This is the peer reviewd version of the followng article:

Time-Correlated Sparsification for Efficient Over-the-Air Model Aggregation in Wireless Federated Learning / Sun, Y.; Zhou, S.; Niu, Z.; Gunduz, D.. - 2022-:(2022), pp. 3388-3393. (2022 IEEE International Conference on Communications, ICC 2022 COEX, kor 2022) [10.1109/ICC45855.2022.9839279].

Institute of Electrical and Electronics Engineers Inc.
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

16/05/2026 00:40

(Article begins on next page)

Time-Correlated Sparsification for Efficient Over-the-Air Model Aggregation in Wireless Federated Learning

Yuxuan Sun*, Sheng Zhou*, Zhisheng Niu*, Deniz Gündüz†
*Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
†Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK
Email: {sunyuxuan, sheng.zhou, niuzhs}@tsinghua.edu.cn, d.gunduz@imperial.ac.uk

arXiv:2202.08420v1 [cs.IT] 17 Feb 2022

***Abstract*—Federated edge learning (FEEL) is a promising distributed machine learning (ML) framework to drive edge intelligence applications. However, due to the dynamic wireless environments and the resource limitations of edge devices, communication becomes a major bottleneck. In this work, we propose time-correlated sparsification with hybrid aggregation (TCS-H) for communication-efficient FEEL, which exploits jointly the power of model compression and over-the-air computation. By exploiting the temporal correlations among model parameters, we construct a global sparsification mask, which is identical across devices, and thus enables efficient model aggregation over-the-air. Each device further constructs a local sparse vector to explore its own important parameters, which are aggregated via digital communication with orthogonal multiple access. We further design device scheduling and power allocation algorithms for TCS-H. Experiment results show that, under limited communication resources, TCS-H can achieve significantly higher accuracy compared to the conventional top-K sparsification with orthogonal model aggregation, with both i.i.d. and non-i.i.d. data distributions.**

I. Introduction

Federated edge learning (FEEL) refers to the implementation of federated learning algorithms [1], [2] in a wireless network, where edge devices train a shared machine learning (ML) model using their local datasets, and periodically communicate with a base station (BS) via wireless channels for global model aggregation. It is considered as a promising paradigm to facilitate edge intelligence, driving numerous applications such as Internet of things, augmented and virtual reality, self driving, and smart network management [3], [4].

However, due to resource limitations in terms of wireless channel bandwidth, transmit power, battery capacity, and computing capability of edge devices, it is challenging to train accurate models in a FEEL system. A particular challenge is the communication bottleneck, which limits the amount of information that can be exchanged between the BS and edge devices, and thus reduces the training performance.

To overcome the communication bottleneck, a straightforward idea is to reduce the traffic load. Model compression techniques, mainly quantization and sparsification, are proposed to reduce the communication load [5]–[9]. With quantization [5], fewer bits, or even a single bit [6], are used to represent each parameter in the ML model. With sparsification, the high dimensional ML model vector is transformed to a sparse vector, where only important elements are kept [7]. The two techniques can also be combined [8]. While a high dimensional mask is required to be transmitted from each device to the BS for conventional sparsification methods, it is shown in [9] that a global mask can be used instead by exploiting the temporal correlations during the convergence of the ML model. Communication load can also be reduced by enabling multiple iterations of local training, and carefully scheduling a subset of devices in each round [10], [11].

Another emerging way of improving the communication efficiency of FEEL is over-the-air computation (OAC) [12]–[16], which exploits the superposition property of the wireless multiple access channel from devices to the BS to average the local models over-the-air. It is shown in [12] that compared with the digital counterpart, an over-the-air FEEL system with N devices can improve the communication efficiency by $O\left(\frac{N}{\log_2 N}\right)$. However, due to the high dimension of the ML model, it is still not practical for the devices to update all the parameters to the BS at each time. A linear projection method is proposed in [13], [14], which projects the model vector to a lower-dimension by a pseudo-random matrix. However, this increases the computational complexity at the receiver, making it difficult for real implementation. Temporal correlations of model parameters are further exploited in [17] to reduce the recovery complexity.

In this paper, we combine the power of model compression with OAC, and propose time-correlated sparsification with hybrid aggregation (TCS-H) to further improve the communication efficiency of FEEL. Our algorithm is built upon the time-correlated sparsification (TCS) method in [9], which exploits the temporal correlations in the model parameters during training, and constructs a global mask for sparsification. As the global mask is identical across devices, we propose to aggregate the corresponding sparsified model differences over-the-air. Meanwhile, local sparsification is used to explore the important parameters specific to each device. With quantization, these sparse vectors, together with the local

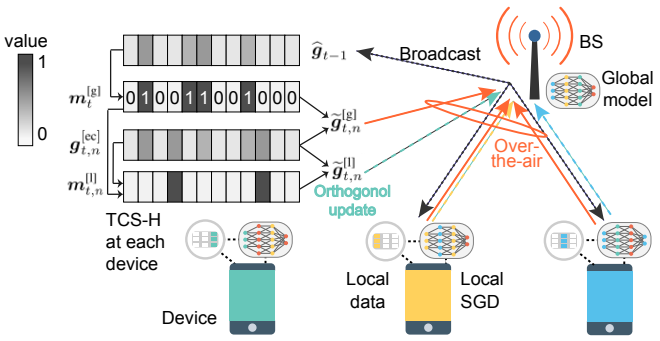

Fig. 1. Illustration of the FEEL system and the proposed TCS-H algorithm.

sparsification masks, are transmitted via digital communication with orthogonal multiple access. Under an average power constraint, we further design the device scheduling and power allocation algorithm. Experiment results on CIFAR-10 dataset show that, under limited communication resources, TCS-H can improve the accuracy by 6.6%, compared with the classical Top-K sparsification with orthogonal model aggregation.

II. Time-Correlated Sparsification with Hybrid Aggregation for FEEL

As shown in Fig. 1, we consider a FEEL system where N devices $\mathcal{N} = \{1, \ldots, N\}$ train a ML model collaboratively, under the coordination of a BS. Each device n owns a local dataset $\mathcal{D}_n$ with D data samples. The local loss function is defined as $F_n(\boldsymbol{w}) \triangleq \frac{1}{D}\sum_{\boldsymbol{\xi}_i \in \mathcal{D}_n} f(\boldsymbol{w}, \boldsymbol{\xi}_i)$, where $\boldsymbol{\xi}_i$ is a data sample in $\mathcal{D}_n$, $\boldsymbol{w} \in \mathbb{R}^d$ is the model vector to be trained, and $f(\boldsymbol{w}, \boldsymbol{\xi}_i)$ is a loss function measuring the fitting performance of $\boldsymbol{w}$ on a single data sample $\boldsymbol{\xi}_i$. The global loss function is defined as $F(\boldsymbol{w}) \triangleq \frac{1}{N}\sum_{n=1}^{N} F_n(\boldsymbol{w})$. The goal of FEEL is $\min_{\boldsymbol{w}} F(\boldsymbol{w})$, by optimizing the model vector $\boldsymbol{w}$.

To train the FEEL task, devices carry out stochastic gradient descent (SGD) locally, and communicate with the BS for global model aggregation in an iterative manner. There are M orthogonal sub-channels for uplink communications, indexed by $\mathcal{M} = \{1, \ldots, M\}$. Each device n has an average power constraint $\bar{P}_n$. A total number of C time slots are available for uplink communications during the whole training process, where the length of a time slot equals that of a symbol. We remark here that, different from the common assumption that limits the training rounds, we mainly focus on the uplink communication bottleneck by limiting the total number of resource blocks to MC. Accordingly, the total number of training rounds, denoted by T_{Algo}, depends on the FEEL algorithm and its parameter settings.

In order to optimize the training performance under the constraints on communication bandwidth and average transmit power, we aim to reduce the communication load while improving the efficiency of model aggregation. Combining model compression with OAC, we propose the TCS-H algorithm for FEEL, as shown in Algorithm 1.

In the initialization phase, the BS and all the devices start from a common model $\boldsymbol{w}_{-1}$. Each device calculates the local gradient using its own dataset, and the BS aggregates these to get the global gradient $\boldsymbol{g}_0$. As the initialization only involves a single round, the communication cost of the initial training phase is not taken into account.

Algorithm 1 TCS-H for FEEL

Initialization: Initial model $\boldsymbol{w}_{-1}$, initial global gradient $\widehat{\boldsymbol{g}}_0 = \boldsymbol{g}_0 = \boldsymbol{w}_0 - \boldsymbol{w}_{-1}$. Let $U_0 = 0$, $t = 1$.
while $\sum_{\tau=0}^{t-1} U_\tau \leq C$ and model is not converged **do**
  BS broadcasts the global model difference $\widehat{\boldsymbol{g}}_{t-1}$ of last round to all the devices.
  Local training at each device $n \in \mathcal{N}$**:**
  Recovers the global model $\boldsymbol{w}_{t-1} \leftarrow \boldsymbol{w}_{t-2} + \widehat{\boldsymbol{g}}_{t-1}$.
  Carries out local SGD for H iterations, and gets local model difference: $\boldsymbol{g}_{t,n} = \boldsymbol{w}_{t,n,H} - \boldsymbol{w}_{t-1}$.
  Error compensation: $\boldsymbol{g}_{t,n}^{[\text{ec}]} = \boldsymbol{g}_{t,n} + \boldsymbol{e}_{t-1,n}$.
  Global mask: $\boldsymbol{m}_t^{[\text{g}]} = S_{\text{top}}(\widehat{\boldsymbol{g}}_{t-1}, K^{[\text{g}]})$.
  Sparsification with global mask: $\widetilde{\boldsymbol{g}}_{t,n}^{[\text{g}]} = \boldsymbol{m}_t^{[\text{g}]} \circ \boldsymbol{g}_{t,n}^{[\text{ec}]}$.
  Local mask: $\boldsymbol{m}_{t,n}^{[\text{l}]} = S_{\text{top}}\left((\boldsymbol{1} - \boldsymbol{m}_t^{[\text{g}]}) \circ \boldsymbol{g}_{t,n}^{[\text{ec}]}, K^{[\text{l}]}\right)$.
  Sparsification and quantization with local mask: $\widetilde{\boldsymbol{g}}_{t,n}^{[\text{l}]} = Q(\boldsymbol{m}_{t,n}^{[\text{l}]} \circ \boldsymbol{g}_{t,n}^{[\text{ec}]}, q)$.
  Model aggregation at the BS:
  Device scheduling and power allocation according to Algorithm 2. Records communication slots U_t.
  Aggregates $\widetilde{\boldsymbol{g}}_{t,n}^{[\text{g}]}$ of all the scheduled devices over-the-air, and receives $\widetilde{\boldsymbol{g}}_{t,n}^{[\text{l}]}$, $\boldsymbol{m}_{t,n}^{[\text{l}]}$ via orthogonal transmissions.
  Aggregates the global model difference $\widehat{\boldsymbol{g}}_t$ according to (2), and updates global model $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} + \widehat{\boldsymbol{g}}_t$.
  $t \leftarrow t + 1$.
  Error cumulation at each device $n \in \mathcal{N}$**:**
  if $n \in \mathcal{N}_t$ **then**
    $\boldsymbol{e}_{t,n} \leftarrow \boldsymbol{g}_{t,n}^{[\text{ec}]} - \widetilde{\boldsymbol{g}}_{t,n}^{[\text{g}]} - \widetilde{\boldsymbol{g}}_{t,n}^{[\text{l}]}$.
  else
    $\boldsymbol{e}_{t,n} \leftarrow \boldsymbol{e}_{t-1,n}$.
  end if
end while

In the main loop, there are three steps in each round.

1) Local SGD: In the t-th training round, each device receives the global model difference $\widehat{\boldsymbol{g}}_{t-1}$ from the BS, and updates the global model from the last round according to $\boldsymbol{w}_{t-1} = \boldsymbol{w}_{t-2} + \widehat{\boldsymbol{g}}_{t-1}$. Each device n then carries out H iterations of local SGD according to

$$\boldsymbol{w}_{t,n,i} = \boldsymbol{w}_{t,n,i-1} - \eta_t \nabla F_n(\boldsymbol{w}_{t,n,i-1}; \mathcal{B}_{t,n,i}), i = 1, \ldots, H,$$

where $\boldsymbol{w}_{t,n,0} = \boldsymbol{w}_{t-1}$, η_t is the learning rate, and $\mathcal{B}_{t,n,i}$ is a randomly sampled mini-batch with batch size B. The local model difference is then calculated according to $\boldsymbol{g}_{t,n} = \boldsymbol{w}_{t,n,H} - \boldsymbol{w}_{t-1}$, and compensated with the cumulated error $\boldsymbol{e}_{t-1,n}$, which yields $\boldsymbol{g}_{t,n}^{[\text{ec}]} = \boldsymbol{g}_{t,n} + \boldsymbol{e}_{t-1,n}$. The cumulated error keeps track of the parameters that are not updated to the BS due to compression, helping to accelerate the training [7].

2) Local Model Compression: We use the TCS method [9] as well as stochastic quantization to reduce the communication load in each round. Let function $S_{\text{top}}(\boldsymbol{x}, K)$ return a mask vector $\boldsymbol{m} \in \{0, 1\}^d$, which indicates the locations of non-zero elements after a top-K sparsification operation on $\boldsymbol{x}$. That is,

if the i-th element of $\boldsymbol{x}$ after a top-K sparsification is not zero, then $\boldsymbol{m}[i]=1$; otherwise $\boldsymbol{m}[i]=0$. The top-K sparsification keeps the K elements of $\boldsymbol{x}$ with largest absolute values, while setting the remaining elements to zero.

Based on the global model difference of the last round, $\widehat{\boldsymbol{g}}_{t-1}$, each device first calculates the global mask $\boldsymbol{m}_t^{[\mathrm{g}]}=S_{\mathrm{top}}(\widehat{\boldsymbol{g}}_{t-1},K^{[\mathrm{g}]})$ with sparsity $K^{[\mathrm{g}]}$. Then, the local model difference $\boldsymbol{g}_{t,n}^{[\mathrm{ec}]}$ is sparsified with the global mask, which yields $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}=\boldsymbol{m}_t^{[\mathrm{g}]}\circ\boldsymbol{g}_{t,n}^{[\mathrm{ec}]}$, where $\circ$ represents the element-wise multiplication. By exploiting the temporal correlations of the model, the global mask is the same for all the devices. This enables OAC for global model aggregation.

Then, each device generates a local mask $\boldsymbol{m}_{t,n}$ to explore the important elements of $\boldsymbol{g}_{t,n}^{[\mathrm{ec}]}$, where $\boldsymbol{m}_{t,n}^{[\mathrm{l}]}=S_{\mathrm{top}}\left((\mathbf{1}-\boldsymbol{m}_t^{[\mathrm{g}]})\circ\boldsymbol{g}_{t,n}^{[\mathrm{ec}]},K^{[\mathrm{l}]}\right)$, and $\mathbf{1}$ is a d-dimension all-one vector. The local model difference is then sparsified with the local mask and then quantized, which is given by $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}=Q(\boldsymbol{m}_{t,n}^{[\mathrm{l}]}\circ\boldsymbol{g}_{t,n}^{[\mathrm{ec}]},q)$. Here, $Q(\boldsymbol{x},q)$ represents a q-bit stochastic quantization function, with examples in [8].

3) Global Model Aggregation: As the positions of non-zero elements of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}$ are the same for all the devices, we propose to use OAC for the global aggregation of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}$ to improve the communication efficiency. Meanwhile, $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ and local masks $\boldsymbol{m}_{t,n}^{[\mathrm{l}]}$ are aggregated via digital communication with orthogonal multiple access; and thus, quantization is introduced above to further reduce the communication load.

Specifically, in the t-th round, a number of N_t devices in set $\mathcal{N}_t\subset\mathcal{N}$ are scheduled for global aggregation according to Algorithm 2, which will be introduced in the next section. For OAC, each device $n\in\mathcal{N}_t$ extracts the non-zero elements of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}$ and partitions them evenly into M segments $\left[\widetilde{\boldsymbol{g}}_{t,n,1}^{[\mathrm{g}]},\cdots,\widetilde{\boldsymbol{g}}_{t,n,M}^{[\mathrm{g}]}\right]$, each with $\left\lceil\frac{K^{[\mathrm{g}]}}{M}\right\rceil$ or $\left\lfloor\frac{K^{[\mathrm{g}]}}{M}\right\rfloor$ elements. The wireless channel gain between the BS and device n in the m-th sub-channel is denoted by $h_{t,n,m}$, which is assumed to be constant during each round. In the m-th sub-channel, each scheduled device transmits $\frac{\sigma_t\widetilde{\boldsymbol{g}}_{t,n,m}^{[\mathrm{g}]}}{h_{t,n,m}}$ synchronously with all the other scheduled devices for OAC, where σ_t is a power scalar that determines the received signal-to-noise ratio (SNR).

At the BS, the received signal over sub-channel m is

$$\boldsymbol{y}_{t,m}=\sum_{n\in\mathcal{N}_t}h_{t,n,m}\frac{\sigma_t\widetilde{\boldsymbol{g}}_{t,n,m}^{[\mathrm{g}]}}{h_{t,m,n}}+\boldsymbol{z}_{t,m}=\sigma_t\sum_{n\in\mathcal{N}_t}\widetilde{\boldsymbol{g}}_{t,n,m}^{[\mathrm{g}]}+\boldsymbol{z}_{t,m},\quad(1)$$

where $\boldsymbol{z}_{t,m}$ is a noise vector, with each element following Gaussian distribution with zero mean and variance σ_0^2. Let $j(i)$ be the index of the i-th non-zero element of $\boldsymbol{m}_t^{[\mathrm{g}]}$. The BS constructs a d-dimension sparse vector $\boldsymbol{y}_t\in\mathbb{R}^d$ based on $\boldsymbol{m}_t^{[\mathrm{g}]}$, where $\boldsymbol{y}_t[j(i)]$ equals the i-th element of $[\boldsymbol{y}_{t,1},\cdots,\boldsymbol{y}_{t,M}]$. Likewise, $[\boldsymbol{z}_{t,1},\cdots,\boldsymbol{z}_{t,M}]$ is mapped to $\boldsymbol{z}_t\in\mathbb{R}^d$.

In the digital communication part, each device $n\in\mathcal{N}$ transmits both $\boldsymbol{m}_{t,n}^{[\mathrm{l}]}$ and the non-zero elements of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ to the BS, requiring $Q^{[\mathrm{l}]}=(\log_2 d+q)K^{[\mathrm{l}]}$ bits. The sub-channel assignment and the corresponding power allocation are given by Algorithm 2.

The global model difference $\widehat{\boldsymbol{g}}_t$ is aggregated at the BS:

$$\begin{aligned}\widehat{\boldsymbol{g}}_t=&\frac{\boldsymbol{y}_t}{\sigma_t N_t}+\frac{1}{N_t}\sum_{n\in\mathcal{N}_t}\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}\\=&\frac{1}{N_t}\sum_{n\in\mathcal{N}_t}\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}+\frac{\boldsymbol{z}_t}{\sigma_t N_t}+\frac{1}{N_t}\sum_{n\in\mathcal{N}_t}\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}.\end{aligned}\quad(2)$$

Finally, error cumulation is carried out at each device. If device n is scheduled, the difference between the transmitted vector and the original vector is accumulated, i.e., $\boldsymbol{e}_{t,n}=\boldsymbol{g}_{t,n}+\boldsymbol{e}_{t-1,n}-\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}-\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$. Otherwise, $\boldsymbol{e}_{t,n}=\boldsymbol{e}_{t-1,n}$. Define U_t as the total number of communication time slots used in the t-th round. If the global model is not converged and the communication resource is not used up, i.e., $\sum_{\tau=1}^{t}U_\tau<C$, the TCS-H algorithm will start the next round.

III. Device Scheduling and Power Allocation

In this section, we introduce the device scheduling and power allocation algorithm for TCS-H. We consider that each device has an average power constraint $\bar{P}_n$ for communication. For OAC, the total power required in the t-th round at device n is $P_{t,n}^{[\mathrm{g}]}=\sum_{m=1}^{M}\left\|\frac{\sigma_t\widetilde{\boldsymbol{g}}_{t,n,m}^{[\mathrm{g}]}}{h_{t,n,m}}\right\|_2^2$, with $U_t^{[\mathrm{g}]}=\left\lceil\frac{K^{[\mathrm{g}]}}{M}\right\rceil$ communication time slots. For digital communication, define $P_{t,n,m}^{[\mathrm{l}]}$ as the power allocated to sub-channel m by device n in round t, and denote the corresponding required slots by $U_{t,n}^{[\mathrm{l}]}$. Then, the average power constraint is given by $\frac{1}{C}\sum_{t=1}^{T_{\mathrm{Algo}}}\left(P_{t,n}^{[\mathrm{g}]}+U_{t,n}^{[\mathrm{l}]}\sum_{m=1}^{M}P_{t,n,m}^{[\mathrm{l}]}\right)\le\bar{P}_n$.

Let $P_{t,n}^{[\mathrm{l}]}=\sum_{m=1}^{M}P_{t,n,m}^{[\mathrm{l}]}$, and $P_{t,n}=P_{t,n}^{[\mathrm{g}]}+U_{t,n}^{[\mathrm{l}]}P_{t,n}^{[\mathrm{l}]}$. Then, at the start of round t, the average power constraint of the n-th device is given by $\bar{P}_{t,n}=\frac{C\bar{P}_n-\sum_{\tau=1}^{t-1}P_{\tau,n}}{C-\sum_{\tau=1}^{t-1}U_\tau}$.

Device Scheduling: As shown in Algorithm 2, after local gradient computation in the t-th round, devices that satisfy the power constraint $P_{t,n}^{[\mathrm{g}]}\le\alpha_{t,n}\bar{P}_{t,n}U_{t,n}^{[\mathrm{g}]}$ are scheduled by the BS for global aggregation. Here, $\alpha_{t,n}>0$ is a coefficient which dynamically adjusts the power allocation policy. $\alpha_{t,n}=1$ indicates a simple myopic policy. If $\alpha_{t,n}>1$, power is allocated in a more aggressive manner in the current round, and vice versa. In this work, we consider a myopic policy and do not further optimize this term, leaving it as a future work. We remark that, the selection of $\alpha_{t,n}$ requires an online device scheduling and power allocation algorithm, which can be designed based on stochastic optimization theories such as Lyapunov optimization. We refer to our previous work [16] for such a design approach.

Sub-Channel Assignment and Power Allocation: For all the scheduled devices $n\in\mathcal{N}_t$, the BS needs to assign sub-channels and allocate power for them to aggregate $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ and $\boldsymbol{m}_{t,n}^{[\mathrm{l}]}$. We consider that each sub-channel is allocated to one device in each round. The goal is to minimize the required communication time slots $U_t^{[\mathrm{l}]}\triangleq\max_{n\in\mathcal{N}_t}U_{t,n}^{[\mathrm{l}]}$, under the power constraints of the scheduled devices. Let $\beta_{t,n,m}=1$ indicate that the m-th sub-channel is assigned to the n-th device for digital communication in round t, and

Algorithm 2 Device Scheduling and Power Allocation Algorithm for TCS-H

1: **Input:** $\bar{P}_{t,n}$, $P_{t,n}^{[\mathrm{g}]}$, $h_{t,n,m}$, $\mathcal{N}_t = \emptyset$, $\mathcal{M}_t = \emptyset$.
2: **Device scheduling:**
3: **for** $n \in \mathcal{N}$ **do**
4: If $P_{t,n}^{[\mathrm{g}]} \leq \alpha_{t,n}\bar{P}_{t,n}U_{t,n}^{[\mathrm{g}]}$, then $\mathcal{N}_t = \mathcal{N}_t \cup \{n\}$.
5: **end for**
6: **Sub-channel assignment and power allocation:**
7: Initial sub-channel assignment by solving a bottleneck matching problem.
8: For $\forall n \in \mathcal{N}_t$, calculate rate $R_{t,n}$ under the current sub-channel assignment.
9: Update $\mathcal{M}_t = \{m|\beta_{t,n,m} = 1, m \in \mathcal{M}\}$. Let $\mathcal{M}_t' = \mathcal{M} - \mathcal{M}_t$ and $\mathcal{N}_t' = \mathcal{N}_t$.
10: **while** $\mathcal{M}_t' \neq \emptyset$ and $\mathcal{N}_t' \neq \emptyset$ **do**
11: $n^\dagger = \arg\min_{n\in\mathcal{N}_t'} R_{t,n}$.
12: $m^\dagger = \arg\max_{m\in\mathcal{M}_t'} r_{t,n^\dagger,m}$.
13: Calculate potential gain $\Delta R_{t,n^\dagger}$ according to (4).
14: **if** $\Delta R_{t,n^\dagger} > 0$ **then**
15: $\beta_{t,n^\dagger,m^\dagger} = 1$, $\mathcal{M}_t = \mathcal{M}_t \cup \{m^\dagger\}$, $\mathcal{M}_t' = \mathcal{M} - \mathcal{M}_t$. Update $R_{t,n^\dagger}$.
16: **else**
17: $\mathcal{N}_t' = \mathcal{N}_t' - \{n^\dagger\}$.
18: **end if**
19: **end while**
20: For $\forall n \in \mathcal{N}_t$, use water-filling algorithm for the rate-optimal power allocation, as shown in (5).
21: **Output:** $\mathcal{N}_t$, $\boldsymbol{\beta}_t$, $\boldsymbol{P}_t$, U_t

$\beta_{t,n,m} = 0$ otherwise. Let $\boldsymbol{\beta}_t = \{\beta_{t,n,m}|n \in \mathcal{N}_t, m \in \mathcal{M}\}$, and $\boldsymbol{P}_t = \{P_{t,n,m}^{[\mathrm{l}]}|n \in \mathcal{N}_t, m \in \mathcal{M}\}$. The problem is:

$$\mathcal{P}1: \min_{\boldsymbol{\beta}_t, \boldsymbol{P}_t} \max_{n\in\mathcal{N}_t} U_{t,n}^{[\mathrm{l}]} \tag{3a}$$

$$\text{s.t. } U_{t,n}^{[\mathrm{l}]} R_{t,n} \geq Q^{[\mathrm{l}]}, \forall n \in \mathcal{N}_t, \tag{3b}$$

$$\sum_{m=1}^{M} P_{t,n,m}^{[\mathrm{l}]} \leq \bar{P}_{t,n}, \forall n \in \mathcal{N}_t, \tag{3c}$$

$$\sum_{n\in\mathcal{N}_t} \beta_{t,n,m} \leq 1, \forall m \in \mathcal{M}, \ \beta_{t,n,m} \in \{0,1\}, \tag{3d}$$

where $R_{t,n} = \sum_{m=1}^{M} \beta_{t,n,m} \log_2\left(1 + \frac{P_{t,n,m}^{[\mathrm{l}]}|h_{t,n,m}|^2}{\sigma_0^2}\right)$ is the achievable communication rate of device n in round t.

Problem $\mathcal{P}1$ is a mixed-integer non-linear programming problem, which is difficult to solve. Inspired by the channel and power allocation policies for orthogonal frequency division multiplexing (OFDM) systems [18], [19], we decouple sub-channel assignment from power allocation, and solve $\mathcal{P}1$ in the following three steps. Note that the number of sub-channels is in general larger than that of the devices, and thus we assume $M \geq N$ in the following.

1) Initial Sub-Channel Assignment: The BS first assigns one sub-channel $m \in \mathcal{M}$ to each device $n \in \mathcal{N}_t$, such that the minimum achievable rate of devices is maximized. Let $r_{t,n,m} = \log_2\left(1 + \frac{\bar{P}_{t,n}|h_{t,n,m}|^2}{\sigma_0^2}\right)$ be the communication rate when a single sub-channel m is allocated to device n. The sub-channels $\mathcal{M}$ and devices $\mathcal{N}_t$ form a complete undirected bipartite graph, where the weight of each edge between $m \in \mathcal{M}$ and $n \in \mathcal{N}_t$ is $r_{t,n,m}$. The assignment problem is called a *bottleneck matching* or *max-min matching* problem, whose optimal solution can be achieved by the threshold policy [20] with complexity $O(M^2N_t^2)$. The key idea of the threshold policy is to activate edges whose weights are larger than a progressively decreasing threshold, and find a maximum cardinality matching given the active edges. The policy terminates when each device is assigned a sub-channel.

2) Remaining Sub-Channel Assignment: The remaining $M - N_t$ sub-channels, denoted by $\mathcal{M}_t'$, are further assigned in a heuristic manner. Let $\mathcal{N}_t'$ be the set of devices whose sum rates $R_{t,n}$ can be improved if more sub-channels are assigned, with $\mathcal{N}_t' = \mathcal{N}_t$ in the beginning. As shown in Lines 10-19 in Algorithm 2, in each iteration, we find device $n^\dagger \in \mathcal{N}_t'$ with the minimum sum rate, and the best remaining channel $m^\dagger \in \mathcal{M}_t'$ associated with $n^\dagger$. The potential gain of the sum rate is approximated under equal power allocation, which is given by

$$\Delta R_{t,n^\dagger} = \sum_{m=1}^{M} \beta'_{t,n^\dagger,m} \log_2\left(1 + \frac{\bar{P}_{t,n^\dagger}|h_{t,n^\dagger,m}|^2}{\sigma_0^2 \sum_{m=1}^{M} \beta'_{t,n^\dagger,m}}\right) - \sum_{m=1}^{M} \beta_{t,n^\dagger,m} \log_2\left(1 + \frac{\bar{P}_{t,n^\dagger}|h_{t,n^\dagger,m}|^2}{\sigma_0^2 \sum_{m=1}^{M} \beta_{t,n^\dagger,m}}\right), \tag{4}$$

where $\beta'_{t,n^\dagger,m^\dagger} = 1$, and $\beta'_{t,n^\dagger,m} = \beta_{t,n^\dagger,m}$ for $\forall m \neq m^\dagger$.

If $\Delta R_{t,n^\dagger}$ is positive, then we assign sub-channel $m^\dagger$ to device $n^\dagger$. Otherwise, assigning more sub-channels is not beneficial to rate increment for device $n^\dagger$, which is thus removed from $\mathcal{N}_t'$. The process is terminated when all the sub-channels are assigned, or no device can benefit from having more sub-channels.

3) Power Allocation: Given sub-channel assignment $\boldsymbol{\beta}_t$, we finally implement the water-filling algorithm for each device to optimize power allocation for rate maximization. For $\forall n \in \mathcal{N}_t$, power is allocated according to

$$P_{t,n,m}^{[\mathrm{l}]} = \begin{cases} \left(\frac{1}{\lambda_{t,n}} - \frac{\sigma_0^2}{|h_{t,n,m}|^2}\right)^+, & \text{if } \beta_{t,n,m} = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $\lambda_{t,n}$ is chosen such that $\sum_{m=1}^{M} P_{t,n,m}^{[\mathrm{l}]} = \bar{P}_{t,n}$, $(x)^+ = x$ if $x > 0$, and $(x)^+ = 0$ otherwise.

Finally, Algorithm 2 outputs the device scheduling $\mathcal{N}_t$, sub-channel assignment $\boldsymbol{\beta}_t$, power allocation $\boldsymbol{P}_t$, and the total number of communication slots U_t consumed in round t.

IV. Convergence Analysis

In this section, we provide a convergence guarantee for the proposed TCS-H algorithm.

Lemma 1. *For $\forall \boldsymbol{x} \in \mathbb{R}^d$, using a global mask $\boldsymbol{m}^{[\mathrm{g}]}$ and a local mask $\boldsymbol{m}^{[\mathrm{l}]}$ for sparsification, and applying a q-bit stochastic quantization as [5] to the local sparsified vector,*

$$\mathbb{E}\left[\|\boldsymbol{x} - \boldsymbol{x} \circ \boldsymbol{m}^{[\mathrm{g}]} - Q(\boldsymbol{x} \circ \boldsymbol{m}^{[\mathrm{l}]}, q)\|_2^2\right]$$

$$\leq \left[1-\left(1-\frac{K^{[\mathrm{l}]}}{2^{2q-2}}\right)\frac{K^{[\mathrm{g}]}+K^{[\mathrm{l}]}}{d}\right]\|\boldsymbol{x}\|_2^2, \quad (6)$$

where $K^{[\mathrm{g}]}$ and $K^{[\mathrm{l}]}$ are the numbers of non-zero elements in $\boldsymbol{m}^{[\mathrm{g}]}$ and $\boldsymbol{m}^{[\mathrm{l}]}$, respectively, $K^{[l]} < 2^{2q-2}$, and the expectation is taken over sparsification and stochastic quantization.

To facilitate the convergence analysis, we follow the literature [5], [7], [8], and assume that the local loss function $F_n(\boldsymbol{w})$ is L-smooth and μ-strongly convex, i.e., for $\forall \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^d$ and $n \in \mathcal{N}$, $F_n(\boldsymbol{v}) - F_n(\boldsymbol{w}) \leq \langle \nabla F_n(\boldsymbol{w}), \boldsymbol{v}-\boldsymbol{w}\rangle + \frac{L}{2}\|\boldsymbol{v}-\boldsymbol{w}\|_2^2$, $F_n(\boldsymbol{v}) - F_n(\boldsymbol{w}) \geq \langle \nabla F_n(\boldsymbol{w}), \boldsymbol{v}-\boldsymbol{w}\rangle + \frac{\mu}{2}\|\boldsymbol{v}-\boldsymbol{w}\|_2^2$. Here, $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ denotes the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$. We also assume that the local stochastic gradient has bounded l_2-norm, i.e., $\mathbb{E}_{\boldsymbol{x}_n \sim \mathcal{D}_n}\left[\left\|\nabla F_n\left(\boldsymbol{w}_{t,n,i}; \boldsymbol{x}_n\right)\right\|_2^2\right] \leq G^2$, $\forall n, t, i$.

Define $\gamma \triangleq \left(1-\frac{K^{[\mathrm{l}]}}{2^{2q-2}}\right)\frac{K^{[\mathrm{g}]}+K^{[\mathrm{l}]}}{d}$, $A_1 \geq \frac{4a\gamma(1-\gamma^2)}{a\gamma-4H}$, where $a > 0$, and the following auxiliary variables: $V_1 = \left(\frac{7\mu H}{2}+6HL\right)\frac{\sigma_0^2 K^{[\mathrm{g}]}}{N^2}$, $V_2 = \frac{HG^2}{BN}$, $V_3 = \left(\frac{3\mu}{2}+\frac{9L}{2}\right)G^2(H-1)H + (14\mu+24L)\frac{A_1 H^3 G^2}{\gamma}$.

The convergence of the proposed TCS-H algorithm under full device participation is given as follows, which mainly shows the impact of the channel noise and model compression.

Theorem 1. *Given the initial global model $\boldsymbol{w}_0$, let $\sigma_t = (a+t)^2$, the learning rate $\eta_t = \frac{8}{\mu H(a+t)}$, and define $\omega_t = (a+t)^2$, when $a > \max\left\{\frac{4H}{\gamma}, \frac{32L}{\mu}\right\}$,*

$$F(\bar{\boldsymbol{w}}_T) - F^* \leq \frac{La^3}{4S_T}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_2^2 + \frac{LT^2}{\mu H S_T}V_1 + \frac{8LT(T+2a+1)}{\mu^2 H^2 S_T}V_2 + \frac{64LT}{\mu^3 H^3 S_T}V_3, \quad (7)$$

where $S_T = \sum_{t=1}^{T}\omega_t$, $\bar{\boldsymbol{w}}_T = \frac{1}{S_T}\sum_{t=1}^{T}\omega_t \boldsymbol{w}_{t-1}$, $\boldsymbol{w}^*$ and F^* are the optimal global model and minimum global loss, respectively.

The proof uses Lemma 1 and follows the perturbed iterate framework in [7], [8]. We remark two major differences compared with the proof in [8]. One is that we further take into account the wireless channel noise, which introduces a new term $\frac{LT^2}{\mu H S_T}V_1$ in the bound. The other is that we consider the convergence of the global model, by folding up the local iterations in the proof. As S_T is at the scale of $O(T^3)$, the bound given in (7) indicates the convergence of TCS-H.

V. Experiments

In this section, we evaluate the proposed TCS-H algorithm for FEEL by considering an image classification task on the CIFAR-10[1] dataset. There are $N = 20$ devices and $M = 25$ sub-channels. We consider both independent and identically distributed (i.i.d.) and non-i.i.d. local data across devices. For the i.i.d. case, the whole training dataset is partitioned into N disjoint subsets uniformly at random. For the non-i.i.d. case, each device stores 2 out of 10 classes of data samples.

[1]https://www.cs.toronto.edu/~kriz/cifar.html

A convolutional neural network (CNN) with the same structure as [16] is trained. The total number of parameters of the considered CNN is 258898. The mini-batch size is 64, and the learning rate is 0.05. Regarding the wireless channel, we consider Rayleigh fading with scale parameter 1, and additive white Gaussian noise with variance $\sigma_0^2 = 10^{-6}$. The average power constraint is $\bar{P}_n = 5\mathrm{mW}$, $\forall n$, and the power scalar is $\sigma_t = 5$, $\forall t$.

We compare the proposed TCS-H algorithm with two benchmarks. 1) Top-K: each device sparsifies the local model difference $\boldsymbol{g}_{t,n}^{[\mathrm{ec}]}$ by keeping $K = K^{[\mathrm{g}]} + K^{[\mathrm{l}]}$ elements with the largest absolute values. Orthogonal digital communication is used for global aggregation. 2) TCS-D [9]: it uses TCS with the same global and local sparsities $K^{[\mathrm{g}]}$ and $K^{[\mathrm{l}]}$, but both $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}$ and $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ are transmitted via orthogonal digital communications. For both Top-K and TCS-D, the number of bits to transmit each element of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{g}]}$ and $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ is the same as that of $\widetilde{\boldsymbol{g}}_{t,n}^{[\mathrm{l}]}$ in TCS-H, represented by q.

Fig. 2 shows the model accuracy achieved by different algorithms in the i.i.d. case, and the corresponding number of communication resource blocks $\sum_{\tau=1}^{t} U_\tau M$ consumed. We run each algorithm for 3000 rounds, with $H = 10$ local iterations in each round. In the legend, $(\phi^{[\mathrm{g}]}, \phi^{[\mathrm{l}]})$ shows the global and local sparsities, i.e., $\phi^{[\mathrm{g}]} = \frac{K^{[\mathrm{g}]}}{d}$ and $\phi^{[\mathrm{l}]} = \frac{K^{[\mathrm{l}]}}{d}$. The default value of q is 16, unless specified. Given $\phi^{[\mathrm{g}]} = 0.2$ and $\phi^{[\mathrm{l}]} = 0.05$, the average fraction of devices scheduled by TCS-H is 0.65, and thus for Top-K and TCS-D, we randomly schedule 13 devices in each round.

When $\phi^{[\mathrm{g}]} = 0.2$ and $\phi^{[\mathrm{l}]} = 0.05$, we find that TCS can maintain similar accuracy as Top-K, while greatly reduce the communication cost without transmitting the global masks. Meanwhile, the proposed TCS-H algorithm with OAC can further save 68% communication resources compared with the TCS-D algorithm with pure orthogonal aggregation. Comparing TCS-H with different sparsities, we find that even when $\phi^{[\mathrm{l}]} = 0.0001$, the TCS-H algorithm can still achieve a similar accuracy as that with $\phi^{[\mathrm{l}]} = 0.05$. This validates the strong temporal correlation of the model, and thus the feasibility of the TCS-H algorithm. Last but not least, TCS-H can outperform the Top-K algorithm in model accuracy and communication efficiency simultaneously. For example, when $\phi^{[\mathrm{l}]} = 0.05$, the model accuracy achieved by TCS-H with $\phi^{[\mathrm{g}]} = 0.5$ is 2.18% higher than that of Top-K with $\phi^{[\mathrm{g}]} = 0.2$, while the number of communication resource blocks consumed by TCS-H is 75% fewer.

In the non-i.i.d. case, we run each algorithm for 20000 rounds, with one local iteration in each round. 11 devices are scheduled by TCS-H on average, which is also set for the Top-K and TCS-D benchmarks. As shown in Fig. 3, under non-i.i.d. data, the proposed TCS-H algorithm can still achieve a higher model accuracy than the Top-K and TCS-D benchmarks with fewer communication resources, which is similar to the i.i.d. case. Moreover, TCS-H has a significant gain when the total communication resource is limited. For example, if the number of communication resource blocks is

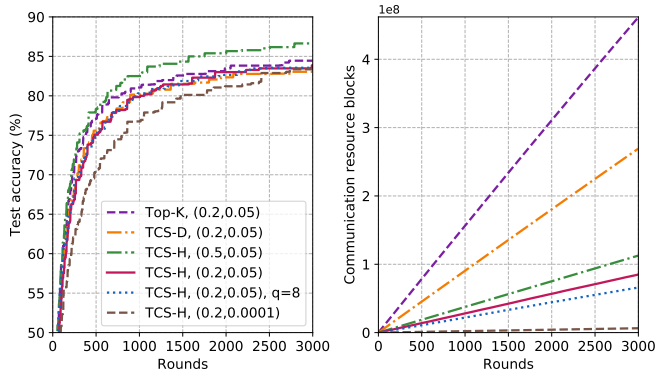

Fig. 2. Model accuracy (left) and the required communication resources (right) of different algorithms on CIFAR-10 dataset with i.i.d. data.

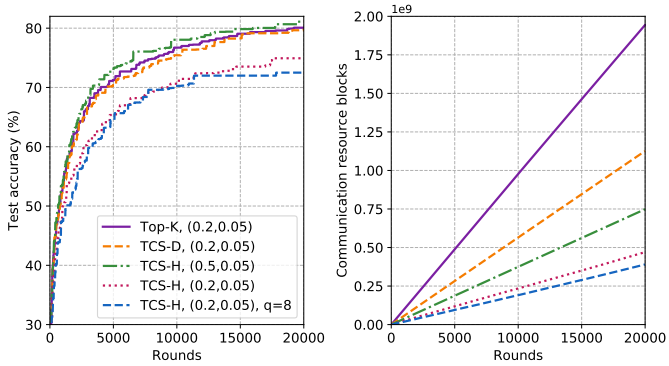

Fig. 3. Model accuracy (left) and the required communication resources (right) of different algorithms on CIFAR-10 dataset with non-i.i.d. data.

limited to 7.5×10^8, then TCS-H, TCS-D and Top-K can train the CNN model for 20000, 13290 and 7664 rounds, respectively. Accordingly, the accuracies achieved by the three algorithms are 81.1%, 77.7% and 74.5%. TCS-H increases the model accuracy by 6.6% compared with Top-K sparsification. Such result validates the superior communication efficiency of TCS-H.

VI. Conclusions

In this work, we have proposed a TCS-H algorithm to improve the communication efficiency of FEEL. By exploiting the temporal correlations among model parameters, a global mask is constructed in each training round, which enables the model aggregation over-the-air. The important parameters of each individual device are further explored and aggregated via orthogonal digital communication. Experiments on CIFAR-10 dataset have shown that, the proposed TCS-H is extremely promising when the wireless communication resource is limited, with a 6.6% accuracy improvement compared to the Top-K sparsification via orthogonal model aggregation. Future directions include the design of online device scheduling policies, and the optimization of the sparsity and quantization levels under various kinds of resource constraints.

References

[1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *NIPS Workshop on Private Multi-Party Machine Learning*, Oct. 2016.

[2] B. McMahan, E. Moore, D. Ramage, et al. "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artificial Intelligence and Statistics (AIStats)*, pp. 1273-1282, Apr. 2017.

[3] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," in *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204-2239, Nov. 2019.

[4] M. Chen, et al., "Distributed learning in wireless networks: Recent progress and future challenges," in *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579-3605, Dec. 2021.

[5] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1709-1720, Dec. 2017.

[6] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimization for non-convex problems," in *Proc. International Conference on Machine Learning (ICML)*, pp. 560-569, Jul. 2018.

[7] S. U. Stich, J. B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4452-4463, Dec. 2018.

[8] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 14668-14679, Dec. 2019.

[9] E. Ozfatura, K. Ozfatura, and D. Gunduz, "Time-correlated sparsification for communication-efficient federated learning," [Online] Available: https://arxiv.org/abs/2101.08837, Jan. 2021.

[10] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," in *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317-333, Jan. 2020.

[11] W. Shi, S. Zhou, Z. Niu, M. Jiang and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453-467, Jan. 2021.

[12] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491-506, Jan. 2020.

[13] M. Mohammadi Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *IEEE Trans. Signal Process.*, vol. 68, pp. 2155-2169, Apr. 2020.

[14] M. Mohammodi Amiri and D. Gunduz, "Federated learning over wireless fading channels," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546-3557, May 2020.

[15] Y. Sun, W. Shi, X. Huang, S. Zhou and Z. Niu, "Edge learning with timeliness constraints: Challenges and solutions," in *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 27-33, Dec. 2020.

[16] Y. Sun, S. Zhou, Z. Niu, and D. Gunduz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227-242, Jan. 2022.

[17] D. Fan, X. Yuan and Y. -J. A. Zhang, "Temporal-structure-assisted gradient aggregation for over-the-air federated edge learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3757-3771, Dec. 2021.

[18] L. Gao, and S. Cui, "Efficient subcarrier, power, and rate allocation with fairness consideration for OFDMA uplink," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1507–1511, 2008.

[19] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks," in *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 226-234, Feb. 2009.

[20] R. S. Garfinkel, "An improved algorithm for the bottleneck assignment problem." *Operations Research*, vol. 19, no. 7, pp. 1747–51, 1971.