的信道增益[9]，其中β_0与d_{km}分别表示参考距离d_0处的路径损耗和两者的实际距离。p_k为传感器k的发射功率，模型采用正交频分复用(Orthogonal Frequency Division Multiplexing, OFDM)技术，忽略同信道干扰。因此，从传感器k到AP m的上行传输速率为

$$R_{km}=\bar{b}_k\log_2\left(1+\frac{\pi_{km}p_k|h_{km}|^2}{N_0}\right) \tag{3}$$

其中，N_0表示噪声功率。

因此，从传感器k到AP m用于任务卸载的传输时延可以表示为

$$T_{km}^{\mathrm{up}}=\frac{\pi_{km}\alpha_k I_k}{R_{km}} \tag{4}$$

(2)本地结果上传。$\beta_k\in[0,1)$表示传感器k的本地计算结果占本地任务的比例，即本地计算结果大小为$\beta_k(1-\alpha_k)I_k(\mathrm{bit})$。因此，从传感器$k$到AP m的本地结果上传时延可以表示为

$$T_{\mathrm{lc}}^{\mathrm{up}}=\frac{\pi_{km}\beta_k(1-\alpha_k)I_k}{R_{km}} \tag{5}$$

2.2.2 下行传输

AP将计算结果发送给执行器，$\underline{b}_l\in[0,B]$表示分配给执行器l用于从AP接收计算结果的带宽。与上行传输类似，$h_{ml}=\beta_0(d_{ml}/d_0)^{-3}$表示AP m和执行器l之间的信道增益。AP要在多个执行器之间分配发射功率来为其服务。p_l表示分配给执行器l的功率，AP最大发射功率为$P_{\max}$，需满足$0\le\sum_{l=1}^{L}\pi_{ml}\underline{p}_l\le P_{\max},\forall m$。因此从AP m到执行器l的下行传输速率为

$$R_{ml}=\underline{b}_l\log_2\left(1+\frac{\pi_{ml}\underline{p}_l|h_{ml}|^2}{N_0}\right) \tag{6}$$

其中，N_0表示噪声功率。

AP将本地计算结果与边缘计算结果整合后共同发送给执行器，λ_k表示下载结果占两部分结果之和的比例，因此执行器l需要接收的结果大小为$O_k=\lambda_k(\beta_k(1-\alpha_k)+\mu_k\alpha_k)I_k$。因此，计算结果从AP m发送到执行器l的下行传输时延为

$$T_{ml}^{\mathrm{dw}}=\frac{\pi_{km}O_k}{R_{ml}} \tag{7}$$

2.3 计算模型

2.3.1 本地计算

传感器k以DT估计处理速率f_k执行本地任务，则本地计算的估计时延可表示为

$$\tilde{T}_k^{\mathrm{se}}=\frac{(1-\alpha_k)C_k}{f_k} \tag{8}$$

DITEN通过历史数据可以预先获取传感器k实际处理速率与DT估计速率之间的偏差$\hat{f}_k$，此时实际时延和DT估计时延之间的差值表示为

$$\Delta T_k^{\mathrm{se}}=\frac{(1-\alpha_k)C_k\hat{f}_k}{f_k(f_k-\hat{f}_k)} \tag{9}$$

因此，本地计算的实际时延为

$$T_k^{\mathrm{se}}=\tilde{T}_k^{\mathrm{se}}+\Delta T_k^{\mathrm{se}} \tag{10}$$

2.3.2 边缘计算

AP m以DT估计处理速率f_{km}^{AP}执行传感器k的卸载任务，则估计处理时延为

$$\tilde{T}_{km}^{\mathrm{AP}}=\frac{\pi_{km}\alpha_k C_k}{f_{km}^{\mathrm{AP}}} \tag{11}$$

从DITEN中得到AP计算资源偏差后，实际处理时延和DT估计处理时延的差值可以表示为

$$\Delta T_{km}^{\mathrm{AP}}=\frac{\pi_{km}\alpha_k C_k \hat{f}_k^{\mathrm{AP}}}{f_{km}^{\mathrm{AP}}(f_{km}^{\mathrm{AP}}-\hat{f}_k^{\mathrm{AP}})} \tag{12}$$

因此，AP计算卸载任务的实际时延为

$$T_{km}^{\mathrm{AP}}=\tilde{T}_{km}^{\mathrm{AP}}+\Delta T_{km}^{\mathrm{AP}} \tag{13}$$

2.4 能耗与时延模型

通信能耗。通信能耗由任务卸载能耗和本地结果上传能耗两部分组成。因此，传感器k的通信能耗可以表示为

$$E_k^{\mathrm{cm}}=\left(T_{km}^{\mathrm{up}}+T_{\mathrm{lc}}^{\mathrm{up}}\right)p_k \tag{14}$$

计算能耗。AP有充足的能源供应，可以忽略边缘计算能耗[9,14]。因此，传感器k的计算能耗即本地计算能耗，可表示为

$$E_k^{\mathrm{cp}}=(1-\alpha_k)\frac{\xi_k}{2}C_k(f_k-\hat{f}_k)^2 \tag{15}$$

其中，ξ_k是由芯片架构决定的有效电容系数[7]。

因此，传感器k的总能耗可计算为

$$E_k^{\mathrm{tot}}=E_k^{\mathrm{cm}}+E_k^{\mathrm{cp}} \tag{16}$$

由全过程任务模型可知，传感器k的任务端到端时延由5部分组成，用$T_k^{\mathrm{E2E}},\forall k$表示。由于通信资源的约束，本地结果上传和任务卸载无法并行传输[9]，本地结果必须等待任务卸载完成后才能上传，因此就导致了端到端时延会出现如图3所示的4种情况，图中$\tau_1,\tau_2,\tau_3,\tau_4,\tau_5$分别表示本地计算、本地结果上传、任务卸载、边缘计算和结果下载的时延。

根据本地计算时延和任务卸载时延之间的关系，将上述4种情况概括为两类：

(1)本地计算时延大于任务卸载时延

如图3(a)和图3(b)所示，$\tau_1>\tau_3$。此时传感器k卸载完任务后，信道有一段时间空闲，需要空闲到本地计算结束后才能将结果上传到AP。此时传感器k的端到端时延为$T_k^{\mathrm{E2E}}=\max\{(\tau_1+\tau_2),(\tau_3+\tau_4)\}+\tau_5$。

(2)本地计算时延小于任务卸载时延

如图3(c)和图3(d)所示，$\tau_1<\tau_3$。此时传感器k本地计算完成后不能立刻将计算结果上传，需要等到任务卸载完成后，才能将本地计算结果上传到AP。因此，这种情况下传感器k的端到端时延为$T_k^{\mathrm{E2E}}=\tau_3+\max\{\tau_2,\tau_4\}+\tau_5$。

综上两种情况，针对传感器k而言，其总的端到端时延可以表示为

$$T_k^{\mathrm{E2E}}=\max\left\{\tau_3+\tau_4,\max\left(\tau_1,\tau_3\right)+\tau_2\right\}+\tau_5 \tag{17}$$

基于以上分析，传感器k的任务J_k从卸载、执行到结果下载的端到端时延可以表示为

$$\begin{aligned}T_k^{\mathrm{E2E}}&=\sum\max\{\tau_3+\tau_4,\max(\tau_1,\tau_3)+\tau_2\}+\tau_5\\&=\sum\max\{T_{km}^{\mathrm{up}}+T_{km}^{\mathrm{AP}},\max(T_k^{\mathrm{se}},T_{km}^{\mathrm{up}})+T_{lc}^{\mathrm{up}}\}\\&\quad+T_{ml}^{\mathrm{dw}}\\&=\sum_{m=1}^{M}\left\{\max\left[\frac{\pi_{km}\alpha_k I_k}{R_{km}}+\frac{\pi_{km}\alpha_k C_k}{f_{km}^{\mathrm{AP}}-\hat{f}_{km}^{\mathrm{AP}}},\right.\right.\\&\quad\max\left(\frac{(1-\alpha_k)C_k}{f_k-\hat{f}_k},\frac{\pi_{km}\alpha_k I_k}{R_{km}}\right)\\&\quad\left.\left.+\frac{\pi_{km}\beta_k\left(1-\alpha_k\right)I_k}{R_{km}}\right]+\frac{\pi_{km}O_k}{R_{ml}}\right\}\end{aligned} \tag{18}$$

2.5 问题建模

本文考虑了任务执行的全过程，综合考虑5阶段时延，时延构成复杂且存在图3所示的时延冲突，因此优化时延在本文框架下更有价值。所以，本文通过联合优化设备关联，上/下行带宽，AP发射功率，卸载因子，以及本地与边缘的估计处理速率$\boldsymbol{f}=\{f_k,f_{km}^{\mathrm{AP}}\}$来最小化系统总的端到端时延。优化问题表述为

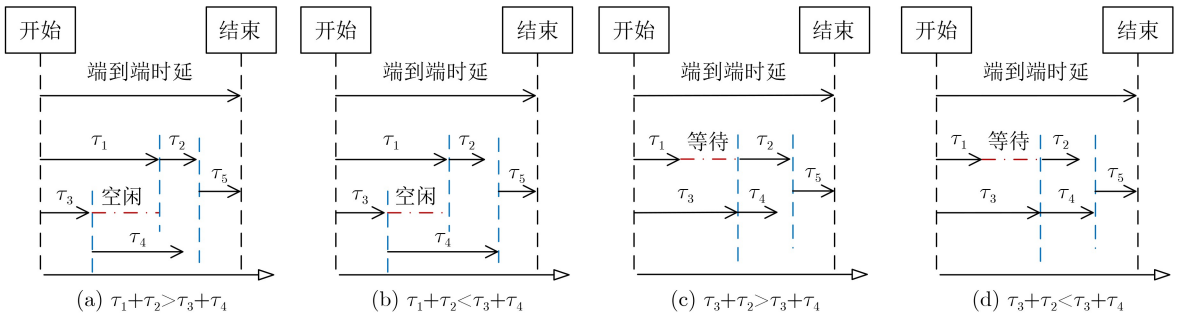

图 3 端到端时延示意图

$$\min_{\pi,\alpha,\bar{b},\underline{b},p,\boldsymbol{f}} \sum_{k=1}^{K}\sum_{m=1}^{M} T^{\mathrm{E2E}}(\pi,\alpha,\bar{b},\underline{b},p,\boldsymbol{f})$$
$$\text{s.t.}\begin{cases}\mathrm{C1}: T_k^{\mathrm{E2E}}(\pi,\alpha,\bar{b},\underline{b},\underline{p},\boldsymbol{f}) \le T_{\max}, \forall k\in\mathcal{K}\\ \mathrm{C2}: E_k^{\mathrm{tot}}\left(\pi,\alpha,\bar{b},f_k\right) \le E_{\max}, \forall k\in\mathcal{K}\\ \mathrm{C3}: \sum_{k=1}^{K}\pi_{km} \le N, \forall m\in\mathcal{M}\\ \mathrm{C4}: \sum\nolimits_{k=1}^{K}\bar{b}_k \le B, \sum\nolimits_{l=1}^{L}\underline{b}_l \le B\\ \mathrm{C5}: 0 \le \sum_{m=1}^{M}\pi_{km}p_l \le P_{\max}, \forall m\in\mathcal{M}\\ \mathrm{C6}: \pi_{km}\in\{0,1\}, \sum\nolimits_{m=1}^{M}\pi_{km}=1, \forall k\in\mathcal{K},\\ \quad m\in\mathcal{M}\\ \mathrm{C7}: 0\le\alpha_k\le 1, \forall k\in\mathcal{K}\\ \mathrm{C8}: f_k \le F_{\max}^{\mathrm{se}}, f_{km}^{\mathrm{AP}} \le F_{\max}^{\mathrm{AP}}, \forall k\in\mathcal{K}, m\in\mathcal{M}\\ \mathrm{C9}: \bar{b}_k, \underline{b}_l, f_k, f_{km}^{\mathrm{AP}} \ge 0, \forall k\in\mathcal{K}, l\in\mathcal{L}, m\in\mathcal{M}\end{cases} \tag{19}$$

问题式(19)中C1和C2为时延和能耗约束；C3约束AP的服务数量；C4约束带宽取值；C5约束AP功率取值；C6为设备关联约束；C7约束卸载因子取值；C8约束计算频率取值；C9为变量的非负约束。

3 问题求解

问题式(19)的目标函数和约束条件非凸非光滑，且二进制变量与连续变量耦合，导致问题式(19)是一个混合整数非凸问题，难以求解。因此，本文采取变量解耦的方法，将原问题分解为计算和通信资源优化、设备关联优化、卸载决策优化以及传输带宽优化4个子问题。然后将内部凸近似方法(ICA)和匈牙利算法(HA)结合，提出了一种交替优化(AO)算法，即ICA-HA-AO算法，迭代地解决原问题。

3.1 计算和通信资源优化

在该子问题中，给定$(\alpha^{(i)},\pi^{(i)},\bar{b}^{(i)},\underline{b}^{(i)})$值的条件下求解问题式(19)，找到下一个最优传感器计算频率以及AP计算频率和发射功率$(\boldsymbol{f}^{(i+1)},\underline{p}^{(i+1)})$。此时，问题式(19)转换成了计算和通信资源优化子问题SP1。

$$\left.\begin{array}{l}\min\limits_{\boldsymbol{f},p|\alpha^{(i)},\pi^{(i)},\bar{b}^{(i)},\underline{b}^{(i)}} \sum\limits_{k=1}^{K}\sum\limits_{m=1}^{M} T^{\mathrm{E2E}}(\boldsymbol{f},p)\\ \text{s.t.}\ \ \mathrm{C1,C3,C5,C8,C9}\end{array}\right\} \tag{20}$$

SP1中约束C1是非凸的，接下来通过内部凸近似[7]方法，将子问题SP1转换为凸优化问题。

凸化约束C1：定义变量$\underline{R}\triangleq\{\underline{R}_{ml}\},\forall m,l$，满足$1/\underline{R}_{ml}(\underline{p})\le\underline{R}_{ml}$，有

$$\begin{aligned}&T_k^{\mathrm{E2E}}(\boldsymbol{f},p)\ge\underline{T}_k^{\mathrm{E2E}}(\boldsymbol{f},\underline{R})\\ &=\sum_{m=1}^{M}\left\{\max\left[\frac{\pi_{km}\alpha_k I_k}{R_{km}}+\frac{\pi_{km}\alpha_k C_k}{f_{km}^{\mathrm{AP}}-\hat{f}_{km}^{\mathrm{AP}}},\right.\right.\\ &\left.\left.\max\left(\frac{(1-\alpha_k)C_k}{f_k-\widehat{f}_k},\frac{\pi_{km}\alpha_k I_k}{R_{km}}\right)\right]+\pi_{km}O_k\underline{R}_{ml}\right\}\end{aligned} \tag{21}$$

则约束C1可以改写为

$$\underline{T}_k^{\mathrm{E2E}}(\boldsymbol{f},\underline{R})\le T_{\max} \tag{22}$$

$$1/\underline{R}_{ml}^{(i)}(\underline{b}^{(i)},p,\pi^{(i)})\le\underline{R}_{ml},\forall m,l \tag{23}$$

此时约束式(22)和式(23)均为凸约束。因此，问题SP1转化成为一个凸问题式(24)。

$$\left.\begin{array}{l}\min\limits_{\boldsymbol{f},p,\underline{R}|\alpha^{(i)},\pi^{(i)},\bar{b}^{(i)},\underline{b}^{(i)}} \sum\limits_{k=1}^{K}\sum\limits_{m=1}^{M}\underline{T}^{\mathrm{E2E}}(\boldsymbol{f},\underline{R})\\ \text{s.t.}\ \ \text{式(22)},\text{式(23)},\mathrm{C3,C5,C8,C9}\end{array}\right\} \tag{24}$$

问题式(24)是一个凸问题，可以通过内点法等凸优化算法解决。求解子问题SP1的步骤见算法1。

3.2 设备关联优化

该子问题在给定$(\alpha^{(i+1)},\boldsymbol{f}^{(i+1)},\underline{p}^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)})$值的条件下求解问题式(19)，来得到最优的传感器与AP关联策略$(\pi^{(i+1)})$。此时，问题式(19)转换成了设备关联优化子问题SP2。

$$\left.\begin{array}{l}\min\limits_{\pi|\alpha^{(i)},\boldsymbol{f}^{(i+1)},p^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)}} \sum\limits_{k=1}^{K}\sum\limits_{m=1}^{M} T^{\mathrm{E2E}}(\pi)\\ \text{s.t.}\ \ \mathrm{C1}\sim\mathrm{C3,C5,C6}\end{array}\right\} \tag{25}$$

由于问题SP2中约束C3, C5, C6是非凸的，且C6中存在二进制变量，导致该问题是NP难的。为解决SP2，可以将二进制变量松弛为连续变量，并将非凸问题转化为凸问题。但松弛约束会使一个传

算法 1 基于内部凸近似的计算与通信资源分配算法

输入：卸载因子α，关联变量π，传输带宽$\bar{b},\underline{b}$，发射功率p以及计算频率$\boldsymbol{f}$；

初始化：迭代次数$i=0$，容忍值ε，最大迭代次数$I_{\max}$

输出：最优功率和计算频率$(p^*,\boldsymbol{f}^*)$

(1) **while 1 do**

(2) 求解问题式(24)的可行解$(p^{(i+1)},\boldsymbol{f}^{(i+1)})$；

(3) $i=i+1$

(4) **if** 收敛**or** $i>I_{\max}$ **then**

(5) **break;**

(6) **end if**

(7) **end while**

感器与多个AP关联，无法满足约束C6。此外，暴力搜索的复杂度过高，在大规模场景下并不适用。相比之下，匹配算法由于其低复杂度等优势，能有效解决该问题。在众多匹配算法中，匈牙利算法相对于霍普克洛夫特-卡普算法等其他算法，实现简单且复杂度低，更适用于解决无权二分图最大匹配问题。因此，本文采用匈牙利算法解决关联问题。但该算法只适用于匹配双方数量相等的一对一匹配问题，而SP2不满足条件。因此，本文采取基于加边补零法的二分图最优匹配方法，通过引入虚拟AP和传感器来解决该问题。

将传感器与AP关联用无向二分图$G=(V,E)$来描述，其中V表示节点集$V=\mathcal{K}\cup\mathcal{M}$，$E$表示边集。约束C2和C6要求传感器集合$\mathcal{K}$中的每一个节点只能与一条边关联，AP集合$\mathcal{M}$中的节点可以与最多$N$条边关联，此时$\mathcal{K}$中的顶点与$\mathcal{M}$中的顶点是多对一关系，相应二分图如图4(a)所示。

为使关联问题满足匈牙利算法的条件，本文通过将AP m虚拟复制的方式得到虚拟AP集合$\mathcal{M}^{(m)},m=1,2,\cdots,M$，由约束C2可得虚拟AP m集合大小为$\left|\mathcal{M}^{(m)}\right|=N$，即每个AP需要虚拟复制$N-1$个。此时关联关系变为AP与单个传感器关联。传感器与虚拟AP关联可以用一个新的无向二分图$G'=(V',E')$表示，其中$V'=\left(\cup_{m=1}^{M}\mathcal{M}^{(m)}\right)\cup\mathcal{K}$，因此关联问题转换成了图4(b)所示的二分图匹配问题。

图4(b)中假设每个AP最多服务3个传感器，因此每一个AP需要虚拟复制2个，例如AP1用虚拟AP集合$M^{(1)}=\{M_1^{(1)},M_2^{(1)},M_3^{(1)}\}$表示。此时，图4(a)中的3个传感器$(K_1,K_2,K_3)$和一个AP$(M_1)$关联，转换成了图4(b)中的3个传感器$(K_1,K_2,K_3)$分别和3个虚拟AP$(M_1^{(1)},M_2^{(1)},M_3^{(1)})$关联。经转换后虚拟AP的数量为$N\times M$。当传感器数量小于虚拟AP数量时$(K\le NM)$，通过添加虚拟传感器节点，使之转换为数量平衡的指派问题，共需要添加$NM-K$个虚拟传感器。经过上述转换后得到一个如图4(c)所示的数量平衡的一对一匹配问题。设虚拟传感器集合为$\mathcal{K}'=\{1,\cdots,K,\cdots,NM\}$，此时子问题SP2等价于问题式(26)。

$$\min_{\pi|\alpha^{(i)},\boldsymbol{f}^{(i+1)},p^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)}}\sum_{\forall k\in\mathcal{K}'}\sum_{\forall m\in\mathcal{M}^{(m)}}T^{\mathrm{E2E}}(\pi)$$
$$\text{s.t.}\begin{cases}\mathrm{C1}:T_k^{\mathrm{E2E}}(\pi)\le T_{\max},\forall k\in\mathcal{K}'\\ \mathrm{C2}:E_k^{\mathrm{tot}}\left(\pi,\alpha,\bar{b},f_k\right)\le E_{\max},\forall k\in\mathcal{K}'\\ \mathrm{C3}:0\le\sum_{l=1}^{L}\pi_{y(l)m}p_l\le P_{\max},\forall m\in\mathcal{M}^{(m)}\\ \mathrm{C4}:\pi_{km}\in\{0,1\},\forall k\in\mathcal{K}',m\in\mathcal{M}^{(m)}\end{cases}\tag{26}$$

基于匈牙利算法解决设备关联子问题SP2的实现步骤如算法2所示。

3.3　卸载决策优化

该子问题在固定$(\pi^{(i+1)},\boldsymbol{f}^{(i+1)},\underline{p}^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)})$取值的条件下，寻找下一个最优的卸载决策值$(\alpha^{(i+1)})$。此时，问题式(19)转换成了卸载决策优化子问题SP3。

$$\left.\begin{aligned}&\min_{\alpha^{(i+1)}|\pi^{(i+1)},\boldsymbol{f}^{(i+1)},p^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)}}\sum_{k=1}^{K}\sum_{m=1}^{M}T^{\mathrm{E2E}}(\alpha)\\&\text{s.t.}\ \ \mathrm{C1},\mathrm{C3},\mathrm{C7}\end{aligned}\right\}\tag{27}$$

算法 2　基于匈牙利算法的传感器与AP关联优化算法

输入：卸载因子α，传输带宽$\bar{b},\underline{b}$，发射功率p以及计算频率$\boldsymbol{f}$；

输出：最优关联策略(π^*)

(1)给定传感器数量K，AP数量M，AP能够服务传感器的最大数量N；

(2)每个AP虚拟成N个，形成数量为NM的虚拟AP集合；

(3)**if** $K<NM$ **then**

(4)　加边补零，增加$NM-K$个虚拟传感器；

(5)**end if**

(6)执行匈牙利算法。

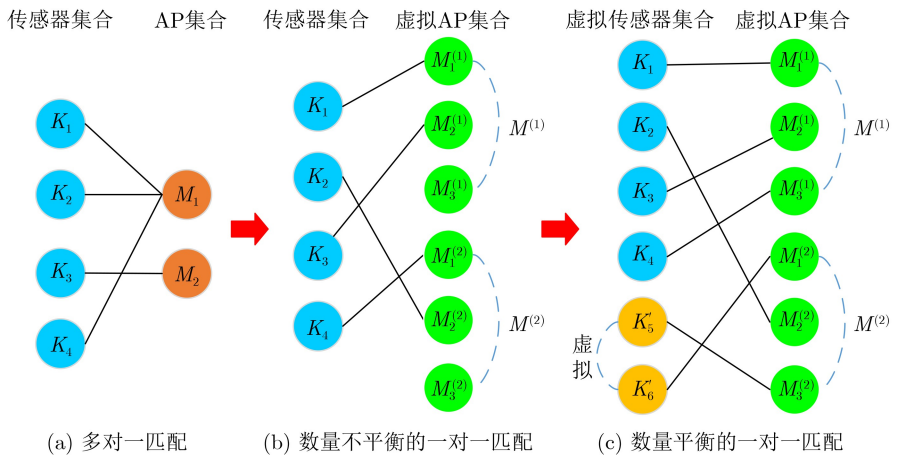

图 4 基于加边补零法的二分图最优匹配过程

子问题SP3是一个线性规划问题，可以很方便地采用线性规划方法解决。

3.4 传输带宽优化

在该子问题中，给定$(\pi^{(i+1)},\alpha^{(i+1)},\boldsymbol{f}^{(i+1)},\underline{p}^{(i+1)})$取值的条件下，来寻找下一个最优的上行和下行带宽取值$(\bar{b}^{(i+1)},\underline{b}^{(i+1)})$。此时，问题式(19)转换成了传输带宽优化子问题SP4。

$$\left.\begin{aligned}&\min_{\bar{b}^{(i+1)},\underline{b}^{(i+1)}|\pi^{(i+1)},\alpha^{(i+1)},\boldsymbol{f}^{(i+1)},p^{(i+1)}}\sum_{k=1}^{K}\sum_{m=1}^{M}T^{\mathrm{E2E}}(\bar{b},\underline{b})\\&\text{s.t.}\ \ \mathrm{C1},\mathrm{C3},\mathrm{C4},\mathrm{C6}\end{aligned}\right\}\tag{28}$$

从问题SP4可以看出，其目标函数是一个凸函数，约束条件为凸集，因此该问题是一个凸优化问题。可以采用内点法等凸优化算法进行有效求解。

3.5 ICA-HA-AO算法

本小节基于对上述4个子问题的求解，提出一种基于内部凸近似方法和匈牙利算法的交替优化算法(ICA-HA-AO)，有效解决了任务端到端时延最小化问题。设问题式(19)的最优解为$(\alpha^*,\pi^*,\bar{b}^*,\underline{b}^*,p^*,\boldsymbol{f}^*)$，ICA-HA-AO算法的具体实现过程如算法3所示。

4 仿真结果与分析

4.1 参数设置与基准方案

本文考虑一个半径为200 m的区域，其中两个AP的坐标为(50, 50)和(–50, 50)，每个AP的最大服务数量$N=6$，覆盖半径为150 m, 10个传感器-执行器对在区域内随机分布。其他重要参数由表1给出。

为了评估所提联合优化方案的有效性，本文将所提算法与下面3种基准方案进行比较。

(1)方案1[6]：基于序列凸逼近方法的通信效率优先方案。该方案优先考虑通信过程优化，假设传感器和AP的处理速率均为满足能量约束下的最大值。

(2)方案2[9]：基于凸差算法(Difference of Convex, DC)的计算效率优先方案。该方案确定AP的发射功率，优先考虑计算资源的最优分配。

(3)方案3[19]：基于交替优化(AO)算法的无本地计算方案。该方案假设传感器的任务全部卸载到AP处执行，研究卸载因子对端到端时延的影响。

4.2 仿真结果与讨论

图5展示了任务所需计算资源与端到端时延的关系。本文方案性能优于基准方案，例如C_k为

算法 3 基于内部凸近似方法和匈牙利算法的ICA-HA-AO算法

输入：卸载因子$\alpha^{(0)}$，关联变量$\pi^{(0)}$，带宽$\bar{b}^{(0)},\underline{b}^{(0)}$，发射功率$p^{(0)}$，计算频率$\boldsymbol{f}^{(0)}$，设置迭代次数$i=0$，容忍值$\varepsilon$和最大迭代次数$I_{\max}$；

输出：最优资源分配$(\alpha^*,\pi^*,\bar{b}^*,\underline{b}^*,p^*,\boldsymbol{f}^*)$

while 1 do

  给定$(\alpha^{(i)},\pi^{(i)},\bar{b}^{(i)},\underline{b}^{(i)})$，利用算法1求解子问题SP1，得到最优传感器计算频率以及AP的计算频率和发射功率$(p^{(i+1)},\boldsymbol{f}^{(i+1)})$；

  给定$(\boldsymbol{f}^{(i+1)},p^{(i+1)},\alpha^{(i)},\bar{b}^{(i)},\underline{b}^{(i)})$，利用算法2求解子问题SP2，得到最优传感器与AP关联策略$(\pi^{(i+1)})$；

  给定$(\boldsymbol{f}^{(i+1)},p^{(i+1)},\pi^{(i+1)},\bar{b}^{(i)},\underline{b}^{(i)})$，利用内点法求解子问题SP3，得到最优卸载决策$(\alpha^{(i+1)})$；

  给定$(\boldsymbol{f}^{(i+1)},p^{(i+1)},\pi^{(i+1)},\alpha^{(i)})$，利用内点法求解子问题SP4，得到最优上下行带宽$(\bar{b}^{(i+1)},\underline{b}^{(i+1)})$；

  $i=i+1$

  if 收敛 **or** $i>I_{\max}$ **then**

    break;

  end if

end while

表 1 仿真参数

参数名	参数值	参数名	参数值
传感器发射功率$\bar{p}_k$	10 dBm[8]	传感器最大计算频率$F_{\max}^{\mathrm{se}}$	3 GHz[7]
AP最大发射功率$P_{\max}$	40 dBm[18]	AP最大计算频率$F_{\max}^{\mathrm{AP}}$	10 GHz[7]
输入任务量I_k	100 Mbit[18]	路径损耗β_0	–30 dB[14]
任务所需计算资源C_k	960×10^6 cycle[7]	参考距离d_0	10 m[9]
传输总带宽B	20 MHz[18]	噪声功率N_0	–174 dBm/Hz[18]
最大时延与能耗$T_{\max},E_{\max}$	2 s, 1.5 J	有效交换电容系数ξ	10^{-28}[9]

70 M cycle时，所提联合优化方案比方案3的时延降低了约0.8 s，比方案1和方案2的时延分别降低了约0.3 s和0.5 s。此外，随着任务所需计算资源的增加，所有方案的时延均会增加，这是因为更高的计算资源不仅会增加本地和边缘的处理时延，还会迫使传感器降低其处理速率以满足能耗约束，从而导致时延进一步增加。

图6显示了传输带宽与端到端时延的关系。可以看出，所提联合优化方案在降低端到端时延方面明显优于其他3种基准方案，而无本地计算的方案3表现最差，尤其当带宽较小时性能的提升更为明显，而随着带宽的增大性能提升略有下降。这是因为带宽大小会影响传输速率，而任务卸载时延受带宽影响较大，因此考虑带宽优化的联合优化方案与其他算法相比时延最多降低了约1.5 s，反映了算法的有效性。

图7展示了边缘服务器处理速率对端到端时延的影响。4种方案的时延均随处理速率的增加而减小，所提联合优化方案性能优于其他3种基准方案。例如当处理速率从10 G cycle/s提升至50 G cycle/s时，所提联合方案的端到端时延下降幅度最大。这是因为AP处理速率增加意味着边缘处理能力增强，就会导致更多的任务选择卸载到边缘处理，这也解释了为什么方案3的性能最接近联合优化方案。

图8显示了任务量与端到端时延之间的关系。所提方案的时延低于其他3种方案。当任务量为9 Mbit时，所提方案的时延比基准方案中最差(方案1)的时延降低了约1.2 s，这是因为只有联合优化方案同时考虑了计算和通信资源分配。此外，方案1受任务量的影响最大，最好与最坏情况相差约1.3 s，这是因为固定计算频率后，在能耗等约束下，边缘处理的时间显著增加，从而导致时延的增加。由于完全卸载后任务都集中在边缘处理，而边缘处理速率要远高于本地处理速率，所以任务量对方案3的影响最小。

图9研究了处理速率的DT偏差对端到端时延的影响。随着DT偏差的增大，端到端时延也增大，该结果符合式(9)和式(12)。其中，当偏差为0.01时系统性能最优，而随着偏差值的增大时延的增幅也越来越大，例如当偏差从0.01增大到0.02时和从0.04增大到0.05时，时延分别增加了约0.13 s和0.23 s。未来随着DT技术的发展，实际值与估计值的偏差一定会逐渐缩小，数字世界会更加接近物理世界。

图10展示了本文方案在不同AP和设备数量下的性能表现。当设备数量相同AP数量递增时，时延不断下降。这是因为当AP较多时，其服务的设备数量减少，设备分配到的资源就更多，从而导致时延降低。但是过度增加AP数量不仅会导致资源浪费，还会增加关联问题的复杂性。因此，在实际部署时应充分权衡AP数量。当AP数量相同设备数

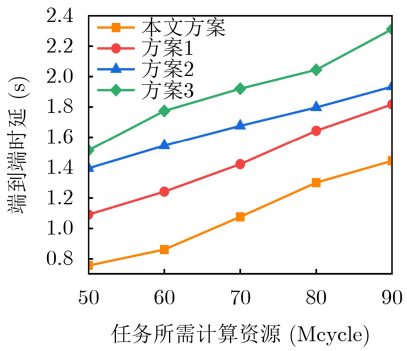

图 5 任务所需计算资源与端到端时延的关系

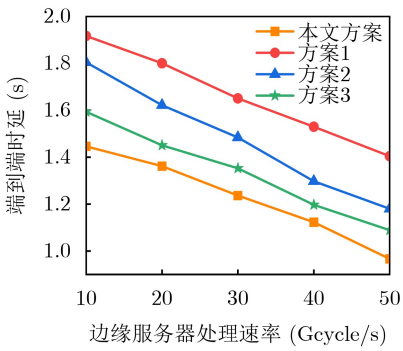

图 7 服务器处理速率与端到端时延的关系

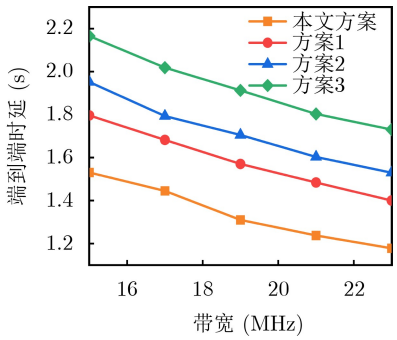

图 6 传输带宽与端到端时延的关系

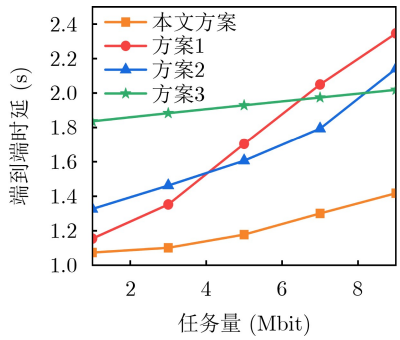

图 8 任务量与端到端时延的关系

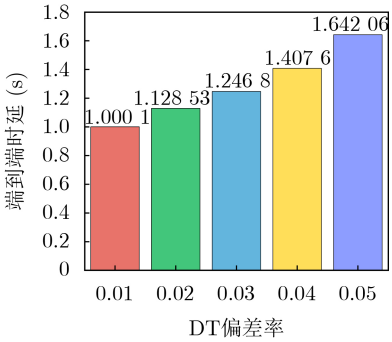

图 9 不同DT偏差对端到端时延的影响

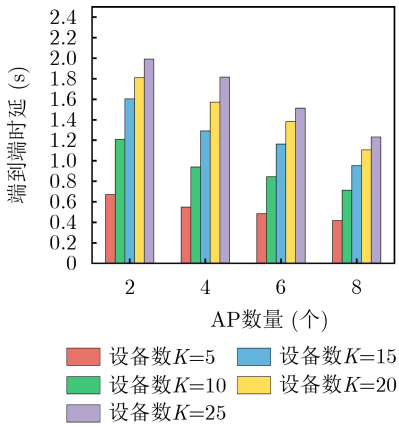

图 10 设备数量不同的端到端时延对比

量递增时，时延也随之增加。但本文方案能够有效控制时延的增幅。例如当AP数为2设备数设置为10, 15和20时，时延增幅从0.39 s降到0.21 s。这是因为在不超过AP负载能力的前提下增加设备时，联合优化方案能够更充分地调度资源，实现更多任务的并行处理。

5 结论

本文研究了数字孪生边缘网络中任务卸载、设备关联与资源分配联合优化问题。首先，建立了数字域DT模型以及任务模型、端到端时延模型等物理域模型，并通过联合优化设备关联、卸载比例以及计算和通信资源分配实现端到端时延最小化。其次，将所提混合整数非凸问题分解为计算和通信资源优化、设备关联优化、卸载决策优化以及传输带宽优化4个子问题，然后基于内部凸近似方法和匈牙利算法提出了ICA-HA-AO算法。最后，仿真结果表明，本文所提方案与参考方案相比能够显著降低端到端时延。在未来，鉴于频谱资源的稀缺性，本文建立的优化问题和所提算法通过适当的变形和拓展，可与MIMO, NOMA等提高频谱资源利用率的技术结合，进一步提高数字孪生边缘网络系统的频谱效率和整体性能。

参 考 文 献

[1] DJIGAL H, XU Jia, LIU Linfeng, *et al.* Machine and deep learning for resource allocation in multi-access edge computing: A survey[J]. *IEEE Communications Surveys & Tutorials*, 2022, 24(4): 2449–2494. doi: 10.1109/COMST.2022.3199544.

[2] MACH P and BECVAR Z. Mobile edge computing: A survey on architecture and computation offloading[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628–1656. doi: 10.1109/COMST.2017.2682318.

[3] WANG Zhiying, SUN Gang, SU Hanyue, *et al.* Low-latency scheduling approach for dependent tasks in MEC-enabled 5G vehicular networks[J]. *IEEE Internet of Things Journal*, 2024, 11(4): 6278–6289. doi: 10.1109/JIOT.2023.3309940.

[4] DENG Xiaoheng, YIN Jian, GUAN Peiyuan, *et al.* Intelligent delay-aware partial computing task offloading for multiuser industrial internet of things through edge computing[J]. *IEEE Internet of Things Journal*, 2023, 10(4): 2954–2966. doi: 10.1109/JIOT.2021.3123406.

[5] MEI Jing, TONG Zhao, LI Kenli, *et al.* Energy-efficient heuristic computation offloading with delay constraints in mobile edge computing[J]. *IEEE Transactions on Services Computing*, 2023, 16(6): 4404–4417. doi: 10.1109/TSC.2023.3324604.

[6] ZHANG Haibo, LIU Xiangyu, XU, Yongjun, *et al.* Partial offloading and resource allocation for MEC-assisted vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2024, 73(1): 1276–1288. doi: 10.1109/TVT.2023.3306939.

[7] VAN HUYNH D, NGUYEN V D, KHOSRAVIRAD S R, *et al.* URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach[J]. *IEEE Transactions on Communications*, 2022, 70(11): 7669–7682. doi: 10.1109/TCOMM.2022.3205692.

[8] VAN HUYNH D, NGUYEN V D, CHATZINOTAS S, *et al.* Joint communication and computation offloading for ultra-reliable and low-latency with multi-tier computing[J]. *IEEE Journal on Selected Areas in Communications*, 2023, 41(2): 521–537. doi: 10.1109/JSAC.2022.3227088.

[9] LI Song, SUN Weibin, SUN Yanjing, *et al.* Energy-efficient task offloading using dynamic voltage scaling in mobile edge computing[J]. *IEEE Transactions on Network Science and Engineering*, 2021, 8(1): 588–598. doi: 10.1109/TNSE.2020.3046014.

[10] ZHANG Yongchao, HU Jia, and MIN Geyong. Digital twin-driven intelligent task offloading for collaborative mobile edge computing[J]. *IEEE Journal on Selected Areas in Communications*, 2023, 41(10): 3034–3045. doi: 10.1109/

JSAC.2023.3310058.

[11] 唐伦, 单贞贞, 文明艳, 等. 工业物联网中数字孪生辅助任务卸载算法[J]. 电子与信息学报, 2024, 46(4): 1296–1305. doi: 10.11999/JEIT230317.
TANG Lun, SHAN Zhenzhen, WEN Mingyan, *et al.* Digital twin-assisted task offloading algorithms for the industrial internet of things[J]. *Journal of Electronics & Information Technology*, 2024, 46(4): 1296–1305. doi: 10.11999/JEIT230317.

[12] TANG Fengxiao, CHEN Xuehan, RODRIGUES T K, *et al.* Survey on digital twin edge networks (DITEN) toward 6G[J]. *IEEE Open Journal of the Communications Society*, 2022, 3: 1360–1381. doi: 10.1109/OJCOMS.2022.3197811.

[13] 张彦, 卢云龙. 数字孪生边缘网络[J]. 中兴通讯技术, 2023, 29(3): 21–25. doi: 10.12142/ZTETJ.202303005.
ZHANG Yan and LU Yunlong. Digital twin edge networks[J]. *ZTE Technology Journal*, 2023, 29(3): 21–25. doi: 10.12142/ZTETJ.202303005.

[14] 苏健, 钱震, 李斌. 数字孪生使能的智能超表面边缘计算网络任务卸载[J]. 电子与信息学报, 2022, 44(7): 2416–2424. doi: 10.11999/JEIT220180.
SU Jian, QIAN Zhen, and LI Bin. Digital twin empowered task offloading for RIS-assisted edge computing networks[J]. *Journal of Electronics & Information Technology*, 2022, 44(7): 2416–2424. doi: 10.11999/JEIT220180.

[15] DAI Yueyue, ZHANG Ke, MAHARJAN S, *et al.* Deep reinforcement learning for stochastic computation offloading in digital twin networks[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(7): 4968–4977. doi: 10.1109/TII.2020.3016320.

[16] LU Yunlong, MAHARJAN S, and ZHANG Yan. Adaptive edge association for wireless digital twin networks in 6G[J]. *IEEE Internet of Things Journal*, 2021, 8(22): 16219–16230. doi: 10.1109/JIOT.2021.3098508.

[17] SUN Wen, ZHANG Haibin, WANG Rong, *et al.* Reducing offloading latency for digital twin edge networks in 6G[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(10): 12240–12251. doi: 10.1109/TVT.2020.3018817.

[18] SUN Yaping, XU Jie, and CUI Shuguang. Joint user association and resource allocation optimization for MEC-enabled IoT networks[C]. Proceedings of the ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, 2022: 4884–4889. doi: 10.1109/ICC45855.2022.9839276.

[19] DONG Rui, SHE Changyang, HARDJAWANA W, *et al.* Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin[J]. *IEEE Transactions on Wireless Communications*, 2019, 18(10): 4692–4707. doi: 10.1109/TWC.2019.2927312.

李　松：男，副教授，研究方向为工业物联网、边缘计算等.
李　顺：男，硕士生，研究方向为移动边缘计算、数字孪生等.
王博文：男，副教授，研究方向为工业物联网、社交物联网等.
孙彦景：男，教授，研究方向为工业物联网、无线资源分配等.

责任编辑：陈　倩

Task Offloading and Resource Allocation Method for End-to-End Delay Optimization in Digital Twin Edge Networks

LI Song[①②]　LI Shun[①]　WANG Bowen[①]　SUN Yanjing[①②]

[①](*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou* 221116, *China*)

[②](*Xuzhou Engineering Research Center of Intelligent Industry Safety and Emergency Collaboration, Xuzhou* 221116, *China*)

Abstract:

Objective　The rapid development of wireless communication and the Internet of Things (IoT) has led to significant growth in compute-intensive and delay-sensitive applications, which impose stricter latency requirements. However, local devices often face challenges in meeting these demands due to limitations in storage, computing power, and battery life. Mobile Edge Computing (MEC) has emerged as a key technology to address these issues. Despite its potential, the dynamic and complex nature of edge networks presents significant challenges in task offloading and resource allocation. DIgital Twin Edge Networks (DITEN), which map digital twins to physical devices in real-time, offer a promising solution. By integrating MEC with Digital Twin (DT) technology, this approach not only alleviates resource limitations in devices but also optimizes resource allocation in the digital domain, minimizing physical resource waste. This paper tackles the End-to-

End (E2E) optimization problem in the offloading, computation, and result feedback process within edge computing networks. A DT-assisted joint task offloading, device association, and resource allocation scheme is proposed for E2E delay optimization, providing theoretical support for improving resource utilization in edge networks.

Methods The optimization problem in this paper involves a non-convex objective function with both binary and continuous constraints, making it a mixed integer non-convex problem. To address this, the original problem is decomposed into four subproblems: computation and communication resource optimization, device association optimization, offloading decision optimization, and transmission bandwidth optimization. Within the Alternating Optimization (AO) framework, the Internal Convex Approximation (ICA) method is applied to convert the non-convex problem into a convex one. Additionally, the many-to-one matching problem is transformed into a one-to-one matching problem, and the Hungarian Algorithm (HA) is employed to solve the device association subproblem. Finally, the ICA-HA-AO is proposed to address the E2E delay optimization problem effectively.

Results and Discussions The ICA-HA-AO algorithm approximates non-convex constraints as convex ones through constraint transformation and iteratively solves the original problem, determining optimal strategies for task offloading, device association, and resource allocation. Simulation results show that the ICA-HA-AO algorithm achieves optimal performance across varying task resource requirements, bandwidth, edge processing rates, and task volumes. Compared to the worst-performing benchmark scheme, delays are reduced by approximately 0.8 s, 1.5 s, 0.5 s, and 1.2 s, respectively (Fig. 5–Fig. 8). As the DT deviation increases, the delay also increases more significantly, with a rise of about 0.13 s when the deviation increases from 0.01 to 0.02, emphasizing the importance of setting the DT deviation (Fig. 9). When the number of devices remains constant and the number of Access Points (APs) increases, the delay continues to decrease, highlighting the significance of AP deployment in practice. Additionally, when the number of APs remains fixed and the number of devices increases, the delay increases accordingly. However, the ICA-HA-AO algorithm effectively controls the rate of delay increase. For instance, when the number of devices is 10, 15, and 20, the delay increase is reduced from 0.39 s to 0.21 s (Fig. 10). These results demonstrate that the ICA-HA-AO algorithm can more efficiently utilize and schedule resources, achieving optimal resource allocation.

Conclusions This paper investigates the joint optimization problem of task offloading, device association, and resource allocation in DITEN. Firstly, within the edge computing network, physical and DT models are established for a network comprising sensors, edge servers, and actuators. A comprehensive task model is developed, and the E2E delay for tasks is derived. The optimization problem for minimizing E2E delay is then formulated, subject to constraints such as power and energy consumption. Secondly, to solve the proposed mixed integer non-convex optimization problem, the original problem is decomposed into four subproblems. Based on the ICA and HA methods, an ICA-HA-AO algorithm is proposed to solve the problem iteratively. Finally, simulation results demonstrate that the proposed ICA-HA-AO algorithm significantly reduces E2E delay and outperforms benchmark schemes. Future work may explore integrating this method with techniques to improve spectrum utilization, thereby further enhancing spectrum efficiency and overall performance in DITEN systems.

Key words: Mobile Edge Computing (MEC); Digital Twin (DT); Task offloading; Resource allocation; Alternating optimization