Digital Twin-Assisted Edge Computation Offloading in Industrial Internet of Things With NOMA

Long Zhang 1,1, Han Wang 2, Hongmei Xue 2, Hongliang Zhang 2, Qilie Liu 2, Dusit Niyato 2, and Zhu Han 2

1Hebei University of Engineering
2Affiliation not available

October 31, 2023

Abstract

Integrating digital twins (DTs) and multi-access edge computing (MEC) is a promising technology that realizes edge intelligence in 6G, which has been recognized as the key enabler for Industrial Internet of Things (IIoT). In this paper, we explore a DT-assisted MEC system for the IIoT scenario where a DT server is created as a digital counterpart of the MEC server, via estimating the computation state of the MEC server within the DT modelling cycle. To achieve energy and spectrally efficient offloading, we consider that IIoT devices communicate with industrial gateways (IGWs) through a non-orthogonal multiple access (NOMA) protocol. Each IIoT device has an industrial computation task that can be executed locally or fully offloaded to IGW. We aim to minimize the total task completion delay of all IIoT devices by jointly optimizing the IGW’s subchannel assignment as well as the computation capacity allocation, edge association, and transmit power control of IIoT device. The resulting problem is shown to be a mixed integer non-convex optimization problem, which is NP-hard and challenging to solve. We decompose the original problem into four solvable sub-problems, and then propose an overall alternating optimization algorithm to solve the sub-problems iteratively until convergence. Validated via simulations, the proposed scheme shows superiority to the benchmarks in reducing the total task completion delay and increasing the percentage of offloading IIoT devices.
Abstract—Integrating digital twins (DTs) and multi-access edge computing (MEC) is a promising technology that realizes edge intelligence in 6G, which has been recognized as the key enabler for Industrial Internet of Things (IIoT). In this paper, we explore a DT-assisted MEC system for the IIoT scenario where a DT server is created as a digital counterpart of the MEC server, via estimating the computation state of the MEC server within the DT modelling cycle. To achieve energy and spectrally efficient offloading, we consider that IIoT devices communicate with industrial gateways (IGWs) through a non-orthogonal multiple access (NOMA) protocol. Each IIoT device has an industrial computation task that can be executed locally or fully offloaded to IGW. We aim to minimize the total task completion delay of all IIoT devices by jointly optimizing the IGW’s subchannel assignment as well as the computation capacity allocation, edge association, and transmit power control of IIoT device. The resulting problem is shown to be a mixed integer non-convex optimization problem, which is NP-hard and challenging to solve. We decompose the original problem into four solvable subproblems, and then propose an overall alternating optimization algorithm to solve the sub-problems iteratively until convergence. Validated via simulations, the proposed scheme shows superiority to the benchmarks in reducing the total task completion delay and increasing the percentage of offloading IIoT devices.

Index Terms—Industrial Internet of Things, multi-access edge computing, digital twin, non-orthogonal multiple access.

I. INTRODUCTION

The recent advancements in Industrial Internet of Things (IIoT) and wireless communications technologies have motivated a variety of delay-sensitive and computation-intensive industrial applications, such as asset tracking, predictive maintenance, and smart factories [1], [2]. The success of these applications enables industries and enterprises to have better efficiency and reliability in their operations. However, due to the limited computation resources and processing capabilities of IIoT devices (e.g., sensors, actuators, and machines), it is a significant challenge for them to run these intensive computing applications with critical latency requirements locally. Multi-access edge computing (MEC) is a promising solution for this challenge, where the whole or a fraction of industrial computation tasks of IIoT devices can be offloaded to MEC servers deployed at the edge of IIoT networks, e.g., computation access points (CAPs), via industrial gateways (IGWs) [3]–[5]. By leveraging the superior computing capabilities of MEC servers, the tasks can be successfully processed in a timely, secure, and efficient manner [6], [7].

On the other hand, energy and spectrally efficient offloading is especially important for the success of MEC in IIoT, which motivates the application of non-orthogonal multiple access (NOMA) to enable computation offloading. Unlike traditional orthogonal multiple access (OMA), NOMA allows the same resource block to be shared by multiple users simultaneously, and further exploits the signal difference on power domain to distinguish the users via the successive interference cancellation (SIC) [8]. For NOMA-enabled task offloading in IIoT, a group of IIoT devices can offload their industrial computation tasks to an IGW more efficiently over the same subchannel for enhancing the computation offloading performance [9]–[11].

However, MEC still comes with a set of challenges in industrial applications. First, with the increasing number of IIoT devices, more computing resources are allocated to the offloaded tasks, thus causing higher overloads of MEC servers. The computation burden of MEC servers may result in the higher network configuration costs and the inefficiency of processing such offloaded tasks timely [12]. Second, the capabilities of IIoT devices for making the right offloading decisions between local computing and edge processing are constrained by their computation offloading policies and computing requirements (e.g., delay budget). The offloading decision made by a single IIoT device is always based on its own performance gain, e.g., the task offloading rate, without accounting for the sheer number of other IIoT devices. Hence, there is a performance bottleneck brought by the large-scale task offloading, which also reveals some concerns particularly with the real-time task processing and the optimized computation decision-making strategy for enabling MEC applications [12]–[14].

Rather than boosting edge intelligence, 6G is envisioned to empower edge clouds created by digitalization and hyper-connectivity of everything. To achieve multi-dimensional data integration and digital visualization, digital twins (DTs) have emerged as one of the most cutting-edge technologies in 6G, providing near-instant unlimited connectivity and ultra-reliable low latency communications [15]. Being a bridge connecting physical and digital spaces, a DT is a high-fidelity virtual...
representation of a physical entity in the digital space, enabled by synchronized data acquisition about its function, model, and processing capability, etc., across the entire life cycle [15–18]. Through a combination of DTs and MEC in industrial applications, the computation state of the MEC systems, e.g., the central processing unit (CPU) clock frequency of an MEC server, can be monitored and predicted in real time within the DT modelling cycle. Such a combination not only improves the offloading decision-making efficiency, but also enhances the computation performance in IIoT. As an emerging architecture, DT-assisted MEC has been recognized as a key enabler for realizing smart manufacturing and Industry 4.0 [19].

A. Motivation and Contributions

Incorporating DTs and NOMA into MEC is not only an extension of traditional MEC, but also a practical application incentive promoted to provide significant performance gains in terms of spectral efficiency (SE), energy efficiency, task completion delay, etc. To fully unleash the potentials of such an envisioned integration design, the following challenges need to be well tackled. First, with NOMA, every IIoT device is capable of being associated with any IGW by multiplexing on the same subchannel with other IIoT devices for task offloading. Hence, an increased complexity incurred by offloading decisions of different tasks makes edge association and subchannel assignment a challenging problem. Second, due to the co-channel interference between IIoT devices served by the same IGW, flexible and efficient resource allocation should be carefully designed to coordinate the intra-cluster interference and optimize the offloading decisions. Third, it is challenging to efficiently process and control large-scale offloaded tasks by the MEC server only, when the enormous number of IIoT devices choose to offload the task via NOMA. With the given network configuration costs, the computing performance, e.g., the task completion delay, can be potentially improved by integrating DTs into MEC systems. Therefore, it is particularly necessary to jointly consider the edge association and resource allocation for further optimizing the offloading decision and enhancing the offloading performance.

Motivated by the above discussions, we focus on a DT-assisted MEC system where IIoT devices employ the NOMA to offload their industrial computation tasks to the IGWs simultaneously. In this regard, we investigate the problem of joint edge association as well as computation and communication resource allocation in such a newly emerging MEC architecture, aiming to minimize the total task completion delay of all IIoT devices. To the best of our knowledge, this paper is the first trial that addresses the problem of joint edge association and resource allocation in the DT-assisted MEC system with NOMA for emerging industrial applications. The major contributions are summarized as follows:

- We design a two-layer DT-assisted MEC architecture for the IIoT system, where a DT server is integrated at the edge layer to act as a digital model of the MEC server, and NOMA is used by each IGW at the end layer to serve the IIoT devices for task offloading. In particular, the DT server is created to assist the offloading decision-making by estimating the state of the MEC server in terms of the CPU clock frequency. We consider the idle and busy states of the MEC server via two hypotheses to determine the computation delay of edge execution assisted by the DT server. Our architecture is the first time in the literature to identify a close coupling of DT-assisted MEC and NOMA-enabled task offloading in IIoT.
- We formulate a joint optimization problem of the IGW’s subchannel assignment as well as the computation capacity allocation, edge association, and transmit power control of IIoT devices, with the objective of minimizing the total completion delay for processing tasks across all IIoT devices, under the constraints of their computing services. We note that the resulted problem is a mixed integer non-convex optimization problem, which is proved to be NP-hard and very difficult to solve directly.
- We then decompose it into four solvable sub-problems: power allocation, subchannel assignment, computation capacity allocation, and edge association optimization. In particular, the subchannel assignment sub-problem is solved by an efficient heuristic algorithm for achieving the maximum achievable rate in total within each NOMA cluster. The sub-problem for optimizing edge association to identify the offloading decision is modeled as a two-sided many-to-one matching game process between IIoT devices and subchannels, and is solved by a many-to-one matching algorithm which yields a stable matching result. An overall alternating optimization algorithm is proposed to solve the sub-problems iteratively, whose convergence is also proved by the non-increasing value of the objective function after each iteration.
- We validate our proposed scheme by extensive numerical simulations. Comparing with the representative benchmark schemes, using the DT to assist the MEC systems in IIoT can contribute to not only the reduction in the total task completion delay, but also the increase in the percentage of offloading IIoT devices.

B. Paper Organization

The rest of this paper is organized as follows. We review the related work in Section II. The system model and problem formulation are described in Section III. Section IV proposes the overall iterative algorithm to solve the optimization problem. Simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

A. Digital Twin in MEC Systems

Due to the great potential of additional computing performance gains, the combination of DTs and MEC has recently received a growing consideration from the research community. In [13], Sun et al. utilized the DTs of MEC servers to assist offloading decision by estimating the states of their physical counterparts and providing the training data for decision agent in deep reinforcement learning (DRL) framework. By taking the edge collaboration into account, Liu et al. in [20] designed a DT-assisted task offloading scheme, depending on the selection of cooperative and reliable MEC servers via the data acquisition about channel state information as well as the
blockchain application in DTs. With the aid of using DTs to capture the running states and behaviors of end devices, Lu et al. in [12] developed a blockchain-enabled federated learning (FL) scheme to enhance learning security and data privacy protection for end devices. In [21], Fan et al. used DTs to create a virtual replica of MEC networks for assisting lane-changing decisions, aiming to make the connected vehicles orchestrate and evaluate lane-changing strategies more intelligently.

More recently, there has been significant interest in studying the DT-assisted MEC in industrial applications. In [13], Duy et al. explored the end-to-end latency minimization problem in DT-assisted MEC for industrial automation, wherein the DTs of MEC servers provides the estimated computation capability by monitoring the current states of physical counterparts. Similar to the proposed DT edge network architecture in [12], Lu et al. in [22] also employed the FL to build the DT models of IIoT devices according to their historical running data, with the benefits of transmission overhead reduction and data privacy enhancement in IIoT. However, all these related works in DT-assisted MEC have not considered the usage states of the MEC server during the DT modelling cycle, when they create the DTs to represent the MEC server in physical space. Note that in practice, there exist idle and busy states regarding the MEC server, such that the probabilities of these two states should be well taken into account in the design of computation performance metrics, e.g., the edge processing delay. By contrast, we denote the idle and busy states of the MEC server by two hypotheses, and with the probabilities of these two hypotheses, we then formulate the computation delay of edge execution for each IIoT device.

B. NOMA-Enabled Task Offloading in MEC Scenarios

Motivated by the benefits of MEC and NOMA, significant amount of research efforts has been spent to integrate NOMA with task offloading into MEC. In [10], Ye et al. presented a hybrid offloading scheme in the NOMA-enabled MEC network, and formulated the successful computation probability maximization problem, revealing the advantages of NOMA over OMA. By identifying the interaction between the differentiated offloading delay and co-channel interference of NOMA users, Sheng et al. in [23] devised a task offloading scheme to minimize the average overall offloading delay by jointly optimizing offloading decision and resource allocation. In [24], Fang et al. formulated an optimization problem to reduce the task completion time by jointly optimizing the task partition ratios and offloading transmit power for each user. In [25], Pham et al. explored the task offloading issue in multi-carrier NOMA-enabled MEC systems and adopted the coalition formation game to find the solution to the formulated total computation overhead minimization problem.

Several recent works have also been devoted to applying NOMA-enabled computation offloading in IIoT systems. In [9], Qian et al. proposed a total energy consumption minimization problem by jointly optimizing NOMA-transmission duration, offloading decision, and computation-resource allocation of each IIoT device. In [11], Zhao et al. presented a DRL-based joint resource allocation and IIoT device orchestration policy via NOMA, aiming to achieve the more accurate model and reduce overhead for MEC-assisted hierarchical FL in IIoT systems. By considering channel states and computation task requests, Tuong et al. in [26] designed an efficient scheme to minimize the average task delay of all IIoT devices by jointly optimizing the subchannel assignment, offloading decision, and computation resource allocation. By the help of other IIoT devices with rich computation resources, Zhu et al. in [27] proposed a machine-to-machine-assisted NOMA-based MEC scheme to enhance computation services, and formulated a system energy consumption minimization problem by jointly optimizing the sleep-scheduling and resource allocation. Although these solutions provide certain insights in applying NOMA to enable task offloading in MEC scenarios efficiently, they cannot capture the effect of digital visualization design on overall computation performance by virtually representing the MEC server. By contrast, this paper considers a combination of DTs and MEC in IIoT by employing NOMA to enable task offloading, aiming to further achieve considerable computation performance gains.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Scenario Description

Consider a DT-assisted MEC architecture for the IIoT scenario as shown in Fig. 1 which consists of two layers with different functionalities and properties, i.e., the end layer and the edge layer. In the end layer, a set $M = \{1, 2, \cdots, M\}$ of IIoT devices are randomly distributed on the ground of a smart factory. We consider a binary offloading policy, such that the industrial computation task of each IIoT device is either executed locally or offloaded for edge processing.$^{1}$ The task of IIoT device $m$ is described by a two-tuple $D_m = (\eta_m, \lambda_m)$, where $\eta_m$ is the size of task-input bits in total to be executed and $\lambda_m$ is the number of required CPU cycles to accomplish task $D_m$, for $m \in M$. In the edge layer, an MEC server is integrated on a central CAP of the smart factory to provide computing services for IIoT devices. We denote $f$, as the

---

$^1$Note that in practice, there may be different types of computation tasks for IIoT devices, which depends on the exact industrial applications. For the highly integrated industrial process across the production life cycle in Industry 4.0, the computation task cannot be partitioned and has to be processed as a whole either locally at the IIoT device or offloaded to the MEC server.
clock frequency of the CPU chip to describe the computation capability of the MEC server.

To make the convergence of edge and end layers, a set \( \mathcal{N} = \{1, 2, \ldots, N\} \) of IGWs, as the industrial wireless access portals, are deployed uniformly and relatively close to the IIoT devices in the smart factory, for offloading the task-input bits from them and then sending the aggregation of raw data to the CAP. Note that in this work, we consider that the IGWs are connected to the CAP via dedicated and high-capacity wired backhauls, e.g., optical fiber or Ethernet \([28]\).

**B. NOMA-Enabled Task Offloading Model**

We focus on one particular type of the task offloading scenario, in which the IIoT devices employ the power-domain NOMA to offload their task-input bits to the IGWs simultaneously. The CAP divides the overall available bandwidth into a set \( \mathcal{K} = \{1, 2, \ldots, K\} \) of subchannels, each with an equally-sized bandwidth of \( B \). By using NOMA, IIoT devices served by the same IGW \( n \) are grouped into a NOMA cluster \( \mathcal{M}^k_n \), where \( |\mathcal{M}^k_n| \) IIoT devices multiplex on the same subchannel \( k \) at the same time for task offloading, for \( \mathcal{M}^k_n \subseteq \mathcal{M} \). In particular, each IIoT device can and only can be grouped into one NOMA cluster, and each subchannel can and only can be assigned to one IGW. For convenience, we define a subchannel assignment indicator for IGW \( n \) and subchannel \( k \), i.e.,

\[
\alpha^k_n = \begin{cases} 
1, & \text{if subchannel } k \text{ is assigned to IGW } n, \\
0, & \text{otherwise.}
\end{cases}
\]

Denote \( j \) as the order of IIoT device \( j \) in NOMA cluster \( \mathcal{M}^k_n \), and \( J_n \) as the number of IIoT devices served by IGW \( n \) in \( \mathcal{M}^k_n \), for \( J_n = |\mathcal{M}^k_n| \). Therefore, there are a set \( \mathcal{J}_n = \{1, 2, \ldots, J_n\} \) of orders in \( \mathcal{M}^k_n \). Without loss of generality, \( J_n \) IIoT devices in NOMA cluster \( \mathcal{M}^k_n \) are sorted by their channel gains in the ascending order, i.e., \( |h^k_{1,n}|^2 \leq |h^k_{2,n}|^2 \leq \cdots \leq |h^k_{J_n,n}|^2 \leq |h^k_{J_n+1,n}|^2 \), where \( h^k_{j,n} \) is the channel coefficient from IIoT device \( j \) to IGW \( n \) on subchannel \( k \), for \( m \in \mathcal{M}^k_n \). By applying the SIC technique, IGW \( n \) first decodes the message from IIoT device \( j \), for \( j < m \) and \( j \in \mathcal{J}_n \), and then removes this message from its received signals, in the order of \( j = 1, 2, \ldots, m-1 \). Through the sequential decoding, the signals from IIoT device \( j \) in \( \mathcal{M}^k_n \) can be treated as the interference, for \( j > m \). As a result, the signal-to-interference-plus-noise ratio (SINR) received at IGW \( n \) on subchannel \( k \) for IIoT device \( m \) in NOMA cluster \( \mathcal{M}^k_n \) can be specified by

\[
\gamma^k_m = \frac{|h^k_{m,n}|^2 p^k_{m,n}}{\sigma^2_n + \sum_{j=m+1}^{J_n} |h^k_{j,n}|^2 p^k_{j,n}},
\]

where \( p^k_{m,n} \) is the transmit power of IIoT device \( m \) to IGW \( n \) on subchannel \( k \), and \( \sigma^2_n \) is the power of the additive white Gaussian noise (AWGN) at IGW \( n \).

To indicate that IIoT device \( m \) is assigned to only one order (e.g., the \( j \)-th order) of NOMA cluster \( \mathcal{M}^k_n \) to associate with IGW \( n \), a binary variable is also introduced, which is given by

\[
\beta^k_{m,n} = \begin{cases} 
1, & \text{if IIoT device } m \text{ is assigned to the } j \text{-th order of } \mathcal{M}^k_n, \\
0, & \text{otherwise.}
\end{cases}
\]

Since each IIoT device can be assigned to at most one order of NOMA cluster for task offloading, the binary offloading policy for IIoT device \( m \) can be then determined as

\[
\sum_{j=1}^{J_n} \sum_{n=1}^{N} \beta^k_{m,n} = 1, \quad \text{task } D_m \text{ is offloaded to IGW } n, \quad \sum_{j=1}^{J_n} \sum_{n=1}^{N} \beta^k_{m,n} = 0, \quad \text{task } D_m \text{ is executed locally.} \tag{4}
\]

Therefore, the achievable rate of IIoT device \( m \) for task offloading can be given by

\[
R_m = \sum_{n=1}^{N} \sum_{j=1}^{J_n} \sum_{k=1}^{K} \alpha^k_n \times B \log_2 \left( 1 + \frac{|p^k_{m,n}|^2}{\sigma^2_n + \sum_{j=m+1}^{J_n} |h^k_{j,n}|^2 p^k_{j,n}} \right) \tag{5}
\]

Denote \( \zeta_m \) as the overhead during task offloading, e.g., data encryption and channel encoding, for IIoT device \( m \). For task \( D_m \), the actual size of task-input bits to be offloaded from IIoT device \( m \) to IGW \( n \) is given by \( \zeta_m \eta_m \). Thereby, the offloading time of IIoT device \( m \) for task \( D_m \) can be derived as

\[
T_m = \frac{\zeta_m \eta_m}{R_m}. \tag{6}
\]

Due to the wired backhauling used for data aggregation as mentioned, we do not consider the time overhead for each IGW to convey the aggregation of raw data to the CAP.

**C. Digital Twin Model**

To optimize the edge execution for the offloaded tasks, we integrate the DT into the edge layer to create a DT server that serves as the real-time digital counterpart of the MEC server. The whole DT framework is designed by a three-dimensional model entity assembly of a physical space, a virtual space, and a connection between physical and virtual spaces, as shown in Fig.1. By help of sensors, the multi-dimensional data, e.g., the function, model, processing capability, etc., are collected at the MEC server in physical space, and are then sent to the DT server in virtual space through the connection \([29]\). By means of the real-time interaction, the offloading decision-making is performed at both MEC and DT servers.

Being a virtual representation of the MEC server, the DT server provides the estimated CPU frequency to reflect the computing performance of the physical counterpart in terms of current computation state. As discussed in \([13], [14]\), the DT server may have an estimated deviation between the real state and the estimated state of the MEC server. Let \( \tilde{f}_e \) be the estimated deviation of CPU frequency to denote the deviation between the physical MEC server and its DT counterpart, which can be given as either positive or negative. From the perspective of the MEC server, the DT server \( \mathcal{D}_e \) can be then formulated as

\[
\mathcal{D}_e \triangleq \Theta \left( \tilde{f}_e; \tilde{f}_e \right), \tag{7}
\]

where \( \tilde{f}_e \) is the estimated CPU frequency of the MEC server.

**D. Computing Model**

1) Local Computing: For local task computing, we use \( f^L_m \) to denote the CPU frequency of IIoT device \( m \) as the local computation capability. Thus, the computation delay of IIoT device \( m \) for processing task \( D_m \) locally can be expressed by
\[ T^L_m = \lambda_m \left( f^L_m \right)^{-1}. \]  

(8)

2) Digital Twin-Assisted Edge Computing: From the DT’s perspective, the usage of MEC server in physical space can be described in terms of two states, idle and busy. The idle state means the computation task is being processed by the MEC server, whereas the busy state indicates the DT server is being used to assist the offloading decision-making for enhancing the computing performance. The idle and busy states of the MEC server are respectively represented by hypotheses \( \mathcal{H}_0 \) and \( \mathcal{H}_1 \), whose probabilities can be estimated as follows

\[ \Pr \{ \mathcal{H}_0 \} = \frac{\xi_0}{\xi_0 + \xi_1} \quad \text{and} \quad \Pr \{ \mathcal{H}_1 \} = \frac{\xi_1}{\xi_0 + \xi_1}, \]  

(9)

where \( \xi_0 \) and \( \xi_1 \) are the transition rates from idle to busy and busy to idle, respectively.

Denote the number of CPU cycles required to process one bit of raw data at the MEC server as \( \phi \), which is determined by the nature of industrial tasks. As a result, the total CPU cycles required to process task \( D_m \) of IIoT device can be written as \( \phi q_m \eta_m \). Similar to [13], we assume that the deviation between the DT server \( \mathcal{D}_e \) and its physical counterpart can be obtained in advance. For IIoT device \( m \), the computation delay gap between DT-assisted edge computing and real execution of MEC server for processing task \( D_m \) can be thus given by

\[ G^C_m = \frac{\phi q_m \eta_m}{q_m \hat{f}_e} - \frac{\phi q_m \eta_m}{q_m \hat{f}_e} = -\frac{\phi q_m \eta_m}{q_m \hat{f}_e} (\hat{f}_e + f_e), \]  

(10)

where \( q_m \) is the proportion of computation capacity allocated to task \( D_m \) of IIoT device \( m \) by the MEC server.

Denote \( T^M_m \) and \( T^{DT}_m \) as the computation delay of real execution of the MEC server and DT-aided edge computing for processing task \( D_m \), respectively. Given delay gap \( G^C_m \) in (10), we can derive \( T^M_m = T^{DT}_m + G^C_m \). Thus, the computation delay of edge processing for IIoT device \( m \) can be written as

\[ T^C_m = \Pr \{ \mathcal{H}_0 \} T^M_m + \Pr \{ \mathcal{H}_1 \} T^{DT}_m = \frac{\xi_0}{\xi_0 + \xi_1} \phi q_m \eta_m (\hat{f}_e + f_e) + \frac{\xi_1}{\xi_0 + \xi_1} \phi q_m \eta_m (\hat{f}_e + f_e) \]  

(11)

As a result, the completion delay for executing task \( D_m \) when IIoT device \( m \) chooses to perform edge computing can be obtained by

\[ T^E_m = T^U_m + T^C_m. \]  

(12)

By integrating the binary offloading policy as obtained in (4), the actual completion delay spent in processing task \( D_m \) for IIoT device \( m \) can be calculated by

\[ T_m = \left( 1 - \sum_{n=1}^{N} \sum_{j=1}^{J_n} \beta^j_{m,n} \right) T^L_m + \sum_{n=1}^{N} \sum_{j=1}^{J_n} \beta^j_{m,n} T^E_m. \]  

(13)

E. Problem Formulation

Denote \( \mathcal{M}_{off} = \bigcup_{m \in \mathcal{N}} \mathcal{M}_m^b \) as the set of IIoT devices that choose to perform edge computing via task offloading. Let us further define \( \alpha = \{ \alpha^n_k, \forall n, k \}, \beta = \{ \beta^j_{m,n}, j \in \mathcal{J}_n, \forall m, n \}, \mathcal{P} = \{ p^k_{m,n}, \forall m, n, k \}, \) and \( \mathcal{Q} = \{ q_m, m \in \mathcal{M}_{off} \} \). By taking into account the constraints of computing services of all IIoT devices, we aim to minimize the total completion delay for processing tasks among them by jointly optimizing the transmit power allocation (i.e., \( \mathcal{P} \)), the subchannel assignment (i.e., \( \alpha \)), the computation capacity allocation (i.e., \( \mathcal{Q} \)), and the edge association (i.e., \( \beta \)). Specifically, the task completion delay minimization problem can be formulated as

\[
\min_{\alpha, \beta, \mathcal{P}, \mathcal{Q}} \sum_{m=1}^{M} T_m \quad \text{s.t.} \quad \sum_{n=1}^{N} \alpha^n_k \leq 1, \forall k \in \mathcal{K}, \quad \alpha^n_k \in \{0,1\}, \forall n \in \mathcal{N}, k \in \mathcal{K},
\]

\[
\sum_{j=1}^{J_n} \beta^j_{m,n} \leq 1, \forall m \in \mathcal{M}, \quad \sum_{j=1}^{J_n} \beta^j_{m,n} \leq J_n, \forall n \in \mathcal{N},
\]

\[
\beta^j_{m,n} \in \{0,1\}, \forall m \in \mathcal{M}, n \in \mathcal{N}, j \in \mathcal{J}_n,
\]

\[
0 \leq p^k_{m,n} \leq P^\text{max}_m, \forall m \in \mathcal{M}, n \in \mathcal{N}, k \in \mathcal{K},
\]

\[
q_m \geq 0, \forall m \in \mathcal{M}_{off},
\]

\[
\sum_{m=1}^{M} q_m \hat{f}_e \leq \hat{f}_e, \forall m \in \mathcal{M}_{off}.
\]

In problem (14), a subchannel can be assigned to an IGW only shown in (14b) and only once as in (14e). (14e) specifies the binary offloading policy by which computation task of each IIoT device can be either processed locally or offloaded for edge execution. (14e) represents that the maximum number of devices IIoT devices are served by each IGW in a NOMA cluster. (14f) is designed to give assignment priority to a lower value of order over all the higher values of order in a NOMA cluster. (14g) details the binary bound of edge association. (14h) ensures that the transmit power of IIoT device cannot exceed its maximum transmit power denoted by \( P^\text{max}_m \). (14i) means that the computation capacity allocated to each IIoT device must be non-negative. Finally, (14j) indicates an upper bound \( \hat{f}_e \) of computation capability for the DT server. The NP-hardness of problem (14) can be also shown as follows.

**Theorem 1.** Problem (14) is NP-hard.

**Proof:** We consider an instance of problem (14), where each subchannel can and only can be assigned to one IGW, and simultaneously, an IIoT device can be assigned to only one order of NOMA cluster to associate with that IGW. Such an instance of problem (14) can be regarded as a three dimension matching process, which has been proved to be NP-complete in [30]. The proof is similar to [30], and we omit it here. ■
Problem (14) is a mixed integer non-convex optimization problem, which is NP-hard and extremely difficult to derive an optimal solution due to the following observations. First, the optimization variables $\alpha$ and $\beta$ for subchannel assignment and edge association are binary, and thereby (14b) and (14g) involve integer constraints. Second, the optimization variables $\alpha$, $\beta$, $P$, and $Q$ are closely coupled in the objective function, which results in non-convexity of (14). Therefore, in the next section, we will develop a low-complexity to solve the problem.

IV. PROPOSED SOLUTION

To solve this NP-hard problem efficiently, in this section, we decouple the original problem (14) into four sub-problems: transmit power optimization, subchannel assignment optimization, computation capacity optimization, and edge association optimization. Then, an overall algorithm is designed via alternatingly optimizing the sub-problems, such that the total task completion delay can be iteratively reduced until convergence.

A. Transmit Power Optimization

Given $\alpha$, $\beta$, and $Q$, the sub-problem for optimizing transmit power allocation $P$ can be expressed as

$$\min_{P} \sum_{m=1}^{J_n} \frac{\phi_m \eta_m}{B} \log_2 \left(1 + \frac{|h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{j=m+1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right)$$

s.t. $0 \leq p_{m,n}^k \leq P_{\text{max}}, \forall m \in M_n^k, n \in N, k \in K$. (15a)

Problem (15) is a non-convex optimization problem since there exist the interference terms in objective function (15a). To make problem (15) efficiently solved, we intend to derive the sub-optimal solution by maximizing the total achievable rate of $J_n$ IIoT devices served by IGW $n$ in NOMA cluster $M_n^k$. To this aim, the maximum achievable rate of all IIoT devices in a given NOMA cluster is analytically derived as the following theorem.

**Theorem 2.** The maximum achievable rate of $J_n$ IIoT devices in total, served by IGW $n$ in $M_n^k$, can be obtained by

$$\max \sum_{m=1}^{J_n} R_m = B \log_2 \left(1 + \frac{1}{\sigma_n^2} \sum_{m=1}^{J_n} |h_{m,n}^k|^2 P_{\text{max}} \right).$$

**Proof:** Please see Appendix A.

Due to the intra-cluster interference from other IIoT devices, the transmit power allocated to each IIoT device is difficult to approach the maximum achievable rate $R_{\text{max}}$ at the same time. To further optimize $p_{m,n}^k$, we first focus on the achievable rate of IIoT device $m$ in $M_n^k$, for $m = J_n$. Due to $R_{J_n} = B \log_2 \left(1 + \frac{1}{\sigma_n^2} |h_{J_n,n}^k|^2 p_{J_n,n}^k \right)$, we thus determine that

$$p_{J_n,n}^k = \left( \frac{1}{|h_{J_n,n}^k|^2} \right)^{\frac{1}{2}} \left(2B^{-1}R_{J_n} - 1 \right).$$

For any $m < J_n$ in $M_n^k$, the use of Theorem 2 yields to

$$R_{m+1} + R_{m+2} + \cdots + R_{J_n} = B \log_2 \left(1 + \frac{1}{\sigma_n^2} \sum_{j=m+1}^{J_n} |h_{j,n}^k|^2 p_{J_n,n}^k \right).$$

Performing some algebraic manipulations, we then obtain

$$p_{m,n}^k = \left(2B^{-1}R_m - 1 \right)^{\frac{1}{2}} \left(2B^{-1}\sum_{j=m+1}^{J_n} |h_{j,n}^k|^2 p_{J_n,n}^k \right).$$

(19)

Since $R_m \leq R_{\text{max}}$, for $m \in M_n^k$, the optimal power $p_{m,n}^k$ as the sub-optimal solution to (15) can be derived by

$$p_{m,n}^k = \frac{\sigma_n^2}{2B^{-1}R_{\text{max}} - 1} \left(2B^{-1}(R_{m+1} + R_{m+2} + \cdots + R_{J_n}) \right).$$

(20)

Using (20), problem (15) can be thus solved as follows

$$\sum_{m=1}^{J_n} R_m = \sum_{m=1}^{J_n} \phi_m \eta_m \frac{1}{B} \log_2 \left(1 + \frac{|h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{j=m+1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right).$$

(21)

B. Subchannel Assignment Optimization

Given $P$, $\beta$, and $Q$, the sub-problem for optimizing subchannel assignment $\alpha$ can be given by

$$\min_{\alpha} \sum_{k=1}^{K} \sum_{m=1}^{J_n} \phi_m \eta_m \alpha_k \frac{1}{B} \log_2 \left(1 + \frac{|h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{j=m+1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right).$$

(22a)

s.t. (14b) and (14c).

(22b)

Problem (22) is a non-convex optimization problem since the constraints (14b) and (14c) are non-concave. In general, there is no efficient method to obtain the optimal solution for such a non-convex problem. We can observe from Theorem 2 that the maximum achievable rate of $J_n$ IIoT devices in total, served by IGW $n$ in NOMA cluster $M_n^k$, can be obtained by the derivation of the optimal power $p_{m,n}^k$. This thus motivates us to find the optimal IGW corresponding to the optimal subchannel which can achieve the maximum achievable rate in total within NOMA cluster.

To reduce the computational complexity, we then design an efficient heuristic algorithm for subchannel assignment. The procedure of obtaining the optimal solution to (22) is outlined in Algorithm 1. Specifically, we denote $K'$ as the set of subchannels remaining to be assigned and $N'$ as the set of the remaining IGWs to be allocated. The optimal subchannel can be found corresponding to the IGW which can achieve the maximum achievable rate in total. The sets $K'$ and $N'$ are accordingly updated with the assignment of subchannels.

C. Computation Capacity Optimization

Given $\alpha$, $\beta$, and $P$, the sub-problem for optimizing computation capacity allocation $Q$ can be formulated as

$$\min_{Q} \sum_{m \in M_n^k} \left( \frac{\phi_m \eta_m}{q_m (f_e + f_r)} + \frac{\xi_1}{\phi_m \eta_m} \frac{1}{f_e + f_r} \right).$$

(23a)

s.t. (14d) and (14e).

(23b)

Before solving (23), we first show that the convexity of this problem in Proposition 1 as given below.

**Proposition 1.** Problem (23) is convex.

**Proof:** Please see Appendix B.
Algorithm 1 Subchannel Assignment Algorithm for Solving Problem (22).

1: Initialization: optimal transmit power $p_{m,n}^k$, $\forall m$, according to (20).
2: Set $N' = N$, $K' = K$, and $p_{m,n}^k = p_{m,n}^k$, $\forall m, n, k$.
3: for all $m \in M_{off}$ do
4: Calculate $\sum_{m \in M_{k}^h} R_m$ according to (21).
5: end for
6: for all $k \in K'$ do
7: $n^* = \arg \max_{n \in N'} \sum_{m \in M_k^h} R_m$, $\forall n \in N'$.
8: Set $\alpha_k^m = 1$.
9: Set $N'' = N'' \setminus \{n^*\}$ and $K'' = K'' \setminus \{k\}$.
10: end for
11: Output: $\alpha^* = \{\alpha_k^m^*\}$.

Since problem (23) is a convex optimization problem, we then adopt the Lagrangian dual decomposition method to solve it to obtain the optimal computation capacity allocation.

Theorem 3. The optimal solution to (23) is given as a closed-form expression, i.e.,

$$q_m^* = \frac{F \sqrt{\phi_m \eta_m}}{\sum_{m \in M_{off}} \sqrt{\phi_m \eta_m}},$$

where $F = \tilde{E}_e F_{max}$.

Proof: Please see Appendix C.

D. Edge Association Optimization

Given $\alpha$, $P$, and $Q$, the sub-problem for optimizing edge association $\beta$ can be written by

$$\min_{\beta} \sum_{m=1}^{M} \left(1 - \sum_{n=1}^{N} \sum_{j=1}^{J_n} \beta_{j,m,n}^k \right) T_m^L + \sum_{n=1}^{N} \sum_{j=1}^{J_n} \beta_{j,m,n}^k T_m^E,$$

subject to (14d), (14e), (14f), and (14g).

Problem (25) is a 0-1 integer programming problem with a non-convex objective function, aiming to find an assignment of either 0 or 1 to binary association variable $\beta$ for choosing either local computing or task offloading. To solve problem (25) with a low-complexity algorithm, we recognize that this problem can be modeled as a two-sided many-to-one matching process. For task offloading, each IoT device can and only can be allowed to access at most one subchannel in a NOMA cluster, and each subchannel can and only can be assigned with at most $J_n$ IoT devices. For local computing, there is no subchannel assigned to the IoT device. Therefore, the IoT devices and the subchannels act as two disjoint sets of players to be mutually matched with purpose of minimizing the total completion delay, thus the solution can be obtained.

Definition 1. Given two disjoint sets, $M$ of IoT devices, and $K$ of subchannels, a bijective function $\Psi : M \cup K \rightarrow M \cup K \cup \{0\}$ is a many-to-one matching such that the following conditions are satisfied for all $m \in M$ and $k \in K$:

1) $\Psi(m) \in K \cup \{0\}$ with $|\Psi(m)| = 1$;
2) $\Psi(k) \in M$ with $|\Psi(k)| \leq J_n$;
3) $m \in \Psi(k) \Leftrightarrow k \in \Psi(m)$.

where $\Psi(m) = \{0\}$ means that the IoT device is matched with subchannel $\{0\}$, thus implying that the computation task is executed locally. Condition 1 implies that each IoT device can and only can be matched with one subchannel. Condition 2 indicates that each subchannel can and only can be matched with at most $J_n$ IoT devices. Condition 3 implies that if IoT device $m$ is matched with subchannel $k$, then subchannel $k$ is also matched with IoT device $m$.

Under this matching framework, each IoT device $m$ only concerns about the actual completion delay for task processing via choosing either local computing (on subchannel $\{0\}$) or task offloading (on subchannel $k$). Thus, the preference value of IoT device $m$ on subchannel $k$ can be defined as

$$V_m(k) = \begin{cases} T_m^U + T_m^C, & \text{if matched to subchannel } k, \\ T_m^L, & \text{if matched to subchannel } \{0\}. \end{cases}$$

While for each subchannel $k$, it concerns about the sum of completion delay of all IoT devices served by IGW $n$ in NOMA cluster $M_n^k$. Therefore, the preference value of subchannel $k$ under matching $\Psi$ can be given by

$$V_k(\Psi(k) = \{m, k\}) = \sum_{m \in M_n^k} (T_m^U + T_m^C).$$

To better indicate the preference of each player over the set of other players, a preference relation $\succ$ is adopted for both IoT devices and subchannels. Specifically, we define $\succ_m$ as the preference relation of IoT device $m$ over the set of subchannels. For any two subchannels $k, k' \in K \cup \{0\}, k \neq k'$, the preference relation of two matchings $\Psi, \Psi'$, for $k = \Psi(m)$ and $k' = \Psi'(m)$, can be written as

$$(k, \Psi) \succ_m (k', \Psi') \Leftrightarrow V_m(k) < V_m(k'),$$

which implies that IoT device $m$ prefers subchannel $k$ than $k'$, only if IoT device $m$ obtains the lower completion delay in task processing on subchannel $k$ than that on subchannel $k'$.

We further denote $\succ_k$ as the preference relation of subchannel $k$ over the set of IoT devices. For any two subsets of IoT devices $M_n^k, M_n^{k'} \subset M, M_n^k \neq M_n^{k'},$ the preference relation of subchannel $k$ on two matchings $\Psi, \Psi'$, for $k = \Psi(m)$ and $k' = \Psi'(m)$, can be given by

$$(M_n^k, \Psi) \succ_k (M_n^{k'}, \Psi') \Leftrightarrow V_k(M_n^k) < V_k(M_n^{k'}),$$

which indicates that subchannel $k$ prefers the subset of IoT devices $M_n^k$ than $M_n^{k'}$, only if the sum of completion delay in $M_n^k$ for subchannel $k$ is lower than that in $M_n^{k'}$.

From (5), (6), and (11), we find that the preference value of each IoT device is related to not only the subchannel it matches, but also the other IoT devices matched to the same subchannel due to the co-channel interference and competition behavior of each IoT device for computation resource. Hence, Definition 1 is a type of many-to-one matching game with externalities (31). To tackle the externalities, we resort to the swap operation between any two IoT devices to exchange their matched subchannels, while keeping other IoT devices matchings unchanged, followed by the idea of swap matching.
Algorithm 2 Many-to-One Matching Based Edge Association Optimization Algorithm for Solving Problem (25).

1: Initialization: Given a feasible matching $\Psi = \Psi_0$.
2: Each IIoT device is randomly matched to any subchannel $k \in K \cup \{0\}$, subject to $|\Psi(m)| = 1$ and $|\Psi(k)| \leq J_n$.
3: Obtain $P$, $\alpha$, and $Q$ according to Algorithm 1 and [23], respectively.
4: repeat
5: for all $m \in M$ do
6: for all $m' \in M$ do
7: Calculate preference values of all involved IIoT devices and subchannels under matching $\Psi^m_m$.
8: if $(m, m')$ is a swap-blocking pair with $m \in \Psi(k)$ and $m' \in \Psi(k')$ then
9: Update $\Psi = \Psi^m_m$.
10: end if
11: end for
12: end for
13: until There exists no swap-blocking pair

Definition 2. Given a matching $\Psi$, a pair of IIoT devices $m, m' \in M$, and subchannels $k, k' \in K \cup \{0\}$ with $m \in \Psi(k)$, $m' \in \Psi(k')$, a swap matching is defined as $\Psi^m_m = \Psi \setminus \{(m, k), (m', k')\} \cup \{(m', k'), (m, k)\}$. Here, a matching $\Psi$ is stable if there exists no swap-matching $\Psi^m_m(k)$, such that
1) $\forall \gamma \in (m, m', k, k')$, $V_e(\Psi^m_m(\gamma)) \leq V_e(\Psi(\gamma))$ and
2) $\exists \gamma \in (m, m', k, k')$, $V_e(\Psi^m_m(\gamma)) < V_e(\Psi(\gamma))$.

From Definition 2, if there exists a swap matching $\Psi^m_m$, a pair of IIoT devices $(m, m')$ is called a swap-blocking pair in $\Psi$ with $m \in \Psi(k)$ and $m' \in \Psi(k')$. To elaborate, if a swap matching between two IIoT devices is approved, the completion delay for task processing of any player involved will not increase, and at least one of the players’ completion delay will decrease. To obtain a stable matching result as given in Definition 2, we propose a many-to-one matching algorithm for deriving the solution to the edge association optimization problem, as shown in Algorithm 2.

Note that in Algorithm 2, we first assume an initial feasible matching, in which IIoT devices and subchannels are randomly matched with each other. Then, the power control, subchannel assignment, and computation capacity allocation is optimized to calculate the preference values of IIoT devices and subchannels. Next, the swap operation for these two IIoT devices is performed if there exists a swap-blocking pair between them. The algorithm continues until there exists no swap-blocking pair, thus yielding a stable matching result.

E. Overall Algorithm Design

In this subsection, we propose an overall algorithm for alternately optimizing these four sub-problems in an iterative way, and obtain a sub-optimal solution of (14). Algorithm 3 presents the sketch of the proposed algorithm. More precisely, in the $l$-th iteration, it first uses Algorithm 2 to obtain the optimal edge association $\beta^*_{l+1}$ with given $\alpha_l$, $P_l$, $Q_l$. Afterwards, we apply Algorithm 1 to obtain the optimal subchannel assignment $\alpha^*_{l+1}$ with given $\beta^*_{l+1}$, $P_l$, $Q_l$. Next, the sub-optimal transmit power $P^*_{l+1}$ is obtained by solving (15), with given $\alpha^*_{l+1}$, $\beta^*_{l+1}$, $Q_l$. Then, we obtain the optimal computation capacity $Q^*_{l+1}$ by solving (23) with given $\alpha^*_{l+1}$, $\beta^*_{l+1}$, $P^*_{l+1}$. As the iterations continue, the total task completion delay of all IIoT devices will gradually decrease. Finally, a sub-optimal solution to (14) can be derived until Algorithm 3 converges to a prescribed accuracy within finite iterations.

1) Convergence: The convergence of Algorithm 3 can be guaranteed as follows.

Proposition 2. Algorithm 3 is guaranteed to converge.

Proof: Denote $\Xi(\alpha_l, \beta_l, P_l, Q_l)$, $\Xi_1(\alpha_l, \beta_l, P_l, Q_l)$, $\Xi_2(\alpha_l, \beta_l, P_l, Q_l)$, and $\Xi_3(\alpha_l, \beta_l, P_l, Q_l)$ as the objective values of (14), (22), (15), and (23) in the $l$-th iteration, respectively. In the $(l+1)$-th iteration, for given $\alpha_l$, $P_l$, $Q_l$, in step 3 of Algorithm 3 we have

$$\Xi(\alpha_l, \beta^*_{l+1}, P_l, Q_l) \geq \Xi(\alpha_l, \beta^*_{l+1}, P_l, Q_l),$$

(30)

where inequality $(a)$ holds due to the derived optimal solution $\beta^*_{l+1}$ of (25). Second, for given $\beta^*_{l+1}$, $P_l$, $Q_l$, in step 4 of Algorithm 3 it follows that

$$\Xi(\alpha_l, \beta^*_{l+1}, P_l, Q_l) \geq \Xi_1(\alpha^*_{l+1}, \beta^*_{l+1}, P_{l+1}, Q_l),$$

(31)

$$\geq \Xi(\alpha^*_{l+1}, \beta^*_{l+1}, P_{l+1}, Q_l),$$

where inequality $(b)$ holds due to the derived optimal solution $\alpha^*_{l+1}$ of (23), and inequality $(c)$ holds since the objective value of (14) is lower-bounded by that of (23). Third, in step 5 and step 6 of Algorithm 3 one obtains

$$\Xi(\alpha^*_{l+1}, \beta^*_{l+1}, P_{l+1}, Q_l) \geq \Xi_2(\alpha^*_{l+1}, \beta^*_{l+1}, P^*_{l+1}, Q_l),$$

(32)

$$\Xi_2(\alpha^*_{l+1}, \beta^*_{l+1}, P^*_{l+1}, Q_l) \geq \Xi_3(\alpha^*_{l+1}, \beta^*_{l+1}, P^*_{l+1}, Q^*_{l+1}),$$

which can be similarly described as in (31). Then, according to (30)-(32), we have

$$\Xi(\alpha_l, \beta_l, P_l, Q_l) \geq \Xi(\alpha^*_{l+1}, \beta^*_{l+1}, P^*_{l+1}, Q^*_{l+1}),$$

(33)
which concludes that the objective value of (14) is non-increasing after each iteration of Algorithm 3. Therefore, the objective function has a lower bound, and will converge to a fixed value after a finite number of iterations, i.e., Algorithm 3 is convergent.

2) Complexity: The complexity of Algorithm 3 is analyzed as follows. In Step 3, Algorithm 2 requires each IIoT device to perform the swap operations until a stable matching can be obtained, and thus the computational complexity of solving (25) is $O(M(K + 1))$. In Step 4, (22) is a linear optimization problem, which can be solved with computational complexity $O\left(\sqrt{K} + \frac{1}{\epsilon}\right)$, where $K$ denotes the numbers of variables (i.e., the subchannels) and $\epsilon$ denotes the iterative accuracy [33]. Therefore, the total computational complexity of Algorithm 3 is $O\left(L(M(K + 1) + \sqrt{K})\right)$, where $L$ represents the number of iterations for alternately and iteratively optimizing the four sub-problems in Algorithm 3.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed algorithm through numerical simulations. For the purpose of showing the advantages of our proposed algorithm in reducing the total task completion delay and increasing the percentage of offloading IIoT devices, we also compare it with those of three benchmark schemes, which are described as follows:

- **Edge Computing via MEC Server Only (EC-MSO):** All the tasks are processed by the physical MEC server only when offloading without the aid of DTs. Note that there is no computation delay gap between DT-assisted edge computing and real execution of the MEC server. In this case, the computation delay of edge computing for IIoT device $m$ can be expressed by

$$T_m^{EC} = \frac{\phi_m \eta_m}{\varrho_m f_e}, \quad \forall m \in \mathcal{M}_{off}. \quad (34)$$

- **Task Offloading via OMA (TO-OMA):** The IIoT devices offload their tasks to the IGWs via OMA, i.e., each sub-channel is assigned to at most one IIoT device. Different from the binary offloading policy via NOMA in (4), the offloading decision of IIoT device $m$ is denoted by $c_{m,n}$, which depicts whether to fully offload for IGW $n$. When IIoT device $m$ decides to offload the task to IGW $n$, we have $c_{m,n} = 1$; otherwise, we have $c_{m,n} = 0$. In this case, the achievable rate of IIoT device $m$ for task offloading can be rewritten as

$$R_m = \sum_{n=1}^{N} c_{m,n} \sum_{k=1}^{K} \alpha_k n B \log_2 \left(1 + \frac{\left|h_{m,n,k}\right|^2 p_{m,n}}{\sigma_n^2}\right). \quad (35)$$

Targeting on the same objective, we just need to update the achievable rate of offloading decision variable $c_{m,n}$ in (14) instead of using the original constraints in (14d)-(14g).

- **Local Computing Only (LCO):** All IIoT devices execute their tasks via local computing only, i.e., $\alpha_k = 0$ and $\sum_{n=1}^{N} \sum_{j=1}^{J_n} \beta_{n,j} = 0, \forall n \in \mathcal{M}, \forall n \in \mathcal{N}, \forall j \in \mathcal{J}_n$. In this case, the total task completion delay of all IIoT devices can be given in closed-form as

$$\sum_{m=1}^{M} \lambda_m \left(f_m^L\right)^{-1}.$$

A. System Settings

The system settings are set as follows unless stated otherwise. For the considered system, the IGWs and IIoT devices are uniformly distributed on the ground of a smart factory, which is given by a 300 m x 300 m square area with the CAP located in the center. Given such a distribution setup of the IGWs and IIoT devices, the minimum distance between them is set to 2 m. We utilize the Rayleigh fading channel model to denote the channel gain between IIoT device and IGW. To better describe the noisy factory environments, we adopt the Rayleigh fading with a shadowing standard deviation of 10 dB, the Gaussian noise with a noise figure of 9 dB, and the elevated non-line-of-sight path loss (in dB) $140 + 36.7 \log_{10} d [\text{km}]$ [26], where $d$ is the distance between IIoT device and IGW.

Throughout the simulations, each IIoT device has a random size of task-input bits, which follows a uniform distribution $\eta_m \in [2,000,5,000]$ in bits. For most of the industrial applications, the number of required CPU cycles to accomplish the task always correlates with the input bit size. We thus consider $\lambda_m \triangleq \eta_m$ for each IIoT device, where $\phi = 1,000$ cycles/bit [23]. We define the overhead during task offloading as $c_m = 1.1$ times over the input bit size $\eta_m$ for IIoT device. The CPU frequency of IIoT device is uniformly distributed in $f_m^L \in [1,2]$ Gcycles/s, while CPU frequency of MEC server is set to $f_e = 50$ Gcycles/s. For the MEC server, the probability of idle state is set to be $Pr\{\mathcal{H}_0\} = 0.5$. Regarding the DT model, the estimated computation deviation $|\hat{f}_e|$ between the MEC server and DT server is chosen within $[0,10]$ in Gcycles/s. The maximum achievable rate of IIoT device is set as $R_{max} = 2 \times 10^4$ bit/s. We define the subchannel’s bandwidth for task offloading as $B=1$ MHz, and also set the maximum transmit power of IIoT device to be $P_{max} = 24$ dBm.

B. Impact of the DT’s Estimated Computation Deviation

We first examine the performance of our proposed algorithm and analyze the effect of DT usage, in terms of DT’s estimated computation deviation $|\hat{f}_e|$. Fig. 3 shows the total task completion delay versus the estimated computation deviation $|\hat{f}_e|$ used in the DT model for different NOMA cluster sizes $J_n$. From Fig. 2, we can see that as $|\hat{f}_e|$ grows, the total task completion delay is always decreasing for three different values of $J_n$. This is because that with the fixed CPU frequency of the MEC server, the larger the estimated computation deviation is, the higher the estimated CPU frequency is provided by the DT server for its physical counterpart, which causes a significant reduction in task completion delay. Meanwhile, the NOMA cluster size $J_n$ markedly affects the value of total completion delay. A larger NOMA cluster size will lead to lower task completion delay in total. This can be explained by the fact that higher SE can be achieved by larger NOMA cluster size $J_n$ for IGW along with offloading more computation tasks from the IIoT devices. In other words, more tasks can be processed on the MEC server assisted by the DT counterpart, such that the total task completion delay is reduced sharply.

The results of percentage of offloading IIoT devices versus the DT’s estimated computation deviation $|\hat{f}_e|$ for different NOMA cluster sizes $J_n$ are shown in Fig. 4. From the results, we note that the percentage of offloading IIoT devices has a
C. Impact of the Number of IIoT Devices

In Fig. 4 we compare the total task completion delay of our proposed algorithm with three benchmark schemes by varying the number of IIoT devices \( M \). It is evident that the total task completion delay increases for all the schemes with an increase in \( M \). One can observe from Fig. 4 that as \( M \) increases, the total task completion delay of the proposed algorithm as well as the benchmarks of EC-MSO and TO-OMA is significantly lower than that of the LCO scheme. To explain, the LCO scheme forces all IIoT devices to handle their tasks via local computing only, thereby causing higher local processing delay compared to the edge execution. Several observations are also drawn as follows: 1) When \( M \leq 10 \), the proposed algorithm as well as the EC-MSO and TO-OMA schemes exhibit similar performance, this is because each IIoT device can exclusively occupy one subchannel to offload task due to the sufficient subchannels (i.e., \( M \leq K = 10 \)); 2) When \( M > 10 \), the proposed algorithm and the EC-MSO scheme outperform the TO-OMA scheme, and the gap between them becomes larger as \( M \) increases, this is because NOMA allows multiple IIoT devices to share the subchannels with larger \( M \), which results in more tasks offloaded for edge computing. From Fig. 4 we also see that the proposed algorithm shows considerably performance gain compared with the EC-MSO scheme. This observation shows that the DT in our scheme can help the MEC server to improve the offloading decision-making efficiency and further to achieve better computational performance than the MEC without DTs.

Fig. 5 plots the impact of the number of IIoT devices \( M \) on the percentage of offloading IIoT devices for our proposed algorithm and three benchmark schemes. It can be seen that the percentage of offloading IIoT devices is 0% for the LCO scheme, regardless of the number of IIoT devices. This can be explained by the definition of the LCO scheme, which requires all IIoT devices execute their tasks via local computing only. Noteworthy, some IIoT devices may not benefit from the DT-aided edge computing, while some IIoT devices have to handle the tasks locally even when they are more likely to choose task offloading. From Fig. 5 we can also observe that when \( M \leq 5 \), the proposed algorithm as well as the benchmarks of EC-MSO and TO-OMA always achieve the same performance in terms of percentage of offloading IIoT devices. These results imply that when there are fewer IIoT devices, they all choose to offload their tasks for edge execution due to superior computation capacity of the MEC server. However, when \( M > 5 \), the percentage of offloading IIoT devices of the proposed algorithm is greater than that of the benchmarks of EC-MSO and TO-OMA, and the gap between them becomes larger as \( M \) increases. Not surprisingly, our proposed algorithm allows

<table>
<thead>
<tr>
<th>Percentage of offloading IIoT devices</th>
<th>DT's estimated computation deviation (CPU cycles/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed algorithm, ( J_n=1 )</td>
<td>Proposed algorithm, ( J_n=2 )</td>
</tr>
<tr>
<td>Proposed algorithm, ( J_n=3 )</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Total task completion delay versus the DT’s estimated computation deviation, when \( M = 25 \), \( N = 15 \), \( K = 15 \), and \( F = 1 \).

Fig. 3. Percentage of offloading IIoT devices versus the DT’s estimated computation deviation, when \( M = 25 \), \( N = 15 \), \( K = 15 \), and \( F = 1 \).

Fig. 4. Total task completion delay versus the number of IIoT devices, when \( N = 10 \), \( K = 10 \), \( |f_e| = 10 \) cycles/s, and \( F = 0.95 \).
more IIoT devices to offload their tasks for DT-aided edge computing, benefiting from the enhanced offloading decision-making efficiency and reduced task completion delay.

D. Impact of the Number of Subchannels

In Fig. 6, we evaluate the total task completion delay versus the number of subchannels $K$ for our proposed algorithm and three benchmark schemes. The figure shows that the total task completion delay for the LCO scheme maintains a constant value, nearly $33\,\text{s}$, regardless of the number of subchannels. This observation confirms the definition of LCO scheme. It is clear from Fig. 6, the proposed algorithm still outperforms the benchmark schemes of EC-MSO and TO-OMA, for different $K$, and the gap between them has a decreasing trend as $K$ increases. This is because as $K$ grows, more spectrum resource blocks available means that the SE for them becomes more closer, thereby reducing the delay gap. Another observation can be drawn as follows: when $K < M = 25$, the total task completion delay of the TO-OMA scheme shows an obviously decreasing trend compared with that of the proposed algorithm and the EC-MSO scheme. The reason behind this is that all the subchannels are occupied for given $M = 25$, when $K$ is smaller, such that more tasks are processed locally, leading to higher task completion delay. We can also see that the delay gap between the proposed algorithm and the EC-MSO scheme keeps in a steady value for different $K$. To explain, NOMA allows one subchannel to be shared by multiple IIoT devices simultaneously, so that there is no significant change in the number of offloading IIoT devices as $K$ increases. In addition, the total task completion delay for the proposed algorithm and the EC-MSO scheme with $B = 1.2\,\text{MHz}$ is observed to be always larger than that of $B = 1\,\text{MHz}$. Such results bolster the importance of choosing an appropriate subchannel’s bandwidth for task offloading to reduce the task completion delay.

E. Impact of the Size of Task-input Bits

Fig. 7 illustrates the comparison of the total task completion delay between our proposed algorithm and three benchmark schemes by varying the size of task-input bits $\eta_m$ of IIoT device. We observe that the total task completion delay increases with $\eta_m$. From Fig. 7, it can be also seen that the total task completion delay with respect to the LCO scheme increases markedly compared with the proposed algorithm as well as the benchmark schemes of EC-MSO and TO-OMA, as $\eta_m$ goes up. This result indicates that, to achieve better computation performance, more tasks should be offloaded to the MEC server for edge processing. Besides, one can easily see that both the proposed algorithm and the EC-MSO scheme have lower task completion delay in total than the TO-OMA scheme, and this verifies our analysis that NOMA allows multiple IIoT devices to share the subchannels, thus achieving higher offloading rates of IIoT devices. From the figure, we also find that with $\eta_m$ increasing, the delay performance of the proposed algorithm outperforms the benchmarks, which shows that our proposed scheme has superiority by enhancing the computing performance via DT. A final important observation is that the performance of the proposed algorithm and the TO-OMA scheme with $\xi_0 = 0.6$ outperforms that of $\xi_0 = 0.5$ in terms of the total task completion delay. To explain, the larger the transition rate $\xi_0$ from idle to busy of the MEC server is, the greater the computing performance is shown by the DT. Therefore, we can conclude that the computation performance presents the considerable gains brought by the DT with higher transition rate.

VI. CONCLUSION

In this paper, we investigated the problem of joint edge association and resource allocation in the DT-assisted MEC system with NOMA for IIoT. Under the constraints of computing services of all IIoT devices, we aimed at minimizing the total task...
completion delay among them by jointly optimizing the IGW’s subchannel assignment as well as the computation capacity allocation, edge association, and power control of IIoT device. Since the resulted problem was a mixed integer non-convex optimization problem and NP-hard, we decoupled it into four sub-problems, namely power optimization, subchannel assignment optimization, computation capacity optimization, and edge association optimization. An efficient iterative algorithm was then designed to find a convergent solution of this problem by alternately optimizing the sub-problems. The convergence and complexity of the proposed algorithm were also analyzed theoretically. As results, we reveal that the proposed scheme achieves considerable performance gains for the total task completion delay of all IIoT devices compared to the other benchmarks. Moreover, two useful insights can be drawn from simulation results: 1) Our proposed scheme significantly achieves lower task completion delay when the DT’s estimated computation deviation and the NOMA cluster size are large enough; 2) Choosing higher estimated computation deviation of DT at larger NOMA cluster size and increasing the number of subchannels at given number of IIoT devices are two effective means to reduce the total task completion delay.

**APPENDIX**

**A. Proof of Theorem 2**

Making use of (35), we have

\[ R_m = B \log_2 \left( 1 + \frac{|h_{m,n}^k| p_{m,n}^k}{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right) \]

\[ = B \log_2 \left( \frac{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k}{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right) \]

\[ = B \log_2 \left( \frac{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k}{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right) \]

\[ - B \log_2 \left( \frac{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k}{\sigma_n^2 + \sum_{j=1}^{J_n} |h_{j,n}^k|^2 p_{j,n}^k} \right). \]  (36)

From (36), after some algebra, we obtain (37), as shown at the bottom of next page. It can be observed from (37) that the total achievable rate increases with \( p_{m,n}^k \). To maximize the total achievable rate in (37), we set the optimal power value as \( p_{m,n}^k = P_{m}^{\text{max}} \). Substituting \( P_{m}^{\text{max}} \) into (37) yields the result.

**B. Proof of Proposition 7**

Since constraints (14i) and (14j) in (23) are convex, we thus focus on examining the convexity of objective function (23a). The second-order derivative of objective function (23a) with respect to \( q_m \) can be obtained by

\[ \frac{\partial^2 U(q_m)}{\partial q_m^2} = \frac{2 \phi_{m,\eta_m} (\tilde{f}_e + \xi_1 \tilde{f}_e)}{q_m^2 \tilde{f}_e (\tilde{f}_e + \tilde{f}_e)} > 0. \]  (38)

where \( U(q_m) \triangleq \phi_{m,\eta_m} \frac{q_m (\tilde{f}_e + \tilde{f}_e)}{q_m (\tilde{f}_e + \tilde{f}_e)} + \xi_1 (q_m \tilde{f}_e (\tilde{f}_e + \tilde{f}_e)). \) The nonnegativity of (38) is guaranteed, which indicates objective function (23a) is convex. Therefore, problem (23) is strictly convex, and also has unique solution. Thus, we conclude the proof.

**C. Proof of Theorem 3**

The Lagrangian function of (23) can be given by

\[ \mathcal{L}(Q, \nu, \vartheta) = \sum_{m \in \mathcal{M}_{off}} \left( \frac{\phi_{m,\eta_m} (\tilde{f}_e + \xi_1 \tilde{f}_e)}{q_m (\tilde{f}_e + \tilde{f}_e)} + \frac{c_0 + \xi_1 q_m \tilde{f}_e (\tilde{f}_e + \tilde{f}_e)}{q_m (\tilde{f}_e + \tilde{f}_e)} \right) - \sum_{m \in \mathcal{M}_{off}} \nu_m q_m + \vartheta \left( \sum_{m \in \mathcal{M}_{off}} q_m \tilde{f}_e - F_{m}^{\text{max}} \right) \]  (39)

where \( \nu \in \{\nu_m, \forall m\} \) and \( \vartheta \) are the Lagrange multipliers associated with constraints (14i) and (14j), respectively.

Problem (23) satisfies the Slater’s condition, such that the strong duality holds for this problem if the duality gap is zero. Let \( q_m^\star \) be the optimal solution to (23). The Karush-Kuhn-Tucker (KKT) conditions can be specifically derived as

\[ \frac{\partial \mathcal{L}(\cdots)}{\partial q_m^\star} = \frac{- \phi_{m,\eta_m} (\tilde{f}_e + \xi_1 \tilde{f}_e)}{q_m^\star (\tilde{f}_e + \tilde{f}_e)} - \nu_m + \vartheta = 0. \]  (40)

\[ \nu_m q_m^\star = 0. \]  (41)

\[ \vartheta \left( \sum_{m \in \mathcal{M}_{off}} q_m^\star \tilde{f}_e - F_{m}^{\text{max}} \right) = 0. \]  (42)

Next, using (41), we have \( \nu_m = 0 \). Combining \( \nu_m = 0 \) and (40), we note that

\[ \frac{\tilde{f}_e + \xi_1 \tilde{f}_e}{\tilde{f}_e + \tilde{f}_e} (\tilde{f}_e + \tilde{f}_e) = \frac{\phi_{m,\eta_m}}{q_m^\star (\tilde{f}_e + \tilde{f}_e)}. \]  (43)

By multiplying both sides of (42) by \( \frac{\phi_{m,\eta_m}}{q_m^\star} \), we then derive the square root as follows

\[ \sqrt{\frac{\phi_{m,\eta_m}}{q_m^\star}} \left( \sum_{m \in \mathcal{M}_{off}} q_m^\star \tilde{f}_e - F_{m}^{\text{max}} \right) = 0. \]  (44)

After some algebra, it follows directly from (44) that the optimal solution \( q_m^\star \) to (23) can be denoted in closed-form as

\[ q_m^\star = \frac{\tilde{f}_e + F_{m}^{\text{max}} \sqrt{\phi_{m,\eta_m}}}{\sum_{m \in \mathcal{M}_{off}} \sqrt{\phi_{m,\eta_m}}}. \]  (45)

By letting \( F = \tilde{f}_e + F_{m}^{\text{max}} \), we further derive \( q_m^\star \) as shown in (24) and this concludes the proof.

**REFERENCES**


\[
\sum_{m=1}^{J} R_m = B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=2}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right) - B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=2}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=3}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right) + \cdots + B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=J-1}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=J}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right) - B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=J}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=J+1}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right)
\]

\[
= B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right)
\]

\[
= B \log_2 \left( \frac{\sigma_n^2 + \sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2 + \sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k} \right)
\]

\[
= B \log_2 \left( 1 + \frac{\sum_{m=1}^{J} |h_{m,n}^k|^2 p_{m,n}^k}{\sigma_n^2} \right).
\]