23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

Distributed Continuous-time Algorithm to Solve A Linear Matrix Equation

Wen Deng, Xianlin Zeng, and Yiguang Hong

I. Introduction

Recently, with the increasing scale and big data of various practical problems in natural science and engineering fields, distributed algorithms over multi-agent networks have attracted a significant amount of research attention (see [1]–[4]). Both discrete-time and continuous-time algorithms (see [1]–[4]) have been proposed and investigated for distributed optimization with various types of constraints. Nowadays, continuous-time algorithms have received much attention in [2]–[6], partially because some continuous-time approaches may provide an effective tool for the optimization analysis and design, though distributed designs for many important problems are still challenging.

The solution to linear algebraic equations of the form $Ax = b$, where A is a matrix and x, b are vectors of appropriate dimensions, in a multi-agent network has attracted much interest and has been extensively studied in [5]–[10]. The significant results in [5]–[9] provided various distributed algorithms with the standard case that each agent knows a few rows of A and b, while [10] proposed a distributed computation approach for another standard case, where each agent has the knowledge of a few columns of matrix A.

As we know, the algorithms to solve matrix equations such as Sylvester equations and Lyapunov equations, with linear algebraic equations as their special cases, are important and rapidly developed (referring to [11], [12] and the references therein for details). Although the distributed computation of $Ax = b$ has been studied in the past several years, the results for distributed computation of general linear matrix equations are quite few, though [13] studied the solution to the matrix equation of the form $AXB = F$. One commonly used form is $\sum_{i=1}^{r} A_i X B_i = \sum_{i=1}^{r} C_i$, and the computation of its solution X plays a fundamental role in many important application problems such as the computation of the generalized Sylvester equations (see [11], [12], [14]). In fact, when the r in the general linear matrix equation is more than two and even larger, the computational complexity increases much if we still deal with the problem with some traditional ideas. Meanwhile, the algorithms in [13] solving the equation $AXB = F$ can not applied directly to the structure which is mentioned in this paper.

The main purpose of this paper is to design a distributed continuous-time algorithm for solving the linear matrix equation $\sum_{i=1}^{r} A_i X B_i = \sum_{i=1}^{r} C_i$, in a distributed way over a multi-agent network. Considering that the computation of a least squares solution can be related to an optimization problem, we also take a distributed optimization perspective to investigate the solution for this linear matrix equation over a multi-agent network. Then we propose a distributed continuous-time algorithm and analyze its convergence with help of some control techniques such as Lyapunov approaches and semi-stability [15].

The contributions of this paper are summarized as follows.

- For a distributed design to solve the linear matrix equation of the form $\sum_{i=1}^{r} A_i X B_i = \sum_{i=1}^{r} C_i$, we propose a distributed computation where each agent i just needs to know A_i, B_i, C_i, communicates with its neighbors, and finally reaches the estimates consensus.
- By using a distributed constrained optimization reformulation, we propose a distributed continuous-time algorithm for the linear matrix equation, specifically, by using augmented Lagrangian functions. The proposed algorithm is able to find a least squares solution to the linear matrix equation for any initial condition under mild assumptions.
- For the distributed algorithm proposed, we provide a rigorous proof for the correctness and convergence of the algorithm to a least squares solution based on saddle-point dynamics of Lagrangian functions and semi-stability theory with mild conditions.

II. Preliminaries and Formulation

We first introduce some notations. Let $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{m\times n}$ denote the set of numbers, the set of $n-$dimensional real column vectors, and the set of $m \times n$ real matrices, respectively. Write $\mathbf{1}_n$ for a vector in $\mathbb{R}^n$ with all its components equal to 1, I_n for the $n \times n$ identity matrix, and $\mathbf{0}_n(\mathbf{0}_{m\times n})$ for the $n-$dimensional column vector ($m \times n$ matrix) with all elements of 0. For $M \in \mathbb{R}^{m\times n}$, we denote M^T, image(M) and $\ker(M)$ as the transpose, the image, and the kernel of the matrix M, respectively. Let $\otimes$ denote the Kronecker product; let col $\{M_1, \dots, M_n\}$ denote a column stack of matrices $M_i, i = 1, \dots, n$, which is $[M_1^T, \dots, M_n^T]^T$. Additionally, denote $||\cdot||$ as the Euclidean norm, and $||\cdot||_F$ as the Frobenius norm of matrices, $\langle \cdot, \cdot \rangle_F$ as the Frobenius inner product of real matrices.

Then we introduce some concept of graph theory. An undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{1, \dots, r\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, $A = [a_{i,j}] \in \mathbb{R}^{n\times n}$ is the adjacency matrix such that $a_{i,j} = a_{j,i} > 0$ if $(i, j) \in \mathcal{E}$ and $a_{i,j} = 0$ otherwise. The Laplacian matrix $L_n = D - A$, where $D \in \mathbb{R}^{n\times n}$ is diagonal with $D_{i,i} = \sum_{i=1}^{n} a_{i,j}$. It is known that, if the undirected graph $\mathcal{G}$ is connected, then $L_n = L_n^T \geq 0$, rank$(L_n) = n - 1$ and $\ker(L_n) = \{k\mathbf{1}_n : k \in \mathbb{R}\}$ [16].

Then we use the definition of semi-stability given in [15]. Consider a dynamical system

$$\dot{x}(t) = \phi(x(t)), \quad x(0) = x_0, \quad t \geq 0, \tag{1}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz continuous. Given a trajectory of (1) $x(t) : [0, \infty) \to \mathbb{R}^d$. The point $z \in \mathbb{R}^d$ is a limit point of a solution $t \to x(t)$ if there exists a positive increasing divergent sequence $\{t_i\}_{i=1}^{\infty} \subset \mathbb{R}$ such that $z = \lim_{i \to \infty} x(t_i)$. We denote the set of all positive limit points by $\Omega(x)$. A set $\mathcal{M}$ is positively invariant with respect to (1) if, for every $x_0 \in \mathcal{M}, \mathcal{M}$ contains the solution $x(t)$ of (1) for all $t \geq 0$ with initial condition x_0.

The following result is a special case of [15, Theorem 3.1].

Lemma 2.1: Let $\mathcal{D}$ be an open positively invariant set with respect to (1) and let $V : \mathcal{D} \to \mathbb{R}$ be a continuously differentiable function, and $x(\cdot)$ be a solution of (1) with $x(0) \in \mathcal{D}$. Assume $\dot{V}(x(t)) \leq 0, \forall x \in \mathcal{D}$ and define $\mathcal{Z} = \{x \in \mathcal{D} : \dot{V}(x) = 0\}$. If every point in the largest invariant subset $\mathcal{M}$ of $\bar{\mathcal{Z}} \cap \mathcal{D}$ is a Lyapunov stable equilibrium point, where $\bar{\mathcal{Z}}$ is the closure of $\mathcal{Z}$, then (1) is semi-stable with respect to $\mathcal{D}$.

It is time to formulate our problem. Consider the linear matrix equation of the form

$$\sum_{i=1}^{r} A_i X B_i = \sum_{i=1}^{r} C_i, \tag{2}$$

where $A_i \in \mathbb{R}^{n \times p}, B_i \in \mathbb{R}^{q \times m}, C_i \in \mathbb{R}^{n \times m}$ for all $i \in \{1, \ldots, r\}$ are known matrices, and $X \in \mathbb{R}^{p \times q}$ is an unknown matrix to be solved. Actually, the equation (2) may not have an exact solution X. When the solution of equation does not exist, we usually consider the minimum norm least squares solution. Meanwhile, the least squares method can work out the exact solution when it exists.

The least squares solution to (2) is defined as the solution to the following optimization problem

$$\min_{X} || \sum_{i=1}^{r} (A_i X B_i - C_i) ||_F^2. \tag{3}$$

Consider the distributed computation of a least squares solution to (2) over a multi-agent network described by an undirected graph $\mathcal{G}$, where matrices A_i, B_i, C_i are known by the i-th agent, for all $i \in \{1, \ldots, r\}$.

In a distributed way, we rewrite equation (3) as

$$\begin{aligned} \min_{\bar{X}} \quad & || \sum_{i=1}^{r} (A_i X_i B_i - C_i) ||_F^2, \\ s.t. \quad & X_i = X_j, \qquad \forall i, j \in \{1, \ldots, r\} \end{aligned} \tag{4}$$

where $\bar{X} = \mathrm{col}\{X_1, \ldots, X_r\} \in \mathbb{R}^{rp \times q}$, X_i is the estimate of the solution to (3) for the i-th agent, and all the agents' estimates will reach consensus finally. In this sense, we consider the following optimization problen:

$$\begin{aligned} \min_{(\bar{X}, \bar{Y})} \quad & \frac{1}{2} \sum_{i=1}^{r} ||Y_i||_F^2 + \frac{1}{2} \sum_{i=1}^{r} \langle \sum_{j=1}^{r} a_{i,j} (Y_i - Y_j), Y_i \rangle_F, \\ s.t. \quad & \sum_{i=1}^{r} (A_i X_i B_i - C_i) = \sum_{i=1}^{r} Y_i, \\ & X_i = X_j, \qquad \forall i, j \in \{1, \ldots, r\} \end{aligned} \tag{5}$$

where $\bar{X} = \mathrm{col}\{X_1, \ldots, X_r\}$ and $\bar{Y} = \mathrm{col}\{Y_1, \ldots, Y_r\}$.

In fact, the equality constraint

$$\sum_{i=1}^{r} (A_i X_i B_i - C_i) = \sum_{i=1}^{r} Y_i \tag{6}$$

is equivalent to

$$A_i X_i B_i - C_i - Y_i + \sum_{j=1}^{r} a_{i,j} (Z_i - Z_j) = 0_{n \times m}, \tag{7}$$

where $a_{i,j}$ is the (i, j)-th element of adjacency matrix of $\mathcal{G}$, which is undirected and connected. And because $\mathcal{G}$ is undirected and connected,

$$X_i = X_j, \qquad \forall i, j \in \{1, \ldots, r\}$$

is equivalent to

$$\sum_{j=1}^{r} a_{i,j} (X_i - X_j) = 0_{p \times q}, \qquad \forall i, j \in \{1, \ldots, r\}.$$

Then we get the problem

$$\begin{aligned} \min_{(\bar{X}, \bar{Y}, \bar{Z})} \quad & \frac{1}{2} \sum_{i=1}^{r} ||Y_i||_F^2 + \frac{1}{2} \sum_{i=1}^{r} \langle \sum_{j=1}^{r} a_{i,j} (Y_i - Y_j), Y_i \rangle_F, \\ s.t. \quad & A_i X_i B_i - C_i - Y_i + \sum_{j=1}^{r} a_{i,j} (Z_i - Z_j) = 0_{n \times m}, \\ & \sum_{j=1}^{r} a_{i,j} (X_i - X_j) = 0_{p \times q}, \quad \forall i \in \{1, \ldots, r\}. \end{aligned} \tag{8}$$

where $\bar{X} = \mathrm{col}\{X_1, \ldots, X_r\}, \bar{Y} = \mathrm{col}\{Y_1, \ldots, Y_r\}$ and $\bar{Z} = \mathrm{col}\{Z_1, \ldots, Z_r\}$.

III. Main Result

By the KKT optimality condition, $(\bar{X}^*, \bar{Y}^*, \bar{Z}^*)$ is a solution to problem (8) if and only if

$$\begin{aligned} & A_i X_i^* B_i - C_i - Y_i^* + \sum_{j=1}^{r} a_{i,j} (Z_i^* - Z_j^*) = 0_{n \times m}, \\ & \sum_{j=1}^{r} a_{i,j} (X_i^* - X_j^*) = 0_{p \times q}, \quad \forall i, j \in \{1, \ldots, r\} \end{aligned} \tag{9}$$

and there are matrices $\Lambda_i^{1*} \in \mathbb{R}^{p\times q}$, $\Lambda_i^{2*} \in \mathbb{R}^{n\times m}$, such that

$$\begin{aligned}
0_{p\times q} &= -A_i^T \Lambda_i^{2*} B_i^T - \sum_{j=1}^{r} a_{i,j}(\Lambda_i^{1*} - \Lambda_j^{1*}),\\
0_{n\times m} &= -Y_i^* - \sum_{j=1}^{r} a_{i,j}(Y_i^* - Y_j^*) + \Lambda_i^{2*},\\
0_{n\times m} &= -\sum_{j=1}^{r} a_{i,j}(\Lambda_i^{2*} - \Lambda_j^{2*}).
\end{aligned} \tag{10}$$

Here we focus on problem (8), and propose a distributed algorithm of agent i as

$$\begin{aligned}
\dot{X}_i(t) =& - A_i^T \Lambda_i^2(t) B_i^T - \sum_{j=1}^{r} a_{i,j}(\Lambda_i^1(t) - \Lambda_j^1(t))\\
& - \sum_{j=1}^{r} a_{i,j}(X_i(t) - X_j(t)), X_i(0) = X_{i0} \in \mathbb{R}^{p\times q},\\
\dot{Y}_i(t) =& - Y_i(t) + \Lambda_i^2(t) - \sum_{j=1}^{r} a_{i,j}(Y_i(t) - Y_j(t)),\\
& Y_i(0) = Y_{i0} \in \mathbb{R}^{n\times m},\\
\dot{Z}_i(t) =& - \sum_{j=1}^{r} a_{i,j}(\Lambda_i^2(t) - \Lambda_j^2(t)), Z_i(0) = Z_{i0} \in \mathbb{R}^{n\times m},\\
\dot{\Lambda}_i^1(t) =& \sum_{j=1}^{r} a_{i,j}(X_i(t) - X_j(t)), \Lambda_i^1(0) = \Lambda_{i0}^1 \in \mathbb{R}^{p\times q},\\
\dot{\Lambda}_i^2(t) =& A_i X_i(t) B_i - C_i - Y_i(t) + \sum_{j=1}^{r} a_{i,j}(Z_i(t)\\
& - Z_j(t)) - \sum_{j=1}^{r} a_{i,j}(\Lambda_i^2(t) - \Lambda_j^2(t)),\\
& \Lambda_i^2(0) = \Lambda_{i0}^2 \in \mathbb{R}^{n\times m}.
\end{aligned} \tag{11}$$

$X_i(t), Y_i(t)$ and $Z_i(t)$ are the estimates of solutions to problem (8) by agent i at time t, and $\Lambda_i^1(t)$ and $\Lambda_i^2(t)$ are the estimates of Lagrangian multipliers for the constraints in (8) by agent i at time t.

Remark 3.1: Algorithm (11) is a primal-dual algorithm, whose primal variables are X_i, Y_i and Z_i, and dual variables are Λ_i^1 and Λ_i^2.

Remark 3.2: Algorithm (11) can be viewed as the saddle-point dynamics of the augmented Lagrangian function L. That is

$$\begin{aligned}
&\dot{X}_i = -\nabla_{X_i} L,\ \dot{Y}_i = -\nabla_{Y_i} L,\ \dot{Z}_i = -\nabla_{Z_i} L,\\
&\dot{\Lambda}_i^1 = \nabla_{\Lambda_i^1} L,\ \dot{\Lambda}_i^2 = \nabla_{\Lambda_i^2} L, \quad \forall i \in \{1, \ldots, r\}.
\end{aligned}$$

Lemma 3.1: Suppose that $\mathcal{G}$ is connected and undirected. $(\bar{X}^*, \bar{Y}^*, \bar{Z}^*)$ is a solution of problem (8) if and only if there exist $\Lambda^{k*} \in \mathbb{R}^{rn\times m}, k = 1, 2$ such that $(\bar{X}^*, \bar{Y}^*, \bar{Z}^*, \Lambda^{1^*}, \Lambda^{2^*})$ is an equilibrium of algorithm (11).

The proof can be obtained following the KKT optimality condition, which is omitted here.

The following shows the convergence of algorithm (11).

Theorem 3.1: If the undirected graph $\mathcal{G}$ is connected, then

1) every equilibrium of algorithm (11) is Lyapunov stable and its trajectory is bounded for any initial condition;
2) the trajectory $(\bar{X}(t), \bar{Y}(t), \bar{Z}(t), \Lambda^1(t), \Lambda^2(t))$ converges to an equilibrium of (11);
3) $\lim_{t\to\infty} X_i(t)$, for all $i \in \{1, \ldots, r\}$ is a least squares solution to problem (2). In addition, if $\lim_{t\to\infty} \sum_i ||Y_i||_F^2 = 0$, $\lim_{t\to\infty} X_i(t)$ is an exact solution to problem (2).

In this extended abstract, the proof is omitted here.

References

[1] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922-938, 2010.
[2] P. Yi, Y. Hong, and F. Liu, "Distributed gradient algorithm for constrained optimization with application to load sharing in power systems," *Systems & Control Letters*, vol. 83, pp. 45-52, 2015.
[3] Z. Qiu, S. Liu, and L. Xie, "Distributed constrained optimal consensus of multi-agent systems," *Automatica*, vol. 68, pp. 209-215, 2016.
[4] X. Zeng, P. Yi, and Y. Hong, "Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5227-5233, 2017.
[5] G. Shi and B. D. O. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659-2674, 2017.
[6] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2863-2878, 2015.
[7] B. Anderson, S. Mou, A. Morse, and U. Helmke, "Decentralized gradient algorithm for solution of a linear equation," arXiv:1509.04538v1.
[8] L. Wang, D. Fullmer, and A. S. Morse, "A distributed algorithm with an arbitrary initialization for solving a linear algebraic equation," in *Americian Control Conference*, Boston, MA, USA, 2016, pp. 1078-1081.
[9] J. Liu, S. Mou, and A. S. Morse, "Asynchronous distributed algorithms for solving linear algebraic equations," *IEEE Transactions on Automatic Control*, DOI: 10.1109/TAC.2017.2714645.
[10] K. Cao, X. Zeng, and Y. Hong, "Continuous-time distributed algorithms for solving linear algebraic equation," in *Proc. of The 36th Chinese Control Conference*, Dalian, China, 2017.
[11] V. Simoncini, "Computational methods for linear matrix equations?' *SIAM Review*, vol. 58, no. 3, pp. 377-441, 2016.
[12] P. Lancaster, "Explicit solutions of linear matrix equations," *SIAM Review*, vol. 12, no. 4, pp. 544-566, 1970.
[13] X. Zeng, S. Liang, Y. Hong, and J. Chen, "Distributed computation of linear matrix equations: An optimization perspective," arXiv:1708.01833.
[14] J. K. Baksalary and R. Kala, "The matrix equation $AXB + CXD = E$," *Linear Algebra and its Applications*, vol. 30, pp. 141-147, 1980.
[15] Q. Hui, W. Haddad, and S. Bhat, "Semistability, finite-time stability, differential inclusions, and discontinuous dynamical systems having a continuum of equilibria," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2465-2470, 2009.
[16] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, New York: Springer-Verlag, 2001.
[17] J. LaSalle, "Some extensions of Liapunov's second method," *IRE Transactions on Circuit Theory*, vol. 7, no. 4, pp. 520-527, 1960.