Journal of Machine Learning Research 21 (2020) 1-39 Submitted 8/19; Revised 1/20; Published 3/20

GADMM: Fast and Communication Efficient Framework for Distributed Machine Learning

Anis Elgabli ANIS.ELGABLI@OULU.FI
Jihong Park JIHONG.PARK@OULU.FI
Amrit S. Bedi AMRITBD@IITK.AC.IN
Mehdi Bennis MEHDI.BENNIS@OULU.FI
Vaneet Aggarwal VANEET@PURDUE.EDU

Editor: Animashree Anandkumar

Abstract

When the data is distributed across multiple servers, lowering the communication cost between the servers (or workers) while solving the distributed learning problem is an important problem and is the focus of this paper. In particular, we propose a fast, and communication-efficient decentralized framework to solve the distributed machine learning (DML) problem. The proposed algorithm, Group Alternating Direction Method of Multipliers (GADMM) is based on the Alternating Direction Method of Multipliers (ADMM) framework. The key novelty in GADMM is that it solves the problem in a decentralized topology where at most half of the workers are competing for the limited communication resources at any given time. Moreover, each worker exchanges the locally trained model only with two neighboring workers, thereby training a global model with a lower amount of communication overhead in each exchange. We prove that GADMM converges to the optimal solution for convex loss functions, and numerically show that it converges faster and more communication-efficient than the state-of-the-art communication-efficient algorithms such as the Lazily Aggregated Gradient (LAG) and dual averaging, in linear and logistic regression tasks on synthetic and real datasets. Furthermore, we propose Dynamic GADMM (D-GADMM), a variant of GADMM, and prove its convergence under the time-varying network topology of the workers.

1. Introduction

Distributed optimization plays a pivotal role in distributed machine learning applications (Ahmed et al., 2013; Dean et al., 2012; Li et al., 2013, 2014) that commonly aims to minimize $\frac{1}{N}\sum_{n=1}^{N} f_n(\boldsymbol{\Theta})$ with N workers. As illustrated in Fig. 1-(a), this problem is often solved by locally minimizing $f_n(\boldsymbol{\theta}_n)$ at each worker and globally averaging their model parameters $\boldsymbol{\theta}_n$'s (and/or gradients) at a parameter server, thereby yielding the global model parameters $\boldsymbol{\Theta}$ (Tsianos et al., 2012). Another way is to formulate the problem as an average consensus problem that minimizes $\frac{1}{N}\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n)$ under the constraint $\boldsymbol{\theta}_n = \boldsymbol{\Theta}, \forall n$ which can be solved using dual decomposition or Alternating Direction Method of Multipliers (ADMM). ADMM is preferable since standard dual decomposition may fail in updating the variables in some cases. For example, if the objective function $f_n(\boldsymbol{\theta}_n)$ is a nonzero affine function of any component in the input parameter $\boldsymbol{\theta}_n$, then the $\boldsymbol{\theta}_n$-update fails, since the

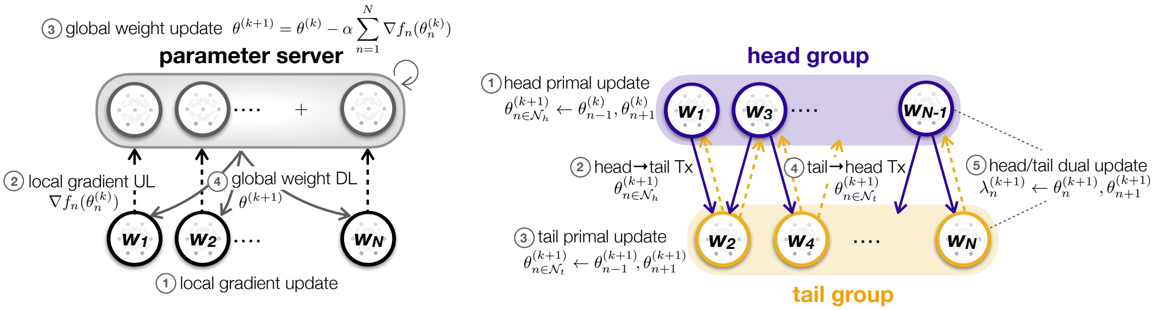
Lagrangian is unbounded from below in $\boldsymbol{\theta}_n$ for most choices of the dual variables (Boyd et al., 2011). However, using ADMM or dual decomposition, an existence of a central entity is necessary.

Such a centralized solution is, however, not capable of addressing a large network size exceeds the parameter server's coverage range. Even if the parameter server has a link to each worker, communication resources may become the bottleneck since, at every iteration, all workers need to transmit their updated models to the server before the server updates the global model and send it to the workers. Hence, as the number of workers increases, the uplink communication resources become the bottleneck. Because of this, we aim to develop a fast and communication-efficient decentralized algorithm, and propose *Group Alternating Direction Method of Multipliers (GADMM).* GADMM solves the problem $\frac{1}{N}\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n)$ subject to $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1}, \forall n \in \{1, \cdots, N-1\}$, in which the workers are divided into two groups (*head* and *Tail*), and each worker in the head (tail) group communicates only with its two neighboring workers from the tail (head) group as shown in Fig. 1-(b). Due to its communication with only two neighbors rather than all the neighbors or a central entity, the communication in each iteration is significantly reduced. Moreover, by dividing the workers into two equal groups, at most half of the workers are competing for the communication resources at every communication round.

Despite this sparse communication where each worker communicates with at most two neighbors, we prove that GADMM converges to the optimal solution for convex functions. We numerically show that its communication overhead is lower than that of state-of-the-art communication-efficient centralized and decentralized algorithms including Lazily Aggregated Gradient (LAG) (Chen et al., 2018), and dual averaging (Duchi et al., 2011) for linear and logistic regression on synthetic and real datasets. Furthermore, we propose a variant of GADMM, Dynamic GADMM (D-GADMM), to consider the dynamic networks in which the workers are moving objects (*e.g.,* vehicles), so the neighbors of each worker could change over time. Moreover, we prove that D-GADMM inherits the same convergence guarantees of GADMM. Interestingly, we show that D-GADMM not only adjusts to dynamic networks, but it also improves the convergence speed of GADMM, *i.e.,* given a static physical topology, keeping on randomly changing the way the connectivity chain is constructed (Fig. 1-(b)) can significantly accelerate the convergence of GADMM. It is worth mentioning that it was shown in (Nedić et al., 2018) as the number of links in the network graph decreases, the convergence speed becomes slower. However, we show that the decrease of the convergence speed of GADMM compared to the standard parameter server-based ADMM (fully connected graph) due to sparsifying the network graph can be compensated by continuously keep changing neighbors and utilize D-GADMM.

2. Related Works and Contributions

Distributed Optimization. There are a variety of distributed optimization algorithms proposed in the literature, such as primal methods (Jakovetić et al., 2014; Nedić and Olshevsky, 2014; Nedić and Ozdaglar, 2009; Shi et al., 2015) and primal-dual methods (Chang et al., 2014a; Koppel et al., 2017; Bedi et al., 2019). Consensus optimization underlies most of the primal methods, while dual decomposition and ADMM are the most popular among the primal-dual algorithms (Glowinski and Marroco, 1975; Gabay and


③ global weight update $\theta^{(k+1)} = \theta^{(k)} - \alpha \sum_{n=1}^{N} \nabla f_n(\theta_n^{(k)})$
parameter server
....
+
② local gradient UL
$\nabla f_n(\theta_n^{(k)})$
④ global weight DL
$\theta^{(k+1)}$
W1
W2
....
WN
① local gradient update
head group
① head primal update
$\theta_{n\in\mathcal{N}_h}^{(k+1)} \leftarrow \theta_{n-1}^{(k)}, \theta_{n+1}^{(k)}$
W1
W3
....
WN-1
② head→tail Tx
$\theta_{n\in\mathcal{N}_h}^{(k+1)}$
④ tail→head Tx
$\theta_{n\in\mathcal{N}_t}^{(k+1)}$
⑤ head/tail dual update
$\lambda_n^{(k+1)} \leftarrow \theta_n^{(k+1)}, \theta_{n+1}^{(k+1)}$
③ tail primal update
$\theta_{n\in\mathcal{N}_t}^{(k+1)} \leftarrow \theta_{n-1}^{(k+1)}, \theta_{n+1}^{(k+1)}$
W2
W4
....
WN
tail group

(a) **Distributed gradient descent** (centralized)

(b) **GADMM** (decentralized)

Figure 1: An illustration of (a) distributed gradient descent with a parameter server and (b) GADMM without any central entity.

Mercier, 1975; Boyd et al., 2011; Jaggi et al., 2014; Ma et al., 2017; Deng et al., 2017). The performance of distributed optimization algorithms is commonly characterized by their computation time and communication cost. The computation time is determined by the per-iteration complexity of the algorithm. The communication cost is determined by: (i) the number of *communication rounds* until convergence, (ii) the number of *channel uses* per communication round, and (iii) the *bandwidth/power* usage per channel use. Note that the number of communication rounds is proportional to the number of iterations; *e.g.,* 2 rounds at every iteration k, for uplink and downlink transmissions in Fig. 1-(a) or for head-to-tail and tail-to-head transmissions in Fig. 1-(b). For a large scale network, the communication cost often becomes dominant compared to the computation time, calling for communication efficient distributed optimization (Zhang et al., 2012; McMahan et al., 2017; Park et al., 2019; Jordan et al., 2018; Liu et al., 2019; Sriranga et al., 2019).

Communication Efficient Distributed Optimization. A vast amount of work is devoted to reducing the aforementioned three communication cost components. To reduce the bandwidth/power usage per channel use, decreasing communication payload sizes is one popular solution, which is enabled by gradient quantization (Suresh et al., 2017), model parameter quantization (Zhu et al., 2016; Sriranga et al., 2019), and model output exchange for large-sized models via knowledge distillation (Jeong et al., 2018). To reduce the number of channel uses per communication round, exchanging model updates can be restricted only to the workers whose computation delays are less than a target threshold (Wang et al., 2018), or to the workers whose updates are sufficiently changed from the preceding updates, with respect to gradients (Chen et al., 2018), or model parameters (Liu et al., 2019). Albeit their improvement in communication efficiency for every iteration k, most of the algorithms in this literature are based on distributed gradient descent, and this limits their required communication rounds to the convergence rate of distributed gradient descent, which is $\mathcal{O}(1/k)$ for differentiable and smooth objective functions and can be as low as $\mathcal{O}(1/\sqrt{k})$ (*e.g.,* when the objective function is non-differentiable everywhere (Boyd et al., 2011)).

On the other hand, primal-dual decomposition methods are shown to be effective in enabling distributed optimization (Jaggi et al., 2014; Boyd et al., 2011; Ma et al., 2017; Glowinski and Marroco, 1975; Gabay and Mercier, 1975; Deng et al., 2017), among which ADMM is a compelling solution that often provides a fast convergence rate with low complexity (Glowinski and Marroco, 1975; Gabay and Mercier, 1975; Deng et al., 2017). It was shown in (Chen et al., 2016) that Gauss-Seidel ADMM (Glowinski and Marroco, 1975) achieves the convergence rate $o(1/k)$. However, this convergence rate is ensured only when the objective function is a sum of two separable convex functions.

Finally, all aforementioned distributed algorithms require a parameter server being connected to every worker, which may induce a costly communication link to some workers or it may not even be feasible particularly for the workers located beyond the server's coverage. In sharp contrast, we aim at developing a decentralized optimization framework ensuring fast convergence without any central entity.

Decentralized Optimization. For decentralized topology, decentralized gradient descent (DGD) has been investigated in (Nedić et al., 2018). Since DGD encounters a lower number of connection per worker compared to parameter-server based GD, it achieves a slower convergence. Beyond GD based approaches, several communication-efficient decentralized algorithms were proposed for both time-variant and invariant topologies. (Duchi et al., 2011; Scaman et al., 2018) proposed decentralized algorithms to solve the problem for time-invariant topology at a convergence rate of $\mathcal{O}(1/\sqrt{k})$. On the other hand, (Lan et al., 2017) proposed a decentralized algorithm that enforces each worker to transmit the updated primal and dual variables at each iteration. Note that, in GADMM, each worker is required to share the primal parameters only per iteration. Finally, it is worth mentioning that a decentralized algorithm was proposed in (He et al., 2018), but that algorithm was studied only for linear learning tasks.

For time-varying topology, there are a few proposed algorithms in the literature. For instance, (Nedić and Olshevsky, 2014) proposed a sub-gradient based algorithm for time-variant directed graph. The algorithm enforces each worker to send two sets of variables to its neighboring nodes per iteration and achieves $\mathcal{O}(1/\sqrt{k})$ convergence rate. In contrast to that, in D-GADMM, only primal variables are shared with neighbors at each iteration. Finally, (Nedic et al., 2017) proposed an algorithm that achieves a linear convergence speed but for strongly convex functions only. Moreover, it also enforces each worker to send more than one set of variables per communication round.

Contribution. We formulate the decentralized machine learning (DML) problem as a constrained optimization problem that can be solved in a decentralized way. Moreover, we propose a novel algorithm to solve the formulated problem optimally for convex functions. The proposed algorithm is shown to be fast and communication-efficient. It achieves significantly less communication overhead compared to the standard ADMM. The proposed GADMM algorithm allows (i) only half of the workers to transmit their updated parameters at each communication round, (ii) the workers update their model parameters in parallel, while each worker communicates only with two neighbors which makes it communication-efficient. Moreover, we propose D-GADMM which has two advantages: (i) it accounts for time-varying network topology, (ii) it improves the convergence speed of GADMM by randomly changing neighbors even when the physical topology is not time-varying. Therefore, D-GADMM integrates the communication efficiency of GADMM which uses only two links per worker (sparse graph) with the fast convergence speed of the standard ADMM with parameter server (star topology with N connection to a central entity). It is worth mentioning that GADMM is closely related to other group-based ADMM methods as in (Wang et al., 2017), but these methods consider more communication links per iteration than our proposed

GADMM algorithm. Notably, the algorithm in (Wang et al., 2017) still relies on multiple central entities, *i.e.,* master workers under a master-slave architecture, whereas GADMM requires no central entity wherein workers are equally divided into head and tail groups.

The rest of the paper is organized as follows. In section 3, we describe the problem formulation. We describe our proposed variant of ADMM (GADMM) and analyze its convergence guarantees in sections 4 and 5 respectively. In section 6, we describe D-GADMM which is an extension of our proposed algorithm to time varying networks. In section 7, we discuss our simulation results comparing GADMM to the considered baselines. Finally, in section 8, we conclude the paper and briefly discuss future directions.

3. Problem Formulation

We consider a network of N workers where each worker is equipped with the task to learn a global parameter $\mathbf{\Theta}$. The aim is to minimize the global convex loss function $F(\mathbf{\Theta})$ which is sum of the local convex, proper, and closed functions $f_n(\mathbf{\Theta})$ for all n. We consider the following optimization problem

$$\min_{\mathbf{\Theta}} F(\mathbf{\Theta}), \quad F(\mathbf{\Theta}) := \sum_{n=1}^{N} f_n(\mathbf{\Theta}), \tag{1}$$

where $\mathbf{\Theta} \in \mathbb{R}^d$ is the global model parameter. Gradient descent algorithm can be used to solve the problem in (1) iterativly in a central entity. The goal here is to solve the problem in a distributed manner. The standard technique used in the literature for distributed solution is consensus formulation of (1) given by.

$$\min_{\mathbf{\Theta}, \{\boldsymbol{\theta}_n\}_{n=1}^{N}} \quad \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n) \tag{2}$$

$$\text{s.t.} \quad \boldsymbol{\theta}_n = \mathbf{\Theta}, \quad \forall\, n. \tag{3}$$

Note that with the reformulation in (2)-(3), the objective function becomes separable across the workers and hence can be solved in a distributed manner. The problem in (2)-(3) is known as the *global consensus problem* since the constraint forces all the variables across different workers to be equal as detailed in (Boyd et al., 2011). The problem in (2)-(3) can be solved using the primal-dual based algorithms as in (Chang et al., 2014b; Touri and Nedic, 2009; Nedić and Ozdaglar, 2009), saddle point algorithms proposed in (Koppel et al., 2017; Bedi et al., 2019), and ADMM-based techniques such as (Glowinski and Marroco, 1975; Boyd et al., 2011; Deng et al., 2017). ADMM forms an augmented Lagrangian which adds a quadratic term to the Lagrange function and breaks the main problem into sub-problems that are easier to solve per iteration. Note that in the ADMM implementation (Boyd et al., 2011; Deng et al., 2017), only the primal variables $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$ can be updated in a distributed manner. However, the step of updating $\mathbf{\Theta}$ requires collecting $\boldsymbol{\theta}_n$ from all workers which is communication inefficient (Boyd et al., 2011).

The problem formulation in (2)-(3) can be solved using standard ADMM (parameter server based-ADMM). The augmented Lagrangian of the optimization problem in (2)-(3) as

$$\boldsymbol{\mathcal{L}}_\rho(\boldsymbol{\Theta}, \{\boldsymbol{\theta}_n\}_{n=1}^N, \boldsymbol{\lambda}) = \sum_{n=1}^N f_n(\boldsymbol{\theta}_n) + \sum_{n=1}^N \langle \boldsymbol{\lambda}_n, \boldsymbol{\theta}_n - \boldsymbol{\Theta} \rangle + \frac{\rho}{2} \sum_{n=1}^N \|\boldsymbol{\theta}_n - \boldsymbol{\Theta}\|^2, \tag{4}$$

where $\boldsymbol{\lambda} := [\boldsymbol{\lambda}_1^T, \cdots, \boldsymbol{\lambda}_N^T]^T$ is the collection of the dual variables, and ρ is a constant adjusting the penalty for the disagreement between $\boldsymbol{\theta}_n$ and $\boldsymbol{\Theta}$. The primal and dual variables under ADMM are updated in the following three steps.

1) At iteration $k+1$, the *primal variable of each workers* is updated as:

$$\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\Theta}^k \rangle + \frac{\rho}{2} \|\boldsymbol{\theta}_n - \boldsymbol{\Theta}^k\|^2 \big], n \in \{1, \cdots, N\} \tag{5}$$

2) After the update in (5), each workers sends its primal variable (updated model) to the parameter server. The *primal variable of the parameter server* is then updated as:

$$\boldsymbol{\Theta}^{k+1} = \frac{1}{N} \sum_{n=1}^N \Big(\boldsymbol{\theta}_n^{k+1} + \frac{1}{\rho} \boldsymbol{\lambda}_n^k \Big). \tag{6}$$

3) After the update in (6), the parameter server broadcasts its primal variable (the updated global model) to all workers. After receiving the global model ($\boldsymbol{\Theta}^{k+1}$) from the parameter server, *each worker locally updates its dual variable* $\boldsymbol{\lambda}_n$ as follows

$$\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\Theta}^{k+1}), n = \{1, \cdots, N\}. \tag{7}$$

Note that standard ADMM requires a parameter server that collects updates from all workers, update a global model and broadcast that model to all workers. Such a scheme may not be a communication-efficient due to: (i) N workers competing for the limited communication resources at every iteration, (ii) the worker with the weakest communication channel will be the bottleneck for the communication rate of the broadcast channel from the parameter server to the workers, (iii) some workers may not be in the coverage zone of the parameter server.

In contrast to standard ADMM, we propose a decentralized algorithm that minimizes the communication cost required per worker by allowing only $N/2$ workers to transmit at every communication round, so the communication resources to each worker are doubled compared to parameter server-based ADMM. Moreover, it limits the communication of each worker to include only two neighbors. We consider the optimization problem in (2)-(3) and rewrite the constraints as follows.

$$\boldsymbol{\theta}^\star := \arg \min_{\{\boldsymbol{\theta}_n\}_{n=1}^N} \sum_{n=1}^N f_n(\boldsymbol{\theta}_n) \tag{8}$$

$$\text{s.t. } \boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1}, \quad n = 1, \cdots, N-1. \tag{9}$$

Here $\boldsymbol{\theta}^\star$ is the optimal and note that $\boldsymbol{\theta}_{n-1}^\star = \boldsymbol{\theta}_n^\star$ and $\boldsymbol{\theta}_n^\star = \boldsymbol{\theta}_{n+1}^\star$ for all n. This implies that each worker n has joint constraints with only two neighbors (except for the two end workers which have only one). Nonetheless, ensuring $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1}$ for all $n \in \{1, \cdots, N-1\}$ at the convergence point yields convergence to a global model parameter that is shared across all workers.

4. Proposed Algorithm: GADMM

We will now describe our proposed algorithm, GADMM, that solves the optimization problem defined in (8)-(9) in a decentralized manner. The proposed algorithm is fast since it allows workers belonging to the same group to update their model parameters in parallel, and it is communication-efficient since it allows workers to exchange variables with a minimum number of neighbors and enjoys a fast convergence rate. Moreover, it allows only half of the workers to transmit their updated model parameters at each communication round. Note that when the number of workers who update their parameters per communication round is reduced to half, the communication physical resources (*e.g.,* bandwidth) available to each worker are doubled when those resources are shared among workers.

The **main idea** of the proposed algorithm is presented in Fig. 1-(b). The proposed GADMM algorithm splits the network nodes (workers) connected with a chain into two groups *head* and *tail* such that each worker in the head's group is connected to other workers through two tail workers. It allows updating the parameters in parallel for the workers in the same group. In one algorithm iterate, the workers in the head group update their model parameters, and each head worker transmits its updated model to its directly connected tail neighbors. Then, tail workers update their model parameters to complete one iteration. In doing so, each worker (except the edge workers) communicates with only two neighbors to update its parameter, as depicted in Fig. 1-(b). Moreover, at any communication round, only half of the workers transmit their parameters, and these parameters are transmitted to only two neighbors.

In contrast to the Gauss-Seidel ADMM in (Boyd et al., 2011), GADMM allows all the head (tail) workers to update their parameters in parallel and still converges to the optimal solution for convex functions as will be shown later in this paper. Moreover, GADMM has much less communication overhead as compared to PJADMM in (Deng et al., 2017) which requires all workers to send their parameters to a central entity at every communication round. Also, GADMM has fewer hyperparameters to control and less computation per iteration than PJADMM. The detailed steps of the proposed algorithm are summarized in Algorithm 1.

To intuitively describe GADMM, without loss of generality, we consider an even N number of workers under their linear connectivity graph shown in Fig. 1-(b), wherein each head (or tail) worker communicates at most with two neighboring tail (or head) workers, except for the edge workers (*i.e.,* first and last workers). With that in mind, we start by writing the augmented Lagrangian of the optimization problem in (8)-(9) as

$$\boldsymbol{\mathcal{L}}_\rho(\{\boldsymbol{\theta}_n\}_{n=1}^N, \boldsymbol{\lambda}) = \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n) + \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1} \rangle + \frac{\rho}{2} \sum_{n=1}^{N-1} \|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}\|^2, \tag{10}$$

Let's divide the N workers into two groups, head $\mathcal{N}_h = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \boldsymbol{\theta}_5, \cdots, \boldsymbol{\theta}_{N-1}\}$, and tail $\mathcal{N}_t = \{\boldsymbol{\theta}_2, \boldsymbol{\theta}_4, \boldsymbol{\theta}_6, \cdots, \boldsymbol{\theta}_N\}$, respectively. The primal and dual variables under GADMM are updated in the following three steps.

1) At iteration $k+1$, the *primal variables of head workers* are updated as:

$$\begin{aligned}\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \boldsymbol{\theta}_{n-1}^k - \boldsymbol{\theta}_n\rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^k\rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_{n-1}^k - \boldsymbol{\theta}_n\|^2 \\ + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^k\|^2\big], n \in \mathcal{N}_h \setminus \{1\} \qquad (11)\end{aligned}$$

Since the first head worker ($n = 1$) does not have a left neighbor ($\boldsymbol{\theta}_{n-1}$ is not defined), its model is updated as follows.

$$\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^k\rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^k\|^2\big], n = 1 \qquad (12)$$

2) After the updates in (11) and (12), head workers send their updates to their two tail neighbors. The *primal variables of tail workers* are then updated as:

$$\begin{aligned}\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n\rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^{k+1}\rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n\|^2 \\ + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}^{k+1}\|^2\big], n \in \mathcal{N}_t \setminus \{N\}. \qquad (13)\end{aligned}$$

Since the last tail worker ($n = N$) does not have a right neighbor ($\boldsymbol{\theta}_{n+1}$ is not defined), its model is updated as follows.

$$\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n-1}^k, \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n\rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n\|^2\big], n = N. \qquad (14)$$

3) After receiving the updates from neighbors, *every worker locally updates its dual variables* $\boldsymbol{\lambda}_{n-1}$ and $\boldsymbol{\lambda}_n$ as follows

$$\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1}), n = \{1, \cdots, N-1\}. \qquad (15)$$

These three steps of GADMM are summarized in Algorithm 1. We remark that when $f_n(\boldsymbol{\theta}_n)$ is convex, proper, closed, and differentiable for all n, the subproblems in (11) and (13) are convex and differentiable with respect to $\boldsymbol{\theta}_n$. That is true since the additive terms in the augmented Lagrangian are the addition of quadratic and linear terms, which are also convex and differentiable.

5. Convergence Analysis

In this section, we focus on the convergence analysis of the proposed algorithm. It is essential to prove that the proposed algorithm indeed converges to the optimal solution of the problem in (8)-(9) for convex, proper, and closed objective functions. The idea to prove the convergence is related to the proof of Gauss-Seidel ADMM in (Boyd et al., 2011), while additionally accounting for the following three challenges: (i) the additional terms

Algorithm 1 Group ADMM (GADMM)

```
**Input**: $N, f_n(\boldsymbol{\theta}_n)$ for all $n, \rho$
**Initialization**:
$\mathcal{N}_h = \{\boldsymbol{\theta}_n \mid n\text{: odd}\}, \mathcal{N}_t = \{\boldsymbol{\theta}_n \mid n\text{: even}\}$
$\boldsymbol{\theta}_n^{(0)} = 0, \boldsymbol{\lambda}_n^{(0)} = 0$ for all $n$
**for** $k = 0, 1, 2, \cdots, K$ **do**
    **Head worker** $n \in \mathcal{N}_h$**:**
        **computes** its primal variable $\boldsymbol{\theta}_n^{k+1}$ via (11) in parallel; and
        **sends** $\boldsymbol{\theta}_n^{k+1}$ to its neighboring workers $n-1$ and $n+1$.
    **Tail worker** $n \in \mathcal{N}_t$**:**
        **computes** its primal variable $\boldsymbol{\theta}_n^{k+1}$ via (13) in parallel; and
        **sends** $\boldsymbol{\theta}_n^{k+1}$ to its neighbor workers $n-1$ and $n+1$.
    **Every worker updates** the dual variables $\boldsymbol{\lambda}_{n-1}^k$ and $\boldsymbol{\lambda}_n^k$ via (15) locally.
**end for**
```

that appear when the problem is a sum of more than two separable functions, (ii) the fact that each worker can communicate with two neighbors only, and (iii) the parallel model parameter updates of the head (tail) workers. We show that the GADMM iterates converge to the optimal solution after addressing all the above-mentioned challenges in the proof. Before presenting the main technical Lemmas and Theorems, we start with the necessary and sufficient optimality conditions, which are the primal and the dual feasibility conditions (Boyd et al., 2011) defined as

$$\boldsymbol{\theta}_n^\star = \boldsymbol{\theta}_{n-1}^\star, n \in \{2, \cdots, N\} \qquad \text{(primal feasibility)} \tag{16}$$

$$\begin{aligned} &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) - \boldsymbol{\lambda}_{n-1}^\star + \boldsymbol{\lambda}_n^\star, n \in \{2, \cdots, N-1\} \\ &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) + \boldsymbol{\lambda}_n^\star, n = 1 \qquad \text{(dual feasibility)} \\ &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) + \boldsymbol{\lambda}_{n-1}^\star, n = N \end{aligned} \tag{17}$$

We remark that the optimal values $\boldsymbol{\theta}_n^\star$ are equal for each n, we denote $\boldsymbol{\theta}^\star = \boldsymbol{\theta}_n^\star = \boldsymbol{\theta}_{n-1}^\star$ for all n. Note that, at iteration $k+1$, we calculate $\boldsymbol{\theta}_n^{k+1}$ for all $n \in \mathcal{N}_t \setminus \{N\}$ as in (13), from the first order optimality condition, it holds that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n-1}^{k+1}) + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1}). \tag{18}$$

Next, rewrite the equation in (18) as

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \left(\boldsymbol{\lambda}_{n-1}^k + \rho(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n^{k+1})\right) + \left(\boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1})\right). \tag{19}$$

From the update in (15), the equation in (19) implies that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, n \in \mathcal{N}_t \setminus \{N\}. \tag{20}$$

Note that for the N-th worker, We calculate $\boldsymbol{\theta}_N^{k+1}$ as in (14), then we follow the same steps, and we get

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1}, n = N. \tag{21}$$

From the result in (20) and (21), it holds that the dual feasibility condition in (17) is always satisfied for all $n \in \mathcal{N}_t$.

Next, consider every $\boldsymbol{\theta}_n^{k+1}$ such that $n \in \mathcal{N}_h \setminus \{1\}$ which is calculated as in (11) at iteration k. Similarly from the first order optimality condition, we can write

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n-1}^k) + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^k). \tag{22}$$

Note that in (22), we don't have all the primal variables calculated at $k+1$ instance. Hence, we add and subtract the terms $\boldsymbol{\theta}_{n-1}^{k+1}$ and $\boldsymbol{\theta}_{n+1}^{k+1}$ in (22) to get

$$\begin{aligned}\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) &- \left(\boldsymbol{\lambda}_{n-1}^k + \rho(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_n^{k+1})\right) + \Big(\boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1})\Big)\\ &+ \rho(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^k) + \rho(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k).\end{aligned} \tag{23}$$

From the update in (15), it holds that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \rho(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^k) + \rho(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k). \tag{24}$$

Following the same steps for the first head worker ($n = 1$) after excluding the terms $\boldsymbol{\lambda}_{n-1}^k$ and $\rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n-1}^k)$ from (22) (worker 1 does not have a left neighbor) gives

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) + \boldsymbol{\lambda}_n^{k+1} + \rho(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k). \tag{25}$$

Let $\mathbf{s}_{n\in\mathcal{N}_h}^{k+1}$, the dual residual of worker $n \in \mathcal{N}_h$ at iteration $k+1$, be defined as follows

$$\mathbf{s}_n^{k+1} = \begin{cases} \rho(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^k) + \rho(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k), \text{ for } n \in \mathcal{N}_h \setminus \{1\} \\ \rho(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^k), \text{ for } n = 1. \end{cases} \tag{26}$$

Next, we discuss about the primal feasibility condition in (16) at iteration $k+1$. Let $\mathbf{r}_{n,n+1}^{k+1} = \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n+1}^{k+1}$ be the primal residual of each worker $n \in \{1, \cdots, N-1\}$. To show the convergence of GADMM, we need to prove that the conditions in (16)-(17) are satisfied for each worker n. We have already shown that the dual feasibility condition in (17) is always satisfied for the tail workers, and the dual residual of tail workers is always zero. Therefore, to prove the convergence and the optimality of GADMM, we need to show that the $\mathbf{r}_{n,n+1}^k$ for all $n = 1, \cdots, N-1$ and $\mathbf{s}_{n\in\mathcal{N}_h}^k$ converge to zero, and $\sum_{n=1}^N f_n(\boldsymbol{\theta}_n^k)$ converges to $\sum_{n=1}^N f_n(\boldsymbol{\theta}^\star)$ as $k \to \infty$. Now we are in position to introduce our first result in terms of Lemma 1.

Lemma 1 *For the iterates $\boldsymbol{\theta}_n^{k+1}$ generated by Algorithm 1, we have*

(i) Upper bound on the optimality gap

$$\sum_{n=1}^N [f_n(\boldsymbol{\theta}_n^{k+1}) - f_n(\boldsymbol{\theta}^\star)] \leq -\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1}, \mathbf{r}_{n,n+1}^{k+1} \rangle + \sum_{n\in\mathcal{N}_h} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^\star - \boldsymbol{\theta}_n^{k+1} \rangle. \tag{27}$$

(ii) Lower bound on the optimality gap

$$\sum_{n=1}^N [f_n(\boldsymbol{\theta}_n^{k+1}) - f_n(\boldsymbol{\theta}^\star)] \geq -\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n+1}^{k+1} \rangle. \tag{28}$$

The detailed proof is provided in Appendix A. The main idea for the proof is to utilize the optimality of the updates in (11) and (13). We derive the upper bound for the objective function optimality gap in terms of the primal and dual residuals as stated in (27). To get the lower bound in (28) in terms of the primal residual, the definition of the Lagrangian is used at $\rho = 0$. The result in Lemma 1 is used to derive the main results in Theorem 2 of this paper presented next.

Theorem 2 *When $f_n(\boldsymbol{\theta}_n)$ is closed, proper, and convex for all n, and the Lagrangian $\boldsymbol{\mathcal{L}}_0$ has a saddle point, for GADMM iterates, it holds that*

(i) the primal residual converges to zero as $k \to \infty$.i.e.,

$$\lim_{k\to\infty} \mathbf{r}_{n,n+1}^k = \mathbf{0}, n \in \{1, \cdots, N-1\}. \tag{29}$$

(ii) the dual residual converges to zero as $k \to \infty$.i.e.,

$$\lim_{k\to\infty} \mathbf{s}_n^k = \mathbf{0}, n \in \mathcal{N}_h. \tag{30}$$

(iii) the optimality gap converges to zero as $k \to \infty$.i.e.,

$$\lim_{k\to\infty} \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^k) = \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star). \tag{31}$$

Proof The detailed proof of Theorem 2 is provided in Appendix B. There are three main steps to prove convergence of the proposed algorithm. For a proper, closed, and convex objective function $f_n(\cdot)$, with Lagrangian $\mathcal{L}_0$ which has a saddle point $(\boldsymbol{\theta}^\star, \{\boldsymbol{\lambda}_n\}_{\forall n})$, we define a Lyapunov function V_k as

$$V_k = 1/\rho \sum_{n=1}^{N-1} \left\| \boldsymbol{\lambda}_n^k - \boldsymbol{\lambda}_n^\star \right\|^2 + \rho \sum_{n \in \mathcal{N}_h \setminus \{1\}} \left\| \boldsymbol{\theta}_{n-1}^k - \boldsymbol{\theta}^\star \right\|^2 + \rho \sum_{n \in \mathcal{N}_h} \left\| \boldsymbol{\theta}_{n+1}^k - \boldsymbol{\theta}^\star \right\|^2. \tag{32}$$

In the proof, we show that V_k is monotonically decreasing at each iteration k of the proposed algorithm. This property is then used to prove that the primal residuals go to zero as $k \to \infty$ which implies that $\mathbf{r}_{n,n+1}^k \to \mathbf{0}$ for all n. Secondly, we prove that the dual residuals converges to zero as $k \to \infty$ which implies that $\mathbf{s}_n^k \to \mathbf{0}$ for all $n \in \mathcal{N}_h$. Note that the convergence in the first and the second step implies that the overall constraint violation due to the proposed algorithm goes to zero as $k \to \infty$. In the final step, we utilize statement (i) and (ii) of Theorem 2 into the results of Lemma 1 to prove that the objective optimality gap goes to zero as $k \to \infty$. ■

Algorithm 2 Dynamic GADMM (D-GADMM)

```
Input: $N, f_n(\cdot)$ for all $n$, $\rho$, and $\tau$
Initialization:
$\mathcal{N}_h = \{\boldsymbol{\theta}_n \mid n\text{: odd}\}, \mathcal{N}_t = \{\boldsymbol{\theta}_n \mid n\text{: even}\}$
$\boldsymbol{\theta}_n^{(0)} = 0, \boldsymbol{\lambda}_n^{(0)} = 0$, for all $n$
for $k = 0, 1, 2, \cdots, K$ do
    if $k$ mod $\tau = 0$ then
        Every worker:
            broadcasts current model parameter
            finds neighbors and refreshes indices $\{n\}$ as explained in Appendix C.
            sends $\boldsymbol{\lambda}_n^k$ to its right neighbor (worker $n_{r,k}$)
    end if
    Head worker $n \in \mathcal{N}_h$:
        computes its primal variable $\boldsymbol{\theta}_n^{k+1}$ via (11) in parallel; and
        sends $\boldsymbol{\theta}_n^{k+1}$ to its neighboring workers $n_{l,k}$ and $n_{r,k}$.
    Tail worker $n \in \mathcal{N}_t$:
        computes its primal variable $\boldsymbol{\theta}_n^{k+1}$ via (13) in parallel; and
        sends $\boldsymbol{\theta}_n^{k+1}$ to its neighbor workers $n_{l,k}$ and $n_{r,k}$.
    Every worker updates the dual variables $\boldsymbol{\lambda}_{n-1}^k$ and $\boldsymbol{\lambda}_n^k$ via (15) locally.
end for
```

6. Extension to Time-Varying Network: D-GADMM

In this section, we present an extension of the proposed GADMM algorithm to the scenario where the set of neighboring workers to each worker is varying over time. Note that the overlay logical topology under consideration is still chain while the physical neighbors are allowed to change. Under this dynamic setting, the execution of the proposed GADMM in Algorithm 1 would be disrupted. Therefore, we propose, D-GADMM (summarized in Algorithm 2) which adjusts to the changes in the set of neighbors.

In D-GADMM, all the workers periodically reconsider their connections after every τ iterations. if neighbors and/or worker assignment to head/tail group change, every worker broadcasts its current model parameter to the new neighbors. We assume that the workers run an algorithm that can keep constructing a communication-efficient logical chain as the underlying physical topology changes, and the design of such an algorithm is not the main focus of the paper. It is worth mentioning that a logical graph that starts at one worker and reaches every other worker only once in the most communication-efficient way is an NP-hard problem. It can be easily shown that this problem can be reduced to a Traveling Salesman Problem (TSP). This is due to the fact that starting from one worker and choosing every next one such that the total communication cost is minimized is exactly equal to starting from one city and reaching every other city such that the total distance is minimized, *i.e.,* the workers in our problem are the cities in TSP, and the communication cost between each pair of workers in our problem is the distance between each pair of cities in TSP. Hence, proposed heuristics to solve TSP (Lenstra and Kan, 1975; Bonomi and Lutton, 1984) can be used to construct the chain in our problem with the aid of a central entity, and then the algorithm continues working on the decentralized way. Decentralized heuristics for TSP have been proposed which can also be used (Peterson, 1990; Dorigo and Gambardella, 1997). However, in this paper, we use a simple decentralized heuristic that we describe in Appendix C. Finally, it is worth mentioning that D-GADMM can still be utilized even if the physical topology does not change. In such a scenario, the workers can agree on a predefined sequence of logical chains, so changing neighbors does not require running an online algorithm, and thus it encounters zero overhead. We observe in section 7 that D-GADMM can improve the convergence speed when it is utilized even when the physical topology does not change.

The detailed steps of D-GADMM is described in Algorithm 2. We note that before the execution, we assume that all the nodes are connected to each other with a chain. Each node is associated with an index n and there exists a link from node 1 to node 2, node 2 to node 3, and so on till $N-1$ to N. For each node n there is an associated primal variable $\boldsymbol{\theta}_n$ from $n=1$ to N and a dual variable $\boldsymbol{\lambda}_n$ from node $n=1$ to $N-1$. Under the dynamic settings, we assume that the nodes at position $n=1$ and N are fixed while the other nodes are allowed to move in the network. This means that instead of having the connection in the order $1-2-3\cdots N$, the nodes are allowed to connect in any order such as $1-5-3-\cdots-4-N$, or $1-4-2-\cdots-5-N$, etc. Alternatively, the neighbors of each node n are no longer fixed under the dynamic settings. To denote this behavior, since the topology is still the chain, we call the left neighbor to node n as $n_{l,k}$ at iteration k and similarly $n_{r,k}$ for the right neighbor node. Therefore, at each iteration k, each node implements the algorithm considering $n_{l,k}$ and $n_{r,k}$ as its neighbors. Note that, when the topology changes at iteration k, every worker n transmits its right dual variable $\boldsymbol{\lambda}_n^k$ to its right neighbor in the new chain to ensure that both neighbors share the same dual variable. Therefore, the right neighbor of each worker n will replace $\boldsymbol{\lambda}_{n_{l,k}}^k$ with the dual variable that is received from its new left neighbor. With that, we show in Appendix D that the algorithm converges to the optimal solution in a similar manner to GADMM.

7. Numerical Results

To validate our theoretical foundations, we numerically evaluate the performance of GADMM in linear and logistic regression tasks, compared with the following benchmark algorithms.

- **LAG-PS** (Chen et al., 2018): A version of LAG where parameter server selects communicating workers.
- **LAG-WK** (Chen et al., 2018): A version of LAG where workers determine when to communicate with the server.
- **Cycle-IAG** (Blatt et al., 2007; Gurbuzbalaban et al., 2017): A cyclic modified version of the incremental aggregated gradient (IAG).
- **R-IAG** (Chen et al., 2018; Schmidt et al., 2017): A non-uniform sampling version of stochastic average gradient (SAG).
- **GD**: Batch gradient descent.
- **DGD** (Nedić et al., 2018) Decentralized gradient descent.

	Linear Regression				**Logistic** Regression			
Iteration	N =14	20	24	26	N =14	20	24	26
LAG-PS	542	8,043	54,249	141,132	21,183	20,038	19,871	20,544
LAG-WK	385	6,444	44,933	121,134	18,584	17,475	17,050	17,477
GADMM	**78**	**292**	**558**	**550**	**120**	**235**	**112**	**160**
GD	524	8,163	55,174	143,651	1,190	1,204	1,181	1,152

	Linear Regression				**Logistic** Regression			
TC	N =14	20	24	26	N =14	20	24	26
LAG-PS	3,183	52,396	363,571	1,035,778	316,570	419,819	495,792	553,493
LAG-WK	**820**	12,369	82,985	241,944	18,786	17,835	17,432	17,915
GADMM	1,092	**5,840**	**13,392**	**14,300**	**696**	**1,962**	**1,030**	**1,712**
GD	7,860	171,423	1,379,350	3,878,577	17,850	25,284	29,525	31,104

Table 1: The required number of iterations (top) and total communication cost (bottom) to achieve the target objective error 10^{-4} for different number of workers, in linear and logistic regression with the real datasets.

- **DualAvg** (Duchi et al., 2011) Dual averaging.

For the tuning parameters, we use the setup in (Chen et al., 2018). For our decentralized algorithm, we consider N workers without any central entity, whereas for centralized algorithms, a uniformly randomly selected worker is considered as a central controller having a direct link to each worker. The performance of each algorithm is measured using:

- the **objective error** $|\sum_{n=1}^{N}\left[f_n(\boldsymbol{\theta}_n^{(k)}) - f_n(\boldsymbol{\theta}^*)\right]|$ at iteration k.
- (ii) The **total communication cost (TC)**. The TC of a decentralized algorithm is $\sum_{t=1}^{T_a}\sum_{n=1}^{N}\mathbf{1}_{n,t}\cdot L_{n,t}^{m}$, where T_a is the number of iterations to achieve a target accuracy a, and $\mathbf{1}_{n,t}$ denotes an indicator function that equals 1 if worker n is sending an update at t, and 0 otherwise. The term $L_{n,t}^{m}$ is the cost of the communication link between workers n and m at communication round t. Next, let $L_{n,t}^{c}$ denote the cost of the communication between worker n and the central controller at t. Then, the TC of a centralized algorithm is $\sum_{t=1}^{T_a}(L_{\mathrm{BC},t}^{c} + \sum_{n=1}^{N}\mathbf{1}_{n,t}\cdot L_{n,t}^{c})$, where $L_{\mathrm{BC},t}^{c}$ and $L_{n,t}^{c}$'s correspond to downlink broadcast and uplink unicast costs, respectively. It is noted that the communication overhead in (Chen et al., 2018) only takes into account uplink costs.
- The total running time (clock time) to achieve objective error a. This metric considers both the communication and the local computation time. We consider $L_{n,t}^{m} = L_{n,t}^{c} = L_{\mathrm{BC},t}^{c} = 1$ unless otherwise specified.

All simulations are conducted using the synthetic and real datasets described in (Dua and Graff, 2017; Chen et al., 2018). The synthetic data for the linear and logistic regression tasks are generated as described in (Chen et al., 2018). We consider 1,200 samples with 50 features, which are evenly split into workers. Next, the real data tests linear and logistic

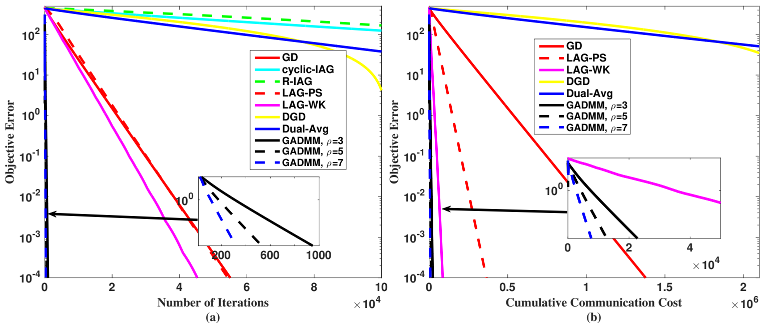


Figure 2: Objective error, total communication cost, and total running time comparison between GADMM and five benchmark algorithms, in *linear* regression with synthetic ($N = 24$) datasets.

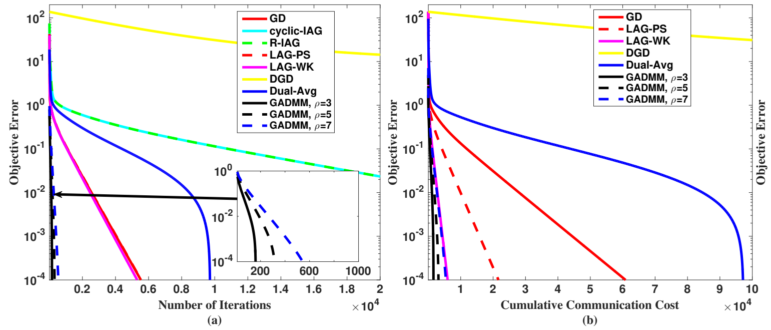


Figure 3: Objective error, total communication cost, and total running time comparison between GADMM and five benchmark algorithms, in *linear* regression with real ($N = 10$) datasets.

regression tasks with **Body Fat** (252 samples, 14 features) and **Derm** (358 samples, 34 features) datasets (Dua and Graff, 2017), respectively. As the real dataset is smaller than the synthetic dataset, we by default consider 10 and 24 workers for the real and synthetic datasets, respectively.

Figs. 2, 3, 4, and 5 corroborate that GADMM outperforms the benchmark algorithms by several orders of magnitudes, thanks to the idea of two alternating groups where each worker communicates only with two neighbors. For linear regression with the synthetic dataset, Fig. 2 shows that all variants of GADMM with $\rho = 3, 5$, and 7 achieve the target objective error of 10^{-4} in less than 1,000 iterations, whereas GD, LAG-PS, and LAG-WK (the closest among baselines) require more than 40,000 iterations to achieve the same target error. Furthermore, the TC of GADMM with $\rho = 3$ and $\rho = 5$ are 6 and 9 times lower than that of LAG-WK respectively. Table 1 shows similar results for different numbers of workers, only except for linear regression with the smallest number of workers (14), in which LAG-WK achieves the lowest TC. We also observe from Figs. 2 and 3 that GADMM outperforms all baselines in terms of the total running time, thanks to the fast convergence. GADMM performs matrix

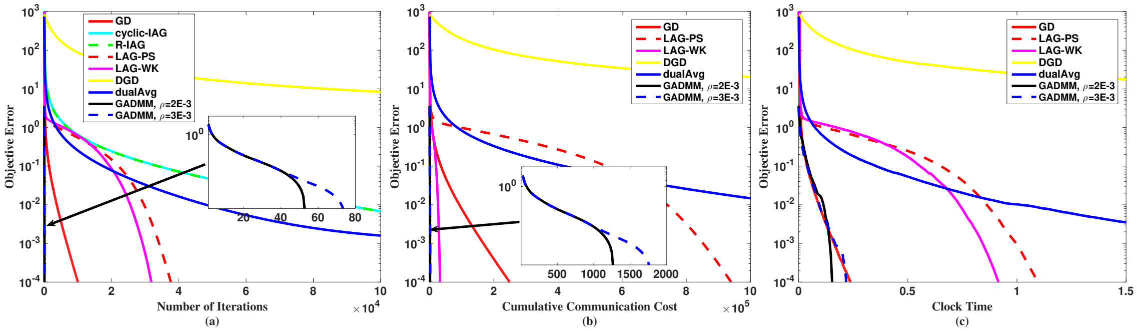


Figure 4: Objective error, total communication cost, and total running time comparison between GADMM and five benchmark algorithms, in *logistic* regression with synthetic ($N = 24$) datasets.

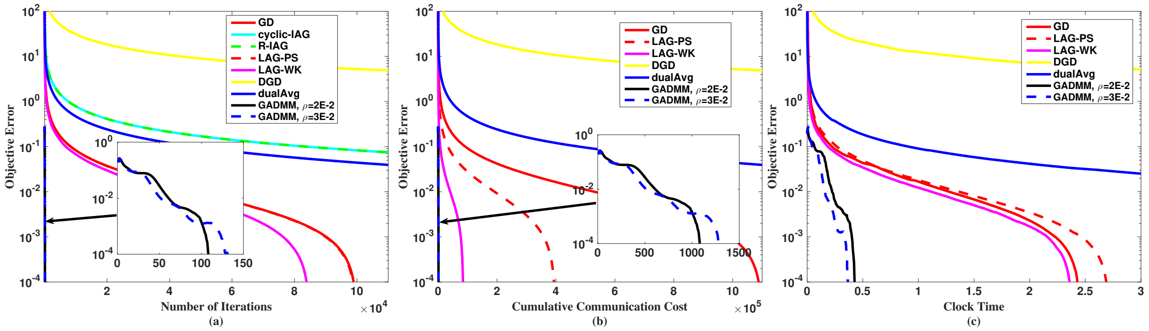
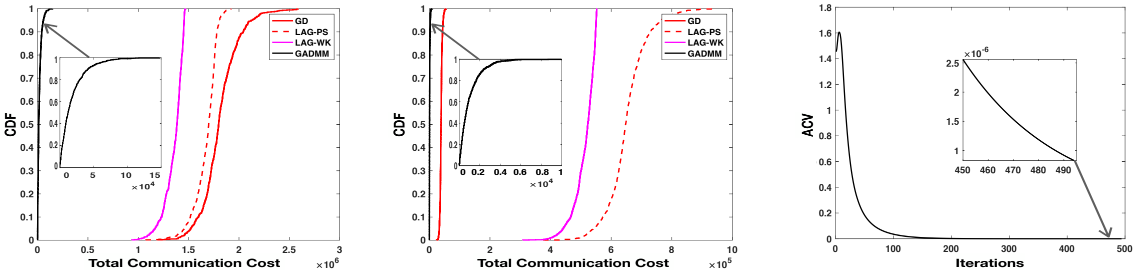


Figure 5: Objective error, total communication cost, and total running time comparison between GADMM and five benchmark algorithms, in *logistic* regression with real ($N = 10$) datasets.

inversion which is computationally complex compared to calculating gradient. However, the computation cost per iteration is compensated by fast convergence.

For logistic regression, Figs. 4 and 5 validate that GADMM outperforms the benchmark algorithms, as in the case of linear regression in Figs. 2 and 3. One thing that is worth mentioning here is shown in Fig 4-(c), where we can see that the total running time of GADMM is equal to the running time of GD. The reason behind this is that the logistic regression problem is not solved in a closed-form expression at each iteration. However, GADMM still significantly outperforms GD in communication-efficiency.

Next, comparing the results in Fig. 2 and Fig. 3, we observe that the optimal ρ depends on the data distribution across workers. Namely, when the local data samples of each worker are highly correlated with the other workers' samples (*i.e.,* Body Fat dataset, Fig. 3), the local optimal of each worker is very close to the global optimal. Therefore, reducing the penalty for the disagreement between θ_n and θ_{n+1} by lowering ρ yields faster convergence. Following the same reasoning, higher ρ provides faster convergence when the local data samples are independent of each other (*i.e.,* synthetic datasets in Fig. 2).



(a) CDF of TC: Linear, real data (Derm)

(b) CDF of TC: Logistic, real data (Derm)

(c) ACV: Logistic, real data (Derm)

Figure 6: The cumulative distribution function (CDF) of total communication cost (TC) in (a) linear and (b) logistic regression by uniformly randomly distributed 24 workers with 1,000 observations, and (c) the average consensus constraint violation (ACV) of GADMM in logistic regression by 4 workers.

Fig. 6-(a) and (b) demonstrate that GADMM is communication efficient under different network topologies. In fact, the TC calculations of GADMM in Table 1 and Fig. 2 rely on a unit communication cost for all communication links, i.e., $L^m_{n,t} = L^c_{n,t} = L^c_{\mathrm{BC},t} = 1$, which may not capture the communication efficiency of GADMM under a generic network topology. Instead, we use the consumed energy per communication iteration as the communication cost metric. We illustrate the cumulative distribution function (CDF) of TC by observing 1,000 different network topologies. At the beginning of each observation, 24 workers are randomly distributed over a 10×10 m^2 square area. In GADMM, the method described in Appendix C is used to construct the logical chain. In centralized algorithms, the worker closest to the center becomes a central worker associating with all the other workers. We assume that the bandwidth is evenly distributed among users, and we also assume that each worker needs a bit rate of 10Mbps to transmit its model in a one-time slot. Therefore, the communication cost per worker per iteration is the amount of energy that worker consumes to achieve the rate of 10Mbps. Note that according to Shannon's formula, the achievable rate is a function of the bandwidth and power, *i.e.,* $R = B \cdot log_2(\frac{P}{d^2 \cdot N_0 \cdot B})$, where B is the bandwidth, P is the communication power, N_0 is the noise spectral density, and d is the distance between the transmitter and the receiver (McKeague, 1981), so we assume a free-space communication link. In our simulations, we assume, $B = 2$MHz, $N_0 = 1E-6$, we find the required power (energy) to achieve 10Mbps over link l at time slot t, and that reflects the communication cost of using link l at time slot t.

The CDF results in Fig. 6-(a) and (b) show that with high probability, GADMM achieves much lower TC in both linear and logistic regression tasks for generic network topology, compared to other baseline algorithms. On the other hand, Fig. 6-(c) validates that GADMM guarantees consensus on the model parameters of all workers when training converges. Indeed, GADMM complies with the constraint $\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1}$ in (3). We observe in Fig. 4-(c) that the average consensus constraint violation (ACV), defined as $\sum_{n=1}^{N-1} |\boldsymbol{\theta}_n^{(k)} - \boldsymbol{\theta}_{n+1}^{(k)}|/N$, goes to zero with the number of iterations. Specifically, AVC becomes 8×10^{-7} after 495 iterations at which the loss becomes 1×10^{-4}. This underpins that GADMM is robust against its consensus violations temporarily at the early phase of training, thereby achieving the average consensus at the end.

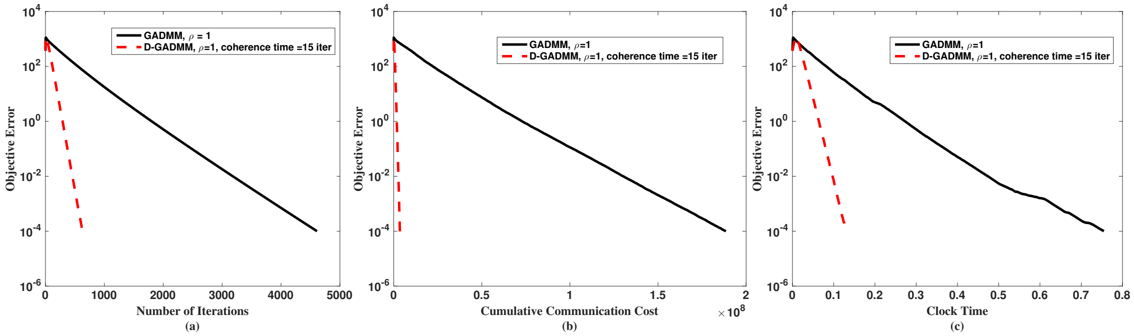
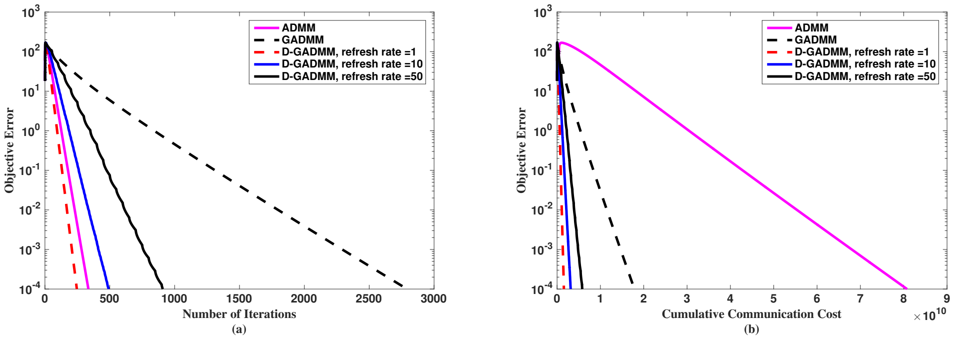


Figure 7: Objective error, total communication cost, and total running time of D-GADMM versus GADMM in linear regression with the synthetic dataset at $\rho = 1$, $N = 50$

We now extend GADMM to D-GADMM, and evaluate its performance under the time-varying network topology. One note to make, in simulating D-GADMM, we do not exchange dual variables between neighbors at every topology change as described in line 10, Algorithm 2. However, as we will show, D-GADMM still converges. Therefore, the extra communication overhead that might be encountered in D-GADMM when workers share their dual variables is avoided and the convergence is still preserved. We change the topology every 15 iterations. Therefore, we assume that the system coherence time is 15 iterations. To simulate the change in the topology, 50 workers are randomly distributed over a 250×250 m^2 square area every 15-th iteration. D-GADMM uses the method described in appendix D which consumes 2 iterations (4 communication rounds) to build the chain. In contrast, GADMM keeps the logical worker connectivity graph unchanged even when the underlying physical topology changes. In linear regression with the synthetic dataset and 50 workers, as observed in Fig. 7, even though D-GADMM consumes two iterations per topology change in building the chain, both the total number of iterations to achieve the objective error of $1E-4$ and the TC of D-GADMM are significantly reduced compared to GADMM. We observe that by changing the neighboring set of each worker more frequently, the convergence speed is significantly improved. Therefore, even for the static scenario in which the physical topology does not change, reconstructing the logical chain every few iterations can significantly improve the convergence speed.

We finally compare both GADMM and D-GADMM with the standard ADMM which requires a parameter server (star topology). Since the topology does not change, we replace “system coherence time” with “refresh rate”. Therefore, the objective of using D-GADMM is not to adapt to topology changes, while to improve the convergence speed of GADMM. To compare the algorithms, we use 24 workers ($N = 24$), and we randomly drop them over a 250×250 m^2 square area. For standard ADMM, we use the worker that is closest to the center of the grid as the parameter server.

As observed from Fig. 8, compared to GADMM, standard ADMM requires fewer iterations to achieve the objective error of $1E-4$, but that comes at significantly higher communication cost as shown in Fig. 8-(b) (4 times higher cost than GADMM). We show that by randomly changing the logical connectivity graph and utilizing D-GADMM, we can reduce the gap in the number of iterations between GADMM and standard ADMM and significantly reduce



(a)

(b)

Figure 8: Objective error, total communication cost, and total running time of D-GADMM, GADMM, and Standared ADMM in linear regression with the synthetic dataset at $\rho = 1$, $N = 24$

the communication cost. In fact, Fig. 8 shows that by changing the logical graph every iteration, D-GADMM converges faster than standard ADMM and achieves a communication cost that is 40 times less. It is worth mentioning that for static physical topology, changing the logical graph comes at zero cost since workers can agree on a predefined pseudorandom sequence in the graph changes. Therefore, every worker knows its neighbors in the next iteration.

8. Conclusions and Future work

In this paper, we formulate a constrained optimization problem for distributed machine learning applications, and propose a novel decentralized algorithm based on ADMM, termed Group ADMM (GADMM) to solve this problem optimally for convex functions. GADMM is shown to maximize the communication efficiency of each worker. Extensive simulations in linear and logistic regression with synthetic and real datasets show significant improvements in convergence rate and communication overhead compared to the state-of-the-art algorithms. Furthermore, we extend GADMM to D-GADMM which accounts for time-varying network topologies. Both analysis and simulations confirm that D-GADMM achieves the same convergence guarantees as GADMM with lower communication overhead under the time-varying topology scenario. Constructing a communication-efficient logical chain may not always be possible; therefore, extending the algorithm to achieve a low communication overhead under an arbitrary topology could be an interesting topic for future study.

Appendix A. Proof of Lemma 1

Proof of statement (i): We note that $f_n(\boldsymbol{\theta}_n)$ for all n is closed, proper, and convex, hence $\boldsymbol{\mathcal{L}}_\rho$ is sub-differentiable. Since $\boldsymbol{\theta}^{k+1}_{n\in\mathcal{N}_h}$ minimizes $\boldsymbol{\mathcal{L}}_\rho(\boldsymbol{\theta}_{n\in\mathcal{N}_h}, \boldsymbol{\theta}^k_{n\in\mathcal{N}_t}, \boldsymbol{\lambda}_n)$, the following must hold true at each iteration $k+1$

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, n \in \mathcal{N}_h \setminus \{1\} \tag{33}$$

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, n = 1 \tag{34}$$

Note that we use (34) for worker 1 since it does not have a left neighbor (*i.e.,* $\boldsymbol{\lambda}_0^{k+1}$ is not defined). However, for simplicity and to avoid writing separate equations for the edge workers (workers 1 and N), we use: $\boldsymbol{\lambda}_0^{k+1} = \boldsymbol{\lambda}_N^{k+1} = 0$ throughout the rest of the proof. Therefore, we can use a single equation for each group (*e.g.,* equation (33) for $n \in \mathcal{N}_h$).

The result in (33) implies that $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_h$ minimizes the following convex objective function

$$f_n(\boldsymbol{\theta}_n) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}_n \rangle. \tag{35}$$

Next, since $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_h$ is the minimizer of (35), then, it holds that

$$f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \le f_n(\boldsymbol{\theta}^\star) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}^\star \rangle \tag{36}$$

where $\boldsymbol{\theta}^\star$ is the optimal value of the problem in (8)-(9). Similarly for $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_t$ satisfies (17) and it holds that

$$f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \le f_n(\boldsymbol{\theta}^\star) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle. \tag{37}$$

Adding (36) and (37), and then taking the summation over all the workers, we get

$$\begin{aligned}
&\sum_{n=1}^N f_n(\boldsymbol{\theta}_n^{k+1}) + \sum_{n\in\mathcal{N}_t} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle + \sum_{n\in\mathcal{N}_h} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \\
&\qquad \le \sum_{n=1}^N f_n(\boldsymbol{\theta}^\star) + \sum_{n\in\mathcal{N}_t} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle + \sum_{n\in\mathcal{N}_h} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}^\star \rangle
\end{aligned} \tag{38}$$

After rearranging the terms, we get

$$\begin{aligned}
\sum_{n=1}^N f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^N f_n(\boldsymbol{\theta}^\star) \le{}& \sum_{n\in\mathcal{N}_t} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle + \sum_{n\in\mathcal{N}_h} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle \\
&- \sum_{n\in\mathcal{N}_t} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle - \sum_{n\in\mathcal{N}_h} \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \\
&+ \sum_{n\in\mathcal{N}_h} \langle \mathbf{s}_{n\in\mathcal{N}_h}^{k+1}, \boldsymbol{\theta}^\star - \boldsymbol{\theta}_n^{k+1} \rangle.
\end{aligned} \tag{39}$$

Note that,

$$\sum_{n\in\mathcal{N}_h}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}_n\rangle =\langle \boldsymbol{\lambda}_{1}^{k+1},\boldsymbol{\theta}_1\rangle - \langle \boldsymbol{\lambda}_{2}^{k+1},\boldsymbol{\theta}_3\rangle + \langle \boldsymbol{\lambda}_{3}^{k+1},\boldsymbol{\theta}_3\rangle + \cdots$$
$$\cdots - \langle \boldsymbol{\lambda}_{N-2}^{k+1},\boldsymbol{\theta}_{N-1}\rangle + \langle \boldsymbol{\lambda}_{N-1}^{k+1},\boldsymbol{\theta}_{N-1}\rangle, \tag{40}$$

and

$$\sum_{n\in\mathcal{N}_t}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}_n\rangle = - \langle \boldsymbol{\lambda}_{1}^{k+1},\boldsymbol{\theta}_2\rangle + \langle \boldsymbol{\lambda}_{2}^{k+1},\boldsymbol{\theta}_2\rangle - \langle \boldsymbol{\lambda}_{3}^{k+1},\boldsymbol{\theta}_4\rangle + \cdots$$
$$\cdots - \langle \boldsymbol{\lambda}_{N-1}^{k+1},\boldsymbol{\theta}_{N}\rangle + \langle \boldsymbol{\lambda}_{N-1}^{k+1},\boldsymbol{\theta}_{N}\rangle. \tag{41}$$

From (40) and (41), at $\boldsymbol{\theta}_n^{k+1}$, it holds that

$$\begin{aligned}
&\sum_{n\in\mathcal{N}_t}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}_n^{k+1}\rangle + \sum_{n\in\mathcal{N}_h}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}_n^{k+1}\rangle\\
&\quad= \langle \boldsymbol{\lambda}_{1}^{k+1},\boldsymbol{\theta}_1^{k+1}-\boldsymbol{\theta}_2^{k+1}\rangle + \langle \boldsymbol{\lambda}_{2}^{k+1},\boldsymbol{\theta}_2^{k+1}-\boldsymbol{\theta}_3^{k+1}\rangle + \cdots + \langle \boldsymbol{\lambda}_{N-1}^{k+1},\boldsymbol{\theta}_{N-1}^{k+1}-\boldsymbol{\theta}_N^{k+1}\rangle\\
&\quad= \langle \boldsymbol{\lambda}_{1}^{k+1},\mathbf{r}_{1,2}^{k+1}\rangle + \langle \boldsymbol{\lambda}_{2}^{k+1},\mathbf{r}_{2,3}^{k+1}\rangle + \cdots + \langle \boldsymbol{\lambda}_{N-1}^{k+1},\mathbf{r}_{N-1,N}^{k+1}\rangle,
\end{aligned} \tag{42}$$

where for the second equality, we have used the definition of primal residuals defined after (24). Similarly, it holds for $\boldsymbol{\theta}^\star$ that

$$\begin{aligned}
&\sum_{n\in\mathcal{N}_t}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}^\star\rangle + \sum_{n\in\mathcal{N}_h}\langle -\boldsymbol{\lambda}_{n-1}^{k+1}+\boldsymbol{\lambda}_{n}^{k+1},\boldsymbol{\theta}^\star\rangle \qquad (43)\\
&\quad= \langle \boldsymbol{\lambda}_{1}^{k+1},\boldsymbol{\theta}^\star\rangle + \langle \boldsymbol{\lambda}_{2}^{k+1}-\boldsymbol{\lambda}_{1}^{k+1},\boldsymbol{\theta}^\star\rangle + \langle \boldsymbol{\lambda}_{3}^{k+1}-\boldsymbol{\lambda}_{2}^{k+1},\boldsymbol{\theta}^\star\rangle + \cdots + \langle \boldsymbol{\lambda}_{N}^{k+1}-\boldsymbol{\lambda}_{N-1}^{k+1},\boldsymbol{\theta}^\star\rangle\\
&\quad= 0.
\end{aligned}$$

The equality in (43) holds since $\boldsymbol{\lambda}_N^{k+1}=\mathbf{0}$. Next, substituting the results from (42) and (43) into (39), we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \leq -\sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_{n}^{k+1},\mathbf{r}_{n,n+1}^{k+1}\rangle + \sum_{n\in\mathcal{N}_h}\langle \mathbf{s}_{n\in\mathcal{N}_h}^{k+1},\boldsymbol{\theta}^\star-\boldsymbol{\theta}_n^{k+1}\rangle, \tag{44}$$

which concludes the proof of statement (i) of Lemma 1.

Proof of statement (ii): The proof of this Lemma is along the similar line as in (Boyd et al., 2011, A.3) but is provided here for completeness. We note that for a saddle point $(\boldsymbol{\theta}^\star,\{\boldsymbol{\lambda}_n^\star\}_{\forall n})$ of $\boldsymbol{\mathcal{L}}_0(\{\boldsymbol{\theta}_n\}_{\forall n},\{\boldsymbol{\lambda}_n\}_{\forall n})$, it holds that

$$\boldsymbol{\mathcal{L}}_0(\boldsymbol{\theta}^\star,\{\boldsymbol{\lambda}_n^\star\}_{\forall n}) \leq \boldsymbol{\mathcal{L}}_0(\boldsymbol{\{}\boldsymbol{\theta}_n^{k+1}\}\forall n,\{\boldsymbol{\lambda}_n^\star\}\forall n) \tag{45}$$

for all n. Substituting the expression for the Lagrangian from (10) on the both sides of (45), we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \leq \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) + \sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_{n}^{\star},\mathbf{r}_{n,n+1}^{k+1}\rangle. \tag{46}$$

After rearranging the terms, we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \geq -\sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_{n}^{\star},\mathbf{r}_{n,n+1}^{k+1}\rangle \tag{47}$$

which is the statement (ii) of Lemma 1.

Appendix B. Proof of Theorem 2

To proceed with the analysis, add (44) and (47), multiply by 2, we get

$$2\sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\sum_{n\in\mathcal{N}_h}\langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^{\star}\rangle \leq 0. \tag{48}$$

By applying, $\boldsymbol{\lambda}_n^{k+1}=\boldsymbol{\lambda}_n^{k}+\rho\mathbf{r}_{n,n+1}^{k+1}$ obtained from the dual update in (15), (48) can be recast as

$$2\sum_{n=1}^{N-1}\langle \boldsymbol{\lambda}_n^{k}+\rho\mathbf{r}_{n,n+1}^{k+1}-\boldsymbol{\lambda}_n^{\star}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\sum_{n\in\mathcal{N}_h}\langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^{\star}\rangle \leq 0. \tag{49}$$

Note that the first term on the left hand side of (49) can be written as

$$\sum_{n=1}^{N-1}2\langle \boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}, \mathbf{r}_{n,n+1}^{k+1}\rangle + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2. \tag{50}$$

Replacing $\mathbf{r}_{n,n+1}^{k+1}$ in the first and second terms of (50) with $\frac{\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{k}}{\rho}$, we get

$$\sum_{n=1}^{N-1}(2/\rho)\langle \boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}, \boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{k}\rangle + (1/\rho)\left\|\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{k}\right\|^2 + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2. \tag{51}$$

Using the equality $\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{k}=(\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star})-(\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star})$, we can rewrite (51) as

$$\begin{aligned}
&\sum_{n=1}^{N-1}(2/\rho)\langle \boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}, (\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star})-(\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star})\rangle + (1/\rho)\left\|(\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star})-(\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star})\right\|^2 + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2\\
&=\sum_{n=1}^{N-1}(2/\rho)\langle \boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}, \boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star}\rangle - (2/\rho)\left\|\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}\right\|^2 + (1/\rho)\left\|\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star}\right\|^2\\
&\qquad - (2/\rho)\langle \boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star}, \boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}\rangle + 1/\rho\left\|\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}\right\|^2 + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 \qquad (52)\\
&=\sum_{n=1}^{N-1}\left[(1/\rho)\left\|\boldsymbol{\lambda}_n^{k+1}-\boldsymbol{\lambda}_n^{\star}\right\|^2 - (1/\rho)\left\|\boldsymbol{\lambda}_n^{k}-\boldsymbol{\lambda}_n^{\star}\right\|^2 + \rho\left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2\right]. \qquad (53)
\end{aligned}$$

Next, consider the second term on the left hand side of (49). From the equality (26), it holds that

$$\begin{aligned}
&2\sum_{n\in\mathcal{N}_h}\langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^{\star}\rangle \qquad (54)\\
&\qquad=\sum_{n\in\mathcal{N}_h\setminus\{1\}}\Big(2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1}-\boldsymbol{\theta}_{n-1}^{k}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^{\star}\rangle\Big) + \sum_{n\in\mathcal{N}_h}\Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1}-\boldsymbol{\theta}_{n+1}^{k}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^{\star}\rangle\Big).
\end{aligned}$$

Note that $\boldsymbol{\theta}_n^{k+1} = -\mathbf{r}_{n-1,n}^{k+1} + \boldsymbol{\theta}_{n-1}^{k+1} = \mathbf{r}_{n,n+1}^{k+1} + \boldsymbol{\theta}_{n+1}^{k+1}, \forall n = \{2, \cdots, N-1\}$, which implies that we can rewrite (54) as follows

$$\begin{aligned} & 2\sum_{n\in\mathcal{N}_h} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}^\star \rangle \\ & \quad = \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star\rangle \Big) \\ & \qquad\qquad + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star\rangle \Big). \end{aligned} \tag{55}$$

Using the equalities,

$$\begin{aligned} \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star =& (\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}) + (\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star), \forall n \in \mathcal{N}_h \setminus \{1\} \\ \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star =& (\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}) + (\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star), \forall n \in \mathcal{N}_h \end{aligned} \tag{56}$$

we rewrite the right hand side of (55) as

$$\begin{aligned} & \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, (\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}) + (\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star))\rangle \Big) \\ & \qquad + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, (\boldsymbol{\theta}_{n}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}) + (\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star))\rangle \Big) \\ & = \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k} \right\|^2 + 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\rangle \Big) \\ & \qquad + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k} \right\|^2 + 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}_{n+1}^\star\rangle \Big). \end{aligned} \tag{57}$$

Further using the equalities

$$\begin{aligned} \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k} =& (\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star), \forall n \in \mathcal{N}_h \setminus \{1\} \\ \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k} =& (\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star), \forall n \in \mathcal{N}_h \end{aligned} \tag{58}$$

we can write (57) as

$$\begin{aligned} & \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k} \right\|^2 + 2\rho\langle (\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star), \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\rangle \Big) \\ & \qquad + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k} \right\|^2 \\ & \qquad\quad + 2\rho\langle (\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star), \boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\rangle \Big) \end{aligned} \tag{59}$$

$$\begin{aligned} & = \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k} \right\|^2 + 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\rangle \\ & \qquad - 2\rho \left\| \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star \right\|^2 \Big) + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle + 2\rho \left\| \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k} \right\|^2 \\ & \qquad + 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\rangle - 2\rho \left\| \boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star \right\|^2 \Big). \end{aligned} \tag{60}$$

After rearranging the terms, we can write

$$
\begin{aligned}
= \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(& -2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 + \rho \left\|(\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star)\right\|^2 \\
& + 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\rangle - 2\rho \parallel \boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star \parallel_2^2 \Big) + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle \\
& + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2 + \rho \left\|(\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star)\right\|^2 \\
& + 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\rangle - 2\rho \left\|\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big).
\end{aligned}
\tag{61}
$$

Next, expanding the square terms in (61), we get

$$
\begin{align}
& 2\sum_{n\in\mathcal{N}_h} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}^\star\rangle \nonumber \\
& \quad = \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 \tag{62} \\
& \qquad + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big) + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle \nonumber \\
& \qquad\quad + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2 + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big). \nonumber
\end{align}
$$

Substituting the equalities from (53) and (62) to the left hand side of (49), we obtain

$$
\begin{align}
& \sum_{n=1}^{N-1} \Big[(1/\rho) \left\|\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star\right\|^2 - (1/\rho) \left\|\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star\right\|^2 + \rho \left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 \Big] \nonumber \\
& \quad + \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 \tag{63} \\
& \qquad + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big) + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle \nonumber \\
& \qquad\quad + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2 + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big) \leq 0. \tag{64}
\end{align}
$$

Multiplying both the sides by -1, we get

$$
\begin{align}
& \sum_{n=1}^{N-1} \Big[-(1/\rho) \left\|\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star\right\|^2 + (1/\rho) \left\|\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star\right\|^2 - \rho \left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 \Big] \nonumber \\
& \quad - \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 \tag{65} \\
& \qquad + \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n-1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big) + \sum_{n\in\mathcal{N}_h} \Big(2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle \nonumber \\
& \qquad\quad + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2 + \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}^\star\right\|^2 - \rho \left\|\boldsymbol{\theta}_{n+1}^{k} - \boldsymbol{\theta}^\star\right\|^2 \Big) \geq 0, \tag{66}
\end{align}
$$

After rearranging the terms in (65) and using the definition of the Lyapunov function in (32), we get

$$\begin{aligned} V_{k+1} \leq V_k - \sum_{n=1}^{N-1} \rho \left\| \mathbf{r}_{n,n+1}^{k+1} \right\|^2 - \Big[\sum_{n \in \mathcal{N}_h \setminus \{1\}} \rho \left\| \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k} \right\|^2 + \sum_{n \in \mathcal{N}_h} \rho \left\| \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k} \right\|^2 \Big] \\ -\Big[\sum_{n \in \mathcal{N}_h \setminus \{1\}} -2\rho \langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1} \rangle + \sum_{n \in \mathcal{N}_h} 2\rho \langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1} \rangle \Big]. \end{aligned} \tag{67}$$

In order to prove that $k+1$ is a one step towards the optimal solution or the Lyapunov function decreases monotonically at each iteration, we need to show that the sum of the inner product terms on the right hand side of the inequality is positive. In other words, we need to prove that the term $\sum_{n \in \mathcal{N}_h \setminus \{1\}} -2\rho \langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1} \rangle + \sum_{n \in \mathcal{N}_h} 2\rho \langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1} \rangle$ is always positive. Note that this term can be written as.

$$\begin{aligned} &\sum_{n \in \mathcal{N}_h \setminus \{1\}} - 2\rho \langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1} \rangle + \sum_{n \in \mathcal{N}_h} 2\rho \langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1} \rangle \qquad (68) \\ &= 2\rho \Big[\langle \mathbf{r}_{1,2}^{k+1}, \boldsymbol{\theta}_2^{k+1} - \boldsymbol{\theta}_2^{k} \rangle - \langle \mathbf{r}_{2,3}^{k+1}, \boldsymbol{\theta}_2^{k+1} - \boldsymbol{\theta}_2^{k} \rangle + \langle \mathbf{r}_{3,4}^{k+1}, \boldsymbol{\theta}_4^{k+1} - \boldsymbol{\theta}_4^{k} \rangle - \langle \mathbf{r}_{4,5}^{k+1}, \boldsymbol{\theta}_4^{k+1} - \boldsymbol{\theta}_4^{k} \rangle + \\ &\qquad \cdots + \mathbf{r}_{N-1,N}^{k+1} (\boldsymbol{\theta}_N^{k+1} - \boldsymbol{\theta}_N^{k}) \Big] \\ &= 2\rho \langle \mathbf{r}_{1,2}^{k+1} - \mathbf{r}_{2,3}^{k+1}, \boldsymbol{\theta}_2^{k+1} - \boldsymbol{\theta}_2^{k} \rangle + 2\rho \langle \mathbf{r}_{3,4}^{k+1} - \mathbf{r}_{4,5}^{k+1}, \boldsymbol{\theta}_4^{k+1} - \boldsymbol{\theta}_4^{k} \rangle + \\ &\qquad \cdots + 2\rho \langle \mathbf{r}_{N-1,N}^{k+1}, \boldsymbol{\theta}_N^{k+1} - \boldsymbol{\theta}_N^{k} \rangle. \end{aligned}$$

We know that $\boldsymbol{\theta}_{n \in \mathcal{N}_t}^{k+1}$ minimizes $f_n(\boldsymbol{\theta}_n) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n \rangle$; hence it holds that

$$f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \leq f_n(\boldsymbol{\theta}_n^{k}) + \langle -\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k} \rangle, \tag{69}$$

Similarly, $\boldsymbol{\theta}_{n \in \mathcal{N}_t}^{k}$ minimizes $f_n(\boldsymbol{\theta}_n) + \langle -\boldsymbol{\lambda}_{n-1}^{k} + \boldsymbol{\lambda}_n^{k}, \boldsymbol{\theta}_n \rangle$, which implies that

$$f_n(\boldsymbol{\theta}_n^{k}) + \langle -\boldsymbol{\lambda}_{n-1}^{k} + \boldsymbol{\lambda}_n^{k}, \boldsymbol{\theta}_n^{k} \rangle \leq f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n-1}^{k} + \boldsymbol{\lambda}_n^{k}, \boldsymbol{\theta}_n^{k+1} \rangle. \tag{70}$$

Adding (69) and (70), we get

$$\langle (-\boldsymbol{\lambda}_{n-1}^{k+1} + \boldsymbol{\lambda}_n^{k+1}) - (-\boldsymbol{\lambda}_{n-1}^{k} + \boldsymbol{\lambda}_n^{k}), \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_n^{k} \rangle \leq 0. \tag{71}$$

Further after rearranging, we get

$$\langle (\boldsymbol{\lambda}_{n-1}^{k} - \boldsymbol{\lambda}_{n-1}^{k+1}) + (\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{k}), \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_n^{k} \rangle \leq 0. \tag{72}$$

By knowing that $\mathbf{r}_{n-1,n}^{k+1} = (1/\rho)(\boldsymbol{\lambda}_{n-1}^{k+1} - \boldsymbol{\lambda}_{n-1}^{k})$ and $\mathbf{r}_{n,n+1}^{k+1} = (1/\rho)(\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{k})$, (72) can be written as

$$-\rho \langle \mathbf{r}_{n-1,n}^{k+1} - \mathbf{r}_{n,n+1}^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_n^{k} \rangle \leq 0, \forall n \in \mathcal{N}_t. \tag{73}$$

The above inequality implies that

$$\rho \langle \mathbf{r}_{n-1,n}^{k+1} - \mathbf{r}_{n,n+1}^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_n^{k} \rangle \geq 0, \forall n \in \mathcal{N}_t. \tag{74}$$

Note that since worker N does not have a right neighbor, $\mathbf{r}_{N,N+1}^{k+1} = \boldsymbol{\lambda}_N^{k+1} = \boldsymbol{\lambda}_N^k = 0$.

Next, for $\rho > 0$. Using the inequality in (74) into (68), we get

$$\sum_{n\in\mathcal{N}_h\setminus\{1\}} - 2\rho\langle \boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \sum_{n\in\mathcal{N}_h} 2\rho\langle \boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}, \mathbf{r}_{n,n+1}^{k+1}\rangle \geq 0. \tag{75}$$

Next, we use the result in (75) into (67) to get

$$V_{k+1} \leq V_k - \sum_{n=1}^{N-1} \rho \left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 - \Big[\sum_{n\in\mathcal{N}_h\setminus\{1\}} \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 + \sum_{n\in\mathcal{N}_h} \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2\Big]. \tag{76}$$

The result in (76) proves that V_{k+1} decreases with k. Now, since $V_k \geq 0$ and $V_k \leq V_0$, it holds that $\left[\sum_{n=1}^{N-1} \rho \left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 + \Big[\sum_{n\in\mathcal{N}_h\setminus\{1\}} \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 + \sum_{n\in\mathcal{N}_h} \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2\Big]\right]$. is bounded. Taking the telescopic sum over k in (76) as limit $K \to \infty$, we get

$$\lim_{K\to\infty} \sum_{k=0}^{K} \left[\sum_{n=1}^{N-1} \rho \left\|\mathbf{r}_{n,n+1}^{k+1}\right\|^2 + \Big[\sum_{n\in\mathcal{N}_h\setminus\{1\}} \rho \left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|^2 + \sum_{n\in\mathcal{N}_h} \rho \left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\|^2\Big]\right] \leq V_0. \tag{77}$$

The result in (77) implies that the primal residual $\mathbf{r}_{n,n+1}^{k+1} \to \mathbf{0}$ as $k \to \infty$ for all $n \in \{1, \cdots, N-1\}$ completing the proof of statement (i) in Theorem 2. Similarly, the norm differences $\left\|\boldsymbol{\theta}_{n-1}^{k+1} - \boldsymbol{\theta}_{n-1}^{k}\right\|$ and $\left\|\boldsymbol{\theta}_{n+1}^{k+1} - \boldsymbol{\theta}_{n+1}^{k}\right\| \to \mathbf{0}$ as $k \to \infty$ which implies that the dual residual $\mathbf{s}_n^k \to \mathbf{0}$ as $k \to \infty$ for all $n \in \mathcal{N}_h$ stated in the statement (ii) of Theorem 2. In order to prove the statement (iii) of Theorem 2, consider the lower and the upper bounds on the objective function optimality gap given by

$$\sum_{n=1}^{N} [f_n(\boldsymbol{\theta}_n^{k+1}) - f_n(\boldsymbol{\theta}^\star)] \leq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1}, \mathbf{r}_{n,n+1}^{k+1}\rangle + \sum_{n\in\mathcal{N}_h} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}^\star - \boldsymbol{\theta}_n^{k+1}\rangle \tag{78}$$

$$\sum_{n=1}^{N} [f_n(\boldsymbol{\theta}_n^{k+1}) - f_n(\boldsymbol{\theta}^\star)] \geq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{\star}, \mathbf{r}_{n,n+1}^{k+1}\rangle. \tag{79}$$

Note that from the results in statement (i) and (ii) of Theorem 2, it holds that the right hand side of the upper bound in (78) converge to zero as $k \to \infty$ and also the right hand side of the lower bound in (79) converges to zero as $k \to \infty$. This implies that

$$\lim_{k\to\infty} \sum_{n=1}^{N} [f_n(\boldsymbol{\theta}_n^{k+1}) - f_n(\boldsymbol{\theta}^\star)] = 0 \tag{80}$$

which is the statement (iii) of Theorem 2.

Appendix C. Method for D-GADMM Chain Construction

To ensure that a chain in a given graph is found in a decentralized way, we use the following method.

- The N workers (N is assumed to be even) share a pseudorandom code that is used every τ seconds, where τ is the system coherence time, to generate a set $\mathcal{H}$ containing $(N/2-2)$ integer numbers, with each number chosen from the set $\{2, \cdots, N-1\}$. In other words, we assume that the topology changes every τ seconds. Note that the generated numbers at $i \cdot \tau$ and $(i+1) \cdot \tau$ time slots may differ. However, at the $i \cdot \tau$-th time slot, the same set of numbers is generated across workers with no communication.

- Every worker with physical index $n \in \big\{\mathcal{H} \cup \{1\}\big\}$ is assigned to the head set. Note that the worker whose physical index 1 is always assigned to the head set. On the other hand, every worker with physical index n such that $n \notin \big\{\mathcal{H} \cup \{1\}\big\}$ assigns itself to the tail set. Therefore, the worker whose physical index N is always assigned to the tail set. Following this strategy, the number of heads will be equal to the number of tails, and are both equal to $N/2$.

- Every worker in the head set broadcasts its physical index alongside a pilot signal. Pilot signal is a signal known to each worker. It is used to measure the signal strength and find neighboring workers.

- Every worker in the tail set calculates its cost of communication to every head based on the received signal strength. For example, the communication cost between head n and tail m is equal to [1/power of the received signal from head n to tail m], in which the link with lower received signal level is more costly, as it is incurring higher transmission power.

- Every worker in the tail set broadcasts a vector of length $N/2$, containing the communication cost to the $N/2$ heads, *i.e.,* the first element in the vector captures the communication cost between this tail and worker 1, since worker 1 is always in the head set, whereas the second element represents the communication cost between this tail and the first index in the head set $\mathcal{H}$ and so on.

- Once head worker $n \in \big\{\mathcal{H} \cup \{1\}\big\}$ receives the communication cost vector from tail workers, it finds a communication-efficient chain that starts from worker 1 and passes through every other worker to reach worker N. In our simulations, we use the following simple and greedy strategy that is performed by every head to ensure they generate the same chain. The strategy is as follows:

  - Find the tail that has the minimum communication cost to worker 1 and create a link between this tail and worker 1.
  - From the remaining set of heads, find the head that has the minimum communication cost to this tail and create a link between this head and the corresponding tail.
  - Follow the same strategy until all workers are connected. When every head follows this strategy, all heads will generate the same chain.

  - Under the following two assumptions: (i) the communication cost between any pair of workers is $< \infty$, and (ii) no two tails have equal communication cost to the same head, this strategy guarantees that every head will generate the same chain.

- Once every head finds its chain, all neighbors share their current models, and D-GADMM is carried out for τ seconds using the current chain.

Note that, the described heuristic requires 4 communication rounds (2 iterations). Finally, it is worth mentioning that this approach has no guarantee to find the most communication-efficient chain. As mentioned in section 6, our focus in this paper is not to design the chain construction algorithm.

Appendix D. Convergence Analysis of D-GADMM

For the dynamic settings, we assume that the first $n = 1$ and the last node $n = N$ are fixed and the others can move at each iterate. Therefore, we denote the neighbors to each node n at iteration k as $n_{l,k}$ and $n_{r,k}$ as the left and the right neighbors , respectively. Note that when, $n_{l,k} = n-1$ and $n_{r,k} = n+1$ for all k, we recover the GADMM implementation. With that in mind, we start by writing the augmented Lagrangian of the optimization problem in (8)-(9) at each iteration k as

$$\boldsymbol{\mathcal{L}}_\rho^k(\{\boldsymbol{\theta}_n\}_{n=1}^N, \boldsymbol{\lambda}) = \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n) + \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}} \rangle + \frac{\rho}{2} \sum_{n=1}^{N-1} \|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n+1}\|^2, \tag{81}$$

where $\boldsymbol{\lambda} := [\boldsymbol{\lambda}_1^T, \cdots, \boldsymbol{\lambda}_{N-1}^T]^T$ is the collection of dual variables. Note that the set of nodes in head $\mathcal{N}_h^k$ and tail $\mathcal{N}_t^k$ will change with k.6 The primal and dual variables under GADMM are updated in the following three steps. The modified algorithm updates are written as

1. At iteration $k+1$, the *primal variables of head workers* are updated as:

$$\begin{aligned}\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n_{l,k}}^k, \boldsymbol{\theta}_{n_{l,k}}^k - \boldsymbol{\theta}_n \rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}}^k \rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_{n_{l,k}}^k - \boldsymbol{\theta}_n\|^2 \\ + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}}^k\|^2\big], n \in \mathcal{N}_h \setminus \{1\}\end{aligned} \tag{82}$$

   Since the first head worker ($n = 1$) does not have a left neighbor ($\boldsymbol{\theta}_{n-1}$ is not defined), its model is updated as follows.

$$\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}}^k \rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{l,k}}^k\|^2\big], n = 1 \tag{83}$$

2. After the updates in (82) and (83), head workers send their updates to their two tail neighbors. The *primal variables of tail workers* are then updated as:

$$\begin{aligned}\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n_{l,k}}^k, \boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n \rangle + \langle \boldsymbol{\lambda}_n^k, \boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}}^{k+1} \rangle + \frac{\rho}{2}\|\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n\|^2 \\ + \frac{\rho}{2}\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_{n_{r,k}}^{k+1}\|^2\big], n \in \mathcal{N}_t \setminus \{N\}.\end{aligned} \tag{84}$$

Since the last tail worker ($n = N$) does not have a right neighbor ($\boldsymbol{\theta}_{n+1}$ is not defined), its model is updated as follows

$$\boldsymbol{\theta}_n^{k+1} = \arg\min_{\boldsymbol{\theta}_n} \big[f_n(\boldsymbol{\theta}_n) + \langle \boldsymbol{\lambda}_{n_{l,k}}^k, \boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n \rangle + \frac{\rho}{2} \|\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n\|^2\big], n = N. \tag{85}$$

3. After receiving the updates from neighbors, *every worker locally updates its dual variables* $\boldsymbol{\lambda}_{n-1}$ and $\boldsymbol{\lambda}_n$ as follows

$$\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^{k+1}), n = \{1, \cdots, N-1\}. \tag{86}$$

Note that when the topology changes, $\boldsymbol{\lambda}_n^k$ of worker n is received from the left neighbor $n_{l,k}$ before updating $\boldsymbol{\lambda}_n^{k+1}$ according to (86). For the proof, we start with the necessary and sufficient optimality conditions, which are the primal and the dual feasibility conditions (Boyd et al., 2011) for each k are defined as

$$\boldsymbol{\theta}_n^\star = \boldsymbol{\theta}_{n_{l,k}}^\star, n \in \{2, \cdots, N\} \qquad \text{(primal feasibility)} \tag{87}$$

$$\begin{aligned} &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) - \boldsymbol{\lambda}_{n_{l,k}}^\star + \boldsymbol{\lambda}_n^\star, n \in \{2, \cdots, N-1\} \\ &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) + \boldsymbol{\lambda}_n^\star, n = 1 \qquad \text{(dual feasibility)} \\ &\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^\star) + \boldsymbol{\lambda}_{n-1}^\star, n = N \end{aligned} \tag{88}$$

We remark that the optimal values $\boldsymbol{\theta}_n^\star$ are equal for each n, we denote $\boldsymbol{\theta}^\star = \boldsymbol{\theta}_n^\star = \boldsymbol{\theta}_{n-1}^\star$ for all n. Note that, at iteration $k+1$, we calculate $\boldsymbol{\theta}_n^{k+1}$ for all $n \in \mathcal{N}_t^k \setminus \{N\}$ as in (13), from the first order optimality condition, it holds that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n_{l,k}}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{l,k}}^{k+1}) + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^{k+1}). \tag{89}$$

Next, rewrite the equation in (89) as

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \big(\boldsymbol{\lambda}_{n_{l,k}}^k + \rho(\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n^{k+1})\big) + \Big(\boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^{k+1})\Big). \tag{90}$$

From the update in (86), the equation in (90) implies that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, n \in \mathcal{N}_t^k \setminus \{N\}. \tag{91}$$

Note that for the N-th worker, we calculate $\boldsymbol{\theta}_N^{k+1}$ as in (14), then we follow the same steps, and we get

$$\mathbf{0} \in \partial f_N(\boldsymbol{\theta}_N^{k+1}) - \boldsymbol{\lambda}_{N_{l,k}}^{k+1}. \tag{92}$$

From the result in (91) and (92), it holds that the dual feasibility condition in (88) is always satisfied for all $n \in \mathcal{N}_t^k$.

Next, consider every $\boldsymbol{\theta}_n^{k+1}$ such that $n \in \mathcal{N}_h^k \setminus \{1\}$ which is calculated as in (82) at iteration k. Similarly from the first order optimality condition, we can write

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n_{l,k}}^k + \boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{l,k}}^k) + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^k). \tag{93}$$

Note that in (93), we don't have all the primal variables calculated at $k+1$ instance. Hence, we add and subtract the terms $\boldsymbol{\theta}_{n_{l,k}}^{k+1}$ and $\boldsymbol{\theta}_{n_{r,k}}^{k+1}$ in (93) to get

$$\begin{aligned}\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \big(\boldsymbol{\lambda}_{n_{l,k}}^k + \rho(\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_n^{k+1})\big) + \Big(\boldsymbol{\lambda}_n^k + \rho(\boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^{k+1})\Big) \\ + \rho(\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_{n_{l,k}}^k) + \rho(\boldsymbol{\theta}_{n_{r,k}}^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^k). \qquad (94)\end{aligned}$$

From the update in (86), it holds that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \rho(\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_{n_{l,k}}^k) + \rho(\boldsymbol{\theta}_{n_{r,k}}^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^k). \qquad (95)$$

Following the same steps for the first head worker ($n = 1$) after excluding the terms $\boldsymbol{\lambda}_0^k$ and $\rho(\boldsymbol{\theta}_1^{k+1} - \boldsymbol{\theta}_0^k)$ from (93) (worker 1 does not have a left neighbor) gives

$$\mathbf{0} \in \partial f_1(\boldsymbol{\theta}_1^{k+1}) + \boldsymbol{\lambda}_1^{k+1} + \rho(\boldsymbol{\theta}_{1_{r,k}}^{k+1} - \boldsymbol{\theta}_{1_{r,k}}^k). \qquad (96)$$

Let $\mathbf{s}_n^{k+1}$, the dual residual of worker $n \in \mathcal{N}_h^k$ at iteration $k+1$, be defined as follows

$$\mathbf{s}_n^{k+1} = \begin{cases} \rho(\boldsymbol{\theta}_{n_{l,k}}^{k+1} - \boldsymbol{\theta}_{n_{l,k}}^k) + \rho(\boldsymbol{\theta}_{n_{r,k}}^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^k), \text{ for } n \in \mathcal{N}_h^k \setminus \{1\} \\ \rho(\boldsymbol{\theta}_{n_{r,k}}^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^k), \text{ for } n = 1. \end{cases} \qquad (97)$$

Next, we discuss about the primal feasibility condition in (87) at iteration $k+1$. Let $\mathbf{r}_{n,n_{r,k}}^{k+1} = \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}_{n_{r,k}}^{k+1}$ be the primal residual of each worker $n \in \{1, \cdots, N-1\}$. To show the convergence of GADMM, we need to prove that the conditions in (87)-(88) are satisfied for each worker n. We have already shown that the dual feasibility condition in (88) is always satisfied for the tail workers, and the dual residual of tail workers is always zero. Therefore, to prove the convergence and the optimality of GADMM, we need to show that the $\mathbf{r}_{n,n_{r,k}}^k$ for all $n = 1, \cdots, N-1$ and $\mathbf{s}_{n\in\mathcal{N}_h^k}^k$ converge to zero, and $\sum_{n=1}^N f_n(\boldsymbol{\theta}_n^k)$ converges to $\sum_{n=1}^N f_n(\boldsymbol{\theta}^\star)$ as $k \to \infty$. We proceed as follows to prove the same.

We note that $f_n(\boldsymbol{\theta}_n)$ for all n is closed, proper, and convex, hence $\boldsymbol{\mathcal{L}}_\rho^k$ is sub-differentiable. Since $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_h^k$ at k minimizes $\boldsymbol{\mathcal{L}}_\rho^k(\boldsymbol{\theta}_{n\in\mathcal{N}_h}, \boldsymbol{\theta}_{n\in\mathcal{N}_t}^k, \boldsymbol{\lambda}_n)$, the following must hold true at each iteration $k+1$, which implies that

$$\mathbf{0} \in \partial f_n(\boldsymbol{\theta}_n^{k+1}) - \boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, n \in \mathcal{N}_h^k \setminus \{1\} \qquad (98)$$

$$\mathbf{0} \in \partial f_1(\boldsymbol{\theta}_1^{k+1}) + \boldsymbol{\lambda}_1^{k+1} + \mathbf{s}_1^{k+1}, n = 1 \qquad (99)$$

Note that we use (99) for worker 1 since it does not have a left neighbor (*i.e.,* $\boldsymbol{\lambda}_0^{k+1}$ is not defined). However, for simplicity and to avoid writing separate equations for the edge workers (workers 1 and N), we use: $\boldsymbol{\lambda}_0^{k+1} = \boldsymbol{\lambda}_N^{k+1} = 0$ throughout the rest of the proof. Therefore, we can use a single equation for each group (*e.g.,* equation (33) for $n \in \mathcal{N}_h^k$).

The result in (98) implies that $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_h^k$ minimizes the following convex objective function

$$f_n(\boldsymbol{\theta}_n) + \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n \rangle. \qquad (100)$$

Next, since $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_h^k$ is the minimizer of (100), then, it holds that

$$f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \leq f_n(\boldsymbol{\theta}^\star) + \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, \boldsymbol{\theta}^\star \rangle \tag{101}$$

where $\boldsymbol{\theta}^\star$ is the optimal value of the problem in (8)-(9). Similarly for $\boldsymbol{\theta}_n^{k+1}$ for $n \in \mathcal{N}_t^k$ satisfies (88) and it holds that

$$f_n(\boldsymbol{\theta}_n^{k+1}) + \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \leq f_n(\boldsymbol{\theta}^\star) + \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle. \tag{102}$$

Add (101) and (102), and then take the summation over all the workers, note that for a given k, the topology in the network is fixed, we get

$$\begin{aligned}
&\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) + \sum_{n \in \mathcal{N}_t^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle + \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \\
&\quad \leq \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) + \sum_{n \in \mathcal{N}_t^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle + \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1} + \mathbf{s}_n^{k+1}, \boldsymbol{\theta}^\star \rangle
\end{aligned} \tag{103}$$

After rearranging the terms, we get

$$\begin{aligned}
\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \leq{} & \sum_{n \in \mathcal{N}_t^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle + \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle \\
& - \sum_{n \in \mathcal{N}_t^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle - \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle \\
& + \sum_{n \in \mathcal{N}_h^k} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}^\star - \boldsymbol{\theta}_n^{k+1} \rangle.
\end{aligned} \tag{104}$$

Note that we can write

$$\sum_{n \in \mathcal{N}_t^t} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle + \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} \rangle = \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1}, \mathbf{r}_{n,n_r^k}^{k+1} \rangle, \tag{105}$$

where for the equality, we have used the definition of primal residuals defined after (95). Similarly, it holds for $\boldsymbol{\theta}^\star$ as

$$\sum_{n \in \mathcal{N}_t^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle + \sum_{n \in \mathcal{N}_h^k} \langle -\boldsymbol{\lambda}_{n_{l,k}}^{k+1} + \boldsymbol{\lambda}_n^{k+1}, \boldsymbol{\theta}^\star \rangle = 0. \tag{106}$$

The equality in (106) holds since $\boldsymbol{\lambda}_N^{k+1} = \mathbf{0}$. Next, substituting the results from (105) and (106) into (104), we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \leq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1}, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle + \sum_{n \in \mathcal{N}_h^k} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}^\star - \boldsymbol{\theta}_n^{k+1} \rangle, \tag{107}$$

which provides an upper bound on the optimality gap. Next, we get the lower bound as follows. We note that for a saddle point $(\boldsymbol{\theta}^\star, \{\boldsymbol{\lambda}_n^\star\}_{\forall n})$ of $\boldsymbol{\mathcal{L}}_0(\{\boldsymbol{\theta}_n\}_{\forall n}, \{\boldsymbol{\lambda}_n\}_{\forall n})$, it holds that

$$\boldsymbol{\mathcal{L}}_0(\boldsymbol{\theta}^\star, \{\boldsymbol{\lambda}_n^\star\}_{\forall n}) \leq \boldsymbol{\mathcal{L}}_0(\{\boldsymbol{\theta}_n^{k+1}\}_{\forall n}, \{\boldsymbol{\lambda}_n^\star\}_{\forall n}). \tag{108}$$

Substituting the expression for the Lagrangian from (81) on the both sides of (108), we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \leq \sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) + \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle. \tag{109}$$

After rearranging the terms, we get

$$\sum_{n=1}^{N} f_n(\boldsymbol{\theta}_n^{k+1}) - \sum_{n=1}^{N} f_n(\boldsymbol{\theta}^\star) \geq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle \tag{110}$$

which provide the lower bound on the optimality gap. Next, we show that both the lower and upper bound converges to zero as → ∞. This would prove that the optimality gap converges to zero with $k \to \infty$.

To proceed with the analysis, add (107) and (110), multiply by 2, we get

$$2\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle + 2 \sum_{n \in \mathcal{N}_h^k} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}^\star \rangle \leq 0. \tag{111}$$

From the dual update in (86), we have $\boldsymbol{\lambda}_n^{k+1} = \boldsymbol{\lambda}_n^k + \rho \mathbf{r}_{n,n_{r,k}}^{k+1}$ and (111) can be written as

$$2\sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}_n^{k} + \rho \mathbf{r}_{n,n_{r,k}}^{k+1} - \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle + 2 \sum_{n \in \mathcal{N}_h^k} \langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1} - \boldsymbol{\theta}^\star \rangle \leq 0. \tag{112}$$

Note that the first term on the left hand side of (112) can be written as

$$\sum_{n=1}^{N-1} 2\langle \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star, \mathbf{r}_{n,n_{r,k}}^{k+1} \rangle + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2 + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2. \tag{113}$$

Replacing $\mathbf{r}_{n,n_{r,k}}^{k+1}$ in the first and second terms of (113) with $\frac{\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^k}{\rho}$, we get

$$\sum_{n=1}^{N-1} (2/\rho)\langle \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star, \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{k} \rangle + (1/\rho) \left\| \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{k} \right\|^2 + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2. \tag{114}$$

Using the equality $\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{k} = (\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star) - (\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star)$, we can write (114) as

$$\begin{aligned}
&\sum_{n=1}^{N-1} (2/\rho)\langle \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star, (\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star) - (\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star) \rangle + (1/\rho) \left\| (\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star) - (\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star) \right\|^2 + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2 \\
&= \sum_{n=1}^{N-1} (2/\rho)\langle \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star, \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star \rangle - (2/\rho) \left\| \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star \right\|^2 + (1/\rho) \left\| \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star \right\|^2 \\
&\qquad - (2/\rho)\langle \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star, \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star \rangle + 1/\rho \left\| \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star \right\|^2 + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2 \qquad (115) \\
&= \sum_{n=1}^{N-1} \Big[(1/\rho) \left\| \boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^\star \right\|^2 - (1/\rho) \left\| \boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^\star \right\|^2 + \rho \left\| \mathbf{r}_{n,n_{r,k}}^{k+1} \right\|^2 \Big]. \qquad (116)
\end{aligned}$$

Next, consider the second term on the left hand side of (112), from the equality (97), it holds that

$$\begin{aligned}
&2\sum_{n\in\mathcal{N}_h^k}\langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^\star\rangle \\
&\quad= \sum_{n\in\mathcal{N}_h\setminus\{1\}} \Big(2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^\star\rangle\Big) + \sum_{n\in\mathcal{N}_h}\Big(2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^\star\rangle\Big).
\end{aligned} \tag{117}$$

Note that $\boldsymbol{\theta}_n^{k+1} = -\mathbf{r}_{n_{l,k},n}^{k+1} + \boldsymbol{\theta}_{n_{l,k}}^{k+1} = \mathbf{r}_{n,n_{r,k}}^{k+1} + \boldsymbol{\theta}_{n_{r,k}}^{k+1}, \forall n = \{2, \cdots, N-1\}$, which implies that we can rewrite the equation in (117) as follows

$$\begin{aligned}
&2\sum_{n\in\mathcal{N}_h^k}\langle \mathbf{s}_n^{k+1}, \boldsymbol{\theta}_n^{k+1}-\boldsymbol{\theta}^\star\rangle \\
&\quad= \sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \mathbf{r}_{n_{l,k},n}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}^\star\rangle\Big) \\
&\qquad\quad + \sum_{n\in\mathcal{N}_h}\Big(2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \mathbf{r}_{n,n_{r,k}}^{k+1}\rangle + 2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}^\star\rangle\Big).
\end{aligned} \tag{118}$$

Using the equalities,

$$\begin{aligned}
\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}^\star =&(\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}) + (\boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star), \forall n\in\mathcal{N}_h^k\setminus\{1\} \\
\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}^\star =&(\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}) + (\boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}^\star), \forall n\in\mathcal{N}_h^k
\end{aligned} \tag{119}$$

we rewrite the right hand side of (118) as

$$\begin{aligned}
&\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \mathbf{r}_{n_{l,k},n}^{k+1}\rangle + 2\rho\|\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}\|^2 + 2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star\rangle\Big) \\
&\quad + \sum_{n\in\mathcal{N}_h^k}\Big(2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \mathbf{r}_{n,n_{r,k}}^{k+1}\rangle + 2\rho\|\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}\|^2 + 2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}_{n_{r,k}}^\star\rangle\Big).
\end{aligned} \tag{120}$$

Further using the equalities

$$\begin{aligned}
\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k} =&(\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star), \forall n\in\mathcal{N}_h^k\setminus\{1\} \\
\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k} =&(\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}^\star), \forall n\in\mathcal{N}_h^k,
\end{aligned} \tag{121}$$

we can write (120) after the rearrangement as

$$\begin{aligned}
&\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}, \mathbf{r}_{n-1,n}^{k+1}\rangle + \rho\|\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}_{n_{l,k}}^{k}\|^2 + \rho\|(\boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star)\|^2 \\
&\qquad + 2\rho\langle \boldsymbol{\theta}_{n_{l,k}}^{k+1}-\boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star\rangle - 2\rho \parallel \boldsymbol{\theta}_{n_{l,k}}^{k}-\boldsymbol{\theta}^\star \parallel_2^2\Big) + \sum_{n\in\mathcal{N}_h^k}\Big(2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}, \mathbf{r}_{n,n_{r,k}}^{k+1}\rangle \\
&\qquad + \rho\|\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}_{n_{r,k}}^{k}\|^2 + \rho\|(\boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}^\star) - (\boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}^\star)\|^2 \\
&\qquad + 2\rho\langle \boldsymbol{\theta}_{n_{r,k}}^{k+1}-\boldsymbol{\theta}^\star, \boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}^\star\rangle - 2\rho\|\boldsymbol{\theta}_{n_{r,k}}^{k}-\boldsymbol{\theta}^\star|^2\Big).
\end{aligned} \tag{122}$$

Next, expanding the square terms in (122), we get the upper bound for the term in (116) as follows

$$\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}, \mathbf{r}^{k+1}_{n-1,n}\rangle + \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\|^2 + \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2 - \rho\|\boldsymbol{\theta}^{k}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2\Big)$$
$$+ \sum_{n\in\mathcal{N}_h^k} \Big(2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}, \mathbf{r}^{k+1}_{n,n_{r,k}}\rangle + \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\|^2 + \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{\star}\|^2 - \rho\|\boldsymbol{\theta}^{k}_{n_{r,k}} - \boldsymbol{\theta}^{\star}\|^2\Big). \tag{123}$$

Substituting the equalities from (116) and (123) to the left hand side of (112), we obtain

$$\sum_{n=1}^{N-1}\Big[-(1/\rho)\|\boldsymbol{\lambda}_n^{k+1} - \boldsymbol{\lambda}_n^{\star}\|^2 + (1/\rho)\|\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^{\star}\|^2 - \rho\|\mathbf{r}^{k+1}_{n,n_{r,k}}\|^2\Big]$$
$$-\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \Big(-2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}, \mathbf{r}^{k+1}_{n_{l,k},n}\rangle + \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\|^2 + \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2 - \rho\|\boldsymbol{\theta}^{k}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2\Big)$$
$$-\sum_{n\in\mathcal{N}_h^k} \Big(2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}, \mathbf{r}^{k+1}_{n,n_{r,k}}\rangle + \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\|^2 + \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{\star}\|^2 - \rho\|\boldsymbol{\theta}^{k}_{n_{r,k}} - \boldsymbol{\theta}^{\star}\|^2\Big)$$
$$\geq 0, \tag{124}$$

Next, consider the Lyapunov function V_k as

$$V_k = 1/\rho \sum_{n=1}^{N-1} \|\boldsymbol{\lambda}_n^{k} - \boldsymbol{\lambda}_n^{\star}\|^2 + \rho \sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \|\boldsymbol{\theta}^{k}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2 + \rho \sum_{n\in\mathcal{N}_h^k} \|\boldsymbol{\theta}^{k}_{n_{l,k}} - \boldsymbol{\theta}^{\star}\|^2. \tag{125}$$

After rearranging the terms in (124) and using the definition of the Lyapunov function in (125), we get

$$V_{k+1} \leq V_k - \sum_{n=1}^{N-1} \rho\|\mathbf{r}^{k+1}_{n,n_{r,k}}\|^2 - \Big[\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\|^2 + \sum_{n\in\mathcal{N}_h^k} \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\|^2\Big]$$
$$-\Big[\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} -2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}, \mathbf{r}^{k+1}_{n_{l,k},n}\rangle + \sum_{n\in\mathcal{N}_h^k} 2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}, \mathbf{r}^{k+1}_{n,n_{r,k}}\rangle\Big]. \tag{126}$$

We rewrite (126) as

$$V_{k+1} \leq V_k - \sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \rho\|\mathbf{r}^{k+1}_{n_l,n}\|^2 + \sum_{n\in\mathcal{N}_h^k} \rho\|\mathbf{r}^{k+1}_{n,n_r}\|^2 - \Big[\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} \rho\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\|^2 + \sum_{n\in\mathcal{N}_h^k} \rho\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\|^2\Big]$$
$$-\Big[\sum_{n\in\mathcal{N}_h^k\setminus\{1\}} -2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}, \mathbf{r}^{k+1}_{n_{l,k},n}\rangle + \sum_{n\in\mathcal{N}_h^k} 2\rho\langle \boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}, \mathbf{r}^{k+1}_{n,n_{r,k}}\rangle\Big]. \tag{127}$$

Next, the equation in (127) can be re-written as

$$\begin{aligned} V_{k+1} \leq V_k - \rho \sum_{n \in \mathcal{N}_h^k \setminus \{1\}} & \left[\|\mathbf{r}^{k+1}_{n_l,n}\|^2 - 2\langle \boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}, \mathbf{r}^{k+1}_{n_{l,k},n} \rangle + \|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\|^2 \right] \\ - \rho \sum_{n \in \mathcal{N}_h^k} & \left[\|\mathbf{r}^{k+1}_{n,n_r}\|^2 + 2\langle \boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}, \mathbf{r}^{k+1}_{n,n_{r,k}} \rangle + \|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\|^2 \right] \end{aligned} \tag{128}$$

Further, we write (128) as

$$V_{k+1} \leq V_k - \rho \left(\sum_{n \in \mathcal{N}_h^k \setminus \{1\}} \|\mathbf{r}^{k+1}_{n_l,n} - (\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}})\|^2 + \sum_{n \in \mathcal{N}_h^k} \|\mathbf{r}^{k+1}_{n,n_r} + (\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}})\|^2 \right) \tag{129}$$

The result in (129) proves that V_{k+1} decreases in each iteration k. Now, since $V_k \geq 0$ and $V_k \leq V_0$, it holds that $\left[\sum_{n \in \mathcal{N}_h^k \setminus \{1\}} \|\mathbf{r}^{k+1}_{n_l,n} - (\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}})\|^2 + \sum_{n \in \mathcal{N}_h^k} \|\mathbf{r}^{k+1}_{n,n_r} + (\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}})\|^2 \right]$ is bounded. Taking the telescopic sum over k in (129) and taking limit $K \to \infty$, we get

$$\lim_{K \to \infty} \sum_{k=0}^{K} \left[\sum_{n \in \mathcal{N}_h^k \setminus \{1\}} \|\mathbf{r}^{k+1}_{n_l,n} - (\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}})\|^2 + \sum_{n \in \mathcal{N}_h^k} \|\mathbf{r}^{k+1}_{n,n_r} + (\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}})\|^2 \right] \leq V_0. \tag{130}$$

The result in (130) implies that the primal residual $\mathbf{r}^{k+1}_{n,n_{r,k}} \to \mathbf{0}$ as $k \to \infty$ for all $n \in \{1, \cdots, N-1\}$. Similarly, the norm differences $\left\|\boldsymbol{\theta}^{k+1}_{n_{l,k}} - \boldsymbol{\theta}^{k}_{n_{l,k}}\right\|$ and $\left\|\boldsymbol{\theta}^{k+1}_{n_{r,k}} - \boldsymbol{\theta}^{k}_{n_{r,k}}\right\| \to \mathbf{0}$ as $k \to \infty$ which implies that the dual residual $\mathbf{s}^k_n \to \mathbf{0}$ as $k \to \infty$ for all $n \in \mathcal{N}_h^k$. In order to prove the convergence to optimal point, , consider the lower and the upper bounds on the objective function optimality gap given by

$$\sum_{n=1}^{N} [f_n(\boldsymbol{\theta}^{k+1}_n) - f_n(\boldsymbol{\theta}^\star)] \leq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}^{k+1}_n, \mathbf{r}^{k+1}_{n,n_{r,k}} \rangle + \sum_{n \in \mathcal{N}_h^k} \langle \mathbf{s}^{k+1}_n, \boldsymbol{\theta}^\star - \boldsymbol{\theta}^{k+1}_n \rangle \tag{131}$$

$$\sum_{n=1}^{N} [f_n(\boldsymbol{\theta}^{k+1}_n) - f_n(\boldsymbol{\theta}^\star)] \geq - \sum_{n=1}^{N-1} \langle \boldsymbol{\lambda}^{\star}_n, \mathbf{r}^{k+1}_{n,n_{r,k}} \rangle. \tag{132}$$

Note that from the results established in this appendix, it holds that the right hand side of the upper bound in (131) converge to zero as $k \to \infty$ and also the right hand side of the lower bound in (132) converges to zero as $k \to \infty$. This implies that

$$\lim_{k \to \infty} \sum_{n=1}^{N} [f_n(\boldsymbol{\theta}^{k+1}_n) - f_n(\boldsymbol{\theta}^\star)] = 0 \tag{133}$$

which is the required result. Hence proved.

References

Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. *In Proceedings of World Wide Web, Rio de Janeiro, Brazil*, May 2013.

Amrit Singh Bedi, Alec Koppel, and Rajawat Ketan. Asynchronous saddle point algorithm for stochastic optimization in heterogeneous networks. *IEEE Transactions on Signal Processing*, 67(7):1742–1757, 2019. ISSN 1053-587X. doi: 10.1109/TSP.2019.2894803.

Doron Blatt, Alfred O Hero, and Hillel Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

Ernesto Bonomi and Jean-Luc Lutton. The n-city travelling salesman problem: Statistical mechanics and the metropolis algorithm. *SIAM review*, 26(4):551–568, 1984.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Tsung-Hui Chang, Mingyi Hong, and Xiangfeng Wang. Multi-agent distributed optimization via inexact consensus admm. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2014a.

Tsung-Hui Chang, Angelia Nedić, and Anna Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automation and Control*, 59(6):1524–1538, 2014b.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. *Advances in Neural Information Processing Systems*, 31:5055–5065, 2018.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. *Advances in Neural Information Processing Systems*, 25:1223–1231, 2012.

Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multi-block admm with $o(1/k)$ convergence. *Journal of Scientific Computing*, 71(2):712–736, 2017.

Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

Daniel Gabay and Bertrand Mercier. *A dual algorithm for the solution of non linear variational problems via finite element approximation*. Institut de recherche d'informatique et d'automatique, 1975.

Roland Glowinski and A Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

Mert Gurbuzbalaban, Asuman Ozdaglar, and Pablo A Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2): 1035–1048, 2017.

Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In *Advances in Neural Information Processing Systems*, pages 4536–4546, 2018.

Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. *Advances in Neural Information Processing Systems*, 27:3068–3076, 2014.

Dušan Jakovetić, Joao Xavier, and José MF Moura. Fast distributed gradient methods. *IEEE Transactions on Automation and Control Automa. Control*, 59(5):1131–1146, 2014.

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *presented at Neural Information Processing Systems Workshop on Machine Learning on the Phone and other Consumer Devices (MLPCD), Montréal, Canada*, 2018. doi: arXiv:1811.11479. URL `http://arxiv.org/abs/1811.11479`.

Michael I. Jordan, Jason D. Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 2018.

Alec Koppel, Brian M Sadler, and Alejandro Ribeiro. Proximity without consensus in online multiagent optimization. *IEEE Transactions on Signal Processing*, 65(12):3062–3077, 2017.

Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pages 1–48, 2017.

Jan Karel Lenstra and AHG Rinnooy Kan. Some simple applications of the travelling salesman problem. *Journal of the Operational Research Society*, 26(4):717–733, 1975.

Mu Li, David G Andersen, and Alexander Smola. Distributed delayed proximal gradient methods. *presented at Neural Information Processing Systems Workshop on Optimization for Machine Learning, Lake Tahoe, NV, USA*, December 2013.

Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. *Advances in Neural Information Processing Systems*, 27:19–27, 2014.

Yaohua Liu, Wei Xu, Gang Wu, Zhi Tian, and Qing Ling. Communication-censored ADMM for decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 67 (10):2565–2579, 2019.

Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

Ian W McKeague. On the capacity of channels with gaussian and non-gaussian noise. *Information and Control*, 51(2):153–173, 1981.

H. Brendan McMahan, Ramage Daniel Moore, Eider, Seth Hampson, and Blaise Agüera yArcas. Communication-efficient learning of deep networks from decentralized data. *In Proceedings of Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA*, April 2017.

Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Trans. Automa. Control*, 60(3):601–615, 2014.

Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automation and Control*, 54(1):48–61, 2009.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5): 953–976, 2018.

Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. Wireless network intelligence at the edge. *to appear in Proceedings of the IEEE [Online]. Early access is available at: https://ieeexplore.ieee.org/document/8865093*, November 2019.

Carsten Peterson. Parallel distributed approaches to combinatorial optimization: benchmark studies on traveling salesman problem. *Neural computation*, 2(3):261–269, 1990.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, 2015.

Nandan Sriranga, Chandra R. Murthy, and Vaneet Aggarwal. A method to improve consensus averaging using quantized admm. In *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019.

Ananda Theertha Suresh, Felix X Yu, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. *Proceedings of Machine Learning Research*, 70:3329–3337, 2017.

Behrouz Touri and Angelia Nedic. Distributed consensus over network with noisy links. *In Proceedings of International Conference on Information Fusion, Seattle, WA, USA*, July 2009.

Konstantinos I. Tsianos, Sean Lawlor, and Michael G. Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. *In Proceedings of Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA*, October 2012.

Huahua Wang, Arindam Banerjee, and Zhi-Quan Luo. Parallel direction method of multipliers. *Advances in Neural Information Processing Systems*, 27:181–189.

Huihui Wang, Yang Gao, Yinghuan Shi, and Ruili Wang. Group-based alternating direction method of multipliers for distributed linear classification. *IEEE Transactions on Cybernetics*, 47(11):3568–3582, 2017.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *ArXiv preprint*, abs/1804.05271, 2018.

Yuchen Zhang, Martin J Wainwright, and John C Duchi. Communication-efficient algorithms for statistical optimization. *Advances in Neural Information Processing Systems*, 25:1502–1510, 2012.

Shengyu Zhu, Mingyi Hong, and Biao Chen. Quantized consensus ADMM for multi-agent distributed optimization. *In Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Shanghai, China*, March 2016.