# Federated Learning with Convex Global and Local Constraints

**Chuan He** HE000233@UMN.EDU
**Le Peng** PENG0347@UMN.EDU
**Ju Sun** JUSUN@UMN.EDU
*University of Minnesota, USA*

## Abstract

This paper considers federated learning (FL) with constraints where the central server and all local clients collectively minimize a sum of local objective functions subject to inequality constraints. To train the model without moving local data at clients to the central server, we propose an FL framework that each local client performs multiple updates using the local objective and local constraints, while the central server handles the global constraints and performs aggregation based on the updated local models. In particular, we develop a proximal augmented Lagrangian (AL) based algorithm, where the subproblems are solved by an inexact alternating direction method of multipliers (ADMM) in a federated fashion. Under mild assumptions, we establish the worst-case complexity bounds of the proposed algorithm. Our numerical experiments demonstrate the practical advantages of our algorithm in solving linearly constrained quadratic programming and performing Neyman-Pearson classification in the context of FL.

## 1. Introduction

Federated learning (FL) has emerged as a prominent distributed machine learning paradigm, finding extensive application across diverse domains. While FL has gained extensive adoption, there have been few developments on FL algorithms capable of handling constraints that incorporate desired properties and prior knowledge. This is despite the frequent occurrence of constrained optimization problems in modern learning tasks (e.g., see [39]). In particular, Neyman-Pearson classification [66] and learning with fairness constraints [2, 13, 49] are two important examples of constrained machine learning problems. We defer a review of constrained optimization problems in modern machine learning to the Appendix B, and associated algorithms in Appendix A.

In the FL literature, many efforts have been devoted to mitigating class imbalance [65] and improving model fairness [15, 18, 19] through the application of constrained optimization models. Nevertheless, these algorithms are often specialized to particular use cases and suffer from a lack of computational complexity guarantees for achieving consensus, optimality, and feasibility in their solutions. The main goals of this paper are twofold: (1) to investigate a general optimization problem with convex constraints in an FL setting; (2) to develop an FL algorithm with complexity guarantees for finding its solution. Specifically, we consider the following general optimization formulation of FL problems with convex global and local constraints [1]:

$$\min_{w}\left\{\sum_{i=1}^{n} f_i(w)+h(w)\right\} \quad \text{s.t.} \quad \underbrace{c_0(w)\le 0}_{\text{global constraint}}, \quad \underbrace{c_i(w)\le 0,\ \ 1\le i\le n}_{\text{local constraints}}, \tag{1}$$

where the functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $1 \le i \le n$, and the mappings $c_i : \mathbb{R}^d \to \mathbb{R}^{m_i}$, $0 \le i \le n$, are convex and continuously differentiable, and $h : \mathbb{R}^d \to (-\infty, \infty]$ is a simple closed convex function. The convexity assumption is necessary for our initial theoretical exploration of FL with constraints. We also explore the applicability of our FL algorithm to classification tasks with nonconvex fairness constraints in Appendix F.3.

The global constraint in (1), namely $c_0(w) \le 0$, refers to a constraint that can be directly accessed by the central server. The local constraints in (1), namely $c_i(w) \le 0$ for $1 \le i \le n$, refer to constraints that depend on the local data that clients used for training the model. Throughout this paper, we assume that

*for each $1 \le i \le n$, the local objective $f_i$ and local constraint $c_i$ are handled solely by the local client $i$, and the central server has access to the global constraint $c_0$.*

This assumption generalizes the one commonly imposed for unconstrained FL, where each local objective function is solely handled by one local client. In addition, our model (1) is tailored for scenarios where local clients have enough amount of reliable data points to establish their own local constraints. Meanwhile, to enhance generalization property, the central server forms a global constraint by incorporating certain public or external data points. Additionally, it is noteworthy that solving an FL problem with $n$ local constraints, such as $c_i(w) \le r$, $1 \le i \le n$, can yield a feasible solution for the coupled constraints involving data points from all local clients, such as $1/n \sum_{i=1}^{n} c_i(w) \le r$.

Due to the sophistication of the constraints in problem (1), existing FL algorithms face challenges when attempting to apply or extend them directly to solve (1). For example, a natural approach for this problem is to adopt existing FL algorithms to minimize the quadratic penalty function associated with (1). However, to ensure global convergence to a solution for (1), it is often necessary to minimize a sequence of penalty functions with sufficiently large penalty parameters, rendering the solution process highly unstable and inefficient (e.g., see [54]). Moreover, in the centralized setting, Lagrangian methods are frequently employed for constrained optimization in deep learning (e.g., see [16]). However, these methods often require careful tuning of initial multipliers and step-sizes for the multipliers. In contrast, we propose an FL algorithm grounded in the proximal augmented Lagrangian (AL) method. This algorithm efficiently and robustly finds an $(\epsilon_1, \epsilon_2)$-KKT solution of (1) for its definition). At each iteration of this algorithm, a fixed penalty parameter is employed, and an approximate solution to a proximal AL subproblem associated with (1) is computed by an inexact alternating direction method of multipliers (ADMM) in a federated manner. We study the worst-case complexity of this algorithm under a *locally Lipschitz* assumption on $\nabla f_i$, $1 \le i \le n$, and $\nabla c_i$, $0 \le i \le n$. Our main contributions are highlighted below.

- We propose a proximal AL based FL algorithm (Algorithm 1) for seeking an approximate KKT solution of problem (1). The proposed algorithm naturally generalizes the current FL algorithms designed for unconstrained finite-sum optimization (see problem (6) below). Under

[1] Distributed optimization with global and local constraints has been studied before in the literature (e.g., see Nedic et al. [53], Zhu and Martínez [78]).

a *locally Lipschitz* condition and mild assumptions, we establish the worst-case complexity for finding an $(\epsilon_1, \epsilon_2)$-KKT solution of problem (1). To the best of our knowledge, the proposed algorithm is the first one for FL with global and local constraints, and its complexity results are entirely new in the literature.

- We conduct numerical experiments by comparing our proximal AL based FL algorithm with existing FL algorithms on several real-world constrained learning tasks including binary classification with specified recall and classification with nonconvex fairness constraints. Our numerical results validate that our FL algorithm can achieve solution quality comparable to the centralized algorithm.
- We propose an inexact ADMM based FL algorithm (Algorithm 2) for solving an unconstrained finite-sum optimization problem (see problem (5) below). Equipped with a newly introduced verifiable termination criterion, Algorithm 2 serves as a subproblem solver for Algorithm 1. We establish a global linear convergence rate for this algorithm under the assumptions of strongly convex local objectives and *locally* Lipschitz continuous gradients.

## 2. A proximal AL based FL algorithm for solving problem (1)

In this section we propose a proximal AL based FL algorithm for solving (1). This algorithm follows a similar framework to a proximal AL method. At each iteration, it applies the inexact ADMM (Algorithm 2) to find an approximate solution $w^{k+1}$ to the proximal AL subproblem associated with problem (1):

$$\min_w \left\{ \ell_k(w) := \sum_{i=1}^n f_i(w) + h(w) + \frac{1}{2\beta} \sum_{i=0}^n [\|[\mu_i^k + \beta c_i(w)]_+\|^2 - \|\mu_i^k\|^2] + \frac{1}{2\beta}\|w - w^k\|^2 \right\}, \quad (2)$$

where $[\cdot]_+$ denotes the nonnegative part of a vector. The multiplier estimates are updated according to the classical scheme: $\mu_i^{k+1} = [\mu_i^k + \beta c_i(w^{k+1})]_+$ for each $0 \le i \le n$.

```
Algorithm 1: A proximal AL algorithm for FL with constraints
Data: tolerances $\epsilon_1, \epsilon_2 \in (0,1)$, $w^0 \in \mathrm{dom}(h)$, $\mu_i^0 \ge 0$ for $0 \le i \le n$, $\bar{s} > 0$, and $\beta > 0$;
Result: A primal-dual solution pair $(w^{k+1}, \mu^{k+1})$;
for $k = 0, 1, 2, \ldots$ do
    Set $\tau_k = \bar{s}/(k+1)^2$;
    Call the inexact ADMM with $(\tau, \tilde{w}^0) = (\tau_k, w^k)$ to find an approximate solution $w^{k+1}$ to
      (3) in a federated manner such that $\mathrm{dist}_\infty(0, \partial\ell_k(w^{k+1})) \le \tau_k$.
    Server update: The central server updates $\mu_0^{k+1} = [\mu_0^k + \beta c_0(w^{k+1})]_+$;
    Communication (broadcast): Each local client $i$ receives $w^{k+1}$ from the server.
    Client update (local): Each local client $i$ updates $\mu_i^{k+1} = [\mu_i^k + \beta c_i(w^{k+1})]_+$.
    Termination: Output $(w^{k+1}, \mu^{k+1})$ and terminate the algorithm if
        $\|w^{k+1} - w^k\|_\infty + \beta\tau_k \le \beta\epsilon_1, \quad \|\mu^{k+1} - \mu^k\|_\infty \le \beta\epsilon_2.$
end
```

$$P_{i,k}(w) := f_i(w) + \frac{1}{2\beta}\|[\mu_i^k + \beta c_i(w)]_+\|^2 + \frac{1}{2(n+1)\beta}\|w - w^k\|^2.$$

Notice that the subproblem (2) can be rewritten as

$$\min_w \left\{ \ell_k(w) = \sum_{i=0}^{n} P_{i,k}(w) + h(w) \right\}, \tag{3}$$

where $P_{i,k}$, $0 \le i \le n$, are defined as

$$P_{i,k}(w) := f_i(w) + \frac{1}{2\beta}[\|[\mu_i^k + \beta c_i(w)]_+\|^2 - \|\mu_i^k\|^2] + \frac{1}{2(n+1)\beta}\|w - w^k\|^2,\ \forall i \ge 0. \tag{4}$$

When Algorithm 2 is applied to solve problem (3), the local merit function $P_{i,k}$, constructed from the local objective $f_i$ and local constraint $c_i$, is handled by the respective local client $i$, while the merit function $P_{0,k}$ is handled by the central server. Hence, Algorithm 2 is well-suited for the FL framework that the local objective $f_i$ and local constraint $c_i$ are handled by the local client $i$, and the central server performs aggregation and handles the global constraint $c_0$.

We now make the following assumptions: (a) The proximal operator for $h$ and the projection onto $\mathbb{R}_+^m$ can be exactly evaluated; (b) The functions $f_i$, $1 \le i \le n$, and mappings $c_i$, $0 \le i \le n$, are continuously differentiable, and $\nabla f_i$, $1 \le i \le n$, and $\nabla c_i$, $0 \le i \le n$, are locally Lipschitz continuous on $\mathbb{R}^d$; (c) The strong duality holds for problems (1) and its dual problem

$$\sup_{\mu \ge 0} \inf_w \{f(w) + h(w) + \langle \mu, c(w) \rangle\}.$$

The following theorem states the worst-case complexity results of Algorithm 1, whose proof is relegated to Appendix D.4.

**Theorem 1** *The number of outer iteration of Algorithm 1 is at most $\mathcal{O}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$, and the total number of inner iterations of Algorithm 1 is at most $\widetilde{\mathcal{O}}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$. In each iteration, Algorithm 1 requires one communication round.*

## 3. An inexact ADMM for FL

In this section we propose an inexact ADMM based FL algorithm for solving a class of finite-sum optimization problems. This algorithm is used as a subproblem solver for the proximal AL based FL algorithm proposed in Algorithm 1. In particular, we consider the following regularized unconstrained finite-sum optimization problem:

$$\min_w \left\{ F_h(w) := \sum_{i=0}^{n} F_i(w) + h(w) \right\}. \tag{5}$$

where $F_i : \mathbb{R}^d \to \mathbb{R}$, $0 \le i \le n$, are continuously differentiable and convex functions. We now make the following additional assumptions on problem (5) throughout this section: (a) The functions $F_i$, $0 \le i \le n$, are continuously differentiable, and moreover, $\nabla F_i$, $0 \le i \le n$, are locally Lipschitz continuous on $\mathbb{R}^d$; (b) The functions $F_i$, $0 \le i \le n$, are strongly convex on $\mathbb{R}^d$, that is, there exists some $\sigma > 0$ such that $\langle \nabla F_i(u) - \nabla F_i(v), u - v \rangle \ge \sigma \|u - v\|^2\ \forall u, v \in \mathbb{R}^d$, $0 \le i \le n$.

The iteration complexity of Algorithm 2 is established in the following theorem, whose proof is relegated to Appendix C.3.

**Theorem 2** *Algorithm 2 terminates in at most $\mathcal{O}(|\log \tau|)$ iterations. Also, Algorithm 2 requires one communication round at each iteration.*

**Algorithm 2:** An inexact ADMM for finite-sum optimization

```
Data: tolerance $\tau \in (0,1]$, $q \in (0,1)$, $\tilde{w}^0 \in \mathrm{dom}(h)$, and $\rho_i > 0$ for $1 \le i \le n$;
Result: A solution $w^{t+1}$;
Set $w^0 = \tilde{w}^0$, and $(u_i^0, \lambda_i^0, \tilde{u}_i^0) = (\tilde{w}^0, -\nabla F_i(\tilde{w}^0), \tilde{w}^0 - \nabla F_i(\tilde{w}^0)/\rho_i)$ for $1 \le i \le n$.
for $t = 0, 1, 2, \ldots$ do
    Server update: The central server finds an approximate solution $w^{t+1}$ to
        $$\min_w \left\{ \varphi_{0,t}(w) = F_0(w) + h(w) + \sum_{i=1}^n \left[ \frac{\rho_i}{2} \|\tilde{u}_i^t - w\|^2 \right] \right\}.$$
    Communication (broadcast): Each local client $i$ receives $w^{t+1}$ from the server.
    Client update (local): Each local client $i$ finds an approximate solution $u_i^{t+1}$ to
        $$\min_{u_i} \left\{ \varphi_{i,t}(u_i) = F_i(u_i) + \langle \lambda_i^t, u_i - w^{t+1} \rangle + \frac{\rho_i}{2} \|u_i - w^{t+1}\|^2 \right\},$$
    and then updates $\lambda_i^{t+1} = \lambda_i^t + \rho_i(u_i^{t+1} - w^{t+1})$, $\tilde{u}_i^{t+1} = u_i^{t+1} + \lambda_i^{t+1}/\rho_i$, and
    $\tilde{\varepsilon}_{i,t+1} = \|\nabla \varphi_{i,t}(w^{t+1}) - \rho_i(w^{t+1} - u_i^t)\|_\infty$.
    Communication: Each local client $i$ sends $(\tilde{u}_i^{t+1}, \tilde{\varepsilon}_{i,t+1})$ back to the central server.
    Termination: Output $w^{t+1}$ and terminate this algorithm if $\varepsilon_{t+1} + \sum_{i=1}^n \tilde{\varepsilon}_{i,t+1} \le \tau$.
end
```

## 4. Numerical experiments

**Linear equality constrained quadratic programming** A description of the experiment setup is deferred to Appendix F.1. The computational results of Algorithm 1 and the centralized proximal AL method (abbreviated as cProx-AL) for solving the randomly generated instances are presented in Table 1. One observes that: 1) both Algorithm 1 and the centralized proximal AL method are capable of finding a solution of similar quality in terms of objective value and constraint violation; 2) Algorithm 1 is as efficient as centralized proximal AL methods as measured by the number of outer iterations; 3) Algorithm 1 can scale-up to a varied number of clients and constraints. To account for all factors, the proposed methods can approximate high-quality solutions as the centralized method.

**Neyman-Pearson classification** A description of the experiment setup is relegated to Appendix F.2. One can see from Table 2 that: 1) Algorithm 1 yields solutions with comparable quality to cProx-AL, both concerning loss for class 0 and class 1, similar to what we observed in the previous experiment. 2) when solving the Neyman-Pearson classification, we can effectively regulate the loss value for the minority class (class 1) to remain below a predefined threshold across all local clients, demonstrating the practical utility of applications such as detecting rare diseases where the performance of minority class is critical. To provide further insights into the optimization progress, we plot the feasibility violation (i.e., loss for class 0) of Algorithm 1 over its outer iterations in

Table 1: Numerical results for linear equality constrained quadratic programming

| | | | objective value | | feasibility violation ($\times 10^{-4}$) | | #outer iterations | |
|---|---|---|---|---|---|---|---|---|
| $d$ | $n$ | $\tilde{m}$ | Algorithm 1 | cProx-AL | Algorithm 1 | cProx-AL | Algorithm 1 | cProx-AL |
| 100 | 1 | 1 | -6.1982 | -6.1982 | 1.1986 | 3.0593 | 5.6 | 5.0 |
| 100 | 5 | 1 | -5.4337 | -5.4348 | 5.5226 | 7.6943 | 6.0 | 6.8 |
| 100 | 10 | 1 | -0.8998 | -0.9021 | 7.7720 | 6.9185 | 8.5 | 13.0 |
| 500 | 1 | 5 | -33.0971 | -33.0971 | 7.5226 | 0.9430 | 5.9 | 5.0 |
| 500 | 5 | 5 | -30.8218 | -30.8220 | 4.4055 | 4.5350 | 4.0 | 4.0 |
| 500 | 10 | 5 | -24.7565 | -24.7583 | 4.4589 | 6.8591 | 5.0 | 5.1 |

Table 2: Numerical results for Neyman-Pearson classification.

| | | | | loss for class 0 | | | loss for class 1 ($\leq 0.2$) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dataset | #class 1/ class 0 | m | $n$ | Algorithm 1 | cProx-AL | Unconstr | ours | | cProx-AL | | Unconstr | |
| | | | | | | | mean | max | mean | max | mean | max |
| breast-cancer-wisc | 240/455 | 20 | 1 | 0.4340 | 0.4343 | 0.1388 | 0.2088 | 0.2088 | 0.2088 | 0.2088 | 0.3266 | 0.3266 |
| | | | 5 | 0.6840 | 0.6846 | 0.1393 | 0.1826 | 0.2030 | 0.1825 | 0.2030 | 0.3280 | 0.4052 |
| | | | 10 | 0.6481 | 0.6484 | 0.1409 | 0.1811 | 0.2067 | 0.1811 | 0.2067 | 0.3276 | 0.5337 |
| | | | 20 | 0.7630 | 0.7633 | 0.1436 | 0.1647 | 0.2057 | 0.1647 | 0.2057 | 0.3256 | 0.5404 |
| adult | 7840/24715 | 21 | 1 | 0.8677 | 0.8643 | 0.2135 | 0.1991 | 0.1991 | 0.1991 | 0.1991 | 0.7886 | 0.7886 |
| | | | 5 | 0.8475 | 0.8544 | 0.2136 | 0.1977 | 0.2019 | 0.1975 | 0.2018 | 0.7887 | 0.8178 |
| | | | 10 | 0.8656 | 0.8668 | 0.2136 | 0.1914 | 0.2003 | 0.1905 | 0.1991 | 0.7885 | 0.8325 |
| | | | 20 | 0.8688 | 0.8768 | 0.2137 | 0.1878 | 0.2028 | 0.1876 | 0.2025 | 0.7882 | 0.8418 |
| monks-1 | 275/275 | 21 | 1 | 1.7794 | 1.7821 | 0.5332 | 0.1966 | 0.1966 | 0.1966 | 0.1966 | 0.6508 | 0.6508 |
| | | | 5 | 1.8229 | 1.8278 | 0.5313 | 0.1909 | 0.2013 | 0.1908 | 0.2009 | 0.6516 | 0.7207 |
| | | | 10 | 1.8457 | 1.8484 | 0.5297 | 0.1834 | 0.2059 | 0.1836 | 0.2064 | 0.6480 | 0.8964 |
| | | | 20 | 2.0595 | 2.0743 | 0.5364 | 0.1560 | 0.2004 | 0.1551 | 0.2005 | 0.6461 | 0.9358 |

Figure 1. We can see that our proposed methods can converge to a feasible solution within a few outer iterations, effectively enforcing consistency across all the clients.

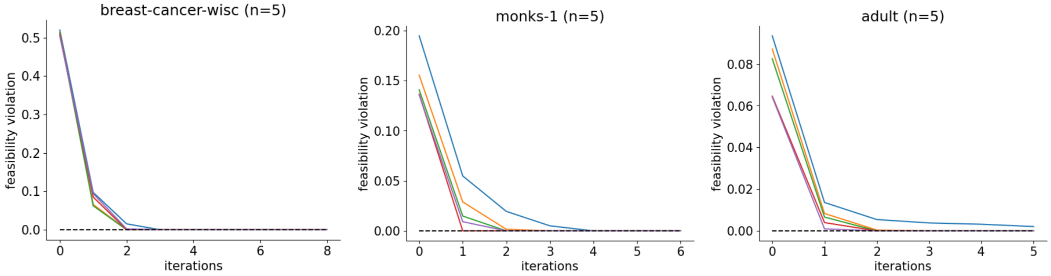

Figure 1: Progression of feasibility violation over iterations on brease-cancer-wisc and monks-1 datasets. Each client's feasibility violation progression is represented by a distinct color.

## References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[3] Naman Agarwal, Nicolas Boumal, Brian Bullins, and Coralia Cartis. Adaptive regularization with cubics on manifolds. *Mathematical Programming*, 188:85–134, 2021.

[4] Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. A primal-dual method for conic constrained distributed optimization problems. *Advances in Neural Information Processing Systems*, 29, 2016.

[5] Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. A distributed ADMM-like method for resource sharing over time-varying networks. *SIAM Journal on Optimization*, 29(4):3036–3068, 2019.

[6] Necdet Serhat Aybat and Garud Iyengar. An augmented Lagrangian method for conic convex programming. *arXiv preprint arXiv:1302.6322*, 2013.

[7] Ernesto G Birgin and José Mario Martínez. Complexity and performance of an augmented Lagrangian algorithm. *Optimization Methods and Software*, 35(5):885–920, 2020.

[8] Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

[9] Richard H Byrd, Robert B Schnabel, and Gerald A Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24(5):1152–1170, 1987.

[10] Coralia Cartis, Nicholas IM Gould, and Ph L Toint. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA Journal of Numerical Analysis*, 32(4):1662–1695, 2012.

[11] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM Journal on Optimization*, 23(3): 1553–1574, 2013.

[12] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods. *SIAM Journal on Numerical Analysis*, 53(2):836–851, 2015.

[13] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 319–328, 2019.

[14] Peter W Christensen and Anders Klarbring. *An introduction to structural optimization*, volume 153. Springer Science & Business Media, 2008.

[15] Lingyang Chu, Lanjun Wang, Yanjie Dong, Jian Pei, Zirui Zhou, and Yong Zhang. Fedfair: Training fair models in cross-silo federated learning. *arXiv preprint arXiv:2109.05662*, 2021.

[16] Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pages 300–332. PMLR, 2019.

[17] Frank E Curtis and Michael L Overton. A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. *SIAM Journal on Optimization*, 22(2):474–500, 2012.

[18] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

[19] Borja Rodríguez Gálvez, Filip Granqvist, Rogier van Dalen, and Matt Seigel. Enforcing fairness in private federated learning via the modified method of differential multipliers. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[20] Yonghai Gong, Yichuan Li, and Nikolaos M Freris. FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2575–2587. IEEE, 2022.

[21] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[22] Geovani Nunes Grapiglia and Ya-xiang Yuan. On the complexity of an augmented Lagrangian method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 41(2):1546–1568, 2021.

[23] Gabriel Haeser, Hongcheng Liu, and Yinyu Ye. Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary. *Mathematical Programming*, 178:263–299, 2019.

[24] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer, New York, 2nd edition, 2009.

[25] Chuan He and Zhaosong Lu. A Newton-CG based barrier method for finding a second-order stationary point of nonconvex conic optimization with complexity guarantees. *SIAM Journal on Optimization*, 33(2):1191–1222, 2023.

[26] Chuan He, Heng Huang, and Zhaosong Lu. A Newton-CG based barrier-augmented Lagrangian method for general nonconvex conic optimization. *arXiv preprint arXiv:2301.04204*, 2023.

[27] Chuan He, Zhaosong Lu, and Ting Kei Pong. A Newton-CG based augmented Lagrangian method for finding a second-order stationary point of nonconvex equality constrained optimization with complexity guarantees. *SIAM Journal on Optimization*, 33(3):1734–1766, 2023.

[28] Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pages 1529–1538. PMLR, 2017.

[29] Kevin Huang, Nuozhou Wang, and Shuzhong Zhang. An accelerated variance reduced extra-point approach to finite-sum VI and optimization. *arXiv preprint arXiv:2211.03269*, 2022.

[30] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[31] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[32] Weiwei Kong, Jefferson G Melo, and Renato DC Monteiro. Iteration complexity of an inner accelerated inexact proximal augmented Lagrangian method based on the classical Lagrangian function. *SIAM Journal on Optimization*, 33(1):181–210, 2023.

[33] Guanghui Lan and Renato DC Monteiro. Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Mathematical Programming*, 155(1-2):511–547, 2016.

[34] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[35] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[36] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[37] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[38] Zichong Li, Pin-Yu Chen, Sijia Liu, Songtao Lu, and Yangyang Xu. Rate-improved inexact augmented Lagrangian method for constrained nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2170–2178. PMLR, 2021.

[39] Buyun Liang, Tim Mitchell, and Ju Sun. NCVX: A user-friendly and scalable package for nonconvex optimization in machine learning. *arXiv preprint arXiv:2111.13984*, 2021.

[40] Peng Lin, Wei Ren, and Yongduan Song. Distributed multi-agent optimization subject to nonidentical constraints and communication delays. *Automatica*, 65:120–131, 2016.

[41] Yi Liu, JQ James, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, 7(8):7751–7763, 2020.

[42] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pages 240–254. Springer, 2020.

[43] Songtao Lu. A single-loop gradient descent and perturbed ascent algorithm for nonconvex functional constrained optimization. In *International Conference on Machine Learning*, pages 14315–14357. PMLR, 2022.

[44] Songtao Lu, Meisam Razaviyayn, Bo Yang, Kejun Huang, and Mingyi Hong. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. *Advances in Neural Information Processing Systems*, 33:2811–2822, 2020.

[45] Zhaosong Lu and Sanyou Mei. Accelerated first-order methods for convex optimization with locally Lipschitz continuous gradient. *SIAM Journal on Optimization*, 33(3):2275–2310, 2023.

[46] Zhaosong Lu and Zirui Zhou. Iteration-complexity of first-order augmented Lagrangian methods for convex conic programming. *SIAM Journal on Optimization*, 33(2):1159–1190, 2023.

[47] Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*, 2020.

[48] 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.

[49] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6): 1–35, 2021.

[50] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019.

[51] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[52] Ion Necoara, Andrei Patrascu, and Francois Glineur. Complexity of first-order inexact Lagrangian and penalty methods for conic convex programming. *Optimization Methods and Software*, 34(2):305–335, 2019.

[53] Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[54] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[55] Maher Nouiehed, Jason D Lee, and Meisam Razaviyayn. Convergence to second-order stationarity for constrained non-convex optimization. *arXiv preprint arXiv:1810.02024*, 2018.

[56] Michael O'Neill and Stephen J Wright. A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees. *IMA Journal of Numerical Analysis*, 41(1): 84–121, 2021.

[57] Andrei Patrascu, Ion Necoara, and Quoc Tran-Dinh. Adaptive inexact fast augmented Lagrangian methods for constrained convex optimization. *Optimization Letters*, 11:609–626, 2017.

[58] Le Peng, Gaoxiang Luo, Andrew Walker, Zachary Zaiman, Emma K Jones, Hemant Gupta, Kristopher Kersten, John L Burns, Christopher A Harle, Tanja Magoc, et al. Evaluation of federated learning variations for COVID-19 diagnosis using chest radiographs from 42 US and European hospitals. *Journal of the American Medical Informatics Association*, 30(1): 54–63, 2023.

[59] Le Peng, Sicheng Zhou, Jiandong Chen, Rui Zhang, Ziyue Xu, and Ju Sun. A systematic evaluation of federated learning on biomedical natural language processing. In *International Workshop on Federated Learning for Distributed Data Mining*, 2023. URL https://openreview.net/forum?id=pLEQFXACNA.

[60] MJD Powell and ya-xiang Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 49(1):189–211, 1991.

[61] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[62] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ Digital Medicine*, 3(1):119, 2020.

[63] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3): e0118432, 2015.

[64] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-IID data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019.

[65] Zebang Shen, Juan Cervino, Hamed Hassani, and Alejandro Ribeiro. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2021.

[66] Xin Tong, Yang Feng, and Anqi Zhao. A survey on Neyman-Pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(2): 64–81, 2016.

[67] Yash Travadi, Le Peng, Xuan Bi, Ju Sun, and Mochen Yang. Welfare and fairness dynamics in federated learning: A client selection perspective. *Statistics and Its Interface*, 2023.

[68] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106: 25–57, 2006.

[69] Han Wang, Siddartha Marella, and James Anderson. Fedadmm: A federated primal-dual algorithm allowing partial participation. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 287–294. IEEE, 2022.

[70] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33:7611–7623, 2020.

[71] Peng Wang, Peng Lin, Wei Ren, and Yongduan Song. Distributed subgradient-based multiagent optimization with more general step sizes. *IEEE Transactions on Automatic Control*, 63 (7):2295–2302, 2017.

[72] Yue Xie and Stephen J Wright. Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints. *Journal of Scientific Computing*, 86:1–30, 2021.

[73] Yangyang Xu. Iteration complexity of inexact augmented Lagrangian methods for constrained convex programming. *Mathematical Programming*, 185:199–244, 2021.

[74] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

[75] Deming Yuan, Shengyuan Xu, and Huanyu Zhao. Distributed primal-dual subgradient method for multiagent optimization via consensus algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(6):1715–1724, 2011.

[76] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. FedPD: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021.

[77] Shenglong Zhou and Geoffrey Ye Li. Federated learning via inexact ADMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[78] Minghui Zhu and Sonia Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2011.

[79] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

## Appendix A. Related works

**FL algorithms for unconstrained optimization:** Federated learning has emerged as a cornerstone technique for privacy-preserved distributed learning since Google proposed the seminal work [48]. Unlike traditional centralized learning methods, FL enables the training of models with distributed edge clients, ranging from small mobile devices like phones [50] to large data providers such as hospitals and banks [42]. This inherent property of privacy preservation aligns seamlessly with the principles upheld by various critical domains, including healthcare [58, 59, 62], finance [42], IoT [50], and transportation [41], where safeguarding data privacy is essential.

FedAvg, introduced by [48], is the first and also the most widely applied FL algorithm. It was proposed for solving the unconstrained finite-sum optimization problem:

$$\min_{w} f(w) = \sum_{i=1}^{n} f_i(w). \tag{6}$$

Since then, many variants have been proposed to tackle various practical issues, such as data heterogeneity [30, 37, 76], system heterogeneity [20, 35, 70], fairness [36], efficiency [31, 64], and incentives [67]. For example, [35] proposed FedProx by adding a proximal term in the local objective to handle clients with different computation capabilities. [30] proposed Scallfold to address the issue of date heterogeneity where local data is non-independent and identically distributed (non-iid). Additionally, ADMM based FL algorithms have been proposed in [1, 20, 69, 76, 77], and these methods have been shown to be inherently resilient to heterogeneity. [61] extended FedAvg by introducing adaptive optimizers for server aggregation, significantly reducing communication costs and improving FL scalability. [36] proposed Ditto, a personalized FL framework that demonstrates improved client fairness and robustness. More variants of FL algorithms and their applications can be found in the survey [34]. Despite the numerous FL algorithms proposed previously, they primarily focus on unconstrained FL problems, leaving a gap between constrained optimization and FL.

**Centralized algorithms for constrained optimization:** Recent years have witnessed fruitful algorithmic developments for constrained optimization in the centralized setting. In particular, there has been a rich literature on inexact AL methods for solving convex constrained optimization problems (e.g., see Aybat and Iyengar [6], Lan and Monteiro [33], Lu and Mei [45], Lu and Zhou [46], Necoara et al. [52], Patrascu et al. [57], Xu [73]). In addition, AL methods and variants have also been extended to solve nonconvex constrained optimization problems (e.g., see Birgin and Martínez [7], Grapiglia and Yuan [22], He et al. [26, 27], Hong et al. [28], Kong et al. [32], Li et al. [38], Lu [43]). Moreover, sequential quadratic programming methods [8, 17], trust-region methods [9, 60], interior point method [68], and extra-point method [29] have also been proposed for solving constrained optimization problems. Furthermore, there have been many recent works on algorithms for finding second-order stationary points of nonconvex constrained optimization problems (e.g., see [3, 10–12, 23, 25–27, 44, 51, 55, 56, 72]).

**Distributed algorithms for constrained optimization:** In another line of research, many algorithms have been developed for distributed optimization with global and local constraints. An early work [53] introduced a distributed projected subgradient algorithm for distributed optimization with local constraints. This work has been extended to handle scenarios involving time-varying directed graphs in [40, 71]. Yet, these methods require each node to compute a projection on the local constraint set, which is applicable only to relatively simple constraints. To address more complicated constraints, distributed primal-dual algorithms were developed in [4, 5] for distributed convex optimization with conic constraints. In addition, primal-dual projected subgradient algorithms [75, 78] have been developed for distributed optimization with global and local constraints. For an overview of algorithmic developments in distributed optimization with constraints, we refer to [74]. We emphasize that the existing algorithms for constrained distributed optimization do not follow the common FL framework where clients perform multiple local updates before aggregating the global model. The algorithm development in this paper follows a distinct trajectory compared to them.

**FL algorithms for constrained learning problems:** Some studies have combined constrained optimization techniques with FL algorithms to tackle complex learning tasks, such as addressing label imbalances and promoting model fairness. For example, [65] proposes an FL algorithm aimed at handling class imbalances, using primal-dual updates through the Lagrangian function. Similarly, [15, 18] propose FL algorithms designed for fairness-constrained learning, also incorporating primal-dual updates using the Lagrangian function. In addition, [19] proposes an FL algorithm for fairness-constrained learning, implementing primal-dual updates with the augmented Lagrangian function. Nonetheless, these studies are tailored to specific applications and do not establish convergence guarantees regarding constraint feasibility, stationarity, or consensus.

## Appendix B. Constrained optimization problems in modern machine learning

In recent years, there has been a growing prominence in solving constrained optimization problems arising in machine learning. Especially, when moving to trustworthy AI and efficient AI, we can see an overwhelming number of problems where explicit constraints have to be enforced and computed during the learning process. For example, robustness evaluation [21], learning with fairness [2, 13, 49], learning with label imbalance [63], neural architecture search [79], topology optimization [14], knowledge-aware machine learning [47] can be formulated as optimization problems with hard constraints. We next present Neyman-Pearson classification and learning with fairness:

**Neyman-Pearson classification:** Consider a binary classification problem, where one is more concerned with the risk of misclassifying one specific class than the other one, as often occurs in medical diagnosis. To address this problem, Neyman-Pearson classification model is proposed as follows (e.g., see [66]):

$$\min_{w} \frac{1}{n_0}\sum_{i=1}^{n_0}\varphi(w,z_{i,0}) \quad \text{s.t.} \quad \frac{1}{n_1}\sum_{i=1}^{n_1}\varphi(w,z_{i,1}) \le r, \tag{7}$$

where $w$ is the weight parameter, $\varphi$ is a loss function, $\{z_{i,0}\}_{i=1}^{n_0}$ and $\{z_{i,1}\}_{i=1}^{n_1}$ are the training data from two separate classes 0 and 1, respectively, and $r > 0$ controls the training error for class 1. The Neyman-Pearson classification model (7) is introduced as a statistical learning model for handling asymmetric training error priorities.

**Learning with fairness:** Incorporating fairness constraints into the training of machine learning models is widely recognized as an important approach to ensure the models' trustworthiness [2, 13, 49]. Training a model with fairness constraints is usually formulated as follows:

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n}\varphi(w,z_i) \quad \text{s.t.} \quad \min_{1\le i\le k} p_j(w,\{z_i\}_{i=1}^n) \ge \rho \max_{1\le j\le k} p_j(w,\{z_i\}_{i=1}^n), \tag{8}$$

where $w$ is the weight parameter, $\varphi$ is a loss function, $p_j$, $1 \le j \le k$, are performance metrics, $\rho \in [0,1]$ is the targeted fairness level, and $\{z_i\}$ is the training data set.

## Appendix C. Proof of Theorem 2

We first derive the following equivalent consensus reformulation of problem (5):

$$\min_{w,u_i} \left\{ \sum_{i=1}^{n} F_i(u_i) + F_0(w) + h(w) \right\} \quad \mathrm{s.\,t.} \quad u_i = w, \quad 1 \le i \le n, \tag{9}$$

Throughout this section, we let $(\tilde{w}^*, u^*)$ be the optimal solution of (9), and $\lambda^*$ be the associated Lagrangian multiplier. Recall from the definition of $\tilde{u}_i^t$ in Algorithm 2 that

$$\tilde{u}_i^t = u_i^t + \lambda_i^t/\rho_i, \quad \forall 1 \le i \le n, t \ge 0. \tag{10}$$

### C.1. Output of Algorithm 2

**Theorem 3** *If Algorithm 2 terminates at iteration t, its output $w^{t+1}$ satisfies*

$$\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) \le \tau.$$

**Proof** By the definition of $F_h$ in (5), one has that

$$\partial F_h(w^{t+1}) = \sum_{i=0}^{n} \nabla F_i(w^{t+1}) + \partial h(w^{t+1}). \tag{11}$$

In addition, notice from (2), (2), and (10) that

$$\begin{aligned}
\partial \varphi_{0,t}(w^{t+1}) &= \nabla F_0(w^{t+1}) + \sum_{i=1}^{n} \rho_i (w^{t+1} - \tilde{u}_i^t) + \partial h(w^{t+1}) \\
&= \nabla F_0(w^{t+1}) + \sum_{i=1}^{n} [\rho_i (w^{t+1} - u_i^t) - \lambda_i^t] + \partial h(w^{t+1}), \\
\nabla \varphi_{i,t}(w^{t+1}) &= \nabla F_i(w^{t+1}) + \lambda_i^t, \quad \forall 1 \le i \le n.
\end{aligned}$$

Combining these with (11), we obtain that

$$\partial F_h(w^{t+1}) = \partial \varphi_{0,t}(w^{t+1}) + \sum_{i=1}^{n} [\nabla \varphi_{i,t}(w^{t+1}) - \rho_i (w^{t+1} - u_i^t)],$$

which together with $\mathrm{dist}_\infty(0, \partial \varphi_{0,t}(w^{t+1})) \le \varepsilon_{t+1}$ (see Algorithm 2) implies that

$$\begin{aligned}
\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) &\le \mathrm{dist}_\infty(0, \partial \varphi_{0,t}(w^{t+1})) + \sum_{i=1}^{n} \|\nabla \varphi_{i,t}(w^{t+1}) - \rho_i (w^{t+1} - u_i^t)\|_\infty \\
&\le \varepsilon_{t+1} + \sum_{i=1}^{n} \tilde{\varepsilon}_{i,t+1}.
\end{aligned}$$

Using this and the termination criterion, we obtain that $\mathrm{dist}_\infty(0, \partial F_h(w^{t+1})) \le \tau$ holds as desired.
■

### C.2. Bounded iterates of Algorithm 2

**Lemma 4** *Let $\{u_i^{t+1}\}_{1\le i\le n, t\in\mathbb{T}}$ and $\{w^{t+1}\}_{t\in\mathbb{T}}$ be all the iterates generated by Algorithm 2, where $\mathbb{T}$ is a subset of consecutive nonnegative integers starting from $0$. Then we have $w^{t+1}\in\mathcal{Q}$ and $u_i^{t+1}\in\mathcal{Q}$ for all $1\le i\le n$ and $t\in\mathbb{T}$, where*

$$\mathcal{Q}=\left\{v:\|v-\tilde{w}^*\|^2\le\frac{n+1}{\sigma^2(1-q^2)}+\frac{1}{\sigma}\sum_{i=1}^n\left(\rho_i\|\tilde{w}^*-\tilde{w}^0\|^2+\frac{1}{\rho_i}\|\nabla F_i(\tilde{w}^*)-\nabla F_i(\tilde{w}^0)\|^2\right)\right\}. \tag{12}$$

**Proof** [Proof of Lemma 4]

Recall from that $F_i$, $0\le i\le n$, are strongly convex with modulus $\sigma>0$. In addition, by (10) and the fact that $\mathrm{dist}_\infty(0,\partial\varphi_{0,t}(w^{t+1}))\le\varepsilon_{t+1}$, one can obtain that there exist some $h^{t+1}\in\partial h(w^{t+1})$ and $\|e_0^{t+1}\|_\infty\le\varepsilon_{t+1}$ such that

$$\begin{aligned}e_0^{t+1}&=\nabla F_0(w^{t+1})+h^{t+1}+\sum_{i=1}^n\rho_i(w^{t+1}-\tilde{u}_i^t)\overset{(10)}{=}\nabla F_0(w^{t+1})+h^{t+1}+\sum_{i=1}^n[\rho_i(w^{t+1}-u_i^t)-\lambda_i^t]\\&=\nabla F_0(w^{t+1})+h^{t+1}+\sum_{i=1}^n[\rho_i(u_i^{t+1}-u_i^t)-\lambda_i^{t+1}].\end{aligned} \tag{13}$$

Using the fact that $\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty\le\varepsilon_{t+1}$, one can see that there exists $\|e_i^{t+1}\|_\infty\le\varepsilon_{t+1}$ such that

$$e_i^{t+1}=\nabla\varphi_{i,t}(u_i^{t+1})\overset{(2)}{=}\nabla F_i(u_i^{t+1})+\lambda_i^t+\rho_i(u_i^{t+1}-w^{t+1})=\nabla F_i(u_i^{t+1})+\lambda_i^{t+1},\quad\forall 1\le i\le n, \tag{14}$$

Recall that $\tilde{w}^*$ and $u^*$ are the optimal solution of (9), and $\lambda^*\in\mathbb{R}^m$ is the associated Lagrangian multiplier. Then there exists $h^*\in\partial h(\tilde{w}^*)$ such that

$$\nabla F_i(u_i^*)+\lambda_i^*=0,\quad\nabla F_0(\tilde{w}^*)+h^*-\sum_{i=1}^n\lambda_i^*=0,\quad u_i^*=\tilde{w}^*,\quad\forall 1\le i\le n. \tag{15}$$

In view of this, (14), and the strong convexity of $F_i$, one can deduce that

$$\begin{aligned}\sigma\|u_i^{t+1}-\tilde{w}^*\|^2&\le\langle u_i^{t+1}-\tilde{w}^*,\nabla F_i(u_i^{t+1})-\nabla F_i(\tilde{w}^*)\rangle=\langle u_i^{t+1}-\tilde{w}^*,-\lambda_i^{t+1}+\lambda_i^*+e_i^{t+1}\rangle\\&\le\langle u_i^{t+1}-\tilde{w}^*,-\lambda_i^{t+1}+\lambda_i^*\rangle+\frac{\sigma}{2}\|u_i^{t+1}-\tilde{w}^*\|^2+\frac{1}{2\sigma}\|e_i^{t+1}\|^2,\end{aligned}$$

where the equality is due to $\tilde{w}^*=u_i^*$, $\nabla F_i(u_i^*)=\lambda_i^*$, and (14), and the last inequality follows from $\langle a,b\rangle\le t/2\|a\|^2+1/(2t)\|b\|^2$ for all $a,b\in\mathbb{R}^d$ and $t>0$. By (13), (15), and the strong convexity of $F_0$, one has that

$$\begin{aligned}\sigma\|w^{t+1}-\tilde{w}^*\|^2&\le\langle w^{t+1}-\tilde{w}^*,\nabla F_0(w^{t+1})+h^{t+1}-\nabla F_0(\tilde{w}^*)-h^*\rangle\\&=\langle w^{t+1}-\tilde{w}^*,\sum_{i=1}^n[\lambda_i^{t+1}-\lambda_i^*-\rho_i(u_i^{t+1}-u_i^t)]+e_0^{t+1}\rangle,\\&\le\langle w^{t+1}-\tilde{w}^*,\sum_{i=1}^n[\lambda_i^{t+1}-\lambda_i^*-\rho_i(u_i^{t+1}-u_i^t)]\rangle+\frac{\sigma}{2}\|w^{t+1}-\tilde{w}^*\|^2+\frac{1}{2\sigma}\|e_0^{t+1}\|^2,\end{aligned}$$

where the first inequality is due to the strong convexity of $F_0$ and the convexity of $h$, the equality is due to (13) and the second relation in (15), and the last inequality follows from $\langle a, b\rangle \le t/2\|a\|^2 + 1/(2t)\|b\|^2$ for all $a, b \in \mathbb{R}^d$ and $t > 0$. Summing up these inequalities and rearranging the terms, we obtain that

$$\begin{aligned}
&\frac{\sigma}{2}(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^{n}\|u_i^{t+1} - \tilde{w}^*\|^2) \\
&\le \langle w^{t+1} - \tilde{w}^*, \sum_{i=1}^{n}[\lambda_i^{t+1} - \lambda_i^* - \rho_i(u_i^{t+1} - u_i^t)]\rangle + \frac{1}{2\sigma}\|e_0^{t+1}\|^2 + \sum_{i=1}^{n}(\langle u_i^{t+1} - \tilde{w}^*, -\lambda_i^{t+1} + \lambda_i^*\rangle + \frac{1}{2\sigma}\|e_i^{t+1}\|^2) \\
&\le \sum_{i=1}^{n}\langle w^{t+1} - u_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^{n}\rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \\
&= \sum_{i=1}^{n}\frac{1}{\rho_i}\langle\lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^{n}\rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \frac{n+1}{2\sigma}\varepsilon_{t+1}^2, \qquad (16)
\end{aligned}$$

where the second inequality is due to $\|e_i^{t+1}\| \le \varepsilon_{t+1}$ for all $0 \le i \le n$ and $t \ge 0$. Notice that the following well-known identities hold:

$$\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle = \frac{1}{2}(\|w^{t+1} - u_i^{t+1}\|^2 - \|w^{t+1} - u_i^t\|^2 + \|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2), \tag{17}$$

$$\langle\lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle = \frac{1}{2}(\|\lambda_i^* - \lambda_i^t\|^2 - \|\lambda_i^* - \lambda_i^{t+1}\|^2 - \|\lambda_i^t - \lambda_i^{t+1}\|^2). \tag{18}$$

These along with (16) imply that

$$\begin{aligned}
&\frac{\sigma}{2}(\|w^{t+1} - \tilde{w}^*\|^2 + \sum_{i=1}^{n}\|u_i^{t+1} - \tilde{w}^*\|^2) + \sum_{i=1}^{n}\frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2 - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \\
&\overset{(16)}{\le} \sum_{i=1}^{n}\frac{1}{\rho_i}\langle\lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^{n}\rho_i\langle w^{t+1} - \tilde{w}^*, u_i^t - u_i^{t+1}\rangle + \sum_{i=1}^{n}\frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2 \\
&\overset{(17)}{\le} \sum_{i=1}^{n}\frac{1}{\rho_i}\langle\lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^{n}\frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2 + \|w^{t+1} - u_i^{t+1}\|^2) \\
&= \sum_{i=1}^{n}\frac{1}{\rho_i}\langle\lambda_i^t - \lambda_i^{t+1}, \lambda_i^{t+1} - \lambda_i^*\rangle + \sum_{i=1}^{n}\frac{1}{2\rho_i}\|\lambda_i^{t+1} - \lambda_i^t\|^2 + \sum_{i=1}^{n}\frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2) \\
&\overset{(18)}{=} \sum_{i=1}^{n}\frac{1}{2\rho_i}(\|\lambda_i^* - \lambda_i^t\|^2 - \|\lambda_i^* - \lambda_i^{t+1}\|^2) + \sum_{i=1}^{n}\frac{\rho_i}{2}(\|\tilde{w}^* - u_i^t\|^2 - \|\tilde{w}^* - u_i^{t+1}\|^2) \\
&= \sum_{i=1}^{n}[(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2) - (\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2)]. \qquad (19)
\end{aligned}$$

Summing up this inequality over $t = 0, \ldots, \bar{t}$, we obtain that

$$\begin{aligned}&\sum_{t=0}^{\bar{t}}\left[\frac{\sigma}{2}\left(\|w^{t+1}-\tilde{w}^*\|^2+\sum_{i=1}^{n}\|u_i^{t+1}-\tilde{w}^*\|^2\right)+\sum_{i=1}^{n}\frac{\rho_i}{2}\|w^{t+1}-u_i^t\|^2-\frac{n+1}{2\sigma}\varepsilon_{t+1}^2\right]\\ &\le\sum_{i=1}^{n}\left[\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^0\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^0\|^2\right)-\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^{\bar{t}+1}\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^{\bar{t}+1}\|^2\right)\right].\end{aligned} \tag{20}$$

Recall from Algorithm 2 that $\varepsilon_{t+1} = q^t$, $u_i^0 = \tilde{w}^0$, and $\lambda_i^0 = -\nabla F_i(\tilde{w}^0)$. Also, notice from (15) that $\tilde{w}^* = u_i^*$ and $\lambda_i^* = -\nabla F_i(u_i^*)$. By these and (20), one can deduce that

$$\begin{aligned}&\frac{\sigma}{2}(\|w^{t+1}-\tilde{w}^*\|^2+\sum_{i=1}^{n}\|u_i^{t+1}-\tilde{w}^*\|^2)\le\frac{n+1}{2\sigma}\sum_{t=0}^{\infty}q^{2t}+\sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^0\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^0\|^2\right)\\ &\le\frac{n+1}{2\sigma(1-q^2)}+\sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^0\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^0\|^2\right)\\ &=\frac{n+1}{2\sigma(1-q^2)}+\sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-\tilde{w}^0\|^2+\frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*)-\nabla F_i(\tilde{w}^0)\|^2\right).\end{aligned}$$

In view of this and the definition of $\mathcal{Q}$ in (12), we can observe that $w^{t+1} \in \mathcal{Q}$ and $u_i^{t+1} \in \mathcal{Q}$ for all $t \in \mathbb{T}$ and $1 \le i \le n$. Hence, the conclusion of this lemma holds as desired. ■

### C.3. Proof of Theorem 2

**Lemma 5** *Assume that $r, c > 0$ and $q \in (0, 1)$. Let $\{a_t\}_{t\ge0}$ be a sequence satisfying*

$$(1+r)a_{t+1} \le a_t + cq^{2t}, \quad \forall t \ge 0. \tag{21}$$

*Then we have*

$$a_{t+1} \le \max\left\{q, \frac{1}{1+r}\right\}^{t+1}\left(a_0+\frac{c}{1-q}\right), \quad \forall t \ge 0. \tag{22}$$

**Proof** It follows from (21) that

$$\begin{aligned}a_{t+1}&\le\frac{1}{1+r}a_t+\frac{1}{1+r}cq^{2t}\le\frac{1}{(1+r)^2}a_{t-1}+\frac{cq^{2(t-1)}}{(1+r)^2}+\frac{cq^{2t}}{1+r}\\ &\le\cdots\le\frac{1}{(1+r)^{t+1}}a_0+\sum_{i=0}^{t}\frac{cq^{2i}}{(1+r)^{t+1-i}}=\frac{1}{(1+r)^{t+1}}a_0+c\sum_{i=0}^{t}\frac{q^i}{(1+r)^{t+1-i}}q^i\\ &\le\frac{1}{(1+r)^{t+1}}a_0+c\max\left\{q,\frac{1}{1+r}\right\}^{t+1}\sum_{i=0}^{t}q^i\\ &\le\frac{1}{(1+r)^{t+1}}a_0+\frac{c}{1-q}\max\left\{q,\frac{1}{1+r}\right\}^{t+1}\le\max\left\{q,\frac{1}{1+r}\right\}^{t+1}\left(a_0+\frac{c}{1-q}\right),\end{aligned}$$

where the fifth inequality is due to $q^i \le \max\{q, 1/(1+r)\}^i$ and $1/(1+r)^{t+1-i} \le \max\{q, 1/(1+r)\}^{t+1-i}$. Hence, the relation (22) holds as desired. ■

**Lemma 6** *Let $\mathcal{Q}$ be defined in (12). Then there exists some $L_{\nabla F} > 0$ such that*

$$\|\nabla F_i(u) - \nabla F_i(v)\| \le L_{\nabla F}\|u - v\|, \quad \forall u, v \in \mathcal{Q}, 0 \le i \le n. \tag{23}$$

**Proof** Notice from (12) that the set $\mathcal{Q}$ is convex and compact. By this and the local Lipschitz continuity of $\nabla F_i$ on $\mathbb{R}^d$, one can verify that there exists some constant $L_{\nabla F} > 0$ such that (23) holds (see also Lemma 1 in [45]). ■

**Lemma 7** *Let $\{w^{t+1}\}_{t\in\mathbb{T}}$ and $\{u_i^{t+1}\}_{1\le i\le n, t\in\mathbb{T}}$ be all the iterates generated by Algorithm 2, where $\mathbb{T}$ is defined in Lemma 4. Then we have*

$$S_t \le q_r^t \left[ S_0 + \frac{1}{1-q}\left( \frac{n+1}{2\sigma} + \sum_{i=1}^{n} \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2} \right) \right], \quad \forall t \ge 0, \tag{24}$$

*where $\sigma$ and $L_{\nabla F}$ are given in the assumptions in Section 3 and Lemma 6, respectively, $q$ and $\rho_i$, $1 \le i \le n$, are inputs of Algorithm 2, and*

$$q_r = \max\left\{ q, \frac{1}{1+r} \right\}, \quad r = \min_{1\le i\le n} \left\{ \frac{\sigma\rho_i}{\rho_i^2 + 2L_{\nabla F}^2} \right\}, \tag{25}$$

$$S_t = \sum_{i=1}^{n} \left( \frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2 \right), \quad \forall t \ge 0. \tag{26}$$

**Proof** Recall from (19) that

$$\begin{aligned}
&\sum_{i=1}^{n} \left( \frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2 \right) \\
&\ge \sum_{i=1}^{n} \left( \frac{\rho_i + \sigma}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2 + \frac{\rho_i}{2}\|w^{t+1} - u_i^t\|^2 \right) + \frac{\sigma}{2}\|w^{t+1} - \tilde{w}^*\|^2 - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2 \\
&\ge \sum_{i=1}^{n} \left( \frac{\rho_i + \sigma}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2 \right) - \frac{n+1}{2\sigma}\varepsilon_{t+1}^2.
\end{aligned} \tag{27}$$

Also, notice from (14), (15), and (23) that

$$\|\lambda_i^* - \lambda_i^{t+1}\|^2 \overset{(14)(15)}{\le} (\|\nabla F_i(\tilde{w}^*) - \nabla F_i(u_i^{t+1})\| + \|e_i^{t+1}\|)^2 \overset{(23)}{\le} 2L_{\nabla F}^2\|\tilde{w}^* - u_i^{t+1}\|^2 + 2\varepsilon_{t+1}^2,$$

which implies that

$$\|\tilde{w}^* - u_i^{t+1}\|^2 \ge \frac{2\rho_i}{\rho_i^2 + 2L_{\nabla F}^2}\left( \frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2 \right) - \frac{2\varepsilon_{t+1}^2}{\rho_i^2 + 2L_{\nabla F}^2}. \tag{28}$$

By this, the definition of $S_t$ in (26), and (27), one has that

$$\begin{aligned}
&S_t + \left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right) q^{2t} \\
&= \sum_{i=1}^n \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2\right) + \left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\varepsilon_{t+1}^2 \\
&\overset{(27)}{\geq} \sum_{i=1}^n \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2\right) + \frac{\sigma}{2}\sum_{i=1}^n \|\tilde{w}^* - u_i^{t+1}\|^2 + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\varepsilon_{t+1}^2 \\
&\overset{(28)}{\geq} \sum_{i=1}^n \left(1 + \frac{\sigma\rho_i}{\rho_i^2 + 2L_{\nabla F}^2}\right)\left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^{t+1}\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^{t+1}\|^2\right) \\
&\geq (1+r)S_{t+1}.
\end{aligned}$$

When $t = 0$, (24) holds clearly. When $t \geq 1$, by the above inequality, (25), and Lemma 5 with $(a_t, c) = (S_t, \frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2})$, we obtain that

$$\begin{aligned}
S_t &\leq \max\left\{q, \frac{1}{1+r}\right\}^t \left[S_0 + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right] \\
&= q_r^t \left[S_0 + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right].
\end{aligned}$$

Hence, the conclusion of this lemma holds as desired. ■

**Proof** [Proof of Theorem 2] By the definition of $\varphi_{i,t}$ in (2), one has that for each $1 \leq i \leq n$,

$$\|\nabla\varphi_{i,t}(w^{t+1}) - \nabla\varphi_{i,t}(u_i^{t+1})\| \leq \|\nabla F_i(w^{t+1}) - \nabla F_i(u_i^{t+1})\| + \rho_i\|w^{t+1} - u_i^{t+1}\| \leq (L_{\nabla F} + \rho_i)\|w^{t+1} - u_i^{t+1}\|,$$

where the second inequality is due to (23) and the fact that $w^{t+1} \in \mathcal{Q}$ and $u_i^{t+1} \in \mathcal{Q}$ for all $1 \leq i \leq n$ (see Lemma 4). By the above inequality and the fact that $\|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty \leq \varepsilon_{t+1}$ (see Algorithm 2), one can obtain that

$$\begin{aligned}
\varepsilon_{t+1} + \sum_{i=1}^n \tilde{\varepsilon}_{i,t+1} &= \varepsilon_{t+1} + \sum_{i=1}^n \|[\nabla\varphi_{i,t}(w^{t+1}) - \rho_i(w^{t+1} - u_i^t)]\|_\infty \\
&\leq \varepsilon_{t+1} + \sum_{i=1}^n \|\nabla\varphi_{i,t}(u_i^{t+1})\|_\infty + \sum_{i=1}^n \|\nabla\varphi_{i,t}(w^{t+1}) - \nabla\varphi_{i,t}(u_i^{t+1})\| + \sum_{i=1}^n \rho_i\|w^{t+1} - u_i^t\| \\
&\leq (n+1)\varepsilon_{t+1} + \sum_{i=1}^n (L_{\nabla F} + \rho_i)\|w^{t+1} - u_i^{t+1}\| + \sum_{i=1}^n \rho_i\|w^{t+1} - u_i^t\|, \qquad (29)
\end{aligned}$$

where the first inequality is due to $\|u\|_\infty \leq \|u\|$ for all $u \in \mathbb{R}^d$ and the triangle inequality. Also, by (19), (24), and (26), one can see that

$$\begin{aligned}
\frac{\sigma}{4}\|w^{t+1} - u_i^{t+1}\|^2 &\leq \frac{\sigma}{2}\|w^{t+1} - \tilde{w}^*\|^2 + \frac{\sigma}{2}\|u_i^{t+1} - \tilde{w}^*\|^2 \overset{(19)}{\leq} \sum_{i=1}^n \left(\frac{\rho_i}{2}\|\tilde{w}^* - u_i^t\|^2 + \frac{1}{2\rho_i}\|\lambda_i^* - \lambda_i^t\|^2\right) \\
&\overset{(26)}{=} S_t \overset{(24)}{\leq} q_r^t \left[S_0 + \frac{1}{1-q}\left(\frac{n+1}{2\sigma} + \sum_{i=1}^n \frac{\sigma}{\rho_i^2 + 2L_{\nabla F}^2}\right)\right]. \qquad (30)
\end{aligned}$$

Using again (19), (24), and (26), we obtain that

$$\begin{aligned}\frac{1}{2}(\sum_{i=1}^{n}\rho_i\|w^{t+1}-u_i^t\|)^2 \le (\sum_{i=1}^{n}\rho_i)(\sum_{i=1}^{n}\frac{\rho_i}{2}\|w^{t+1}-u_i^t\|^2) &\overset{(19)}{\le} (\sum_{i=1}^{n}\rho_i)\sum_{i=1}^{n}(\frac{\rho_i}{2}\|\tilde{w}^*-u_i^t\|^2+\frac{1}{2\rho_i}\|\lambda_i^*-\lambda_i^t\|^2)\\ &\overset{(26)}{=} (\sum_{i=1}^{n}\rho_i)S_t \overset{(24)}{\le} (\sum_{i=1}^{n}\rho_i)q_r^t\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right],\end{aligned} \tag{31}$$

where the first inequality is due to the Cauchy-Schwarz inequality. Combining (29) with (30) and (31), we obtain that

$$\begin{aligned}&\varepsilon_{t+1}+\sum_{i=1}^{n}\tilde{\varepsilon}_{i,t+1}\\ &\le (n+1)q^t+\left(\frac{2}{\sqrt{\sigma}}\sum_{i=1}^{n}(L_{\nabla F}+\rho_i)+\sqrt{2\sum_{i=1}^{n}\rho_i}\right)\left[S_0+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right]^{1/2} q_r^{t/2}.\end{aligned} \tag{32}$$

Recall from Algorithm 2 and (15) that $(u_i^0,\lambda_i^0)=(\tilde{w}^0,-\nabla F_i(\tilde{w}^0))$ and $\lambda_i^*=-\nabla F_i(\tilde{w}^*)$. By these and (26), one has

$$S_0=\sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-\tilde{w}^0\|^2+\frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*)-\nabla F_i(\tilde{w}^0)\|^2\right). \tag{33}$$

For convenience, denote

$$\begin{aligned}b=&\left(\frac{2}{\sqrt{\sigma}}\sum_{i=1}^{n}(L_{\nabla F}+\rho_i)+\sqrt{2\sum_{i=1}^{n}\rho_i}\right)\\ &\times\left[\sum_{i=1}^{n}\left(\frac{\rho_i}{2}\|\tilde{w}^*-\tilde{w}^0\|^2+\frac{1}{2\rho_i}\|\nabla F_i(\tilde{w}^*)-\nabla F_i(\tilde{w}^0)\|^2\right)+\frac{1}{1-q}\left(\frac{n+1}{2\sigma}+\sum_{i=1}^{n}\frac{\sigma}{\rho_i^2+2L_{\nabla F}^2}\right)\right]^{1/2}.\end{aligned}$$

Using this, (32), and (33), we obtain that

$$\varepsilon_{t+1}+\sum_{i=1}^{n}\tilde{\varepsilon}_{i,t+1}\le (n+1)q^t+bq_r^{t/2}\le (n+1+b)q_r^{t/2}.$$

where the last inequality is due to $q\le q_r<1$. This along with the termination criterion implies that the number of iterations of Algorithm 2 is bounded above by

$$\left\lceil\frac{2\log(\tau/(n+1+b))}{\log q_r}\right\rceil+1=\mathcal{O}(|\log\tau|). \tag{34}$$

Hence, the conclusion of this theorem holds as desired. ■

We observe from the proof of Theorem 2 that the number of iterations of Algorithm 2 is bounded by the quantity in (34).

## Appendix D. Proof of Theorem 1

We define the Lagrangian function associated with problem (1) as

$$l(w,\mu)=\begin{cases} f(w)+h(w)+\langle\mu,c(w)\rangle & \text{if } w\in\mathrm{dom}(h) \text{ and } \mu\geq 0,\\ -\infty & \text{if } w\in\mathrm{dom}(h) \text{ and } \mu\ngeq 0,\\ \infty & \text{if } w\notin\mathrm{dom}(h),\end{cases}$$

Then one can verify that (e.g., see equation (17) in Lu and Zhou [46])

$$\partial l(w,\mu)=\begin{cases}\begin{pmatrix}\nabla f(w)+\partial h(w)+\nabla c(w)\mu\\ c(w)-\mathcal{N}_{\mathbb{R}^m_+}(\mu)\end{pmatrix} & \text{if } w\in\mathrm{dom}(h) \text{ and } \mu\geq 0,\\ \emptyset & \text{otherwise.}\end{cases} \tag{35}$$

We also define the set-valued operator associated with problems (1) as

$$\mathcal{T}_l:(w,\mu)\to\{(u,\nu)\in\mathbb{R}^d\times\mathbb{R}^m:(u,-\nu)\in\partial l(w,\mu)\},\quad \forall(w,\mu)\in\mathbb{R}^d\times\mathbb{R}^m\,. \tag{36}$$

In view of (35), (36), and the definition of KKT solution, we observe that finding an KKT solution of problems (1) is equivalent to solving the inclusion problem (see ([46])):

$$\text{Find}\quad (w,\mu)\in\mathbb{R}^d\times\mathbb{R}^m\quad \text{such that}\quad (0,0)\in\mathcal{T}_l(w,\mu). \tag{37}$$

Let $f_0(w)\equiv 0$ throughout this section. From Lemma 1 in [46], one can observe that

$$\nabla P_{i,k}(w)=\nabla f_i(w)+\nabla c_i(w)[\mu_i^k+\beta c_i(w)]_+ +\frac{1}{(n+1)\beta}(w-w^k),\quad \forall 0\leq i\leq n. \tag{38}$$

### D.1. Local Lipschitz continuity of $\nabla P_{i,k}$

**Lemma 8** *The gradients $\nabla P_{i,k}$, $0\leq i\leq n$, are locally Lipschitz continuous on $\mathbb{R}^d$.*

**Proof** Fix an arbitrary $w\in\mathbb{R}^d$ and a bounded open set $\mathcal{U}_w$ containing $w$. We suppose that $\nabla f_i$ is $L_{w,1}$-Lipschitz continuous on $\mathcal{U}_w$, and $\nabla c_i$ is $L_{w,2}$-Lipschitz continuous on $\mathcal{U}_w$. Also, let $U_{w,1}=\sup_{w\in\mathcal{U}_w}\|c_i(w)\|$ and $U_{w,2}=\sup_{w\in\mathcal{U}_w}\|\nabla c_i(w)\|$. By (4), and (38) one has for each $0\leq i\leq n$ and $u,v\in\mathcal{U}_w$ that

$$\begin{aligned}\|\nabla P_{i,k}(u)-\nabla P_{i,k}(v)\| &\overset{(38)}{\leq} \|\nabla f_i(u)-\nabla f_i(v)\|+\|\nabla c_i(u)-\nabla c_i(v)\|\|[\mu_i^k+\beta c_i(u)]_+\|\\ &\quad +\|[\mu_i^k+\beta c_i(u)]_+-[\mu_i^k+\beta c_i(v)]_+\|\|\nabla c_i(v)\|+\frac{1}{(n+1)\beta}\|u-v\|\\ &\leq L_{w,1}\|u-v\|+(\|\mu_i^k\|+\beta U_{w,1})L_{w,2}\|u-v\|\\ &\quad +\beta\|c_i(u)-c_i(v)\|\|\nabla c_i(v)\|+\frac{1}{(n+1)\beta}\|u-v\|\\ &\leq\left[L_{w,1}+(\|\mu_i^k\|+\beta U_{w,1})L_{w,2}+\beta U_{w,2}^2+\frac{1}{(n+1)\beta}\right]\|u-v\|.\end{aligned}$$

Therefore, $\nabla P_{i,k}(u)$ is locally Lipschitz continuous on $\mathbb{R}^d$, and the conclusion holds as desired. ■

## D.2. Output of Algorithm 1

**Theorem 9** *If Algorithm 1 successfully terminates, its output $(w^{k+1}, \mu^{k+1})$ is an $(\epsilon_1, \epsilon_2)$-KKT solution of problem (1).*

**Proof** Notice from (2) that

$$\ell_k(w) = f(w) + h(w) + \frac{1}{2\beta}[[\mu^k + \beta c(w)]_+^2 - \|\mu^k\|^2] + \frac{1}{2\beta}\|w - w^k\|^2.$$

By this, (35), and the fact that $\mu^{k+1} = \Pi_{\mathcal{K}^*}(\mu^k + \beta c(w^{k+1}))$, one has

$$\begin{aligned} \partial\ell_k(w^{k+1}) - \frac{1}{\beta}(w^{k+1} - w^k) &= \nabla f(w^{k+1}) + \partial h(w^{k+1}) + \nabla c(w^{k+1})[\mu^k + \beta c(w)]_+ \\ &= \nabla f(w^{k+1}) + \partial h(w^{k+1}) + \nabla c(w^{k+1})\mu^{k+1} = \partial_w l(w^{k+1}, \mu^{k+1}). \end{aligned} \tag{39}$$

Using similar arguments as for the second relation of equation (52) in [46], we obtain that

$$\frac{1}{\beta}(\mu^{k+1} - \mu^k) \in \partial_\mu l(w^{k+1}, \mu^{k+1}). \tag{40}$$

In view of this and (39), one can see that

$$\mathrm{dist}_\infty(0, \partial_w l(w^{k+1}, \mu^{k+1})) \overset{(39)}{\le} \mathrm{dist}_\infty(0, \partial\ell_k(w^{k+1})) + \frac{1}{\beta}\|w^{k+1} - w^k\|_\infty \le \tau_k + \frac{1}{\beta}\|w^{k+1} - w^k\|_\infty \le \epsilon_1,$$

$$\mathrm{dist}_\infty(0, \partial_\mu l(w^{k+1}, \mu^{k+1})) \le \frac{1}{\beta}\|\mu^{k+1} - \mu^k\|_\infty \le \epsilon_2.$$

These along with (35) imply that $(w^{k+1}, \mu^{k+1})$ is an $(\epsilon_1, \epsilon_2)$-KKT solution of problem (1), which proves this theorem as desired. ■

## D.3. Bounded iterates of Algorithm 1

**Lemma 10 (Bounded iterates of Algorithm 1)** *Let $\{w^k\}_{k\in\mathbb{K}}$ be all the iterates generated by Algorithm 1, where $\mathbb{K}$ is a subset of consecutive nonnegative integers starting from 0. Then we have $w^k \in \mathcal{Q}_1$ for all $k \in \mathbb{K}$, where*

$$\mathcal{Q}_1 = \{w \in \mathbb{R}^d : \|w - w^*\| \le r_0 + 2\bar{s}\beta\}, \quad r_0 = \|(w^0, \mu^0) - (w^*, \mu^*)\|, \tag{41}$$

*and $w^0$, $\mu^0$, $\bar{s}$, and $\beta$ are inputs of Algorithm 1.*

We define $\mathbb{K} - 1 = \{k - 1 : k \in \mathbb{K}\}$, and for any $0 \le k \in \mathbb{K} - 1$, define

$$w_*^k = \arg\min_w \ell_k(w), \quad \mu_*^k = \Pi_{\mathcal{K}^*}(\mu^k + \beta c(w_*^k)). \tag{42}$$

The following lemma shows that the update from $(w^k, \mu^k)$ to $(w^{k+1}, \mu^{k+1})$ can be viewed as applying an inexact proximal point algorithm (PPA) to the inclusion problem (37). Its proof can be found in Lemma 5 in [46].

**Lemma 11** *Let $\{(w^k,\mu^k)\}_{k\in\mathbb{K}}$ be generated by Algorithm 1. Then for any $k\in\mathbb{K}$, we have*

$$\|(w^{k+1},\mu^{k+1})-\mathcal{J}_\beta(w^k,\mu^k)\|\le\beta\tau_k,$$

*where $\mathcal{J}_\beta=(\mathcal{I}+\beta\mathcal{T}_l)^{-1}$, $\mathcal{I}$ is the identity mapping, and $\mathcal{T}_l$ is defined in (36).*

The following lemma establishes some properties of $(w^k,\mu^k)$ and $(w_*^k,\mu_*^k)$. Its proof can be found in Lemma 14 in [45].

**Lemma 12** *Let $\{(w^k,\mu^k)\}_{k\in\mathbb{K}}$ be generated by Algorithm 1, where $\mathbb{K}$ is defined in Lemma 10. Let $(w_*^k,\mu_*^k)$ be defined in (42) for all $0\le k\le\mathbb{K}-1$. Then the following relations hold.*

$$\|(w^k,\mu^k)-(w_*^k,\mu_*^k)\|^2+\|(w_*^k,\mu_*^k)-(w^*,\mu^*)\|^2\le\|(w^k,\mu^k)-(w^*,\mu^*)\|^2,\quad\forall 0\le k\le\mathbb{K}-1,$$
$$\|(w^k,\mu^k)-(w^*,\mu^*)\|\le\|(w^0,\mu^0)-(w^*,\mu^*)\|+\beta\sum_{j=0}^{k-1}\tau_j,\quad\forall 0\le k\in\mathbb{K}.$$

Notice from Algorithm 1 that $\tau_k=\bar{s}/(k+1)^2$ for all $k\ge0$. Therefore, one has $\sum_{j=0}^{\infty}\tau_j\le2\bar{s}$. In view of this and Lemma 12, we observe that

$$\|w^k-w^*\|\le r_0+2\bar{s}\beta,\quad\|\mu^k-\mu^*\|\le r_0+2\bar{s}\beta,\quad\forall0\le k\in\mathbb{K},\tag{43}$$
$$\|w^k-w_*^k\|\le r_0+2\bar{s}\beta,\quad\|w_*^k-w^*\|\le r_0+2\bar{s}\beta,\quad\forall0\le k\in\mathbb{K}-1.\tag{44}$$

where $r_0$ is defined in (41), and $\beta$ and $\bar{s}$ are inputs of Algorithm 1. The first relation in (43) leads to the conclusion that $w^k\in\mathcal{Q}_1$ for all $k\in\mathbb{K}$, which immediately implies that Lemma 10 holds.

### D.4. Proof of Theorem 1

We provide a technical lemma concerning the convergence rate of an inexact PPA applied to a monotone inclusion problem. Its proof can be found in Lemma 3 in [46].

**Lemma 13** *Let $\mathcal{T}:\mathbb{R}^p\rightrightarrows\mathbb{R}^q$ be a maximally monotone operator and $z^*\in\mathbb{R}^p$ such that $0\in\mathcal{T}(z^*)$. Let $\{z^k\}$ be a sequence generated by an inexact PPA, starting with $z^0$ and obtaining $z^{k+1}$ be approximately evaluating $\mathcal{J}_\beta(z^k)$ such that*

$$\|z^{k+1}-\mathcal{J}_\beta(z^k)\|\le e_k$$

*for some $\beta>0$ and $e_k\ge0$, where $\mathcal{J}_\beta=(\mathcal{I}+\beta\mathcal{T})^{-1}$ and $\mathcal{I}$ is the identity operator. Then, for any $K\ge1$, we have*

$$\min_{K\le k\le2K}\|z^{k+1}-z^k\|\le\frac{\sqrt{2}\left(\|z^0-z^*\|+2\sum_{k=0}^{2K}e_k\right)}{\sqrt{K+1}}.$$

Recall from (41) that $\mathcal{Q}_1$ is a compact set. We let

$$U_{\nabla f}=\sup_{w\in\mathcal{Q}_1}\max_{1\le i\le n}\|\nabla f_i(w)\|,\quad U_{\nabla c}=\sup_{w\in\mathcal{Q}_1}\max_{0\le i\le n}\|\nabla c_i(w)\|,\quad U_c=\sup_{w\in\mathcal{Q}_1}\max_{0\le i\le n}\|c_i(w)\|.\tag{45}$$

**Lemma 14** *Let $\{w^{k,t+1}\}_{t\in\mathbb{T}_k}$ and $\{u_i^{k,t+1}\}_{1\le i\le n, t\in\mathbb{T}_k}$ be all the iterates generated by Algorithm 2 for solving the subproblem (3) at the kth iteration of Algorithm 1, where $\mathbb{T}_k$ is a consecutive non-negative integers starting from 0. Then we have $w^{k,t+1}\in\mathcal{Q}_2$ and $u_i^{k,t+1}\in\mathcal{Q}_2$ for all $t\in\mathbb{T}_k$ and $1\le i\le n$, where*

$$\mathcal{Q}_2=\left\{v:\|v-u\|^2\le\frac{(n+1)^3\beta^2}{(1-q^2)}+(n+1)\beta\sum_{i=1}^n\left[\rho_i(r_0+2\bar{s}\beta)^2+\frac{4}{\rho_i}U_{\nabla P}^2\right],u\in\mathcal{Q}_1\right\}, \tag{46}$$

$$U_{\nabla P}=U_{\nabla f}+\frac{2(r_0+2\bar{s}\beta)}{(n+1)\beta}+U_{\nabla c}\left(\|\mu^*\|+r_0+2\bar{s}\beta+\beta U_c\right). \tag{47}$$

**Proof** By (38) and the definition of $P_{i,k}$ in (4), one has for all $w\in\mathcal{Q}_1$ and $1\le i\le n$ that

$$\begin{aligned}\|\nabla P_{i,k}(w)\| &\overset{(38)}{=} \|\nabla f_i(w)+\nabla c_i(w)\Pi_{\mathcal{K}_i^*}(\mu_i^k+\beta c_i(w))+\frac{1}{(n+1)\beta}(w-w^k)\|\\ &\le \|\nabla f_i(w)\|+\|\nabla c_i(w)\|\|\Pi_{\mathcal{K}_i^*}(\mu_i^k+\beta c_i(w))\|+\frac{1}{(n+1)\beta}\|w-w^k\|\\ &\overset{(45)}{\le} U_{\nabla f}+U_{\nabla c}(\|\mu_i^*\|+\|\mu_i^k-\mu_i^*\|+\beta U_c)+\frac{1}{(n+1)\beta}(\|w-w^*\|+\|w^k-w^*\|)\\ &\overset{(41)(43)}{\le} U_{\nabla f}+U_{\nabla c}\left(\|\mu^*\|+r_0+2\bar{s}\beta+\beta U_c\right)+\frac{2(r_0+2\bar{s}\beta)}{(n+1)\beta}\overset{(47)}{=}U_{\nabla P}.\end{aligned} \tag{48}$$

Recall that Algorithm 2 with $(\tilde{w}^0,\tau)=(w^k,\tau_k)$ is applicable to the subproblem (3). In addition, recall that the subproblem (3) has an optimal solution $w_*^k$ (see (42)), $P_{i,k}$, $0\le i\le n$, are strongly convex with modulus $1/[(n+1)\beta]$. By Lemma 4 with $(F_i,\tilde{w}^*,\tilde{w}^0,\sigma)=(P_{i,k},w_*^k,w^k,1/[(n+1)\beta])$, we obtain that $w^{k,t+1}\in\widetilde{\mathcal{Q}}$ and $u_i^{k,t+1}\in\widetilde{\mathcal{Q}}$ for all $t\in\mathbb{T}_k$ and $1\le i\le n$, where

$$\widetilde{\mathcal{Q}}=\left\{v:\|v-w_*^k\|^2\le\frac{(n+1)^3\beta^2}{(1-q^2)}+(n+1)\beta\sum_{i=1}^n\left(\rho_i\|w_*^k-w^k\|^2+\frac{1}{\rho_i}\|\nabla P_{i,k}(w_*^k)-\nabla P_{i,k}(w^k)\|^2\right)\right\}. \tag{49}$$

Notice from (44) that $\|w_*^k-w^k\|\le r_0+2\bar{s}\beta$. It follows from (41), (43), and (44) that $w^k,w_*^k\in\mathcal{Q}_1$. By these, (48), and (49), one has that

$$\widetilde{\mathcal{Q}}\subseteq\left\{v:\|v-w_*^k\|^2\le\frac{(n+1)^3\beta^2}{(1-q^2)}+(n+1)\beta\sum_{i=1}^n\left[\rho_i(r_0+2\bar{s}\beta)^2+\frac{4}{\rho_i}U_{\nabla P}^2\right]\right\}.$$

This along with (46) and the fact that $w_*^k\in\mathcal{Q}_1$ implies that the conclusion of this lemma holds as desired. ■

Let $L_{\nabla f,2}$ be the Lipschitz constant of $\nabla f_i$, $1\le i\le n$, on $\mathcal{Q}_2$, and $L_{\nabla c,2}$ be the Lipschitz constant of $\nabla c_i$, $0\le i\le n$, on $\mathcal{Q}_2$. Also, we let

$$U_{\nabla c,2}=\sup_{w\in\mathcal{Q}_2}\max_{0\le i\le n}\|\nabla c_i(w)\|,\quad U_{c,2}=\sup_{w\in\mathcal{Q}_2}\max_{0\le i\le n}\|c_i(w)\|. \tag{50}$$

We define

$$L_{\nabla P,2} = L_{\nabla f,2} + (\|\mu^*\| + r_0 + 2\bar{s}\beta + \beta U_{c,2})L_{\nabla c,2} + \beta U_{\nabla c,2}^2 + \frac{1}{(n+1)\beta}. \tag{51}$$

By the local Lipschitz continuity of $\nabla f_i$, $1 \le i \le n$, and $\nabla c_i$, $0 \le i \le n$, and a similar argument as in the proof of Lemma 6, one can observe that $L_{\nabla f,2}$, $L_{\nabla c,2}$, and $L_{\nabla P,2}$ are well-defined.

To proceed, we next show that $\nabla P_{i,k}$, $0 \le i \le n$, are $L_{\nabla P,2}$-Lipschitz continuous on $\mathcal{Q}_2$. By the definitions of $L_{\nabla f,2}$ and $L_{\nabla c,2}$, (4), (38), (43), (50), and (51), one has that for all $u, v \in \mathcal{Q}_2$ and $0 \le i \le n$,

$$\begin{aligned}
\|\nabla P_{i,k}(u) - \nabla P_{i,k}(v)\| &\overset{(38)}{\le} \|\nabla f_i(u) - \nabla f_i(v)\| + \|[\mu_i^k + \beta c_i(w)]_+\|\|\nabla c_i(u) - \nabla c_i(v)\| \\
&\quad + \|[\mu_i^k + \beta c_i(u)]_+ - [\mu_i^k + \beta c_i(v)]_+\|\|\nabla c_i(v)\| + \frac{1}{(n+1)\beta}\|u - v\| \\
&\overset{(50)}{\le} L_{\nabla f,2}\|u - v\| + (\|\mu_i^*\| + \|\mu_i^k - \mu_i^*\| + \beta U_{c,2})L_{\nabla c,2}\|u - v\| \\
&\quad + \beta U_{\nabla c,2}^2\|u - v\| + \frac{1}{(n+1)\beta}\|u - v\| \\
&\overset{(43)}{\le} \left[L_{\nabla f,2} + (\|\mu^*\| + r_0 + 2\bar{s}\beta + \beta U_{c,2})L_{\nabla c,2} + \beta U_{\nabla c,2}^2 + \frac{1}{(n+1)\beta}\right]\|u - v\| \\
&\overset{(51)}{=} L_{\nabla P,2}\|u - v\|.
\end{aligned} \tag{52}$$

**Proof** [Proof of Theorem 1] We first derive an upper bound for the number of outer iterations of Algorithm 1. Recall that $\sum_{j=0}^{\infty} \tau_j = 2\bar{s}$. It follows from Lemmas 11 and 13 that

$$\begin{aligned}
\min_{K \le k \le 2K} \frac{1}{\beta}\|(w^{k+1}, \mu^{k+1}) - (w^k, \mu^k)\| &\le \frac{\sqrt{2}\left(\|(w^0, \mu^0) - (w^*, \mu^*)\| + 2\beta \sum_{j=0}^{\infty} \tau_j\right)}{\beta\sqrt{K+1}} \\
&\le \frac{\sqrt{2}\left(\|(w^0, \mu^0) - (w^*, \mu^*)\| + 4\bar{s}\beta\right)}{\beta\sqrt{K+1}} = \frac{\sqrt{2}\,(r_0 + 4\bar{s}\beta)}{\beta\sqrt{K+1}},
\end{aligned}$$

which then implies that

$$\begin{aligned}
&\min_{K \le k \le 2K} \left\{\tau_k + \frac{1}{\beta}\|w^{k+1} - w^k\|_\infty\right\} \le \frac{\bar{s}}{(K+1)^2} + \frac{\sqrt{2}\,(r_0 + 4\bar{s}\beta)}{\beta\sqrt{K+1}} \le \left[\bar{s} + \frac{\sqrt{2}\,(r_0 + 4\bar{s}\beta)}{\beta}\right]\frac{1}{\sqrt{K+1}}, \\
&\min_{K \le k \le 2K} \frac{1}{\beta}\|\mu^{k+1} - \mu^k\|_\infty \le \frac{\sqrt{2}\,(r_0 + 4\bar{s}\beta)}{\beta\sqrt{K+1}}.
\end{aligned}$$

We see from these and the termination criterion that the number of outer iterations of Algorithm 1 is at most

$$K_{\epsilon_1,\epsilon_2} = 2\left[\bar{s} + \frac{\sqrt{2}(r_0 + 4\bar{s}\beta)}{\beta}\right]^2 \max\{\epsilon_1^{-2}, \epsilon_2^{-2}\} = \mathcal{O}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\}).$$

We next derive an upper bound for the total number of inner iterations of Algorithm 1. Recall from (4) that $P_{i,k}$, $0 \le i \le n$, are strongly convex with modulus $1/[(n+1)\beta]$. In addition, notice from Lemma 8 that $P_{i,k}$, $0 \leq i \leq n$, are locally Lipschitz continuous on $\mathbb{R}^d$. Therefore, Algorithm 2 is applicable to the subproblem (3).

From Lemma 14, we see that all iterates generated by Algorithm 2 for solving (3) lie in $\mathcal{Q}_2$. Also, in view of (52), we see that $\nabla P_{i,k}$, $1 \leq i \leq n$, are $L_{\nabla P,2}$-Lipschitz continuous on $\mathcal{Q}_2$. Therefore, by Theorem 2 with $(\tau, F_i, \sigma, L_{\nabla F}, \tilde{w}^*, \tilde{w}^0) = (\tau_k, P_{i,k}, 1/[(n+1)\beta], L_{\nabla P,2}, w^k_*, w^k)$ and the discussion at the end of Appendix C.3, one can see that the number of iterations of Algorithm 2 for solving (3) is no more than

$$T_k = \left\lceil \frac{2\log(\tau_k/(n+1+b_k))}{\log \tilde{q}_r} \right\rceil + 1 \tag{53}$$

where

$$\tilde{q}_r = \max\left\{q, \frac{1}{1+\tilde{r}}\right\}, \quad \tilde{r} = \min_{1\leq i\leq n}\left\{\frac{\rho_i}{(n+1)\beta(\rho_i^2 + 2L^2_{\nabla P,2})}\right\},$$

$$b_k = \left(2\sqrt{(n+1)\beta}\sum_{i=1}^n (L_{\nabla P,2} + \rho_i) + \sqrt{2(\sum_{i=1}^n \rho_i)}\right.$$
$$\times \left[\sum_{i=1}^n\left(\frac{\rho_i}{2}\|w^k_* - w^k\|^2 + \frac{1}{2\rho_i}\|\nabla P_{i,k}(w^k_*) - \nabla P_{i,k}(w^k)\|^2\right) + \frac{1}{1-q}\left(\frac{(n+1)^2\beta}{2} + \frac{1}{(n+1)\beta}\sum_{i=1}^n \frac{1}{\rho_i^2 + 2L^2_{\nabla P,2}}\right)\right]$$

Recall from (43), (44), and the definitions of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ that $w^k_*, w^k \in \mathcal{Q}_1 \subseteq \mathcal{Q}_2$. It then follows that

$$\frac{\rho_i}{2}\|w^k_* - w^k\|^2 + \frac{1}{2\rho_i}\|\nabla P_{i,k}(w^k_*) - \nabla P_{i,k}(w^k)\|^2 \overset{(52)}{\leq} \frac{\rho_i^2 + L^2_{\nabla P,2}}{2\rho_i}\|w^k_* - w^k\|^2 \overset{(44)}{\leq} \frac{\rho_i^2 + L^2_{\nabla P,2}}{2\rho_i}(r_0 + 2\bar{s}\beta)^2.$$

Then one has $b_k \leq \bar{b}$, where

$$\bar{b} = \left(2\sqrt{(n+1)\beta}\sum_{i=1}^n (L_{\nabla P,2} + \rho_i) + \sqrt{2(\sum_{i=1}^n \rho_i)}\right.$$
$$\left.\times \left[\sum_{i=1}^n \frac{\rho_i^2 + L^2_{\nabla P,2}}{2\rho_i}(r_0 + 2\bar{s}\beta)^2 + \frac{1}{1-q}\left(\frac{(n+1)^2\beta}{2} + \frac{1}{(n+1)\beta}\sum_{i=1}^n \frac{1}{\rho_i^2 + 2L^2_{\nabla P,2}}\right)\right]^{1/2}\right).$$

By $b_k \leq \bar{b}$, $\tau_k = \bar{s}/(k+1)^2$, $k \leq K_{\epsilon_1,\epsilon_2}$, and (53), one has that

$$T_k \leq \left\lceil \frac{2\log((n+1+\bar{b})(K_{\epsilon_1,\epsilon_2}+1)^2/\bar{s})}{\log(\tilde{q}_r^{-1})} \right\rceil + 1.$$

Therefore, by $K_{\epsilon_1,\epsilon_2} = \mathcal{O}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\})$, one can see that the total number of inner iterations of Algorithm 1 is at most

$$\sum_{k=0}^{K_{\epsilon,\epsilon_2}} T_k \leq (K_{\epsilon_1,\epsilon_2}+1)\left(\left\lceil \frac{2\log((n+1+\bar{b})(K_{\epsilon_1,\epsilon_2}+1)^2/\bar{s})}{\log(\tilde{q}_r^{-1})} \right\rceil + 1\right) = \widetilde{\mathcal{O}}(\max\{\epsilon_1^{-2}, \epsilon_2^{-2}\}).$$

This completes the proof as desired. ■

## Appendix E. A centralized proximal AL method

In this part, we present a centralized proximal AL method (see Algorithm 2 in [46]) for solving the convex constrained optimization problem:

$$\min_w \; f(w) + h(w) \quad \text{s. t.} \quad c(w) \leq 0, \tag{54}$$

where the function $f : \mathbb{R}^d \to \mathbb{R}$ and the mapping $c : \mathbb{R}^d \to \mathbb{R}^m$ are continuous differentiable and convex.

**Algorithm 3:** A centralized proximal AL method for solving problem (54)

```
**Input**: tolerances $\epsilon_1, \epsilon_2 \in (0,1)$, $w^0 \in \mathrm{dom}(h)$, $\mu^0 \geq 0$, and $\beta > 0$. **for** $k = 0, 1, 2, \ldots$ **do**
    Find an approximate solution $w^{k+1}$ to the proximal AL subproblem:
        $\min_w \left\{ \ell_k(w) = f(w) + h(w) + \frac{1}{2\beta}\left(\|[\mu^k + \beta c(w)]_+\|^2 - \|\mu^k\|^2\right) + \frac{1}{2\beta}\|w - w^k\|^2 \right\}$
    such that
        $\mathrm{dist}_\infty(0, \partial \ell_k(w^{k+1})) \leq \tau_k.$
    Update the Lagrangian multiplier:
        $\mu^{k+1} = [\mu^k + \beta c(w^{k+1})]_+.$
    Output $(w^{k+1}, \mu^{k+1})$ and terminate the algorithm if
        $\|w^{k+1} - w^k\|_\infty + \beta\tau_k \leq \beta\epsilon_1, \qquad \|\mu^{k+1} - \mu^k\|_\infty \leq \beta\epsilon_2.$
**end**
```

## Appendix F. Experiment description

### F.1. Linear equality constrained quadratic programming

In this subsection we consider the linear equality constrained quadratic programming problem:

$$\min_w \sum_{i=1}^n \left( \frac{1}{2} w^T A_i w + b_i^T w \right) \quad \text{s. t.} \quad C_i w + d_i = 0, \quad 0 \leq i \leq n, \tag{55}$$

where $A_i \in \mathbb{R}^{d \times d}$, $1 \leq i \leq n$, are positive semidefinite, $b_i \in \mathbb{R}^d$, $1 \leq i \leq n$, $C_i \in \mathbb{R}^{\tilde{m} \times d}$, $0 \leq i \leq n$, and $d_i \in \mathbb{R}^{\tilde{m}}$, $0 \leq i \leq n$.

For each $(d, n, \tilde{m})$, we randomly generate 10 instances of problem (55). In particular, for each $1 \leq i \leq n$, we first randomly generate matrix $A_i$ by letting $A_i = U_i D_i U_i^T$, where $D_i \in \mathbb{R}^{d \times d}$ is a diagonal matrix, whose diagonal entries are randomly generated according to the uniform distribution over $[5, 10]$ and $U_i \in \mathbb{R}^{d \times d}$ is a randomly generated orthogonal matrix. We then randomly generate $b_i$, $1 \leq i \leq n$, $C_i$, $0 \leq i \leq n$, and $d_i$, $0 \leq i \leq n$, with all entries chosen from the standard normal distribution.

The computational results of Algorithm 1 and the centralized proximal AL method (abbreviated as cProx-AL) for solving the randomly generated instances are presented in Table 1. Our aim is to apply Algorithm 1 and a centralized proximal AL method to find a $(10^{-3}, 10^{-3})$-KKT solution of problem (55), and compare their performances. In particular, we solve the convex quadratic programming subproblems (2) and (2) arising in Algorithm 1 by seeking a root to the linear equation derived from equating the gradient to zero. In addition, the centralized proximal AL method follows the same framework as Algorithm 1 except that the $w^{k+1}$ is obtained by directly seeking a root to the linear equation derived from equating the gradient of (3) to zero. We set parameters for Algorithm 1 and the centralized proximal AL method as $w^0 = (1, \ldots, 1)^T$, $\mu_i^0 = (0, \ldots, 0)^T$ $\forall 0 \le i \le n$, $\bar{s} = 0.01$ and $\beta = 1$. We also set $\rho_i = 1$ $\forall 1 \le i \le n$ for Algorithm 2.

In detail, the value of $d$, $n$, and $\tilde{m}$ is listed in the first three columns, respectively. For each triple $(d, n, \tilde{m})$, the average objective value, the average feasibility violation, the average number of outer iterations, and the average total number of iterations over 10 random instances are given in the rest columns.

### F.2. Neyman-pearson classification

We consider the Neyman-Pearson binary classification problem:

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \phi(w; \{x_j^{(i0)}\}_{1 \le j \le m_{i0}}) \quad \text{s.t.} \quad \phi(w; \{x_j^{(i1)}\}_{1 \le j \le m_{i1}}) \le r_i, \quad 1 \le i \le n, \tag{56}$$

where $\{x_j^{(i0)}\}_{1 \le j \le m_{i0}}$ and $\{x_j^{(i1)}\}_{1 \le j \le m_{i1}}$ are the sets of samples in client $i$ associated with labels 0 and 1, respectively, and $\phi$ is the binary logistic loss (see Section 4.4.1 in Hastie et al. [24])

$$\phi(w; \{x_j^{(is)}\}_{1 \le j \le m_{is}}) = \frac{1}{m_{is}} \sum_{j=1}^{m_{is}} \left[ -s w^T x_j^{(is)} + \log(1 + e^{w^T x_j^{(is)}}) \right], \quad s \in \{0, 1\}.$$

We consider three real-world datasets, namely 'breast-cancer-wisc', 'adult', and 'monks-1', from the UCI repository. For each dataset, we conducted an imbalanced classification task that minimizes the binary classification loss while ensuring the loss for class 1 (minority) less than a threshold $r = 0.2$. To simulate the FL setting, we divided each dataset into $n$ folds, mimicking distributed clients each holding the same amount of data with equal imbalanced ratios. Each experiment is repeated three times to account for randomness.

We compare Algorithm 1 for solving the Neyman-Pearson classification model (56) and Algorithm 2 for directly minimizing the unconstrained binary classification model:

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \left[ \phi(w; \{x_j^{(i0)}\}_{1 \le j \le m_{i0}}) + \phi(w; \{x_j^{(i1)}\}_{1 \le j \le m_{i1}}) \right]. \tag{57}$$

In particular, we apply Algorithm 1 to find an $(10^{-2}, 10^{-2})$-KKT solution of problem (56). In addition, we apply Algorithm 2 to find an approximate solution of (57) such that the gradient of the objective is less than $10^{-2}$. We set the parameters for Algorithms 2 and 1 the same as the experiments for linearly constrained quadratic programming.

The computational results for solving the Neyman-Pearson classification and unconstrained logistic regression using three real-world datasets are presented in Table 2. In detail, the first four columns of Table 2 represent the names of the dataset, numbers of samples in class 1 and 0, number of features, and number of clients. In the last two columns, we present the losses for class 0 and class 1, respectively, which include results computed from the Neyman-Pearson classification and the unconstrained logistic regression. We include the mean and max loss values for class 1 among all local clients.

### F.3. Classification with fairness constraints

In this subsection we consider the classification with global and local fairness constraints:

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\phi(w;(x_j^{(i)},y_j^{(i)})) \tag{58a}$$

$$\text{s.t.} \ -r_i \le \frac{1}{\tilde{m}_i}\sum_{j=1}^{\tilde{m}_i}\phi(w;(\tilde{x}_j^{(i)},\tilde{y}_j^{(i)})) - \frac{1}{\hat{m}_i}\sum_{j=1}^{\hat{m}_i}\phi(w;(\hat{x}_j^{(i)},\hat{y}_j^{(i)})) \le r_i, \quad 0 \le i \le n. \tag{58b}$$

where $\phi$ is the logistic loss defined as in (F.2), $(x_j^{(i)},y_j^{(i)}) \in \mathbb{R}^d \times \{0,1\}$, $1 \le j \le m_i$, are the feature-label pairs at client $i$. For each client $i$, the local dataset $\{(x_j^{(i)},y_j^{(i)})\}_{1\le j\le m_i}$ is divided into two sensitive groups $\{(\tilde{x}_j^{(i)},\tilde{y}_j^{(i)})\}_{1\le j\le \tilde{m}_i}$ and $\{(\hat{x}_j^{(i)},\hat{y}_j^{(i)})\}_{1\le j\le \hat{m}_i}$. The global dataset at the central server also includes two sensitive groups of samples $\{(\tilde{x}_j^{(0)},\tilde{y}_j^{(0)})\}_{1\le j\le \tilde{m}_0}$ and $\{(\hat{x}_j^{(0)},\hat{y}_j^{(0)})\}_{1\le j\le \hat{m}_0}$.

We consider the real-world dataset named ‘adult-b’ consisting of a training set and a testing set.[2] Each sample in this dataset has 39 features and a binary label. We conducted a binary classification task with fairness constraints that control the loss disparity between two sensitive groups of samples. We allocate 22,654 samples from the training set to the local dataset at clients, and 5,659 samples from the testing set to form the global dataset at the central server. To simulate an FL setting, we partitioned each dataset into $n$ folds, ensuring an equal number of samples at each client.

We apply Algorithm 1 and cProx-AL to find an $(10^{-3},10^{-3})$-KKT solution of problem (58). cProx-AL is presented in Algorithm 3, where $w^{k+1}$ is obtained by applying L-BFGS method built in scipy.optimize.minimize to solve the subproblem. We run 10 trials of Algorithm 1 and cProx-AL, where for each run, both algorithms have the same initial point $w^0$, randomly chosen from the unit Euclidean sphere. We set the other parameters for Algorithm 1 and the cProx-AL method as $\mu_i^0 = (0,\ldots,0)^T \ \forall 0 \le i \le n$, $\bar{s} = 0.001$ and $\beta = 10$. We also set $\rho_i = 10^8 \ \forall 1 \le i \le n$ for Algorithm 2.

The computational results for solving problems (58) are presented in Table 3. In detail, the first column of Table 3 represents the number of clients. In the last two columns, we present the classification loss and loss disparity, respectively, which include results computed from the classification model with fairness constraints in (58). By computing the average of 10 random trials, we include the relative difference of the objective value between Algorithm 1 and cProx-AL, and also the mean and max loss disparity (absolute difference of losses for two sensitive groups) among all clients and the central server. The respective standard deviations are listed in parentheses. Comparing the classification loss and loss disparity of Algorithm 1 and cProx-AL in Table 3 reveals that both Algorithm 1 and cProx-AL can yield solutions of similar quality. Given the small standard deviation,

[2] This dataset can be found in https://github.com/heyaudace/ml-bias-fairness/tree/master/data/adult.

Table 3: Numerical results for problem (58).

| $n$ | objective value | | | loss disparity ($\leq 0.1$) | | | |
|---|---|---|---|---|---|---|---|
| | Algorithm 1 | cProx-AL | relative difference | Algorithm 1 | | cProx-AL | |
| | | | | mean | max | mean | max |
| 1 | 0.37 (9.83e-05) | 0.37 (4.14e-05) | 1.97e-03 (2.53e-04) | 0.10 (1.14e-04) | 0.10 (1.36e-04) | 0.10 (3.69e-06) | 0.10 (5.38e-06) |
| 5 | 0.37 (3.99e-03) | 0.37 (4.05e-03) | 1.86e-03 (4.69e-04) | 0.09 (5.34e-05) | 0.10 (7.51e-05) | 0.09 (3.68e-05) | 0.10 (4.36e-06) |
| 10 | 0.37 (6.39e-03) | 0.37 (6.52e-03) | 2.39e-03 (8.40e-04) | 0.08 (1.68e-04) | 0.10 (2.15e-05) | 0.08 (1.52e-04) | 0.10 (6.56e-06) |
| 20 | 0.38 (9.46e-03) | 0.37 (9.86e-03) | 4.61e-03 (2.43e-03) | 0.08 (9.75e-05) | 0.10 (1.01e-04) | 0.08 (4.90e-05) | 0.10 (6.06e-06) |

we observe that the convergence behavior of Algorithm 1 remains stable across 10 trial runs. These observations demonstrate the ability of Algorithm 1 to solve the problem in an FL framework stably without compromising solution quality, and it also implies the potential of our algorithm in solving FL problems with particular nonconvex constraints.

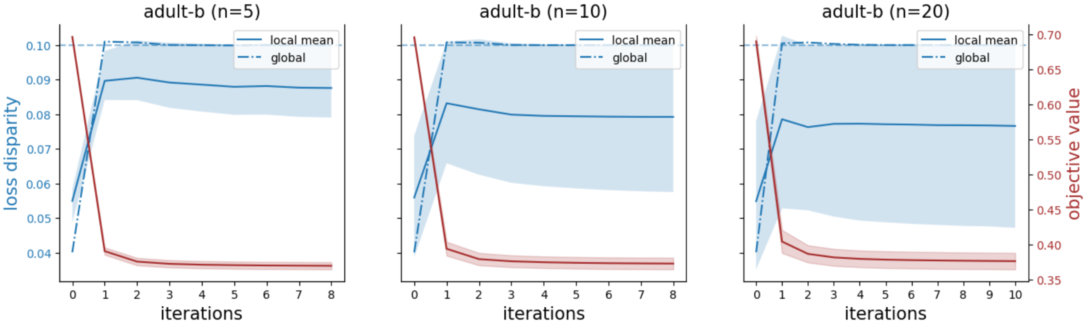

Figure 2: Convergence behavior of loss disparity and classification loss across all local clients in one random trial, over the outer iterations of Algorithm 1 on the adult dataset. The solid blue and brown lines indicate the convergence behavior of the average loss disparity and classification loss over all clients, respectively. The blue and brown shaded areas indicate thregions between the maximum value and minimum value of loss disparity and classification loss over all clients, respectively. The blue dashdot line indicates the convergence behavior of the global loss disparity in the central server.

Figure 2 shows the convergence behavior of loss disparity and classification loss across all local clients in one random trial, over the outer iterations of Algorithm 1. From this figure, we see that our proposed method consistently relegates the loss disparities (local/global constraints) on all clients and the central server to a level below a threshold ($\leq 0.1$) while also consistently minimizing the classification losses (local objectives) on all local clients.