



研究与开发

工业互联网中基于任务紧急程度的资源分配算法

邹虹^{1,2,3}, 卓赛^{1,2,3}, 张鸿^{1,2,3}, 张明兴^{1,2,3}, 吴大鹏^{1,2,3}

- (1. 重庆邮电大学通信与信息工程学院, 重庆 400065;
2. 先进网络与智能互联技术重庆市高校重点实验室, 重庆 400065;
3. 泛在感知与互联重庆市重点实验室, 重庆 400065)

摘要: 工业互联网中任务的生成通常具有连续性和周期性, 并且任务对时延要求很高, 这给系统成本带来了挑战。为应对这一挑战, 提出了一种基于任务紧急程度的成本最小化资源分配算法。通过遗传算法优化任务的卸载策略和系统的资源分配策略, 对于卸载的任务, 根据任务的紧急程度进行调度, 并在满足时延要求的前提下计算任务的最优发射功率。仿真结果表明, 所提算法有效改善了系统总能耗成本。

关键词: 工业互联网; 资源分配; 遗传算法

中图分类号: TN929

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2024018

Task urgency-based resource allocation algorithm in industrial Internet of things

ZOU Hong^{1,2,3}, ZHUO Sai^{1,2,3}, ZHANG Hong^{1,2,3}, ZHANG Mingxing^{1,2,3}, WU Dapeng^{1,2,3}

1. School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2. Advanced Network and Intelligent Connection Technology Key Laboratory of Chongqing Education Commission of China, Chongqing 400065, China
3. Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing 400065, China

Abstract: In the industrial Internet of things, the generation of tasks is observed to exhibit both continuity and periodicity, along with stringent latency requirements. These characteristics posed challenges to system's cost-efficiency. To address these challenges, a cost minimization resource allocation algorithm based on the urgency of tasks was proposed. By employing a genetic algorithm, the task offloading strategy and the system's resource allocation strategy

收稿日期: 2023-09-01; **修回日期:** 2023-11-20

基金项目: 国家自然科学基金资助项目 (No.62271096, No.U20A20157); 重庆市教委科学技术研究项目 (No.KJQN202000626); 重庆市高校创新研究群体项目 (No.CXQT20017); 重邮信通青创团队支持计划项目 (No.SCIE-QN-2022-04); 重庆市博士后研究项目特别资助项目 (No.2021XM3058); 重庆市自然科学基金创新发展联合基金资助项目 (No.CSTB2023NSCQ-LZX0134)

Foundation Items: The National Natural Science Foundation of China (No.62271096, No.U20A20157), Science and Technology Research Program of Chongqing Municipal Education Commission (No.KJQN202000626), University Innovation Research Group of Chongqing (No.CXQT20017), Youth Innovation Group Support Program of ICE Discipline of CQUPT (No.SCIE-QN-2022-04), Chongqing Postdoctoral Science Special Foundation (No.2021XM3058), Chongqing Natural Science Foundation of China (No.CSTB2023NSCQ-LZX0134)



were optimized. For offloaded tasks, they were scheduled according to their level of urgency. Additionally, the optimal transmission power for each task was calculated to meet latency constraints. Simulation results indicate that the proposed algorithm effectively reduces the overall energy cost of the system.

Key words: IIoT, resource allocation, genetic algorithm

0 引言

工业物联网(industrial Internet of things, IIoT)通过互联各类工业资源使得资源分配与任务调度得以高度优化,从而显著提升了生产效率和服务质量^[1]。各种IIoT应用场景已经普遍采用机器学习(machine learning, ML)解决方案,如缺陷诊断、故障检测、产品分类等^[2]。以产品分类为例,感应设备捕捉产品的图像并进行预处理,然后使用深度神经网络(deep neural network, DNN)对预处理后的数据进行推理最终得到控制指令,执行器执行该指令^[3]。然而,IIoT任务对时延要求高,为应对这一挑战,可将任务卸载到边缘网络进行处理。

5G边缘计算与工业融合之后,逐步成为支撑工业生产的基础设施^[4]。在边缘网络中,边缘服务器(edge server, ES)的计算和存储能力优于终端设备,可以部署拥有更多神经元和隐藏层的全尺寸DNN(full-size deep neural network, FDNN)模型。终端设备上受硬件条件的限制无法部署FDNN模型,但可以通过权值修剪技术修剪不太重要的权值以降低计算复杂度和模型大小^[5],将修剪后的DNN模型称为轻量级DNN(light-weight deep neural network, LDNN)模型。这两个模型在处理相同的任务时,LDNN消耗的计算资源比FDNN少^[6],但LDNN的推理精度降低了。任务卸载到ES时会带来额外的传输时延以及传输能耗,如何制定合理的卸载决策是非常重要的。

在IIoT场景中,设备需要持续进行数据处理和通信,这增加了能源消耗的压力^[7]。这些特点不仅要求设备具有高效的处理能力,还必须长时

间稳定运行,因此,高能耗成为一个不可忽视的问题。这不仅增加了运营成本,还可能导致能源浪费和环境负担,因此,在IIoT中,对能耗的优化也是至关重要的。

目前,已存在一些工作研究了IIoT场景中的ML任务的资源分配问题。文献[8]以最小化时延成本和推理错误成本为目标,提出了一种精度感知的联合任务卸载和资源分配方案,任务的卸载决策制定过程考虑了当前任务的精度要求以及系统的队列长度情况。文献[9]提出了一种新的计算卸载框架,用于IIoT的分层ML任务分配,并根据设备和ES上的ML模型的复杂性、计算能力以及当前到来任务数据的质量,提出了一个分段凸优化问题以最小化任务执行总时延。文献[10]考虑了数据大小对推理精度的影响,提出了一种基于端边合作的IIoT联合任务推理的DNN部署、任务卸载、数据大小控制、计算资源分配和无线信道分配方案。文献[11]以最大化任务的推理精度为目标,考虑了任务质量对DNN模型的推理精度影响,通过高效的资源利用为推理任务选择合适的DNN模型,从而在满足严格时延要求的同时有效提高任务推理的平均精度。文献[12]提出了一种高效迭代优化方案联合优化用户的卸载决策、发射功率、计算资源和子信道资源分配来最小化能耗。

从现有研究工作中看,部分研究者已经对ML任务执行时延和推理精度问题进行了研究,然而某些IIoT场景中的任务只需要在规定的最大可容忍时延内得到处理即可。此外,DNN推理错误所带来的资源消耗也是部分研究者所忽视的问题。因此,本文提出了一种基于任务紧急程度的成本最小化资源分配(task urgency based cost mini-

mization resource allocation, TUBCMRA) 算法, 考虑了不同 DNN 模型推理不同质量任务时的精度, 在保证任务执行时延的前提下, 最小化任务执行的能耗成本以及任务推理错误带来的成本。本文的主要贡献如下。

(1) 构建了一个单 ES 的计算卸载框架。根据任务的质量和紧急程度、当前系统的队列积压情况等因素, 联合优化卸载决策、发射功率分配、资源分配和任务调度策略。

(2) 由于所提的问题是一个混合整数非线性规划问题, 无法直接求解, 因此本文结合遗传算法 (genetic algorithm, GA) 将每轮迭代的求解过程分为 3 个阶段进行求解。

1 系统模型与问题描述

工业部件分类模型如图 1 所示。该模型主要分为设备层和边缘层。设备层由 N 组 IIoT 设备组成, 设备集合表示为 $N=\{1, 2, \dots, N\}$, 每组设备由传送带、感应器以及执行器构成, 感应器上部署有离线训练完成的 LDNN 模型。边缘层由基站

(base station, BS) 和边缘服务器 (edge server, ES) 组成, ES 上部署了多种 FDNN 模型。每组设备通过无线链路连接到 BS, BS 将收集到的数据发送给 ES 进行处理。

对于每组设备而言, 传送带上运载着多个工业部件, 感应器收集每个工业部件的数据信息并生成任务, 该任务可以由感应器上低精度的 LDNN 推理, 也可以卸载到 ES 由高精度的 FDNN 推理。系统中可能存在多组设备生成同种类型的任务, 任务类型表示为 $J=\{1, 2, \dots, J\}$, 任务类型相同的设备集合用 N_j 表示。任务推理完成后会生成相应的控制指令, 将发送到设备上的执行器执行。由于控制指令的大小远小于任务大小, 因此控制指令的传输过程可忽略不计^[9]。该场景下的任务调度和资源分配可描述为: 系统根据任务的质量以及网络条件决定的卸载策略, 为确保任务执行满足时延约束, 系统会分配适当的计算资源。对于卸载到 ES 的任务, 系统还会根据任务的紧急程度进行调度。

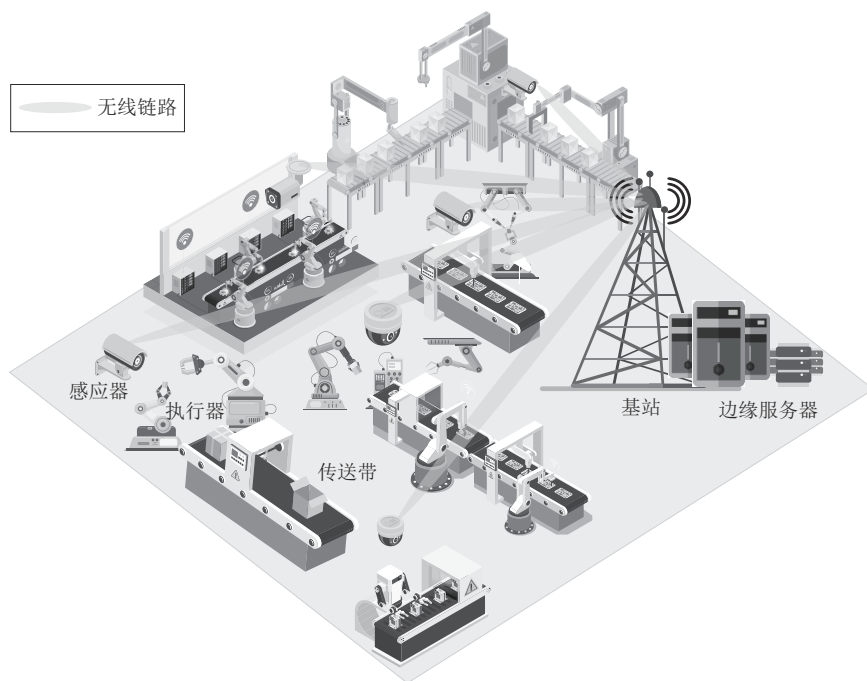


图1 工业部件分类模型



1.1 任务模型

本文考虑了 T 个时隙，每个时隙的长度为 τ ，时隙集合可以表示为 $T=\{0,1,2,\dots,T-1\}$ 。IIoT 设备 i 在时隙 t 中生成的第 j 类任务表示为 $\{D_{ij}(t), T_{ij}(t), \xi_{ij}(t)\}$ ， $D_{ij}(t)$ 表示设备 i 在时隙 t 生成的任务大小（单位：byte）； $T_{ij}(t)$ 表示该任务的最大可容忍时延； $\xi_{ij}(t)$ 表示任务质量， $\xi_{ij}(t)=1$ 表示任务 i 为有噪任务， $\xi_{ij}(t)=0$ 表示任务 i 为无噪任务。不同质量的任务与 DNN 模型的映射关系见表 1。

表 1 不同质量的任务与 DNN 模型的映射关系

对比项	LDNN	FDNN
无噪数据	p_{UL}	p_{UD}
有噪数据	p_{NL}	p_{ND}

1.2 计算模型

1.2.1 本地计算

当任务由本地的 LDNN 推理时，卸载决策变量 $o_i(t)=0$ ，任务执行时延包括任务推理时延和排队时延，推理时延可以表示为：

$$\tau_i^l(t) = \frac{\mu_j^l D_{ij}(t)}{f_i^l(t)} + \tau_{ij}^{ql}(t) \quad (1)$$

其中， $\mu_j^l D_{ij}(t)$ 表示第 j 类任务在 LDNN 模型上推理的计算量， $f_i^l(t)$ 表示设备 i 在时隙 t 分配的计算资源， $\tau_{ij}^{ql}(t)$ 表示任务排队时延。

本地执行的任务以任务队列的形式进入缓冲区，设备 i 的任务队列 $U_i(t)$ 在 $t+1$ 时隙更新为：

$$U_i(t+1) = \left[U_i(t) - \frac{f_i^l(t)\tau}{\mu_j^l} \right]^+ + (1 - o_i(t))D_{ij}(t) \quad (2)$$

其中， $[x]^+ = \max(x, 0)$ ， τ 是时隙长度， $f_i^l(t)\tau/\mu_j^l$ 表示当前时隙设备 i 能够处理的任务， $(1 - o_i(t))D_{ij}(t)$ 表示卸载到 ES 的任务。任务的排队时延可以计算为：

$$\tau_{ij}^{ql}(t) = \frac{\mu_j^l U_i(t)}{f_i^l(t)} \quad (3)$$

本地计算的能耗取决于消耗的 CPU 周期

数^[13]，可表示为：

$$e_i^l(t) = \kappa \mu_j^l D_{ij}(t) f_i^l(t)^2 \quad (4)$$

其中， κ 表示设备上的芯片架构的有效电容系数。

1.2.2 边缘计算

边缘计算分为数据传输和任务推理两个阶段。对于数据传输阶段而言，假设 IIoT 设备与 BS 的连接基于正交频分多址，则信道间的干扰被消除。基站的总带宽为 B ，平均分配给每个设备，设备的发射功率表示为 $p_i(t)$ ，则传输速率表示为：

$$R_i(t) = \frac{B}{N} \log \left(1 + \frac{p_i(t)h}{\sigma^2} \right) \quad (5)$$

其中， h 表示信道增益， σ^2 表示噪声功率。基于式 (5)，任务传输时延表达式为：

$$\tau_{ij}^{\text{trans}}(t) = \frac{D_{ij}(t)}{R_i(t)} \quad (6)$$

传输能耗可根据传输时延计算得出：

$$e_i^{\text{trans}}(t) = p_i(t) \tau_{ij}^{\text{trans}}(t) \quad (7)$$

同种类型的任务要在相同 DNN 模型上推理，ES 上同一时隙下不同 FDNN 模型的任务排队情况如图 2 所示，不同类型的任务以队列的形式进入对应 FDNN 模型的缓冲区，各 FDNN 模型串行推理到来的任务。ES 推理任务的过程中，给每种 FDNN 模型分配适当的计算资源。ES 总的计算资源用 $f_{\max}^e$ 表示，时隙 t 中分配给 FDNN $_j$ 的计算资源表示为 $f_j^e(t)$ ，则任务 i 的推理时延为：

$$\tau_{ij}^{\text{com}}(t) = \frac{\mu_j^e \sum_{m=1}^{s_i} D_{ij}^m(t)}{f_j^e(t)} \quad (8)$$

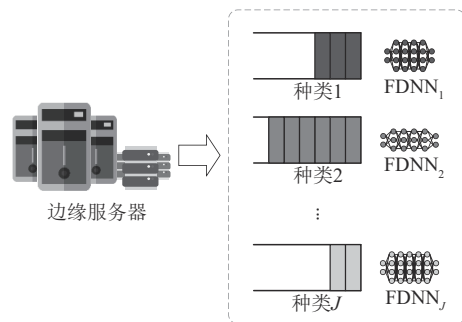


图2 ES上同一时隙下不同DNN模型的任务排队情况

将ES上第 j 类任务的队列长度记为 $G_j(t)$, 则 $t+1$ 时隙中的队列长度应更新为:

$$G_j(t+1) = \left[G_j(t) - \frac{f_j^e(t)\tau}{\mu_j^e} \right]^+ + \sum_{i \in N_j} (1 - o_i(t)) D_{ij}(t) \quad (9)$$

其中, $f_j^e(t)\tau/\mu_j^e$ 表示FDNN $_j$ 在时隙 t 中能够处理的任务大小, $\sum_{i \in N_j} (1 - o_i(t)) D_{ij}(t)$ 表示时隙 t 中卸载到ES的第 j 类任务大小之和。任务 i 在FDNN $_j$ 上的排队时延为:

$$\tau_{ij}^{qe}(t) = \frac{\mu_j^e G_j(t)}{f_j^e(t)} \quad (10)$$

因此, 边缘计算总时延为:

$$\tau_{ij}^e(t) = \tau_{ij}^{\text{trans}}(t) + \tau_{ij}^{qe}(t) + \tau_{ij}^{\text{com}}(t) \quad (11)$$

1.3 推理模型

任务的推理精度与推理该任务的DNN模型以及任务的质量有关, 结合表1给出的不同质量的任务与DNN模型的映射关系可将任务推理精度表示为:

$$p_{ij}(t) = o_i(t)(\xi_{ij}(t)p_{\text{NL}} + (1 - \xi_{ij}(t))p_{\text{UL}}) + (1 - o_i(t))(\xi_{ij}(t)p_{\text{ND}} + (1 - \xi_{ij}(t))p_{\text{UD}}) \quad (12)$$

若DNN模型推理得到错误的控制指令, 则会导致经济损失^[8], 将任务 i 的经济损失定义为 $\omega e_{ij}(t)$, 其中 ω 为惩罚系数, 文献[9]指出, ML任务推理错误带来的惩罚大于当前任务执行成本, 因此, $\omega \geq 1$ 。任务 i 由推理得到错误的控制指令而造成的经济损失为:

$$\pi_{ij}^{\text{err}}(t) = (1 - p_{ij}(t))\omega e_{ij}(t) \quad (13)$$

其中, $(1 - p_{ij}(t))$ 为任务 i 推理错误的概率, $e_{ij}(t)$ 为任务 i 执行总能耗:

$$e_{ij}(t) = (1 - o_i(t))e_i^l(t) + o_i(t)e_i^{\text{trans}}(t) \quad (14)$$

每个任务的成本包括能耗成本和错误推理的惩罚, 则时隙 t 中所有任务的成本之和为:

$$E(t) = \sum_{i=1}^N e_{ij}(t) + \pi_{ij}^{\text{err}}(t) \quad (15)$$

1.4 问题描述与问题转化

本文目标是在满足每个任务的时延约束的前提下, 最小化系统的时间平均总成本。优化问题如下。

$$\begin{aligned} P1: & \text{minimize} \quad \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E[E(t)] \\ & \{O, F^l, F^e, P, S\} \\ \text{s.t.} & \text{C1: } o_i(t) \in \{0, 1\}, \forall i \in N, \forall t \in T \\ & \text{C2: } 0 \leq f_i^l(t) \leq f_i^{\max} \\ & \text{C3: } 0 \leq f_j^e(t) \leq f_e^{\max} \\ & \text{C4: } \sum_{j=1}^J f_j^e(t) \leq f_e^{\max} \\ & \text{C5: } 0 \leq p_i(t) \leq p_i^{\max} \\ & \text{C6: } \tau_{ij}(t) \leq T_{ij}(t), \forall i \in N_j, j \in J \\ & \text{C7: } \lim_{t \rightarrow \infty} \frac{E\{U_i(t)\}}{t} = 0, \lim_{t \rightarrow \infty} \frac{E\{G_j(t)\}}{t} = 0 \end{aligned} \quad (16)$$

其中, C1表示卸载决策变量约束; C2表示设备的计算资源分配约束; C3与C4表示ES的计算资源分配约束; C5表示设备的传输功率分配的约束; C6表示任务的最大可容忍时延约束, $\tau_{ij}(t)$ 为任务 i 的总执行时延, 可表示为:

$$\tau_{ij}(t) = o_i(t)\tau_{ij}^l(t) + (1 - o_i(t))\tau_{ij}^e(t) \quad (17)$$

C7表示设备以及ES上的队列稳定性约束。

P1是一个随机优化问题, 直接对其进行求解需要提前获取所有时隙中的参数, 并且约束C7与 t 耦合, 因此很难直接进行求解。为解决这些问题, 可根据李雅普诺夫优化理论^[14]对问题进行转化。

定义系统中的队列集合 $A(t) = \{U_i(t), G_j(t) | \forall i \in N, j \in J\}$, 因此, 李雅普诺夫二次函数可表示为:

$$L(A(t)) = \frac{1}{2} \sum_{i=1}^N U_i^2(t) + \frac{1}{2} \sum_{j=1}^J G_j^2(t) \quad (18)$$

其中, $U_i(0) = 0, (\forall i \in N), G_j(0) = 0, (\forall j \in J)$ 。引入李雅普诺夫漂移函数表示相邻时隙的队列变化情况, 李雅普诺夫漂移函数可表示为:

$$\Delta(A(t)) = E[L(A(t+1)) - L(A(t)) | A(t)] \quad (19)$$

为了在保证系统稳定性的前提下最小化系统的能耗成本, 将时隙 t 中的漂移加惩罚函数定义为:



$$\Delta(A(t)) + VE[E(t)|A(t)] \quad (20)$$

其中, $V > 0$ 是衡量能耗成本与李雅普诺夫漂移的权重, V 的值越大表示越要牺牲队列的稳定性来最小化系统的能耗成本, 反之亦然。问题 P1 可转化为每个时隙中的优化问题 P2:

$$\begin{aligned} \text{P2: minimize } & \Delta(A(t)) + VE[E(t)|A(t)] \\ & \{O, F^l, F^e, P, S\} \\ \text{s.t.} & \text{C1} \sim \text{C6} \end{aligned} \quad (21)$$

定理 1 给定时间隙 t 中的卸载策略以及资源分配策略可以得到 $\Delta(A(t))$ 的上界为^[8]:

$$\begin{aligned} \Delta(A(t)) \leq & B + E \left[\sum_{i=1}^N U_i(t) \left((1 - o_i(t)) D_{ij}(t) - \frac{f_i^l(t)\tau}{\mu_j^l} \right) \right] \\ & + E \left[\sum_{j=1}^J G_j(t) \left(\sum_{i \in N_j} o_i(t) D_{ij}(t) - \frac{f_j^e(t)\tau}{\mu_j^e} \right) | A(t) \right] \end{aligned} \quad (22)$$

$$\begin{aligned} \text{其中, } B = & 1/2 \sum_{i=1}^N \left(\left(f_{\max}^l \tau / \mu_j^l \right)^2 + \left(D_{ij}^{\max} \right)^2 \right) + \\ & 1/2 \sum_{j=1}^J \left(\left(f_{\max}^e \tau / \mu_j^e \right)^2 + \left(|N_j| D_{ij}^{\max} \right)^2 \right). \end{aligned}$$

定理 1 表明问题 P2 可转化为漂移函数的上界加惩罚函数的最小化。忽略式 (22) 中的常数项 B , 将问题 P2 转化为 P2.1:

$$\begin{aligned} \text{P2.1: minimize } & F(O, F^l, F^e, P, S) = \\ & \{O, F^l, F^e, P, S\} \\ & \sum_{i=1}^N U_i(t) \left((1 - o_i(t)) D_{ij}(t) - \frac{f_i^l(t)\tau}{\mu_j^l} \right) + \\ & \sum_{j=1}^J G_j(t) \left(\sum_{i \in N_j} o_i(t) D_{ij}(t) - \frac{f_j^e(t)\tau}{\mu_j^e} \right) + VE(t) \\ \text{s.t.} & \text{C1} \sim \text{C7} \end{aligned} \quad (23)$$

2 算法设计

本节提出了一种三阶段的优化算法。第一阶段, 使用 GA 确定卸载策略以及资源分配策略。第二阶段, 根据任务的紧急程度对任务进行调度。第三阶段, 根据前两阶段求得的卸载策略、资源分配策略以及任务调度策略计算每个设备的

最优发射功率分配策略。通过多轮迭代, 得到优化问题的稳定收敛解。

2.1 任务调度策略

综合考虑 ES 上每种 FDNN 模型在同一时隙中到达任务的特性, 对任务进行调度。卸载任务在推理之前需经历传输和排队两个阶段, 因此, 可求得任务 i 的剩余最大可容忍时延为:

$$\tau_{ij}^{\text{rem}}(t) = T_i - \tau_{ij}^{\text{qe}}(t) - \frac{D_{ij}(t)}{R_i(t)} \quad (24)$$

当任务 i 的计算时延等于 $\tau_{ij}^{\text{rem}}(t)$ 时, 所需要的计算资源为:

$$\text{de}_i(t) = \frac{\mu_j^e D_{ij}(t)}{\tau_{ij}^{\text{rem}}(t)} \quad (25)$$

在实际中, 同一时隙内每种 FDNN 模型上到来的任务获得的计算资源相同, 所以 $\text{de}_i(t)$ 可以作为衡量任务紧急程度的指标。 $\text{de}_i(t)$ 的值越大表示任务 i 的紧急程度越高, 则任务 i 应该优先执行。

2.2 发射功率分配

当给定卸载策略、资源分配策略以及任务调度策略后, 可将 P2.1 转化为:

$$\begin{aligned} \text{minimize } & \sum_{i=1}^N F(O^*, F^{l*}, F^{e*}, P, S^*) = \\ \text{minimize } & \sum_{i=1}^N \frac{o_i(t) p_i(t) D_{ij}(t)}{\frac{B}{N} \ln \left(1 + \frac{p_i(t) h}{\sigma^2} \right)} \\ \text{s.t.} & \text{C5, C6} \end{aligned} \quad (26)$$

由于每个设备的发射功率分配是独立的, 上述问题可以分解成 N 个子问题单独进行求解。

定理 2 当 $p_i(t) \in [0, p_i^{\max}]$ 时, $F_i(O^*, F^{l*}, F^{e*}, P, S^*)$ 在 $[0, p_i^{\max}]$ 内是单增函数

证明 因为 $F_i(O^*, F^{l*}, F^{e*}, P, S^*)$ 的导数 F_i' 的值在 $[0, p_i^{\max}]$ 内满足 $F_i' > 0$, 所以 $F_i(O^*, F^{l*}, F^{e*}, P, S^*)$ 在 $[0, p_i^{\max}]$ 内是关于 $p_i(t)$ 的增函数。

由定理 2 可知, 在满足约束 C6 的前提下,

$p_i(t)$ 的下界为最优解, 当 $p_i^{\max}$ 不满足约束C6时, $p_i^{\max}$ 为最优解, 根据式(26)求得 $p_i(t)$ 的下界为:

$$p_i(t) = \frac{\sigma^2}{h} \left(\frac{ND_{ij}(t)}{2^{B(T_i - \tau_{ij}^{qc}(t))} - 1} \right) \quad (27)$$

2.3 任务卸载和资源分配策略

2.3.1 算法初始化

在GA中, 将待优化的决策变量作为染色体。第一次迭代时, 传入的参数是初始化变量, 后续迭代中传入的是上次迭代结束后生成的子代种群参数。基于卸载决策变量和资源分配决策变量的特点, 采用浮点数编码的方法, 每个染色体可表示为:

$$\text{Ch}_l = [\mathbf{O}, \mathbf{F}^l, \mathbf{F}^e]^T \quad (28)$$

适应度值是衡量染色体的优良程度的标准, P2.1的目标是求 $F(\mathbf{O}, \mathbf{F}^l, \mathbf{F}^e, \mathbf{P}, \mathbf{S})$ 的最小值, 所以将适应度函数定义为:

$$y_k = \begin{cases} -F(\mathbf{O}, \mathbf{F}^l, \mathbf{F}^e, \mathbf{P}, \mathbf{S}) & , \text{染色体}k \text{满足约束} \\ -(F_b + \varepsilon_k) & , \text{染色体}k \text{不满足约束} \end{cases} \quad (29)$$

其中, F_b 为当前种群中满足时延约束C7且适应度值最低的染色体。 ε_k 是用来衡量当前染色体对应的总时延与最大可容忍时延之间的差距的惩罚因子, 表示为:

$$\varepsilon_k = \lambda \sum_{i=1}^N [\tau_{ij}(t) - T_i]^+ \quad (30)$$

其中, λ 表示惩罚权重。

2.3.2 遗传算子

交叉能够实现不同染色体之间优良基因的重组, 从而加速算法寻找全局最优解的速度, 多点交叉能够创建更多种类的基因组合, 提供更大的搜索空间, 因此, 本节采用多点交叉。变异是GA中的一个关键操作, 可以为种群引入新的信息, 避免陷入局部最优。变异逻辑如下。

$$f_i^l = \begin{cases} f_i^l + \alpha(f_{\max} - f_i^l) & , \beta \geq 0.5 \\ f_i^l - \alpha(f_i^l - f_{\min}) & , \beta < 0.5 \end{cases} \quad (31)$$

其中, α 和 β 为随机数, α 表示变异的步长, β 决定变异的方向。

算法1 TUBCMRA算法

输入: $D_{ij}(t), T_i(t), \zeta_{ij}(t), t$

初始化: 种群数量 L , 最大迭代次数 $g_{\max}$,

交叉概率 p_c , 变异概率 p_m

while 迭代次数 $< g_{\max}$ do

根据式(29)求解每个染色体的适应度值 y_k

根据每个染色体的 y_k , 使用锦标赛选择法选择 L 个染色体

随机生成交叉的染色体对

if rand() $> p_c$

随机生成两个交叉点交换两点之间的基因

end if

if rand() $> p_m$

根据式(31)执行变异逻辑

end if

根据式(25)计算 $de_i(t)$, 根据任务紧急程度制定任务调度策略

根据式(27)计算每个任务的发射功率

更新 $F(\mathbf{O}, \mathbf{F}^l, \mathbf{F}^e, \mathbf{P}, \mathbf{S})$ 值最小的策略集合 $\{\mathbf{O}, \mathbf{F}^l, \mathbf{F}^e, \mathbf{P}, \mathbf{S}\}$

$g = g + 1$

end while

输出: 种群中最优个体对应的任务执行策略

3 仿真分析

假设共有25组IIoT设备, 每组设备中部署有离线训练完成的LDNN模型, 总共有6种任务, 设



备随机分布在占地面积为 $300\text{ m}\times 300\text{ m}$ 的工厂中,附近有一个BS覆盖所有的设备,信道总带宽 $B=15\text{ MHz}$,噪声功率谱密度 $N_0=-174\text{ dBm/Hz}$ ^[6],ES的总计算资源为 $f_{\max}^e=16\text{ GHz}$ ^[8],种群个数 $L=20$ 。其他仿真参数见表2。将本文提出的TUBCMRA算法与以下方案进行了对比。

表2 仿真参数

仿真参数 ^[16-17]	数值	仿真参数	数值
p_{UL}	[0.915, 0.938]	$D_{ij}(t)$	[0.2, 0.5] MB
p_{NL}	[0.876, 0.897]	p_c	0.6
p_{ND}	[0.929, 0.954]	p_m	0.02
p_{UD}	[0.915, 0.938]	h	4×10^{-13}

(1) 不考虑任务质量 (quality ignore, QI): 任务卸载过程中不考虑任务质量的影响。

(2) 随机执行 (execute randomly, ER): 对于同一时隙卸载的任务按随机序列执行。

(3) 有噪任务卸载, 无噪任务本地处理 (noisy offload unnoisy locally execute, NOULE)。

边缘队列和设备队列随时隙的变化情况如图3所示。从图3中可以看出,在初始状态,设备以及边缘上的队列长度快速增加,然后趋于稳定。可以看出,在对原问题进行李雅普诺夫优化后,采用TUBCMRA算法求解能够保证设备层和边缘层任务队列的稳定性。

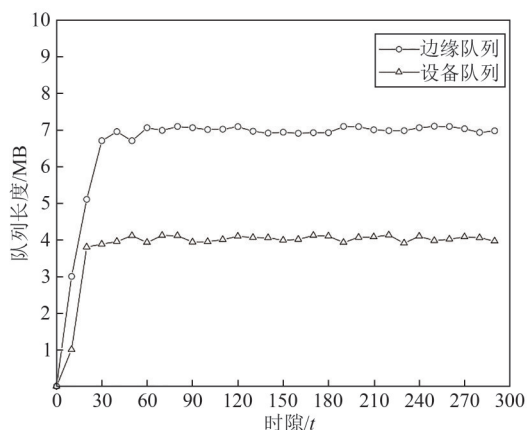


图3 边缘队列和设备队列随时隙的变化情况

总能耗成本和任务的平均数据量大小的关系如图4所示。图4中4个方案的总能耗成本都随着平均任务的增大而增加。随着平均任务变大,任务本地处理的计算量以及任务卸载时的传输时延都会变大,从而增加了任务执行的能耗成本。所提TUBCMRA算法的总能耗成本是最小的,并且当平均任务越大时该算法的优势越明显。当平均任务为0.5 MB时,TUBCMRA算法的总能耗成本分别要比QI、ER、NOULE方案分别低16%、22%、42%。这是因为TUBCMRA算法根据任务的紧急程度进行了调度,减少了紧急任务的等待时间,任务对传输时延的要求变得相对宽松,分配较低的发射功率也能满足时延约束,从而减少了任务传输能耗。

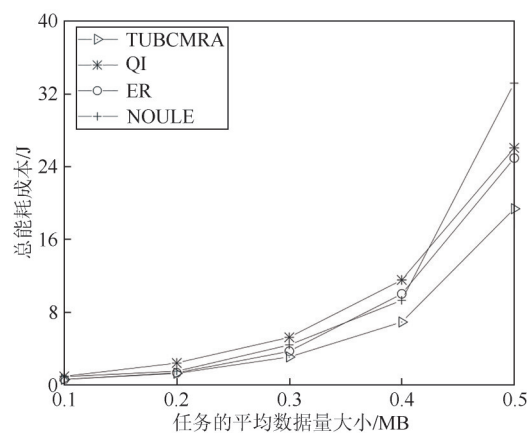


图4 总能耗成本和任务的平均数据量大小的关系

惩罚系数 ω 和总能耗成本的关系如图5所示,所有方案的总成本都随着 ω 的增加而增加,因为在总能耗成本的表达式中 ω 的增加将直接导致总成本的增加。方案QI没有考虑任务质量的影响,总能耗成本是最高的。所提TUBCMRA算法结合了任务的质量对任务的调度策略以及资源分配策略进行了联合优化, ω 越大表示会有更多任务被卸载到ES以降低推理错误带来的影响,并且还会根据卸载任务的紧急程度进行调度。当 ω 为6时,TUBCMRA算法对应的总能耗

成本相较于ER、NOULE、QI方案分别降低了19%、30%和37%。

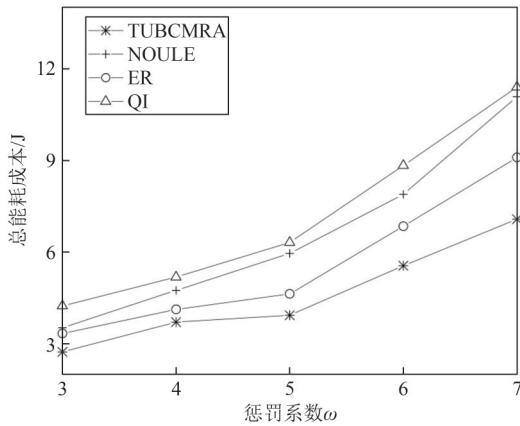


图5 惩罚系数 ω 和总能耗成本的关系

ES的计算资源和总能耗成本的关系如图6所示。当ES上的计算资源增加时,可以减少卸载任务的执行时延,从而满足更多任务的时延要求,更多任务可以卸载到ES处理以减少任务推理错误的加权成本。QI和NOULE方案的总能耗成本改善不明显,因为QI方案没有考虑任务质量的影响,不能增加卸载到ES的任务数量,NOULE方案固定了卸载策略,增加ES的总计算资源只能改善卸载时的传输能耗。TUBCMRA算法与ER方案在ES的计算资源增加时会将更多的任务卸载到ES处理以减小任务推理错误的影响。并且,TUBCMRA算法考虑了卸载任务的紧急程度,所以能耗成本最小。

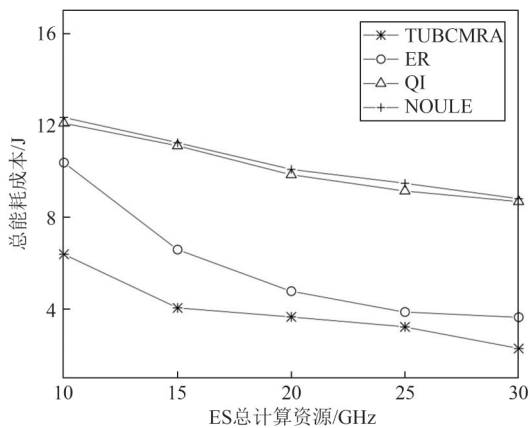


图6 ES的计算资源和总能耗成本的关系

4 结束语

本文针对IIoT中的工业部件分类场景提出了一种资源分配算法,以优化系统能耗成本为目标制定任务调度与资源分配策略。由于所提问题与时间耦合,采用李雅普诺夫优化技术将问题与时间解耦。在每个时隙中,使用GA求解卸载策略以及资源分配策略。对于卸载任务,根据任务的紧急程度进行调度并在满足时延要求的前提下计算最优发射功率。仿真结果表明,与其他方案相比,所提TUBCMRA算法能够有效改善工业部件分类场景中的系统能耗成本。在未来,可将当前工作拓展到多ES场景,考虑每个ES的DNN模型部署策略以及各ES之间的资源共享,以改善系统运行成本。

参考文献:

- [1] QIU T, CHI J C, ZHOU X B, et al. Edge computing in industrial Internet of things: architecture, advances and challenges[J]. IEEE Communications Surveys & Tutorials, 2020, 22(4): 2462-2488.
- [2] SISINNI E, SAIFULLAH A, HAN S, et al. Industrial Internet of things: challenges, opportunities, and directions[J]. IEEE Transactions on Industrial Informatics, 2018, 14(11): 4724-4734.
- [3] LIANG F, YU W, LIU X, et al. Toward edge-based deep learning in industrial Internet of things[J]. IEEE Internet of Things Journal, 2020, 7(5): 4329-4341.
- [4] 王峰, 于青民, 黄颖, 等. 工业互联网网络关键技术与发展研究[J]. 电信科学, 2022, 38(7): 106-113.
WANG F, YU Q M, HUANG Y, et al. Research on key technology and development of industrial Internet network[J]. Telecommunications Science, 2022, 38(7): 106-113.
- [5] HAN S, MAO H Z, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[EB]. 2015: arXiv: 1510.00149.
- [6] WU W, YANG P, ZHANG W T, et al. Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning[J]. IEEE Transactions on Industrial Informatics, 2021, 17(7): 4988-4998.
- [7] 赵尚维康. 工业物联网中基于边缘计算和强化学习的计算卸载方法研究[D]. 南京: 南京邮电大学, 2022.
ZHAO S W K. Research on computing unloading method



based on edge computing and reinforcement learning in industrial Internet of things[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2022.

- [8] FAN W H, LI S M, LIU J, et al. Joint task offloading and resource allocation for accuracy-aware machine-learning-based IIoT applications[J]. IEEE Internet of Things Journal, 2023, 10(4): 3305-3321.
- [9] YANG B, CAO X L, LI X F, et al. Mobile-edge-computing-based hierarchical machine learning tasks distribution for IIoT[J]. IEEE Internet of Things Journal, 2020, 7(3): 2169-2180.
- [10] FAN W H, CHEN Z Y, HAO Z B, et al. DNN deployment, task offloading, and resource allocation for joint task inference in IIoT[J]. IEEE Transactions on Industrial Informatics, 2023, 19(2): 1634-1646.
- [11] ZHANG W T, YANG D, PENG H X, et al. Deep reinforcement learning based resource management for DNN inference in industrial IoT[J]. IEEE Transactions on Vehicular Technology, 2021, 70(8): 7605-7618.
- [12] 绳韵, 许晨, 郑光远. 基于NOMA的超密集MEC网络任务卸载和资源分配方案[J]. 电信科学, 2022, 38(2): 35-46.
SHENG Y, XU C, ZHENG G Y. Task offloading and resource allocation in NOMA-based ultra-dense MEC networks[J]. Telecommunications Science, 2022, 38(2): 35-46.
- [13] WANG K Z, YANG K, MAGURAWALAGE C S. Joint energy minimization and resource allocation in C-RAN with mobile cloud[J]. IEEE Transactions on Cloud Computing, 2018, 6(3): 760-770.
- [14] NEELY M J. Stochastic network optimization with application to communication and queueing systems[M]. Cham: Springer International Publishing, 2010.
- [15] ZHANG W T, YANG D, WANG H C. Data-driven methods for predictive maintenance of industrial equipment: a survey[J]. IEEE Systems Journal, 2019, 13(3): 2213-2227.
- [16] LUO J H, WU J X, LIN W Y. ThiNet: a filter level pruning method for deep neural network compression[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 5068-5076.

[作者简介]



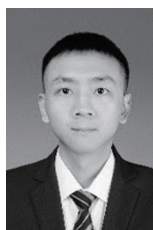
邹虹 (1970-), 女, 重庆邮电大学通信与信息工程学院副教授, 主要研究方向为工业物联网、光无线融合网络等。



卓赛 (1999-), 男, 重庆邮电大学通信与信息工程学院硕士生, 主要研究方向为工业物联网。



张鸿 (1987-), 男, 博士, 重庆邮电大学通信与信息工程学院讲师, 主要研究方向为工业物联网、光无线融合网络等。



张明兴 (1999-), 男, 重庆邮电大学通信与信息工程学院硕士生, 主要研究方向为工业物联网。



吴大鹏 (1979-), 男, 博士, 重庆邮电大学通信与信息工程学院教授, 主要研究方向为工业物联网、泛在无线网络等。