Anarchic Convex Federated Learning

Dongsheng Li, Xiaowen Gong
Department of Electrical and Computer Engineering
Auburn University
Auburn, AL 36849, USA
{dzl0093,xgong}@auburn.edu

***Abstract*—The rapid advances in federated learning (FL) in the past few years have recently inspired a great deal of research on this emerging topic. Existing work on FL often assume that clients participate in the learning process with some particular pattern (such as balanced participation), and/or in a synchronous manner, and/or with the same number of local iterations, while these assumptions can be hard to hold in practice. In this paper, we propose AFLC, an Anarchic Federated Learning algorithm for Convex learning problems, which gives maximum freedom to clients. In particular, AFLC allows clients to 1) participate in arbitrary rounds; 2) participate asynchronously; 3) participate with arbitrary numbers of local iterations. The proposed AFLC algorithm enables clients to participate in FL efficiently and flexibly according to their needs, e.g., based on their heterogeneous and time-varying computation and communication capabilities. We characterize performance bounds on the learning loss of AFLC as a function of clients' local model delays and local iteration numbers. Our results show that the convergence error can be made arbitrarily small by choosing appropriate learning rates, and the convergence rate matches that of existing benchmarks. The results also characterize the impacts of clients' various parameters on the learning loss, which provide useful insights. Numerical results demonstrate the efficiency of the proposed algorithm.**

I. Introduction

As an emerging paradigm of machine learning (ML), *federated learning* (FL) carries out model training in a distributed manner [1]: Instead of collecting data from a possibly large number of devices to a central server in the cloud for training, FL trains a global ML model by aggregating local ML models computed distributedly across edge devices based on their local data. One significant advantage of FL is to preserve the privacy of individual devices' data. Moreover, since only local ML models rather than local data are sent to the server, the communication costs can be greatly reduced. Furthermore, FL can exploit substantial computation capabilities of ubiquitous smart devices.

In order to fully realize the potential of FL, several challenges need to be addressed due to salient features of FL. First, FL implies heterogeneous local data across clients, so that local models computed by clients from their local data can be diverse. Therefore, it is difficult for clients' local models to achieve convergence. Second, in contrast to conventional distributed ML where nodes typically communicate after every local computation iteration, clients in FL can perform multiple local iterations of computation before communicating their local models. While this feature can reduce communication costs of FL, it may slow down the convergence of the global model due to local model drifts (which is observed in many prior works such as [2]–[4]).

Besides data heterogeneity and multiple local iterations, FL also faces some challenges due to heterogeneous and time-varying computation and communication capabilities of clients' devices. First of all, clients may not be able to participate in every round of the entire learning process. This is particularly the case for cross-device FL where many clients only have resource-constrained mobile devices which are sometimes not available to perform local computations and/or communications with the FL server. Moreover, due to heterogeneity in computation and communication capabilities, even a client is able to participate in learning, it may be impossible or inefficient for all clients to complete their local computations and also communications of their local models in every round of the learning process in a synchronous manner. As a result, clients may need to compute and communicate their local models asynchronously. Furthermore, even clients can participate in FL synchronously in a round, they may perform different numbers of local iterations of computation, based on their computation capabilities. Such heterogeneous computation configuration can improve the efficiency of clients in FL, especially when there are stragglers. However, existing work on FL only considered some of the issues discussed above, but not all the issues at the same time.

In this paper, we explore Anarchic Federated Learning (AFL) which addresses all the challenges of FL as discussed above, by imposing minimum control on how clients participate in FL. In particular, AFL allows clients to 1) participate in arbitrary rounds; 2) participate asynchronously; 3) participate with arbitrary numbers of local iterations. By giving maximum freedom to clients, AFLC enables clients to participate in FL efficiently and flexibly according to their needs, e.g., based on their heterogeneous and time-varying computation and communication capabilities. We focus on AFL for convex learning problems (i.e., the objective function is convex), which has not been studied before for AFL. The convex setting of FL is of great importance: Although cutting-edge learning problems (such as deep learning) are typically non-convex in general, they are often convex locally and thus can be well approximated by convex problems. Although AFL for non-convex learning problems has been studied very recently in [5],

The work of Dongsheng Li and Xiaowen Gong were supported by NSF CAREER grant No. 2145031.

978-1-6654-9427-4/23/$31.00 ©2023 IEEE

there are some non-trivial differences for the convex setting that need to be addressed. In particular, the learning accuracy in the convex setting is quantified by the normed distance between the model found by the algorithm and the optimal model, which is quite different from that in the non-convex setting (which is the normed gradient). As a result, the major techniques used in the convergence analysis for the convex setting are significantly different from those in [5], e.g., the convexity of the learning problem needs to be utilized.

The main contributions of this paper are summarized as follows:

- We propose AFLC, an Anarchic Federated Learning algorithm for Convex learning problems, which allows clients to 1) participate in arbitrary rounds; 2) participate asynchronously; 3) participate with arbitrary numbers of local iterations. The proposed AFLC algorithm enables clients to participate in FL efficiently and flexibly according to their needs, e.g., based on their heterogeneous and time-varying computation and communication capabilities. One key idea in the algorithm design of AFLC is to use the most recent local model of a client to update the global model in a round, if the client does not participate or have not completed its local computation and/or communication in that round.
- We conduct convergence analysis for the AFLC algorithm by characterizing performance bounds on the learning loss as a function of clients' local model delays, and local iteration numbers. Our results show that the convergence error can be made arbitrarily small by choosing appropriate learning rates, and the convergence rate matches that of existing benchmarks. The results also characterize the impacts of clients' parameters on the learning loss, which provide useful insights.
- We evaluate the performance of the proposed AFLC algorithms by conducting numerical experiments for FL benchmarks. The experimental results demonstrate the efficiency of the proposed algorithms.

The remainder of this paper is organized as follows. Section II reviews related work. In Section III, we propose a anarchic federated learning algorithm for convex learning problems. In Section IV, we analyze the convergence of the proposed AFLC algorithm. Numerical results based on experiments are provided in Section VI.

II. Related Work

FL has emerged as a disruptive computing paradigm for ML by democratizing the learning process to potentially many individual users using their end devices [1], [6]–[9]. The past few years have seen tremendous research on FL. In the following, we discuss recent work on FL from three different aspects that are mostly related to this paper.

Federated Learning with Partial Client Participation. One major challenge for FL is that clients may not always participate throughout the entire learning process. This is especially true for cross-device FL where many clients have resource-constrained mobile devices which are sometimes not possible or too costly to perform local computations and/or communicate local/global models with the server. Many recent works [5], [8]–[10] studied FL where only some of all clients participate in learning in a round. Most of these studies [8], [10] assumed that clients' participation is balanced (e.g., the set of participating clients are randomly selected from all clients), such that each client has the same probability of participation. Under this assumption, it has been shown that FL algorithms can achieve a vanishing convergence error. However, in the general case where clients' participation can be arbitrary, there is a non-vanishing convergence error due to the worst-case client participation. This paper not only considers arbitrary client participation, but also asynchronous participation and heterogeneous local iteration numbers of clients.

Asynchronous Federated Learning. Many existing work [8]–[10] on FL studied synchronous algorithms where participating clients perform local computations and exchange local models in the same round (note that synchronous FL can also have partial client participation). However, synchronous algorithms can be inefficient as some clients may have to wait for other clients to complete their computations and/or communications, especially when there are stragglers due to heterogeneous computation and communication capabilities of clients. In this case, asynchronous algorithms [5], [11] are more efficient where a client can start its local computations in one round while completing the communication of its local model in another round. In this paper, besides asynchronous learning, we also consider arbitrary client participation and heterogeneous local iteration numbers.

Federated Learning with Heterogeneous Computations. One salient feature of FL is that clients can have heterogeneous computation capabilities. As a result, it is more efficient and flexible to allow clients to use different computation configurations. Some existing work on FL [5], [12] considered clients who use different mini-batch sizes, different local iteration numbers, and/or different learning model structures, etc. This paper considers clients with different local iteration numbers as well as arbitrary client participation and asynchronous algorithms.

III. Anarchic Federated Learning for Convex Learning Problem

In this section, we first present the system setting and the problem formulation of the FL system we consider. Then we describe AFLC, an Anarchic Federated Learning algorithm for Convex learning problems.

A. System Setting and Problem Formulation

Consider a FL system with an FL server and N clients in set $\mathcal{N}$ who collaboratively train a ML model with distributed local data in an asynchronous manner. The goal of the FL system is to minimize the training loss, which is given by the following optimization problem:

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \triangleq \sum_{k\in\mathcal{N}} p_k F_k(\boldsymbol{w}),$$

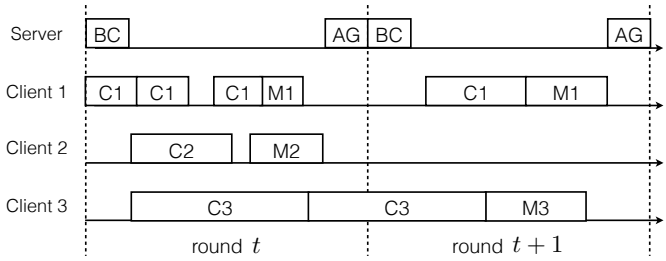

Fig. 1. Schedule of AFLC: BC is global model broadcast, AG is local model aggregation, each C block represents a local Computation iteration, each M block represents a local model coMmunication.

where $F(\boldsymbol{w})$ is the global loss function, $\boldsymbol{w}$ is the model parameter, $F_k(\boldsymbol{w})$ is the local loss function determined by client k's local dataset, p_k is the coefficient of client k's local loss function, and $\sum_{k\in\mathcal{N}} p_k = 1$.

Let τ_t^k be the delay of the client k's most recent local model received at the server at the end of round t (i.e., the difference between t and the index of the round when client k received the global model from the server to compute this most recent local model). If client k receives the global model $\boldsymbol{w}_{t-1}$ in round t, and then completes its local computations and communication of its local model $\boldsymbol{w}_{t-1}^k$ to the server in round t, then $\tau_t^k = 1$ (e.g., the delay of client 1 in round $t+1$ in Fig 1 is $\tau_{t+1}^1 = (t+1) - t = 1$); otherwise, $\tau_t^k > 1$ (e.g., in Fig 1, the delays of clients 2 and 3 in round $t+1$ are both $\tau_{t+1}^2 = \tau_{t+1}^3 = (t+1) - (t-1) = 2$).

After receiving the global model from the server, each client uses the global model to perform one or multiple local iterations of computation, each of which is given by

$$\boldsymbol{w}_{t,h+1}^k \triangleq \boldsymbol{w}_{t,h}^k - \eta_{t,h} \nabla F_k(\boldsymbol{w}_{t,h}^k, \xi_{t,h}^k),\ h = 0, 1, ..., H_t^k - 1,$$

where h is the local iteration index, $\xi_{t,h}^k$ is a sample uniformly chosen from the client k's local dataset. Let H_t^k be the maximum number of local iterations of computation for client k based on the global model it received in round t.

We use $\Gamma = \sum_{k\in\mathcal{N}} p_k (F^* - F_k^*)$ to quantify the non independent and identically distributed (non-IID) degree of the local data among all clients [8]. If $\Gamma = 0$, then the local data are IID, otherwise, they are non-IID case. The larger Γ is, the higher non-IID degree is. In addition, we do not allow a client to never update her model to the server, which means there exists a maximum delay constraint.

B. Algorithm Design of AFLC

We propose the AFLC algorithm as described in Algorithm 1. The schedule of the proposed AFLC algorithm is described in Fig 1. We consider two types of clients in each round t: 1) *updating clients* $\mathcal{N}_t^1$ who have completed their local computations and also communications of their local models to the server in round t, so that their local models are used to update the global model in round t (e.g., clients 1 and 2 for round t, and clients 1 and 3 for round $t+1$ in Fig 1); 2) *non-updating clients* $\mathcal{N}_t^0$ who do not participate (i.e., do not perform any local computation or any communication) in round t (e.g., client 2 for round $t+1$ in Fig 1), or who participate but have not completed their local computations or communications of their local models to the server in round t (e.g., client 3 for round t in Fig 1). Note that clients' local computations and communications with the server are allowed to be *intermittent and spanning multiple rounds* (such as clients 1, 2, and 3 in Fig 1), which accommodates the time-varying computation and communication capabilities of clients.

Algorithm 1 Anarchic Federated Learning for Convex Learning Problems (AFLC)

For each client $k \in \mathcal{N}$**:**
If starting to participate, receive the latest global model $\boldsymbol{w}_\mu$ broadcast by the server with timestamp (round index) μ, and set $\boldsymbol{w}_{\mu,0}^k = \boldsymbol{w}_\mu$;
for $h = 0$ to $H_\mu^k - 1$ **do**
  $\boldsymbol{w}_{\mu,h+1}^k \triangleq \boldsymbol{w}_{\mu,h}^k - \eta_{\mu,h} \nabla F_k(\boldsymbol{w}_{\mu,h}^k, \xi_{\mu,h}^k)$;
end for
Sum local stochastic gradients as $\boldsymbol{G}_\mu^k = \sum_{h=0}^{H_\mu^k - 1} \nabla F_k(\boldsymbol{w}_{\mu,h}^k, \xi_{\mu,h}^k)$;
Send the local model $\boldsymbol{G}_\mu^k$ to the server;
For the server, in each round t**:**
Broadcast the latest global model $\boldsymbol{w}_{t-1}$ to all clients;
Receive the local model $\boldsymbol{w}_{t-\tau_t^k}^k$ from each updating client $k \in \mathcal{N}_t^1$;
Retrieve the most recent local model $\boldsymbol{w}_{t-\tau_t^l}^l$ from the server's memory for each non-updating client $l \in \mathcal{N}_t^0$;
Update the global model as $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_{t+1} \left(\sum_{k\in\mathcal{N}_t^1} p_k \boldsymbol{G}_{t-\tau_t^k}^k + \sum_{l\in\mathcal{N}_t^0} p_l \boldsymbol{G}_{t-\tau_t^l}^l \right)$;

The server is responsible for local model aggregation. In each round, the server first broadcasts the latest global model to all clients (only clients who start to participate in this round receive this global model). Then the server receives local models from updating clients, and retrieves the most recent local models of non-updating clients from the server's memory. These most recent local models have been kept by the server since they were last updated by the server, when the non-updating clients in this round last communicated their local models to the server. Finally, the server aggregates the local models from both updating clients and non-updating clients, to update the global model.

IV. Convergence Analysis of AFLC

In this section, we first make some assumptions on the function F_1, F_2, ..., F_N. Then we provide the theoretical analysis of the proposed algorithm and some insights. Due to space limitation, all the proofs of results in this paper are provided in our online technical report [13].

Assumption 1: (*L-Smoothness*). Each local objective function is L-smooth, that is, $\forall x, y$

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2} ||y - x||^2.$$

Assumption 2: (*μ-Strongly Convex*). Each local objective function is μ-strongly convex, that is, $\forall x, y$

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}||y - x||^2.$$

Assumption 3: (*Bounded Gradient Variance*). The stochastic gradient at each client is an unbiased estimator of the local gradient: $E[f_i(\boldsymbol{w}_t^k,\ \xi_t^k)] = \nabla F_i(\mathbf{x})$, and has bounded variance

$$E\Big[\ \big\|\nabla F_i(\boldsymbol{w}_t^k,\ \xi_t^k) - \nabla F_i(\boldsymbol{w}_t^k)\big\|^2\ \Big] \leq \sigma_k^2,$$

$$\sigma_k^2 \geq 0, \sum_{k\in\mathcal{N}} p_k\sigma_k < \sigma \quad \forall k \in \mathcal{N}.$$

Assumption 4: (*Bounded Gradient*). The deviation between local and global gradient satisfies:

$$E\Big[\ \big\|\nabla F_k(\boldsymbol{w}_t^k,\ \xi_t^k)\big\|^2\ \Big] \leq G^2,$$

$$\forall k \in \mathcal{N},\quad \forall t \in \{1,\ 2,\ ,\ ...,\ T\}.$$

Assumption 5: (*Bounded Delay*). The period of two adjacent global updates for any client is not greater than τ_{max}, i.e. $t - \tau_t^k + 1 \leq \tau_{max},\ \forall k \in \mathcal{N},\quad \forall t \in \{1,\ 2,\ ,\ ...,\ T\}$, where τ_t^k is the asynchronous model delay which represents that client k uses the model at the round $t - \tau_t^k$ to compute his local update and update to the server at the round t.

Assumption 1, 2 and 5 are standard and commonly used in the literature on learning and optimization [3], [5], [14]. For Assumption 3, the boundedness of local stochastic gradients' variances is also a common assumption for prior work on FL with non-IID datasets [15]–[17]. Assumption 4 is used in some works [8].

To satisfy Assumption 5, each client should participate in at least the first round of the learning process, so that the FL server can have the most recent local model for each client in each round. This condition is necessary since, otherwise, there does not exist an FL algorithm that can always guarantee a vanishing convergence error. To see this, when this condition does not hold, we present a lower bound on the convergence error, which is achieved in the worst-case scenario of client participation.

Proposition 1: (Convergence error lower bound with arbitrary client participation) Let Ω_r characterize heterogeneity of clients' local data. Suppose client participation can be arbitrary. There exist loss functions satisfying Assumptions 1-3, and a particular client participation process, such that for any convergent FL algorithm, its convergence error is lower bounded by:

$$E\Big[\ \|F(\boldsymbol{w}_t) - F^*\|^2\ \Big] \geq \mathcal{O}(\Omega_r).$$

Remark 1: Intuitively, in the worst-case scenario when some client never participates in any round, then the global model can only be driven towards the local optimal models of the clients who participate, and there is no way to shift the global model towards the global optimal model. The lower bound above is achieved in such worst-case scenario (see our online technical report [13] for details).

Next we analyze the convergence of the proposed AFLC algorithm as below.

Theorem 1: (*Convergence error of AFLC*) Suppose Assumptions 1- 5 hold. If we set $\gamma = \max\{\frac{8L}{\mu},\ H_{max}\}$ where H_{max} is the maximum local iteration numbers among all users in all rounds, and set the learning rate $\eta_t = \frac{2}{\mu(\gamma+t)}$, then the training loss of AFLC given by Algorithm 1 is upper bounded by:

$$E\Big[\ \|F(\boldsymbol{w}_t) - F^*\|^2\ \Big] \leq \frac{2L}{\mu(\gamma+t)}\left(\frac{B}{\mu} + 2L\left\|\boldsymbol{w}_1 - \boldsymbol{w}^*\right\|^2\right), \tag{1}$$

where $B = 6L\Gamma + \sigma^2 + (2+\mu\eta_t)G^2\sum_{k\in\mathcal{N}} K^k_{t,\tau_t^k}$ and $K^k_{t,\tau_t^k} = \sum_{i=t-\tau_t^k}^{t} p_k\eta_i^2(H_i^k - 1)^2$.

Remark 2: Theorem 1 shows that the proposed Algorithm 1 can converge to the optimal value (rather than an error neighborhood) in the sense that the convergence error can be made arbitrarily small if the number of rounds t is large enough. It has been shown in prior work [18] that FL with arbitrary client participation results in a non-vanishing convergence error. This is due to an objective function drift under the worst-case scenario of client participation, regardless of the choices of learning rates and local iteration numbers. In our proposed AFLC algorithm, we use the most recent local model from a client in a round if the server does not receive a local model update from that client in that round. In this way, we show that the objective function drift can be addressed, despite of using the most recent local model rather than the actual local model from the client if the server would receive a local model update from that client in that round. In fact, the error between the most recent local model and the actual local model can be properly controlled by choosing an appropriate learning rate.

Remark 3: We observe that the convergence error bound depends on agents' local iteration numbers $\{H_i\}$, and the bound increases as $\{H_i\}$ increase. Intuitively, due to clients' heterogeneous data, more local computation iterations drives each client's local model more towards its local optimal model and possibly away from the global optimal model (also known as "local drifts" in existing works on FL [7], [8]). As a result, the convergence error bound increases as the local iteration numbers go up. Note that the term involving B goes to 0 when clients' local iteration numbers are all equal to 1 (i.e., $H_i = 1, \forall i$).

We also observe that the error bound increases as agents' local model delays $\{t_i\}$ increase. This is because, as the local model delay increases, there is more error in the most recent local model used in the AFLC algorithm compared to the actual local model without any delay. Therefore, the error increases when the delay is higher.

Remark 4: We note that existing works on FL predominantly considered non-convex learning problems [5], [19], while only a few of them studied the convex setting [8]. In particular, a very recent work [5] proposed anarchic federated learning algorithms for the non-convex setting. Although the algorithms in [5] are similar to the AFLC algorithms proposed in this paper, there are some non-trivial differences for the convex setting in the convergence analysis of the algorithms. In particular, the learning accuracy in the convex setting is quantified by the normed distance between the model found by the algorithm and the optimal model, which is quite different from that in the non-convex setting (which is the normed gradient of the objective function). As a result, the major techniques

used in the convergence analysis for the convex setting are significantly different from those in [5]. In particular, the convexity of the training loss functions of the learning problem is utilized to establish an important bound in the convergence analysis. Also, a properly designed diminishing stepsize needs to be used here to achieve a vanishing convergence error. In contrast, a constant stepsize is used in [5] for the non-convex setting, which can achieve a vanishing convergence error.

Remark 5: It is also worth noting that, to deal with heterogeneous local iteration numbers of clients, a couple of recent studies [5], [10] proposed to scale the aggregation weights of clients' local models according to their local iteration numbers, so as to achieve a vanishing convergence error. However, the AFLC algorithm in this paper can do so without using this technique, which is due to the differences of the convex setting here compared to the non-convex case studied in [5], [10].

Corollary 1: (*Convergence Rate*). With a decaying learning rate $\eta_t = \frac{2}{\mu(\gamma+t)}$, the convergence rate of AFLC is $\mathcal{O}(\frac{1}{T})$.

Remark 6: It has been shown that an asynchnorous FL algorithm under the non-convex setting can achieve a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ (e.g., AsyncCommSGD [20], AFA-CD [5]). A synchnoronous FL algorithm under the convex setting usually achieve a convergence rate of $\mathcal{O}(\frac{1}{T})$ (e.g., FedAvg-non-IID [8], Fedprox [21]). As our AFLC algorithm which is asynchnorous and under the convex setting can reach a convergence rate of $\mathcal{O}(\frac{1}{T})$, it matches that of the existing synchnoronous algorithms for convex learning and outperforms the existing asynchnorous algorithms for non-convex learning.

V. Numerical Experiments

In this section, we conduct simulations to verify the theoretical findings and evaluate the efficiency of the proposed algorithm. We first describe the simulation setup, and then present the results and analysis.

A. Simulation Setup

In this section, we run the image recognition program on the minist database using the ASUS laptop to evaluate the effectiveness of AFLC and verify our theoretical results. The MNIST is a database of handwritten digits which is used for training image processing programs. The database is also widely used for training and testing in the field of machine learning [22]. We first examine the convergence rate of 4 algorithms. We perform the simulations in terms of 3 design variables, which are the number of clients updating with the server per round (N), the degree of non-IID (Γ), the maximum local iteration numbers (H_t^{max}), and the maximum local update delay (τ_{max}). We study the relationship between the convergence rate and these 3 variables, respectively.

B. Simulation Results

1) Convergence rate: As shown in Fig 2, we check the test accuracy of 4 different algorithms, which are AFA-CD [5], the proposed algorithm, FedAsync [23], and FedAvg [1]. The test accuracy shows that our algorithm is efficient and accurate. The accuracy of our algorithm is similar to the AFA-CD.

2) Impact of the number of clients updating with the server per round: As shown in Fig 3, fast-k refers to the server collecting the fast-k clients who finish their local computation under the maximum delay constraint in the IID setting. It is shown that more clients participating in a round can speed up the training process because more participating clients can make more clients compute with a smaller delays model. When there are only a few clients can update their model in each round, then there must exist a straggler with a higher delay, which can decline in aggregation results.

3) Impact of the number of the degree of non-IID: It is hard to get the exact value of Γ in (1), so we use 'IID dataset', 'Non-IID Data with balanced samples and labels', 'Non-IID Dataset with unbalanced label samples', 'Non-IID Dataset with unbalanced samples', where balanced samples mean all clients' number of each label is equal in their dataset, and balanced labels mean all clients label types are same. It is easy to find that the degree of non-IID in these 4 cases is increasing. It is shown in Fig 4 that the degree of non-IID affects the convergence rate, but it does not affect the final results, which meet our analyses.

4) Impact of the maximum local iteration numbers: Fig 5 shows that increasing the maximum local iteration numbers will decrease the convergence rate of the FL. However, in asynchronous federated learning, simply increasing local iteration numbers can sometimes make bad effects on the training loss, because some stragglers with larger local iteration numbers can ruin the results of the global model. In addition, increasing local iteration numbers cannot improve the final training accuracy which meets our theoretical analyses.

5) Impact of the maximum local update delay: As shown in Fig 6, the value of the global loss function increases with increasing delay time. For a synchronous FL ($t_{max} = 0$), which obviously has the lowest global loss among all settings. At the beginning of the FL training, the difference between the 4 different maximum local update delays is relatively small, since we only constrain the maximum value of the local update delay, resulting in a smaller difference in user choices at the beginning of the training. When the training is nearly finished, the difference becomes larger and larger due to the straggler clients.

VI. Conclusion and Future Work

FL is an emerging topic in ML/AI and networking that has recently received tremendous interests. In this paper, we propose AFLC, which allows clients to 1) participate in arbitrary rounds; 2) participate asynchronously; 3) participate with arbitrary numbers of local iterations. The AFLC algorithm enables clients to participate in FL efficiently and flexibly according to their needs, e.g., based on their heterogeneous and time-varying computation and communication capabilities. We characterized performance bounds on the learning loss of AFLC as a function of clients' local model delays, and local iteration numbers, which show that the learning loss can be made arbitrarily small by choosing appropriate learning rates. We demonstrated the efficiency of the AFLC via numerical

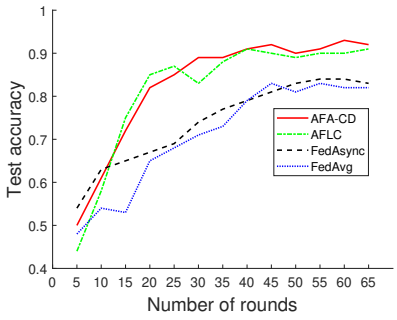

Fig. 2. Test accuracy of different algorithms.

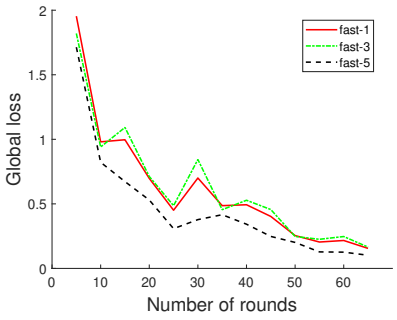

Fig. 3. Impact of the number of clients updating with the server per round (N).

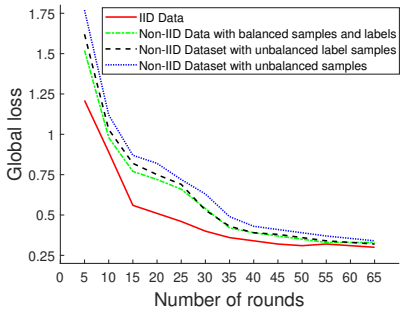

Fig. 4. Impact of the non-IID degree (Γ): The non-IID degree is increasing across the 4 schemes.

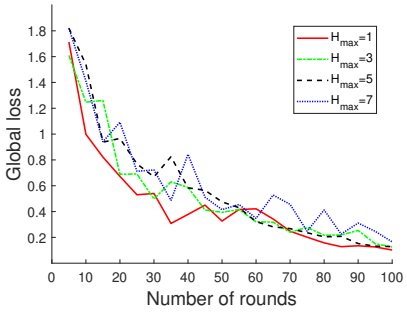

Fig. 5. Impact of the maximum local iteration number (H_t^{max}).

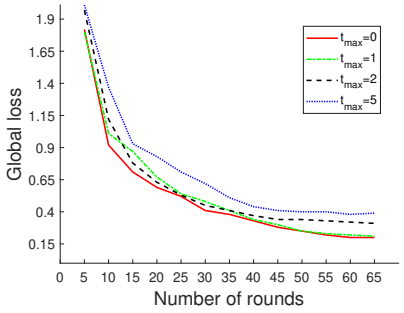

Fig. 6. Impact of the maximum local update delay (τ_{max}).

results. For future work, we will explore AFL in other settings of FL, such as for decentralized networks of clients. These cases will be more challenging to study due to the complex communication structure.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning (ICML)*, 2020.

[3] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[4] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[5] H. Yang, X. Zhang, P. Khanduri, and J. Liu, "Anarchic federated learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2022.

[6] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," in *SysML Conference*, 2019.

[7] S. U. Stich, "Local SGD converges fast and communicates little," in *International Conference on Learning Representations (ICLR)*, 2019.

[8] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *International Conference on Learning Representations (ICLR)*, 2020.

[9] S. Wang and M. Ji, "A unified analysis of federated learning with arbitrary client participation," in *Advances in Neural Information Processing Systems (NIPS)*, 2022.

[10] L. Zhu, H. Lin, Y. Lu, Y. Lin, and S. Han, "Delayed gradient averaging: Tolerate the communication latency for federated learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2021.

[11] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 2737–2745.

[12] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2019.

[13] Technical report of anarchic convex federated learning. [Online]. Available: https://auburn.box.com/s/ohlj56yjzy9btzm17oe1llqdgocn2v2e

[14] Z. Chai, Y. Chen, L. Zhao, Y. Cheng, and H. Rangwala, "Fedat: A communication-efficient federated learning method with asynchronous tiers under non-iid data," *ArXiv*, 2020.

[15] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[16] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[17] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[18] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5234–5249, 2021.

[19] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019, pp. 1387–1395.

[20] D. Avdiukhin and S. Kasiviswanathan, "Federated learning under arbitrary communication patterns," in *International Conference on Machine Learning*. PMLR, 2021, pp. 425–435.

[21] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, vol. 3, p. 3, 2018.

[22] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[23] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.