

Scalable, Distributed Algorithms for Solving Linear Equations via Double-Layered Networks

Xuan Wang[✉], Shaoshuai Mou[✉], and Brian D. O. Anderson[✉]

Abstract—This paper proposes scalable, distributed algorithms for solving linear equations by integrating two mechanisms, termed consensus and conservation, in double-layered multiagent networks. The multiagent network considered in this paper is composed of clusters and each cluster consists of an aggregator and a sub-network of agents. By achieving consensus and conservation through agent-agent communications in the same cluster and aggregator-aggregator communications among different clusters, respectively, distributed algorithms are devised for agents to cooperatively achieve a solution to the overall linear equation. These algorithms outperform existing algorithms, including but not limited to the following aspects—first, each agent does not have to know as much as a complete row or column of the overall equation; second, each agent only needs to control as few as two scalar states when the number of clusters and the number of agents are sufficiently large; third, the dimensions of agents' states in the proposed algorithms do not have to be the same (while in contrast, algorithms based on the idea of standard consensus inherently require all agents' states to be of the same dimension). Both analytical proof and simulation results are provided to validate exponential convergence of the proposed distributed algorithms in solving linear equations.

Index Terms—Distributed algorithms, double-layered framework, linear equations, multiagent networks.

I. INTRODUCTION

DISTRIBUTED control of multiagent networks has recently received a significant amount of research attention, the goal of which is to accomplish global objectives through local coordinations [1]. *Consensus*, which drives all agents in the network to reach an agreement regarding a certain quantity [2], [3], has served as a basis in deriving many distributed

algorithms for optimization [4]–[8], synchronization of coupled oscillators [9], multirobot formation control [10], cooperative sensing [11], etc. Most recently, consensus has motivated distributed algorithms for solving linear algebraic equations [12]–[17], which achieves efficiency by decomposing a large system of linear equations into smaller ones that can be cooperatively solved by a network of agents. Compared with distributed equation solvers, such as Gaussian belief propagation [18] and Neumann series approximation [19], which require the linear equation to be either diagonally dominant or positive definite, the consensus-based distributed linear equation solvers [12]–[17] are applicable to a much larger class of linear equations. Elegant and powerful as the idea of consensus is, there has been rather limited application of consensus-based algorithms into situations when coordination among agents requires more than reaching consensus, especially when conservation requirements are involved. Different from consensus, *conservation* is a constraint that the sum of certain functions of agents' states needs to be constant [20]. Various types of conservation arise in many engineering applications, including conservation of resources in distributed allocation [21], conservation of total energy in controlling hybrid vehicles [22], conservation of flows in traffic control [23], conservation of linear and angular momentum in formation control [24], etc. Recognition of the potential of conservation in complementing consensus has motivated us to integrate both consensus and conservation together in one framework with the goal of combining their advantages together for achieving efficiency of distributed coordination.

One natural way to achieve such integration is by layered coordination [25], [26], which has been proven to be a powerful tool in many similar situations.

For example, practical tasks involving a large number of robots can be achieved by coordination in the planning layer, executive layer, and/or behavior layer [27]. Complicated optimization problems can be solved by coordination through layers, each of which iterates on its own subsets of decision variables using local information to achieve individual optimality [28]. Deep learning algorithms can be established by grouping neural nodes into multiple layers to achieve different functions including feature extraction, collection, comparison, and fusion [29]. Compared with single-layered networks, double-layered networks provide a natural description for quantifying the interconnectivity between different categories of connections [30], improve accuracy, [31] and efficiency [32]. This has motivated us to employ double-layered frameworks for the integration of consensus and conservation.

Manuscript received March 19, 2019; accepted May 11, 2019. Date of publication May 27, 2019; date of current version February 27, 2020. This work was supported by fundings from Northrop Grumman Corporation (NGC-REALM and NGCRC). The work of B. D. O. Anderson was supported in part by Data-61 CSIRO, Canberra, ACT 2601, Australia, and also with the Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 2600, Australia (e-mail: Brian.Anderson@anu.edu.au).

X. Wang and S. Mou are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA (e-mail: wang3156@purdue.edu; mous@purdue.edu).

B. D. O. Anderson is with Hangzhou Dianzi University, Hangzhou 310018, China, with Data-61 CSIRO, Canberra, ACT 2601, Australia, and also with the Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT 2600, Australia (e-mail: Brian.Anderson@anu.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2019.2919101

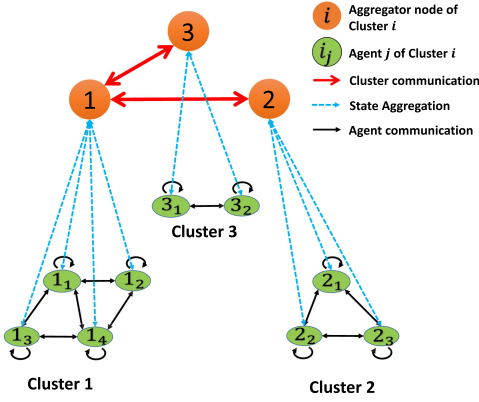


Fig. 1. Example of a double-layered multiagent network.

In this paper, we consider a double-layered multiagent network composed of clusters where each cluster consists of one aggregator and a network of agents (as will be detailed later in Fig. 1). Double-layered structures have played a significant role in distributed consensus [33]–[37] as well as distributed synchronization under communication delays or disturbances [38]–[41]. By selecting one or more agents as leaders from each cluster and sparsely connecting these leaders, clusters are able to coordinate for achieving unconstrained consensus [35]–[37]. This approach is, however, not directly applicable to developing distributed linear equation solvers, the goal of which is not only to achieve a consensus but a consensus subject to a group of linear constraints. Thus, we introduce one aggregator in each cluster, which has no computational burden but is able to collect and distribute information within the cluster. Information exchange between clusters is achieved through the upper-level network of aggregators. Such a technique of mixed use of heterogeneous agents has proved its efficiency in mobile communication networks [42]. The double-layered framework enables achieving two different types of coordination simultaneously—namely, one layer taking care of consensus while the other for conservation. This allows us to develop distributed algorithms for solving linear equations, which outperform existing distributed linear equation solvers [12]–[17], in a number of aspects, including but not limited to the following:

- 1) each agent does not have to know as much as a complete row or column of the overall equation;
- 2) each agent only needs to control as few as two scalar states when the number of clusters and the number of agents in each cluster (rather than in the whole network), are equal to the number of rows and columns of the overall equation, respectively;
- 3) the dimensions of agents' state vectors do not have to be the same, which is in contrast to algorithms based on the idea of standard consensus. These consensus-based algorithms inherently require all agents' states to be of the same dimension [12]–[16].

The rest of this paper is organized as follows. In Section II, we describe the structure and information flow of a double-layered framework, and formulate the problem of solving a linear equation via the structure. According to the particular choice of different goals in the cluster layer and the agent layer, we then

present two different types of distributed algorithms for solving linear equations in Sections III and IV, respectively. Both algorithms are distributed and converge exponentially fast. Analytical proofs and simulations are provided in these two sections. In Section V, we introduce a special case using a homogeneous partition of the linear equation, for which the proposed two algorithms can be simplified to yield the same update and can be implemented via a single-layered grid network. We finally conclude in Section VI and provide proofs of lemmas in Appendix.

Notation: Throughout this paper, we let $\mathbf{1}_r$ denote a vector in $\mathbb{R}^r$ with all its components equal to 1; let I_r denote the $r \times r$ identity matrix. Let M' , $\ker M$, and $\text{image } M$ denote the conjugate transpose, the kernel, and the image of a matrix M , respectively (most but not all matrices and vectors will be real). Let $\mathbf{x}_i = \text{col} \{x_{i1}, \dots, x_{ir}\}$ denote a column stack of vectors x_{ij} , $j = 1, \dots, r$. Let $\text{col} \{A_1, A_2, \dots, A_r\}$ denote a column stack of matrices A_i , $i = 1, 2, \dots, r$, which is $[A_1' A_2' \dots A_r']'$. Let $\text{diag} \{A_1, A_2, \dots, A_r\}$ denote the block diagonal matrix with A_i the i th diagonal block entry, $i = 1, 2, \dots, r$. Let $\otimes$ denote the Kronecker product.

II. PROBLEM FORMULATION

Consider a double-layer multiagent network consisting of a number of c clusters. Each cluster i is composed of one aggregator denoted by i and a number of c_i agents denoted by $i_1, i_2, \dots, i_{c_i}$. An aggregator is able to collect and distribute information with all agents in the same cluster. Further, we suppose each aggregator i is also able to receive information from certain other aggregators which are called i 's *aggregator-neighbors* and denoted by $\mathcal{N}_i$; also, we assume $i \in \mathcal{N}_i$. The neighbor relations of aggregators can be characterized by a c -node graph $\mathbb{G}$, in which there is an arc from $\bar{i}$ to i if and only if $\bar{i} \in \mathcal{N}_i$. Here, aggregators, which are not required to perform any computations, are introduced only for the purpose of information exchange among clusters including collecting information and distributing information within the cluster where the aggregator is located. In order to minimize the information overheads and the work load of aggregators, we do not require all information in a cluster to be shared with other clusters through communications between aggregators, as will be seen later, in Remarks 1 and 2. Within each cluster i , each agent i_j , $j = 1, \dots, c_i$, is able to receive information from certain agents, which are called agent i_j 's *agent-neighbors* denoted by $\mathcal{N}_{ij}$, specially $i_j \in \mathcal{N}_{ij}$. The neighbor relations within cluster i can be characterized by a c_i -node graph $\mathbb{G}_i$, in which there is an arc from $\bar{i}_j$ to i_j if and only if $\bar{i}_j \in \mathcal{N}_{ij}$. Suppose all $\mathbb{G}_i$, $i = 1, 2, \dots, c$ and $\mathbb{G}$ are connected and bidirectional.¹ One example of such a double-layer multiagent network is shown in Fig. 1. Consistently with the description above, we emphasize that no aggregator is a node of $\mathbb{G}_i$ and no agent is a node of $\mathbb{G}$.

Consider an overall linear equation

$$A\mathbf{x} = \mathbf{b}$$

where $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Suppose $A\mathbf{x} = \mathbf{b}$ has at least one solution. Suppose each agent i_j knows part of the overall

¹Here we use the term bidirectional instead of undirected to emphasize the two-way nature of information flows.

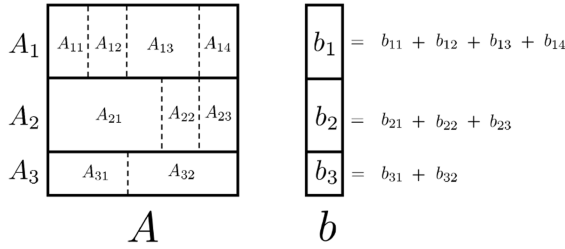


Fig. 2. Example of the relation between agents' locally available information and the overall equation for the network of Fig. 1. The various submatrices and subvectors do not have to be scalar.

linear equation (in accord with rules given in the next section), which might not be as much as a complete row or column of A . Let each agent i_j control a state vector $x_{ij}(t)$, while the cluster aggregators do not control any state and only play the role of providing communications. The problem of interest is to develop a distributed update for each agent such that all $x_{ij}(t)$ converge to constant vectors x_{ij}^* , $j = 1, 2, \dots, c_i$ and $i = 1, 2, \dots, c$, which jointly, via an arrangement set out in the next section, form a solution to $Ax = b$.

III. GLOBAL-CONSENSUS AND LOCAL-CONSERVATION

Suppose each agent i_j in a cluster i knows $A_{ij} \in \mathbb{R}^{m_i \times n_{ij}}$ and $b_{ij} \in \mathbb{R}^{m_i}$ such that the collection of them

$$[A_{i1} \ A_{i2} \ \dots \ A_{ic_i}] = A_i \in \mathbb{R}^{m_i \times n} \quad \sum_{j=1}^{c_i} b_{ij} = b_i \in \mathbb{R}^{m_i} \quad (1)$$

are a block row of the overall equation, where

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_c \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix}. \quad (2)$$

Here, b_i result from a partition of b . The b_{ij} could be any vectors that satisfy (1). One simple choice of such b_{ij} is obtained by setting one b_{ij} equal to b_i and the others equal to zero. In addition, one has

$$\sum_{j=1}^{c_i} n_{ij} = n, \quad i = 1, 2, \dots, c \quad \sum_{i=1}^c m_i = m. \quad (3)$$

Note that $n_{ij} = 1$ and $m_i = 1$ are permitted, but are of course not required. Consistent with the setup of Fig. 1, an example of how each agent's locally available information A_{ij}, b_{ij} is related to the overall equation $Ax = b$ is shown in Fig. 2.

Suppose each agent i_j controls a state vector $x_{ij}(t) \in \mathbb{R}^{n_{ij}}$. In this section, we aim to devise a distributed update for each agent i_j 's state $x_{ij}(t)$ to converge exponentially fast to a constant vector x_{ij}^* such that:

- 1) All x_{ij}^* , $j = 1, 2, \dots, c_i$ within each cluster i , $i = 1, 2, \dots, c$, satisfy the following:

$$\text{Local conservation: } \sum_{j=1}^{c_i} (A_{ij} x_{ij}^* - b_{ij}) = 0 \quad (4)$$

- 2) All $x_i^* = \text{col} \{x_{i1}^*, \dots, x_{ic_i}^*\}$, $i = 1, 2, \dots, c$, among all clusters in the network reach a consensus x^* , that is

$$\text{Global consensus: } x_1^* = x_2^* = \dots = x_c^* = x^*. \quad (5)$$

From (1) and (4) one has $A_i x_i^* = b_i$. This and the global consensus in (5) imply $Ax^* = b$. All x_{ij}^* satisfying the local conservation (4) and the global consensus (5) are said to form a *consensus-conservation solution* x^* to $Ax = b$. Note that however large the matrix A may be, it is always possible to conceive of a network structure in which the state vector size and the degree of every agent in each subnetwork are bounded independently of the size of A . This is a crucial scalability property.

A. Update

Let $x_i(t) \in \mathbb{R}^n$ denote a column collection of agent states in cluster i , $i = 1, 2, \dots, c$, that is

$$x_i(t) = \text{col} \{x_{i1}(t), \dots, x_{ic_i}(t)\}. \quad (6)$$

Aggregator i , through communication with the agents i_j maintains a copy of $x_i(t)$. This vector of course does depend on the size of A ; it is inevitable that if one is solving $Ax = b$ some part of the network must be designed to hold the solution x . In our case, all aggregators will hold the solution in the limit as t goes to infinity. To achieve a consensus-conservation solution x^* , it is sufficient to achieve $A_i x_i(t) = b_i$ while all $x_i(t)$, $i = 1, 2, \dots, c$, reach a consensus. In order to allow the aggregator to decompose the global consensus of $x_i(t)$ into relations involving agents' states, we let $E_{ij} \in \mathbb{R}^{n_{ij} \times n}$ denote a matrix consisting of rows from I_n such that $\text{col} \{E_{ij}, j = 1, 2, \dots, c_i\} = I_n$ and

$$A_{ij} = A_i E'_{ij}.$$

Then one has

$$x_{ij}(t) = E_{ij} x_i(t).$$

It follows that all $x_i(t)$ reaching a consensus is equivalent to requiring that $\forall i = 1, \dots, c$ and $\forall k \in \mathcal{N}_i$

$$x_{ij}(t) \rightarrow E_{ij} x_k(t). \quad (7)$$

To achieve the local conservation (4), one also introduces an additional *coordination state* $z_{ij}(t) \in \mathbb{R}^{m_i}$ associated with and stored by each agent i_j . As noted earlier, m_i may even be 1, and in any case, the inclusion of $z_{ij}(t)$ as a quantity managed by agent i_j does not destroy the scalability property associated with the arrangement, either by virtue of its dimension or, as will be seen below, by virtue of its calculation. Then we propose the following update for each agent i_j , $i = 1, 2, \dots, c$ and $j = 1, 2, \dots, c_i$:

$$\begin{aligned} \dot{x}_{ij} = & -A'_{ij} \left(A_{ij} x_{ij} - b_{ij} - \sum_{i_k \in \mathcal{N}_{ij}} (z_{ij} - z_{ik}) \right) \\ & - \sum_{k \in \mathcal{N}_i} (x_{ij} - E_{ij} x_k) \end{aligned} \quad (8)$$

$$\dot{z}_{ij} = A_{ij} x_{ij} - b_{ij} - \sum_{i_k \in \mathcal{N}_{ij}} (z_{ij} - z_{ik}) \quad (9)$$

where the first line of update (8) and (9) aim to achieve the local conservation (4) while the second line of update (8) aims to achieve the global consensus (5). One natural generalization to the proposed updates (8) and (9) is achievable by assigning different weights to controls for the local conservation and the global consensus, respectively, with the aim of achieving faster convergence. Achieving optimal choice of such weights might, however, require more than locally available information, which will not be discussed in this paper.

Remark 1: Note immediately that in implementation of (8) and (9), information about x_{ij} is only shared between clusters while information about z_{ij} is only shared among agents within the same cluster. That is, each aggregator i is able to access $x_k(t)$ for some k , depending on the graph $\mathbb{G}$, through aggregator communications. Of course, $x_k(t)$ is collected by its aggregator neighbor $k \in \mathcal{N}_i$; aggregator i distributes $E_{ij}x_k(t)$ to each agent i_j of cluster i . Note that $|\mathcal{N}_i|$ can be bounded independently of the size of A through the design of the aggregator network with graph $\mathbb{G}$. Within each cluster i , each agent i_j only needs to access its neighbors' coordination state $z_{ik}, i_k \in \mathcal{N}_{ij}$, through agent-agent communication in cluster i . Thus, the proposed updates (8) and (9) are obviously distributed in the sense that only communications among aggregator-neighbors and agent-neighbors are involved. Compared with existing consensus-based distributed linear equation solvers [12]–[16], distributed updates (8) and (9):

- 1) individual agents in the distributed solvers require much less knowledge of the overall equation and control states of much smaller dimension. For a given $A \in \mathbb{R}^{m \times n}$ of fixed size, each agent i_j knows $A_{ij} \in \mathbb{R}^{m_i \times n_{ij}}$ and $b_{ij} \in \mathbb{R}^{m_i}$, and controls states $x_{ij}(t) \in \mathbb{R}^{n_{ij}}$, and $z_{ij}(t) \in \mathbb{R}^{m_i}$. Sizes of all these locally available matrices and state vectors might change with respect to the number of clusters and the number of agents in each cluster (but as already noted, can be bounded independently of the size of A). To see why this is so, we note from (3) and partitions in Fig. 2 that increasing c and c_i leads to the decreases of m_i and n_{ij} , respectively. Specially, when the number of clusters is m and the number of agents within each cluster is n , that is, $c = m$ and $c_i = n$, each agent only needs to know two scalar entries $A_{ij} \in \mathbb{R}$, $b_{ij} \in \mathbb{R}$ and updates two scalar states—namely, $x_{ij}(t) \in \mathbb{R}$, $z_{ij}(t) \in \mathbb{R}$;
- 2) allow all agents' state vectors to be of different dimensions while in contrast consensus-based distributed linear equation solvers require all agents to control states of the same size. Thus the proposed updates might be applied in networks of heterogeneous agents with different capabilities of storage.

B. Main Result

Before proceeding, we first derive a compact form of (8) and (9). Toward this end, we let $z_i(t) \in \mathbb{R}^{c_i m_i}$ denote the column collection of all agents' coordination states in cluster i , that is

$$z_i(t) = \text{col} \{z_{i1}(t), \dots, z_{ic_i}(t)\} \quad i = 1, 2, \dots, c. \quad (10)$$

Let

$$\bar{A}_i = \text{diag} \{A_{i1}, \dots, A_{ic_i}\} \quad \bar{b}_i = \text{col} \{b_{i1}, \dots, b_{ic_i}\} \quad (11)$$

and

$$\bar{L}_{\mathbb{G}_i} = L_{\mathbb{G}_i} \otimes I_{m_i} \quad i = 1, 2, \dots, c \quad (12)$$

with $L_{\mathbb{G}_i}$ the Laplacian matrix of the c_i -node connected and bidirectional graph $\mathbb{G}_i$. Recalling $\text{col} \{E_{ij}, j = 1, 2, \dots, c_i\} = I_n$ and (6), one can write (8) and (9) as

$$\dot{x}_i = -\bar{A}'_i (\bar{A}_i x_i - \bar{b}_i - \bar{L}_{\mathbb{G}_i} z_i) - \sum_{k \in \mathcal{N}_i} (x_i - x_k) \quad (13)$$

$$\dot{z}_i = \bar{A}_i x_i - \bar{b}_i - \bar{L}_{\mathbb{G}_i} z_i \quad (14)$$

for $i = 1, 2, \dots, c$. Each equation pair in the above describes what is occurring at a particular cluster. Now let $x = \text{col} \{x_1, \dots, x_c\}$, $z = \text{col} \{z_1, \dots, z_c\}$

$$\hat{A} = \text{diag} \{\bar{A}_1, \dots, \bar{A}_c\} \quad \hat{b} = \text{col} \{\bar{b}_1, \dots, \bar{b}_c\}, \quad (15)$$

$$\hat{L} = \text{diag} \{\bar{L}_{\mathbb{G}_1}, \dots, \bar{L}_{\mathbb{G}_c}\} \quad \hat{L}_{\mathbb{G}} = L_{\mathbb{G}} \otimes I_n \quad (16)$$

with $L_{\mathbb{G}}$ Laplacian matrix of the c -node connected graph $\mathbb{G}$. Equations (13) and (14) can be further rewritten in the following compact form:

$$\dot{x} = -\hat{A}' (\hat{A} x - \hat{b} - \hat{L} z) - \hat{L}_{\mathbb{G}} x \quad (17)$$

$$\dot{z} = \hat{A} x - \hat{b} - \hat{L} z. \quad (18)$$

Or equivalently

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = Q \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \hat{A}' \hat{b} \\ -\hat{b} \end{bmatrix} \quad (19)$$

with

$$Q = \begin{bmatrix} -\hat{A}' \hat{A} - \hat{L}_{\mathbb{G}} & \hat{A}' \hat{L} \\ \hat{A} & -\hat{L} \end{bmatrix}. \quad (20)$$

To analyze the convergence of (19) we need the following lemma to characterize eigenvalues of Q .

Lemma 1: Let

$$M = \begin{bmatrix} -M'_1 M_1 - M_2 & M'_1 M_3 \\ M_1 & -M_3 \end{bmatrix}$$

where the M_i are real, $i = 1, 2, 3$, and M_2 and M_3 are positive semidefinite. Then all eigenvalues of M are real negative or zero. Moreover, if 0 is an eigenvalue of M , it must be nondefective.²

The proof of Lemma 1 will be given in the Appendix. By this lemma and by establishing the convergence of the linear time-invariant system (19) to a constant steady state, one has the following main result.

Theorem 1: Suppose $Ax = b$ has at least one solution, and the graphs $\mathbb{G}_i$, $i = 1, 2, \dots, c$, and $\mathbb{G}$ are connected and bidirectional. Then under the distributed updates (8), (9), all $x_{ij}(t)$ with $i = 1, 2, \dots, c$ and $j = 1, 2, \dots, c_i$ converge exponentially fast to constant vectors x_{ij}^* satisfying (4) and (5), which jointly form a consensus-conservation solution x^* to $Ax = b$.

²An eigenvalue is nondefective if and only if its algebraic multiplicity equals its geometric multiplicity. In other words, the Jordan block corresponding to a nondefective eigenvalue is diagonal.

Proof of Theorem 1: We first prove that there exists a constant vector $\text{col}\{\hat{x}, \hat{z}\}$ which is an equilibrium of (19). Recall there exists a constant vector $y \in \mathbb{R}^n$ such that $Ay = b$. From the definitions of A_{ij}, b_{ij} in (1) and (2) and E_{ij} , one has

$$\sum_{j=1}^{c_i} (A_{ij} E_{ij} y - b_{ij}) = 0 \quad i = 1, 2, \dots, c.$$

This equation and definitions of $\bar{A}_i, \bar{b}_i$ in (11) lead to

$$(\mathbf{1}'_{c_i} \otimes I_{m_i}) (\bar{A}_i y - \bar{b}_i) = 0 \quad i = 1, 2, \dots, c. \quad (21)$$

Note that $\bar{L}_{G_i} = L_{G_i} \otimes I_{m_i}$, where L_{G_i} is the Laplacian matrix of a c_i -node connected and bidirectional graph G_i . Then

$$\text{image } \bar{L}_{G_i} = \ker(\mathbf{1}'_{c_i} \otimes I_{m_i}) \quad i = 1, 2, \dots, c \quad (22)$$

From (21) and (22), one has

$$(\bar{A}_i y - \bar{b}_i) \in \text{image } \bar{L}_{G_i} \quad i = 1, 2, \dots, c. \quad (23)$$

Then there exists a constant vector $\hat{z}_i \in \mathbb{R}^{c_i m_i}$ such that

$$\bar{A}_i y - \bar{b}_i - \bar{L}_{G_i} \hat{z}_i = 0 \quad i = 1, 2, \dots, c. \quad (24)$$

Let $\hat{x} = \mathbf{1}_c \otimes y$. Note that $\hat{L}_G = L_G \otimes I_n$ with L_G the Laplacian matrix of the c -node connected and bidirectional graph G . Then

$$\hat{L}_G \hat{x} = 0. \quad (25)$$

Let $\hat{z} = \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_c\}$. From (24) and definitions of $\hat{A}, \hat{b}$ in (15), one has

$$\hat{A} \hat{x} - \hat{b} - \hat{L} \hat{z} = 0. \quad (26)$$

This equation and (25) imply that $\text{col}\{\hat{x}, \hat{z}\}$ is an equilibrium of (19).

Second, we analyze the convergence of the error

$$e(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} - \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix}. \quad (27)$$

From (19) and the fact that $\text{col}\{\hat{x}, \hat{z}\}$ is an equilibrium of (19), one has

$$\dot{e} = Qe. \quad (28)$$

From Lemma 1, the structure of Q in (20) and the fact that the Laplacian matrices $\hat{L}$ and $\hat{L}_G$ are symmetric and positive semidefinite, one concludes that all eigenvalues of Q are real negative or zero. Moreover, if zero is an eigenvalue of Q , it must be nondefective. Thus there exists a constant vector $q \in \ker Q$ such that $e(t)$ of the linear time-invariant system (28) converges to q exponentially fast[43]. Thus $\text{col}\{x(t), z(t)\}$ converges exponentially fast to a constant vector $\text{col}\{\hat{x}^*, \hat{z}^*\}$, where

$$\begin{bmatrix} \hat{x}^* \\ \hat{z}^* \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix} + q \quad q \in \ker Q. \quad (29)$$

Partition the constant vector $\hat{x}^*$ such that

$$\hat{x}^* = \text{col}\{x_1^*, \dots, x_c^*\} \quad (30)$$

where $x_i^* \in \mathbb{R}^n$ is further partitioned as

$$x_i^* = \text{col}\{x_{i1}^*, x_{i2}^*, \dots, x_{ic_i}^*\} \quad (31)$$

with $x_{ij}^* \in \mathbb{R}^{n_{ij}}$. Evidently, we have that $x_{ij}(t)$ converges to x_{ij}^* exponentially fast. In the following, one only needs to show that all these x_{ij}^* satisfy the local conservation in (4) and the global consensus in (5).

From (29) and the property that $\text{col}\{\hat{x}, \hat{z}\}$ is an equilibrium of (19), one concludes that $\text{col}\{\hat{x}^*, \hat{z}^*\}$ is also an equilibrium of (19). It follows that:

$$0 = -\hat{A}' (\hat{A} \hat{x}^* - \hat{b} - \hat{L} \hat{z}^*) - \hat{L}_G \hat{x}^* \quad (32)$$

$$0 = \hat{A} \hat{x}^* - \hat{b} - \hat{L} \hat{z}^*. \quad (33)$$

Partition $\hat{z}^* = \text{col}\{z_1^*, z_2^*, \dots, z_c^*\}$ with $z_i^* \in \mathbb{R}^{c_i m_i}$. From (33) and the definitions of $\hat{A}, \hat{b}, \hat{L}$ in (15) and (16), one has

$$\bar{A}_i x_i^* - \bar{b}_i - \bar{L}_{G_i} z_i^* = 0 \quad i = 1, 2, \dots, c. \quad (34)$$

From the definitions of $\bar{A}_i, \bar{b}_i, \bar{L}_{G_i}$ in (11) and (12), one can rewrite (34) as

$$\begin{bmatrix} A_{i1} x_{i1}^* \\ A_{i2} x_{i2}^* \\ \vdots \\ A_{ic_i} x_{ic_i}^* \end{bmatrix} - \begin{bmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{ic_i} \end{bmatrix} - (L_{G_i} \otimes I_{m_i}) z_i^* = 0. \quad (35)$$

Premultiplying by $\mathbf{1}'_{c_i} \otimes I_{m_i}$ on both sides of (35), one has

$$\sum_{j=1}^{c_i} (A_{ij} x_{ij}^* - b_{ij}) - [(\mathbf{1}'_{c_i} L_{G_i}) \otimes I_{m_i}] z_i^* = 0.$$

Since L_{G_i} is the Laplacian of c_i -node connected bidirectional graph G_i , one has $\mathbf{1}'_{c_i} L_{G_i} = 0$. Thus

$$\sum_{j=1}^{c_i} (A_{ij} x_{ij}^* - b_{ij}) = 0. \quad (36)$$

In addition, from (32) and (33), one has

$$\hat{L}_G \hat{x}^* = 0$$

where $\hat{L}_G = L_G \otimes I_m$ with L_G the Laplacian matrix of the c -node connected and bidirectional graph G . Thus there exists a constant vector x^* such that

$$\hat{x}^* = \mathbf{1}_c \otimes x^*. \quad (37)$$

Together with (30), this implies

$$x_1^* = x_2^* = \dots = x_c^* = x^* \quad (38)$$

with x_i^* a collection of x_{ij}^* as defined in (31). From (36) and (38) one concludes that all x_{ij}^* satisfy the local conservation in (4) and the global consensus in (5).

Therefore, the $x_{ij}(t)$ converge exponentially fast to constant vectors x_{ij}^* which form a consensus-conservation solution x^* to $Ax = b$. This completes the proof. ■

C. Simulation

We utilize the double-layer network as in Fig. 1 to solve the following linear equation $Ax = b$, which is partitioned

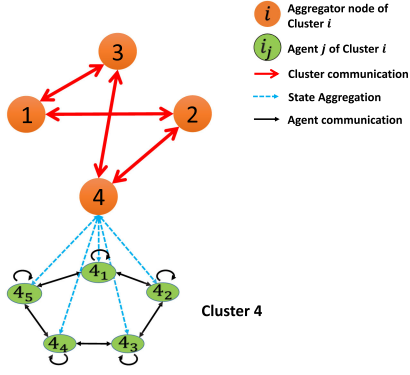


Fig. 3. Another double-layered multiagent network. For clarity, the details of agent communications are depicted for only one cluster.

according to the structure as in Fig. 2 with details as follows.

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 \\ 2 & -1 & 3 & -3 & 0 \\ -6 & 2 & 6 & 1 & -2 \\ 2 & -6 & 3 & 7 & 0 \\ 4 & -4 & 4 & 8 & 2 \end{bmatrix}, b = \begin{bmatrix} 8 \\ 8 \\ 11 \\ -17 \\ -9 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

Suppose each agent i_j knows A_{ij} and b_{ij} , and we employ the updates (8) and (9) with arbitrary initializations. Let

$$V(t) = \frac{1}{2} \sum_{i=1}^c \left\| \begin{bmatrix} x_{i1}(t) \\ \vdots \\ x_{ic_i}(t) \end{bmatrix} - x^* \right\|_2^2$$

where $x^* = [0.77 \ 2.79 \ 1.98 \ -1.10 \ 0.38]^T$ is a solution to $Ax = b$. Thus $V(t)$ measures the closeness of all agent states to forming a consensus-conservation solution. We then utilize the double-layer network as in Fig. 3, which results from adding one additional cluster to Fig. 1, to solve the same linear equation with partitions as follows (for clarity, the details of agent communications are depicted for only one cluster).

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 \\ 2 & -1 & 3 & -3 & 0 \\ -6 & 2 & 6 & 1 & -2 \\ 2 & -6 & 3 & 7 & 0 \\ -3 & -4 & 0 & -2 & 6 \end{bmatrix}, b = \begin{bmatrix} 8 \\ 8 \\ 11 \\ -17 \\ -9 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

Simulations in Fig. 4 suggests that $V(t)$ converges exponentially fast to 0, and thus, all $x_{ij}(t)$ converge exponentially fast to constant vectors that form a consensus-conservation solution x^* in both cases. This is in accord with Theorem 1. Moreover, the convergence rate can also be affected by the number of partitions and the network topology. Intuitively, more partitions of the overall linear equations require more clusters for implementing the proposed updates, which leads to slower convergence as suggested by Fig. 4.

IV. GLOBAL-CONSERVATION AND LOCAL-CONSENSUS

In the previous section, agents in the same cluster collectively know a block row of the overall matrix A as indicated in Fig. 2. In this section, we consider a different situation in which agents

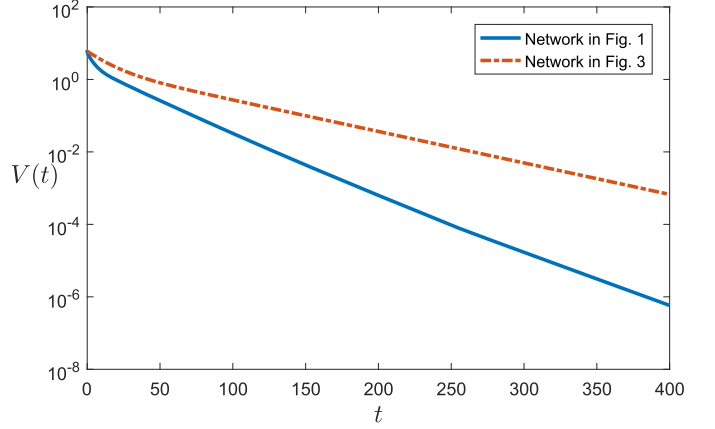


Fig. 4. Evolution of $V(t)$ under the proposed updates (8) and (9).

$$\begin{array}{c} A_1 \quad A_2 \quad A_3 \\ \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \\ A_{14} & A_{24} & A_{34} \end{bmatrix} \\ A \end{array} \quad \begin{array}{c} b \\ b = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \end{bmatrix} + \begin{bmatrix} b_{21} \\ b_{22} \\ b_{23} \\ b_{24} \end{bmatrix} + \begin{bmatrix} b_{31} \\ b_{32} \\ b_{33} \\ b_{34} \end{bmatrix} \\ b = b_1 + b_2 + b_3 \end{array}$$

Fig. 5. Example of the relation between agents' locally available information and the overall equation for the network of Fig. 1.

in the same cluster collectively know a block column of A , for which a different coordination arrangement will be required in the cluster-layer and the agent-layer, as will be shown later.

Suppose each agent i_j in cluster i knows $A_{ij} \in \mathbb{R}^{m_{ij} \times n_i}$, $b_{ij} \in \mathbb{R}^{m_{ij}}$ such that the collection of them

$$\begin{bmatrix} A_{i1} \\ A_{i2} \\ \vdots \\ A_{ic_i} \end{bmatrix} = A_i \in \mathbb{R}^{m \times n_i} \quad \begin{bmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{ic_i} \end{bmatrix} = b_i \in \mathbb{R}^m \quad (39)$$

are parts of the overall linear equation $Ax = b$, where

$$A = [A_1 \ A_2 \ \cdots \ A_c] \quad b = \sum_{i=1}^c b_i. \quad (40)$$

Note that while b is given, the b_i 's are not; they can be any vectors that satisfy (40) (a particular case is that one b_i equals b and the others are zero). Besides, one has

$$\sum_{j=1}^{c_i} m_{ij} = m, \quad i = 1, 2, \dots, c \quad \sum_{i=1}^c n_i = n. \quad (41)$$

An example of the relation between agents' locally available information A_{ij} , b_{ij} and the overall equation $Ax = b$ is shown in Fig. 5. Note that the symbols A_{ij} and b_i here are used differently from their use in previous sections but the notation is convenient.

Suppose each agent i_j controls a state vector $x_{ij}(t) \in \mathbb{R}^{n_i}$. In this section, we aim to devise a distributed update for each

agent i_j 's state $x_{ij}(t)$ to converge exponentially fast to a constant vector x_{ij}^* , $i = 1, 2, \dots, c$ and $j = 1, 2, \dots, c_i$, such that:

- 1) All x_{ij}^* , $j = 1, 2, \dots, c_i$, within each cluster i reach a consensus x_i^* , that is

$$\text{Local consensus: } x_{i1}^* = x_{i2}^* = \dots = x_{ic_i}^* = x_i^* \quad (42)$$

- 2) All x_i^* , $i = 1, 2, \dots, c$ among all clusters in the network satisfy the following:

$$\text{Global conservation: } \sum_{i=1}^c (A_i x_i^* - b_i) = 0. \quad (43)$$

Let $\mathbf{x}^* = \text{col} \{x_1^*, x_2^*, \dots, x_c^*\}$ be the column collection of the consensus value to which all agents in the same cluster converge. From (40) and (43) one has $A\mathbf{x}^* = b$. Thus the x_{ij}^* satisfying the local consensus (42) and the global conservation (43) are said to form a *conservation-consensus solution* $\mathbf{x}^*$ to $A\mathbf{x} = b$.

Note that here all x_{ij}^* in the same cluster are the same, which is a part of the solution to the overall equation, while, in contrast, in the consensus-conservation solution defined in the previous section, the x_{ij}^* in each cluster i jointly form a solution to the overall equation $A\mathbf{x} = b$.

A. Update

In order to achieve the global conservation, we employ the networks $\mathbb{G}$ and $\mathbb{G}_i$ described in Section II and introduce an additional state $z_{ij}(t) \in \mathbb{R}^{m_{ij}}$ at each agent i_j . Let $E_{ij} \in \mathbb{R}^{m_{ij} \times m}$ consist of rows of the identity matrix I_m such that $\text{col} \{E_{ij}, j = 1, 2, \dots, c_i\} = I_m$ and

$$A_{ij} = E_{ij} A_i.$$

Let $\mathbf{z}_i(t) \in \mathbb{R}^m$ denote the column of all coordination states in cluster i , $i = 1, 2, \dots, c$, that is

$$\mathbf{z}_i = \text{col} \{z_{i1}, z_{i2}, \dots, z_{ic_i}\}.$$

Then

$$z_{ij} = E_{ij} \mathbf{z}_i.$$

Suppose each cluster aggregator i is able to access its neighbor aggregator's coordination state $\mathbf{z}_k(t)$, $k \in \mathcal{N}_i$ through aggregator communication and then distributes $E_{ij} \mathbf{z}_k(t)$ to each agent i_j of cluster i . Within each cluster i , each agent i_j is able to access to its neighbors' state x_{ik} , $i_k \in \mathcal{N}_{ij}$, through agent-agent communication in cluster i . Then one proposes the following update for each agent i_j , $i = 1, 2, \dots, c$ and $j = 1, 2, \dots, c_i$:

$$\begin{aligned} \dot{x}_{ij} = & -A'_{ij} \left(A_{ij} x_{ij} - b_{ij} - \sum_{k \in \mathcal{N}_i} (z_{ij} - E_{ij} \mathbf{z}_k) \right) \\ & - \sum_{i_k \in \mathcal{N}_{ij}} (x_{ij} - x_{ik}) \end{aligned} \quad (44)$$

$$\dot{\mathbf{z}}_i = A_i \mathbf{z}_i - b_i - \sum_{k \in \mathcal{N}_i} (z_{ij} - E_{ij} \mathbf{z}_k) \quad (45)$$

where the first line of update (44) and (45) aim to achieve global conservation in (43) while the second line of (44) aims to achieve the local consensus in (42).

Remark 2: Note immediately that in implementation of (44) and (45), information about z_{ij} is only shared between clusters while information about x_{ij} is only shared among agents within the same cluster. Thus, each aggregator i is able to access $\mathbf{z}_k(t)$ through aggregator communications when $\mathbf{z}_k(t)$ is collected by its aggregator neighbor $k \in \mathcal{N}_i$; aggregator i distributes $E_{ij} \mathbf{z}_k(t)$ to each agent i_j of cluster i . Within each cluster i , each agent i_j only needs to access its neighbors' coordination state x_{ik} , $i_k \in \mathcal{N}_{ij}$, through agent-agent communication in cluster i . And thus the proposed updates (44) and (45) are obviously distributed in the sense that only communications among aggregator-neighbors and agent-neighbors are involved. Moreover, the agents i_j have storage and communication requirements which are fully scalable with the size of A . As before, the aggregator nodes will store vectors of dimension determined by the dimension of A . However, as already noted, the number of neighbors of an aggregator node does not need to grow with the dimension of A . Compared with existing consensus-based distributed linear equation solvers [12]–[16], distributed updates (8) and (9):

- 1) individual agents in the distributed solvers require much less knowledge of the overall equation and control states of much smaller dimension. For a given overall linear equation with $A \in \mathbb{R}^{m \times n}$, each agent i_j knows $A_{ij} \in \mathbb{R}^{m_{ij} \times n_i}$ and $b_{ij} \in \mathbb{R}^{m_{ij}}$, and controls states $x_{ij}(t) \in \mathbb{R}^{n_i}$, and $z_{ij}(t) \in \mathbb{R}^{m_{ij}}$. Sizes of these locally available matrices and state vectors could change with respect to the number of clusters and the number of agents in each cluster. From (41) and partitions in Fig. 5, one has that increasing c_i and c leads to the decreases of m_{ij} and n_i , respectively. Specially, when the number of clusters is n and the number of agents within each cluster is m , that is, $c = n$ and $c_i = m$, each agent only needs to know two scalar entries $A_{ij} \in \mathbb{R}$, $b_{ij} \in \mathbb{R}$ and updates two scalar states—namely, $x_{ij}(t) \in \mathbb{R}$, $z_{ij}(t) \in \mathbb{R}$;
- 2) allow all agents' state vectors to be of different dimensions, which is the same as the distributed updates (8) and (9) in the previous section.

B. Main Result

Before proceeding, we first derive a compact form of (44) and (45). Toward this end, we let $\mathbf{x}_i \in \mathbb{R}^{c_i n_i}$ denote the column collection of all agents' states in cluster i , $i = 1, 2, \dots, c$, that is

$$\mathbf{x}_i = \text{col} \{x_{i1}, x_{i2}, \dots, x_{ic_i}\}.$$

Let

$$\bar{A}_i = \text{diag} \{A_{i1}, \dots, A_{ic_i}\} \quad \bar{L}_{\mathbb{G}_i} = L_{\mathbb{G}_i} \otimes I_{n_i} \quad (46)$$

with $L_{\mathbb{G}_i}$ the Laplacian matrix of the c_i -node connected graph $\mathbb{G}_i$. From (44) and (45), and $\text{col} \{E_{ij}, j = 1, 2, \dots, c_i\} = I_m$, one has

$$\dot{\mathbf{x}}_i = -\bar{A}'_i \left(\bar{A}_i \mathbf{x}_i - b_i - \sum_{k \in \mathcal{N}_i} (\mathbf{z}_i - \mathbf{z}_k) \right) - \bar{L}_{\mathbb{G}_i} \mathbf{x}_i \quad (47)$$

$$\dot{\mathbf{z}}_i = \bar{A}_i \mathbf{x}_i - b_i - \sum_{k \in \mathcal{N}_i} (\mathbf{z}_i - \mathbf{z}_k) \quad (48)$$

for $i = 1, 2, \dots, c$. Let $\mathbf{x} = \text{col} \{\mathbf{x}_1, \dots, \mathbf{x}_c\}$ and $\mathbf{z} = \text{col} \{\mathbf{z}_1, \dots, \mathbf{z}_c\}$

$$\hat{A} = \text{diag} \{\bar{A}_1, \dots, \bar{A}_c\} \quad \hat{b} = \text{col} \{b_1, \dots, b_c\} \quad (49)$$

$$\hat{L} = \text{diag} \{\bar{L}_{\mathbb{G}_1}, \dots, \bar{L}_{\mathbb{G}_c}\} \quad \hat{L}_{\mathbb{G}} = L_{\mathbb{G}} \otimes I_m \quad (50)$$

with $L_{\mathbb{G}}$ the Laplacian matrix of the c -node connected graph $\mathbb{G}$. Equations (47) and (48) can be written in the following compact form:

$$\dot{\mathbf{x}} = -\hat{A}'(\hat{A}\mathbf{x} - \hat{b} - \hat{L}_{\mathbb{G}}\mathbf{z}) - \hat{L}\mathbf{x} \quad (51)$$

$$\dot{\mathbf{z}} = \hat{A}\mathbf{x} - \hat{b} - \hat{L}_{\mathbb{G}}\mathbf{z} \quad (52)$$

which is

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{z}} \end{bmatrix} = Q \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} + \begin{bmatrix} \hat{A}'\hat{b} \\ -\hat{b} \end{bmatrix} \quad (53)$$

with

$$Q = \begin{bmatrix} -\hat{A}'\hat{A} - \hat{L} & \hat{A}'\hat{L}_{\mathbb{G}} \\ \hat{A} & -\hat{L}_{\mathbb{G}} \end{bmatrix}. \quad (54)$$

Theorem 2: Suppose $Ax = b$ has at least one solution, and the graphs $\mathbb{G}_i$, $i = 1, 2, \dots, c$ and $\mathbb{G}$ are connected and bidirectional. Then under the distributed updates (44) and (45), all $x_{ij}(t)$ with $i = 1, 2, \dots, c$ and $j = 1, 2, \dots, c_i$ converge exponentially fast to constant vectors x_{ij}^* which satisfy the local consensus (42) and the global conservation (43), and thus, form a conservation-consensus solution $\mathbf{x}^*$ to $Ax = b$.

Proof of Theorem 2: We first prove that there exists a constant vector $\text{col} \{\hat{\mathbf{x}}, \hat{\mathbf{z}}\}$ which is an equilibrium of (53). Since there exists a constant vector $\mathbf{y} \in \mathbb{R}^n$ such that $A\mathbf{y} = b$, using the definition of A_i, b_i in (40), one has

$$\begin{bmatrix} A_1 & A_2 & \dots & A_c \end{bmatrix} \mathbf{y} = \sum_{i=1}^c b_i.$$

Partition $\mathbf{y} = \text{col} \{\mathbf{y}_1, \dots, \mathbf{y}_c\}$ with $\mathbf{y}_i \in \mathbb{R}^{n_i}$. Then one has

$$\sum_{i=1}^c (A_i \mathbf{y}_i - b_i) = 0.$$

It follows from this and (39) that:

$$\sum_{i=1}^c \left(\begin{bmatrix} A_{i1} \\ \vdots \\ A_{ic_i} \end{bmatrix} \mathbf{y}_i - \begin{bmatrix} b_{i1} \\ \vdots \\ b_{ic_i} \end{bmatrix} \right) = 0.$$

This and the definitions of $\bar{A}_i$ in (46) imply

$$\sum_{i=1}^c (\bar{A}_i \mathbf{y}_i - b_i) = 0 \quad (55)$$

with $\mathbf{y}_i = \mathbf{1}_{c_i} \otimes \mathbf{y}_i$. Let $\hat{\mathbf{x}} = \text{col} \{\mathbf{y}_1, \dots, \mathbf{y}_c\}$. From (55) and the definitions of $\hat{A}, \hat{b}$ in (49), one has

$$(\mathbf{1}'_c \otimes I_m) (\hat{A}\hat{\mathbf{x}} - \hat{b}) = 0. \quad (56)$$

Recall that $\hat{L}_{\mathbb{G}} = L_{\mathbb{G}} \otimes I_m$ with $L_{\mathbb{G}}$ the Laplacian matrix of a c -node connected, bidirectional graph $\mathbb{G}$. Then

$$\text{image } \hat{L}_{\mathbb{G}} = \ker (\mathbf{1}'_c \otimes I_m) \quad (57)$$

which with (56) implies

$$(\hat{A}\hat{\mathbf{x}} - \hat{b}) \in \text{image } \hat{L}_{\mathbb{G}}. \quad (58)$$

Then there exists a constant vector $\hat{\mathbf{z}}$ such that

$$\hat{A}\hat{\mathbf{x}} - \hat{b} - \hat{L}_{\mathbb{G}}\hat{\mathbf{z}} = 0. \quad (59)$$

In addition, since $\bar{L}_{\mathbb{G}_i} = L_{\mathbb{G}_i} \otimes I_{n_i}$ with $L_{\mathbb{G}_i} \otimes I_{n_i}$ the Laplacian matrix of the c_i -node connected graph $\mathbb{G}_i$, and since $\mathbf{y}_i = \mathbf{1}_{c_i} \otimes \mathbf{y}_i$, one has

$$\bar{L}_{\mathbb{G}_i} \mathbf{y}_i = 0, i = 1, 2, \dots, c.$$

Then because $\hat{L} = \text{diag} \{\bar{L}_{\mathbb{G}_1}, \dots, \bar{L}_{\mathbb{G}_c}\}$ and $\hat{\mathbf{x}} = \text{col} \{\mathbf{y}_1, \dots, \mathbf{y}_c\}$, one has

$$\hat{L}\hat{\mathbf{x}} = 0 \quad (60)$$

together with (59), this implies that $\text{col} \{\hat{\mathbf{x}}, \hat{\mathbf{z}}\}$ is an equilibrium of (53).

Second, we analyze the convergence of the error

$$\mathbf{e}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}} \end{bmatrix}. \quad (61)$$

From (53) and the fact that $\text{col} \{\hat{\mathbf{x}}, \hat{\mathbf{z}}\}$ is an equilibrium of (53), one has

$$\dot{\mathbf{e}} = Q\mathbf{e}. \quad (62)$$

From Lemma 1, the structure of Q in (54) and the fact that Laplacian matrices $\hat{L}$ and $\hat{L}_{\mathbb{G}}$ are symmetric and positive semidefinite, one has all eigenvalues of Q are real negative or 0. Moreover, if 0 is an eigenvalue of Q , it must be nondefective. Thus there exists a constant vector $\mathbf{q} \in \ker Q$ such that $\mathbf{e}(t)$ of the linear time-invariant error system (62) converges to $\mathbf{q}$ exponentially fast[43]. Thus $\text{col} \{\mathbf{x}(t), \mathbf{z}(t)\}$ converges exponentially fast to a constant vector $\text{col} \{\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*\}$, where

$$\begin{bmatrix} \hat{\mathbf{x}}^* \\ \hat{\mathbf{z}}^* \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}} \end{bmatrix} + \mathbf{q} \quad \mathbf{q} \in \ker Q. \quad (63)$$

Partition the constant vector $\hat{\mathbf{x}}^*$ such that

$$\hat{\mathbf{x}}^* = \text{col} \{\bar{\mathbf{x}}_1^*, \dots, \bar{\mathbf{x}}_c^*\} \quad (64)$$

where $\bar{\mathbf{x}}_i^* \in \mathbb{R}^{c_i n_i}$ is further partitioned as

$$\bar{\mathbf{x}}_i^* = \text{col} \{x_{i1}^*, x_{i2}^*, \dots, x_{ic_i}^*\} \quad (65)$$

with $x_{ij}^* \in \mathbb{R}^{n_i}$. Evidently, $x_{ij}(t)$ converges to x_{ij}^* exponentially fast. In the following, one only needs to show that all these x_{ij}^* satisfy the local consensus (42) and the global conservation (43).

From (63) and the property that $\text{col} \{\hat{\mathbf{x}}, \hat{\mathbf{z}}\}$ is an equilibrium of (53), one has $\text{col} \{\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*\}$ is an equilibrium of (53). Then

$$0 = -\hat{A}'(\hat{A}\hat{\mathbf{x}}^* - \hat{b} - \hat{L}_{\mathbb{G}}\hat{\mathbf{z}}^*) - \hat{L}\hat{\mathbf{x}}^* \quad (66)$$

$$0 = \hat{A}\hat{\mathbf{x}}^* - \hat{b} - \hat{L}_{\mathbb{G}}\hat{\mathbf{z}}^*. \quad (67)$$

It follows that:

$$\hat{L}\hat{x}^* = 0. \quad (68)$$

From this, (64) and the definition of $\hat{L}$, one has

$$\bar{L}_{G_i} \bar{x}_i^* = 0 \quad i = 1, 2, \dots, c. \quad (69)$$

Note that $\bar{L}_{G_i} = L_{G_i} \otimes I_{n_i}$ with L_{G_i} the Laplacian matrix of a connected bidirectional graph G_i . Then there must be a constant vectors $x_i^* \in \mathbb{R}^{n_i}$ such that

$$\bar{x}_i^* = \mathbf{1}_{c_i} \otimes x_i^*. \quad (70)$$

This and (65) imply

$$x_{i1}^* = x_{i2}^* = \dots = x_{ic_i}^* = x_i^* \quad (71)$$

for $i = 1, 2, \dots, c$.

From (70), the partition of A_i in (39) and $\bar{A}_i = \text{diag}\{A_{i1}, \dots, A_{ic_i}\}$ in (46), one has

$$\bar{A}_i \bar{x}_i^* = A_i x_i^*. \quad (72)$$

From (67) and the definitions of $\hat{A}, \hat{b}, \hat{L}_G$, one has

$$\begin{bmatrix} \bar{A}_1 \bar{x}_1^* \\ \bar{A}_2 \bar{x}_2^* \\ \vdots \\ \bar{A}_c \bar{x}_c^* \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix} - (L_G \otimes I_m) \hat{z} = 0. \quad (73)$$

This equality and (72) imply

$$\begin{bmatrix} A_1 x_1^* \\ A_2 x_2^* \\ \vdots \\ A_c x_c^* \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_c \end{bmatrix} - (L_G \otimes I_m) \hat{z} = 0. \quad (74)$$

Premultiplying by $\mathbf{1}'_c \otimes I_m$ on both sides of (74), one has

$$\sum_{i=1}^c (A_i x_i^* - b_i) - [(\mathbf{1}'_c L_G) \otimes I_m] \hat{z} = 0. \quad (75)$$

Note that L_G is the Laplacian matrix of a c -node connected and bidirectional graph G , one has $\mathbf{1}'_c L_G = 0$. Thus

$$\sum_{i=1}^c (A_i x_i^* - b_i) = 0. \quad (76)$$

From (71) and (76), one sees all x_{ij}^* satisfy the local consensus (42) and the global conservation (43). Therefore all $x_{ij}(t)$ converge to constant vectors, and thus, form a conservation-consensus solution x^* to $Ax = b$. This completes the proof. ■

C. Simulations

We utilize the double-layer network as in Fig. 1 to solve the linear equation $Ax = b$, which is partitioned

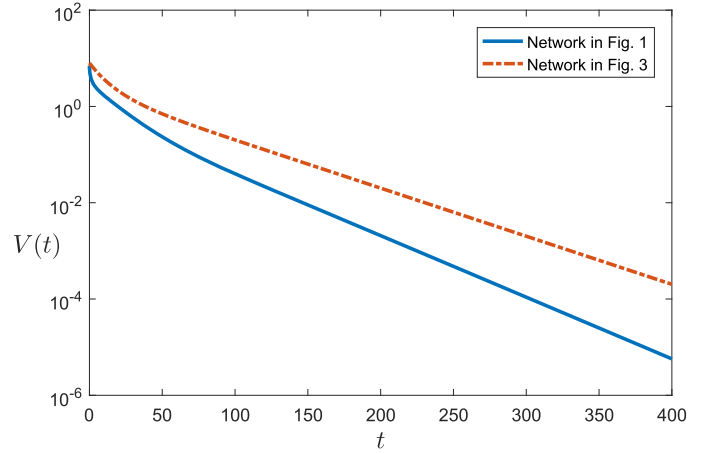


Fig. 6. Evolution of $V(t)$ under distributed updates (44) and (45).

according to the structure in Fig. 5 with details as follows.

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 \\ 2 & -1 & 3 & -3 & 0 \\ -6 & 2 & 6 & 1 & -2 \\ 2 & -6 & 3 & 7 & 0 \\ 4 & -4 & 4 & 8 & 2 \end{bmatrix}, b = \begin{bmatrix} 8 \\ 8 \\ 11 \\ -17 \\ -9 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \\ -5 \\ -5 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \\ 5 \\ -9 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \\ -3 \\ -3 \end{bmatrix}$$

Suppose each agent i_j knows A_{ij} and b_{ij} , and we employ the updates (44) and (45) with arbitrary initializations. Let

$$V(t) = \frac{1}{2} \sum_{i=1}^c \sum_{j=1}^{c_i} \|x_{ij}(t) - x_i^*\|_2^2$$

where $x^* = [0.77 \ 2.79 \ 1.98 \ -1.10 \ 0.38]'$ is a solution to $Ax = b$. Thus, $V(t)$ measures the closeness of all agent states to forming a conservation-consensus solution. We again utilize the double-layer network as in Fig. 3, with the following partition.

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 \\ 2 & -1 & 3 & -3 & 0 \\ -6 & 2 & 6 & 1 & -2 \\ 2 & -6 & 3 & 7 & 0 \\ -3 & -4 & 0 & -2 & 6 \end{bmatrix}, b = \begin{bmatrix} 8 \\ 8 \\ 11 \\ -17 \\ -9 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 5 \\ -9 \\ -4 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \\ 0 \\ -8 \\ -5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Simulations are shown in Fig. 6, which suggests that $V(t)$ converges exponentially fast to zero, and thus, all $x_{ij}(t)$ converge exponentially fast to constant vectors that form a conservation-consensus solution x^* in both cases. This is in accord with Theorem 2. Again, more partitions require more clusters for implementing the proposed updates, and this is expected to lead to slower convergence, as suggested by Fig. 6.

V. SIMPLIFIED NETWORK STRUCTURE UNDER HOMOGENEOUS PARTITION

In this section, we will consider a homogeneous partition of the overall equation $Ax = b$ as in Fig. 7 and will show that the two distributed algorithms developed in previous sections boil down to be identical; moreover, it can be implemented in a single-layered grid network G as in Fig. 8.

Let A be partitioned into r blocked rows and c blocked columns with each $A_{ij} \in \mathbb{R}^{m_i \times n_j}$ for $i = 1, 2, \dots, r$ and $j =$

$$\begin{array}{c}
 A_1 \\
 A_2 \\
 A_3
 \end{array}
 \begin{array}{|c|c|c|c|}
 \hline
 A_{11} & A_{12} & A_{13} & A_{14} \\
 \hline
 A_{21} & A_{22} & A_{23} & A_{24} \\
 \hline
 A_{31} & A_{32} & A_{33} & A_{34} \\
 \hline
 \end{array}
 \begin{array}{c}
 b_1 \\
 b_2 \\
 b_3
 \end{array}
 = \begin{array}{l}
 b_{11} + b_{12} + b_{13} + b_{14} \\
 b_{21} + b_{22} + b_{23} + b_{24} \\
 b_{31} + b_{32} + b_{33} + b_{34}
 \end{array}$$

$A \quad b$

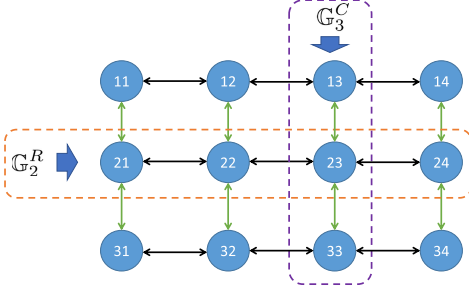
Fig. 7. Homogeneous partition of the equation, with $r = 3$, $c = 4$.

Fig. 8. Single-layered grid network without clusters or aggregators.

1, 2, ..., c ; correspondingly, the b vector is partitioned into sub-vectors b_{ij} such that $b = \text{col} \{b_1, \dots, b_r\} = b$ with $\sum_{j=1}^c b_{ij} = b_i$. One example of such a homogeneous partition is shown in Fig. 7 with $r = 3$ and $c = 4$. Consider a grid network $\mathbb{G}$ consisting of a number of rc agents labeled as i_j with $i = 1, 2, \dots, r$ and $j = 1, 2, \dots, c$, as in Fig. 8. Let $\mathbb{G}_i^R$ denote the i th row-subnetwork of $\mathbb{G}$, which consists of agents $i_1, i_2, \dots, i_c$ and all edges among these agents; and let $\mathbb{G}_j^C$ denote the j th column-subnetwork of $\mathbb{G}$, which consists of agents $1_j, 2_j, \dots, r_j$ and all edges among these agents. Suppose $\mathbb{G}_i^R, i = 1, 2, \dots, r$, and $\mathbb{G}_j^C, j = 1, 2, \dots, c$, are all undirected and connected. Suppose each agent i_j knows $A_{ij} \in \mathbb{R}^{m_i \times n_j}$ and $b_{ij} \in \mathbb{R}^{m_i}$, and controls a state vector $x_{ij}(t) \in \mathbb{R}^{n_i}$. The problem of interest in this section is to develop a distributed update for each $x_{ij}(t)$ converges exponentially fast to a constant x_{ij}^* such that

$$\text{Row-conservation: } \sum_{j=1}^c (A_{ij} x_{ij}^* - b_{ij}) = 0 \quad \forall i = 1, 2, \dots, r. \quad (77)$$

and

$$\text{Column-consensus: } x_{1j}^* = \dots = x_{rj}^* = x_j^* \quad \forall j = 1, 2, \dots, c. \quad (78)$$

Motivated by the two distributed updates developed in previous sections, we propose the following:

$$\begin{aligned}
 \dot{x}_{ij} = & -A'_{ij} \left(A_{ij} x_{ij} - b_{ij} - \sum_{ik \in \mathcal{N}_{ij}^R} (z_{ij} - z_{ik}) \right) \\
 & - \sum_{kj \in \mathcal{N}_{ij}^C} (x_{ij} - x_{kj})
 \end{aligned} \quad (79)$$

$$\dot{z}_{ij} = A_{ij} x_{ij} - b_{ij} - \sum_{ik \in \mathcal{N}_{ij}^R} (z_{ij} - z_{ik}) \quad (80)$$

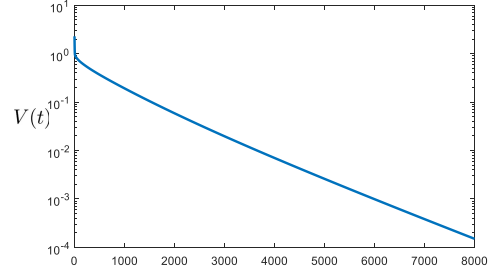


Fig. 9. Single-layered network without clusters or aggregators.

where $z_{ij} \in \mathbb{R}^{m_i}$ is an additional state vector introduced agent at i_j for achieving row-conservation; $\mathcal{N}_{ij}^R$ and $\mathcal{N}_{ij}^C$ denote the neighbor set of agent i_j in $\mathbb{G}_{ij}^R$ and $\mathbb{G}_{ij}^C$, respectively.

Note that the only difference between distributed updates (79) and (80), and distributed updates (8) and (9) is that $\sum_{k \in \mathcal{N}_{ij}^C} (x_{ij} - x_{kj})$ replaces $\sum_{k \in \mathcal{N}_i} (x_{ij} - E_{ij} x_k)$. By looking at each $\mathbb{G}_i^R$ in the single-layered grid as a cluster i , one sees that $E_{ij} x_k$ plays the same role as x_{kj} . Thus distributed updates (8) and (9) become the distributed updates (79) and (80) in the single-layered grid network. By Theorem 1, one has local conservations in (4) and the global consensus (5), which are equivalent to (77) and (78), respectively, in the single-layered grid network. It follows that $x^* = \text{col} \{x_1^*, \dots, x_c^*\}$ is a solution to $Ax = b$. Similar conclusion can also be drawn by looking at distributed updates (79) and (80) as a special case of (44) and (45). To sum up, one has

Corollary 1: Under the distributed updates (79) and (80) in a single-layered grid network, all $x_{ij}(t)$ with $i = 1, 2, \dots, r$; $j = 1, \dots, c$ converge exponentially fast to constant vectors x_{ij}^* satisfying (77) and (78), and thus, $x^* = \text{col} \{x_1^*, \dots, x_c^*\}$ is a solution to $Ax = b$.

Remark 3: The distributed updates in a single-layered grid network in this section can only be applied to the case that the overall linear equation is partitioned homogeneously and the number of agents is equal to the number of partitions. For more general partitions, one needs to go back to (8) and (9) or (44) and (45), for which dimensions of x_{ij} or z_{ij} are not the same.

According to the pattern of Fig. 7, we utilize a network of 100 agents with $r = c = 10$. Consider a random linear equation $Ax = b$, where $A \in \mathbb{R}^{1000 \times 1000}$, $b \in \mathbb{R}^{1000}$. The matrix A is partitioned into 100 blocks with each $A_{ij} \in \mathbb{R}^{10 \times 10}$, the vector b is partitioned accordingly with each $b_{ij} \in \mathbb{R}^{10}$. Define

$$V(t) = \frac{1}{2} \sum_{i=1}^{10} \sum_{j=1}^{10} \|x_{ij}(t) - x_i^*\|_2^2$$

where $x^* = \text{col} \{x_1^*, \dots, x_c^*\}$ is the true solution to the linear equation.

Simulations are shown in Fig. 9, which suggests that $V(t)$ converges exponentially fast to zero. This validates Corollary 1.

VI. CONCLUSION

This paper has devised distributed algorithms in a double-layered multiagent framework for solving linear equations, which consists of clusters and each cluster is composed of an aggregator and a network of agents. In these distributed

algorithms, each agent is not required to know as much as a complete row or column of the overall linear equation. Both analytical proof and simulation results are provided to validate exponential convergence. Future work includes generalization of the proposed algorithms to time-varying directed networks, application to achieving least square solutions, investigation of the impact when different weights are assigned to the conservation and consensus, and distributed algorithms in networks of more than two layers.

APPENDIX

Proof of Lemma 1: Let λ denote any eigenvalue of M with a nonzero eigenvector $\text{col}\{\mathbf{u}, \bar{\mathbf{u}}\}$. Then

$$M \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} \quad (81)$$

with

$$M = \begin{bmatrix} -M_1' M_1 - M_2 & M_1' M_3 \\ M_1 & -M_3 \end{bmatrix}.$$

Let $\bar{M} = \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix} M$. Then one has

$$\bar{M} = \begin{bmatrix} -M_1' M_1 - M_2 & M_1' M_3 \\ M_3' M_1 & -M_3' M_3 \end{bmatrix}$$

which can be written as

$$\bar{M} = - \begin{bmatrix} M_1' \\ -M_3' \end{bmatrix} \begin{bmatrix} M_1 & -M_3 \end{bmatrix} - \begin{bmatrix} M_2 & 0 \\ 0 & 0 \end{bmatrix}. \quad (82)$$

Thus $\bar{M}$ is negative semidefinite. Premultiplying by $\begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix}$ on both sides of (81), one has

$$\begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \bar{M} \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}. \quad (83)$$

First, we prove that λ must be real by contradiction. Suppose $\lambda = \alpha + \beta i$ where $\beta \neq 0$. Since $\bar{M}$ is negative semidefinite, then the imaginary part of the left-hand side of (83) is zero; so is the imaginary part of the right-hand side. It follows that:

$$\beta \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = 0.$$

Since $\beta \neq 0$ there follows $\mathbf{u}'\mathbf{u} + \bar{\mathbf{u}}'M_3'\bar{\mathbf{u}} = 0$. Recall that M_3 is positive semidefinite. Hence $\mathbf{u} = 0$, $M_3'\bar{\mathbf{u}} = 0$. Taken with (81) and noting $\lambda \neq 0$ since $\beta \neq 0$, one has $\bar{\mathbf{u}} = 0$. This and the assumption that $\mathbf{u} = 0$ contradict to the fact that $\text{col}\{\mathbf{u}, \bar{\mathbf{u}}\}$ is nonzero. Thus $\beta = 0$. Therefore, λ is real. From this, (83), $\bar{M}$ is negative semidefinite and M_3 is positive semidefinite, one further has $\lambda \leq 0$.

Second, if $\lambda = 0$ is an eigenvalue of M , we prove that it must be nondefective by contradiction. Suppose $\lambda = 0$ is defective, then there exists a nonzero vector $\text{col}\{\mathbf{v}, \bar{\mathbf{v}}\}$ such that

$$M \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} \quad (84)$$

and

$$M \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = 0. \quad (85)$$

Premultiplying by $\begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix}$ on both sides of (84), one has

$$\begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix}' \bar{M} \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} = (\mathbf{u}'\mathbf{u} + \bar{\mathbf{u}}'M_3'\bar{\mathbf{u}}). \quad (86)$$

Premultiplying by $\begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix}$ on both sides of (85), one has

$$\bar{M} \begin{bmatrix} \mathbf{u} \\ \bar{\mathbf{u}} \end{bmatrix} = 0. \quad (87)$$

This and the fact that $\bar{M}$ is symmetric imply that the left-hand side of (86) is zero. Then

$$(\mathbf{u}'\mathbf{u} + \bar{\mathbf{u}}'M_3'\bar{\mathbf{u}}) = 0 \quad (88)$$

from which, using the fact that M_3 is positive semidefinite, one has

$$\mathbf{u} = 0 \quad M_3'\bar{\mathbf{u}} = 0. \quad (89)$$

Premultiplying by $\begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix}' \begin{bmatrix} I & 0 \\ 0 & M_3' \end{bmatrix}$ on both sides of (84) one has

$$\begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix}' \bar{M} \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}'\mathbf{u} \\ \bar{\mathbf{v}}'M_3'\bar{\mathbf{u}} \end{bmatrix}.$$

The right-hand side is zero by (89). Thus

$$\begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix}' \bar{M} \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} = 0. \quad (90)$$

Together with (76), this yields

$$M_1\mathbf{v} - M_3\bar{\mathbf{v}} = 0 \quad M_2\mathbf{v} = 0. \quad (91)$$

From this and the definition of M , one has

$$M \begin{bmatrix} \mathbf{v} \\ \bar{\mathbf{v}} \end{bmatrix} = 0.$$

By (84), this yields $\text{col}\{\mathbf{u}, \bar{\mathbf{u}}\} = 0$, contradicting the assumption that $\text{col}\{\mathbf{u}, \bar{\mathbf{u}}\}$ is a nonzero eigenvector. Thus, $\lambda = 0$ is nondefective. ■

REFERENCES

- [1] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [2] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [3] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 575–600, 2008.
- [4] A. Nedic and A. Ozdaglar, "Distributed sub-gradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
- [5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.
- [6] T. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, Jun. 2014.
- [7] X. Zeng, P. Yi, and Y. Hong, "Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5227–5233, Oct. 2017.
- [8] Q. Liu, S. Yang, and Y. Hong, "Constrained consensus algorithms with fixed step size for distributed convex optimizations over multiagent networks," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4259–4265, Aug. 2017.

- [9] F. Dorfler, M. Chertkov, and F. Bullo, "Synchronization in complex oscillator networks and smart grids," *Proc. Nat. Acad. Sci.*, vol. 110, no. 6, pp. 2005–2010, 2013.
- [10] X. Chen, M. A. Belabbas, and T. Basar, "Controllability of formations over directed time-varying graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 4362–4367, Sep. 2017.
- [11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
- [12] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2863–2878, Nov. 2015.
- [13] 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)," *Syst. Control Lett.*, vol. 91, pp. 21–27, 2016.
- [14] B. D. O. Anderson, S. Mou, U. R. Helmke, and A. S. Morse, "Decentralized gradient algorithm for solution of a linear equation," *Numer. Algebr. Control Optim.*, vol. 6, no. 3, pp. 319–328, 2016.
- [15] X. Wang, S. Mou, and D. Sun, "Improvement of a distributed algorithm for solving linear equations," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3113–3117, Apr. 2017.
- [16] G. Shi, B. D. O. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2659–2674, Jun. 2017.
- [17] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed algorithm for least square solutions," *IEEE Trans. Autom. Control*, to be published. doi: 10.1109/TAC.2019.2894588.
- [18] O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev, "Gaussian belief propagation solver for systems of linear equations," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 1863–1867.
- [19] C. E. Lee, A. Ozdaglar, and D. Shah, "Asynchronous approximation of a single component of the solution to a linear system," 2014, arXiv:1411.2647.
- [20] X. Wang and S. Mou, "A distributed algorithm for achieving the conservation principle," in *Proc. Amer. Control Conf.*, 2018, pp. 5863–5867.
- [21] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 53–63, Sep. 2009.
- [22] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1242–1251, Nov. 2008.
- [23] M. Treiber and A. Kesting, *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin, Germany: Springer-Verlag, 2012.
- [24] Z. Sun, S. Mou, B. D. O. Anderson, and C. Yu, "Conservation and decay laws in distributed coordination control systems," *Automatica*, vol. 87, pp. 1–7, 2018.
- [25] M. D. Domenico *et al.*, "Mathematical formulation of multilayer networks," *Phys. Rev. X*, vol. 3, no. 4, 2013, Art. no. 041022.
- [26] S. Boccaletti *et al.*, "The structure and dynamics of multilayer networks," *Phys. Rep.*, vol. 544, no. 1, pp. 1–122, 2014.
- [27] R. Simmons *et al.*, "A layered architecture for coordination of mobile robots," in *Multi-Robot Systems: From Swarms to Intelligent Automata*, Dordrecht, Netherlands: Springer, 2002, pp. 103–112.
- [28] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.
- [29] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.
- [30] S. Gomez, A. Diaz-Guilera, J. Gomez-Gardenes, C. J. Perez-Vicente, Y. Moreno, and A. Arenas, "Diffusion dynamics on multiplex networks," *Phys. Rev. Lett.*, vol. 110, no. 2, 2013, Art. no. 028701.
- [31] J. M. Jeanne and R. I. Wilson, "Convergence, divergence, and reconvergence in a feedforward network improves neural speed and accuracy," *Neuron*, vol. 88, no. 5, pp. 1014–1026, 2015.
- [32] K. I. Tsianos and M. G. Rabbat, "Multiscale gossip for efficient decentralized averaging in wireless packet networks," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2137–2149, May 2013.
- [33] J. Chow and P. Kokotovic, "Time scale modeling of sparse dynamic networks," *IEEE Trans. Autom. Control*, vol. 30, no. 8, pp. 714–722, Aug. 1985.
- [34] I. C. Morărescu, S. Martin, A. Girard, and A. Muller-Gueudin, "Coordination in networks of linear impulsive agents," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2402–2415, Sep. 2016.
- [35] M. C. Bragagnolo, I.-C. Morărescu, J. Daafouz, and P. Riedinger, "Reset strategy for consensus in networks of clusters," *Automatica*, vol. 65, pp. 53–63, Mar. 2016.
- [36] E. Bıyık and M. Arcak, "Area aggregation and time-scale modeling for sparse nonlinear networks," *Syst. Control Lett.*, vol. 57, no. 2, pp. 142–149, Feb. 2008.
- [37] D. Romeres, F. Dörfler, and F. Bullo, "Novel results on slow coherency in consensus and power networks," in *Proc. Eur. Control Conf.*, Jul. 2013, pp. 742–747.
- [38] A. Hu, J. Cao, M. Hu, and L. Guo, "Cluster synchronization in directed networks of non-identical systems with noises via random pinning control," *Physica A: Statist. Mech. Appl.*, vol. 395, pp. 537–548, Feb. 2014.
- [39] Y. Wang and J. Cao, "Cluster synchronization in nonlinearly coupled delayed networks of non-identical dynamic systems," *Nonlinear Anal.: Real World Appl.*, vol. 14, no. 1, pp. 842–851, Feb. 2013.
- [40] C. Yang, J. Cao, T. Huang, J. Zhang, and J. Qiu, "Guaranteed cost boundary control for cluster synchronization of complex spatio-temporal dynamical networks with community structure," *Sci. China Inf. Sci.*, vol. 61, no. 5, May 2018, Art. no. 052203.
- [41] W. He, B. Zhang, Q. L. Han, F. Qian, J. Kurths, and J. Cao, "Leader-following consensus of nonlinear multiagent systems with stochastic sampling," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 327–338, Feb. 2017.
- [42] G. Mao, Z. Zhang, and B. D. O. Anderson, "Cooperative content dissemination and offloading in heterogeneous mobile networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6573–6587, Aug. 2016.
- [43] H. L. Trentelman, A. A. Stoorvogel, and M. Hautus, *Control Theory for Linear Systems*. London, U.K.: Springer-Verlag, 2012.



USA.



Xuan Wang received the B.S. and M.S. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively. Between February 2013 and July 2013, he was an exchange student with the College of Electrical Engineering & Computer Science, National Taiwan University, Taipei, Taiwan, China.

Currently, he is working toward the Ph.D. degree at the School of Aeronautics and Astronautics at Purdue University, West Lafayette, IN,

Shaoshuai Mou received the Ph.D. degree from Yale University, New Haven, CT, USA, in 2014.

After working as a Postdoctoral Associate with MIT, Cambridge, MA, USA, he joined Purdue University, West Lafayette, IN, USA, as an Assistant Professor in the School of Aeronautics and Astronautics in 2015. His research interests include distributed algorithms for control/optimizations/learning, multi-agent networks, UAV collaborations, perception and autonomy, resilience and cyber-security.



Brian D. O. Anderson received the Ph.D. in electrical engineering from Stanford University, Stanford, CA, USA, in 1966.

He is a Distinguished Professor with the Australian National University, Canberra, Australia, and a Distinguished Researcher with National ICT, Canberra, Australia.

Dr. Anderson was the recipient of the IEEE Control Systems Award of 1997, the 2001 IEEE James H. Mulligan, Jr. Education Medal, and the Bode Prize of the IEEE Control System Society in 1992, as well as several IEEE and other best paper prizes. He is a Fellow of the Australian Academy of Science, the Australian Academy of Technological Sciences and Engineering, the Royal Society, and a foreign member of the U.S. National Academy of Engineering. He holds honorary doctorates from a number of universities, including Université Catholique de Louvain, Belgium, and ETH Zurich, Zurich, Switzerland. He is a Past President of the International Federation of Automatic Control and the Australian Academy of Science.