2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed least square approach for solving a multiagent linear algebraic equation

Viet Hoang Pham[1] and Hyo-Sung Ahn[1]

Abstract— **This paper considers a linear algebraic equation over a multiagent network. The coefficient matrix is partitioned into multiple blocks; each agent only knows a subset of these blocks in different row and column partitions. Based on a proximal ADMM algorithm, we design a distributed method for every agent to find its corresponding parts in one least square solution of the considered linear algebraic equation. In every iteration of the designed method, each agent uses only its information and communicates with its neighbors. We show that the designed method achieves an exponentially fast convergence for an arbitrarily initial setup. Numerical simulations in MATLAB are provided to verify the effectiveness of the designed method.**

I. Introduction

The control and optimization problems of multiagent systems have become popular in many applications. The main reason is inherent complexity of large-scale problems and memory dispersion with the growth of big data. Depending on the communication network among agents, a multiagent system can have a hierarchical or distributed scheme. In a hierarchical setup, one master agent plays as a coordinator. This agent communicates with and/or assigns tasks to all others. In a distributed setup, agents work in parallel, and each agent can exchange information with only its neighbors. Because of the scalability and privacy protection requirements, distributed ones have become more desired recently.

Many fundamental problems in various engineering applications can be deduced into finding one exact or least square solution of a linear algebraic equation $\mathbf{Hz} = \mathbf{h}$ where $\mathbf{H}$ and $\mathbf{h}$ are coefficient matrix and vector. For example, parameter estimation [1] and data filtering [2]. Various distributed methods have been developed to solve the linear algebraic equation $\mathbf{Hz} = \mathbf{h}$ when each agent knows only a subset of rows in $\mathbf{H}$ and $\mathbf{h}$. The first approach considers constrained consensus problems [3]–[9]. By using an orthogonal projection of the kernel of the local coefficient matrix, the estimated solution at every iteration of each agent always satisfies its linear equations. Distributed consensus schemes are exploited to drive estimated solutions of all agents to converge to an agreement. The method in [3] is continuous-time, and the ones in [4]–[9] are discrete-time. The convergence of these methods is guaranteed if the considered linear algebraic equation has at least one exact solution. Another approach considers the problem of solving a distributed linear algebraic equation as finding the optimal solution of a multiagent optimization problem [9]–[15]. The local cost function in each agent is the total square error of its local linear equations, and the constraints are added to force the agreement for estimated solutions among neighboring agents. Various distributed optimization algorithms can be applied to solve the optimization problem. Some of them can be listed by subgradient-based methods (as in [10]–[13]), zero-gradient-sum algorithm (as in [14]). When the considered linear algebraic equation has one or several exact solutions, all agents can asymptotically determine one of such points [15]. If the exact solution does not exist, the final converged solution is one least square solution, which minimizes the total square errors, of the considered linear algebraic equation [9]–[14].

In the works mentioned above [3]–[15], the authors consider a distributed linear algebraic equation $\mathbf{Hz} = \mathbf{h}$ assuming that the coefficient matrix $\mathbf{H}$ is divided into multiple row partitions and each agent knows one of these partitions. Consequently, in their proposed algorithms, except for the one in [5], every agent controls a state vector having the same dimension as the unknown vector $\mathbf{z}$ in its update. In [5], the dimension of the state vector controlled by one agent can be reduced when considering the sparsity in its row partition. Different from [3]–[15], the authors in [16] assume that the coefficient matrix $\mathbf{H}$ is first divided into either multiple row partitions or column partitions. Then, each of these partitions is further divided into multiple blocks. Only one agent can access each block. These setups are more suitable than those in [3]–[15] when the coefficient matrix $\mathbf{H}$ has large rows and columns. However, the algorithms proposed in [16] require a hierarchical structure. Each agent belongs to a cluster corresponding to one row or column partition. Each cluster has one aggregator to collect information on agents in this cluster and exchange information with aggregators of others.

This paper considers a linear algebraic equation $\mathbf{Hz} = \mathbf{h}$ when the rows and columns of the coefficient matrix $\mathbf{H}$ are distributed over a network of agents. Let the matrix $\mathbf{H}$ be divided into multiple blocks. Each agent only knows a subset of the divided blocks, which can be located in many row and column partitions. Unlike [16], we do not require an aggregator to collect information from agents corresponding to the same row or column partition. For the row partitions whose data are dispensed across multiple agents, we introduce some virtual variables to transform their linear equations into local constraints of individual agents combining coupling constraints among neighboring agents. Consequently, we obtain a distributed optimization problem whose optimal solution is one least square solution

[1] School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju, Korea. E-mails: vietph@gist.ac.kr; hyosung@gist.ac.kr.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

of the considered linear algebraic equation. Then we apply a proximal ADMM algorithm [17] to design a distributed method for finding the optimal solution of the obtained distributed optimization problem. Each agent must only use its information and communicate with its neighbors. Since the designed method is discrete-time, the asymptotic convergence of the ADMM-based method implies the exponential convergence of the estimated solution to the optimal one. In addition, our designed method has no requirements for step-size selection or initial point setup.

The remainder of this paper is organized as follows. In Section II, we introduce distributed linear algebraic equation setup and optimization-based approach. Then we formulate a distributed optimization problem whose optimal solution can be used to determine the least square solution of the considered linear algebraic equation in Section III. Based on a proximal ADMM algorithm [17], Section IV designs a distributed method for solving the formulated optimization problem. Numerical simulations are provided in Section V to illustrate the effectiveness of the designed method. Section VI concludes this paper. In this paper, we skip the proofs of our theorem and lemma due to space limitations. The detailed analysis can be found in the full version [18].

A. Notations

We use $\mathbb{R}, \mathbb{R}^n$, and $\mathbb{R}^{m\times n}$ to denote the set of real numbers, the set of n-dimension real vectors, and the set of real matrices having m rows and n columns. Denote by $\mathbf{1}_n$ and $\mathbf{0}_n$ the vectors in $\mathbb{R}^n$ whose all elements are 1 and 0, respectively. Let $\mathbf{I}_n$ be the identity matrix in $\mathbb{R}^{n\times n}$ and $\mathbf{0}_{m\times n}$ be the zero matrix in $\mathbb{R}^{m\times n}$. When the dimensions are clear, the subscripts of the matrices $\mathbf{I}_n, \mathbf{0}_{m\times n}$, and the vector $\mathbf{1}_n, \mathbf{0}_n$ can be removed. Denote by $\mathbf{H}^T$ the transpose matrix of $\mathbf{H}$. We write $\mathbf{H} \succ 0$ ($\succeq 0$) means $\mathbf{H}$ is a positive (semi-)definite matrix. Define $||\mathbf{x}||^2_{\mathbf{H}} = \mathbf{x}^T\mathbf{H}\mathbf{x}$. When $\mathbf{H} = \mathbf{I}$, we write it as $||\mathbf{x}||^2$ for simplicity. If x is a scalar, we write $|x| = ||x||$. We use $|\mathbb{S}|$ to denote the cardinality of the set $\mathbb{S}$.

Let $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\}$ be the list of m vectors, we define the column vector $\operatorname{col}\mathcal{H} = \operatorname{col}\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\} = \left[\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_m^T\right]^T$. For a list of matrices, $\mathcal{H} = \{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m\}$, we use $\operatorname{blkdiag}\mathcal{H}$ to denote the block-diagonal matrix whose main blocks are matrices in $\mathcal{H}$. In addition, we define two matrices $\operatorname{blkcol}\mathcal{H} = \operatorname{blkcol}\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m\} = \begin{bmatrix}\mathbf{H}_1^T & \mathbf{H}_2^T & \cdots & \mathbf{H}_m^T\end{bmatrix}^T$, $\operatorname{blkrow}\mathcal{H} = \operatorname{blkrow}\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m\} = \begin{bmatrix}\mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_m\end{bmatrix}$.

II. Problem Formulation

A. Distributed linear algebraic equation

Consider a linear algebraic equation

$$\mathbf{H}\mathbf{z} = \mathbf{h}, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^n$ is unknown vector, $\mathbf{H} \in \mathbb{R}^{m\times n}$ and $\mathbf{h} \in \mathbb{R}^m$ are coefficients matrix and vector, respectively. Let $\mathbf{z}^{opt}$ be one least square solution to (1). Then $\mathbf{z}^{opt}$ is one optimal solution to the following optimization problem.

$$\min_{\mathbf{z}} \frac{1}{2}\left|\left|\mathbf{H}\mathbf{z} - \mathbf{h}\right|\right|^2. \tag{2}$$

In general, the linear algebraic equation (1) can have one or more least square solutions. Define $\Psi^{opt} = \frac{1}{2}||\mathbf{H}\mathbf{z}^{opt} - \mathbf{h}||^2$. We have $\Psi^{opt} = 0$ if the linear algebraic equation (1) has at least one exact solution. If $\mathbf{H}$ is a full column rank matrix, we have $\mathbf{z}^{opt} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{h}$ is unique and $\Psi^{opt} = \frac{1}{2}\mathbf{h}^T\mathbf{h} - \frac{1}{2}\mathbf{h}^T\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{h}$.

In this paper, we are interested in determining one least square solution to the linear algebraic equation (1) when the coefficient matrix $\mathbf{H}$ can be subdivided into multiple blocks $\mathbf{H}_{kl}$, where $1 \le k \le K$ and $1 \le l \le L$, as in (3) and each nonzero block is known by only one agent in a network of N agents.

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12} & \cdots & \mathbf{H}_{1L}\\ \mathbf{H}_{21} & \mathbf{H}_{22} & \cdots & \mathbf{H}_{2L}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{H}_{K1} & \mathbf{H}_{K2} & \cdots & \mathbf{H}_{KL}\end{bmatrix}. \tag{3}$$

The coefficient matrix $\mathbf{H}$ in (3) includes KL blocks corresponding to K row and L column partitions. As each agent i, where $1 \le i \le N$, knows some nonzero blocks, we have $KL \ge N$ in general setting. We define $\mathcal{A}[\mathbf{H}_{kl}]$ as an operator to express the agent knowing the block $\mathbf{H}_{kl}$. Since there may be some zero blocks in $\mathbf{H}$, we conventionally define $\mathcal{A}[\mathbf{H}_{kl}] = 0$ if $\mathbf{H}_{kl} = \mathbf{0}$. Note that 0 is not an agent. For each agent i, we define the relation between the block $\mathbf{H}_{kl}$ and this agent by $\delta_i[\mathbf{H}_{kl}]$ where $\delta_i[\mathbf{H}_{kl}] = 1$ if $\mathcal{A}[\mathbf{H}_{kl}] = i$ and $\delta_i[\mathbf{H}_{kl}] = 0$, otherwise. Let $\mathcal{B}_k^{(i)} = \{\mathbf{H}_{kl} : \delta_i[\mathbf{H}_{kl}] = 1, \forall 1 \le l \le L\}$ be the set of blocks in the row partition k known by the agent i, where $1 \le k \le K$ and $1 \le i \le N$.

For each row partition k, where $1 \le k \le K$, of the coefficient matrix $\mathbf{H}$, we define the index list of agents that have information of blocks in this row partition as $\mathcal{R}_k = \{\mathcal{A}[\mathbf{H}_{k1}], \mathcal{A}[\mathbf{H}_{k2}], \dots, \mathcal{A}[\mathbf{H}_{kL}]\}$. Let $\mathcal{C}^l$ be the index list of agents that have information of blocks in the column partition l, i.e., $\mathcal{C}^l = \{\mathcal{A}[\mathbf{H}_{1l}], \mathcal{A}[\mathbf{H}_{2l}], \dots, \mathcal{A}[\mathbf{H}_{Kl}]\}$, where $1 \le l \le L$. As $\mathcal{R}_k$ and $\mathcal{C}^l$, where $1 \le k \le K$ and $1 \le l \le L$, are lists, they can contain duplicated indexes. We define the set $\mathbb{S}(\mathcal{R}_k)$ derived from the list $\mathcal{R}_k$ by removing the index 0 and merging duplicated indexes into one element. Similarly, we define $\mathbb{S}(\mathcal{C}^l)$ for the list $\mathcal{C}^l$, where $1 \le l \le L$. We can obtain an equivalent linear algebraic equation to (1) if we exchange rows in the matrix $\mathbf{H}$ and the vector $\mathbf{h}$. So, the rows in $\mathbf{H}$ can be interchanged such that the ones dispensed across the same list of agents are grouped in the same row partition. Without loss of generality, we assume that $\mathcal{R}_k \ne \mathcal{R}_{k'}$ if $k \ne k'$ for all $1 \le k, k' \le K$.

B. Optimization approach

Denote by m_k, n_l the dimensions of the block $\mathbf{H}_{kl} \in \mathbb{R}^{m_k \times n_l}$, where $1 \le k \le K$ and $1 \le l \le L$. We have $m = \sum_{k=1}^{K} m_k$ and $n = \sum_{l=1}^{L} n_l$. According to the partition of the coefficient matrix $\mathbf{H}$, the unknown vector $\mathbf{z}$ and the coefficient vector $\mathbf{h}$ can be partitioned as $\mathbf{z} = \operatorname{col}\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L\}$ and $\mathbf{h} = \operatorname{col}\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K\}$ where

$$\mathbf{z}_l = \operatorname{blkrow}\left\{\mathbf{0}_{n_l \times \sum_{j=0}^{l-1} n_j}, \mathbf{I}_{n_l}, \mathbf{0}_{n_l \times \left(n - \sum_{j=0}^{l} n_j\right)}\right\}\mathbf{z},$$

$$\mathbf{h}_k = \text{blkrow}\left\{\mathbf{0}_{m_k \times \sum_{j=0}^{k-1} m_j}, \mathbf{I}_{m_k}, \mathbf{0}_{m_k \times \left(m - \sum_{j=0}^{k} m_j\right)}\right\}\mathbf{h}.$$

Here we set $m_0 = n_0 = 0$ conventionally. Assume that each agent $i \in \mathbb{S}(\mathcal{C}^l)$ has a copy $\mathbf{z}_l^{(i)}$ for the part $\mathbf{z}_l$ of the unknown vector $\mathbf{z}$. We have

$$\sum_{l=1}^{L} \mathbf{H}_{kl}\mathbf{z}_l = \sum_{l=1,\mathbf{H}_{kl}\neq\mathbf{0}}^{L} \mathbf{H}_{kl}\mathbf{z}_l^{(\mathcal{A}[\mathbf{H}_{kl}])}, \forall 1 \le k \le K. \tag{4}$$

If $\mathcal{A}[\mathbf{H}_{kl}] \neq 0$, the vector $\mathbf{H}_{kl}\mathbf{z}_l^{\mathcal{A}[\mathbf{H}_{kl}]}$ is private information of the agent $\mathcal{A}[\mathbf{H}_{kl}]$. To rewrite the error of the linear equations corresponding to the row partition k as the sum of local private functions, we make the following assumption:

Assumption 1: For every row partition k, where $1 \le k \le K$, each agent $i \in \mathbb{S}(\mathcal{R}_k)$ can access to a part of $\mathbf{h}_k$ (i.e., can have partial information of $\mathbf{h}_k$). Denote this part as $\mathbf{h}_{i,k}$. However, the sum of these parts $\sum_{i\in\mathbb{S}(\mathcal{R}_k)} \mathbf{h}_{i,k} = \mathbf{h}_k$.

Under Assumption 1, we have

$$\sum_{l=1}^{L} \mathbf{H}_{kl}\mathbf{z}_l - \mathbf{h}_k = \sum_{i\in\mathbb{S}(\mathcal{R}_k)} \Big(\sum_{\mathbf{H}_{kl}\in\mathcal{B}_k^{(i)}} \mathbf{H}_{kl}\mathbf{z}_l^{(i)} - \mathbf{h}_{i,k}\Big), \tag{5}$$

for all $1 \le k \le K$. So, we have an equivalent problem of the optimization problem (2) as follows.

$$\min_{\mathbf{z}_l^{(i)}} \frac{1}{2} \sum_{k=1}^{K} \left|\left| \sum_{i\in\mathbb{S}(\mathcal{R}_k)} \left(\sum_{\mathbf{H}_{kl}\in\mathcal{B}_k^{(i)}} \mathbf{H}_{kl}\mathbf{z}_l^{(i)} - \mathbf{h}_{i,k} \right) \right|\right|^2 \tag{6a}$$

$$\text{s.t. } \mathbf{z}_l^{(i)} = \mathbf{z}_l^{(j)}, \forall i, j \in \mathbb{S}(\mathcal{C}^l), i \neq j, \forall 1 \le l \le L. \tag{6b}$$

It is clear that the optimal cost value $\Psi^{opt} = \frac{1}{2}||\mathbf{H}\mathbf{z}^{opt} - \mathbf{h}||^2$ is also the optimal value for the cost function in the problem (6). In addition, each optimal solution to the optimization problem (6) can be used to determine one least square solution (2) of the linear algebraic equation (1).

C. Communication graph

In this paper, we use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to illustrate the communication network among agents. In which, $\mathcal{V} = \{1, 2, \dots\}$ is the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. Each node $i \in \mathcal{V} = \{1, 2, \dots, N\}$ corresponds to one agent while an edge $(i, j) \in \mathcal{E}$ represents that the agent i can exchange information with the agent j. As $\mathcal{G}$ is an undirected graph, $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$ for every pair of nodes $i, j \in \mathcal{V}$. The neighbor set of the agent $i \in \mathcal{V}$ is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Since agents need to cooperate to find the optimal solution of the problem (6), the communication graph $\mathcal{G}$ is required to be connected. That means, for any pair $i, j \in \mathcal{V}$, there is at least one sequence of edges $(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)$ such that $i_1 = i, i_k = j$ and $i_l \in \mathcal{V}, (i_{l-1}, i_l) \in \mathcal{E}, \forall 2 \le l \le k$. Let $\tilde{\mathcal{V}}$ be a node subset, i.e., $\tilde{\mathcal{V}} \subset \mathcal{V}$. The subgraph induced by $\tilde{\mathcal{V}}$ is defined as $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ where $\tilde{\mathcal{E}} = \{(i, j) \in \mathcal{E} : i, j \in \tilde{\mathcal{V}}\}$. We define $\mathcal{G}_k = (\mathbb{S}(\mathcal{R}_k), \mathcal{E}_k)$ as the subgraph induced by the node set $\mathbb{S}(\mathcal{R}_k)$ for each row partition k, where $1 \le k \le K$. Similarly, we define $\mathcal{G}^l = (\mathbb{S}(\mathcal{C}^l), \mathcal{E}^l)$ for each column partition l, where $1 \le l \le L$. The following assumption is necessary.

Assumption 2: The communication graph satisfies that

1) $\mathcal{G}^l$ is a connected graph for all $1 \le l \le L$,
2) $\mathcal{G}_k$ is a connected graph for all $1 \le k \le K$.

The first item in Assumption 2 is to allow agents in the set $\mathbb{S}(\mathcal{C}^l)$ can cooperatively achieve an agreement on the part $\mathbf{z}_l$ of the unknown vector $\mathbf{z}$. Under this assumption, the constraints in (6b) can be converted into coupling constraints among neighboring agents as follows.

$$\mathbf{z}_l^{(i)} = \mathbf{z}_l^{(j)}, \forall j \in \mathcal{N}_i \cap \mathbb{S}(\mathcal{C}^l), \forall i \in \mathbb{S}(\mathcal{C}^l), \tag{7}$$

for all $1 \le l \le L$. When $i \in \mathbb{S}(\mathcal{R}_k)$, the agent i takes part in the linear equations corresponding to the row partition k as shown in (5). In general, there may be some row partitions $1 \le k \le K$ where $\mathbb{S}(\mathcal{R}_k) \neq \mathcal{N}_i$ for all $i \in \mathbb{S}(\mathcal{R}_k)$. Since communication is only available between two neighboring agents, it is difficult for one agent to collect complete information on the linear equations corresponding to these row partitions. The second item in Assumption 2 facilitates a transformation of these linear equations into local linear equations of individual agents combining coupling constraints among neighboring agents, as presented later in Section III. Summarizing this section, we state the main problem of this paper as follows.

Problem 1: Assume that the communication graph of the considered multiagent network satisfies Assumption 2. Design a distributed method for each agent to find its corresponding parts in one optimal solution of the constrained optimization problem (6) while using only its information and communicating with its neighbors in $\mathcal{N}_i$.

III. DISTRIBUTED OPTIMIZATION REFORMULATION

Let $\bar{\mathbf{z}}_i = \text{col}\{\mathbf{z}_l^{(i)} : i \in \mathbb{S}(\mathcal{C}^l), \forall 1 \le l \le L\}$ be the stacked vector of parts corresponding to the agent $i \in \mathcal{V}$ in the unknown vector $\mathbf{z}$. For each neighboring agent $j \in \mathcal{N}_i$, define the matrix $\mathbf{P}_{ij}$ such that $\mathbf{P}_{ij}\bar{\mathbf{z}}_i = \text{col}\{\mathbf{z}_l^{(i)} : i, j \in \mathbb{S}(\mathcal{C}^l), \forall 1 \le l \le L\}$. The constraints in (7) are equivalent to the following one:

$$\mathbf{P}_{ij}\bar{\mathbf{z}}_i = \mathbf{P}_{ji}\bar{\mathbf{z}}_j, \forall (i, j) \in \mathcal{E}. \tag{8}$$

Let $\Psi(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \dots, \bar{\mathbf{z}}_N)$ be the cost function in (6a). The optimization problem (6) can be rewritten as

$$\min_{\bar{\mathbf{z}}_i, \forall i\in\mathcal{V}} \Psi(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \dots, \bar{\mathbf{z}}_N) \text{ s.t. (8)}. \tag{9}$$

Consider a row partition k where there is only one agent i in the set $\mathbb{S}(\mathcal{R}_k)$, i.e., $\mathbb{S}(\mathcal{R}_k) = \{i\}$. Under Assumption 1, the agent i knows the vector $\mathbf{h}_k$. In this case, we have

$$\sum_{l=1}^{L} \mathbf{H}_{kl}\mathbf{z}_l - \mathbf{h}_k = \bar{\mathbf{A}}_{i,k}\bar{\mathbf{z}}_i - \bar{\mathbf{a}}_{i,k}, \text{ if } \mathbb{S}(\mathcal{R}_k) = \{i\}, \tag{10}$$

where $\bar{\mathbf{a}}_{i,k} = \mathbf{h}_k$ and $\bar{\mathbf{A}}_{i,k} = \text{blkrow}\Big\{\mathbf{H}_{kl}\delta_i[\mathbf{H}_{kl}] : i \in \mathbb{S}(\mathcal{R}_k) \cap \mathbb{S}(\mathcal{C}^l), \forall 1 \le l \le L\Big\}$. Define by $\mathbb{M} = \{\epsilon_1, \epsilon_2, \dots, \epsilon_M\}$ the set of row partitions whose corresponding sets consist of two or more agents, i.e., $|\mathbb{S}(\mathcal{R}_{\epsilon_k})| \ge 2$ for all $\epsilon_k \in \mathbb{M}$. For each row partition $\epsilon_k \in \mathbb{M}$ and agent

$i \in \mathcal{V}$, we define the matrix $\bar{\mathbf{B}}_{i,\epsilon_k} = \text{blkrow}\Big\{\mathbf{H}_{\epsilon_k l}\delta_i[\mathbf{H}_{\epsilon_k l}] : i \in \mathbb{S}(\mathcal{R}_{\epsilon_k}) \cap \mathbb{S}(\mathcal{C}^l), \forall \leq l \leq L\Big\}$ and $\bar{\mathbf{b}}_{i,\epsilon_k} = \mathbf{h}_{i,\epsilon_k}$. We have

$$\sum_{l=1}^{L} \mathbf{H}_{\epsilon_k l}\mathbf{z}_l - \mathbf{h}_{\epsilon_k} = \sum_{i\in\mathbb{S}(\mathcal{R}_{\epsilon_k})} \left(\bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i - \bar{\mathbf{b}}_{i,\epsilon_k}\right). \quad (11)$$

Let $\mathbf{u}_{\epsilon_k}$ be the averaged error of the coupling linear equations in the row partition $\epsilon_k \in \mathbb{M}$, i.e.,

$$\mathbf{u}_{\epsilon_k} = \frac{1}{|\mathbb{S}(\mathcal{R}_{\epsilon_k})|} \sum_{i\in\mathbb{S}(\mathcal{R}_{\epsilon_k})} \left(\bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i - \bar{\mathbf{b}}_{i,\epsilon_k}\right), \forall \epsilon_k \in \mathbb{M}. \quad (12)$$

For every $\epsilon_k \in \mathbb{M}$, we define the vector $\mathbf{v}_{ij,\epsilon_k}$ for each edge $(i,j) \in \mathcal{E}_{\epsilon_k}$ such that

$$\mathbf{u}_{\epsilon_k} = \bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i - \bar{\mathbf{b}}_{i,\epsilon_k} + \sum_{j\in\mathcal{N}_{i,\epsilon_k}} \left(\mathbf{v}_{ij,\epsilon_k} - \mathbf{v}_{ji,\epsilon_k}\right), \quad (13)$$

for all $i \in \mathbb{S}(\mathcal{R}_{\epsilon_k})$, where $\mathcal{N}_{i,\epsilon_k} \equiv \mathcal{N}_i \cap \mathbb{S}(\mathcal{R}_{\epsilon_k})$. We remark here that there always exists a set $\{\mathbf{v}_{ij,\epsilon_k} : (i,j) \in \mathcal{E}_{\epsilon_k}\}$ that satisfying (13). Indeed, from (13), we have

$$\sum_{j\in\mathcal{N}_{i,\epsilon_k}} \left(\mathbf{v}_{ij,\epsilon_k} - \mathbf{v}_{ji,\epsilon_k}\right) = \mathbf{u}_{\epsilon_k} - \bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i + \bar{\mathbf{b}}_{i,\epsilon_k}, \quad (14)$$

for all $i \in \mathbb{S}(\mathcal{R}_{\epsilon_k})$. Consider (14) as a system of linear equations with $\{\mathbf{v}_{ij,\epsilon_k} : (i,j) \in \mathcal{E}_{\epsilon_k}\}$ is the set of unknown variables. It is clear that there are $|\mathbb{S}(\mathcal{R}_{\epsilon_k})|$ linear equations and $2|\mathcal{E}_{\epsilon_k}|$ variables. Since $\mathcal{G}_{\epsilon_k} = (\mathbb{S}(\mathcal{R}_{\epsilon_k}), \mathcal{E}_{\epsilon_k})$ is a connected graph, we have $|\mathcal{E}_{\epsilon_k}| \geq |\mathbb{S}(\mathcal{R}_{\epsilon_k})|$. This implies that the linear algebraic equation (14) has infinite roots.

Let $\mathbf{v}^{(i)}_{ij,\epsilon_k}$ be the copy of $\mathbf{v}_{ij,\epsilon_k} - \mathbf{v}_{ji,\epsilon_k}, \forall (i,j) \in \mathcal{E}_{\epsilon_k}$, in the agent $i \in \mathbb{S}(\mathcal{R}_{\epsilon_k})$. Under the second item in Assumtion 2, the coupling linear equations in (12) implies that $\mathbf{u}_{\epsilon_k} = \bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i + \sum_{j\in\mathcal{N}_{i,\epsilon_k}} \mathbf{v}^{(i)}_{ij,\epsilon_k} - \bar{\mathbf{b}}_{i,\epsilon_k}$ if the constraints in (15) are satisfied for all $j \in \mathcal{N}_{i,\epsilon_k}$, where $i \in \mathbb{S}(\mathcal{R}_{\epsilon_k})$.

$$\sum_{l\in\mathcal{N}_{i,\epsilon_k}} \mathbf{v}^{(i)}_{il,\epsilon_k} + \bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i - \bar{\mathbf{b}}_{i,\epsilon_k} = \sum_{l\in\mathcal{N}_{j,\epsilon_k}} \mathbf{v}^{(j)}_{jl,\epsilon_k} + \bar{\mathbf{B}}_{j,\epsilon_k}\bar{\mathbf{z}}_j - \bar{\mathbf{b}}_{j,\epsilon_k}, \quad (15a)$$

$$\mathbf{v}^{(i)}_{ij,\epsilon_k} + \mathbf{v}^{(j)}_{ij,\epsilon_k} = \mathbf{0}. \quad (15b)$$

Define the set $\mathcal{M}_i = \{\epsilon_k \in \mathbb{M} : i \in \mathbb{S}(\mathcal{R}_{\epsilon_k})\}$ and the stacked vector $\bar{\mathbf{v}}_i = \text{col}\big\{\text{col}\big\{\mathbf{v}^{(i)}_{ij,\epsilon_k} : j \in \mathcal{N}_{i,\epsilon_k}\big\} : \epsilon_k \in \mathcal{M}_i\big\}$ for each agent $i \in \mathcal{V}$. In addition, we define the stacked matrix $\mathbf{A}_i$ and the stacked vector $\mathbf{a}_i$ for each agent $i \in \mathcal{V}$ as

$$\mathbf{A}_i = \text{blkcol}\big\{\bar{\mathbf{A}}_{i,k} : \mathbb{S}(\mathcal{R}_k) = \{i\}, \forall 1 \leq k \leq K\big\},$$
$$\mathbf{a}_i = \text{blkcol}\big\{\bar{\mathbf{a}}_{i,k} : \mathbb{S}(\mathcal{R}_k) = \{i\}, \forall 1 \leq k \leq K\big\}.$$

For simple notation, we assume $\mathbf{A}_i = \mathbf{0}$ and $\mathbf{a}_i = \mathbf{0}$ for agent $i \in \mathcal{V}$ if $\mathbb{S}(\mathcal{R}_k) \neq \{i\}, \forall 1 \leq k \leq K$, i.e., the agent i does not know all nonzero blocks in any row partition of the coefficient matrix $\mathbf{H}$. From (10), (12) and (15), we have $\Psi(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \ldots, \bar{\mathbf{z}}_N) = \sum_{i\in\mathcal{V}} \Psi_i(\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i)$ where $\Psi_i(\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i) = \frac{1}{2}\big|\big|\mathbf{A}_i\bar{\mathbf{z}}_i - \mathbf{a}_i\big|\big|^2 + \sum_{\epsilon_k\in\mathcal{M}_i} \frac{|\mathbb{S}(\mathcal{R}_{\epsilon_k})|}{2}\Big|\Big|\sum_{j\in\mathcal{N}_{i,\epsilon_k}} \mathbf{v}^{(i)}_{ij,\epsilon_k} + \bar{\mathbf{B}}_{i,\epsilon_k}\bar{\mathbf{z}}_i - \bar{\mathbf{b}}_{i,\epsilon_k}\Big|\Big|^2$. Consider the following optimization problem:

$$\min_{\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i, \forall i\in\mathcal{V}} \sum_{i\in\mathcal{V}} \Psi_i(\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i) \text{ s.t. } (8), (15) \forall \epsilon_k \in \mathbb{M}. \quad (16)$$

We have the following result:

Lemma 1: Let $\text{col}\{\text{col}\{\bar{\mathbf{z}}^{opt}_i, \bar{\mathbf{v}}^{opt}_i\} : i \in \mathcal{V}\}$ be one optimal solution to the problem (16). Then $\bar{\mathbf{z}}^{opt}_i, \forall i \in \mathcal{V}$, is the stacked vector of the parts corresponding to the agent i in a least square solution to the linear algebraic equation (1).

Proof: See [18]. ■

Remark 1: In [18], we provide an example to illustrate the definitions (of lists and sets) in distributed setup and the reformulation of distributed linear algebraic equation (10-11) into the distributed optimization problem (16).

IV. Distributed ADMM-based method

In this section, we apply the proximal ADMM algorithm proposed in [17] to design a distributed method for solving the optimization problem (16). The designed method is guaranteed to have exponential convergence.

A. ADMM based solution

Define $\mathbf{x}_i = \text{col}\{\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i\}$ as the collection of all local variables of the agent $i \in \mathcal{V}$. Let $\bar{n}_i$ be the length of the vector $\bar{\mathbf{z}}_i$ for every agent $i \in \mathcal{V}$. We have $\bar{\mathbf{z}}_i = \text{blkrow}\{\mathbf{I}_{\bar{n}_i}, \mathbf{0}\}\mathbf{x}_i, \forall i \in \mathcal{V}$. Define $\mathbf{u}^{(i)}_{\epsilon_k}$ as the copy of the vector $\mathbf{u}_{\epsilon_k}$ in (13) and the stacked vector $\bar{\mathbf{u}}_i = \text{col}\{\mathbf{u}^{(i)}_{\epsilon_k} : \epsilon_k \in \mathcal{M}_i\}$. Then we have $\bar{\mathbf{u}}_i = \mathbf{C}_i\mathbf{x}_i - \mathbf{c}_i$ where $\mathbf{c}_i = \text{col}\{\bar{\mathbf{b}}_{i,\epsilon_k} : \epsilon_k \in \mathcal{M}_i\}$ and $\mathbf{C}_i = \begin{bmatrix} \text{blkrow}\{\bar{\mathbf{B}}^T_{i,\epsilon_k} : \epsilon_k \in \mathcal{M}_i\} \\ \text{blkdiag}\big\{\text{blkcol}\big\{\mathbf{I}_{m_k} : j \in \mathcal{N}_{i,\epsilon_k}\big\} : \epsilon_k \in \mathcal{M}_i\big\} \end{bmatrix}^T$. The local cost function $\Psi_i(\bar{\mathbf{z}}_i, \bar{\mathbf{v}}_i)$ of the agent $i \in \mathcal{V}$ has the quadratic form as

$$\Psi_i(\bar{\mathbf{z}}_i, \bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i) = \frac{1}{2}\mathbf{x}_i^T\mathbf{Q}_i\mathbf{x}_i + \mathbf{q}_i^T\mathbf{x}_i + \mathbf{a}_i^T\mathbf{a}_i + \mathbf{D}_i\mathbf{c}_i^T\mathbf{c}_i.$$

where $\mathbf{Q}_i = \text{blkdiag}\Big\{\mathbf{A}_i^T\mathbf{A}_i, \mathbf{0}\Big\} + \mathbf{D}_i\mathbf{C}_i^T\mathbf{C}_i$, $\mathbf{q}_i = \text{col}\{-\mathbf{A}_i^T\mathbf{a}_i, \mathbf{0}\} + \mathbf{D}_i\mathbf{C}_i^T\mathbf{c}_i$, and $\mathbf{D}_i = \text{blkdiag}\{|\mathbb{S}(\mathcal{R}_{\epsilon_k})| : \epsilon_k \in \mathcal{M}_i\}$. It is easy to verify that $\mathbf{Q}_i$ is a symmetric and positive semidefinite matrix, $\forall i \in \mathcal{V}$.

For two agents $i, j \in \mathbb{S}(\mathcal{R}_{\epsilon_k})$, where $(i,j) \in \mathcal{E}$ and $\epsilon_k \in \mathbb{M}$, we assume that they choose the pair of matrices $\mathbf{D}_{ij,\epsilon_k}$ and $\mathbf{D}_{ji,\epsilon_k}$ such that if $\mathbf{D}_{ij,\epsilon_k} = \mathbf{I}$ then $\mathbf{D}_{ji,\epsilon_k} = -\mathbf{I}$ and if $\mathbf{D}_{ij,\epsilon_k} = -\mathbf{I}$ then $\mathbf{D}_{ji,\epsilon_k} = \mathbf{I}$ and vice versa. Define the matrices $\mathbf{E}_{ij}, \forall j \in \mathcal{N}_i$, such that

$$\mathbf{E}_{ij}\mathbf{x}_i = \text{col}\{\mathbf{P}_{ij}\bar{\mathbf{z}}_i, \text{col}\{\mathbf{u}^{(i)}_{\epsilon_k}, \mathbf{D}_{ij,\epsilon_k}\mathbf{v}^{(i)}_{ij,\epsilon_k} : \epsilon_k \in \mathcal{M}_i \cap \mathcal{M}_j\}\}. \quad (17)$$

It is clear that $\mathbf{E}_{ij}\mathbf{x}_i = \mathbf{E}_{ji}\mathbf{x}_j, \forall (i,j) \in \mathcal{E}$. The optimization problem (16) can be rewritten in the following form:

$$\min_{\substack{\mathbf{x}_i, \forall i\in\mathcal{V} \\ (\mathbf{y}_{ij}, \mathbf{y}_{ji})\in\mathcal{Y}_{ij}, \forall (i,j)\in\mathcal{E}}} \sum_{i\in\mathcal{V}} \left(\frac{1}{2}\mathbf{x}_i^T\mathbf{Q}_i\mathbf{x}_i + \mathbf{q}_i^T\mathbf{x}_i\right) \quad (18a)$$

$$\text{s.t. } \mathbf{E}_{ij}\mathbf{x}_i = \mathbf{y}_{ij}, \forall j \in \mathcal{N}_i, \forall i \in \mathcal{V}. \quad (18b)$$

where $\mathcal{Y}_{ij} = \{(\mathbf{y}_{ij}, \mathbf{y}_{ji}) : \mathbf{y}_{ij} = \mathbf{y}_{ji}\}$ for all $(i,j) \in \mathcal{E}$. The coupling constraints (8) and (15) between two neighboring agents i and j, where $(i,j) \in \mathcal{E}$, are replaced by the equality constraints in (18b) combining the constraint $(\mathbf{y}_{ij}, \mathbf{y}_{ji}) \in \mathcal{Y}_{ij}$. It is easy to verify that the problem (18) has the form of the convex optimization problem (19).

$$\min_{\mathbf{x}, \mathbf{y}\in\mathcal{Y}} \Psi_x(\mathbf{x}) + \Psi_y(\mathbf{y}) \text{ s.t. } \mathbf{F}_x\mathbf{x} + \mathbf{F}_y\mathbf{y} = \mathbf{f}, \quad (19)$$

with the variable vectors $\mathbf{x} = \text{col}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{y} = \text{col}\{\text{col}\{\mathbf{y}_{ij}, \mathbf{y}_{ji}\} : (i,j) \in \mathcal{E}\}$, the cost functions $\Psi_x(\mathbf{x}) = \sum_{i \in \mathcal{V}} \left\{ \frac{1}{2}\mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \mathbf{q}_i^T \mathbf{x}_i \right\}, \Psi_y(\mathbf{y}) = 0$, and $\mathcal{Y} = \prod_{(i,j)\in\mathcal{E}} \mathcal{Y}_{ij}$. Define $\boldsymbol{\lambda}_{ij}$, $\forall j \in \mathcal{N}_i$, as the dual variables corresponding to the equality constraints in (18b) of the agent $i \in \mathcal{V}$. These dual variables can be stacked into a vector $\boldsymbol{\lambda} = \text{col}\{\text{col}\{\boldsymbol{\lambda}_{ij}, \boldsymbol{\lambda}_{ji}\} : (i,j) \in \mathcal{E}\}$. The augmented Lagrangian function of the problem (18) is given by $\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{i \in \mathcal{V}} \left\{ \frac{1}{2}\mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \mathbf{q}_i^T \mathbf{x}_i + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} ||\mathbf{E}_{ij}\mathbf{x}_i - \mathbf{y}_{ij} - \frac{1}{\rho}\boldsymbol{\lambda}_{ij}||^2 \right\}$ where $\rho > 0$ is the penalty parameter.

The proximal ADMM algorithm proposed in [17] for solving the convex optimization problem (19) is given in (20) where $\mathbf{G}$ is a symmetric and positive semidefinite matrix, and the initial point $(\mathbf{x}(0), \mathbf{y}(0), \boldsymbol{\lambda}(0))$ can be chosen arbitrarily such that $\mathbf{y}(0) \in \mathcal{Y}$.

$$\mathbf{x}(s+1) = \arg\min_{\mathbf{x}} \Big\{ \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}(s), \boldsymbol{\lambda}(s)) + \tfrac{1}{2} ||\mathbf{x} - \mathbf{x}(s)||_{\mathbf{G}}^2 \Big\}, \quad (20a)$$

$$\mathbf{y}(s+1) = \arg\min_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}_\rho(\mathbf{x}(s+1), \mathbf{y}, \boldsymbol{\lambda}(s)), \quad (20b)$$

$$\boldsymbol{\lambda}(s+1) = \boldsymbol{\lambda}(s) - \rho\Big(\mathbf{F}_x\mathbf{x}(s+1) + \mathbf{F}_y\mathbf{y}(s+1) - \mathbf{f}\Big). \quad (20c)$$

B. Detailed algorithm

Due to the separation of the augmented Lagrangian function $\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda})$, the equations (20a) and (20b) are equivalent to the equations (21) and (22), respectively. Here, the matrix $\mathbf{G} = \text{blkdiag}\{\mathbf{G}_i : i \in \mathcal{V}\}$ where $\mathbf{G}_i$ is a positive semidefinite matrix chosen by the agent $i \in \mathcal{V}$. Assume that $\mathbf{G}_i$ is chosen such that the matrix $\hat{\mathbf{Q}}_i$ (defined in (23b)) is positive definite, the update (21) is equivalent to (23), $\forall i \in \mathcal{V}$.

$$\mathbf{x}_i(s+1) = -\hat{\mathbf{Q}}_i^{-1}\hat{\mathbf{q}}_i(s), \quad (23a)$$

$$\text{where } \hat{\mathbf{Q}}_i = \mathbf{Q}_i + \mathbf{G}_i + \rho \sum_{j\in\mathcal{N}_i} \mathbf{E}_{ij}^T\mathbf{E}_{ij}, \quad (23b)$$

$$\hat{\mathbf{q}}_i(s) = \mathbf{q}_i - \mathbf{G}_i\mathbf{x}_i(s) - \sum_{j\in\mathcal{N}_i} \mathbf{E}_{ij}^T \left(\rho\mathbf{y}_{ij}(s) + \boldsymbol{\lambda}_{ij}(s)\right). \quad (23c)$$

Consider the optimization problem in (22). From its KKT conditions, we have

$$\begin{aligned} \mathbf{y}_{ij}(s+1) =& \frac{1}{2}\left(\mathbf{E}_{ij}\mathbf{x}_i(s+1) - \frac{1}{\rho}(\boldsymbol{\lambda}_{ij}(s)\right) \\ &+ \frac{1}{2}\left(\mathbf{E}_{ji}\mathbf{x}_j(s+1) - \frac{1}{\rho}\boldsymbol{\lambda}_{ji}(s)\right), \quad (24) \end{aligned}$$

for all $(i,j) \in \mathcal{E}$. According to (20c), the update of the dual variables $\boldsymbol{\lambda}_{ij}, \forall(i,j) \in \mathcal{E}$, are given as

$$\boldsymbol{\lambda}_{ij}(s+1) = \boldsymbol{\lambda}_{ij}(s) - \rho\left(\mathbf{E}_{ij}\mathbf{x}_i(s+1) - \mathbf{y}_{ij}(s+1)\right). \quad (25)$$

We provide Algorithm 1 to present the implementation of our designed method for one agent. It is easy to verify that this algorithm requires only local information for each agent except the common penalty parameter ρ as global information. However, this information can be easily achieved by the agreement among agents. In the iteration update (23-25), every agent $i \in \mathcal{V}$ uses only information belonging to itself or received from its neighbors in $\mathcal{N}_i$.

Algorithm 1 Distributed method for agent $i \in \mathcal{V}$ to find $\bar{\mathbf{z}}_{\mathcal{S}_i}^{opt}$.

```
Agree a common positive penalty parameter $\rho > 0$
Choose $\mathbf{G}_i \succeq 0$ such that $\hat{\mathbf{Q}}_i \succ 0$ (given in (23b))
Initiate: $s = 0, \mathbf{x}_i(0) = \mathbf{0}$ and $\mathbf{y}_{ij}(0) = \mathbf{0}, \forall j \in \mathcal{N}_i$.
Repeat until convergence:
    $\mathbf{x}_i(s+1) \leftarrow (23)$,
    Send $\mathbf{E}_{ij}\mathbf{x}_i(s+1) - \frac{1}{\rho}\boldsymbol{\lambda}_{ij}(s)$ to all $j \in \mathcal{N}_i$,
    Receive $\mathbf{E}_{ji}\mathbf{x}_j(s+1) - \frac{1}{\rho}\boldsymbol{\lambda}_{ji}(s)$ from all $j \in \mathcal{N}_i$,
    $\mathbf{y}_i(s+1) \leftarrow (24), \forall j \in \mathcal{N}_i$,
    $\boldsymbol{\lambda}_{ij}(s+1) \leftarrow (25), \forall j \in \mathcal{N}_i$,
    $s \leftarrow s + 1$,
Finish algorithm: $\mathbf{z}_i^{output} = \text{blkrow}\{\mathbf{I}_{\bar{n}_i}, \mathbf{0}\}\mathbf{x}_i(s)$
```

Let $\mathbf{w}_i(s) = \text{col}\Big\{\mathbf{x}_i(s), \text{col}\big\{\mathbf{y}_{ij}(s), \boldsymbol{\lambda}_{ij}(s) : j \in \mathcal{N}_i\big\}\Big\}$ be the stacked vector corresponding to the estimated solution of the agent $i \in \mathcal{V}$ at the iteration step s in the update (23-25). Based on the convergence results of the proximal ADMM algorithm in [17], [19], we have the following theorem:

Theorem 1: Let every agent $i \in \mathcal{V}$ apply Algorithm 1. The estimated solution $\mathbf{w}(s) = \text{col}\{\mathbf{w}_i(s) : i \in \mathcal{V}\}$ converges exponentially to $\mathbf{w}^\infty$ where $\mathbf{w}^\infty = \text{col}\{\mathbf{w}_i^\infty : i \in \mathcal{V}\}$ is one optimal solution of the optimization problem (18).

Proof: See [18]. ■

According to Theorem 1, $\bar{\mathbf{z}}_i(s) = \text{blkrow}\{\mathbf{I}_{\bar{n}_i}, \mathbf{0}\}\mathbf{x}_i(s)$ converges exponentially to $\bar{\mathbf{z}}_i^\infty = \text{blkrow}\{\mathbf{I}_{\bar{n}_i}, \mathbf{0}\}\mathbf{x}_i^\infty, \forall i \in \mathcal{V}$. Lemma 1 guarantees that the stacked vector $\bar{\mathbf{z}}_i^\infty, \forall i \in \mathcal{V}$, consists of parts corresponding to the agent i in one least square solution of the linear algebraic equation (1).

V. Numerical simulations

This section considers a distributed linear algebraic equation, which can be represented in the form of (10-11), over a network of 24 agents. Assume that each agent has a local variables vector $\bar{\mathbf{z}}_i \in \mathbb{R}^{20}$, $\forall i \in \{1, 2, \dots, 24\}$, and there is one coupling row partition across all agents, i.e., $\mathbb{M} = \{\epsilon_1\}$ and $\mathcal{R}_{\epsilon_1} = \{1, 2, \dots, 24\}$. Let the communication graph be given in Fig. 1 and $\bar{\mathbf{B}}_{i,\epsilon_1} = \begin{bmatrix}\mathbf{0}_{5\times 15} & \mathbf{I}_5\end{bmatrix}, \forall i \in \{1, 2, \dots, 24\}$. Every vector $\bar{\mathbf{b}}_{i,\epsilon_1}$ is chosen randomly. In addition, we assume that the first 5 elements in local variable vectors $\bar{\mathbf{z}}_i, \forall i \in \{1, 2, \dots, 24\}$, are common variables.

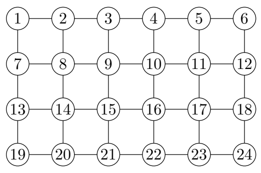

Fig. 1: Communication graph: nodes represent agents and edges represent available communication among agents.

We use MATLAB to test the effectiveness of our designed method in finding least square solutions to one hundred numerical examples. In each example, we choose $\mathbf{A}_i \in \mathbb{R}^{20\times 20}, \mathbf{a}_i \in \mathbb{R}^{20}$ randomly, $\forall i \in \{1, 2, \dots, 24\}$. The ADMM-based method (23-25) is run with the selection of $\rho = 1$. Fig. 2 shows the simulation results

$$\mathbf{x}_i(s+1) = \arg\min_{\mathbf{x}_i} \left(\frac{1}{2}\mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \mathbf{q}_i^T \mathbf{x}_i + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \left|\left| \mathbf{E}_{ij}\mathbf{x}_i - \mathbf{y}_{ij}(s) - \frac{1}{\rho}\boldsymbol{\lambda}_{ij}(s) \right|\right|^2 + \frac{1}{2}\left|\left| \mathbf{x}_i - \mathbf{x}_i(s) \right|\right|_{\mathbf{G}_i} \right), \forall i \in \mathcal{V}. \quad (21)$$

$$(\mathbf{y}_{ij}, \mathbf{y}_{ji})(s{+}1) = \arg\min_{\mathbf{y}_{ij}=\mathbf{y}_{ji}} \left(\left|\left| \mathbf{E}_{ij}\mathbf{x}_i(s+1) - \mathbf{y}_{ij} - \frac{1}{\rho}\boldsymbol{\lambda}_{ij}(s) \right|\right|^2 + \left|\left| \mathbf{E}_{ji}\mathbf{x}_j(s+1) - \mathbf{y}_{ji} - \frac{1}{\rho}\boldsymbol{\lambda}_{ji}(s) \right|\right|^2 \right), \forall (i,j) \in \mathcal{E}. \quad (22)$$

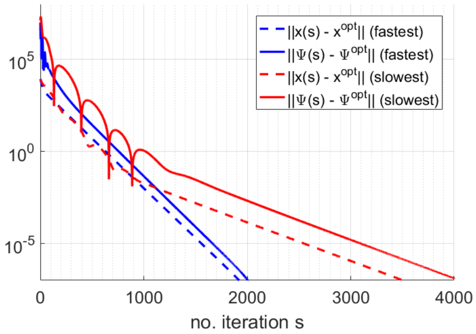

Fig. 2: The convergence to the least square solution of the second example. Algorithm 1 requires 370 iterations (23-25).

of two examples having the fastest (blue lines) and the slowest (red lines) convergence rates. In this figure, the solid lines represent the convergence of the estimated solution by Algorithm 1 to one optimal solution $\mathbf{x}^{opt}$ of the problem (18). The dashed lines illustrate to the evolution of the function $|\Psi(s) - \Psi^{opt}|$ where $\Psi(s) = \sum_{i=1}^{24} \left(\mathbf{x}_i(s)^T \mathbf{Q}_i \mathbf{x}_i(s) + \mathbf{q}_i^T \mathbf{x}_i(s) + \mathbf{a}_i^T \mathbf{a}_i + \mathbf{D}_i \mathbf{c}_i^T \mathbf{c}_i\right)$. It is easy to verify the exponential convergence of the estimated solution to the optimal one. A practical solution can be achieved after about 2000 iterations. In addition, we measure the computation time required by each agent in every iteration step. The maximum time required by one agent for 2000 iterations is 0.5 second. In [18], we provide more numerical simulations to compare the effectiveness of our designed method with some existing works.

VI. Conclusion

This paper considers a linear algebraic equation $\mathbf{Hz} = \mathbf{h}$ where rows and columns of the coefficient matrix $\mathbf{H}$ are dispensed across a network of agents. By introducing some virtual variables, the problem of finding one least square solution is reformulated into a distributed optimization problem. Then, we apply a proximal ADMM algorithm to design a distributed method for agents to solve the formulated optimization problem cooperatively. The estimated solution of every agent converges exponentially to its corresponding parts in one exact least square solution of the considered linear algebraic equation.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2022R1A2B5B03001459.

References

[1] P. Ma, “A new partially-coupled recursive least squares algorithm for multivariate equation-error systems,” *International Journal of Control, Automation and Systems*, vol. 21, no. 6, pp. 1828–1839, 2023.

[2] S. Liu, Y. Zhang, L. Xu, F. Ding, A. Alsaedi, and T. Hayat, “Extended gradient-based iterative algorithm for bilinear state-space systems with moving average noises by using the filtering technique,” *International Journal of Control, Automation and Systems*, vol. 19, no. 4, pp. 1597–1606, 2021.

[3] J. Liu, X. Chen, T. Basar, and A. Nedic, “A continuous-time distributed algorithm for solving linear equations,” in *Proc. of the 2016 American Control Conference (ACC)*, 2016.

[4] S. Mou, J. Liu, and A. S. Morse, “A distributed algorithm for solving a linear algebraic equation,” *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2863–2878, 2015.

[5] S. Mou, Z. Lin, L. Wang, D. Fullmer, and A. S. Morse, “A distributed algorithm for efficiently solving linear equations and its applications (special issue jcw),” *Systems & Control Letters*, vol. 91, pp. 21–27, 2016.

[6] O. Cihan, “Rapid solution of linear equations with distributed algorithms over networks,” *IFAC-PapersOnLine*, vol. 52, no. 25, pp. 467–471, 2019.

[7] J. Liu, A. S. Morse, A. Nedic, and T. Basar, “Exponential convergence of a distributed algorithm for solving linear algebraic equations,” *Automatica*, vol. 83, pp. 37–46, 2017.

[8] J. Liu, S. Mou, and A. S. Morse, “Asynchronous distributed algorithms for solving linear algebraic equations,” *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 372–385, 2018.

[9] G. Shi, B. D. O. Anderson, and U. Helmke, “Network flows that solve linear equations,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659–2674, 2017.

[10] X. Wang, J. Zhou, S. Mou, and M. J. Corless, “A distributed algorithm for least squares solutions,” *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4217–4222, 2019.

[11] Y. Liu, C. Lageman, B. O. D. Anderson, and G. Shi, “An arrow–hurwicz–uzawa type flow as least squares solver for network linear equations,” *Automatica*, vol. 100, pp. 187–193, 2019.

[12] T. Yang, J. George, J. Qin, X. Yi, and J. Wu, “Distributed least squares solver for network linear equations,” *Automatica*, vol. 113, 2020.

[13] M. Jahvani and M. Guay, “A distributed least-squares solver for linear systems of algebraic equations,” *IFAC-PapersOnLine*, vol. 55, no. 7, pp. 804–808, 2022.

[14] S. Li, W. Ai, J. Wu, and Q. Feng, “A fixed-time distributed algorithm for least square solutions of linear equations,” *International Journal of Control, Automation and Systems*, vol. 19, no. 3, pp. 1311–1318, 2021.

[15] P. Wang, Y. Cao, N. Yu, W. Ren, J. Lian, and D. Wu, “Distributed and communication-efficient solutions to linear equations with special sparse structure,” *System & Control Letters*, vol. 160, 2022.

[16] X. Wang, S. Mou, and B. O. D. Anderson, “Scalable, distributed algorithms for solving linear equations via double-layered networks,” *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1132–1143, 2020.

[17] B. He and X. Yuan, “On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers,” *Numer. Math.*, vol. 130, pp. 567 – 577, 2015.

[18] V. H. Pham and H.-S. Ahn, “Distributed least square solution method to linear algebraic equations over multiagent networks,” *arXiv preprint arXiv:2303.07884*, 2023.

[19] X. Pan, Z. Liu, and Z. Chen, “Linear convergence of admm under metric subregularity for distributed optimization,” *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2513–2520, April 2023.