Spectrum Allocation and Device Association in Federated Learning-Enabled Industrial IoT via Hypergraph Matching

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

29-07-2021 / 02-08-2021

CITATION

Wu, Suiyuan; Zhang, Long; Wang, Yao; Han, Zhu (2021): Spectrum Allocation and Device Association in Federated Learning-Enabled Industrial IoT via Hypergraph Matching. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.15074076.v1>

DOI

[10.36227/techrxiv.15074076.v1](https://doi.org/10.36227/techrxiv.15074076.v1)

Spectrum Allocation and Device Association in Federated Learning-Enabled Industrial IoT via Hypergraph Matching

Suiyuan Wu, Long Zhang, *Member, IEEE*, Yao Wang, and Zhu Han, *Fellow, IEEE*

Abstract—In this paper, a joint spectrum allocation and device association problem is investigated for a federated learning aided hierarchical Industrial Internet of Things (IIoT) system for smart factory. To achieve the optimization jointly, we design a weighted learning utility maximization problem, which is a 0-1 integer linear programming problem. To solve this problem, we convert it into a weighted 3D hypergraph model by capturing the 3D mapping relation for IIoT device, subchannel, and edge server. A local search algorithm is then presented to find a 3D hypergraph matching with maximum total weights as the suboptimal solution. Simulation results demonstrate the superior performance of the proposed algorithm compared with the greedy algorithm in the system learning utility.

I. INTRODUCTION

Recently, the paradigm of industrial Internet of things (IIoT) has been considered as one of the most promising approaches capable of improving the production efficiency and economic benefits, thus attracting significant attention from academia, manufacturers, and plant designers [1]. With this paradigm, a large number of IIoT devices (e.g., sensors, robots, actuators, machines, etc.) in smart factories are connected and synchronized through the combination of enabling technologies such as sensing, communications, computing, caching, machine learning, big data, analytics, and so on [2]–[4]. Thus, IIoT can monitor, collect, exchange, analyze, and transmit data directly to drive the unprecedented levels of efficiency, productivity, and performance in industrial operations.

The generated data in IIoT is typically transmitted to either edge nodes for time-critical applications, or the central cloud for further processing and inferencing. With the exponential growth in the number of IIoT devices, a huge volume of data is generated in IIoT system. In this case, both volume of the data and its associated performance needs to be well analyzed to accommodate for the IIoT applications. The requirement thus motivates us to configure, manage, and control the data in a smarter and more agile manner. Then, real-time and flexible decision making in IIoT can be achieved effectively. Therefore, machine learning (ML) approaches offer the promising solutions to analyze huge amounts of data [2], [5]. However, due to constrained communication resources and privacy concerns, it is impractical for all the IIoT devices engaged in learning

to send all of their sensing data to a remote cloud. To address the challenges, as a distributed learning framework, federated learning (FL) has been developed to allow the IIoT devices to cooperatively train an ML model locally without sharing their data to the server [6].

Based on the above motivations, we find that the exploration of FL for IIoT has become highly valuable. Several works are devoted to FL-enabled IIoT system. In [7], the authors studied the resource allocation problem for FL-enabled IIoT, and formulated a training latency minimization problem by jointly considering the active device selection and resource allocation. In [8], the authors explored the FL cost minimization problem in smart industries by capturing both the latency and packet error rate of FL. A communication-efficient FL framework in IIoT system was proposed in [9] for anomaly detection of IIoT devices. Moreover, the authors in [10] also developed a communication-efficient FL scheme in digital twin aided IIoT system. However, due to the power constraint of IIoT devices, the tradeoff between learning performance and energy consumption is significant to the overall system, which has not been fully addressed in literature. Therefore, in this paper, we consider a FL-aided hierarchical IIoT system for smart factory, and particularly, investigate a joint spectrum allocation and device association problem to maximize the weighted learning utility of the system. Main contributions of our work include:

- We formulate the learning utility maximization problem as a 0-1 integer linear programming problem by jointly optimizing binary variables for subchannel allocation and IIoT device association.
- By capturing a 3D mapping relation for IIoT device, subchannel, and edge server, this problem is transformed into a weighted 3D hypergraph model, wherein the hyperedge weight is modeled as the IIoT device's learning utility.
- We convert the problem to find a 3D hypergraph matching with maximum total weights, and propose a local search (LS) based algorithm to obtain a suboptimal solution.

The rest of the paper is organized as follows. We introduce the system model and formulate the optimization problem in Section II. Section III presents the problem transformation and algorithm design. Simulation results are provided in Section IV. Finally, Section V concludes this paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Overview

As shown in Fig. 1, we consider a FL-enabled hierarchical IIoT system for smart factory, which consists of three layers with different functionalities and resources, i.e., end, edge, and cloud layers. In the end layer, N IIoT devices, denoted by a set

This work was supported in part by NSFC (61901115), Hebei Program for Top-notch Young Talents (BJ2017037), and NSF of Hebei (F2019402206).

S. Wu, L. Zhang, and Y. Wang are with the School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China.

Z. Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea.

Corresponding author: Long Zhang (email: zhanglong@hebeu.edu.cn).

Fig. 1. An example of the hierarchical IIoT system for smart factory.

$\mathcal{N} = \{1, \dots, N\}$, are randomly deployed on the ground to collect real-time data (i.e., sensing data) for ubiquitous situation awareness. In the edge layer, M edge servers, denoted by a set $\mathcal{M} = \{1, \dots, M\}$, are uniformly attached on the room ceiling of smart factory. The edge server not only serves as an access point to provide wireless access for IIoT devices via uplink, but also sends the URLLC control data to the associated IIoT devices via downlink. As the core of the entire IIoT system, the cloud server is located at the core network in the cloud layer, to provide high-performance intelligent services and resources, e.g., computing, caching, security, etc. We focus on two kinds of data utilized in the IIoT system, i.e.:

- *Sensing data* generated from the IIoT device has higher data volume but relatively smaller time sensitivity. With sensing data, IIoT devices cooperatively perform the FL for big data analysis in predictive maintenance.
- *Control data* sent from the edge server has smaller data capacity with more time sensitivity. IIoT devices receive control data from edge servers for factory automation.

B. Transmission Model

1) *Sensing Data Transmission via Uplink*: For uplink, N IIoT devices in the end layer send their sensing data to M edge servers in the edge layer over the available spectrum resource block. For simplicity, the orthogonal frequency division multiple access is adopted for scheduling the simultaneous access of IIoT devices. The interference between subchannels during the uplink transmission can be accordingly ignored. The available resource block is equally divided to K orthogonal subchannels, denoted by a set $\mathcal{K} = \{1, \dots, K\}$. Let B be the bandwidth of each subchannel. We assume the channel state information (CSI) is perfectly known at each edge server. For convenience, let us define a binary variable as follows to show the subchannel assignment constraint for IIoT device n :

$$a_{n,k} = \begin{cases} 1, & \text{if subchannel } k \text{ is assigned to IIoT device } n, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that each subchannel is assigned to only one IIoT device and each IIoT device can occupy at most one subchannel. So we have $\sum_{n=1}^N a_{n,k} \leq 1, \forall k$ and $\sum_{k=1}^K a_{n,k} \leq 1, \forall n$. Assume that one IIoT device can associate at most one edge server, and edge server m can serve at most N_m IIoT devices due to the hardware limitation. In order to represent the association

relationship between IIoT device n and edge server m , a binary variable is also introduced, which can be defined by:

$$b_{n,m} = \begin{cases} 1, & \text{if device } n \text{ is associated with edge server } m, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Note that $\sum_{m=1}^M b_{n,m} \leq 1, \forall n$, and $\sum_{n=1}^N b_{n,m} \leq N_m, \forall m$. Then, the uplink achievable rate between IIoT device n and edge server m on subchannel k can be obtained as:

$$R_{n,m}^k = \sum_{m=1}^M b_{n,m} \sum_{k=1}^K a_{n,k} B \log_2 \left(1 + \frac{P_{n,m} h_{n,m}}{\sigma_m^2} \right), \quad (3)$$

where $P_{n,m}$ is the transmit power of IIoT device n to edge server m , $h_{n,m}$ is the channel gain from IIoT device n to edge server m , and σ_m^2 is the variance of AWGN at edge server m . As in [3], $h_{n,m} = 10^{-PL_{n,m}[\text{dB}]/10}$, where $PL_{n,m}[\text{dB}] = A \log_{10}(d_{n,m}) + B + E \log_{10}(f_c/5) + X$, where $d_{n,m}$ is the distance between IIoT device n and edge server m , f_c is the carrier frequency, and A, B, E are the constants depending on the propagation model [3].

2) *Control Data Transmission via Downlink*: Due to higher transmit power of edge server and smaller data size of control data, we neglect the time consumed for sending the control data to IIoT devices via downlink. In this paper, we only consider the reliability requirement for control data transmission. The received SNR at IIoT device n from edge server m is given as $\gamma_{m,n} = P_m h_{m,n} / \sigma_n^2$, where P_m is the transmit power of edge server m , σ_n^2 is the variance of AWGN at IIoT device n , $h_{m,n}$ is the channel gain from edge server m to IIoT device n , which is equal to $h_{n,m}$ due to the channel reciprocity. To ensure the reliable transmission of control data, we have:

$$\gamma_{m,n} \geq b_{n,m} \gamma_{\min}, \quad \forall n, m, \quad (4)$$

where $\gamma_{\min}$ is the minimum SNR threshold.

C. FL Model

1) *Local Training*: Each IIoT device n owns a local dataset $D_n = \{(\mathbf{x}_j, y_j)\}$, where $\mathbf{x}_j$ is j -th input vector and y_j is the corresponding labeled output of $\mathbf{x}_j$ for the FL task of IIoT device n , for $j = 1, \dots, |D_n|$. Note that IIoT device n needs to train a machine learning model ω_n in an iterative manner until a predefined local accuracy θ . The model ω_n captures the performance of learning task with loss function defined as:

$$F_n(\omega) = \frac{1}{|D_n|} \sum_{j=1}^{|D_n|} f_n(\mathbf{x}_j, y_j, \omega_n), \quad (5)$$

where $f_n(\mathbf{x}_j, y_j, \omega_n)$ is the loss function depending on the specific learning task. Moreover, the local iteration number is given by $L(\theta) = \mu \log(\frac{1}{\theta})$, where μ is a constant associated with the data size and learning task. In each iteration, IIoT device n needs to train its model according to the gradient descent algorithm which is given by $\omega_n(t) = \omega_n(t-1) - \eta \nabla F_n(\omega_n(t-1))$, where η is the learning rate. Therefore, we can derive the computation time of IIoT device n as:

$$t_n^{\text{cmp}} = \frac{L(\theta) Z(|D_n|) c_n}{f_n}, \quad (6)$$

where $Z(x)$ is the data size of x in bit, c_n is the number of CPU cycles required for computing one bit data, and f_n is the CPU frequency of IIoT device n . We assume that the edge server has perfect knowledge of computing capacity and energy state of IIoT devices. Therefore, the computation energy consumption of IIoT device n can be given by [11]:

$$e_n^{\text{cmp}} = L(\theta) c_n Z(|D_n|) \zeta f_n^2, \quad (7)$$

where ζ is the energy consumption coefficient depending on the chip architecture of IIoT device.

2) *Model Transmission*: After the training of ω_n , IIoT device n will then transmit the model to its associated edge server, which will generate the delay and energy consumption. The delay and energy consumption of IIoT device n for model transmission can be respectively described as:

$$t_{n,m}^{\text{com}} = \frac{Z(\omega_n)}{R_{n,m}}, \quad (8)$$

$$e_{n,m}^{\text{com}} = P_{n,m} t_{n,m}^{\text{com}}. \quad (9)$$

Note that the unreliable IIoT channel condition has a significant effect on the model transmission, e.g., packet errors. When the received model contains the errors, edge server m will not use that model for aggregation. More sub-global round is needed to get convergence, resulting in more delay and energy consumption. Specifically, the packet error rate between IIoT device n and edge server m is determined by:

$$E_{n,m}^{\text{error}} = a_{n,k} b_{n,m} \left(1 - \exp \left(\frac{-\lambda (I_{C_n} + \sigma_m^2)}{P_{n,m} h_{n,m}} \right) \right), \quad (10)$$

where λ is the waterfall threshold.

3) *Model Aggregation*: After receiving the local models from IIoT devices, edge server m performs the aggregations and generates an updated sub-global model ω_m according to $\omega_m = \sum_{n=1}^N b_{n,m} \omega_n |D_n| / \sum_{n=1}^N b_{n,m} |D_n|$. Then edge server m broadcasts ω_m to every associated IIoT device for the next sub-global round training. To achieve a sub-global accuracy ε , edge servers should perform $I(\varepsilon, \theta)$ times iteration training. As in [12], the iteration number is thus specified by:

$$I(\varepsilon, \theta) = \frac{\phi(\log(1/\varepsilon))}{1 - \theta}, \quad (11)$$

where ϕ is a constant related to learning task. Since edge server has much higher transmit power, we ignore the down-link communication in the FL model. Thus the total energy consumption of IIoT device n in one global round is given as:

$$e_{n,m} = I(\varepsilon, \theta) \sum_{m=1}^M b_{n,m} (e_n^{\text{cmp}} + e_{n,m}^{\text{com}}). \quad (12)$$

After $I(\varepsilon, \theta)$ times iteration training of sub-global model, edge server m then sends ω_m to the cloud server in the cloud layer for global model aggregation and obtain global model g . For simplicity, we ignore the delay and energy consumption in the edge and cloud servers due to their powerful transmission

and computing capabilities. The total delay of IIoT device n takes for the FL in each sub-global round is written by:

$$t_{n,m} = \sum_{m=1}^M b_{n,m} (t_n^{\text{cmp}} + t_{n,m}^{\text{com}}). \quad (13)$$

Due to the synchronous of the FL, the model aggregation depends on the last user which finishes the model training and transmission. Thus, each IIoT device must meet the delay constraint, given by $t_{n,m} \leq \Delta T$, $\forall n$, where $\Delta T > 0$ is a fixed value. Note that the increasing number of the participants in the FL will significantly improve the learning performance. As a result, we employ the involved number of the IIoT devices to characterize the learning speed of the FL as follows:

$$L = \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K a_{n,k} b_{n,m}. \quad (14)$$

D. Problem Formulation

By applying the FL for the sensing data flow, more involved IIoT devices contributes to the improved learning performance of the FL, but resulting in the increased energy consumption of the IIoT devices. Moreover, the proper selection of the IIoT devices also impacts the packet error rate between IIoT devices and edge servers. Keeping this fact in mind, both the energy consumption and packet error rate are critical to the IIoT devices during the FL procedure. Our objective is thus to identify the balance among the learning performance of the FL, the energy consumption and packet error rate relevant to the IIoT devices by optimizing binary variables for subchannel allocation and IIoT device association in the IIoT system.

To capture such a balance, we design a weighted learning utility function of the system, which is specifically defined as:

$$U_{\text{FL}}(a_{n,k}, b_{n,m}) = L - \alpha_1 E - \alpha_2 E^{\text{error}}, \quad (15)$$

where E is total energy consumption in one global round, E^{error} is the total packet error rate, and α_1 and α_2 are the weighting factors of energy consumption and packet error rate associated with the IIoT devices, respectively. Particularly, the total energy consumption in one global round is given by:

$$E = \sum_{n=1}^N e_{n,m} = \sum_{n=1}^N \sum_{m=1}^M I(\varepsilon, \theta) b_{n,m} (e_n^{\text{cmp}} + e_{n,m}^{\text{com}}). \quad (16)$$

Meanwhile, the total packet error rate can be calculated as:

$$E^{\text{error}} = \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K E_{n,m}^{\text{error}}. \quad (17)$$

Under the above setup, we aim to maximize the weighted learning utility function of the system by jointly optimizing binary variables for subchannel allocation and IIoT device association. This optimization problem is formulated as:

$$\max_{a_{n,k}, b_{n,m}} U_{\text{FL}}(a_{n,k}, b_{n,m}) \quad (18)$$

$$\text{s.t. C1: } \sum_{n=1}^N a_{n,k} \leq 1, \forall k, \quad \text{C2: } \sum_{k=1}^K a_{n,k} \leq 1, \forall n,$$

$$\begin{aligned}
\text{C3} : \sum_{m=1}^M b_{n,m} &\leq 1, \forall n, \quad \text{C4} : \sum_{n=1}^N b_{n,m} \leq N_m, \forall m, \\
\text{C5} : \gamma_{m,n} &\geq b_{n,m} \gamma_{\min}, \forall n, m, \quad \text{C6} : t_{n,m} \leq \Delta T, \forall n, \\
\text{C7} : a_{n,k}, b_{n,m} &\in \{0, 1\}, \forall n, k, m.
\end{aligned}$$

In (18), C1 and C2 reflect the binary constraints of subchannel allocation for each IIoT device. C3 shows that each IIoT device can associate at most one edge server. C4 is provided to indicate the maximum association constraint of each edge server. C5 is the reliable transmission requirement of control data flow. C6 is used for the delay constraint of each IIoT device in one sub-global round. C7 dictates the binary constraints for subchannel allocation variable and device association variable.

III. ALGORITHM DESIGN VIA HYPERGRAPH MATCHING

The problem in (18) is a pure 0-1 integer linear programming problem due to the binary variables involved in objective function and the linear constraints C1-C6. Therefore, problem (18) is NP-hard and computationally intractable. Especially, with the increasing scale of the IIoT system wherein N, M, K become very large, it is difficult to optimally solve the problem directly with low computational complexity via the traditional exhaustive method. However, we observe that there exists an inherent coupling constraint for subchannel allocation and device association in problem (18), which typically falls into a one-to-one-to-one (3D) mapping relation for IIoT device, subchannel, and edge server. To make it more tractable, we thus transform problem (18) into a weighted 3-uniform hypergraph model by jointly capturing such coupling constraints and the system-level weighted learning utility function.

To identify such a mapping relation, we develop the virtual elements for edge servers in the edge layer. Specifically, we devise $N_m - 1$ virtual edge servers for edge server m to satisfy the binary variable requirement of device association due to the constraints C3 and C4, for $N_m > N$. Thus, there are MN_m edge servers and virtual edge servers in the edge layer to associate the IIoT devices in the end layer. Note that the conventional graph used to describe the pairwise relation between objects is not applicable for the 3D relation among objects. Therefore, we apply the hypergraph model in which every hyperedge can be formulated as a subset of the vertex set to characterize the 3D mapping among multiple objects.

A. Hypergraph Construction

Definition 1. A weighted hypergraph $\mathcal{H}$ is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \varpi_l(\mathcal{E}_l))$, where $\mathcal{V}$ is a finite set of vertices, $\mathcal{E}$ is a family of non-empty subsets of $\mathcal{V}$ (i.e., hyperedge set), and $\varpi_l(\mathcal{E}_l)$ is a positive real number associated with hyperedge $\mathcal{E}_l \in \mathcal{E}$, called the weight of hyperedge $\mathcal{E}_l$, for $l = 1, 2, \dots, |\mathcal{E}|$.

For ease of exposition, let us utilize $\mathcal{Q}$ to represent the set of MN_m edge servers and virtual edge servers in the edge layer. Combining the sets of IIoT devices and subchannels, an element in $\mathcal{N}, \mathcal{K}$, and $\mathcal{Q}$ stands for a vertex in weighted hypergraph $\mathcal{H}$. Thus, the union of $\mathcal{N}, \mathcal{K}$, and $\mathcal{Q}$ refers to the vertex set $\mathcal{V}$, i.e., $\mathcal{V} = \mathcal{N} \cup \mathcal{K} \cup \mathcal{Q}$. With this in mind, we

employ the combination $\{n, k, q\}$ to characterize a hyperedge $\mathcal{E}_l$ in $\mathcal{H}$, subject to constraints C1-C6 simultaneously, i.e., $\mathcal{E}_l = \{n, k, q\}$, for $n \in \mathcal{N}, k \in \mathcal{K}, q \in \mathcal{Q}, l = 1, 2, \dots, |\mathcal{E}|$. Note that such a combination $\{n, k, q\}$ obviously captures the 3D mapping relation for IIoT device, subchannel, and edge server in the considered IIoT system. Additionally, the weight of hyperedge $\mathcal{E}_l$ is defined as the learning utility function of IIoT device n in one global round, i.e.:

$$\varpi_l(\mathcal{E}_l) = (a_{n,k}b_{n,q} - \alpha_1 e_{n,q} - \alpha_2 E_{n,q}^{\text{error}}), \forall n, k, q. \quad (19)$$

B. Algorithm Design

Based on the weighted hypergraph $\mathcal{H}$, our target for the weighted learning utility maximization problem is converted to find a 3D hypergraph matching that is a collection of N vertex-disjoint hyperedges with the maximum total weights. Unfortunately, it is NP-hard to find a weight-maximum subset of vertex-disjoint hyperedges in the 3D hypergraph matching. Inspired by the LS strategy with an increasing approximation ratio in polynomial time [13], we thereby use the LS to design an efficient iterative algorithm for obtaining a suboptimal solution to problem (18).

Definition 2. A representative graph $\mathcal{G}$ is an ordinary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where a vertex $v_l \in \mathcal{V}$ in $\mathcal{G}$ denotes a hyperedge $\mathcal{E}_l \in \mathcal{E}$ of $\mathcal{H}$, and an ordinary edge in $\mathcal{E}$ denotes the adjacent hyperedges in $\mathcal{H}$ if they intersect with each other at least one vertex of $\mathcal{H}$. The weight of vertex $v_l \in \mathcal{V}$ in $\mathcal{G}$ equals the weight of hyperedge $\mathcal{E}_l$ of $\mathcal{H}$, i.e., $\varpi_l(\mathcal{E}_l)$.

Definition 3. For a representative graph $\mathcal{G}$, a ϵ -claw is an induced subgraph $\mathcal{G}_\epsilon$, whose center vertex connects to ϵ independent vertices, called talons, which forms an independent set $\mathcal{I}_{\mathcal{G}_\epsilon}$ with ϵ vertices.

Let $\Phi(\mathcal{G})$ and $\varpi(\Phi)$ be an initial independent set of $\mathcal{G}$ and the sum of weights for all the vertices in $\Phi(\mathcal{G})$, respectively. Note that an initial independent set refers to an initial matching in original hypergraph $\mathcal{H}$. To obtain an initial independent set $\Phi(\mathcal{G})$, we first use a greedy algorithm to approximately derive an weight-maximum initial subset of edge-disjoint vertices in $\mathcal{G}^1$. Then we propose an LS-based algorithm to search for ϵ -claw iteratively, aiming to improve the sum of weights of initial matching. Through the iterative searching, we thus obtain the weight-maximum 3D hypergraph matching result, including N vertex-disjoint hyperedges.

For tractability, let us use $\psi(\mathcal{I}_{\mathcal{G}_\epsilon}, \Phi(\mathcal{G}))$ to denote the set of adjacent vertices of independent set $\mathcal{I}_{\mathcal{G}_\epsilon}$ with ϵ vertices in $\Phi(\mathcal{G})$. The procedure of the proposed LS-based algorithm is summarized in Algorithm 1. Based on Algorithm 1, we obtain a weight-maximum subset $\mathcal{E}^*$ which has N vertex-disjoint hyperedges. Thereby, the suboptimal solution to the weighted learning utility maximization problem is thus achieved, and the utility function can be given by $U_{\text{FL}}(a_{n,k}, b_{n,m}) = \varpi(\Phi)$.

IV. SIMULATION RESULTS

In this section, we validate the learning performance of the proposed LS-based algorithm with simulations. We consider

¹Due to the space limitation, specific detail about the greedy algorithm is omitted here, and readers can refer to [13] for more details.

Algorithm 1 LS-based Algorithm for Hypergraph Matching

```

1: Transform the initial hypergraph  $\mathcal{H}$  into a representative graph  $\mathcal{G}$ .
2: Using greedy algorithm in [13] to obtain an initial independent set  $\Phi(\mathcal{G})$ .
3: repeat
4:   Sort all vertices of  $\Phi(\mathcal{G})$  in an ascending order with vertex weight.
   Set  $l = 1$ .
5:   repeat
6:     Find all the adjacent vertices of  $l$ -th vertex  $\Phi[l] \in \Phi(\mathcal{G})$ , and let
      $\psi_{\Phi[l]}$  be the set of adjacent vertices of  $\Phi[l]$ .
7:     Sort all the vertices of  $\psi_{\Phi[l]}$  in a descending order with vertex
     weight.
8:     Set  $l$ -th vertex  $\Phi[l]$  as the center vertex of  $\epsilon$ -claw  $\mathcal{G}_\epsilon$ .
9:     for  $\epsilon = 1 \rightarrow 3$  do
10:      Search for  $\epsilon$ -claw  $\mathcal{G}_\epsilon$  from  $\psi_{\Phi[l]}$  in  $\mathcal{G}$ .
11:      if  $\varpi((\Phi(\mathcal{G}) - \psi(\mathcal{I}_{\mathcal{G}_\epsilon}, \Phi(\mathcal{G}))) \cup \mathcal{I}_{\mathcal{G}_\epsilon}) > \varpi(\Phi)$  then
12:         $\Phi(\mathcal{G}) = (\Phi(\mathcal{G}) - \psi(\mathcal{I}_{\mathcal{G}_\epsilon}, \Phi(\mathcal{G}))) \cup \mathcal{I}_{\mathcal{G}_\epsilon}$ , go to step 3.
13:      end if
14:    end for
15:     $l = l + 1$ .
16:   until  $l \geq |\Phi(\mathcal{G})|$ .
17: until  $|\Phi(\mathcal{G})| = N$ .
18: return  $\mathcal{E}^* = \{\mathcal{E}_1^*, \mathcal{E}_2^*, \dots, \mathcal{E}_N^*\}$ .

```

a hierarchical IIoT system, consisting of $N=50$ IIoT devices randomly deployed on the ground and $M=5$ edge servers uniformly attached on the room ceiling with a height of 10m. We set the parameters in propagation model $A=18.7$, $B=46.8$, and $E=20$. Unless other specified, some default parameters are set as $K=40$, $B=180\text{kHz}$, $N_m=12$, $\sigma_n^2=-20\text{dBm}$, $\sigma_m^2=-20\text{dBm}$, $\gamma_{\min}=0.066$, $P_{n,m}=23\text{dBm}$, $P_m=43\text{dBm}$, $Z(\omega)=2.5\text{KB}$, $Z(|D_n|)=30\text{KB}$, $\theta=0.9$, $c_n=40$, $\zeta=10^{-27}$, $\lambda=0.023\text{dB}$, $\Delta T=500\text{ms}$, and $\varepsilon=0.9$. The CPU frequency f_n ranges in the interval $[10, 20]\text{MHz}$.

Fig. 2 compares the learning utility performance between greedy algorithm and our proposed algorithm under the varying number of IIoT devices. We can observe that our proposed algorithm provides a higher utility than that of greedy algorithm. The reason is that by searching for ϵ -claw repeatedly, the proposed algorithm achieves a sub-optimal solution, which has better performance than greedy algorithm. When $N \leq 40$, as the increasing number of IIoT devices, the utility increases rapidly for both algorithms. That is because more involved devices significantly improve the learning performance. When $N > 40$, the learning utility is growing slowly. This result further provides a hint to choose appropriate number of IIoT devices to further improve the system performance.

V. CONCLUSIONS

This paper studied the joint spectrum allocation and device association problem in FL-enabled IIoT system. We formulated the weighted learning utility maximization problem as the 0-1 integer linear programming problem. Based on the 3D mapping relation for IIoT device, subchannel, and edge server, this problem was transformed into the weighted hypergraph model. We proposed the LS-based algorithm to identify the hypergraph matching result for subchannel allocation and IIoT device association. Simulation results show that our proposed algorithm improves the learning utility of the system compared with the greedy algorithm.

Fig. 2. Learning utility of the system vs. number of the IIoT devices.

REFERENCES

- [1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Inform.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
- [2] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang, "A survey on 5G millimeter wave communications for UAV-assisted wireless networks," *IEEE Access*, vol. 7, pp. 117460–117504, Jul. 2019.
- [3] H. Yang, A. Alphones, W.-D. Zhong, C. Chen, and X. Xie, "Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5565–5576, Aug. 2020.
- [4] J. Yang, C. Ma, B. Jiang, G. Ding, G. Zheng, and H. Wang, "Joint optimization in cached-enabled heterogeneous network for efficient industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 831–844, May 2020.
- [5] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, "Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11016–11040, Jul. 2021.
- [6] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
- [7] W. Gao, Z. Zhao, G. Min, Q. Ni, and Y. Jiang, "Resource allocation for latency-aware federated learning in industrial internet-of-things," *IEEE Trans. Ind. Inform.*, 2021, Early Access.
- [8] L. U. Khan, M. Alsenswi, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "Resource optimized federated learning-enabled cognitive internet of things for smart industries," *IEEE Access*, vol. 8, pp. 168854–168864, Sep. 2020.
- [9] Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, "Communication-efficient federated learning for anomaly detection in industrial internet of things," in *Proc. IEEE (GLOBECOM)*, Taipei, Taiwan, Dec. 2020.
- [10] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning for digital twin edge networks in industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 17, no. 8, pp. 5709–5718, Aug. 2021.
- [11] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
- [12] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.
- [13] L. Zhang, H. Zhang, L. Yu, H. Xu, L. Song, and Z. Han, "Virtual resource allocation for mobile edge computing: A hypergraph matching approach," in *Proc. IEEE (GLOBECOM)*, Waikoloa, HI, Dec. 2019.