Appendix

Contents

A Analysis and Proof

Here, we provide theoretical analysis in details and present proofs on how FedMDFG can enhance fairness and guarantee the convergence. In all analyses and proofs, we assume that all local objectives in federated learning are differentiable and smooth.

A.1 Analysis of Fair Descent Direction

We first give the analysis of how to calculate a fair descent direction, and then show the proof of Theorem 1 of the paper.

In this paper, we consider federated learning as a multi-objective optimization:

$$\min_{\omega} \ (L_1(\omega), L_2(\omega), \cdots, L_m(\omega)). \tag{1}$$

When the federated learning model is deemed unfair at round t, we propose a fair-enhanced strategy to obtain a fair descent direction, and we establish a temporary problem (2):

$$\min_{\omega} \ (L_1(\omega), L_2(\omega), \cdots, L_m(\omega), L(\omega)^T h^t), \tag{2}$$

where p is the fairness guidance vector, i.e., $p = (1, \cdots, 1)$. h^t is the opposite normalized vector of the projection of p on the normal plane of $L(\omega^t)$, i.e., h^t =normalize$(\frac{p^T L(\omega^t)}{\|L(\omega^t)\|^2} L(\omega^t) - p)$.

To determine the common descent direction d^t of Problem (2), we solve the following Problem:

$$\begin{aligned} &\max_{\lambda} -\tfrac{1}{2}\lambda^T (Q^T Q)\lambda \\ &s.t. \ \ \textstyle\sum_{i=1}^{|\lambda|} \lambda_i = 1, \\ &\qquad \lambda_i \geq 0, \forall i = 1, 2, \cdots, |\lambda|, \end{aligned} \tag{3}$$

where Q=concat$(\nabla L(\omega^t), \nabla L(\omega^t) h^t)$, $\lambda \in \mathbb{R}^{m+1}$. Since $Q^T Q$ is positive-semidefinite, we can always get the optimal solution of Problem (3). In practice, we utilize cvxopt [1], which is a famous solver, to get a numerical solution of Problem (3) by setting an optimality tolerance $\tau = 10^{-6}$.

Based on [2] and the KKT conditon, Problem (3) is the dual form of the following problem, and thus $d^t = -Q\lambda$.

$$\begin{aligned} &(d^t, \alpha^t) = \arg \min_{d^t \in \mathbb{R}^n, \alpha^t \in \mathbb{R}} \alpha^t + \tfrac{1}{2}\|d^t\|^2, \\ &s.t. \quad \nabla L_i(\omega^t)^T d^t \leq \alpha^t, i = 1, \cdots, m, \\ &\qquad (\nabla L(\omega^t) h^t)^T d^t \leq \alpha^t. \end{aligned} \tag{4}$$

According to [2], the solution of Problem (4) will satisfy:

1. If ω^t is Pareto critical, then $d^t = \mathbf{0}$ and $\alpha^t = 0$.
2. If ω^t is not Pareto critical, then

$$\begin{aligned} &\alpha^t \leq -(1/2)\|d^t\|^2 < 0, \\ &\nabla L_i(\omega^t)^T d^t \leq \alpha^t, i = 1, \cdots, m, \\ &(\nabla L(\omega^t) h^t)^T d^t \leq \alpha^t. \end{aligned} \tag{5}$$

Hence, when ω^t is not Pareto critical in Problem (2), then d^t is a common descent direction, i.e., d^t not only can drive $L_i(\omega^{t+1}) < L_i(\omega^t), i = 1, \cdots, m$, but also is able to force $L(\omega^{t+1})^T h^t < L(\omega^t)^T h^t$.

Theroem 1 of the paper states that we can always find such a d^t when ω^t is not Pareto critical in Problem (1). This also implies that there is no situation in which ω^t is Pareto critical in Problem (2) but not Pareto critical in Problem (1).

We present Theorem 1 of the paper as follows for convenience:

Theorem 1. For a non-Pareto critical solution ω^t of Problem (1), there always exists a direction d^t, such that $\nabla L_i(\omega^t)^T d^t < 0, i = 1, \cdots, m$ and $(\nabla L(\omega^t) h^t)^T d^t < 0$.

Proof: Denote $v{=}\nabla L(\omega^t)^T d^t$, we have $v{<}\mathbf{0}$ according Lemma 1 (mentioned in the paper).

Besides, as ω^t is not Pareto critical for Problem (1), we have $L(\omega^t){>}\mathbf{0}$.

Since $h^t \perp L(\omega^t)$, not all element of h^t is smaller than 0.

Therefore, there always exists a d^t that $vh^t{<}0$, i.e., $(\nabla L(\omega^t) h^t)^T d^t < 0$. Q.E.D.

So there exists $\eta_0 > 0$ such that for $\omega^{t+1} = \omega^t + \eta_0 d^t$, it satisfies $L_i(\omega^{t+1}) < L_i(\omega^t), \forall i$ and $L(\omega^{t+1})^T h^t < L(\omega^t)^T h^t = 0$. Now we discuss why d^t can drive fairer.

Denote ϕ_f as the fairer area formed by all the possible local objective vectors L_f that satisfying $\varphi(L_f, p) < \varphi(L(\omega^t), p)$. This fairer area can be seen in Fig. 2 (b) of the paper. Denote Π_r as a plane formed by all the local objective vectors $L(\omega^t + \eta_r d^t), \forall \eta_r > 0$. Since $L(\omega^{t+1})^T h^t < L(\omega^t)^T h^t = 0$, Π_r must lie across ϕ_f, so there must exist $\eta^t > 0$ that $L(\omega^t + \eta^t d^t)$ is inside ϕ_f. Thus, d^t can drive the model fairer. In conclusion, d^t obtained by the above method is a fair descent direction.

Remark. To prevent all local objectives $L_i(\omega)$ from being identical, we only utilize the fair-enhanced strategy when the model is judged unfair, i.e., out of the tolerable fair area. Otherwise, if all local objectives $L_i(\omega)$ are the same, no direction can force the model fairer, which will affect the model to be trained further.

A.2 Proof of Convergence

Assume that all clients are selected at each round, we prove the convergence of FedMDFG in two parts.

Lemma A.2 [2]. If L is differentiable and $\nabla L(\omega) d < \mathbf{0}$, then there exists some $\epsilon > 0$ (which may depends on ω, d, and β) such that

$$L(\omega + \eta_r d) < L(\omega) + \beta \eta_r \nabla L(\omega) d, \tag{6}$$

for any $\eta_r \in (0, \epsilon)$.

Lemma A.2 indicates that we can always find a step size that make the Armijo condition [3] satisfied.

(Part 1.) If the Armijo condition is satisfied in the step size line search at each round, then according to Theorem 2 of the paper, which is also presented as follows, FedMDFG can converge to a Pareto critical point of Problem (1).

Theorem 2. Suppose that the Armijo condition in the step size line search process is satisfied, then every accumulation point of the sequence $(w^{(t)})_t$ produced by FedMDFG is a Pareto critical point.

Inspired by [2], we give the proof as follows:

Proof: Let y be an accumulation point of the sequence $(\omega^{(t)})_t$ and let $d(y)$ and $\alpha(y)$ be the solution of the following problem at y.

$$\begin{aligned} &(d^t, \alpha^t) = \arg \min_{d^t \in \mathbb{R}^n, \alpha^t \in \mathbb{R}} \alpha^t + \tfrac{1}{2}\|d^t\|^2, \\ &s.t. \quad \nabla L_i(\omega^t)^T d^t \leq \alpha^t, i = 1, \cdots, m. \end{aligned} \tag{7}$$

According to Lemma 1 mentioned in the paper, it is enough to prove that $\alpha(y) = 0$. Obviously, the sequence $(L(\omega^{(t)}))_t$ is componentwise strictly decreasing and satisfies

$$\lim_{t\to\infty} L(\omega^{(t)}) = L(y). \tag{8}$$

Thus,

$$\lim_{t\to\infty} ||L(\omega^{(t)}) - L(\omega^{(t+1)})|| = \mathbf{0}. \tag{9}$$

However,

$$L(\omega^{(t)}) - L(\omega^{(t+1)}) \geq -\eta_t \beta \nabla L(\omega^{(t)})^T d^{(t)} \geq \mathbf{0}, \tag{10}$$

and thus

$$\lim_{t\to\infty} \eta_t \nabla L(\omega^{(t)})^T d^{(t)} = \mathbf{0}. \tag{11}$$

Take a subsequence $(\omega^{(t_u)})_u$ converging to y. Consider two cases:

$$\lim_{u\to\infty} \sup \eta_{t_u} > 0, \tag{12}$$

and

$$\lim_{u\to\infty} \sup \eta_{t_u} = 0. \tag{13}$$

[Case 1] There Exists a subsequence $(\omega^{(t_l)})_l$ converging to y and satisfying

$$\lim_{l\to\infty} \eta_{t_l} = \bar{\eta} > 0. \tag{14}$$

By (11), we can conclude:

$$\lim_{l\to\infty} \nabla L(\omega^{(t_l)})^T d^{t_l} = 0, \tag{15}$$

which also implies that:

$$\lim_{l\to\infty} \alpha(\omega^{(t_l)}) = 0. \tag{16}$$

Since $\omega \to \alpha(\omega)$ is continuous, then we conclude that $\alpha(y) = 0$, y is Pareto critical.

[Case 2] According to [2], $(d^{(t_u)})_u$ is bounded. Take a subsequence $(\omega^{(t_r)})_r$ of $(\omega^{(t_u)})_u$, such that the sequence $(d^{(t_r)})_r$ also converges to some $\bar{d}$.

For all r we have

$$\max_i (L(\omega^{(t_r)})^T d^{(t_r)})_i \leq \tau\alpha(d^{(t_r)}) < \mathbf{0}. \tag{17}$$

Passing onto the $\lim r \to \infty$, we obtain

$$\frac{1}{\tau}\max_i (L(y)^T \bar{d})_i \leq \alpha(y) \leq \mathbf{0}. \tag{18}$$

Take some $q \in \mathbb{N}$. For r large enough,

$$\eta_{t_r} < \frac{1}{2^q}, \tag{19}$$

meaning that the Armijo condition is not satisfied for $\eta = \frac{1}{2^q}$, i.e., $\exists r$

$$L(\omega^{(t_r)} + \frac{1}{2^q} d^{(t_r)}) > L(\omega^{(t_r)}) + \beta\frac{1}{2^q}\nabla L(\omega^{(t_r)})^T d^{(t_r)}. \tag{20}$$

Passing onto the $\lim r \to \infty$, we have

$$L_j(y + \frac{1}{2^q}\bar{d}) \geq L_j(y) + \beta\frac{1}{2^q}(\nabla L(y)^T \bar{d})_j, \tag{21}$$

for at least one $j \in \{1, ..., m\}$. This inequality is true for any $q \in \mathbb{N}$. Together with Lemma A.2, we have:

$$\max_i (\nabla L(y)^T \bar{d})_i \geq \mathbf{0} \tag{22}$$

which implies $\alpha(y) = 0$. Hence y is Pareto critical. Q.E.D.

(Part 2.) If the learning rate η is set so large in practice that the confirmed step size η^t cannot satisfy the Armijo condition but at least gratifies the stopping criterion $\|L(\omega^t + \eta^t d^t)\|_1 < \|L(\omega^t)\|_1$, then because $(|L(\omega^t)||_1)_t$ is decreasing and bounded by 0, in accordance with Monotone Convergence Theorem [4], FedMDFG can converge to a local optimum of the following problem:

$$\min_\omega \sum_{i=1}^{m} L_i(\omega), \tag{23}$$

which is equivalent to prove that it can converge to a local optimum of the federated learning problem:

$$\min_\omega \mathbf{G}(L_1(\omega), L_2(\omega), \cdots, L_m(\omega)), \tag{24}$$

where $\mathbf{G}$ is an aggregation function.

A.3 Scale the Length of the Direction

Before using the obtained direction d^t, we scale its length to that of another direction d_r calculated by applying a method similar to FedSGD [5], i.e., $d_r = -\sum_i^{|S_t|} \frac{1}{|S_t|} g_i$. The reason is $\|d^t\| \leq \|g_r\|$ is always true when $Q = \nabla L(\omega^t)$.

Proof: Denote $\lambda^* \in \mathbb{R}^m$ is the optimal solution of Problem (3) where $Q = \nabla L(\omega^t)$. Denote $\lambda_r \in \mathbb{R}^m$ that $d_r = -Q\lambda_r$. Since the objective of Problem (3) is equivalent to minimize $\|d\|^2$, λ^* is the optimal solution while λ_r is not, then $\|d^t\|^2 \leq \|d_r\|^2$ holds true and thus $\|d^t\| \leq \|g_r\|$ is always true.

B Additional and Complete Experiment Results

Neural Network Models

We list the network architectures of the models utilized in the paper.

(1) CNN for MNIST:

```
CNN(
    (conv2d_1): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (relu): ReLU()
    (max_pooling): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (conv2d_2): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (relu): ReLU()
    (flatten): Flatten(start_dim=1, end_dim=-1)
    (linear_1): Linear(in_features=3136, out_features=512, bias=True)
    (relu): ReLU()
    (linear_2): Linear(in_features=512, out_features=10, bias=True)
)
```

(2) CNN for CIFAR-10 and CIFAR-100:

```
CNN(
    (encoder): Sequential(
        (0): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
        (1): ReLU()
        (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (3): Conv2d(64, 64, kernel_size=(5, 5), stride=(1, 1))
        (4): ReLU()
        (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (decoder): Sequential(
        (0): Linear(in_features=1600, out_features=384, bias=True)
        (1): Dropout(p=0.2, inplace=False)
        (2): ReLU()
        (3): Linear(in_features=384, out_features=192, bias=True)
        (4): Dropout(p=0.5, inplace=False)
        (5): ReLU()
        (6): Linear(in_features=192, out_features=10, bias=True)
    )
)
```

(3) Multilayer perceptron (MLP) for Fashion MNIST:

```
MLP(
    (fc1): Linear(in_features=784, out_features=200, bias=True)
    (relu): ReLU()
    (fc2): Linear(in_features=200, out_features=200, bias=True)
    (relu): ReLU()
    (fc3): Linear(in_features=200, out_features=10, bias=True)
)
```

(4) LeNet for CIFAR-10:

```
LeNet(
    (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))
    (relu): ReLU()
    (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
    (relu): ReLU()
    (fc1): Linear(in_features=400, out_features=120, bias=True)
    (relu): ReLU()
    (fc2): Linear(in_features=120, out_features=84, bias=True)
    (relu): ReLU()
    (fc3): Linear(in_features=84, out_features=10, bias=True)
)
```

This model is utilized in Appendix B.2 to compare the efficiency of the algorithms under different network models.

Datasets

We summarize the dartasets and tasks in the Table. 1, and describe the data partitions in detail.

Table 1: Summary of datasets

Datasets	Models	Tasks
MNIST [6]	CNN	10-class classification
Fashion MNIST [7]	MLP	10-calss classification
CIFAR-10 [8]	CNN	10-class classification
CIFAR-100 [8]	CNN	100-class classification

We design 3 different data partitions scenarios:

- Normal case: Sort all data records based on their classes, divide them into 200 shards, and assign each of 100 clients 2 shards randomly without replacement.
- Mutex case: Each client has and only as data of one class.

- Unbalanced case: For CIFAR-10 and FMNIST, 10 clients are randomly split into five groups: group 1 has one client; Each of group $2, 3, 4$ has two clients; the last group has 3 clients. Then, we aggregate all clients from each group into a new client.

Baselines

We list the baselines used in our experiments.

- FedAvg [5]: A traditional federated learning algorithm.
- AFL [9]: Strive for decent fairness by protecting the worst-case performance on clients.
- qFedAvg [10]: Enhance fairness by employing a fair resource allocation mechanism.
- FedFV [11]: Mitigate the conflicts across the local gradients to increase fairness.
- TERM [12]: Reweight the local model parameters by considering flexible trade-offs between accuracy and fairness.
- Ditto [13]: A federated learning method that aims to enhance fairness and robustness through personalization.
- FedMGDA+ [14]: Aim to find a common descent direction through an extended multiple gradient descent algorithm.

Hyper-parameters

The full hyper-parameters of the algorithms are listed as Table. 2. The first one of each parameter set is regarded as the default for each algorithm. We observe the learning rate $\eta \in \{0.01, 0.05, 0.1\}$ with decay 0.999 per round. We take the best performance of each method for the comparison.

Table 2: Hyper-parameters of different methods

Method	Hyper-parameters
AFL	$\eta_\lambda \in \{0.1, 0.01\}$
qFedAvg	$q \in \{0.1, 1.0, 5.0\}$
FedFV	$\alpha \in \{0.1, 0.2\}$, $\tau \in \{1, 2\}$
TERM	$t \in \{1, 5\}$
Ditto	$\lambda \in \{0.1, 1.0\}$
FedMGDA+	$\epsilon \in \{0.1, 1.0\}$
FedMDFG	$\theta \in \{\pi/16, \pi/32\}$, $s \in \{5, 3, 1\}$

B.1 Accuracy and Fairness

The complete results of Table 2 from the publication are listed in Table. 5. Moreover, we compare the algorithms on Fashion MNIST with batch size 50. (See Table. 6.) The reported results are averaged over 5 runs with different random seeds. For the normal case, the mutex case, and the unbalanced case, 10%, 50%, and 60% of clients are randomly sampled at each communication round.

Fairness indicator of test accuracy is calculated by $arccos(\frac{A \cdot p}{\|A\|\|p\|})$, where $p = \mathbf{1}$ and A is a vector that A_i represents the test accuracy of client i. It indicates the angle (radian) between the test accuracy vector A and the fair guidance vector p. A model is fairer if its fairness indicator value is smaller.

B.2 Efficiency and Convergence

The experiments of convergent efficiency are conducted in the mutex case with 100% clients online at each communication round, which can show the performance without being affected by clients' dropout.

We first present complete figures of the efficiency and convergence experiments, as seen in Fig. 1. Note that in Fig. 1 (b), since FedFV suffers a significant reduction on the performance, we further dedign a smaller learning rate ($\eta = 0.001$) to stabilize it, denoted as FedFV'. But that will slow down the convergence.

We further test the convergence efficiency in the mutex case on Fashion MNIST by using MLP (shown in Fig. 2) and CIFAR-10 by using CNN. LeNet, a different type of CNN with more limited parameters, is also used in the comparison in order to assess the impact of the network we employ. The results can be seen in Fig. 3. Note that for Fig. 3 (b), the learning rate of FedFV (denoted as FedFV') is tuned to be 0.0001, because its performance is quite unbalance and even cannot converge if we set $\eta \in \{0.001, 0.01, 0.05, 0.1\}$. From Fig. 3 (a) and (b) we can observe that FedMDFG have the best performance by using different network models.

Note that the performance of the algorithm's model can be severely shaken in the mutex case because the distribution of data on clients is quite different from one another and the algorithm cannot guarantee that the direction is fair descent and the step size is appropriate. The convergence would occur considerably more slowly if we tuned a smaller learning rate, as was already mentioned. FedMDFG, on the other hand, has the benefit of relying less on artificially adjusting the learning rate because the step size line search offers a means of determining the best step size to use for better training the model. However, The previous methods cannot use the step size line search since they cannot guarantee the direction is fair descent.

B.3 Robustness

We further show more experimental results on robustness. The experiments settings follow the main test, while setting 5% dishonest clients. Table. 3 shows the results, where "clean" means all clients are honest. Compared with the results under 10% dishonest clients' attacks, most of the compared algorithms have better performances. But some of them still obtain a model with only about 10% average test accuracy.

Table 3: Average test accuracy (Fairness indicator) under clients' attacks in the normal case on FMNIST.

	clean	A1	A2	A3
FedAvg	.859(.100)	.490(.381)	.100(1.11)	.612(.271)
AFL	.834(.087)	.100(1.13)	.100(1.13)	.773(.121)
qFedAvg	.855(.106)	.100(1.10)	.366(.740)	.625(.260)
FedFV	.862(.091)	.636(.285)	.191(.958)	.652(.311)
TERM	.672(.201)	.521(.374)	.466(.570)	.673(.202)
Ditto	.860(.095)	.550(.332)	.842(.117)	.669(.278)
FedMGDA+	.714(.216)	.681(.267)	.715(.212)	.100(1.11)
FedMDFG	**.875(.065)**	**.878(.079)**	**.875(.067)**	**.871(.067)**

B.4 Direction comparison

We compare the directions calculated by gradient-based federated algorithms: FedMGDA+, FedFV, and our proposed algorithm. We randomly generate a local gradient for each client each time, and calculate the rate of successfully obtaining a common descent direction in 1 million tries. Table. 4 shows the results, where m represents the number of clients, and n is the dimension of model's parameters. We can observe that FedMDFG can 100% find common descent directions in the five cases.

Table 4: Probability of finding a common descent directions in different cases.

	$m = 2$ $n = 2$	$m = 3$ $n = 3$	$m = 10$ $n = 100$	$m = 100$ $n = 10$	$m = 100$ $n = 1000$
FedMGDA+	1.00	0.56	1.00	0.00	0.97
FedFV	1.00	0.76	1.00	0.00	1.00
FedMDFG	1.00	1.00	1.00	1.00	1.00

References

[1] Martin Andersen, Joachim Dahl, and Lieven Vandenberghe. Cvxopt: Convex optimization. *Astrophysics Source Code Library*, pages ascl–2008, 2020.

[2] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51(3):479–494, 2000.

[3] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[4] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

[5] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[10] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[11] Z. Wang, X. Fan, J. Qi, C. Wen, and R. Yu. Federated learning with fair averaging. In *IJCAI 2021*, 2021.

[12] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. In *International Conference on Learning Representations*, 2021.

[13] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368, 2021.

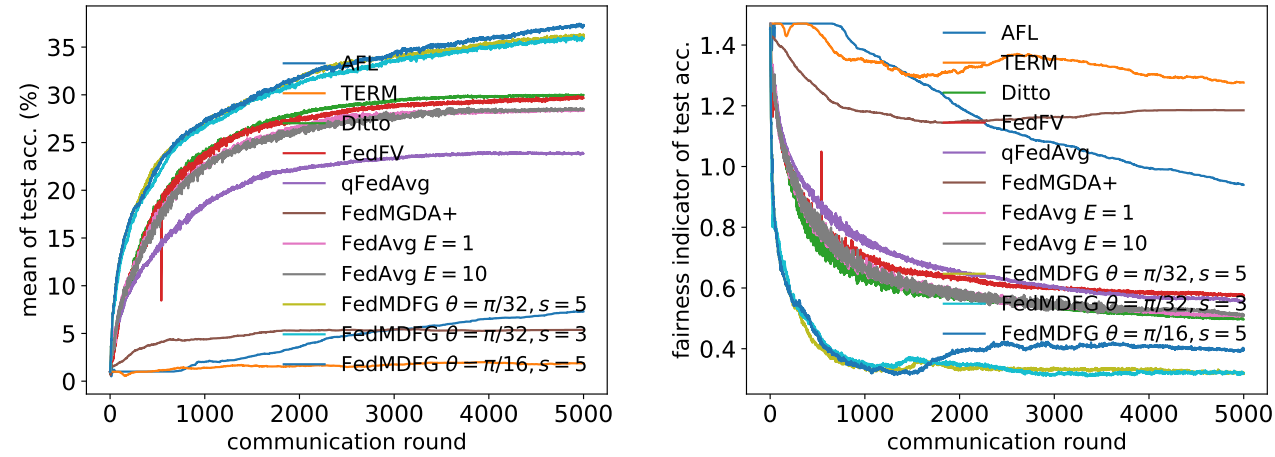
[14] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9(4):2039–2051, 2022.

Table 5: The average, the fairness indicator, the standard deviation (SD), the minimum, and the maximum of test accuracy across clients in normal, mutex, and unbalanced cases on CIFAR-10 with batch size 50 over 2000 rounds. Local epochs $E = 1$.

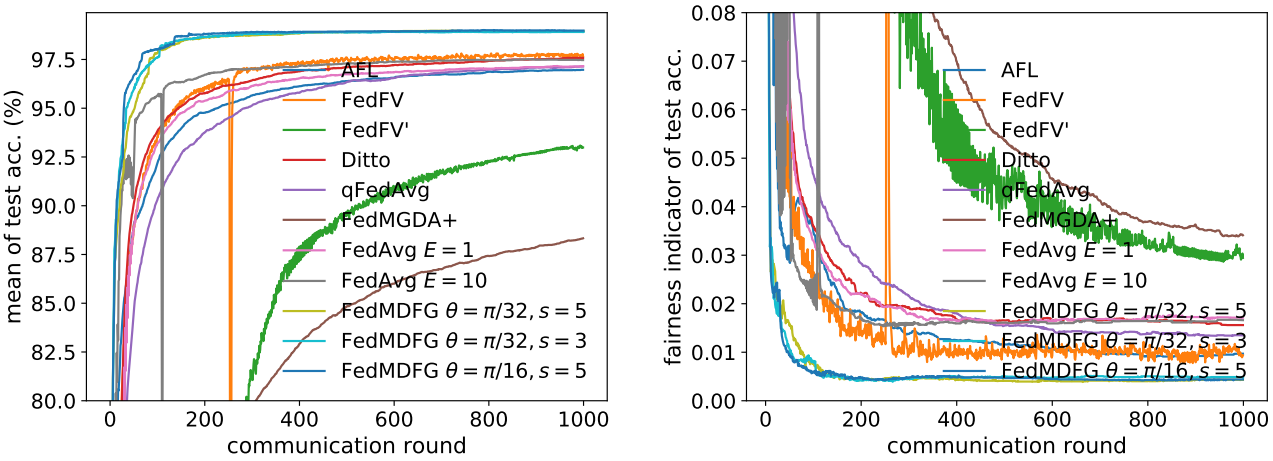
	Normal					Mutex					Unbalanced				
Methods	acc.	fair	SD	min	max	acc	fair	SD	min	max	acc.	fair	SD	min	max
FedAvg	.683	.202	.137	.300	.870	.323	.728	.288	.300	.688	.547	.473	.280	.178	.907
AFL$\|_{\lambda=0.01}$	.681	.141	.096	.470	.880	.328	.589	.219	.111	.824	.526	.538	.314	.097	.876
AFL$\|_{\lambda=0.1}$	.689	.139	.096	.460	.880	.278	.637	.206	.014	.607	.525	.374	.206	.253	.798
qFedAvg$\|_{q=0.1}$	.667	.174	.117	.350	.860	.408	.339	.144	.158	.676	.558	.451	.270	.153	.839
qFedAvg$\|_{q=1.0}$	.595	.161	.097	.260	.780	.303	.538	.181	.103	.687	.535	.255	.139	.398	.778
qFedAvg$\|_{q=5.0}$	.446	.182	.082	.260	.650	.301	.291	.118	.162	.435	.420	.535	.249	.067	.835
FedFV$\|_{a=0.1,\tau=1}$	.684	.177	.122	.330	.890	.374	.582	.246	.022	.771	.561	.307	.178	.348	.763
FedFV$\|_{a=0.1,\tau=2}$	.689	.171	.119	.370	.900	.347	.600	.237	.038	.849	.554	.259	.147	.334	.728
FedFV$\|_{a=0.2,\tau=2}$	.685	.183	.127	.380	.910	.330	.606	.229	.018	.646	.532	.432	.237	.181	.763
TERM$\|_{t=1}$	.529	.176	.094	.310	.740	.338	.297	.115	.186	.515	.579	.263	.156	.425	.858
TERM$\|_{t=5}$	.520	.156	.082	.310	.700	.309	.389	.126	.141	.614	.557	.261	.149	.335	.724
Ditto$\|_{\lambda=0.1}$	.609	.201	.124	.320	.820	.310	.662	.242	.019	.690	.537	.470	.273	.169	**.920**
Ditto$\|_{\lambda=0.5}$	.614	.200	.124	.280	.880	.290	.716	.252	.000	.714	.534	.355	.198	.217	.759
FedMGDA+$\|_{\epsilon=0.1}$	.507	.201	.103	.290	.760	.324	.463	.161	.122	.639	.522	.286	.154	.252	.666
FedMGDA+$\|_{\epsilon=1.0}$	.505	.201	.103	.280	.740	.324	.462	.161	.125	.640	.524	.367	.201	.221	.810
FedMDFG$\|_{\theta=\frac{\pi}{32},s=5}$	**.746**	**.100**	**.075**	.550	**.910**	.690	**.165**	**.115**	.466	.881	.659	**.174**	**.116**	.505	.860
FedMDFG$\|_{\theta=\frac{\pi}{16},s=5}$	.743	.103	.077	**.580**	.890	.685	.209	.146	**.480**	**.894**	**.661**	.181	.121	**.551**	.878
FedMDFG$\|_{\theta=\frac{\pi}{16},s=3}$	.743	.101	.076	.560	.910	**.691**	.203	.126	.398	.867	.659	.178	.118	.513	.822
FedMDFG$\|_{\theta=\frac{\pi}{16},s=1}$	.742	.101	.075	.580	.890	.689	.202	.141	.439	.859	.660	.195	.130	.415	.803

Table 6: The average, the fairness indicator, the standard deviation (SD), the minimum, and the maximum of test accuracy across clients in normal, mutex, and unbalanced cases on Fashion MNIST with batch size 50 over 2000 rounds. Local epochs $E = 1$.

	Normal					Mutex					Unbalanced				
Methods	acc.	fair	SD	min	max	acc.	fair	SD	min	max	acc.	fair	SD	min	max
FedAvg	.850	.108	.092	.630	.990	.760	.324	.256	.166	.970	.763	.239	.186	.420	.937
AFL$\|_{\lambda=0.01}$	.842	.128	.108	.580	.990	.696	.281	.201	.427	.995	.777	.232	.183	.470	**.976**
AFL$\|_{\lambda=0.1}$	.836	.156	.131	.450	.990	.727	.345	.261	.145	.996	.690	.342	.246	.344	.964
qFedAvg$\|_{q=0.1}$	.848	.111	.095	.590	.990	.767	.248	.195	.326	.949	.759	.151	.116	.546	.879
qFedAvg$\|_{q=1.0}$	.824	.110	.091	.530	.970	.714	.242	.176	.297	.896	.781	.083	.066	.680	.872
qFedAvg$\|_{q=5.0}$	.740	.127	.095	.520	.910	.629	.257	.165	.415	.951	.646	.068	.044	.568	.693
FedFV$\|_{a=0.1,\tau=1}$	.847	.116	.099	.570	.990	.783	.314	.254	.132	.969	.780	.139	.109	.590	.927
FedFV$\|_{a=0.1,\tau=2}$	.861	.117	.101	.510	.990	.778	.252	.201	.280	.970	.795	.232	.188	.466	.965
FedFV$\|_{a=0.2,\tau=2}$	.858	.115	.099	.540	.990	.695	.358	.260	.034	.974	.760	.246	.191	.397	.959
TERM$\|_{t=1}$	.809	.128	.104	.470	.970	.757	.216	.166	.377	.908	.734	.111	.082	.597	.833
TERM$\|_{t=5}$	.806	.103	.083	.580	.940	.747	.183	.138	.475	.899	.749	.109	.082	.676	.853
Ditto$\|_{\lambda=0.1}$	.852	.110	.941	.590	.970	.753	.274	.212	.238	.943	.737	.180	.134	.510	.851
Ditto$\|_{\lambda=0.5}$	.856	.117	.101	.470	.990	.705	.394	.293	.115	.926	.730	.186	.137	.541	.926
FedMGDA+$\|_{\epsilon=0.1}$	.795	.164	.132	.310	.970	.633	.377	.251	.217	.963	.765	.214	.166	.469	.970
FedMGDA+$\|_{\epsilon=1.0}$	.800	.161	.130	.330	.970	.635	.355	.236	.247	.961	.778	.207	.164	.500	.960
FedMDFG$\|_{\theta=\frac{\pi}{32},s=5}$	.875	.077	.067	**.710**	**.990**	.851	**.108**	**.092**	.691	.971	.839	.089	.075	.698	.914
FedMDFG$\|_{\theta=\frac{\pi}{16},s=5}$	.876	.077	.068	.720	.990	.855	.112	.097	**.707**	**.980**	**.838**	.088	**.074**	.702	.919
FedMDFG$\|_{\theta=\frac{\pi}{16},s=3}$	.878	.075	.066	.690	.990	.851	.120	.103	.633	.984	.848	.076	.065	.737	.927
FedMDFG$\|_{\theta=\frac{\pi}{16},s=1}$	**.879**	**.075**	**.066**	.690	.980	**.861**	.113	.097	.680	.980	.848	**.067**	.568	**.752**	.921

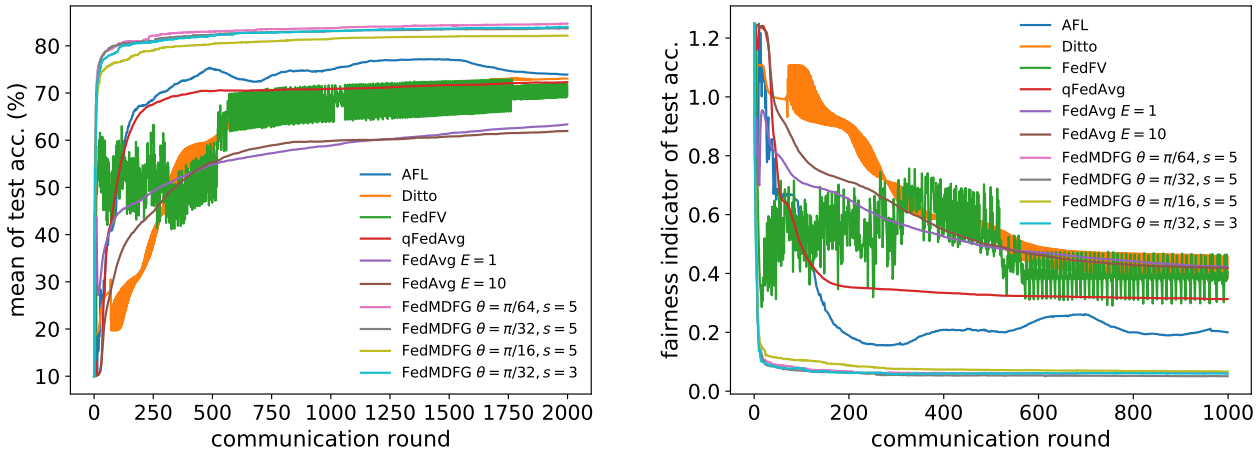


(a) CIFAR-100

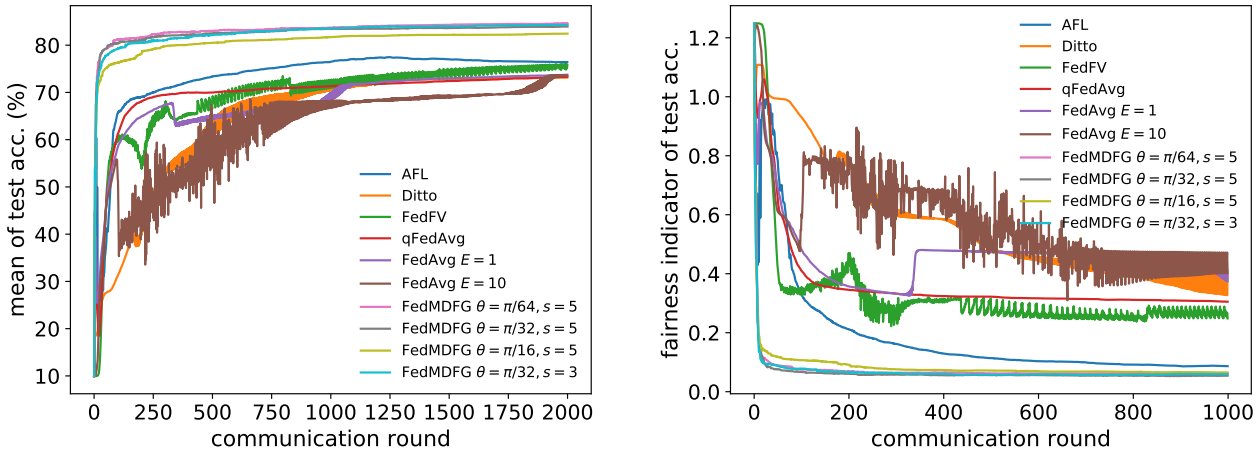


(b) MNIST

Figure 1: The mean (left) and the fairness indicator (right) of the test accuracy across all clients in the mutex case on (a) CIFAR-100 and (b) MNIST with batch size 200 and local epochs $E = 1$. 100% of clients are selected at each round. The reported results are averaged over 5 runs with different random seeds.

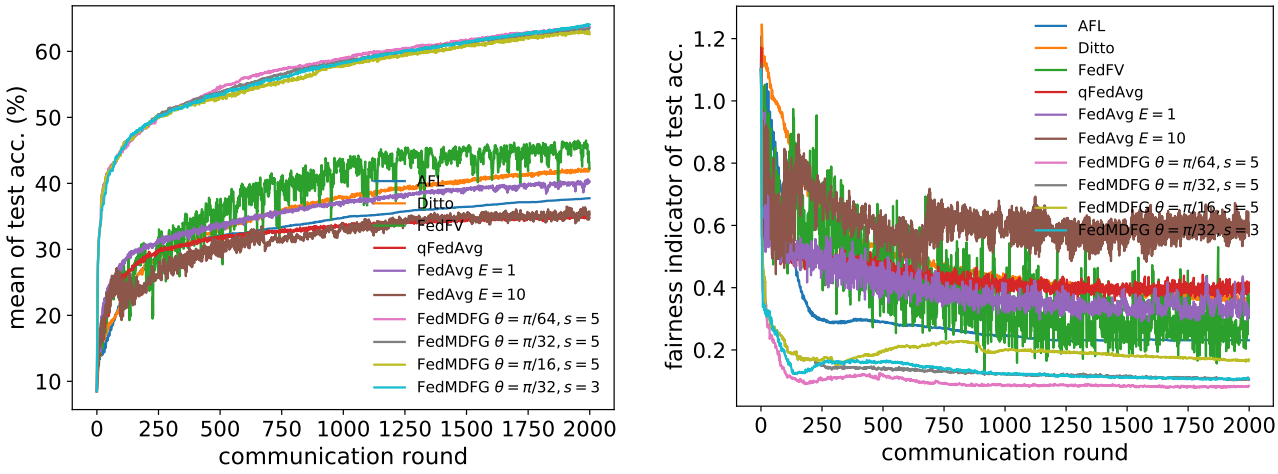


(a) Fashion MNIST with batch size 50.

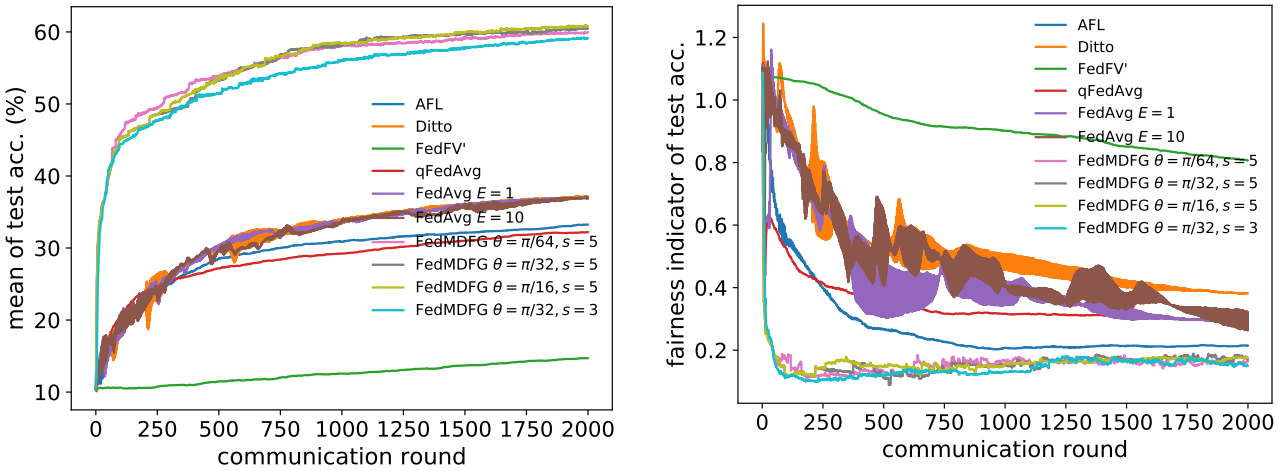


(b) Fashion MNIST with batch size 200.

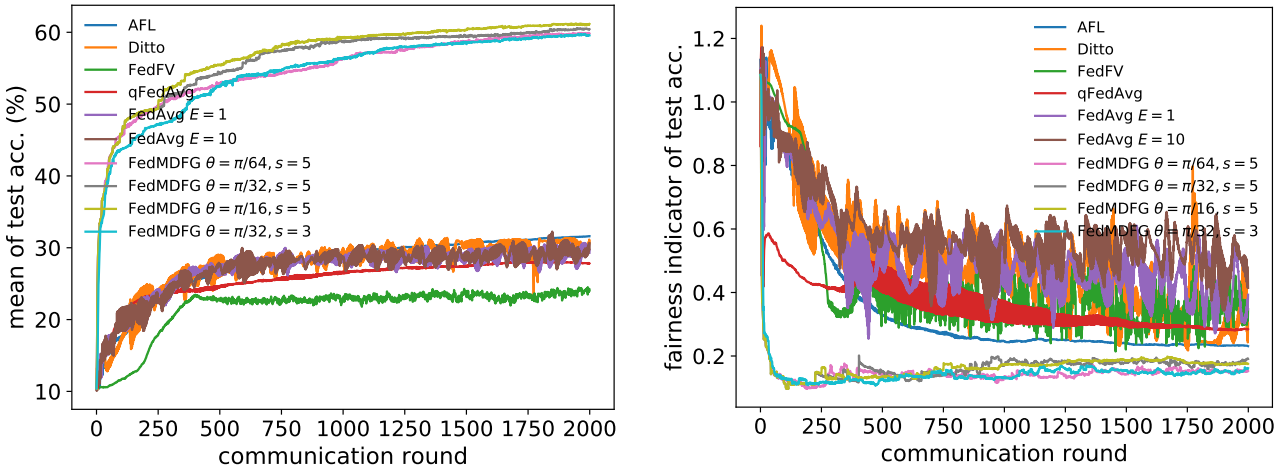
Figure 2: The mean (left) and the fairness indicator (right) of the test accuracy across all clients in the mutex case on Fashion MNIST. Local epochs $E = 1$. 100% of clients are selected at each round. The reported results are averaged over 5 runs with different random seeds.



(a) CIFAR-10 batch size 200, CNN



(b) CIFAR-10 batch size 200, LeNet



(c) CIFAR-10 batch size 50, LeNet

Figure 3: The mean (left) and the fairness indicator (right) of the test accuracy across all clients in the mutex case with local epochs $E = 1$. 100% of clients are selected at each round. The reported results are averaged over 5 runs with different random seeds.