Federated Learning for Wireless Networks: Accuracy v.s. Energy Tradeoff

Nguyen H. Tran, *Member, IEEE,* Minh N. H. Nguyen, Wei Bao

***Abstract*—On-UE AI**

I. Introduction

A. Federated Learning

According to the data minimization principle on consumer data privacy proposed by the White House report in 2012 [39], recently there are increasing interests in a machine learning technique that exploits the participation of serveral mobile phone users, called Federated Learning [**?**], [**?**], [**?**], [**?**]. This learning technique allows the users to collaboratively build a shared learning model while preserving all training data on the user equipments (UEs). Specifically, UEs compute the updates to the current global model on their local training data, which will be aggregated and feedback by a central server, e.g. in the datacenters, so that all UEs have access to the same global model in order to compute their new updates. This process is repeated until an accuracy level of the learning model is reached. By this way, the main benefit of Federated Learning is that the user data privacy is protected because local training data are never uploaded to the cloud, thus decoupling the machine learning from acquiring, storing, and training data in datacenters as conventional approaches.

The key enablers to the Federated Learning are UEs, e.g., smart phones and tables, of which the local data is valuable for learning models, which in turn can greatly improve the user quality of experience. For example, location-based services that needs a massive buit-up database of user-location applications, such as the navigator app Waze, can help users to avoid heavy-traffic roads and thus reduce the congestion. Furthermore, the modern smart UEs can be considered as a personal computer with powerful integrated neural processors (e.g., Snapdragon 845 of Qualcomm) for heavy computing tasks, and plethora of sensors (e.g., cameras, microphones, and GPS) for collecting a wealth amount of data, which ensures the feasibility of Federated Learning to foster more intelligent applications. However, most of the users' data are privacy sensitive and large in nature so it is risky and unnecessary to log data to datacenters for the purpose of model training, which calls for the urgent needs of this technique development.

There are many reasons to support Federated Learning: First, the recent focus on the network edge computing for future applications such that the conventional centralized learning approach is not preferable for on-the-fly applications that require ultra low latency, such as flying a drone, controlling a self-driving car. To perform these delicate tasks, future wireless systems will need to make even more decisions at the network edge (closer to UEs), more quickly and more reliably, even when they lose connectivity. Second, Data Network Effects

However, with all the promising benefits, Federated Learning also comes with new performance metrics as well as challenges to tackle. In the nutshell, the concept of Federated Learning can be considered as the marriage of machine learning (prediction accuracy and data size) and distributed computing (computation capacity and convergence time) over wireless networks (bandwidth, uncertainty of channels, and transmission power). This complex marriage creates a new design in which *machine learning is shifted from datacenters to the network edge* where mobile UEs communicate their updates to train a global model, considering the *communication and computation latencies, link reliability, prediction accuracy, and data size, data privacy, and power constraints of UEs* altogether, which are not concerned by conventional approaches.

B. Contributions

Taking into account these challenges, we provide first-of-its-kind "Federated Learning over Wireless Networks" model design and analysis, which can be summarized as follows

- *Model design of Federated Learning over wireless networks.* We first propose the system model that stylizes the Federated Learning for wireless networks, in which each UE can allocate their computing power based on an optimized accuracy level, and then transmit their update for aggregation in a wireless time-sharing manner. We consider two extremes of objective interest in our model: i) the global learning model's main objective is minimizing the learning time convergence, and ii) the UEs, due to their power and bandwidth constraints, are concerned about their energy usage for update computation and aggregating communication. While minimizing learning time requires judiciously choosing the learning accuracy, the energy of UEs depends on computing and transmission power, which however affects to the learning time and thus creates the couplings between decisions on UEs. Especially, this model is more complicated when we consider the heterogeneity of UEs in terms of power capacities and energy cost coefficients, which affect the trade-off between computation and communication time of UEs.

N.H. Tran and Wei Bao are with the School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia (email: {nguyen.tran, wei.bao}@sydney.edu.au).

M.N.H. Nguyen is with the Department of Computer Science and Engineering, Kyung Hee University, Korea (email: minhnhn@khu.ac.kr).

- Characterize the solutions of problem: Despite non-convex nature of the problem, we exploit the special structure the the problem to decompose it into three sub-problems, correspondingly to communication phase, computation phase, and prediction accuracy phase. Our findings are ....

II. Related Works

Due to Big Data applications and complex models such as Deep Learning, training machine learning models needs to be distributed over multiple machines, giving rise to the researches on decentralized machine learning [?] (Reddi et al., 2016; Ma et al., 2017; Shamir et al., 2014; Zhang & Lin, 2015; Dean" (Konečný et al 2016:1)"et al., 2012; Chilimbi et al., 2014). However, most of the algorithms in these works are designed for machines with balanced and i.i.d. data and connected to high-throughput networks such as datacenters.

From a different motivation, Federated Learning (and related on-device intelligence approaches), which has attracted a lot of attention recently [?], [?], [?], [?], [?], [?], exploits the collaboration of mobile devices (or UEs) that can be large in number, slow and/or unstable in Internet connections, and have non-i.i.d. and unbalanced data locally. However, most of these works focused on designing algorithms to improve the convergence of learning time, unconcerned about other limiting factors of mobile UEs, such as *wireless communication and energy-limited* nature of mobile UEs that can affect the performance of FEDL. To fill this gap, we study how the computation of communication characteristics of UEs can affect to their energy consumption and the learning time convergence of FEDL, considering *heterogeneous* UEs in terms of data size, computing and transmission power capabilities.

Mobile edge computing, offloading

III. System Model

As in Fig. **??**, we consider a wireless multi-user system consisting of one base station (BS) and a set $\mathcal{N}$ of N users with UEs. Each participating UE n stores a local data set $\mathcal{D}_n$, with its size is denoted by D_n. Then, we can define the total data size by $D = \sum_{n=1}^{N} D_n$. In an example of the supervised learning setting, at UE n, $\mathcal{D}_n$ defines the collection of data samples given as a set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_n}$, where $x_i \in \mathbb{R}^d$ is an input sample vector with d features, and $y_i \in \mathbb{R}$ is the labeled output value for the sample x_i. The data can be generated through the usage of UE, for example, via interactions with apps. With these UEs' data, several machine learning applications can be employed for wireless networks such as predicting the BS's load in next hours for dynamic BS load balancing, or predicting the next hovering position of drones such that their coverage is optimized.

In a typical learning problem, for a sample data $\{x_i, y_i\}$ with input x_i (e.g., the response time of various apps inside the UE), the task is to find the *model parameter* w that characterizes the output y_i (e.g., label of BS load, such as high or low, in next hours) with the loss function $f_i(w)$. Some examples of the loss function are $f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2, y_i \in \mathbb{R}$ for linear regression and $f_i(w) = \{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$ for support vector machine. Then, the loss function on the data set of UE n is defined as

$$J_n(w) := \frac{1}{D_n} \sum\nolimits_{i \in \mathcal{D}_n} f_i(w). \tag{1}$$

Then the learning model is the minimizer of the following global loss function on all UEs' data

$$\min_{w \in \mathbb{R}^d} J(w) := \sum\nolimits_{n=1}^{N} \frac{D_n}{D} J_n(w) \tag{2}$$

A. Federated Learning over Wireless Networks

In this section, based on [?], we adapt the Federated Learning to the wireless networks, namely FEDL, as the following algorithm.

FEDL:

1) At the UE side, there are two phases:
   Computation. Each UE n solves its local problem
   $$w_n^{(t)} = \arg\min_{w_n \in \mathbb{R}^d} F_n\Big(w_n \mid w^{(t-1)}, \nabla J^{(t-1)}\Big) \tag{3}$$
   with a local accuracy $0 < \theta < 1$ (i.e., $||\nabla F_n(w^{(t)}|| \leq \theta\, ||\nabla F_n(w^{(t-1)}||)$).
   Communication. All UEs share the wireless environment to transmit $w_n^{(t)}$ and the gradient $\nabla J_n^{(t)}$ to BS.
2) At the BS, the following information is aggregated
   $$w^{(t+1)} = \frac{1}{N} \sum\nolimits_{n=1}^{N} w_n^{(t)} \tag{4}$$
   $$\nabla J^{(t+1)} = \frac{1}{N} \sum\nolimits_{n=1}^{N} \nabla J_n^{(t)} \tag{5}$$
   and feedback to all UEs. This process is iterative until a global accuracy $0 < \varepsilon < 1$ is achieved (i.e., $||\nabla J(w^{(t)}|| \leq \varepsilon ||\nabla J(w^{(t-1)}||)$).

We briefly summarize the algorithm FEDL. To solve problem (2), the Federated Learning uses an iterative approach that requires a number of *global iterations* to achieve an global accuracy level ε. In each global iteration, there are interactions between two sides of UEs and BS: A participating UE, in each computation phase, will minimize its objective $F_n(w_n)$ in (3) using local training data $\mathcal{D}_n$. Minimizing F_n also takes multiple *local iterations* up to an accuracy threshold θ, which is common to all UEs. The computation phase is *synchronous* such that all UEs have to finish solving their local problem before entering the communication phase to transmit their updates to the BS by using a wireless medium sharing scheme (e.g., time-sharing similar to TDMA).
The BS then aggregates the local model parameters and gradients, w_n and the ∇J_n, $\forall n$, respectively, to update and then broadcast the global model parameter and gradient, w and the ∇J according to (4) and (5), respectively, which are required for participating UEs to minimize their $F_n(w_n)$[1], in the next global iteration. We see that the BS does not access the local data $\mathcal{D}_n$, $\forall n$, hence preserving data privacy.

[1]One example, from [], is $F_n(w_n) = J_n(w_n) - (\nabla J_n^{(t-1)} - \beta_1 \nabla J^{(t-1)})^T w_n + \frac{\beta_2}{2} ||w_n - w_n^{(t-1)}||^2$ where $\beta_1, \beta_2 \geq 0$ are parameters.

For strongly convex objective $J(w)$, the general upper bound on global iterations is shown to be [?]

$$K(\varepsilon, \theta) = \frac{\mathcal{O}(\log(1/\epsilon))}{1-\theta} \tag{6}$$

which not only depends on global accuracy ε but also the local accuracy θ. For example, when ε and θ are small (more accurate), FEDLneeds to runs more global iterations. On the other hand, each global iteration consists of both computation and uplink communication time. Since the downlink bandwidth is larger than that of uplink and the BS power is much higher than UE's transmission power, the downlink time is negligible compared to uplink time and thus is not considered. *The computation time, however, depends on the number of local iterations, which is upper bounded by* $\mathcal{O}(\log(1/\theta))$, *for a wide range of iterative algorithms to solve* (3) *such as stochastic gradient descent [?]. We note that FEDLperformance does not depend on which algorithms used in computation phase (i.e., either gradient descent or others are fine) as long as the convergence time of that algorithm is upper-bounded by* $\mathcal{O}(\log(1/\theta))$. Denote the time of one local iteration by T_{cmp}, then the computation time in one global iteration is $v\log(1/\theta)T_{cmp}$ for some constant $v > 0$. Denoting the communication time in one global iteration by T_{com}, the total time of one global iteration of FEDLis defined as

$$T_{glob}(T_{cmp}, T_{com}, \theta) := T_{com} + v\log(1/\theta)T_{cmp}. \tag{7}$$

In this work, we consider a fixed global accuracy ϵ, so we normalize $\mathcal{O}(\log(1/\epsilon))$ to 1 so that $K(\theta) = \frac{1}{1-\theta}$ for ease of presentation. Furthermore, we also normalize v to 1 since we can absorb v into T_{cmp} as the upper bound of one local computation iteration. Thus the upper bound of FEDLlearning time is $K(\theta)\, T_{glob}(\theta)$.

In the next sections we will present how computation and communication time relate to UEs' energy consumption.

B. Computation Model

We denote the number of CPU cycles for UE n to execute one sample of data by c_n, which can be measured offline [?]. Since all samples $\{x_i, y_i\}_{i\in\mathcal{D}_n}$ have the same size, the number of CPU cycles required for UE n to run one local iteration is $c_n D_n$. Denote the CPU-cycle frequency of the UE n by f_n. Then the CPU energy consumption of UE n for one local iteration of computation can be expressed as follows [?]

$$E_n^{cmp}(f_n) = \sum\nolimits_{i=1}^{c_n D_n} \frac{\alpha_n}{2} f_n^2 = \frac{\alpha_n}{2} c_n D_n f_n^2, \tag{8}$$

where $\alpha_n/2$ is the effective capacitance coefficient of UE n's computing chipset. Furthermore, the computation time per local iteration of the UE n is $\frac{c_n D_n}{f_n}$, $\forall n$. We denote the vector of $f_n, \forall n$, by $f \in \mathbb{R}^n$.

C. Communication Model

In FEDL, regards to the communication phase of UEs, we consider a TDMA model for UEs' wireless medium sharing. We note that TDMA is not restrictive to FEDLmodel because other schemes, such as OFDMA, can be applied here without affecting the hindsights. The achievable transmission rate (bits/s) of UE n is defined as follows:

$$r_n = B\log_2\big(1 + \frac{h_n p_n}{N_0}\big), \tag{9}$$

where B is the bandwidth, N_0 is the background noise, p_n is the transmission power, and h_n is the channel gain of the UE n. We assume that h_n is constant during the learning time of FEDL. In Section V, we also treat the case of varying h_n due to random nature of wireless link. Denote the fraction of communication time T_{com}allocated to UE n by τ_n, and the size (in bits) of both w_n and ∇J_n by s_n. Because the dimensions of vectors w_n and ∇J_n are fixed, we assume that their sizes are constant during FEDLlearning time. Then the transmission rate of each UE n is

$$r_n = s_n/\tau_n, \tag{10}$$

which is shown to be the most energy-efficient transmission policy with deadline [?]. Thus, to transmit s_n within a time τ_n, the UE n's energy consumption is

$$E_n^{com}(\tau_n) = \tau_n p_n = \tau_n\, p_n(s_n/\tau_n), \tag{11}$$

where according to (9) and (10), the power function is as follows:

$$p_n(s_n/\tau_n) := \frac{N_0}{h_n}\big(2^{\frac{s_n/\tau_n}{B}} - 1\big). \tag{12}$$

We denote the vector of $\tau_n, \forall n$, by $\tau \in \mathbb{R}^n$.

D. Problem formulation

Define the total energy consumption of all UEs for each global iteration by E_{glob}, which is expressed as follows:

$$E_{glob}(f, \tau, \theta) := \sum\nolimits_{n=1}^{N} E_n^{com}(\tau_n) + \log(1/\theta) E_n^{cmp}(f_n). \tag{13}$$

Then, we consider the following problem

$$\textsf{OPT} : \min. \quad K(\theta)\big[E_{glob}(f, \tau, \theta) + \kappa\, T_{glob}(T_{cmp}, T_{com}, \theta)\big] \tag{14}$$

$$\text{s.t.} \quad \sum\nolimits_{n=1}^{N} \tau_n \le T_{com}, \tag{15}$$

$$\max_n \frac{c_n D_n}{f_n} \le T_{cmp},\ \forall n \in \mathcal{N}, \tag{16}$$

$$f_n^{min} \le f_n \le f_n^{max},\ \forall n \in \mathcal{N}, \tag{17}$$

$$p_n^{min} \le p_n(s_n/\tau_n) \le p_n^{max},\ \forall n \in \mathcal{N}, \tag{18}$$

$$0 < \theta < 1. \tag{19}$$

In FEDL, the goal is to minimize both UEs' energy consumption and total learning time, which, however, can be conflicted. For example, the UEs can save its energy by setting the lowest frequency level all the time, but this will certainly increase the learning time. Therefore, to strike the balance between energy cost and learning time, the weight κ (Joules/second), used in the objective of OPTas an amount of additional energy cost that FEDLis willing to bear for one unit of learning time to be reduced, captures the Pareto-optimal tradeoff between the UEs' energy cost and the total

learning time of FEDL. For example, when most of the UEs are plugged in, then UE energy cost is not the main concern, thus κ can be large.
While constraint (15) captures the time-sharing uplink transmission of UEs, constraint (16) says that the the computing time in one local iteration is determined by the "bottleneck" UE (e.g., with large local data size and low CPU frequency). The feasible ranges of CPU-frequency and transmit power of UEs are imposed by constraints (17) and (18), respectively. We note that (17) and (18) also capture the *heterogeneity* of UEs with different types of CPU and transmit chipsets. The last constraint restricts the feasible range of the local accuracy.

Furthermore, we assume that

$$\max_n \frac{c_n D_n}{f_n^{max}} \leq \min_n \frac{c_n D_n}{f_n^{min}}. \tag{20}$$

This assumption guarantees that the UEs' data size is not so much "uneven" so that any UE with its maximum CPU frequency still can finish the local computation faster than any UE with minimum CPU frequency.

IV. Solutions for OPT

We see that the objective of OPTis non-convex. However, in this section we will characterize its optimal solution by decomposing it into multiple sub-problems.

We consider the first case when θ is fixed, then problem OPTcan be decomposed into two sub-problems as follows:

$$\text{SUB1:}\quad \text{min.}\quad \sum_{n=1}^{N} E_n^{cmp}(f_n) + \kappa T_{cmp} \tag{21}$$

$$\text{s.t.}\quad \max_n \frac{c_n D_n}{f_n} \leq T_{cmp},\ \forall n \in \mathcal{N}, \tag{22}$$

$$f_n^{min} \leq f_n \leq f_n^{max},\ \forall n \in \mathcal{N}. \tag{23}$$

and

$$\text{SUB2: min.}\quad \sum_{n=1}^{N} E_n^{com}(\tau_n) + \kappa T_{com} \tag{24}$$

$$\text{s.t.}\quad \sum_{n=1}^{N} \tau_n \leq T_{com}, \tag{25}$$

$$p_n^{min} \leq p_n(s_n/\tau_n) \leq p_n^{max},\ \forall n \in \mathcal{N}. \tag{26}$$

While SUB1is a CPU-cycle control problem for the computation time and energy minimization, SUB2can be considered as an uplink power control to determine the UEs' fraction of time sharing to minimize the UEs energy and time of FEDL's communication phase. It can be observed that both SUB1and SUB2are **convex** problems. Thus, we next characterize the solutions of these two sub-problems using KKT conditions in the following.

Lemma 1. *The solution of SUB1is as follows*
a) If $\kappa \geq \sum_{n=1}^{N} \alpha_n (f_n^{max})^3$, then

$$T_{cmp}^* = T_{min} := \max_n \frac{c_n D_n}{f_n^{max}}, \forall n \tag{27}$$

$$f_n^* = f_n^{max}, \forall n : \frac{c_n D_n}{f_n^{max}} = T_{cmp}^* \tag{28}$$

$$f_n^* = \frac{c_n D_n}{T_{cmp}^*} = \frac{c_n D_n}{\max_n \frac{c_n D_n}{f_n^{max}}}, \forall n : \frac{c_n D_n}{f_n^{max}} < T_{cmp}^*. \tag{29}$$

Algorithm 1 Finding $\mathcal{N}'$ in Lemma 1, case c)

```
1: Sort UEs such that $\frac{c_1 D_1}{f_1^{min}} \geq \frac{c_2 D_2}{f_2^{min}} \ldots \geq \frac{c_N D_N}{f_N^{min}}$
2: $\mathcal{M} = \mathcal{N}, i = N$;
3: while $i > 0$ do
4:     $T_{cmp}^*(\mathcal{M}) = \left(\frac{\sum_{m \in \mathcal{M}} \alpha_m (c_m D_m)^3}{\kappa}\right)^{1/3}$
5:     if $\frac{c_i D_i}{f_i^{min}} < T_{cmp}^*(\mathcal{M})$, then $\mathcal{M} = \mathcal{M} - i$;
6:     end if
7:     $i = i - 1$
8: end while
9: $\mathcal{N}' = \mathcal{M}$.
```

b) If $\kappa \leq \sum_{n=1}^{N} \alpha_n (f_n^{min})^3$, then

$$T_{cmp}^* = T_{max} := \max_n \frac{c_n D_n}{f_n^{min}} \tag{30}$$

$$f_n^* = f_n^{min}, \forall n \tag{31}$$

c) If $\sum_{n=1}^{N} \alpha_n (f_n^{min})^3 < \kappa < \sum_{n=1}^{N} \alpha_n (f_n^{max})^3$, then there is a subset $\mathcal{N}' \subseteq \mathcal{N}$ such that $f_n^{min} < f_n < f_n^{max}$, $\forall n \in \mathcal{N}'$

$$T_{cmp}^* = \left(\frac{\sum_{n \in \mathcal{N}'} \alpha_n (c_n D_n)^3}{\kappa}\right)^{1/3} \tag{32}$$

$$f_n^* = \frac{c_n D_n}{T_{cmp}^*}, \forall n \in \mathcal{N}' \tag{33}$$

$$f_n^* = f_n^{min},\ \forall n \in \mathcal{N} - \mathcal{N}', \tag{34}$$

where $\mathcal{N} - \mathcal{N}' = \{n \,|\, \frac{c_n D_n}{f_n^{min}} < T_{cmp}^*\}$.

From this result, we see that there are three cases corresponding to the solution of problem SUB1.

a) **High κ for prioritized computation time minimization**: The optimal $T_{cmp}^*$is the minimum computation time T_{min} of FEDLachieved in (27), which is determined by the most "bottleneck" UEs (i.e., heavy-loaded data and/or low f_n^{max}) which runs their maximum CPU-cycle in (28). The other "non-bottleneck" UEs should adjust a "right" CPU-cycle as in (29) to save the energy yet still maintain their computing time the same as T_{min}.
b) **Low κ for prioritized computation energy minimization**: All UE runs their CPU at the lowest cycle frequency (31), thus $T_{cmp}^*$is the maximum computation time T_{max} of FEDLdetermined by the last UEs that finish their computation (30) with their minimum frequency.
c) **Medium κ for balancing computation time and energy minimization**: In this case, $T_{cmp}^*$is strictly inside $[T_{min}, T_{max}]$, determined in (32) by a set $\mathcal{N}'$ of UEs with their optimal CPU-cycle frequency (33) strictly inside the feasible set. Other UEs with light-loaded data can run at the most saving energy mode yet still finish their task before the optimal computation time of FEDLas in (34).

Finding $\mathcal{N}'$.

Before characterizing the solution of SUB2, from (12) and (26), we first define two bounded values for τ_n

$$\tau_n^{max} := \frac{s_n}{B\log_2(h_n N_0^{-1} p_n^{min} + 1)} \tag{35}$$

$$\tau_n^{min} := \frac{s_n}{B\log_2(h_n N_0^{-1} p_n^{max} + 1)}, \tag{36}$$

which are the maximum and minimum fraction of T_{com}that UE n can achieve by transmitting with its minimum and maximum power, respectively.

We also define a new function $g_n : \mathbb{R} \rightarrow \mathbb{R}$ as

$$g_n(\kappa) = \frac{s_n/B}{1 + W\Big(\frac{\kappa N_0^{-1} h_n - 1}{2} \ln 2\Big)/\ln 2}, \tag{37}$$

where $W(.)$ is the Lambert W-function. We can consider $g_n(.)$ as an indirect "power control" function that helps UE n determine the amount of time it should transmit s_n by adjust its power based on the weight κ. This function is strictly decreasing (thus its inverse function $g_n^{-1}(\cdot)$ exists) reflecting that when we put more priotity on minimizing the communication time (i.e., high κ), UE n should raise the power to finish its transmission with less time (i.e., low τ_n).

Lemma 2. *The solution of SUB2is as follows*
a) If $\kappa \geq g_n^{-1}(\tau_n^{min})$, then

$$\tau_n^* = \tau_n^{min} \tag{38}$$

b) If $\kappa \leq g_n^{-1}(\tau_n^{max})$, then

$$\tau_n^* = \tau_n^{max} \tag{39}$$

c) If $g_n^{-1}(\tau_n^{max}) < \kappa < g_n^{-1}(\tau_n^{min})$, then

$$\tau_n^{min} < \tau_n^* = g_n(\kappa) < \tau_n^{max}. \tag{40}$$

And $T_{com}^* = \sum_{n=1}^N \tau_n^*$.

This lemma can be explained in an economics fashion as follows. If we consider the FEDLsystem as the buyer and UEs as sellers with the UE powers as commodities, then the inverse function $g_n^{-1}(\cdot)$ is interpreted as the price of energy cost that UE n is willing to receive by providing power service for FEDLto reduce the learning time. There are two properties of this function: i) price increases with repect to its power, and ii) price sensitivity depends on UEs characteristics, e.g., UEs with better channel quality can have lower price while UEs with larger data size s_n will have higher price. Thus, each UE n will compare its energy price $g_n^{-1}(\cdot)$ with the "offer" price κ of the system to decide how much power it is willing to "sell". Then, there are three cases corresponding to the solution of problem SUB2.

a) **High offer**: If the offer price κ is higher than UE n maximum price request $g_n^{-1}(\tau_n^{min})$, UE n will sell its highest service by transmitting with the maximum power p_n^{max}.
b) **Low offer**: If the offer price κ is lower than UE n minimum price request $g_n^{-1}(\tau_n^{max})$, UE n will sell its lowest service by transmitting with the minimum power p_n^{max}.
c) **Medium offer**: If the offer price κ is within the range of UE n's acceptable price range, UE n will find a power level such that the corresponding energy price will match the offer price.

While SUB1and SUB2solutions share a similar threshold-based dependence on κ, we observe their differences as follows. In SUB1solution, the optimal CPU-cycle frequency of UE n depends on the optimal T_{cmp}^*, which in turn depends on the loads (i.e., $\frac{c_n D_n}{f_n}$, $\forall n$) of all UEs. Thus all UE load information are required for the computation phase. On the other hand, in SUB2solution, each UE n can independently choose its optimal power by comparing its price function $g_n^{-1}(\cdot)$ with κ so that collecting UE information is not needed. The reason is that the synchronization of computation time in constraint (22) of SUB1requires all UE loads, whereas the UEs' time-sharing constraint (25) of SUB2can be decoupled by comparing with the fixed "offer" price κ.

Next, we observe that the solutions of SUB1and SUB2have no dependence on θ so that the optimal T_{com}^*, T_{cmp}^*, f^*, and τ^* can be determined based on κ according to Lemmas 1 and 2. Thus we consider the third sub-problem of OPTand its solution in what follows.

SUB3

$$\begin{aligned} \text{min.} \quad & K(\theta)\big[E_{glob}(f^*, \tau^*, \theta) + \kappa\, T_{glob}(T_{cmp}^*, T_{com}^*, \theta)\big] \\ \text{s.t.} \quad & 0 < \theta < 1. \end{aligned} \tag{41}$$

Lemma 3. *There exists a unique solution θ^* of problem SUB3satisfying the following equation:*

$$\frac{1}{\eta} = \log\big(e^{1/\theta^*}\theta^*\big) \tag{42}$$

where

$$\eta := \frac{\sum_{n=1}^N E_n^{cmp}(f_n^*) + \kappa T_{cmp}^*}{\sum_{n=1}^N \big[E_n^{cmp}(f_n^*) + E_n^{com}(\tau_n^*)\big] + \kappa\big[T_{cmp}^* + T_{com}^*\big]}. \tag{43}$$

From its definition η is the fraction of total computation cost (including all UE energy and computing time) in one local iteration of the computation phase over the aggregated communication and computation costs in each global iteration. We then have some observations from Lemma 3. First, according to (42), it can be shown that $0 < \theta^* < \eta < 1$. Thus, if computation cost is small compared to communication cost (e.g., $\eta < 0.1$), then θ^* is even smaller so UEs have to run a large number of local iterations in computation phase. This way can reduce the number of global iterations $K(\theta^*)$ due to the expensive communication cost.

We need to have a figure for Lemma 3 illustration.

Theorem 1. *The global optimal solution of OPTis the combined solutions of three sub-problems SUB1, SUB2, and SUB3.*

The proof of this theorem is straighforward. The idea is to use the KKT condition to find the stationary points of OPT. Then we can decompose the KKT condition equations into three groups, each of them matches exactly to the KKT condition of SUB1, SUB2, and SUB3, which can be solved for unique closed-form solution as in Lemma 1, 2, and 3, respectively. Thus this unique stationary point is also the global optimal solution of OPT.

We then have some discussions about the combined solutions of OPT. First, we see that SUB1and SUB2solutions can be characterized independently, which can be explained that each UE often has two separate processors: one CPU for mobile applications and another baseband processor for radio control function. Second, SUB1and SUB2do not depend on θ. While the communication phase of SUB2is clearly not affected by the local accuracy parameter, SUB2consider the computation cost in one local iteration. However, the solutions of SUB1and SUB2, which reveal how much communication cost is more expensive than computation cost, are decisive factors to determine the optimal level of local accuracy. Therefore, we can sequentially solve SUB1and SUB2first, then SUB3to achieve the optimal solutions of OPT.

V. Extenstions

1) Define

$$\kappa_{min} = \min\Big\{g_n^{-1}(\tau_n^{max}), \sum\nolimits_{n=1}^{N} \alpha_n (f_n^{min})^3\Big\} \quad (44)$$

$$\kappa_{max} = \max\Big\{g_n^{-1}(\tau_n^{min}), \sum\nolimits_{n=1}^{N} \alpha_n (f_n^{max})^3\Big\} \quad (45)$$

$$(46)$$

The Golden search algorithm can find good κ^*.

2) Consider the outage probability when channel is varying.

VI. Case Studies

VII. Future works

When UEs don't have enough computing power, it can offload the data and learning task, but with differential privacy concerning. Some scenarios, due to the UE's limited resources, a machine learning algorithm needs to run simultaneously on the phone and remotely in the network. Doing so harnesses both the individual intelligence (on-UE AI) and the collective intelligence (cloud AI) and allows the UE to tap the abundant storage and computing power of the network for better and faster inference [**?**].

Another important challenge for enabling on-UE AI pertains to system design. While classical machine learning is centered on maximizing the average reward (or average cost function) for every agent, on-UE AI is more prone to uncertainty and randomness due to limited access to training data, unreliable links between UEs, and the latency that is added when a UE offloads a task to the cloud or its peers.

This issue of local and remote computing is referred to as task offloading, where a task is carried out locally on the UE, remotely in the network, or both. Finding the best strategy which takes into account the application's requirements, neural learning model, power usage, and network congestion is a fundamental problem that engineers are still working to solv

How to choose a subset of UEs that can perform?

When the UEs are strategic, how to incentivize them to participate?

"If additional privacy is needed, randomization techniques from dierential privacy can be used. The centralized algorithm could be modied to produce a dierentially private model [17, 33, 1], which allows the model to be released while protecting the privacy of the individuals contributing updates to the training process. If protection from even a malicious (or compromised) coordinating server is needed, techniques from local dierential privacy can be applied to privatize the individual updates [32]. Details of this are beyond the scope of the current work, but it is a promising direction for future research" (Konečný et al 2016:2)

VIII. Conclusions

We studied the

Appendix

A. Proof of Lemma 1

The Lagrangian of SUB1is

$$\mathcal{L} = \sum\nolimits_{n=1}^{N} \Big[E_n^{cmp}(f_n) + \lambda_n(\frac{c_n D_n}{f_n} - T_{cmp}) + \mu_n(f_n - f_n^{max}) - \nu_n(f_n - f_n^{min})\Big] + \kappa \log(1/\theta) T_{cmp} \quad (47)$$

where λ_n, μ_n, ν_n are non-negative dual variables. Then the KKT condition is as follows:

$$\frac{\partial \mathcal{L}}{\partial f_n} = \frac{\partial E_n^{cmp}}{\partial \tau_n} - \lambda_n \frac{c_n D_n}{f_n^2} + \mu_n - \nu_n = 0,\ \forall n \quad (48)$$

$$\frac{\partial \mathcal{L}}{\partial T_{cmp}} = \kappa - \sum\nolimits_{n=1}^{N} \lambda_n = 0, \quad (49)$$

$$\mu_n(f_n - f_n^{max}) = 0,\ \forall n \quad (50)$$

$$\nu_n(f_n - f_n^{min}) = 0,\ \forall n \quad (51)$$

$$\lambda_n(\frac{c_n D_n}{f_n} - T_{cmp}) = 0.\ \forall n \quad (52)$$

From (49), we have

$$\kappa = \sum\nolimits_{n=1}^{N} \lambda_n^*. \quad (53)$$

a) If $\kappa \geq \sum_{n=1}^{N} \alpha_n (f_n^{max})^3$, then there exists a non-empty set $\mathcal{N}' = \{n | \lambda_n^* \geq \alpha_n (f_n^{max})^3\}$ such that

$$\frac{\partial E_n^{cmp}(f_n^*)}{\partial f_n} - \lambda_n^* \frac{c_n D_n}{{f_n^*}^2} \leq 0, \forall n \in \mathcal{N}' : f_n^* \leq f_n^{max}. \quad (54)$$

Then from (48) we must have $\mu_n^* - \nu_n^* \geq 0, \forall n \in \mathcal{N}'$. Thus, from (50) we must have $f_n^* = f_n^{max}\ \forall n \in \mathcal{N}'$. By definition $T_{cmp}^* = \max_n \frac{c_n D_n}{f_n^{max}}$, and we see that $\mathcal{N}' = \{n : \frac{c_n D_n}{f_n} = T_{cmp}^*\}$.
Also, $\lambda_n^* > 0, \forall n$. Otherwise, if there is an n with $\lambda_n^* = 0$ from (48) we must have $\mu_n^* - \nu_n^* < 0$ then $f_n^* = f_n^{min}$, a contradiction with $\frac{c_n D_n}{f_n^{min}} \leq T_{cmp}^* = \max_n \frac{c_n D_n}{f_n^{max}}$ due to Assumption (20).
On the other hand, for the remaining UEs belong to the set $\mathcal{N} - \mathcal{N}' = \{n | \lambda_n^* < \alpha_n (f_n^{max})^3\}$. We see that

$$\frac{c_n D_n}{f_n^{max}} < T_{cmp}^*,\ \forall n \in \mathcal{N} - \mathcal{N}'. \quad (55)$$

Since T^*_{cmp} must be the same for all n with $\lambda^*_n > 0$ according to (52), we obtain

$$f^*_n = \frac{c_n D_n}{T^*_{cmp}} = \frac{c_n D_n}{\max_n \frac{c_n D_n}{f_n^{max}}}, \ \forall n \in \mathcal{N} - \mathcal{N}', \tag{56}$$

$$\lambda^*_n = \alpha_n \Big(\frac{c_n D_n}{\max_n \frac{c_n D_n}{f_n^{max}}} \Big)^3 < \alpha_n f_n^{max 3}, \ \forall n \in \mathcal{N} - \mathcal{N}'. \tag{57}$$

b) If $\kappa \leq \sum_{n=1}^N \alpha_n (f_n^{min})^3$, then there exists a non-empty set $\mathcal{N}'' = \{n | \lambda^*_n \leq \alpha_n (f_n^{min})^3\}$, then we have

$$\frac{\partial E_n^{cmp}(f^*_n)}{\partial f_n} - \lambda^*_n \frac{c_n D_n}{{f^*_n}^2} \geq 0, \forall n \in \mathcal{N}'' : f^*_n \geq f_n^{min}. \tag{58}$$

Then from (48) we must have $\mu^*_n - \nu^*_n \leq 0, \forall n \in \mathcal{N}''$. Thus, from (50) we must have $f^*_n = f_n^{min}$, and $T^*_{cmp} = \max_n \frac{c_n D_n}{f_n^{min}}, \ \forall n \in \mathcal{N}''$. For all n with $\frac{c_n D_n}{f_n^{min}} < T^*_{cmp}$, from (52), $\lambda^*_n = 0$, thus from (48) we must have $\mu^*_n - \nu^*_n < 0 \implies f^*_n = f_n^{min}$, which can be explained that increasing its CPU frequency more than f_n^{min} will increase its energy cost. Therefore, we have $f^*_n = f_n^{min}, \ \forall n \in \mathcal{N}$.

c) If $\sum_{n=1}^N \alpha_n (f_n^{min})^3 < \kappa < \sum_{n=1}^N \alpha_n (f_n^{max})^3$, then there exists a non-empty set $\mathcal{N}' = \{n | \alpha_n (f_n^{min})^3 < \lambda^*_n < \alpha_n (f_n^{max})^3\}$, then the following equation

$$\frac{\partial E_n^{cmp}(f_n)}{\partial f_n} - \lambda^*_n \frac{c_n D_n}{{f_n}^2} = 0, \forall n \in \mathcal{N}'. \tag{59}$$

has its root $f'_n = \Big(\frac{\lambda^*_n}{\alpha_n}\Big)^{1/3}$ such that

$$f_n^{min} < f'_n < f_n^{max}. \tag{60}$$

Then from (50) and (51), we must have $\mu^*_n = \nu^*_n = 0$, with which and (48) we have $f^*_n = f'_n, \forall n \in \mathcal{N}'$.

Now we analyze T^*_{cmp}. Since $\lambda^*_n > 0$,

$$T^*_{cmp} = \frac{c_n D_n}{f^*_n} = c_n D_n \Big(\frac{\alpha_n}{\lambda^*_n}\Big)^{1/3}, \forall n \in \mathcal{N}'. \tag{61}$$

Combining the above equation with (53), we have

$$T^*_{cmp} = \Bigg(\frac{\sum_{n \in \mathcal{N}'} \alpha_n (c_n D_n)^3}{\kappa} \Bigg)^{1/3} \tag{62}$$

$$f^*_n = \frac{c_n D_n}{T_{cmp}}, \forall n \in \mathcal{N}'. \tag{63}$$

On the other hand, we have $f^*_n = f_n^{min}$, for the set of n such that $\lambda^*_n \leq \alpha_n f_n^{min}$ by previous analysis, in which either $\lambda^*_n = 0$ when $\frac{c_n D_n}{f_n^{min}} < T^*_{cmp}$ or $\lambda^*_n = \alpha_n (f_n^{max})^3$ when $\frac{c_n D_n}{f_n^{min}} = T^*_{cmp}$.

Finally, we have $f^*_n = f_n^{max}$ for the set of n such that $\lambda^*_n \geq \alpha_n (f_n^{max})^3$ by previous analysis, in which $\lambda^*_n = \alpha_n (f_n^{max})^3$ when $\frac{c_n D_n}{f_n^{max}} = T^*_{cmp}$. We note that there exists no UE having $\lambda^*_n > \alpha_n (f_n^{max})^3$, because either $\frac{c_n D_n}{f_n^{max}} < T^*_{cmp}$ or $\frac{c_n D_n}{f_n^{max}} > T^*_{cmp}$ leads to contradiction.

B. Proof of Lemma 2

The Lagrangian of SUB2is

$$\mathcal{L} = \sum\nolimits_{n=1}^N E_n^{com}(\tau_n) + \lambda(\sum\nolimits_{n=1}^N \tau_n - T_{com}) \tag{64}$$
$$+ \sum\nolimits_{n=1}^N \mu_n(\tau_n - \tau_n^{max}) - \sum\nolimits_{n=1}^N \nu_n(\tau_n - \tau_n^{min}) + \kappa T_{com}$$

where λ, μ_n, ν_n are non-negative dual variables. Then the KKT condition is as follows:

$$\frac{\partial \mathcal{L}}{\partial \tau_n} = \frac{\partial E_n^{com}}{\partial \tau_n} + \lambda + \mu_n - \nu_n = 0, \ \forall n \tag{65}$$

$$\frac{\partial \mathcal{L}}{\partial T_{com}} = \kappa - \lambda = 0, \tag{66}$$

$$\mu_n(\tau_n - \tau_n^{max}) = 0, \ \forall n \tag{67}$$

$$\nu_n(\tau_n - \tau_n^{min}) = 0, \ \forall n \tag{68}$$

$$\lambda(\sum\nolimits_{n=1}^N \tau_n - T_{com}) = 0. \tag{69}$$

From (66), we see that $\lambda^* = \kappa$. Let $x := \frac{s_n}{\tau_n B}$, we first consider the equation

$$\frac{\partial E_n^{com}}{\partial \tau_n} + \lambda^* = 0 \tag{70}$$

$$\Leftrightarrow \frac{N_0}{h_n}\big(2^x - 1 - x 2^x\big) = -\lambda^* = -\kappa \tag{71}$$

$$\Leftrightarrow 2^x(x-1) = \kappa N_0^{-1} h_n - 1 \tag{72}$$

$$\Leftrightarrow 2^{x-1}(x-1) = \frac{\kappa N_0^{-1} h_n - 1}{2} \tag{73}$$

$$\Leftrightarrow x = 1 + \frac{W\big(\frac{\kappa N_0^{-1} h_n - 1}{2} \ln 2\big)}{\ln 2} \tag{74}$$

$$\Leftrightarrow \tau_n = g_n(\kappa) = \frac{s_n/B}{1 + \frac{W\big(\frac{\kappa N_0^{-1} h_n - 1}{2} \ln 2\big)}{\ln 2}}. \tag{75}$$

Because $W(.)$ is strictly increasing when $W(.) > -\ln 2$, $g_n(\kappa)$ is strictly decreasing and positive, and so is its inverse function

$$g_n^{-1}(\tau_n) = -\frac{\partial E_n^{com}(\tau_n)}{\partial \tau_n}. \tag{76}$$

Then we have

a) If $g_n(\kappa) \leq \tau_n^{min} \Leftrightarrow \kappa \geq g_n^{-1}(\tau_n^{min})$, then we have

$$\kappa = \lambda^* \geq g_n^{-1}(\tau_n^{min}) \geq -\frac{\partial E_n^{com}}{\partial \tau_n}\Big|_{\tau_n^{min} \leq \tau_n}. \tag{77}$$

Thus, according to (65), $\mu^*_n - \nu^*_n \leq 0$. Because both μ^*_n and ν^*_n cannot be positive, we have $\mu^*_n = 0$ and $\nu^*_n \geq 0$. Then we consider two cases of ν^*: a) $\nu^*_n > 0$, from (68), $\tau^*_n = \tau_n^{min}$, and b) $\nu^*_n = 0$, from (65), we must have $\kappa = g_n^{-1}(\tau_n^{min})$, and thus $\tau^*_n = \tau_n^{min}$.

b) If $g_n(\kappa) \geq \tau_n^{max} \Leftrightarrow \kappa \leq g_n^{-1}(\tau_n^{max})$, then we have

$$\kappa = \lambda^* \leq g_n^{-1}(\tau_n^{max}) \leq -\frac{\partial E_n^{com}}{\partial \tau_n}\Big|_{\tau_n \leq \tau_n^{max}}. \tag{78}$$

Thus, according to (65), $\mu^*_n - \nu^*_n \geq 0$, inducing $\nu^*_n = 0$ and $\mu^*_n \geq 0$. With similar reasoning as above, we have $\tau^*_n = \tau_n^{max}$.

c) If $\tau_n^{min} < g_n(\kappa) < \tau_n^{max} \Leftrightarrow g_n^{-1}(\tau_n^{max}) < \kappa < g_n^{-1}(\tau_n^{min})$, then from (67) and (68), we must have $\mu_n^* = \nu_n^* = 0$, with which and (65) we have

$$\tau_n^* = g_n(\kappa). \tag{79}$$

Finally, with $\lambda^* = \kappa > 0$, from (69) we have $T_{com}^* = \sum_{n=1}^{N} \tau_n^*$.

C. *Proof of Lemma 3*

We consider the first and second derivatives of objective of problem SUB3, denoted by $C(\theta)$, as follows

$$\frac{dC}{d\theta} = \frac{A}{(1-\theta)^2} + \frac{B}{(1-\theta)^2}\log(1/\theta) - \frac{B}{(1-\theta)\theta} \tag{80}$$

$$\frac{d^2C}{d\theta^2} = \frac{2A\theta^2 + 2B\theta^2\log(1/\theta) + B(1-\theta^2)}{(1-\theta)^3\theta^2}, \tag{81}$$

where $A = \sum_{n=1}^{N} E_n^{com}(\tau_n) + \kappa T_{com}$ and $B = \sum_{n=1}^{N} E_n^{cmp}(f_n) + \kappa T_{cmp}$. Since $\frac{d^2C}{d\theta^2} > 0$, $\forall \theta \in (0,1)$, $C(\theta)$ is a strictly convex function and its unique solution can be found as follow

$$\frac{dC}{d\theta} = 0 \Leftrightarrow \frac{A}{B} + 1 = 1/\theta - \log(1/\theta) \tag{82}$$

$$\Leftrightarrow 1/\eta = \log\big(e^{1/\theta}\theta\big) \tag{83}$$

References