A Distributed Algorithm for Efficiently Solving Linear Equations and Its Applications (Special Issue JCW)

S. Mou^{a,*}, Z. Lin^b, L. Wang^c, D. Fullmer^c, A. S. Morse^c

^a*School of Aeronautics and Astronautics, Purdue University, United States.*

^b*College of Electrical Engineering, Zhejiang University, China*

^c*Department of Electrical Engineering, Yale University, United States*

Abstract

A distributed algorithm is proposed for solving a linear algebraic equation $Ax = b$ over a multi-agent network, where $A \in \mathbb{R}^{\bar{n} \times n}$ and the equation has a unique solution $x^* \in \mathbb{R}^n$. Each agent knows only a subset of the rows of $[A \ b]$, controls a state vector $x_i(t)$ of size smaller than n and is able to receive information from its nearby neighbors. Neighbor relations are characterized by time-dependent directed graphs. It is shown that for a large class of time-varying networks, the proposed algorithm enables each agent to recursively update its own state by only using its neighbors' states such that all $x_i(t)$ converge exponentially fast to a specific part of x^* of interest to agent i . Applications of the proposed algorithm include solving the least square solution problem and the network localization problem.

Keywords: Distributed Algorithms; Least-Square Solution; Network Localization.

1. Motivations

A natural idea for solving large systems of linear algebraic equations is to decompose them into smaller ones which can then be solved by a

[☆]The research of A. S. Morse, L. Wang and D. Fullmer is supported by the US Air Force Office of Scientific Research and the National Science Foundation. The research of Z. Lin is supported by the National Natural Science Foundation of China under Grant 61273113.

*Corresponding author

Email addresses: mous@purdue.edu (S. Mou), linz@zju.edu.cn (Z. Lin), lili.wang@yale.edu (L. Wang), daniel.fullmer@yale.edu (D. Fullmer), as.morse@yale.edu (A. S. Morse)

multi-agent network. Since autonomous agents in a network are usually physically separated from each other, each agent can only communicate with its nearby neighbors. There are typically communication constraints on the information flow across a multi-agent network. This consequently precludes centralized processing and gives rise to distributed algorithms for solving linear equations $Ax = b$ simultaneously by m agents [1, 2, 3, 4], in each agent i knows $[A_i \ b_i]$ of the partitioned matrix

$$[A \ b] = \begin{bmatrix} A_1 & b_1 \\ A_2 & b_2 \\ \vdots & \vdots \\ A_m & b_m \end{bmatrix}, \quad A \in \mathbb{R}^{\bar{n} \times n} \quad (1)$$

current estimates of the equation's solution x^* generated by its neighbors, and nothing more. Rather than go through the intermediate step of reformulating the problem of solving linear equations an optimization problem [5, 6, 7, 8, 9] or a constrained consensus problem [10], the algorithm in [3] has tackled the problem directly based on a so-called "agreement principle", which, on the one hand limits each agent's state update to satisfy its own private equation and on the other causes all agents' states to reach a consensus. Implementing these algorithms requires each agent to transmit at each time step a vector of dimension at least n to each of its current neighbors. In certain applications, to be described in this paper, it is not necessary for each agent to determine all of the entries of x^* , especially when the matrix A have zero columns. There is the possibility that each agent may be able to determine its own favorite part of the entries of x^* by transmitting estimates of dimensions smaller than n . The aim of this paper is to present such an algorithm for solving the linear equation which requires less communication between agents than the algorithm in [3] by exploiting the special structure of A .

The rest of the paper is organized as follows: We formulate the problem of interest in Section 2 and present an algorithm to solve it in Section 3. Analysis to prove the effectiveness of the proposed algorithm is provided in Section 4. Two application examples are discussed in Section 5 and we conclude in Section 6. Proof for all propositions and lemmas is given in Section 7.

Notation: Let $\mathbf{m} = \{1, 2, \dots, m\}$. Let $\mathbf{1}_m$ denote the $m \times 1$ vector with all elements equal to 1. Let I_m denote the $m \times m$ identity matrix. Let I denote the identity matrix of a proper size. Let M' and $|M|$ denote the transpose

and the 2-norm of the matrix M , respectively. Throughout the paper, we let $\text{col}\{\cdots\}$ denote a column stack of all elements in it and let $\text{col}\{A_i, i \in \mathcal{R}\}$, where $\mathcal{R}$ is a set of positive integers, denote a stack of A_i with the index in a top-down ascending order. For two sets $\mathcal{R}_1$ and $\mathcal{R}_2$, we let $\mathcal{R}_1 \cap \mathcal{R}_2$ and $\mathcal{R}_1 \cup \mathcal{R}_2$ denote their intersection and union, respectively. Let $\mathcal{R}_1/\mathcal{R}_2$ denote the set by deleting all the elements belonging to $\mathcal{R}_2$ from $\mathcal{R}_1$. Let $\text{diag}\{A_1, A_2, \dots, A_m\}$ represent a block diagonal matrix with A_k as its k th diagonal block. Let $\mathcal{G}_{sa}$ denote the set of all directed graphs with m vertices which have self-arcs at all vertices. Let $\gamma(M)$ denote the graph of a square matrix $M \in \mathbb{R}^{m \times m}$ with nonnegative entries, which is an m vertex directed graph defined so that (j, i) is an arc from j to i in the graph just in case the ij th entry of M is non-zero.

2. Problem Formulation

Given a linear equation $Ax = b$, where $A \in \mathbb{R}^{\bar{n} \times n}$ and $b \in \mathbb{R}^{\bar{n}}$ can be partitioned as in (1), we suppose each A_i admits a block partition of the form $A_i = [A_{i1} \ A_{i2} \ \cdots \ A_{im}]$, $i \in \mathbf{m}$, where some of the $A_{ij} \in \mathbb{R}^{\bar{n}_i \times n_j}$ may be zero matrices. To represent the pattern of non-zero blocks in A , we let $\mathcal{R}_i$ and $\mathcal{C}_i$ denote the set of positive integers such that $A_{ij} \neq 0$ for each $j \in \mathcal{R}_i$ and $A_{ji} \neq 0$ for each $j \in \mathcal{C}_i$, respectively, $i \in \mathbf{m}$. For example, if

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & 0 & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad (2)$$

one has $\mathcal{R}_1 = \{1, 2\}$ and $\mathcal{C}_1 = \{1, 2, 3\}$. Without loss of generality, we assume that A is partitioned such that there are no zero block rows or columns. That is, $\mathcal{R}_i \neq \emptyset$ and $\mathcal{C}_i \neq \emptyset$ for all $i \in \mathbf{m}$. The existence of zero blocks in A_i leads to the following factorization

$$A_i = \bar{A}_i R_i. \quad (3)$$

Here $\bar{A}_i \in \mathbb{R}^{\bar{n}_i \times m_i}$ with $m_i = \sum_{k \in \mathcal{R}_i} n_k$ is resulted from deleting zero blocks from A_i ,

$$R_i = \text{col}\{E_k, k \in \mathcal{R}_i\} \in \mathbb{R}^{m_i \times n}$$

with

$$E_k = \begin{bmatrix} 0_{n_k \times \sum_{i=k-1} n_i} & I_{n_k \times n_k} & 0_{n_k \times \sum_{i=k+1}^m n_i} \end{bmatrix} \in \mathbb{R}^{n_k \times n}, \quad k \in \mathbf{m}.$$

Consider a network of m agents that are able to receive information from their “neighbors”. Here by a neighbor of agent i is meant any agent

within agent i 's reception range. We write $\mathcal{N}_i(t)$ for the labels of agent i 's neighbors at time t and we always take agent i as a neighbor of itself, that is, $i \in \mathcal{N}_i(t)$. Neighbor relations can be conveniently characterized by a directed graph $\mathbb{N}(t)$ with m vertices and a set of arcs defined so that there is an arc in $\mathbb{N}(t)$ from vertex j to i just in case that agent j is a neighbor of agent i .

Let x^* denote the unique solution to $Ax = b$. Corresponding to the pattern of zero-blocks in A , we partition $x^* = \text{col} \{x_1^*, x_2^*, \dots, x_m^*\}$ with $x_i^* \in \mathbb{R}^{n_i}$. Let $y_i = R_i x^*$, that is,

$$y_i = \text{col} \{x_k^*, k \in \mathcal{R}_i\}, \quad i \in \mathbf{m}. \quad (4)$$

Then y_i is a part of x^* , which may become agent i 's specific interest in certain situations. This leads us the following problem:

Problem 1: Suppose each agent i controls a state vector $x_i(t) \in \mathbb{R}^{m_i}$ to satisfy its own private function $\bar{A}_i x_i(t) = b_i$. Devise a local rule for each agent to update its own state only using state vectors from its neighbors such that all $x_i(t)$ converge exponentially fast to the constant vector y_i .

3. The Algorithm

In this section, we will present a distributed algorithm to solve Problem 1.

Suppose time is discrete in that t takes values in $\{1, 2, \dots\}$ and all agents operate synchronously. At $t = 1$, each agent i picks $x_i(1)$ to satisfy $\bar{A}_i x_i(1) = b_i$. For $t \geq 1$, we restrict the updating of $x_i(t)$ to iterations of the form

$$x_i(t+1) = x_i(t) + K_i u_i(t)$$

where the columns of K_i form a basis for the kernel of $\bar{A}_i$. Then no matter what $u_i(t)$ is, each $x_i(t)$ will obviously satisfy $\bar{A}_i x_i(t) = b_i$. If additionally there exists a constant vector x^* such that

$$x_i(t) = R_i x^*, \quad i \in \mathbf{m}, \quad (5)$$

one by $A_i = \bar{A}_i R_i$ has $A_i x^* = b_i$ for all $i \in \mathbf{m}$. Then $Ax^* = b$.

From the definitions of E_k and R_i , one notes that the k th sub-state of $R'_i x_i(t)$ is $E_k R'_i x_i(t)$. To show there exists a constant vector x^* satisfying (5), one only needs to show that there exists a constant vector x^* such that

$$E_k R'_i x_i(t) = E_k x^*, \quad k \in \mathcal{R}_i, \quad i \in \mathbf{m}. \quad (6)$$

To accomplish this it suffices to show

$$E_k R'_i x_i(t) = E_k R'_j x_j(t), \quad j \in \mathcal{N}_i(t) \cap \mathcal{C}_k, \quad k \in \mathcal{R}_i, \quad i \in \mathbf{m}. \quad (7)$$

This is clearly a consensus problem. which can be solved by choosing $x_i(t+1)$ such that

$$E_k R'_i x_i(t+1) = \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k R'_j x_j(t), \quad k \in \mathcal{R}_i$$

where the $w_{ijk}(t)$ are called *weights* chosen to be such that $w_{ijk}(t) \geq 0$ for $t \geq 1$ and for all $i, j, k \in \mathbf{m}$; $w_{ijk}(t) \neq 0$ if and only if $k \in \mathcal{R}_i \cap \mathcal{R}_j$; and

$$\sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) = 1, \quad k \in \mathcal{R}_i, \quad i \in \mathbf{m}. \quad (8)$$

Inspired by the consensus literature [11, 12, 13, 14, 15], we choose $u_i(t)$ to minimize the square of the 2-norm of

$$\text{col} \{ E_k R'_i x_i(t+1) - \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k R'_j x_j(t), \quad k \in \mathcal{R}_i \}. \quad (9)$$

From $R_i = \text{col} \{ E_k, k \in \mathcal{R}_i \}$ and $R_i R'_i = I$, one has (9) is equivalent to

$$x_i(t+1) - v_i(t) \quad (10)$$

where

$$v_i(t) = \text{col} \{ \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k R'_j x_j(t), \quad k \in \mathcal{R}_i \} \quad (11)$$

Choosing $u_i(t)$ to minimize the square of the 2-norm of (10) leads to an update for agent i in the form of

$$x_i(t+1) = x_i(t) - P_i (x_i(t) - v_i(t)) \quad (12)$$

where P_i denotes the orthogonal projection matrix to the kernel of $\bar{A}_i$.

60 4. Main Result

In this section we will prove that the algorithm described in Section 3 solves Problem 1 by driving each $x_i(t)$ to converge exponentially fast to $R_i x^*$. First, we translate the state $x_i(t)$ to a new shift state

$$e_i(t) = x_i(t) - R_i x^*, \quad i \in \mathbf{m}, \quad t \geq 1,$$

which can be interpreted as the error between $x_i(t)$ and $R_i x^*$. From $\bar{A}_i R_i x^* = b_i$ and $\bar{A}_i x_i(t) = b_i$, one has $e_i(t) \in \text{Im } P_i$ because of $\text{Im } P_i = \ker \bar{A}_i$. Thus $P_i e_i(t) = e_i(t)$ since $P_i^2 = P_i$. By multiplying P_i and then adding $-P_i R_i x^*$ to both sides of (12), one has

$$e_i(t+1) = P_i (v_i(t) - R_i x^*), \quad t \geq 1. \quad (13)$$

For each $k \in \mathcal{R}_i$ and $j \in \mathcal{N}_i(t) \cap \mathcal{C}_k$, one has

$$E_k x^* = E_k R'_j R_j x^*$$

from which, $R_i = \text{col} \{E_k, k \in \mathcal{R}_i\}$ and (8), one has

$$\begin{aligned} R_i x^* &= \text{col} \{E_k x^*, k \in \mathcal{R}_i\} \\ &= \text{col} \left\{ \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k x^*, k \in \mathcal{R}_i \right\} \\ &= \text{col} \left\{ \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k R'_j R_j x^*, k \in \mathcal{R}_i \right\} \end{aligned}$$

which together with (11) implies

$$v_i(t) - R_i x^* = \text{col} \left\{ \sum_{j \in \mathcal{N}_i(t) \cap \mathcal{C}_k} w_{ijk}(t) E_k R'_j e_j(t), k \in \mathcal{R}_i \right\} \quad (14)$$

Note that for each $i \in m$ and $k \in \mathcal{R}_i$, one has $k \notin \mathcal{R}_j$ for $j \in \mathcal{N}_i(t)/\mathcal{C}_k$. Thus

$$w_{ijk}(t) = 0, \quad j \in \mathcal{N}_i(t)/\mathcal{C}_k$$

Then (14) can be rewritten as

$$\begin{aligned} v_i(t) - R_i x^* &= \text{col} \left\{ \sum_{j \in \mathcal{N}_i(t)} w_{ijk}(t) E_k R'_j e_j(t), k \in \mathcal{R}_i \right\} \\ &= \sum_{j \in \mathcal{N}_i(t)} \text{col} \{w_{ijk}(t) E_k, k \in \mathcal{R}_i\} R'_j e_j(t) \\ &= \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t) R_i R'_j e_j(t) \end{aligned}$$

where

$$W_{ij}(t) = \text{diag} \{w_{ij i_1}(t) I_{n_{i_1}}, w_{ij i_2}(t) I_{n_{i_2}}, \dots, w_{ij i_{r_i}}(t) I_{n_{i_{r_i}}}\}$$

where $\{i_1, i_2, \dots, i_{r_i}\} = \mathcal{R}_i$. Then (13) becomes

$$e_i(t+1) = P_i \sum_{j \in \mathcal{N}_i(t)} S_{ij}(t) e_j(t), \quad t \geq 1, \quad i \in \mathbf{m} \quad (15)$$

where

$$S_{ij}(t) = W_{ij}(t)R_iR'_j, \quad j \in \mathcal{N}_i.$$

Note that if $j \in \mathcal{N}_i(t)$ but $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, one has $R_iR'_j = 0$ and thus $S_{ij}(t) = 0$. For $j \in \mathcal{N}_i(t)$ but $\mathcal{R}_i \cap \mathcal{R}_j \neq \emptyset$, we suppose $\mathcal{R}_i \cap \mathcal{R}_j = \{s_1, s_2, \dots, s_q\}$ such
65 that $s_1, s_2, \dots, s_q$ are the k_1 th, k_2 th, ..., k_q th elements in $\mathcal{R}_i$, respectively, and the $\bar{k}_1$ th, $\bar{k}_2$ th, ..., $\bar{k}_q$ th elements in $\mathcal{R}_j$, respectively. It is easy to verify that $S_{ij}(t) \in \mathbb{R}^{m_i \times m_j}$ is a $r_i \times r_j$ block matrix such that its $k_1\bar{k}_1$ th, ..., $k_q\bar{k}_q$ th blocks are $w_{ijs_1}(t)I_{n_{s_1}}, \dots, w_{ijs_q}(t)I_{n_{s_q}}$ and all the other blocks are zeros.

As a second step, we combine these m update equations into one linear recursion equation with state vector $e(t) = \text{col} \{e_1(t), e_2(t), \dots, e_m(t)\}$. It is straightforward to verify from (15) that

$$e(t+1) = PS(t)e(t) \quad (16)$$

where

$$P = \text{diag} \{P_1, P_2, \dots, P_m\} \quad (17)$$

and $S(t)$ is an $m \times m$ block matrix with the ij th block equal to $S_{ij}(t)$ for
70 $j \in \mathcal{N}_i(t)$ and zero otherwise. Note that P is an orthogonal projection matrix. We will use this fact without special mentioning in the sequel.

To prove the proposed algorithm to solve Problem 1, it is sufficient to show that $e(t)$ converges exponentially fast to 0. Obviously conditions for such convergence depend on the connectivity of neighbor graphs. In the
75 following, we will provide such conditions for both the case that the neighbor graph $\mathbb{N}(t)$ is time-invariant and time-varying.

4.1. Time-invariant Neighbor Graphs

In this section we turn to the case that the neighbor graph $\mathbb{N}$ is time-invariant. We let $\mathbb{N}_i$ denote the subgraph of $\mathbb{N}$ with node set $\mathcal{C}_i$ and all the edges among nodes of $\mathcal{C}_i$, $i \in \mathbf{m}$. Let us agree to say a *bidirectional* graph is a graph in which i is a neighbor of j whenever j is an neighbor of i . For each $i_k \in \mathcal{R}_i$ and $i \in \mathbf{m}$, the non-zero weights defined in (8) are chosen to be fixed, that is,

$$w_{iji_k} = \begin{cases} 1 - \frac{d_{i_k}}{d_i}, & j = i_k, \\ \frac{1}{d_i}, & \text{otherwise,} \end{cases} \quad (18)$$

where d_{i_k} denotes the number of elements in $\mathcal{N}_i \cap \mathcal{C}_{i_k}$ for each $i_k \in \mathcal{R}_i$ and $d_i = \max\{d_{i_k}, i_k \in \mathcal{R}_i\}$. Note that the weights in (18) can be chosen only

80 using information available to agent i without requiring any cooperation among agents. The main result in the case of time-invariant graphs is the following theorem:

Theorem 1. *Suppose $Ax = b$ has a unique solution. Suppose each agent i updates its state $x_i(t)$ according to (12), where all the weights are fixed as in (18). If $\mathbb{N}$ is such that each $\mathbb{N}_i$, $i \in \mathbf{m}$, is bidirectional and strongly 85 connected, then all $x_i(t)$ converge exponentially fast to the constant vector y_i , where y_i is as defined in (4).*

To prove Theorem 1, it is sufficient to show the following proposition, proof of which will be provided in Section 7.

90 **Proposition 1.** *The spectral radius of PS is less than 1.*

4.2. Time-varying Neighbor Graphs

To establish conditions for convergence of $e(t)$ to 0 in the case that neighbor graphs $\mathbb{N}(t)$ are time-varying, we need the idea of graph composition. By the *composition* of a directed graph $\mathbb{G}_p \in \mathcal{G}_{sa}$ with a directed graph 95 $\mathbb{G}_q \in \mathcal{G}_{sa}$, written as $\mathbb{G}_q \circ \mathbb{G}_p$ is meant the directed graph in $\mathcal{G}_{sa}$ with arc set defined so that (i, j) is an arc in the composition just in case that there is a vertex k such that (i, k) is an arc in $\mathbb{G}_p$ and (k, j) is an arc in $\mathbb{G}_q$. Composition is defined so that for any pair of non-negative $m \times m$ matrices M_1, M_2 with graphs $\gamma(M_1), \gamma(M_2) \in \mathcal{G}_{sa}$, $\gamma(M_2 M_1) = \gamma(M_2) \circ \gamma(M_1)$.

For each neighbor graph $\mathbb{N}(t)$, we let $\mathbb{N}_i(t)$, $i \in \mathbf{m}$, denote its subgraph with the node set $\mathcal{C}_i$ and all the edges among nodes in $\mathcal{C}_i$. Let us agree to say that an infinite sequence of neighbor graphs $\mathbb{N}(1), \mathbb{N}(2), \dots$, are *locally repeatedly jointly strongly connected* if for some positive integer l and each integer $k \geq 1$, the composed graph

$$\mathbb{H}_{ik} = \mathbb{N}_i(kl) \circ \mathbb{N}_i(kl - 1) \circ \dots \circ \mathbb{N}_i(kl - l + 1) \quad (19)$$

100 is strongly connected for all $i \in \mathbf{m}$.

In order to guarantee convergence of $e(t)$ to 0, we also need to impose constraints on the choice of weights. We suppose that all the weights are chosen such that $S(t)$ is doubly stochastic. Although focus of this paper is not about how to obtain such a doubly stochastic $S(t)$, we are aware of 105 various ways to do so. If each $\mathbb{N}(t)$ is bidirectional, each agent usually needs to know at least the upper bound of the number of its neighbor's neighbors.

Additional cooperations among agents are usually required when it comes to directed graphs as illustrated in [16]. Now we are ready to give our main result as follows:

110 **Theorem 2.** *Suppose $Ax = b$ has a unique solution. Suppose each agent i updates its state $x_i(t)$ according to (12). If the sequence of neighbor graphs $\mathbb{N}(t)$, $t \geq 1$, are locally repeatedly jointly strongly connected, and the weights are chosen such that $S(t)$ is doubly stochastic for each $t \geq 1$, then all $x_i(t)$ converge exponentially fast to the constant vector y_i , where y_i is as defined*
 115 *in (4).*

To prove Theorem 2, it is sufficient to show that $e(t)$ converges to 0 exponentially fast. In view of (16), it is clear enough to show that the infinite matrix product

$$PS(t)PS(t-1) \cdots PS(1) \quad (20)$$

converges to the zero matrix exponentially fast as t goes to infinity. Toward this end, we break the infinite matrix product (20) into contiguous sub-products

$$W_k = PS(kl)PS(kl-1) \cdots PS(kl-l+1), \quad k \geq 1 \quad (21)$$

with l chosen such that $\mathbb{H}_{ik}$ as defined in (19) is strongly connected. To prove Theorem 2, one only needs to prove Proposition 2, proof of which will be given in Section 7:

Proposition 2. $|W_k| < 1$ for each $k \geq 1$.

120 5. Applications

In this section, we will apply the proposed algorithm in solving two problems as follows.

5.1. Least Square Solution

Linear equations arising from many engineering problems are usually overdetermined. Especially in distributed parameter estimation [17], observation equations are in the form of $A_i x = b_i + \eta_i, i = 1, 2, \dots, m$, where x is a parameter to be estimated and η_i is a term modeling the measurement noise to b_i . Because of the measurement noise, this type of linear equations usually do not have a solution for which the least square solution is desired. As mentioned in Section 1, a generalization of the algorithm proposed in [3] for solving such overdetermined linear equations does not scale well with the size of the network. Such limitation can be eliminated by using the algorithm proposed in this paper to solve the following problem:

Problem 2: Suppose $Ax = b$ has no solution and $A'A$ is non-singular, where $A = \text{col} \{A_1, A_2, \dots, A_m\}$ and $b = \text{col} \{b_1, b_2, \dots, b_m\}$. Each agent i knows A_i and b_i , $i \in \mathbf{m}$ and controls a state vector $x_i(t)$. Devise a local rule for each agent i to update its state vector $x_i(t)$ using only $x_j(t)$ from its nearby neighbors such that part of $x_i(t)$ converges to be the unique solution of $A'Ax = A'b$.

To solve Problem 2, we first pick up an m -vertex tree graph $\mathbb{T}$ and orient it. Let H denote the $m \times (m-1)$ incidence matrix of $\mathbb{T}$ with the i th row denoted by h_i . Instead of solving $A_i x = b_i$, agent i is assigned with the task of solving $[A_i' A_i \quad h_i \otimes I] x = A_i' b_i$. Then the m -agent network is able to solve $M\bar{x} = q$ with

$$M = \begin{bmatrix} A_1' A_1 & & \\ \vdots & H \otimes I & \\ A_m' A_m & & \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} A_1' b_1 \\ \vdots \\ A_m' b_m \end{bmatrix}.$$

We claim that a solution to $M\bar{x} = q$ must exist and the first elements of $\bar{x}$ denoted by x constitute the solution to $A'Ax = A'b$. To understand why this is so, first note that the block rows of $H \otimes I$ sum to zero because the rows of H sum to zero. Thus if E is a product of elementary row matrices which adds the first $(m-1)$ block rows of $H \otimes I$ to the last, then $E(H \otimes I)$ must be of the form

$$E(H \otimes I) = \begin{bmatrix} D \\ 0 \end{bmatrix}_{nm \times (m-1)n}$$

where D is a square matrix. Moreover D must be nonsingular because the rank $E(H \otimes I) = \text{rank } H \otimes I$ and $\text{rank } H \otimes I = (m-1)n$. This last rank

identity is a consequence of the fact that the rank of an incidence matrix of an m vertex connected graph, namely $\text{rank } H$, equals $m - 1$.

Next observe that the set of solutions to $M\bar{x} = q$ is the same as the set of solutions to $EM\bar{x} = Eq$. But

$$EM = \begin{bmatrix} A'_1 A_1 & & \\ & \ddots & \\ A'_{m-1} A_{m-1} & & D \\ & A'A & 0 \end{bmatrix} \quad \text{and} \quad Eq = \begin{bmatrix} A'_1 b_1 \\ & \ddots \\ A'_{m-1} b_{m-1} \\ A'b \end{bmatrix}.$$

Moreover EM is obviously nonsingular, so a solution to $EM\bar{x} = Eq$ and consequently $M\bar{x} = q$ exists. Note in addition, that since such a solution must also satisfy $EM\bar{x} = Eq$, x must satisfy $A'Ax = A'b$. Therefore x solves the least squares problem.

In order to solve Problem 2, it is sufficient to solve $M\bar{x} = q$, where $M \in \mathbb{R}^{mn \times mn}$. Since the incidence matrix H has many zero entries, there must be the same number of zero block matrices in M . Instead of letting each agent i control a state vector in $\mathbb{R}^{mn}$, we let each $x_i(t) \in \mathbb{R}^{r_i n}$, $i \in \mathbf{m}$, where r_i denote the number of non-zero blocks in the i th block row of M . Let $\bar{M}_i$ be resulted from deleting all the zero blocks from the i th block row of M . Suppose each agent i update its state vectors according to (12) with $\bar{M}_i x_i(1) = A'_i b_i$ and P_i the orthogonal projection matrix to the kernel of $\bar{M}_i$. If the neighbor graphs satisfy the constraints defined in Theorem 2 and Theorem 1, all $x_i(t)$ will reach a local consensus $\bar{x}$, where $M\bar{x} = q$. Then Problem 2 is solved since the first elements of $\bar{x}$ is a solution to $A'Ax = A'b$.

5.2. Network Localization

Since positional information is usually crucial for agents in a spatially distributed multi-agent network to identify the meaning of the data collected, the *network localization problem* has recently attracted much attention [18, 19, 20], the goal of which is to enable each agent to achieve its own position in a global coordinate system by only communicating with its nearby neighbors.

Problem 3: Given a fixed $(m + \bar{m})$ -agent network consisting of a number of m free agents with labels $\mathbf{m} = \{1, 2, \dots, m\}$ and a number of $\bar{m}$ anchor agents with $\bar{m} \geq 3$. Each anchor agent knows its own position in a global coordinate system while each free agent only knows an estimate of its position. Suppose each agent i is able to receive information from certain other agents denoted by $\bar{\mathcal{N}}_i$. We always assume $i \in \bar{\mathcal{N}}_i$. The neighbor relationship can be described by a $(m + \bar{m})$ -node graph $\mathbb{G}$ such that there is a directed arc from j to i in

175 $\mathbb{G}$ if and only if $j \in \bar{\mathcal{N}}_i$. Let $\mathbb{N}$ denote the m -agent network of all free agents in $\mathbb{G}$ and all arcs among the free agents in $\mathbb{G}$. Suppose $\mathbb{G}$ is such that there are at least three disjoint paths from anchor agents to each free agent in $\mathbb{G}$ and in addition $\mathbb{N}$ is connected and bidirectional. The localization problem of interest here is to enable each agent to update its own estimates by only using estimates from its nearby neighbors such that all agents in the network converge to achieve their positions.

To solve the above Problem 3, we let $p_i \in \mathbb{R}^2$ denote the position agent i in $\mathbb{G}$. With inter-agent distance measurements, one can have [21],

$$p_i = \sum_{j \in \bar{\mathcal{N}}_i / \{i\}} a_{ij} p_j \quad (22)$$

where the a_{ij} are called *barycentric coordinates* such that $\sum_{j \in \bar{\mathcal{N}}_i / \{i\}} a_{ij} = 1$. The authors of [21] have given explicit formulas to compute a_{ij} using the Cayley-Menger determinants under the assumption that each agent lies in the convex hull of its neighbors, which has recently been relaxed by [20]. Let $\mathcal{N}_i$ denote the set of neighbors of agent i in $\mathbb{N}$. Then $i \in \mathcal{N}_i$ because of the self-arcs. By (22) one has

$$\sum_{j \in \mathcal{N}_i} a_{ij} p_j = b_i, \quad i \in \mathbf{m} \quad (23)$$

where $a_{ii} = -1$, the a_{ij} , $j \neq i, j \in \mathcal{N}_i$ are the barycentric coordinates, and

$$b_i = - \sum_{j \in \bar{\mathcal{N}}_i / \mathcal{N}_i} a_{ij} p_j, \quad i \in \mathbf{m} \quad (24)$$

with $\bar{\mathcal{N}}_i / \mathcal{N}_i$ denoting the subset of $\bar{\mathcal{N}}_i$ whose elements are not in $\mathcal{N}_i$, that is, it is the set of agent i 's neighbors in $\mathbb{G}$ which are anchor agents. Let $p = \text{col} \{p_1, p_2, \dots, p_m\}$. Then p must be a solution of the following linear equation

$$Ap = b. \quad (25)$$

180 Here, A is an $m \times m$ block matrix with the ij th block given by $a_{ij} \otimes I_2$, where $a_{ii} = -1$, the a_{ij} , $j \neq i, j \in \mathcal{N}_i$, are the barycentric coordinates and $a_{ij} = 0$ if $j \notin \mathcal{N}_i$, and $b = \text{col} \{b_1, b_2, \dots, b_m\}$ with b_i given by (24). By employing the algorithm developed in [20], each agent i is able to know the i th block row of (A, b) . It has been also shown in [20] that if there are at least three
185 disjoint paths in $\mathbb{G}$ from anchor agents to each free agent, all the free agents are localizable, that is, the equation (25) has a unique solution.

Let each agent i in $\mathbb{N}$ controls a state vector

$$x_i(t) = \text{col} \{x_{ii_1}(t), x_{ii_2}(t), \dots, x_{ii_{d_i}}(t)\} \in \mathbb{R}^{2d_i}$$

where $\{i_1, i_2, \dots, i_{d_i}\} = \mathcal{N}_i$. Since $\mathbb{N}$ is connected and bidirectional, the graph $\mathbb{N}_i$, which is the subgraph of $\mathbb{N}$ corresponding to the vertices $\mathcal{N}_i$, is also connected and bidirectional. By Theorem 1, the algorithm developed in Section 3 drives all $x_i(t)$ to converge to a local consensus $p = \text{col} \{p_1, p_2, \dots, p_m\}$, which is a solution to (25). Then each $p_i \in \mathbb{R}^2$ is the position of free agent i in the global coordinate system, which is the limit vector that $x_{ii}(t)$ converges to and is achieved by agent i exponentially fast.

6. Conclusion

In this paper we have described a distributed algorithm to enable a network of m agents to solve linear equations in the form of $Ax = b$ with $A \in \mathbb{R}^{\bar{n} \times n}$. Compared with [3], a major advantage of the proposed algorithm is that each agent controls a state vector in size smaller than n . In future we will consider challenging communication issues arising from implementation [22, 23], and provide analytical proof for the effectiveness of the proposed algorithm in the case when $Ax = b$ has more than one solutions, which is true indicated by simulations.

7. Appendix

To prove Proposition 1 and Proposition 2, one needs the following lemmas.

Lemma 1. *Suppose $Ax = b$ has a unique solution. If there exists a vector $\bar{v}$ such that $Pv = v$ and $v = \text{col} \{R_1\bar{v}, R_2\bar{v}, \dots, R_m\bar{v}\}$, where P is as defined in (17) and all R_i as defined in Section 2, then $\bar{v} = 0$.*

Proof of Lemma 1: Since $Pv = v$, one has $(I - P)v = 0$. Then

$$(I - P_i)R_i\bar{v} = 0, \quad i \in \mathbf{m}$$

which and the definitions of P_i imply

$$\bar{A}_i R_i \bar{v} = 0, \quad i \in \mathbf{m}$$

Note in addition that $\bar{A}_i R_i = A_i$. Then

$$A_i \bar{v} = 0, \quad i \in \mathbf{m}.$$

It follows that $A\bar{v} = 0$. Since $Ax = b$ has a unique solution, one has $\bar{v} = 0$.
 210 Thus $v = 0$. We complete the proof. ■

Lemma 2. *Let $M_t = PS_tPS_{t-1}\cdots PS_1$, where $P^2 = P$ and $|S_t| \leq 1$. If $M_tM'_tv = v$, then $Pv = v$ and*

$$S_t \cdots S_2S_1S'_1S'_2 \cdots S'_tv = v. \quad (26)$$

Proof of Lemma 2: From $M_tM'_tv = v$ and $P^2 = P$, it is straightforward to verify that $Pv = v$. In the following we will prove (26) by induction.

For $t = 1$, one has $M_1 = PS_1$. Since $M_1M'_1v = v$, one has $PS_1S'_1Pv = v$ which and $Pv = v$ imply that $v'S_1S'_1v = v'v$. From $|S_1| \leq 1$ and the
 215 min-max theorem [24], one has $S_1S'_1v = v$. Then (26) is true for $t = 1$.

Now we suppose (26) is true for $t = k$, $k \geq 1$, and we check whether it is true for $t = k+1$. Since $M_{k+1}M'_{k+1}v = v$, one has $PS_{k+1}M_kM'_kS'_{k+1}Pv = v$, which and $Pv = v$ imply

$$v'S_{k+1}M_kM'_kS'_{k+1}v = v'v \quad (27)$$

Note in addition from $|S_{k+1}| \leq 1$ that

$$v'v \geq v'S_{k+1}S'_{k+1}v$$

from which and (27) one has

$$(S'_{k+1}v)'M_kM'_k(S'_{k+1}v) \geq (S'_{k+1}v)'(S'_{k+1}v) \quad (28)$$

Since $|M_k| \leq 1$ because of $|P| = 1$ and $|S_k| \leq 1$, one has

$$v'S_{k+1}M_kM'_kS'_{k+1}v = v'S_{k+1}S'_{k+1}v \quad (29)$$

By using the min-max theorem again, one has

$$M_kM'_k\bar{v} = \bar{v} \quad (30)$$

where $\bar{v} = S'_{k+1}v$. From (30) and the induction assumption on $t = k$, one has

$$\bar{v}'S_k \cdots S_2S_1S'_1S'_2 \cdots S'_k\bar{v} = \bar{v}'\bar{v}$$

That is,

$$v'S_{k+1}S_k \cdots S_2S_1S'_1S'_2 \cdots S'_kS'_{k+1}v = v'S_{k+1}S'_{k+1}v \quad (31)$$

From (27) and (29), one has

$$v' S_{k+1} S'_{k+1} v = v' v$$

Then (31) becomes

$$v' S_{k+1} S_k \cdots S_2 S_1 S'_1 S'_2 \cdots S'_k S'_{k+1} v = v' v.$$

From

$$|S_{k+1} S_k \cdots S_2 S_1 S'_1 S'_2 \cdots S'_k S'_{k+1}| \leq 1$$

and the min-max theorem again, one has

$$S_{k+1} S_k \cdots S_2 S_1 S'_1 S'_2 \cdots S'_k S'_{k+1} v = v.$$

Then (26) holds for $t = k + 1$. We complete the proof. ■

Lemma 3. *Let*

$$\bar{S} = S(kl) S(kl - 1) \cdots S(kl - l + 1), \quad (32)$$

where $S(t)$, $t = kl - l + 1, \dots, kl$, are those appearing in (21). If $\bar{S} \bar{S}' v = v$, there exists a vector $\bar{v}$ such that $v = \text{col} \{R_1 \bar{v}, R_2 \bar{v}, \dots, R_m \bar{v}\}$.

Proof of Lemma 3: By the definition of $S(t)$, one has there exists a permutation matrix Q such that

$$Q' S(t) Q = \text{diag} \{F_1(t) \otimes I_{n_1}, F_2(t) \otimes I_{n_2}, \dots, F_m(t) \otimes I_{n_m}\}$$

where $\gamma(F_i) = \mathbb{N}_i(t)$ and each F_i is a doubly stochastic matrix, $i \in \mathbf{m}$. Then

$$Q' \bar{S} Q = \text{diag} \{\bar{F}_1 \otimes I_{n_1}, \bar{F}_2 \otimes I_{n_2}, \dots, \bar{F}_m \otimes I_{n_m}\} \quad (33)$$

where $\bar{F}_i = F_i(kl) F_i(kl - 1) \cdots F_i(kl - l + 1)$, $i \in \mathbf{m}$, is doubly stochastic and $\gamma(\bar{F}_i) = \mathbb{H}_{i,k}$ is strongly connected because $\mathbb{N}(t)$ is assumed to be locally jointly strongly connected in Theorem 2. By (33), one has

$$Q' \bar{S}' Q = \text{diag} \{\bar{F}'_1 \otimes I_{n_1}, \bar{F}'_2 \otimes I_{n_2}, \dots, \bar{F}'_m \otimes I_{n_m}\}$$

It follows that

$$\begin{aligned} Q' \bar{S} \bar{S}' Q &= (Q' \bar{S} Q)(Q' \bar{S}' Q) \\ &= \text{diag} \{(\bar{F}_1 \bar{F}'_1) \otimes I_{n_1}, (\bar{F}_2 \bar{F}'_2) \otimes I_{n_2}, \dots, (\bar{F}_m \bar{F}'_m) \otimes I_{n_m}\} \end{aligned}$$

where $\bar{F}_i \bar{F}_i'$ is doubly stochastic and $\gamma(\bar{F}_i \bar{F}_i')$ is strongly connected since $\gamma(\bar{F}_i)$ is. Since $\bar{S} \bar{S}' v = v$, one has

$$(Q' \bar{S} \bar{S}' Q) Q' v = Q' v$$

Then there must exist vectors $v_1, v_2, \dots, v_m$ such that

$$Q' v = \text{col} \{ \mathbf{1}_{c_1} \otimes v_1, \mathbf{1}_{c_2} \otimes v_2, \dots, \mathbf{1}_{c_m} \otimes v_m \}$$

That is,

$$v = Q \text{col} \{ \mathbf{1}_{c_1} \otimes v_1, \mathbf{1}_{c_2} \otimes v_2, \dots, \mathbf{1}_{c_m} \otimes v_m \}. \quad (34)$$

Let $\bar{v} = \text{col} \{ v_1, v_2, \dots, v_m \}$. One has from (34) and definitions of Q and R_i that

$$v = \text{col} \{ R_1 \bar{v}, R_2 \bar{v}, \dots, R_m \bar{v} \}.$$

220 We complete the proof. ■

Proof of Proposition 1: Since the weights are chosen according to (18), S can be written as

$$S = D^{-1} M \quad (35)$$

where $D = \text{diag} \{ d_1 I_{n_1}, d_2 I_{n_2}, \dots, d_m I_{n_m} \}$ and M is symmetric because each $\mathbb{N}_i$ is bidirectional. From $P D^{-\frac{1}{2}} = D^{-\frac{1}{2}} P$, one has PS , $P\bar{S}$ and $P\bar{S}P$ have the same non-zero eigenvalues, where

$$\bar{S} = D^{-\frac{1}{2}} M D^{-\frac{1}{2}}.$$

Since M is symmetric, $P\bar{S}P$ is symmetric. Thus all eigenvalues of $P\bar{S}P$, PS , and $P\bar{S}$ are real.

Let λ be any non-zero eigenvalue of $P\bar{S}$ with the corresponding non-zero eigenvector v . Then

$$P\bar{S}v = \lambda v. \quad (36)$$

Multiplying P to both sides of (36), one has

$$P\bar{S}v = \lambda Pv$$

which and (36) imply

$$Pv = v. \quad (37)$$

Multiplying v' to both sides of (36), one has

$$v' \bar{S} v = \lambda v' v. \quad (38)$$

From the fact that $\bar{S}$ is symmetric and the min-max theorem [24], one has

$$\lambda_{\min}(\bar{S}) \leq \lambda \leq \lambda_{\max}(\bar{S}).$$

Note that $\bar{S}$ and S have the same non-zero eigenvalues, where $\bar{S}$ is symmetric and S is stochastic. Then all eigenvalues of S are real and

$$\lambda_{\min}(S) \leq \lambda \leq 1 \quad (39)$$

To prove the spectral radius of PS is strictly less than 1, it is sufficient to show

$$\lambda_{\min}(S) > -1 \quad (40)$$

and

$$\lambda \neq 1. \quad (41)$$

By the definition of S , there exists a permutation matrix Q such that

$$Q'SQ = \text{diag} \{E_1 \otimes I_{n_1}, E_2 \otimes I_{n_2}, \dots, E_m \otimes I_{n_m}\} \quad (42)$$

where each E_i is stochastic and $\gamma(E_i) = \mathbb{N}_i$ is strongly connected. Then (40) is true.

We prove (41) by contradiction. Suppose 1 is an eigenvalue of $P\bar{S}$ with the corresponding eigenvector v . In view of (37), (38) and the min-max theorem, one has

$$\bar{S}v = v, \quad Pv = v.$$

Note that $\bar{S} = D^{\frac{1}{2}}SD^{-\frac{1}{2}}$. Then

$$SD^{-\frac{1}{2}}v = D^{-\frac{1}{2}}v. \quad (43)$$

It follows that

$$(Q'SQ)(Q'D^{-\frac{1}{2}}v) = Q'D^{-\frac{1}{2}}v.$$

That is, $Q'D^{-\frac{1}{2}}v$ is an eigenvector of $Q'SQ$ corresponding to the eigenvalue 1, which and (42) imply that there exist vectors $v_1, v_2, \dots, v_m$ such that

$$Q'D^{-\frac{1}{2}}v = \text{col} \{\mathbf{1}_{c_1} \otimes v_1, \mathbf{1}_{c_2} \otimes v_2, \dots, \mathbf{1}_{c_m} \otimes v_m\}$$

225 Let $\hat{v} = \text{col} \{R_1\bar{v}, R_2\bar{v}, \dots, R_m\bar{v}\}$ where $\bar{v} = \text{col} \{v_1, v_2, \dots, v_m\}$. One has

$$\begin{aligned} v &= D^{\frac{1}{2}}Q \text{col} \{\mathbf{1}_{c_1} \otimes v_1, \mathbf{1}_{c_2} \otimes v_2, \dots, \mathbf{1}_{c_m} \otimes v_m\} \\ &= D^{\frac{1}{2}}\hat{v}. \end{aligned}$$

from which, $Pv = v$ and $PD^{\frac{1}{2}} = D^{\frac{1}{2}}P$, one has

$$P\hat{v} = \hat{v}$$

By Lemma 1, one has $\bar{v} = 0$. Then $\hat{v} = 0$ and $v = 0$. Thus (41) is true. We complete the proof. ■

Proof of Proposition 2: Since P is an orthogonal projection and each $S(t)$, $t \geq 1$, is doubly stochastic, one has

$$|W_k| \leq 1. \quad (44)$$

To prove Proposition 2, one only needs show that 1 is not an eigenvalue of $W_k W_k'$. We will prove this by contradiction. Suppose there exists a $v \neq 0$ such that

$$W_k W_k' v = v$$

where W_k is as defined in (21). By Lemma 2, one has

$$Pv = v$$

and $v' \bar{S} \bar{S}' v = v' v$, where $\bar{S}$ is as defined in (32). Note that $\bar{S} \bar{S}'$ is symmetric and $|\bar{S}| \leq 1$. One has

$$\bar{S} \bar{S}' v = v \quad (45)$$

By Lemma 3, there exists a vector $\bar{v}$ such that $v = \text{col} \{R_1 \bar{v}, R_2 \bar{v}, \dots, R_m \bar{v}\}$, from which, $Pv = v$ and Lemma 1, one has $v = 0$. Thus 1 can not be an
230 eigenvalue of $W_k W_k'$. Therefore $|W_k| < 1$. We complete the proof. ■

References

- [1] S. Mou, A. S. Morse, A fixed-neighbor, distributed algorithm for solving a linear algebraic equation, In Proceeding of European Control Conference (2013) 2269–2273.
- 235 [2] S. Mou, J. Liu, A. S. Morse, A distributed algorithm for solving a linear algebraic equation, In Proceeding of the 51st Annual Allerton Conference on Communication, Control, and Computing (2013) 267–274.
- 240 [3] S. Mou, J. Liu, A. S. Morse, A distributed algorithm for solving a linear algebraic equation, IEEE Transactions on Automatic Control 60 (11) (2015) 2863–2878.

- [4] B. D. O. Anderson, S. Mou, A. S. Morse, U. Helmke, Decentralized gradient algorithm for solution of a linear equation, Numerical Algebra, Control and Optimization Submitted.
- 245 [5] A. Nedic, A. Ozdaglar, Distributed sub-gradient methods for multi-agent optimization, IEEE Transactions on Automatic Control (2009) 48–61.
- [6] D. Jakovetic, J. M. F. Moura, J. Xavier, Fast distributed gradient methods, IEEE Transactions on Automatic Control (5) (2014) 1131–1146.
- 250 [7] J. C. Duchi, A. Agarwal, M. J. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling, IEEE Transactions on Automatic Control (3) (2012) 592–606.
- [8] T. Chang, A. Nedic, A. Scaglione, Distributed constrained optimization by consensus-based primal-dual perturbation method, IEEE Transactions on Automatic Control (6) (2014) 1524–1538.
- 255 [9] B. Gharesifard, J. Cortes, Continuous-time distributed convex optimization on directed graphs, IEEE Transactions on Automatic Control To appear.
- [10] A. Nedic, A. Ozdaglar, P. A. Parrilo, Constrained consensus and optimization in multi-agent networks, IEEE Transactions on Automatic Control (2010) 922–938.
- 260 [11] A. Jadbabaie, J. Lin, A. S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Transactions on Automatic Control 46 (6) (2003) 988–1001.
- 265 [12] W. Ren, R. W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Transactions on Automatic Control (5) (2005) 655–661.
- [13] W. Ren, On consensus algorithms for double-integrator dynamics, IEEE Transactions on Automatic Control 53 (2008) 1503–1509.
- 270 [14] M. Cao, A. S. Morse, B. D. O. Anderson, Agree asynchronously, IEEE Transactions on Automatic Control 53 (8) (2008) 1826–1838.
- [15] M. Cao, A. S. Morse, B. D. O. Anderson, Reaching a consensus in a dynamically changing environment: a graphical approach, SIAM Journal on Control and Optimization 47 (2008) 575–600.

- 275 [16] A. D. Dominguez-Garcia, C. N. Hadjicostis, Distributed matrix scaling
and application to average consensus in directed graphs, *IEEE Trans-*
actions on Automatic Control (3) (2014) 667–681.
- [17] S. Kar, J. M. F. Moura, K. Ramanan, Distributed parameter estima-
tion in sensor networks: Nonlinear observation models and imperfect
280 communication, *IEEE Transactions on Information Theory* (2012) 1 –
52.
- [18] P. Barooah, J. P. Hespanha, Estimation from relative measurements:
Electrical analogy and large graphs, *IEEE Transactions on Signal Pro-*
cessing 56 (6) (2008) 2181–2193.
- 285 [19] G. Zhu, J. Hu, A distributed continuous-time algorithm for network lo-
calization using angle-of-arrival information, *Automatica* 50 (1) (2014)
53–63.
- [20] Y. Diao, Z. Lin, M. Fu, A barycentric coordinate based distributed
localization algorithm for sensor networks, *IEEE Transactions on Signal*
290 *Processing* 62 (18) (2014) 4760–4771.
- [21] U. A. Khan, S. Kar, J. M. F. Moura, Distributed sensor localization
in random environments using minimal number of anchor nodes, *IEEE*
Transactions on Signal Processing 57 (5) (2009) 2000–2016.
- 295 [22] C. Li, J. Jose, X. Wu, Distributed-fountain network code (DFNC) for
content delivery in vehicular networks, *Proceeding of the 10th ACM In-*
ternational Workshop on Vehicular inter-networking, systems, and ap-
plications (2013) 31–40.
- [23] J. Liu, S. Mou, A. S. Morse, An asynchronous distributed algorithm
for solving a linear algebraic equation, *Proc. of the 52nd Conference on*
300 *Decision and Control* (2013) 5409–5414.
- [24] R. C. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge University
Press, New York, 1985.