An Accelerated Asynchronous Distributed Method for Convex Constrained Optimization Problems

Nazanin Abolfazli
Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721
nazaninabolfazli@arizona.edu

Afrooz Jalilzadeh
Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721
afrooz@arizona.edu

Erfan Yazdandoost Hamedani
Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721
erfany@arizona.edu

arXiv:2302.11040v1 [math.OC] 21 Feb 2023

***Abstract*—We consider a class of multi-agent cooperative consensus optimization problems with local nonlinear convex constraints where only those agents connected by an edge can directly communicate, hence, the optimal consensus decision lies in the intersection of these private sets. We develop an asynchronous distributed accelerated primal-dual algorithm to solve the considered problem. The proposed scheme is the first asynchronous method with an optimal convergence guarantee for this class of problems, to the best of our knowledge. In particular, we provide an optimal convergence rate of $\mathcal{O}(1/K)$ for suboptimality, infeasibility, and consensus violation.**

***Index Terms*—Multi-agent distributed optimization, asynchronous algorithm, constrained optimization, convergence rate**

I. Introduction

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ denote a connected undirected graph of N computing nodes where $\mathcal{N} \triangleq \{1, \ldots, N\}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the set of edges. We consider the following constrained optimization problem over network $\mathcal{G}$:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \sum_{i\in\mathcal{N}} \varphi_i(x) \triangleq f_i(x) + \rho_i(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \ i \in \mathcal{N}, \end{aligned} \tag{1}$$

where x denotes the global decision variable; $\rho_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a possibly *non-smooth* convex function with easy-to-compute proximal map; $f_i : \mathbb{R}^n \to \mathbb{R}$ is a *smooth* convex function; and $g_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ is a vector-valued convex function. We assume that each agent $i \in \mathcal{N}$ has only access to local information, i.e., f_i, ρ_i, and g_i. Our objective is to develop an efficient algorithm with a convergence guarantee for solving (1) in a decentralized fashion using the computing nodes $\mathcal{N}$ and exchanging information only along the edges $\mathcal{E}$.

Decentralized optimization over communication networks has various applications. Here we discuss a few applications. 1) In *multi-agent control design*, consider computing an optimal consensus decision satisfying the constraint of each agent $i \in \mathcal{N}$, involving some *uncertain parameter* $q_i \in \mathbb{R}^n$, with at least $1 - \epsilon$ probability, i.e., $\min_x\{\sum_{i\in\mathcal{N}} f_i(x) \mid \mathbb{P}(\{q_i : q_i^\top x \leq b_i\}) \geq 1-\epsilon, i \in \mathcal{N}\}$. This problem can be formulated as a minimization over an intersection of ellipsoids under a particular distribution of the uncertain parameters [1]. 2) In *multi-agent localization in sensor networks* one needs to collaboratively locate a target $\bar{x} \in \mathbb{R}^n$. Suppose each agent $i \in \mathcal{N}$ has a directional sensor that can detect a target when it belongs to $\mathcal{X}_i = \{y_i : \ \|A_i y_i - b_i\|_2 \leq \eta_i\}$. Let $\mathcal{I} = \{i \in \mathcal{N} : \ \bar{x} \in \mathcal{X}_i\}$. Therefore, the target location $\bar{x}$ can be estimated by solving $\min_x\{\|x\|_2 : \ x \in \cap_{i\in\mathcal{I}}\mathcal{X}_i\}$. 3) In *distributed robust optimization*, the goal is to solve $\min_{x\in X}\{\sum_{i\in\mathcal{N}} f_i(x) \mid g_i(x,q) \leq 0, \ \forall q \in Q, \ \forall i \in \mathcal{N}\}$ where Q represents an uncertainty set. Using a scenario-based approach [2], this problem can be reformulated as (1).

Recently, there have been many studies on developing distributed algorithms for convex constrained consensus optimization problems subject to (non)linear constraints with a convergence rate guarantee [3]–[5]. However, the update of such algorithms is in a synchronous fashion which requires access to a global clock, thus largely limiting their applicability. Due to the lack of such an assumption, i.e, an agent has to work based on its own clock, the development of asynchronous algorithms is of prime importance. Additionally, in many applications, networks are vulnerable to certain possible link failures and some agents may not implement any operation at a certain time instant. Therefore, asynchronous implementation is crucial in process of communication and computation. Next, we briefly summarize the related research, and then we state our contributions.

A. Literature Review

In the past few years, numerous research studies have been conducted concerning distributed optimization methods. In the absence of constraints, various methods under both static and time-varying communication network have been studied, such as [6]–[8] to name a few. For convex optimization problems with easy-to-project local constraints, in [9] authors introduced a distributed random projection algorithm, that can be employed by multiple agents connected over a time-varying network, while a proximal minimization perspective is proposed in [10]. In a time-varying setting, [11] suggested a projected subgradient method to solve distributed convex optimization problems. For minimizing multi-agent convex optimization problems with linearly coupled constraints over networks, the author in [12] proposed (primal) randomized

The first and third authors would like to acknowledge support from NSF ECCS-2127696.

block-coordinate descent methods. Moreover, there have been several methods proposed considering a nonconvex objective function subject to easy-to-project or linear constraints such as [13]–[17].

Recently primal-dual methods have become a popular approach for solving distributed optimization problems (see e.g., [3], [18]). In particular, Alternating Directions Method of Multipliers (ADMM) is the basis of many effective distributed algorithms such as those presented in [19]–[21]. Due to the fact that the primal variables must achieve successive minimizations at each iteration, distributed ADMM is computationally expensive [21]. To address this issue, the linearized ADMM algorithms have been presented in [21] and [22].

Although the conventional implementation of distributed algorithms requires synchronous communication between agents, distributed systems may commonly use asynchronous communication networks between nodes. Various asynchronous versions of distributed optimization algorithms have been studied in the literature [23]–[27] with a convergence rate guarantee focusing on easy-to-project or linear constraint sets. However, there are far fewer studies considering general nonlinear constraints. In particular, very recently there have been few studies focusing on developing ADMM-based methods for solving (1) (with possibly nonconvex functions) such as [28]–[30] where the agent's updates are asynchronous. However, none of these methods present a convergence rate guarantee for their algorithms.

Next, we outline the contributions of our paper.

II. Contributions

We consider a class of distributed optimization subject to agent-specific nonlinear constraints. We propose an accelerated primal-dual algorithm with asynchronous updates where each agent has to work based on its own clock and does not need to have access to the global clock. By assuming a composite convex structure on the primal functions and convex constraints, we show that our proposed algorithm converges to an optimal solution at a rate of $\mathcal{O}(1/K)$ in terms of suboptimality, infeasibility, and consensus violation. To the best of our knowledge, the proposed scheme is the first asynchronous method with a momentum acceleration that achieves the optimal convergence rate for the considered setting.

Organization of the paper. In Section III, we provide the main assumptions and definitions, required for the convergence analysis. Next, in Sections IV and V, we introduce AD-APD method and show the convergence rate of $\mathcal{O}(1/K)$ for both suboptimality and infeasibility. Finally, in Section VI we compare the performance of the proposed algorithm with a competitive scheme.

III. Preliminaries

Consider problem (1), where f_i, g_i and ρ_i satisfy the following assumption for any $i \in \mathcal{N}$.

Assumption III.1. *For each $i \in \mathcal{N}$, (i) f_i is differentiable on an open set containing $\mathbf{dom}(\rho_i)$ with a Lipschitz continuous gradient ∇f_i and Lipschitz constant L_i^f. (ii) g_i is differentiable with Lipschitz continuous Jacobian matrix $\mathbf{J}g_i \in \mathbb{R}^{n\times m_i}$ with constant L_i^g. (iii) $\mathbf{dom}(\rho_i) \triangleq \{x \in \mathbb{R}^n \mid \rho_i < \infty\}$ is bounded.*

Before discussing our communication network and the related assumptions, first, we introduce important notations.

A. Notations

Throughout the paper, $\|.\|$ denotes the Euclidean norm. Given a convex function f, let $\mathrm{prox}_f(w) \triangleq \mathrm{argmin}_v\{f(v) + \frac{1}{2}\|v-w\|^2\}$ denote the proximal operator of function f. Let $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denote the identity matrix and $\mathbf{1}_n \in \mathbb{R}^n$ denote the vector of ones. Let $\otimes$ denote Kronecker product and $[x]_+ \triangleq \max\{0, x\}$. For any matrix $A \in \mathbb{R}^{n\times m}$, $A_{i:} \in \mathbb{R}^{1\times m}$ and $A_{:j} \in \mathbb{R}^{n\times 1}$ denotes i-th row and j-th column of A, respectively, and a_{ij} denotes row i and column j of matrix A. For any set of vectors $\{x_i\}_{i\in\mathcal{N}} \subset \mathbb{R}^n$, $\mathbf{x} = [x_i]_{i\in\mathcal{N}} \in \mathbb{R}^{nN}$ denotes the concatenation of those vectors. Moreover, for a given set of matrices $A_i \in \mathbb{R}^{n_i\times m_i}$, for $i \in \mathcal{N}$, $\mathbf{diag}([A_i]_{i\in\mathcal{N}}) \in \mathbb{R}^{n\times m}$ denotes a block diagonal matrix whose diagonal blocks are A_i's where $(n, m) = \sum_{i\in\mathcal{N}}(n_i, m_i)$.

Remark III.1. *Based on Assumption III.1, the boundedness of the domain implies that for any $i \in \mathcal{N}$, g_i is a Lipschitz continuous function and we denote the constant with C_i.*

Next, we define some notations based on the constants introduced in Assumption III.1 and Remark III.1.

Definition 1. *Given a set of parameters τ_i, γ_i, and σ_i for $i \in \mathcal{N}$, let $\mathcal{T} \triangleq \mathbf{diag}([\frac{1}{\tau_i}\mathbf{I}_n]_{i\in\mathcal{N}})$, $\mathcal{S} \triangleq \mathbf{diag}([\frac{1}{\sigma_i}\mathbf{I}_{m_i}]_{i\in\mathcal{N}})$, $\Gamma \triangleq \mathbf{diag}([\frac{1}{\gamma_i}\mathbf{I}_n]_{i\in\mathcal{N}})$, and $\mathcal{B} \triangleq \mathbf{diag}(\mathcal{S}, \Gamma)$. Moreover, we define $\mathbf{C} \triangleq \mathbf{diag}([C_i\mathbf{I}_{m_i}]_{i\in\mathcal{N}})$, $\mathbf{D} \triangleq \mathbf{diag}([(C_i+\delta_i)\mathbf{I}_n]_{i\in\mathcal{N}})$, and $\Delta \triangleq \mathbf{diag}([\delta_i\mathbf{I}_n]_{i\in\mathcal{N}})$.*

Definition 2. *Let $\varphi(\mathbf{x}) \triangleq \sum_{i\in\mathcal{N}}\varphi_i(x_i) : \mathbb{R}^{nN} \to \mathbb{R}$ and $g(\mathbf{x}) \triangleq [g_i(x_i)]_{i\in\mathcal{N}} : \mathbb{R}^{nN} \to \mathbb{R}^m$ where $m \triangleq \sum_{i\in\mathcal{N}} m_i$.*

B. Communication network

We consider a multi-agent system where the agents combine their own information state with those received from their neighbors to update their state. Suppose nodes i and j can exchange information only if $(i,j) \in \mathcal{E}$ or $(j,i) \in \mathcal{E}$, and each node $i \in \mathcal{N}$ has a *private* (local) cost function φ_i and constraint function g_i. The set of neighboring nodes of agent i is denoted by $\mathcal{N}_i \triangleq \{j \in \mathcal{N} \mid (i,j) \in \mathcal{E}$ or $(j,i) \in \mathcal{E}\}$. The weighted matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$ is a nonnegative matrix such that $w_{ij} > 0$ if $j \in \mathcal{N}_i$ and $w_{ij} = 0$ otherwise.

We assume that each node $i \in \mathcal{N}$ has a local clock $t_i \in \mathbb{R}_+$ and a randomly generated waiting time T_i. Each node i will remain *idle* while $\tau_i < T_i$ and switches to the *awake* mode when $\tau_i = T_i$ after which it runs the local computations, resets $t_i = 0$ and draws a new realization of the random variable T_i. Formally, we make the following assumptions on the communication architecture.

Assumption III.2. *The waiting times T_i between consecutive events are i.i.d. random variables with the same exponential* *distribution. Moreover, only one node can be awake at each time instant.*

C. Problem Reformulation

Let $x_i \in \mathbb{R}^n$ denote the local decision vector of node $i \in \mathcal{N}$. We can reformulate (1) as $\min_{\mathbf{x}}\{\varphi(\mathbf{x}) \mid g(\mathbf{x}) \leq 0,\ x_i = x_j\ \forall(i,j) \in \mathcal{E}\}$. Furthermore, we can describe the consensus constraint as a linear constraint, i.e., $(V \otimes \mathbf{I}_n)\mathbf{x} = \mathbf{0}$ for some $V \in \mathbb{R}^{N\times N}$. We consider the following condition on the consensus constraint matrix V.

Assumption III.3. *For any $\mathbf{x} \in \mathbb{R}^{nN}$, $(V \otimes \mathbf{I}_n)\mathbf{x} = \mathbf{0}$ if and only if there exists $x \in \mathbb{R}^n$ such that $\mathbf{x} = \mathbf{1}_N \otimes x$. Moreover, for any $i \in \mathcal{N}$, there exists $\delta_i > 0$ such that $\|V_{i:}\|_1 \leq \delta_i$.*

Remark III.2. *For any mixing matrix W with eigenvalues in $(-1, 1]$, e.g., Laplacian-based and Metropolis mixing matrices, let $V = \alpha(\mathbf{I}_N - W)$ for any $\alpha > 0$. It implies that Assumption III.3 is satisfied with $\delta_i = 2\alpha(1 - w_{ii})$.*

Furthermore, we consider the following standard regularity assumption on problem 1.

Assumption III.4. *The duality gap for (1) is zero, and a primal-dual solution $(x^*, \mathbf{y}^*)$ to (1) exists. Moreover, the dual solution is bounded, i.e., $\exists B > 0$ such that $\|\mathbf{y}^*\| \leq B$.*

Remark III.3. *Note that Assumption III.4 holds in practice under mild conditions as it is studied in [31]. For instance, when Slater condition holds the agents can collectively compute a Slater point and use it to find a dual bound in a distributed manner.*

Now using Lagrangian duality, we equivalently write the following saddle point formulation.

$$\min_{\mathbf{x}\in\mathbb{R}^{nN}} \max_{\substack{\mathbf{y}\in\mathbb{R}^m_+ \\ \boldsymbol{\lambda}\in\mathbb{R}^{nN}}} \mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) \triangleq \varphi(\mathbf{x}) + \langle g(\mathbf{x}),\mathbf{y}\rangle + \langle \boldsymbol{\lambda}, \mathbf{V}\mathbf{x}\rangle \quad (2)$$

where $\mathbf{V} \triangleq V \otimes \mathbf{I}_n$. Next, we develop a distributed primal-dual method with convergence guarantee for solving (2).

IV. Proposed Method

In this section, we study the asynchronous distributed implementation of the accelerated primal-dual (APD) algorithm to solve (2). We propose an asynchronous distributed accelerated primal-dual (AD-APD) algorithm whose iterations can be computed in a decentralized way, via the node-specific computations as in Algorithm 1. In particular, at each iteration, one agent goes to "awake" mode uniformly at random and updates its local decision variables by taking dual accent steps with momentum accelerations following a proximal-gradient descent using the most updated dual decision variables. Moreover, each agent combines the local information with its neighbors using the consensus constraint matrix. Moreover, our proposed method includes a new linear combination of dual gradient iterates that can recover APD for $N = 1$.

Algorithm 1 Asynchronous Distributed Accelerated Primal-Dual Algorithm (AD-APD)

Input: $[\tau_i, \sigma_i, \gamma_i]_{i\in\mathcal{N}}$, $(\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0) \in \mathbb{R}^{nN} \times \mathbb{R}^m \times \mathbb{R}^{nN}$

For $k \geq 0$,

`IDLE`:

while $t_i < T_i$ **do**
  Do Nothing
end while
Go to `AWAKE`

`AWAKE`:

Receive $\lambda_j^k, x_j^k, x_j^{k-1}$ from neighbors $(j \in \mathcal{N}_i)$

$y_i^{k+1} \leftarrow \max\{\mathbf{0}, y_i^k + 2N\sigma_i\big(g_i(x_i^k) - \frac{(2N-1)}{2N} g_i(x_i^{k-1})\big)\}$

$\lambda_i^{k+1} \leftarrow \lambda_i^k + \gamma_i \sum_{j\in\mathcal{N}_i\cup\{i\}} v_{ij}(2Nx_j^k - (2N-1)x_j^{k-1})$

$x_i^{k+1} \leftarrow \mathrm{prox}_{\tau_i f_i}\big(x_i^k - \tau_i\big(\mathbf{J}g_i(x_i^k)^\top y_i^{k+1} + v_{ii}\lambda_i^{k+1} + \sum_{j\in\mathcal{N}_i} v_{ij}\lambda_j^k\big)\big)$

Send $\lambda_i^{k+1}, x_i^{k+1}, x_i^k$ to neighbors

Set $t_i = 0$, get new realization T_i and go to `IDLE`

V. Convergence Analysis

In the following theorem, we state the convergence rate of AD-APD in terms of the Lagrangian error metric, and then in Corollary V.1.1, the convergence rate in terms of the suboptimality, infeasibility, and consensus violation is shown.

Theorem V.1. *Suppose Assumptions III.1 - III.4 hold and $\{\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k\}_{k\geq 0}$ is the sequence generated by AD-APD stated in Algorithm 1 with step-sizes selected such that $\tau_i \leq \frac{1}{2(C_i+\delta_i)+L_i^f+BL_i^g}$, $\sigma_i \leq \frac{1}{3C_i}$ and $\gamma_i \leq \frac{1}{3\delta_i}$. Then it holds for any $(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) \in \mathbb{R}^{nN} \times \mathbb{R}^m_+ \times \mathbb{R}^{nN}$ and $K \geq 1$ that*

$$\begin{aligned}&\mathbb{E}\big[\mathcal{L}(\bar{\mathbf{x}}^K,\mathbf{y},\boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x},\bar{\mathbf{y}}^K,\bar{\boldsymbol{\lambda}}^K)\big] \\ &\leq \frac{N}{2(K+N-1)}\Big(\big\|\mathbf{x}^0-\mathbf{x}\big\|^2_{\mathcal{T}+\mathbf{D}} + \big\|\mathbf{y}^0-\mathbf{y}\big\|^2_{\mathcal{S}+\mathbf{C}} \\ &\quad + \big\|\boldsymbol{\lambda}^0-\boldsymbol{\lambda}\big\|^2_{\Gamma+\Delta} + \tfrac{N-1}{N}\big(\mathcal{L}(\mathbf{x}^0,\mathbf{y},\boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x},\mathbf{y}^0,\boldsymbol{\lambda}^0)\big)\Big),\end{aligned}$$

where $(\bar{\mathbf{x}}^K,\bar{\mathbf{y}}^K,\bar{\boldsymbol{\lambda}}^K) \triangleq \frac{1}{K+N-1}\big(\sum_{k=1}^{K-1}(\mathbf{x}^k,\mathbf{y}^k,\boldsymbol{\lambda}^k) + N(\mathbf{x}^K,\mathbf{y}^K,\boldsymbol{\lambda}^K)\big)$.

Proof. The proof is presented in section V-A. □

Corollary V.1.1. *Under premises of Theorem V.1, for any $K \geq 1$, the following holds.*

$$\begin{aligned}&\big|\varphi(\bar{\mathbf{x}}^K) - \varphi(\mathbf{x}^*)\big| \leq \mathcal{O}\left(\tfrac{N}{K+N-1}\right), \\ &\|\boldsymbol{\lambda}^*\|\,\big\|V\bar{\mathbf{x}}^K\big\| + \sum_{i\in\mathcal{N}} \|y_i^*\|\,\big\|[g_i(\bar{\mathbf{x}}^K)]_+\big\| \leq \mathcal{O}\left(\tfrac{N}{K+N-1}\right).\end{aligned}$$

Proof. The proof follows the same steps as in [32, Corollary 4.2.]. □

Before proving Theorem V.1, we state a standard technical lemma for the proximal gradient step which is a trivial extension of Property 1 in [33]. Then we provide a one-step analysis of the algorithm in Lemma V.3.

Lemma V.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed convex function. Given $\bar{x} \in \mathbf{dom}\, f$ and $t > 0$, let*

$$x^+ = \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + \tfrac{t}{2}\|x - \bar{x}\|^2.$$

Then for all $x \in \mathbb{R}^n$, the following inequality holds:

$$f(x) + \tfrac{t}{2}\|x - \bar{x}\|^2 \geq f(x^+) + \tfrac{t}{2}\|x^+ - \bar{x}\|^2 + \tfrac{t}{2}\|x - x^+\|^2.$$

Before we proceed, we define some notations to facilitate the proof.

Definition 3. *Let function Φ be the smooth part of the objective function in (2), i.e., $\Phi_i(x_i, y_i, \boldsymbol{\lambda}) \triangleq f_i(x_i) + \langle g_i(x_i), y_i\rangle + \langle (V_{i:} \otimes \mathbf{I}_n)\boldsymbol{\lambda}, x_i\rangle$ and $\Phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) \triangleq \sum_{i\in\mathcal{N}} \Phi_i(x_i, y_i, \boldsymbol{\lambda})$. Moreover, we define $\mathbf{z} \triangleq [\mathbf{y}^\top\ \boldsymbol{\lambda}^\top]^\top$.*

Lemma V.3. *Let $\{\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k\}_{k\geq 0}$ be the sequence generated by AD-APD, stated in Algorithm 1. Suppose Assumptions III.1 and III.2 hold and $\mathcal{T}$ and $\mathcal{B}$ are defined in Definition 1. Let $\mathbf{z} \triangleq [\mathbf{y}^\top, \boldsymbol{\lambda}^\top]^\top$, then for any $(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) \in \mathbb{R}^{nN} \times \mathbb{R}^m_+ \times \mathbb{R}^{nN}$,*

$$\begin{aligned}
&\mathbb{E}^k[\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})] \leq (N-1)\big(\mathcal{H}^k \\
&- \mathbb{E}^k[\mathcal{H}^{k+1}]\big) + \langle \mathbf{u}^k, \mathbf{z}^k - \mathbf{z}\rangle - \mathbb{E}^k[\langle \mathbf{u}^{k+1}, \mathbf{z}^{k+1} - \mathbf{z}\rangle] \\
&+ \tfrac{N}{2}\mathbb{E}^k\big[\, \|\mathbf{x} - \mathbf{x}^k\|^2_{\mathcal{T}} - \|\mathbf{x} - \mathbf{x}^{k+1}\|^2_{\mathcal{T}} - \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2_{\mathcal{T}-\mathbf{L}^\Phi} \\
&+ \|\mathbf{x}^{k-1} - \mathbf{x}^k\|^2_{2\mathbf{D}}\,\big] + \tfrac{N}{2}\mathbb{E}^k\big[\, \|\mathbf{z} - \mathbf{z}^k\|^2_{\mathcal{B}} \\
&- \|\mathbf{z} - \mathbf{z}^{k+1}\|^2_{\mathcal{B}} - \|\mathbf{z}^k - \mathbf{z}^{k+1}\|^2_{\widetilde{\mathcal{B}}}\,\big], \qquad (3)
\end{aligned}$$

where $\mathcal{H}^k \triangleq \rho(\mathbf{x}^k) + \Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k)$, $\widetilde{\mathcal{B}} \triangleq \mathcal{B} - 2\mathbf{C}_2/N$, $\mathbf{D} \triangleq \mathbf{diag}([(C_i + \delta_i)\mathbf{I}_n]_{i\in\mathcal{N}})$, $\mathbf{C}_2 \triangleq \mathbf{diag}([C_i\mathbf{I}_{m_i}]_{i\in\mathcal{N}}, [\delta_i\mathbf{I}_n]_{i\in\mathcal{N}})$, and $\mathbf{u}^k \triangleq \nabla_{\mathbf{z}}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) - (2N-1)\nabla_{\mathbf{z}}\Phi(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}, \boldsymbol{\lambda}^{k-1})$.

Proof. We begin the proof by defining auxiliary sequences $\{\tilde{\mathbf{x}}^k, \tilde{\boldsymbol{\lambda}}^k\}_{k\geq 1} \subseteq \mathbb{R}^{nN}$ and $\{\tilde{\mathbf{y}}^k\}_{k\geq 1} \subseteq \mathbb{R}^m$ representing centralized updates and compare them with the sequences generated by the proposed algorithm. Note that these auxiliary sequences are never actually computed in the implementation of the algorithm. In particular, we define the following for all $i \in \mathcal{N}$

$$\tilde{y}_i^{k+1} \triangleq \max\{\mathbf{0}, y_i^k + \sigma_i(\nabla_{y_i}\Phi_i(x_i^k, y_i^k, \boldsymbol{\lambda}^k) + s_i^k)\}, \qquad (4)$$

$$\tilde{\lambda}_i^{k+1} \triangleq \lambda_i^k + \gamma_i(\nabla_{\lambda_i}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) + r_i^k), \qquad (5)$$

$$\tilde{x}_i^{k+1} \triangleq \mathbf{prox}_{\tau_i\rho_i}(x_i^k - \tau_i\nabla_{x_i}\Phi_i(x_i^k, y_i^{k+1}, \boldsymbol{\lambda}^{k+1})), \qquad (6)$$

where $s_i^k \triangleq (2N-1)(\nabla_{y_i}\Phi_i(x_i^k, y_i^k, \boldsymbol{\lambda}^k) - \nabla_{y_i}\Phi_i(x_i^{k-1}, y_i^{k-1}, \boldsymbol{\lambda}^{k-1}))$ and $r_i^k \triangleq (2N-1)(\nabla_{\lambda_i}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) - \nabla_{\lambda_i}\Phi(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}, \boldsymbol{\lambda}^{k-1}))$. Applying Lemma V.2 on (6) implies that

$$\begin{aligned}
&\rho_i(\tilde{x}_i^{k+1}) - \rho_i(x_i) \leq \Big\langle \nabla_{x_i}\Phi_i(x_i^k, y_i^{k+1}, \boldsymbol{\lambda}^{k+1}), x_i - \tilde{x}_i^{k+1}\Big\rangle \\
&+ \tfrac{1}{2\tau_i}\big[\, \|x_i - x_i^k\|^2 - \|x_i - \tilde{x}_i^{k+1}\|^2 - \|x_i^k - \tilde{x}_i^{k+1}\|^2\,\big]. \qquad (7)
\end{aligned}$$

Using Lipschitz continuity of ∇f_i and $\mathbf{J}g_i$ and boundedness of sequence $\{\mathbf{y}^k\}_{k\geq 0}$ we conclude that $\nabla_{x_i}\Phi_i(x_i^k, y_i^{k+1}, \boldsymbol{\lambda}^{k+1})$ is Lipschitz continuous with constant $L_i^\Phi = L_i^f + BL_i^g$. Therefore,

$$\begin{aligned}
&\Big\langle \nabla_{x_i}\Phi_i(x_i^k, y_i^{k+1}, \boldsymbol{\lambda}^{k+1}), x_i - \tilde{x}_i^{k+1}\Big\rangle \leq \Phi_i(x_i, y_i^{k+1}, \boldsymbol{\lambda}^{k+1}) \\
&- \Phi_i(\tilde{x}_i^{k+1}, y_i^{k+1}, \boldsymbol{\lambda}^{k+1}) + \tfrac{L_i^\Phi}{2}\|\tilde{x}_i^{k+1} - x_i^k\|^2. \qquad (8)
\end{aligned}$$

Combining (7) and (8), and summing over $i \in \mathcal{N}$ we obtain

$$\begin{aligned}
&\rho(\tilde{\mathbf{x}}^{k+1}) - \rho(\mathbf{x}) \leq \Phi(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \Phi(\tilde{\mathbf{x}}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}) \\
&+ \tfrac{1}{2}\big[\, \|\mathbf{x} - \mathbf{x}^k\|^2_{\mathcal{T}} - \|\mathbf{x} - \tilde{\mathbf{x}}^{k+1}\|^2_{\mathcal{T}} - \|\mathbf{x}^k - \tilde{\mathbf{x}}^{k+1}\|^2_{\mathcal{T}-\mathbf{L}^\Phi}\,\big],
\end{aligned}$$

where $\mathbf{L}^\Phi \triangleq \mathbf{diag}([L_i^\Phi\mathbf{I}_n]_{i\in\mathcal{N}})$. Note that at each iteration of the algorithm only one agent is awake, i.e., one component of each decision variable is updated, therefore, for any function $\psi : \mathbb{R}^{nN} \to \mathbb{R}$ we have $\mathbb{E}^k[\psi(\mathbf{x}^{k+1})] = \frac{1}{N}\psi(\tilde{\mathbf{x}}^{k+1}) + (1 - \frac{1}{N})\psi(\mathbf{x}^k)$ and one can deduce similar results for $\mathbf{y}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$. Now using this fact and concavity and smoothness of $\Phi(\mathbf{x}, \cdot, \cdot)$ from the last inequality we can conclude that

$$\begin{aligned}
&\mathbb{E}^k[\rho(\mathbf{x}^{k+1}) + \Phi(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}) - \rho(\mathbf{x}) - \Phi(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})] \leq \\
&(N-1)(\rho(\mathbf{x}^k) + \Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) - \mathbb{E}^k[\rho(\mathbf{x}^{k+1}) \\
&+ \Phi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})]) \\
&+ \mathbb{E}^k\Big[\Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}), \mathbf{z} - \mathbf{z}^{k+1}\Big\rangle\Big] + A_1^k \\
&+ (N-1)\mathbb{E}^k\Big[\Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k), \mathbf{z}^{k+1} - \mathbf{z}^k\Big\rangle\Big], \qquad (9)
\end{aligned}$$

where $A_1^k \triangleq \frac{1}{2}\big[\, \|\mathbf{x} - \mathbf{x}^k\|^2_{\mathcal{T}} - \|\mathbf{x} - \tilde{\mathbf{x}}^{k+1}\|^2_{\mathcal{T}} - \|\mathbf{x}^k - \tilde{\mathbf{x}}^{k+1}\|^2_{\mathcal{T}-\mathbf{L}^\Phi}\,\big]$.

Using a similar argument as in (7), one can obtain the following inequalities for the updates in (4) and (5), respectively. Indeed for any $y_i \in \mathbb{R}^{m_i}_+$,

$$\begin{aligned}
&0 \leq \Big\langle \nabla_{y_i}\Phi_i(x_i^k, y_i^k, \boldsymbol{\lambda}^k) + s_i^k, \tilde{y}_i^{k+1} - y_i\Big\rangle \\
&+ \tfrac{1}{2\sigma_i}\big[\, \|y_i - y_i^k\|^2 - \|y_i - \tilde{y}_i^{k+1}\|^2 - \|\tilde{y}_i^{k+1} - y_i^k\|^2\,\big] \qquad (10) \\
&0 = \Big\langle \nabla_{\lambda_i}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) + r_i^k, \tilde{\lambda}_i^{k+1} - \lambda_i\Big\rangle + \tfrac{1}{2\gamma_i}\big[\|\lambda_i - \lambda_i^k\|^2 \\
&- \|\lambda_i - \tilde{\lambda}_i^{k+1}\|^2 - \|\tilde{\lambda}_i^{k+1} - \lambda_i^k\|^2\big]. \qquad (11)
\end{aligned}$$

Next, we sum (10) and (11) over $i \in \mathcal{N}$ and add the resulting inequality to (9). Then, using definition of $\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$ and $\mathcal{H}^k$, and that $\mathbb{E}^k[h(\mathbf{y}^{k+1})] = \frac{1}{N}h(\tilde{\mathbf{y}}^{k+1}) + (1 - \frac{1}{N})h(\mathbf{y}^k)$, we obtain

$$\begin{aligned}
&\mathbb{E}^k[\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})] \leq (N-1)[\mathcal{H}^k \\
&- \mathbb{E}^k[\mathcal{H}^{k+1}]] + \Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1}), \mathbf{z} - \mathbf{z}^{k+1}\Big\rangle \\
&+ \Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k) + \mathbf{q}^k, \tilde{\mathbf{z}}^{k+1} - \mathbf{z}\Big\rangle + A_1^k + A_2^k + A_3^k \\
&+ (N-1)\mathbb{E}^k\Big[\Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k), \mathbf{z}^{k+1} - \mathbf{z}^k\Big\rangle\Big], \qquad (12)
\end{aligned}$$

where $\mathbf{q}^k \triangleq [{\mathbf{s}^k}^\top, {\mathbf{r}^k}^\top]^\top$. Moreover, A_2^k, A_3^k are defined similar to A_1^k as the sum of terms containing norm squares in (10) and (11) over $i \in \mathcal{N}$, respectively. To simplify the notation, we will use $\nabla_{\mathbf{z}}\Phi^k \triangleq \nabla_{\mathbf{z}}\Phi(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k)$. One can easily observe that $\mathbf{q}^k = (2N-1)(\nabla_{\mathbf{z}}\Phi^k - \nabla_{\mathbf{z}}\Phi^{k-1})$. Next, we deal with the two inner product terms in (12) as follows.

$$\begin{aligned}
&\mathbb{E}^k\Big[\left\langle \nabla_{\mathbf{z}}\Phi^{k+1}, \mathbf{z} - \mathbf{z}^{k+1} \right\rangle + \left\langle \nabla_{\mathbf{z}}\Phi^{k} + \mathbf{q}^k, \tilde{\mathbf{z}}^{k+1} - \mathbf{z} \right\rangle \\
&\quad + (N-1) \left\langle \nabla_{\mathbf{z}}\Phi^{k}, \mathbf{z}^{k+1} - \mathbf{z}^k \right\rangle \Big] \\
&= \mathbb{E}^k\Big[\left\langle \nabla_{\mathbf{z}}\Phi^{k+1} - (2N-1)\nabla_{\mathbf{z}}\Phi^{k}, \mathbf{z} - \mathbf{z}^{k+1} \right\rangle \Big] \\
&\quad + \left\langle \mathbf{q}^k - 2(N-1)\nabla_{\mathbf{z}}\Phi^{k}, \mathbf{z}^k - \mathbf{z} \right\rangle + N\mathbb{E}^k[\langle \mathbf{q}^k, \mathbf{z}^{k+1} - \mathbf{z}^k \rangle] \\
&= \left\langle \mathbf{u}^k, \mathbf{z}^k - \mathbf{z} \right\rangle - \mathbb{E}^k[\langle \mathbf{u}^{k+1}, \mathbf{z}^{k+1} - \mathbf{z} \rangle] \\
&\quad + N\mathbb{E}^k[\langle \mathbf{q}^k, \mathbf{z}^{k+1} - \mathbf{z}^k \rangle], \qquad (13)
\end{aligned}$$

where $\mathbf{u}^k = \mathbf{q}^k - 2(N-1)\nabla_{\mathbf{z}}\Phi^k$. Note that at each iteration only one agent's decision variables are updated, hence, $\mathbf{q}^k = q_i^k$ for $i = i_{k-1}$. Therefore, using Young's inequality and Lipschitz continuity of g_i and $\|V_{i:}\| \leq \delta_i$ we conclude that for $i = i_{k-1}$,

$$\begin{aligned}
&N \left\langle \mathbf{q}^k, \mathbf{z}^{k+1} - \mathbf{z}^k \right\rangle = N \left\langle q_i^k, z_i^{k+1} - z_i^k \right\rangle \\
&= N(2N-1)\big(\left\langle g_i(x_i^k) - g_i(x_i^{k-1}), y_i^k - y_i^{k-1} \right\rangle \\
&\quad + \left\langle V_{i:}(x_i^k - x_i^{k-1}), \lambda_i^{k+1} - \lambda_i^k \right\rangle \big) \\
&\leq \tfrac{N(2N-1)}{2}\Big((C_i + \delta_i) \left\|x_i^k - x_i^{k-1}\right\|^2 + C_i \left\|y_i^{k+1} - y_i^k\right\|^2 \\
&\quad + \delta_i \left\|\lambda_i^{k+1} - \lambda_i^k\right\|^2 \Big).
\end{aligned}$$

Note that at iteration k only $z_{i_k}^{k+1}$ is updated where i_k is chosen with probability $1/N$. Hence, one can readily observe that $z_{i_{k-1}}^{k+1} \neq z_{i_{k-1}}^k$ with probability $1/N^2$ and $z_{i_{k-1}}^{k+1} = z_{i_{k-1}}^k$ otherwise. Therefore, taking conditional expectation from the above inequality imply that

$$\begin{aligned}
N\mathbb{E}^k[\langle \mathbf{q}^k, \mathbf{z}^{k+1} - \mathbf{z}^k \rangle] &\leq \tfrac{2N-1}{2}\mathbb{E}^k\big[\left\|\mathbf{x}^k - \mathbf{x}^{k-1}\right\|_{\mathbf{D}}^2 \big] \\
&\quad + \tfrac{2N-1}{2N}\mathbb{E}^k\big[\left\|\mathbf{z}^{k+1} - \mathbf{z}^k\right\|_{\mathbf{C}_2}^2 \big]. \qquad (14)
\end{aligned}$$

Finally, combining (14) and (13) with (12), and using the fact that $\left\|\tilde{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|^2 = N\mathbb{E}^k[\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|^2]$ the result can be concluded. ☐

A. Proof of Theorem V.1

Now we are ready to prove the main result. Consider the inequality obtained in (3). Taking expectations from both sides, using the step-size selection implying $\widetilde{\mathcal{B}} \succeq 0$ and $\widetilde{\mathcal{T}} \succeq 0$, summing the resulting inequality from $k = 0$ to $K-1$ and dividing by K we obtain

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})] \leq \\
&(N-1)\big(\mathcal{H}^0 - \mathbb{E}[\mathcal{H}^K]\big) + \left\langle \mathbf{u}^0, \mathbf{z}^0 - \mathbf{z} \right\rangle - \mathbb{E}[\langle \mathbf{u}^K, \mathbf{z}^K - \mathbf{z} \rangle] \\
&+ \tfrac{N}{2}\mathbb{E}\big[\left\|\mathbf{x} - \mathbf{x}^0\right\|_{\mathcal{T}}^2 - \left\|\mathbf{x} - \mathbf{x}^K\right\|_{\mathcal{T}}^2 - \left\|\mathbf{x}^{K-1} - \mathbf{x}^K\right\|_{\mathcal{T}-\mathbf{L}^{\Phi}}^2 \big] \\
&+ \tfrac{N}{2}\mathbb{E}\big[\left\|\mathbf{z} - \mathbf{z}^0\right\|_{\mathcal{B}}^2 - \left\|\mathbf{z} - \mathbf{z}^K\right\|_{\mathcal{B}}^2 \big]. \qquad (15)
\end{aligned}$$

We notice that function $\Phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$ is linear in $(\mathbf{y}, \boldsymbol{\lambda})$, i.e., $\nabla_{\mathbf{z}}\Phi(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \nabla_{\mathbf{z}}\Phi(\mathbf{x}, \bar{\mathbf{y}}, \bar{\boldsymbol{\lambda}})$ for any $\mathbf{y}, \bar{\mathbf{y}}, \boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}}$; therefore, we can show the following relations for any $k \geq 0$,

$$\begin{aligned}
&(N-1)(\mathcal{H}^k - \mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})) + \left\langle \mathbf{u}^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&= (N-1)\big(\mathcal{L}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})\big) + \left\langle \mathbf{q}^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&\quad - (N-1) \left\langle \nabla_{\mathbf{z}}\Phi^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&= (N-1)\big(\mathcal{L}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})\big) + \left\langle \mathbf{q}^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&\quad - (N-1) \left\langle \nabla_{\mathbf{z}}\Phi^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&\quad \pm (N-1) \Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k), \mathbf{z}^k - \mathbf{z} \Big\rangle \\
&= (N-1)\big(\mathcal{L}(\mathbf{x}^k, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k)\big) + \left\langle \mathbf{q}^k, \mathbf{z}^k - \mathbf{z} \right\rangle \\
&\quad + (N-1) \Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k) - \nabla_{\mathbf{z}}\Phi^k, \mathbf{z}^k - \mathbf{z} \Big\rangle. \qquad (16)
\end{aligned}$$

Next, we provide upper bounds for the two inner products on the right-hand side of (16) similar to (14).

$$\begin{aligned}
&|\left\langle \mathbf{q}^k, \mathbf{z}^k - \mathbf{z} \right\rangle| \leq \tfrac{2N-1}{2}\big(\left\|\mathbf{x}^k - \mathbf{x}^{k-1}\right\|_{\mathbf{D}}^2 + \left\|\mathbf{z}^k - \mathbf{z}\right\|_{\mathbf{C}_2}^2 \big) \\
&(N-1)\Big| \Big\langle \nabla_{\mathbf{z}}\Phi(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\lambda}^k) - \nabla_{\mathbf{z}}\Phi^k, \mathbf{z}^k - \mathbf{z} \Big\rangle \Big| \leq \\
&\quad \tfrac{N-1}{2}\big(\left\|\mathbf{x}^k - \mathbf{x}\right\|_{\mathbf{D}}^2 + \left\|\mathbf{z}^k - \mathbf{z}\right\|_{\mathbf{C}_2}^2 \big).
\end{aligned}$$

Now, with the help of above inequalities in (16) once for $k = 0$ and once for $k = K$, the fact that $\mathbf{q}^0 = 0$, and using the resulting inequality within (15) we obtain

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[\mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^{k+1}, \boldsymbol{\lambda}^{k+1})] \\
&\leq (N-1)(\mathcal{L}(\mathbf{x}^0, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^0, \boldsymbol{\lambda}^0)) \\
&- (N-1)(\mathcal{L}(\mathbf{x}^K, \mathbf{y}, \boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x}, \mathbf{y}^K, \boldsymbol{\lambda}^K)) \\
&+ \tfrac{N}{2}\mathbb{E}\big[\left\|\mathbf{x} - \mathbf{x}^0\right\|_{\mathcal{T}+\mathbf{D}}^2 - \left\|\mathbf{x} - \mathbf{x}^K\right\|_{\mathcal{T}-\mathbf{D}}^2 - \left\|\mathbf{x}^{K-1} - \mathbf{x}^K\right\|_{\widetilde{\mathcal{T}}}^2 \big] \\
&+ \tfrac{N}{2}\mathbb{E}\big[\left\|\mathbf{z} - \mathbf{z}^0\right\|_{\mathcal{B}+\mathbf{C}_2}^2 - \left\|\mathbf{z} - \mathbf{z}^K\right\|_{\mathcal{B}-3\mathbf{C}_2}^2 \big],
\end{aligned}$$

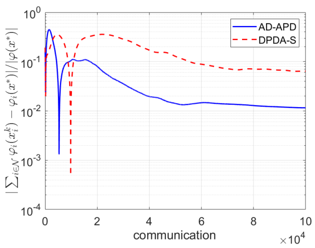
where $\widetilde{\mathcal{T}} = \mathcal{T} - \mathbf{L}^{\Phi} - 2\mathbf{D}$. Finally, rearranging the terms in the aforementioned inequality and dropping the negative terms due to the step-size selection lead to the desired result. ☐

VI. Numerical Experiments

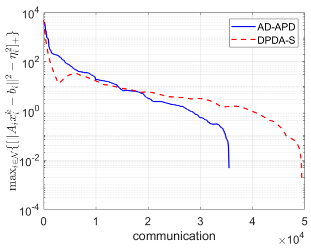
In this section, we consider a distributed localization problem to test the performance of our proposed algorithm. Given a set of local ellipsoids $\mathcal{X}_i \triangleq \{x \in [-1,1]^n \mid \|A_i x - b_i\| \leq \eta_i\}$, for $i \in \mathcal{N}$, where $A_i \in \mathbb{R}^{p_i \times n}$, $b_i \in \mathbb{R}^{p_i}$, and $\eta_i > 0$, the goal is to solve the following optimization problem: $\min_{x\in\mathbb{R}^n}\{\sum_{i\in\mathcal{N}} f_i(x) \mid x \in \bigcap_{i=1}^N \mathcal{X}_i\}$, over a network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. To highlight the benefit of our method, we compare ours with a synchronous distributed primal-dual method (DPDA-S) in [5].

In this experiments, we set $n = 100$, $N = 50$, $p_i = 50$, and $f_i(x) = \frac{1}{2}\|x\|^2$ for all $i \in \mathcal{N}$. We generate a vector $\bar{x} \in \mathbb{R}^n$ such that its entries are i.i.d with uniform distribution on $[-1, 1]$. For each $i \in \mathcal{N}$, A_i is generated with a standard Gaussian distribution, η_i uniformly at random on $[1, 2]$, and $b_i = A_i x + \epsilon_i$ where ϵ_i is generated with a normal distribution of mean zero and variance of 0.01. Moreover, to generate the network $\mathcal{G}$, we generated a random small-world network, i.e., we create a cycle over nodes, then we add $N/2$ edges at random with uniform probability. This leads to a connected graph with $|\mathcal{E}| = 75$.

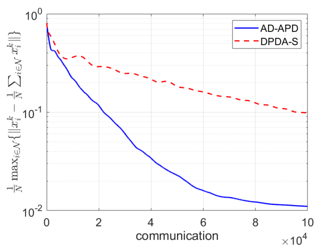
The results are depicted in Figure 1 in terms of suboptimality, infeasibility, and consensus violation versus the number of communications. Note that DPDA-S performs N communications at each iteration while AD-APD performs only one communication per iteration. From Figure 1, we see within the same number of communications AD-APD with asynchronous updates has a better performance than DPDA-S with synchronous updates.



(a) suboptimality



(b) infeasibility



(c) consensus violation

Fig. 1: Comparison of AD-APD and DPDA-S.

References

[1] C. M. Lagoa, X. Li, and M. Sznaier, "Probabilistically constrained linear programs and risk-adjusted controller design," *SIAM Journal on Optimization*, vol. 15, no. 3, pp. 938–951, 2005.

[2] K. You, R. Tempo, and P. Xie, "Distributed algorithms for robust convex optimization via the scenario approach," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 880–895, 2018.

[3] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[4] N. S. Aybat and E. Yazdandoost Hamedani, "A primal-dual method for conic constrained distributed optimization problems," *Advances in neural information processing systems*, vol. 29, 2016.

[5] E. Y. Hamedani and N. S. Aybat, "A decentralized primal-dual method for constrained minimization of a strongly convex function," *IEEE Transactions on Automatic Control*, 2021.

[6] D. Jakovetić, J. M. F. Xavier, and J. M. Moura, "Convergence rates of distributed nesterov-like gradient methods on random networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 868–882, 2013.

[7] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[8] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[9] S. Lee and A. Nedić, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 221–229, 2013.

[10] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Distributed constrained optimization and consensus in uncertain networks via proximal minimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1372–1387, 2017.

[11] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[12] I. Necoara, "Random coordinate descent algorithms for multi-agent convex optimization over networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 2001–2012, 2013.

[13] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE transactions on automatic control*, vol. 58, no. 2, pp. 391–405, 2012.

[14] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[15] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 788–794.

[16] H.-T. Wai, A. Scaglione, J. Lafond, and E. Moulines, "A projection-free decentralized algorithm for non-convex optimization," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2016, pp. 475–479.

[17] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1529–1538.

[18] C.-X. Shi and G.-H. Yang, "Augmented Lagrange algorithms for distributed optimization over multi-agent networks via edge-based method," *Automatica*, vol. 94, pp. 55–62, 2018.

[19] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.

[20] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[21] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[22] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2017.

[23] D. Jakovetic, J. Xavier, and J. M. Moura, "Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3889–3902, 2011.

[24] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 892–904, 2015.

[25] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2095–2106, 2016.

[26] P. Latafat, A. Bemporad, and P. Patrinos, "Plug and play distributed model predictive control with dynamic coupling: A randomized primal-dual proximal algorithm," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1160–1165.

[27] L. Xiao, A. W. Yu, Q. Lin, and W. Chen, "DSCOVR: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1634–1691, 2019.

[28] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "A distributed asynchronous method of multipliers for constrained nonconvex optimization," *Automatica*, vol. 103, pp. 243–253, 2019.

[29] H. Li, Q. Lü, G. Chen, T. Huang, and Z. Dong, "Distributed constrained optimization over unbalanced directed networks using asynchronous broadcast-based algorithm," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1102–1115, 2020.

[30] W. Tang and P. Daoutidis, "Fast and stable nonconvex constrained distributed optimization: the ellada algorithm," *Optimization and Engineering*, vol. 23, no. 1, pp. 259–301, 2022.

[31] N. S. Aybat and E. Y. Hamedani, "A distributed ADMM-like method for resource sharing over time-varying networks," *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 3036–3068, 2019.

[32] E. Y. Hamedani and N. S. Aybat, "A primal-dual algorithm with line search for general convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 31, no. 2, pp. 1299–1329, 2021.

[33] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, vol. 2, no. 3, 2008.