Anarchic Federated Bilevel Optimization

Dongsheng Li, Ye Zhu, Xiaowen Gong, Shiwen Mao, Yang Zhou
Department of Electrical and Computer Engineering
Auburn University, Auburn, AL 36849
Email: {dzl0093,yzz0211,xgong,szm0001,yangzhou}@auburn.edu

***Abstract*—Rapid federated bilevel optimization (FBO) developments have attracted much attention in various emerging machine learning and communication applications. Existing work on FBO often assumes that clients participate in the learning process with some particular pattern (such as balanced participation), and/or in a synchronous manner, and/or with homogeneous local iteration numbers, which might be hard to hold in practice. This paper proposes a novel Anarchic Federated Bilevel Optimization (AFBO) algorithm, which allows clients to 1) participate in any inner or outer rounds; 2) participate asynchronously; and 3) participate with any number of local iterations. The AFBO algorithm enables clients to participate in FBO training flexibly. We provide a theoretical analysis of the learning loss of AFBO for both cases of non-convex and strongly convex loss functions. The convergence results of the AFBO algorithm match that of the existing benchmarks. Numerical studies are conducted to verify the effectiveness of AFBO.**

***Index Terms*—bilevel optimization, federated learning, asynchronous, partial participation.**

I. Introduction

Bilevel Optimization (BO) involves two levels of optimization tasks, with one optimization problem nested within the other. The outer optimization problem is often called the leader's (upper-level) optimization problem, and the inner optimization problem is often called the follower's (lower-level) optimization problem. This two-level optimization can be viewed as a constrained optimization problem, where the lower-level optimization problem can be viewed as the constraints of the upper-level optimization problem. BO has been widely used in many important applications, such as meta-learning, hyper-parameter optimization, model selection, adversarial networks, game theory, and reinforcement learning. Recently, some studies have provided non-asymptotic analyses for BO, such as BSA [1], TTSA [2], ALSET [3] and so on.

Federated Bilevel Optimization (FBO) is to run BO problems in a federated system. It allows a large number of clients to perform computations in parallel, rather than in series or with one client, which can not only save training time but also keep the data private. [1], [2], [4] proposed methods to compute the estimated hyper-gradient. After that, FBO has received much attention [5]–[7]. For example, [8] studied the federated meta-learning problems, [9] provided a federated hyperparameter optimization approach, and [10] improved the fairness of federated learning using a bilevel method. In addition, [11] studied the decentralized stochastic bilevel problem, and [12] proposed an asynchronous distributed bilevel algorithm.

To fully realize the potential of FBO, several challenges need to be addressed, such as heterogeneous and/or time-varying computation and communication capabilities of clients, and the clients' partial and/or asynchronous participation. First of all, clients' heterogeneous computation and communication capabilities lead to a large waiting time when using synchronous algorithms. Furthermore, if the fast clients (with a large computation rate or small communication time) run more local iterations, then the training result will converge to a local optimal instead of a global optimal. In addition, clients may not be able to participate in each round during the entire learning process.

This paper provides a novel Anarchic Federated Bilevel Optimization (AFBO) algorithm to address flexible asynchronous participation, heterogeneous local iteration numbers, and non-IID data simultaneously in FBO. Specifically, AFBO imposes minimum control over how clients participate in FBO by allowing them to 1) participate in any outer or inner rounds (respectively); 2) participate asynchronously across outer rounds or inner rounds (respectively); 3) participate with any numbers of local iterations; 4) participate with arbitrary dataset distributions. By giving clients maximum freedom, AFBO enables them to participate flexibly, accommodating their heterogeneous and time-varying computation and communication capabilities. Meanwhile, the asynchronous communication structure of AFBO can greatly reduce the wall-clock time, as clients synchronize their local models with the server using the most recent data stored in the server instead of waiting for stragglers. Furthermore, due to the two-level structure of the optimization problem, the double-loop algorithm is proposed, thus double asynchronous delay is the essential parts to be analysis, which is much more challenge than previous asynchronous works. Our contributions are summarized as follows.

- We propose AFBO, which allows clients to participate asynchronously in any rounds with heterogeneous local iteration numbers and dataset distributions. AFBO allows clients to participate in FBO efficiently and flexibly with their heterogeneous system parameters and datasets. The algorithm design of AFBO involves some key techniques, including using clients' most recent local gradients, and adjusting learning rates when the delays of inner loop and outer loop increase.

This work was supported by U.S. NSF grants ECCS-2121215, CNS-2319342, and CNS-2107190.

TABLE I
COMPARISON FOR DIFFERENT FBO ALGORITHMS. S REPRESENTS SYNCHRONOUS AND AS REPRESENTS ASYNCHRONOUS.

Related Work	Outer Loop	Inner Loop	Partial Participation	Convergence Rate	Data Heterogeneity	Linear Speedup
[13]	S	S	$\times$	$\mathcal{O}(\varepsilon^{-1.5}n^{-1})$	$\times$	$\surd$
[14]	S	S	$\times$	$\mathcal{O}(\varepsilon^{-1.5})$	$\surd$	$\times$
[6]	S	S	$\times$	$\mathcal{O}(\varepsilon^{-2})$	$\surd$	$\times$
[5]	S	S	$\surd$	$\mathcal{O}(\varepsilon^{-2}n^{-1})$	$\surd$	$\surd$
[15]	S	S	$\surd$	$\mathcal{O}(\varepsilon^{-2}n^{-1})$	$\surd$	$\surd$
[12]	AS	S	$\surd$	$\mathcal{O}(\varepsilon^{-2})$	$\times$	$\times$
This paper	AS	AS	$\surd$	$\mathcal{O}(\varepsilon^{-2}n^{-1})$	$\surd$	$\surd$

- We provide convergence analyses for AFBO with the settings of strongly convex and non-convex upper-level objectives respectively. Our results show that the AFBO algorithm can achieve a convergence rate of $\mathcal{O}(\frac{1}{T})$ for the strongly convex upper-level objective, and a convergence rate of $\mathcal{O}(\sqrt{\frac{1}{T}})$ as well as a linear convergence speedup ($\mathcal{O}(\sqrt{\frac{1}{mT}})$) for the non-convex upper-level objective, which matches that of existing benchmarks. The results also characterize the impacts of clients' local iteration numbers, local model delays, and global model delays on learning loss.
- We conduct numerical experiments to verify the effectiveness of AFBO. The experimental results demonstrate the efficiency of the AFBO.

The remainder of this paper is organized as follows. Section II reviews related work. In Section III, we present the system model and algorithm design of AFBO. In Section IV, we provide the convergence analysis of the AFBO algorithm. Numerical results and conclusions are provided in Section V and Section VI, respectively.

II. Related Work

Bilevel Optimization. The BO problem was first introduced by [16]. Some recent works [17] assumed there is an analytical solution to the lower-level optimization problem, and then the BO problem can be reduced to a single-level problem. However, it is not always possible to find an analytical solution for lower-level problems. [18] replaced the lower-level optimization problem with optimal surrogate under some sufficient conditions (e.g., KKT conditions). Then, the bilevel problem can be reformulated as a single-level constrained optimization problem. However, the resulting problem could be hard to solve since it often involves a large number of constraints [19]. Then, [1], [2], [4] proposed gradient-based methods, which compute the hyper-gradient (or the estimation of hyper-gradient), i.e., $\frac{\partial F(x,y)}{\partial x} + \frac{\partial F(x,y)}{\partial y}\frac{\partial y}{\partial x}$, and use gradient descent (GD) or stochastic gradient descent (SGD) methods to solve the bilevel optimization problems. As far as we know, most BO papers focus on synchronous communication with homogeneous local iterations in inner rounds, which is limited in real-world applications.

Federated Bilevel Optimization. Most existing bilevel optimization algorithms focus on centralized settings and require collecting massive amounts of data from distributed edge clients. This may give rise to data privacy risks and communication bottlenecks [20]. In federated bilevel problems, it is challenging to approximate the hyper-gradient (i.e., $\frac{1}{M}\sum_{i=1}^{M}\frac{\partial F_i(x,y)}{\partial y}\frac{\partial y}{\partial x} \neq \frac{\partial F(x,y)}{\partial y}\frac{\partial y}{\partial x}$). [21] proposed automatic machine learning, which is a powerful tool for approximating the hyper-gradient. [5] studied data heterogeneity in federated bilevel problems under a synchronous setting with one local iteration in the inner loop. [22] provided an algorithm ShroFBO to reduce communication costs and allow heterogeneous local computation in the synchronous FBO system. [14] used momentum-based variance reduced local-SGD to reach a communication-efficient FBO. [23] proposed a different FL algorithm based on Local-SVRG to obtain exact gradient information and achieve lower communication complexity. [24] proposed a new backward updating mechanism to collaboratively learn the model without privacy leakage in an asynchronous vertical federated learning system. [25] provided some low communication complexity algorithms to solve FBO problems through the variance-reduction technique. [12] proposed the ADBO algorithm to solve the bilevel optimization problem in an asynchronous distributed manner. [5] proposed a synchronous partial participation FBO based on the parallel hyper-gradient estimator. [15] proposed a synchronous flexible FBO algorithm that has no sub-loops to improve communication efficiency. [26] proposed an efficient estimation of the hyper-gradient in the distributed setting to achieve a communication complexity $\mathcal{O}(\epsilon^{-1})$, a sample complexity $O(\epsilon^{-1.5})$ and a linear speed-up. [27] proposed a distributed BO algorithm to achieve a communication complexity $\mathcal{O}(\epsilon^{-2})$. This paper considers a federated bilevel optimization problem with *partial participation*, and *heterogeneous local iteration numbers* in an *asynchronous* federated communication setting-AFBO, which has several major differences compared to general BO works. In addition, our paper considers a two-level optimization problem (e.g., meta-learning), which is different from the existing FL works (e.g., AFL).

III. Anarchic Federated Bilevel Optimization

A. System Setting and Problem Formulation

A general bilevel optimization in a federated learning system is formulated as follows,

$$\min_{x\in\mathbb{R}^p} \Phi(x) := f(x, y^*(x)) := \frac{1}{m}\sum_{i\in\mathcal{M}} f_i(x, y^*(x)) \tag{1a}$$

$$s.t.\;\; y^*(x) = \text{argmin}_{y\in\mathbb{R}^q} g(x,y) := \frac{1}{m}\sum_{i\in\mathcal{M}} g_i(x,y), \tag{1b}$$

where $f_i(x,y) = \mathbb{E}\left[f_i(x,y;\xi_i)\right]$ and $g_i(x,y) = \mathbb{E}\left[g_i(x,y;\zeta_i)\right]$ are stochastic upper- and lower-level loss functions of client i, respectively. $\mathcal{M}$ is the set of clients, and $y^*(x)$ is the optimal solution of the lower-level problem. The goal of problem (1) is to minimize the objective function $\Phi(x)$ with respect to (w.r.t.) x, where $y^*(x)$ is obtained by solving the lower-level minimization problem.

The key step to find the solution of problem (1) is to exactly estimate the $\nabla\Phi$ as follows,

$$\nabla\Phi(x) = \nabla_x f(x,y^*(x)) - \nabla^2_{xy} g(x,y^*(x)) \left[\nabla^2_{yy} g(x,y^*(x))\right]^{-1} \nabla_y f(x,y^*(x)), \tag{2}$$

where $\nabla^2_{yy} g(x,y)$ is defined as the Hessian matrix of g w.r.t. y and $\nabla^2_{xy} g(x,y)$ is

$$\nabla^2_{xy} g(x,y) := \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial y_1} g(x,y) & ... & \frac{\partial^2}{\partial x_1 \partial y_q} g(x,y) \\ & ... & \\ \frac{\partial^2}{\partial x_p \partial y_1} g(x,y) & ... & \frac{\partial^2}{\partial x_p \partial y_q} g(x,y) \end{bmatrix}.$$

Since it is hard to find $y^*(x)$ in each step, we usually use a surrogate to efficiently approximate the hyper-gradient $\nabla\Phi$ in (2), denoted as

$$\bar{\nabla}\Phi(x^t) = \nabla_x f(x^t, y^{t+1}) - \nabla^2_{xy} g(x^t, y^{t+1}) \left[\nabla^2_{yy} g(x^t, y^{t+1})\right]^{-1} \nabla_y f(x^t, y^{t+1}),$$

where t is the outer loop round index.

Let $H_i(x^t, y^{t+1})$ be a hyper-gradient estimator of client i to approximate $\nabla f(x^t, y^{t+1})$ and $\mathcal{F}^t := \sigma\{y^0,\ x^0,\ ...,\ y^t,\ x^t,\ y^{t+1}\}$ denotes the filtration that captures all the randomness up to the t-th outer loop. We denote $\overline{H}_i(x^t, y^{t+1}) := \mathbb{E}[H_i(x^t, y^{t+1})|\mathcal{F}^t]$ and $\overline{H}(x^t, y^{t+1}) := \mathbb{E}[\frac{1}{m}\sum_{i\in\mathcal{M}} \overline{H}_i(x^{t-\tau_i^t}, y^{t-\rho_i^t+1})]$, where τ_i^t (ρ_i^t) is the outer (inner) rounds asynchronous delay of client i in round t, respectively.

B. Algorithm Design of AFBO

The challenges in solving the bilevel problem in problem 1 lie in computing the federated hyper-gradient $\nabla\Phi(x) = (1/m)\sum_{i=1}^m \nabla f_i(x, y^*(x))$, whose explicit form can be obtained as follows via implicit differentiation. To calculate the above equation, we need to overcome some difficulties in the federated setting. (i) It is required to approximate the minimizer $y^*(x)$ of the lower-level problem, which can introduce a bias due to the client drift, especially when there are heterogenous local iteration numbers and partial participation; (ii) The computation of a series of global Hessian-vector products lies in a nonlinear manner (i.e., $\frac{1}{M}\sum_{i=1}^M \frac{\partial F_i(x,y)}{\partial y}\frac{\partial y}{\partial x} \neq \frac{\partial F(x,y)}{\partial y}\frac{\partial y}{\partial x}$), which can introduce a large estimation variance; (iii) The federated hyper-gradient estimation may suffer from a bias due to both the upper- and lower-level client drifts including partial participation, asynchronous delay, and unbalanced computing and/or communication abilities, etc. To address those challenges, we propose a double-loop scheme Anarchic Federated Bilevel Optimization (AFBO) algorithm. We use the following federated hyper-gradient estimation:

$$\begin{aligned} &H_i(x^t, y^{t+1}) = \nabla_x f_i(x^t, y^{t+1}; \phi_i^t) \\ &- \left[\frac{1}{m}\sum_{j=1}^m \nabla^2_{xy} g_j(x^{t-\tau_j^t}, y^{t-\rho_j^t+1}; \varsigma_i^{t-\rho_j^t+1})\right] \\ &\times \frac{m}{l_{g,1}} \prod_{l=1}^{U} \left(I - \frac{1}{l_{g,1} m}\sum_{j=1}^m \nabla^2_{yy} g_j(x^{t-\tau_j^t}, y^{t-\rho_j^t+1}; \zeta_i^{t-\rho_j^t+1})\right) \\ &\times \nabla_y f_i(x^{t-\tau_i^t}, y^{t-\rho_i^t+1}; \xi_i^{t-\tau_i^t+1}). \end{aligned}$$

Algorithm 1 Anarchic Federated Bilevel Optimization (AFBO)

```
1:  input: full client index set $\mathcal{M}$, initial point $(x_0,\ y_0)$ local computation delays $\{\tau_i^t|\ i\in\mathcal{M},\ t\in[1,T]\}$, inner loop iterations $\{K^t|\ t\in[1,T]\}\}$ where $K^t = \max_i\{K_i^t|\ \forall i\in\mathcal{M}\}$,the number of local iterations $\{E_i^{t,k}|\forall i\in\mathcal{M},\ t\in[0,T-1]\}$
2:  for Round $t = 0$ to $T-1$ do
3:      for $k = 0$ to $K^t - 1$ do
4:          for Client $i\in\mathcal{M}$ in parallel do
5:              for $e = 0$ to $E_i^{t,k} - 1$ do
6:                  Client $i$ computes $G_{i,e}^{t,k} = \nabla_y g_i(x^t, y_{i,e}^{t,k}; \xi_{i,e}^{t,k})$;
7:              end for
8:              Client $i$ sends $G_i^{t,k} = \frac{1}{E_i^{t,k}}\sum_{e=0}^{E_i^{t,k}-1} G_{i,e}^{t,k}$ to the server;
9:              Server records the delay of each client, i,e., set $\rho_i^k = 1$ and store $G_i^{t,k}$ on the server for updating clients and set $\rho_i^k = \rho_i^{k-1} + 1$ for non-updating clients;
10:             Server computes $G^{t,k} = \frac{1}{m}\sum_{i\in\mathcal{M}} G_i^{t,k-\rho_i^k}$, $y^{t,k+1} = y^{t,k} - \beta_{t,k} G^{t,k}$, and broadcasts $y^{t,k+1}$;
11:         end for
12:         Client $i$ computes $H_i^t$ = APHE$(x^t, y^{t,K^t-1})$ and sends $H_i^t$ to the server;
13:     end for
14:     Server records the delay of each client, i,e., set $\tau_i^t = 1$ and store $H_i^t$ on the server for updating clients and $\tau_i^t = \tau_i^{t-1} + 1$ for non-updating clients;
15:     Server computes $H^t = \frac{1}{m}\sum_{i\in\mathcal{M}} H_i^{t-\tau_i^t}$, $x^{t+1} = x^t - \eta_t H^t$, and broadcasts $x^{t+1}$;
16: end for
```

Note that AFBO chooses $U = m$ as the server keeps the most recent updates of all clients, while previous works (e.g., FedMBO, FedNest) usually choose U from $\{0,\ ...,\ m-1\}$ uniformly at random for synchronous FBO to keep the communication effective. Then we use these updates to make all clients "participate" in all rounds and aggregate $\nabla f(x)$ through $\nabla f(x^t) = (1/m)\sum_{i\in\mathcal{M}} H_i(x^{t-\tau_i^t}, y^{t-\rho_i^t+1})$, where τ_i^t is the time delay of client i's last communication with the server in round t. By using the most recent updates of all clients, the asynchronous bias can be eliminated, which is proved in Theorem 4.1. In addition, AFBO uses aligned update (i.e., $\frac{1}{E_i^{t,k}}\sum_{c=0}^{E_i^{t,k}-1}\nabla_y g_{i,c}$) to correct the bias from heterogeneous local iteration numbers. In this way, AFBO has the most flexibility during the FBO training process. The key difference between AFBO and FedMBO [5] is that AFBO allows both inner loop and outer loop asynchronous communication and does not need to sample participated clients in each round as the server holds the most recent updates of all clients.

Algorithm 2 Anarchic Parallel Hyper-gradient Estimator (APHE)

```
 1: Client $i$ receive $y^{t+1}$ from server;
 2: Client $i$ computes $d_i = \nabla_x f_i(x^t, y^{t+1}; \xi_i)$
 3: Server computes and broadcasts $p_{i,0}$, where $p_{i,0} = \frac{m}{l_{g,1}}\left[\frac{1}{N}\sum_{j=1}^m \nabla_y f_j(x^{t-\tau_j^t}, y^{t-\rho_j^t+1}; \theta_i)\right]$;
 4: for Client $i \in \mathcal{M}$ in parallel do
 5:     for $l = 0$ to $m$ do
 6:         $G = I - \frac{1}{l_{g,1}}\left[\frac{1}{m}\sum_{j=1}^m \nabla_{yy}^2 g_j(x^{t-\tau_j^t}, y^{t-\rho_j^k+1}; \zeta_{i,l})\right]$;
 7:         $p_{i,l} = Gp_{i,l-1}$;
 8:         $W_i = \frac{1}{m}\sum_{j=1}^m \nabla_{xy}^2 g_j(x^{t-\tau_j^t}, y^{t-\rho_j^t+1}; \phi_i)$;
 9:     end for
10: end for
11: Return: $H_i^t = d_i - W_i \times p_{i,N}$.
```

C. Procedure of AFBO

In this section, we develop AFBO, which is formally described in Algorithm 1. We first define two types of clients in round t: 1) updating clients who have completed both their local computations and 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; 2) non-updating clients who do not participate (i.e., do not perform any local computation or any communication) in round t, or who participate but have not completed their local computations or communications of their local models to the server in round t.

Then we present the full procedure of AFBO. Specifically, the server receives the update from updating clients, and retrieves the most recent local models of non-updating clients from the server's memory (since they were lastly updated to the server when the non-updating clients in this round do their communication in previous rounds) to aggregate the global model and broadcasts the latest global model to the updating clients. After clients receive the global model, clients start their inner rounds of training. For the client i receiving the global model, she uses the federated hyper-gradient estimation method to estimate the Hessian inverse matrix. After she performs $E_i^{t,k}$ local iterations of inner rounds SGD, she aligns her local updates (i.e., $G_i^{t,k} = \frac{1}{E_i^{t,k}}\sum_{e=0}^{E_i^{t,k}-1} G_{i,e}^{t,k}$), and computes H_i^t using Algorithm 2, then she sends H_i^t to the server to update global model. Clients and the server repeat these processes until the global model converges to an ϵ-optimal global model.

IV. Convergence Analysis of AFBO

A. Assumptions

Definition 4.1: A function $h : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous with constant L, if $\|h(x) - h(y)\| \le L\,\|x-y\|\,,\ \forall x,\ y \in \mathbb{R}$.

Definition 4.2: A function $h : \mathbb{R} \to \mathbb{R}$ is strongly convex with constant μ, if $h(y) - h(x) \ge \langle \nabla h(x), y-x\rangle + \frac{\mu}{2}||y-x||^2,\ \forall x,\ y \in \mathbb{R}$.

Definition 4.3: A solution x is ϵ-accurate stationary point if $\mathbb{E}\left[\|\nabla f(x)\|^2\right] \le \epsilon$, where x is the output of an algorithm.

Let $z = (x, y) \in \mathbb{R}^{p+q}$ denote all parameters. Throughout this paper, we make the following assumptions on inner/outer objectives.

Assumption 4.1: (Lipschitz properties) For all $i \in \mathcal{M}$: $f_i(x)$, $\nabla f_i(z)$, $\nabla g_i(z)$, $\nabla^2 g_i(z)$ are $l_{f,0}$, $l_{f,1}$, $l_{g,0}$, $l_{g,1}$, $l_{g,2}$-Lipshitz continuous, respectively.

Assumption 4.2: (Strong convexity) For all $i \in \mathcal{M}$: $g_i(x, y)$ is μ_g-strongly convex in y for any fixed $x \in \mathcal{N}^q$.

Assumption 4.3: (Unbiased estimators) For all $i \in \mathcal{M}$: $\nabla f_i(z;\xi)$, $\nabla g_i(z;\zeta)$, $\nabla^2 g_i(z;\zeta)$ are unbiased estimators of $\nabla f_i(z)$, $\nabla g_i(z)$, $\nabla^2 g_i(z)$, respectively.

Assumption 4.4: (Bounded local SGD variances). For all $i \in \mathcal{M}$: there exist constants σ_f^2, $\sigma_{g,1}^2$, and $\sigma_{g,2}^2$, such that $\mathbb{E}_\xi\left[\|\nabla f_i(z;\xi) - \nabla f_i(z)\|^2\right] \le \sigma_f^2$, $\mathbb{E}_\zeta\left[\|\nabla g_i(z;\xi) - \nabla g_i(z)\|^2\right] \le \sigma_{g,1}^2$, $\mathbb{E}_\zeta\left[\|\nabla^2 g_i(z;\xi) - \nabla^2 g_i(z)\|^2\right] \le \sigma_{g,2}^2$.

Assumption 4.5: (Bounded global variances) There exists a constant σ_g, such that $\mathbb{E}\left[\|\nabla g_i(z) - \nabla g(z)\|^2\right] \le \sigma_g^2$, where the expectation $\mathbb{E}$ is taken over the client index i.

Assumption 4.6: (Bounded maximum delay) There exists a constant τ_M and ρ_M, such that all clients must participate in training at least every τ_M round for the outer loop and every ρ_M round for the inner loop.

These assumptions are common in the bilevel optimization literature [1], [19], [28]. Assumption 4.1 requires that the inner and outer functions are well-behaved. Assumption 4.2 supposes the strong convexity of the inner objective, implying a unique solution to the inner minimization in (1). Assumption 4.6 is necessary in many asynchronous works, such as [29], [30].

B. Convergence Analysis

Next, we present a theoretical performance guarantee for the AFBO algorithm via convergence analysis.

Theorem 4.1: Suppose Assumption 4.1 to 4.6 hold, and pick $\eta_t \leq \min\{a_1, \frac{T}{2l_{g,1}a_2}\}$ and $\beta_{t,k} < \frac{a_2\eta_t}{T}$, where $a_1 = \frac{1}{2L_f+4M_fL_y+\frac{2M_fL_{yx}}{L_y\alpha}}$ and $a_2 = \frac{5M_fL_y}{\mu_g} + \frac{\alpha L_{yx}\hat{D}_f a_1}{2\mu_g}$. Let $W^t := f(x^t) + \frac{M_f}{L_y}\left\|y^t - y^*(x^t)\right\|^2$, $V^t = \left\|x^t - x^*\right\|^2 + \frac{L_f}{\mu_f}\left\|y^*(x^t) - y^t\right\|^2$ and W^* is the optimal point of function W. Then, the sequence generated by the AFBO algorithm satisfies:

If $\{f_i(x)\}_{i\in\mathcal{M}}$ are μ_f-strongly convex, then

$$\mathbb{E}[V^T] = (1-\eta_t\mu_f)^T\mathbb{E}[V^0] + C\frac{1-(1-\eta_t\mu_f)^{T-1}}{\eta_t\mu_f},$$

where $C = \frac{2\eta_t}{\mu_f}b^2 + \frac{6\eta_t^2}{m}(\hat{D}_f\sum_{i=1}^m \tau_i^t + \hat{\sigma}_f + 4\hat{D}_f) + (2\eta_t^2 + \frac{L_fU_1}{\mu_f})\frac{\hat{D}}{m} + \frac{4L_fU_2(\sigma_g^2+\sigma_{g,1}^2)}{m\mu_fE^t}\sum_{k=0}^{K^t-1}\beta_{t,k}^2 + \frac{L_fU_3}{\mu_f}\hat{\sigma}_f$.

If $\{f_i(x)\}_{i\in\mathcal{M}}$ are non-convex, then

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \\
\leq&\frac{2}{\bar{\eta}T}\left[\mathbb{E}[W^0] - \mathbb{E}[W^*]\right] + b^2 \\
&+ \frac{8}{\bar{\eta}T}\sum_{t=0}^{T-1}(\frac{M_fU_2}{L_y} + \eta_tM_f^2)(\sigma_g^2+\sigma_{g,1}^2)\sum_{k=0}^{K^t-1}\beta_{t,k}^2 \\
&+ \frac{1}{\bar{\eta}T}\sum_{t=0}^{T-1}(\frac{3\eta_t^2L_f}{m} + \frac{M_fU_3\hat{\sigma}_f}{L_y})\hat{\sigma}_f \\
&+ \frac{1}{\bar{\eta}T}\sum_{t=0}^{T-1}(\sum_{i=1}^{m}\tau_i^t + 4)\frac{3L_f\eta_t^2\hat{D}_f}{m},
\end{aligned}$$

where $U_2 = 1 + 4M_fL_y\eta_t + \frac{\alpha L_{yx}\hat{D}_f^2\eta_t^2}{2}$, $U_3 = \frac{L_y^2\eta_t^2}{m} + \frac{L_{yx}\eta_t^2}{2\alpha m}$ for any $\alpha > 0$, $b = \kappa_g l_{f,1}(1-\frac{1}{\kappa_g})^N$, $\bar{\eta} = \frac{1}{T}\sum_{t=0}^{T-1}\eta_t$, τ_i^t is the asynchronous delay of the outer loop.

Proof: Due to the space limit, we provide all the proofs in our technical report [31].

Remark 1: It shows that the convergence error bound for the strongly convex case consists of two parts: a vanishing term that decreases and goes to 0 as the number of rounds T increases, and the non-vanishing (constant) term depending on the parameters of the problem instance and are independent of T. The vanishing term decreases much faster in the strongly convex case than in the non-convex case. For the non-vanishing term, all components are controlled by stepsize (η_t), which means it is unbiased. Note that the convergence error bound for the non-convex case consists of five parts: a vanishing term and four non-vanishing (constant) terms. The first non-vanishing term depends on the Hessian inverse approximation accuracy. The first non-vanishing term relates to the total number of participating clients (m), SGD variances of the lower level ($\sigma_{g,1}$), and the variance of local and global gradients (σ_g). The third non-vanishing term depends on SGD variances ($\hat{\sigma}_f$) and the total number of clients (m). The last non-vanishing term depends on the upper bound of the estimation of the overall objective gradient ($\hat{D}_f$), the average asynchronous delay ($\frac{1}{m}\sum_{i=1}^m \tau_i^t$), and the total number of clients (m). The decay rate of the vanishing term matches that of the typical SGD methods.

Remark 2: We observe that the first non-vanishing term involves both the variance of the local and the global gradient in the lower-level (σ_g) and the lower-level local gradient variance ($\sigma_{g,1}$), and depends on the total number of clients (m). This error term is due to the variance of stochastic gradients, and it increases as the SGD variance increases and decreases as the total number of clients increases. In addition, the lower non-IID heterogeneity can decrease the error bound of this term. If we run more inner loops, then the error of the first term will decrease. It is also highly affected by the stepsize of the inner loop, which requires us to choose a sufficiently small learning rate. The second term only depends on the rounds of Hessian-vector calculation. The third term is related to the SGD variance of the overall objective ($\hat{\sigma}_f$) and depends on the total number of clients (m). The last term of the non-vanishing term involves the bound of the SGD variance of the overall objective ($\hat{D}_f$), the total number of clients (m), and the average asynchronous delay ($\frac{1}{m}\sum_{i=1}^m \tau_i^t$). The smaller average asynchronous delay decreases the bound of the last term. It can be found that the upper bound of the convergence rate is increasing as the average outer loop delay for all rounds and all clients increases. Although there are 4 non-vanishing terms, only the last term involves the asynchronous delay (i.e., $\frac{1}{\bar{\eta}T}\sum_{t=0}^{T-1}(\sum_{i=1}^m \tau_i^t + 4)\frac{3L_f\eta_t^2\hat{D}_f}{m}$). When it turns to a synchronous FBO (i.e., $\tau_i^t = 1$), the convergence bound becomes $\frac{1}{\bar{\eta}T}\sum_{t=0}^{T-1}(\frac{3(m+4)L_f\eta_t^2}{m})\hat{D}_f$, which matches previous works [1], [6]. When the asynchronous delay (τ_i^t) increases, we need to choose a smaller step size to keep a similar convergence bound. Intuitively, we need to use a smaller step size to make the delayed gradients affect less global gradients. To make all the non-vanishing terms small, sufficiently small learning rates η_t and $\beta_{t,k}$, a large number of inner loops, and a large number of clients should be chosen. Based on Theorem 4.1, we obtain the following convergence rate for the proposed AFBO algorithm with a proper choice of the learning rate.

Corollary 4.1: Let $K^t\beta_t^2 \leq \frac{1}{T}$ and $N = \mathcal{O}(logT)$.

For the strongly convex case, picking the stepsize $\eta_t = \sqrt{\frac{1}{T}}$, then it yields

$$\mathbb{E}\left[\left\|x^t - x^*\right\|^2\right] = \mathcal{O}(\frac{1}{T}), \quad \mathbb{E}\left[\left\|y^*(x^t) - y^t\right\|^2\right] = \mathcal{O}(\frac{1}{T}).$$

For the non-convex case, picking the stepsize $\eta_t = \sqrt{\frac{m}{T}}$, then it yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla f(x^t)\right\|^2\right] \leq \mathcal{O}(\frac{1}{\sqrt{mT}} + \frac{1}{m\sqrt{mT}} + \frac{1}{T}).$$

Remark 3: Corollary 4.1 implies that to achieve an ε-optimal solution for both the lower-level and upper-level problems for the convex case, the sample complexity of AFBO is $\mathcal{O}(\varepsilon^{-2})$, which matches the previous results in FedNest [6], and BA [1]. A synchronous FL algorithm under the convex setting usually achieves a convergence rate of $\mathcal{O}(\frac{1}{T})$ (e.g., FedAvg-non-IID [32]), which matches that of the existing synchronous algorithms for convex learning.

Remark 4: It has been shown that asynchronous FL algorithms under the non-convex setting can achieve a convergence rate of $\mathcal{O}(1/\sqrt{T})$ (e.g., AFA-CD [29]). As our asynchronous algorithm can reach a convergence rate of $\mathcal{O}(1/\sqrt{T})$, it matches that of the existing asynchronous algorithms. In addition, the major term in the upper bound $\mathcal{O}(1/\sqrt{mT})$, which shows it achieves a linear speedup. Compared with FedNest [6], our complexity has the same dependence on ϵ, but a better dependence on m due to the linear speedup.

C. Proof Sketch

In this subsection, we highlight the key steps of the proof towards Theorem 4.1 as well as the differences between our analysis and the existing results. The challenges of our proof mainly lie in obtaining an upper bound for the errors of the asynchronous local gradient and the global gradient. Due to the space limitation, we defer the proof of the convex case in the appendix.

To prove Theorem 4.1, we find the difference between two continuous Lyapunov functions, which is

$$\begin{aligned}\mathbb{E}[W^{t+1}] - \mathbb{E}[W^t] &= f(x^{t+1}) - f(x^t) \\ &+ \frac{M_f}{L_y}(\|y^{t+1} - y^*(x^{t+1})\|^2 - \|y^t - y^*(x^t)\|^2).\end{aligned} \tag{3}$$

The difference in (3) consists of two terms: the first term quantifies the descent of the overall objective function $[f(x)]$; the second term characterizes the descent of the lower-level errors $[y^t - y^*(x^t)]$ and $[y^{t+1} - y^*(x^{t+1})]$. Our proof can be summarized as the descent of the overall objective function, and the upper bound of the lower-level problem.

Lemma 4.1: Suppose Assumptions 4.1, 4.2, 4.3, and 4.4 hold. We have

$$\begin{aligned}&\mathbb{E}[f(x^{t+1})] - \mathbb{E}[f(x^t)] \\ \le&(L_f\eta_t^2 - \frac{\eta_t}{2})\mathbb{E}\left[\|\overline{H}(x^t, y^{t+1})\|^2\right] - \frac{\eta_t}{2}\|\nabla f(x^t)\|^2 \\ &+ \eta_t M_f^2 \mathbb{E}\left[\|y^{t+1} - y^*(x^t)\|^2\right] + \eta_t b^2 \\ &+ \frac{3L_f\eta_t^2}{m}(\hat{D}_f \sum_{i=1}^m \tau_i^t + \hat{\sigma}_f + 4\hat{D}_f).\end{aligned}$$

Lemma 4.1 shows that the gradient descent of the overall objective function is related to the delay of all clients ($\sum_{i=1}^m \tau_i^t$). It is because we use the most recent updates of all clients to compensate for the bias of asynchronous communication. If we use synchronous communication ($\tau_i^t = 1$), the above result can match the general synchronous FBO works.

Lemma 4.2: Suppose Assumptions 4.1, 4.2, 4.3, 4.4, 4.5 hold. Letting $\beta_t \le \frac{1}{2l_{g,1}}$, we have

$$\begin{aligned}\mathbb{E}\left[\|y^{t+1} - y^*(x^t)\|^2\right] &\le 4(\sigma_g^2 + \sigma_{g,1}^2)\sum_{k=0}^{K^t-1}\beta_{t,k}^2 \\ &+ \left(\prod_{k=0}^{K^t-1}(1 - \beta_{t,k}\mu_g)\right)\mathbb{E}\left[\|y^t - y^*(x^t)\|^2\right].\end{aligned}$$

Lemma 4.2 shows that the error of the two consecutive inner-loop rounds is arbitrarily smaller when we select the appropriate inner-loop step size.

V. Numerical Experiments

In this section, we conduct experiments on hyper-parameter optimization tasks in the distributed setting to evaluate the performance of the proposed AFBO and validate our theoretical results. All experiments are implemented in Matlab 2023a on an ASUS laptop with an Nvidia GeForce GTX GPU. Note that the current experiments and the results in other related works are all simulations on a single laptop and simulated for distributed communication. The linear speedup improvement can be shown by implementing the model and the algorithms on a distributed setting with multiple machines.

To prove the efficiency of the proposed AFBO algorithm, we use a similar setting in [12].

A. Data Hyper-Cleaning Task

Following [19], [33], the proposed AFBO algorithm is compared with ADBO [12] and distributed bilevel optimization method FedNest [6] on the distributed data hyper-cleaning task [34] on MNIST datasets. Data hyper-cleaning involves training a classifier in a contaminated environment where each training data label is changed to a random class number with a probability (i.e., the corruption rate). In addition, we further consider the effect of heterogeneous data distribution on the training performance. The distributed data hyper-cleaning problem can be expressed as,

$$\begin{aligned}\min F(\boldsymbol{\phi}, \boldsymbol{\omega}) &= \sum_{i=1}^m \frac{1}{|D_i^{val}|}\sum_{(\boldsymbol{x}_j, y_j)\in D_i^{val}} L(\boldsymbol{x}_j^T\boldsymbol{\omega}, y_j) \\ s.t.\ \boldsymbol{\omega} &= \arg\min_{\boldsymbol{\omega}'} f(\boldsymbol{\phi}, \boldsymbol{\omega}') \\ &= \sum_{i=1}^m \frac{1}{|D_i^{tr}|}\sum_{(\boldsymbol{x}_j, y_j)\in D_i^{tr}} \sigma(\phi_j)L(\boldsymbol{x}_j^T\boldsymbol{\omega}', y_j) + \|\boldsymbol{\omega}'\|^2\end{aligned}$$

where D_i^{tr} and D_i^{val} denote the training and validation datasets on i-th client, espectively. $(\boldsymbol{x}_j, y_j)$ denotes the j-th data and label. $\sigma(.)$ is the sigmoid function, L is the cross-entropy loss, and m is the number of clients in the distributed system. In the MNIST dataset, we set $m = 18$ and $\tau = 10$. As in [35], we assume that the communication delay of each client obeys the heavy-tailed distribution. The proposed AFBO is compared with the state-of-the-art distributed bilevel optimization method FedNest, ADBO and SDBO (Synchronous Distributed Bilevel Optimization, i.e., ADBO without asynchronous setting). The test accuracy results of the 4 algorithms with IID and non-IID datasets are shown in Fig. 1 and Fig. 2. We can observe that the proposed AFBO is the most efficient algorithm. Since the asynchronous setting is considered in AFBO, the server can update its variables once it receives updates from updating clients, It allows multiple local iterations, which makes full use of clients' computing resources.

B. Regularization Coefficient Optimization Task

Following [28], we compare the performance of AFBO with baseline algorithms FedNest, SDBO, and ADBO on the regularization coefficient optimization task using Covertype datasets. The distributed regularization coefficient optimization problem is defined as,

$$\min F(\boldsymbol{\phi}, \boldsymbol{\omega}) = \sum_{i=1}^{m} \frac{1}{|D_i^{val}|} \sum_{(\boldsymbol{x}_j, y_j) \in D_i^{val}} L(\boldsymbol{x}_j^T \boldsymbol{\omega}, y_j)$$

$$s.t.\ \boldsymbol{\omega} = \arg\min_{\boldsymbol{\omega}'} f(\boldsymbol{\phi}, \boldsymbol{\omega}')$$

$$= \sum_{i=1}^{m} \frac{1}{|D_i^{tr}|} \sum_{(\boldsymbol{x}_j, y_j) \in D_i^{tr}} L(\boldsymbol{x}_j^T \boldsymbol{\omega}', y_j) + \sum_{j=1}^{n} \phi_j(\omega_j')$$

where $\boldsymbol{\phi} \in \mathbb{R}^n$, $\omega \in \mathbb{R}^n$ and L, respectively denote the regularization coefficient, model parameter, and logistic loss, and $\omega' = [\omega_1', \ ..., \ \omega_n']$. In the regularization coefficient optimization task with the Covertype dataset, we set $N = 18$ and $\tau = 10$. We also assume that the delay of each client obeys the heavy-tailed distribution. Firstly, we compare the performance of the proposed AFBO, ADBO, SDBO, and FedNest in terms of test accuracy on the Covertype dataset. The results on the Covertype dataset are shown in Fig. 3, which shows that AFBO achieves the best performance among all the schemes. Next, we assume there are at most five stragglers in the distributed system, and the mean of (communication + computation) delay of stragglers is five times the delay of normal clients. The result is shown in Fig. 4. It is found that the efficiency of the synchronous distributed algorithms (FedNest and SDBO) has been significantly affected, while the proposed AFBO and ADBO suffer slightly from the straggler problem since they are asynchronous methods and only consider the updating clients.

C. Convergence under Different Delays

In the previous sections, we assume there are only outer loop asynchronous delays, it shows that AFBO performs similarly to the ADBO and AFBO performs better than other algorithms. In this section, we allow there exists both inner loop and outer loop asynchronous delays. In the MNIST dataset, we set $m = 18$, $\tau = 10$, and $\rho = 10$. In the regularization coefficient optimization task with the Covertype dataset, we set $N = 18$, $\tau = 10$, and $\rho = 10$. As in [35], we assume that the communication delay of each client obeys the heavy-tailed distribution. We first compare AFBO, ADBO, SDBO, FedNest, and Prometheus [36] algorithms' performance on the IID MINIST dataset, then we conduct the experiments on the non-IID MINIST dataset and the Covertype dataset.

Fig. 5, and Fig. 6 show that AFBO performs best among all algorithms. In addition, it shows that all algorithms expect AFBO to suffer an obvious convergence degeneration as there exists an inner loop asynchronous delay. Among them, synchronous algorithms (e.g., SDBO, FedNest, and Prometheus) degenerate more than AFBO and ADBO. Moreover, from Fig. 5, we can find that convergence degeneration is much more obvious in the non-IID MINIST dataset than in the IID MINIST dataset. This is because the bias of using the most recent update of a client is much larger in the non-IID MINIST dataset case than in the IID MINIST dataset case. Intuitively, a reusing of past gradients inducts a new bias of SGD, as some of the SGD gradients use more than others. Finally, Fig. 5 and Fig. 6, show that the convergence degeneration in the Covertype dataset is slighter than in the MINIST dataset.

D. Comparison with AFL

In this subsection, we compare the performance of AFL [29] and AFBO. We use AFL and AFBO to perform the data hyper-cleaning task in IID and non-IID MINIST datasets. We set $m = 18$, $\tau = 10$, and $\rho = 10$ for AFBO algorithm and set $m = 18$ and $\tau = 10$ for AFL algorithm. From Fig 7, and Fig 8, we can see that AFL performs worse in FBO tasks. This is because AFL does not design a distributed estimator for the hyper-parameter ($\nabla\Phi$). The difference between $\frac{1}{M}\sum_{i=1}^{M} \frac{\partial F_i(x,y)}{\partial y}\frac{\partial y}{\partial x}$ and $\frac{\partial F(x,y)}{\partial y}\frac{\partial y}{\partial x}$ leads to the low training accuracy of AFL performing FBO tasks. From the training loss aspect, it shows that the convergence rates and speeds of AFBO and AFL are similar. The reason is that both AFL and AFBO use the most recent gradients stored in the server memory, and both algorithms can achieve linear speed-up. Moreover, both AFL and AFBO consider the effect of the dataset's non-IID degree, thus Fig 7 and Fig 8 show that there is little convergence degeneration as the dataset's non-IID degree increases.

VI. Conclusion

In this paper, we proposed a double-loop scheme Anarchic Federated Bilevel Optimization (AFBO) algorithm, which enables clients to flexibly participate in federated bilevel optimization training according to their heterogeneous and time-varying computation and communication capabilities, and also efficiently by improving utilization of their computation and communication resources. We provided a convergence analysis, which shows that the performance of AFBO matches that of the existing benchmarks. We also conducted simulations using real-world datasets to demonstrate the efficiency of AFBO.

References

[1] S. Ghadimi and M. Wang, "Approximation methods for bilevel programming," *arXiv:1802.02246*, 2018.

[2] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, "A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic," *SIAM Journal on Optimization*, vol. 33, no. 1, pp. 147–180, 2023.

[3] T. Chen, Y. Sun, and W. Yin, "Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] R. Liao, Y. Xiong, E. Fetaya, L. Zhang, K. Yoon, X. Pitkow, R. Urtasun, and R. Zemel, "Reviving and improving recurrent back-propagation," in *International Conference on Machine Learning (ICML)*, 2018.

[5] M. Huang, D. Zhang, and K. Ji, "Achieving linear speedup in non-iid federated bilevel learning," in *International Conference on Machine Learning (ICML)*, 2023.

[6] D. A. Tarzanagh, M. Li, C. Thrampoulidis, and S. Oymak, "Fednest: Federated bilevel, minimax, and compositional optimization," in *International Conference on Machine Learning (ICML)*, 2022.

[7] J. Li, F. Huang, and H. Huang, "Local stochastic bilevel optimization with momentum-based variance reduction," *arXiv:2205.01608*, 2022.

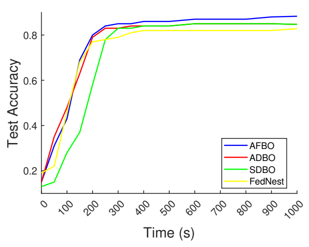

Fig. 1. Test accuracy on IID MINIST.

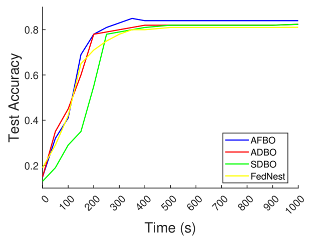

Fig. 2. Test accuracy on non-IID MINIST.

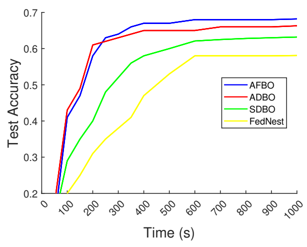

Fig. 3. Test accuracy on Covertype.

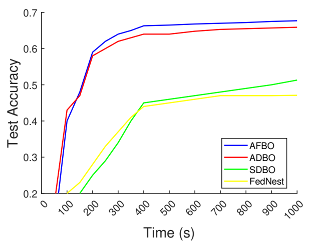

Fig. 4. Test accuracy on Covertype.

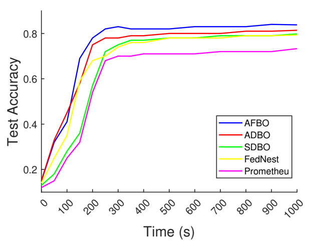

Fig. 5. Test accuracy vs time on non-IID MINIST for heterogenous delays.

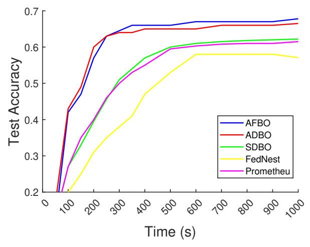

Fig. 6. Test accuracy vs time on Covertype for heterogenous delays.

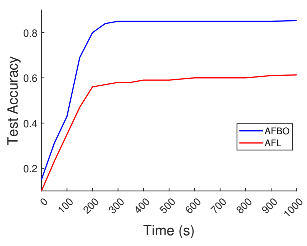

Fig. 7. Test accuracy on IID MINIST.

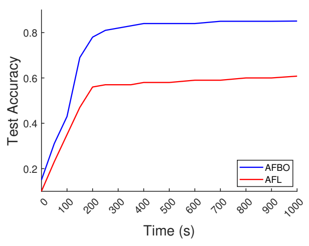

Fig. 8. Test accuracy on non-IID MINIST.

[8] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv:2002.07948*, 2020.

[9] M. Khodak, R. Tu, T. Li, L. Li, M.-F. F. Balcan, V. Smith, and A. Talwalkar, "Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[10] Y. Zeng, H. Chen, and K. Lee, "Improving fairness via federated learning," *arXiv:2110.15545*, 2021.

[11] X. Chen, M. Huang, S. Ma, and K. Balasubramanian, "Decentralized stochastic bilevel optimization with improved per-iteration complexity," in *International Conference on Machine Learning (ICML)*, 2023.

[12] Y. Jiao, K. Yang, T. Wu, D. Song, and C. Jian, "Asynchronous distributed bilevel optimization," in *International Conference on Learning Representations (ICLR)*, 2023.

[13] H. Gao, "On the convergence of momentum-based algorithms for federated bilevel optimization problems," 2022.

[14] F. Huang, "Fast adaptive federated bilevel optimization," 2022.

[15] Y. Yang, P. Xiao, and K. Ji, "Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[16] J. Bracken and J. T. McGill, "Mathematical programs with optimization problems in the constraints," *Operations research*, vol. 21, no. 1, pp. 37–44, 1973.

[17] Y. Zhang, G. Zhang, P. Khanduri, M. Hong, S. Chang, and S. Liu, "Revisiting and advancing fast adversarial training through the lens of bi-level optimization," in *International Conference on Machine Learning (ICML)*, 2022.

[18] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2017.

[19] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design," in *International Conference on Machine Learning (ICML)*, 2021.

[20] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 63–78, 2021.

[21] T. Okuno, A. Takeda, A. Kawana, and M. Watanabe, "Hyperparameter learning via bilevel nonsmooth optimization," *arXiv:1806.01520*, 2018.

[22] Y. Yang, P. Xiao, and K. Ji, "Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning," *arXiv:2305.19442*, 2023.

[23] Y. Huang, Q. Lin, N. Street, and S. Baek, "Federated learning on adaptively weighted nodes by bilevel optimization," *arXiv:2207.10751*, 2022.

[24] Q. Zhang, B. Gu, C. Deng, and H. Huang, "Secure bilevel asynchronous vertical federated learning with backward updating," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[25] J. Li, F. Huang, and H. Huang, "Communication-efficient federated bilevel optimization with local and global lower level problems," *arXiv:2302.06701*, 2023.

[26] ——, "Communication-efficient federated bilevel optimization with global and local lower level problems," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[27] H. Gao, B. Gu, and M. T. Thai, "On the convergence of distributed stochastic bilevel optimization algorithms over a network," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[28] T. Chen, Y. Sun, Q. Xiao, and W. Yin, "A single-timescale method for stochastic bilevel optimization," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

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

[30] 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 (NeurIPS)*, 2020.

[31] "Technical report of AFBO," Technical Report, 2024. [Online]. Available: https://auburn.box.com/s/3sxpvye2q8nldec26h9wuc5vyd1ca12s

[32] 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.

[33] J. Yang, K. Ji, and Y. Liang, "Provably faster algorithms for bilevel optimization," 2021.

[34] X. Chen, M. Huang, and S. Ma, "Decentralized bilevel optimization," *arXiv:2206.05670*, 2022.

[35] A. Cohen, A. Daniely, Y. Drori, T. Koren, and M. Schain, "Asynchronous stochastic optimization robust to arbitrary delays," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[36] Z. Liu, X. Zhang, P. Khanduri, S. Lu, and J. Liu, "Prometheus: Taming sample and communication complexities in constrained decentralized stochastic bilevel learning," in *International Conference on Machine Learning (ICML)*, 2023.