Paper Review: A Distributed Algorithm for Solving a Linear Algebraic Equation

Tomoki Koike

***Abstract*—We will analyze and dissect a paper by Shaoshuai Mou, Ji Liu, and A. Stephen Morse [1]. This paper focuses on the development of a distributed algorithm based on the graph theory and consensus theories that solves a real linear algebraic equation in the shape of $Ax = b$. This equation is ubiquitous throughout any science application, and therefore, is a field that has been well investigated but is constantly improved upon with newly proposed methods. This paper casts its original theory of solving this linear equation by partitioning rows as a subset and quantifying each as an agent. Then each agents will be updating its state recursively by the state estimation by receiving information from neighboring agents.**

***Index Terms*—Autonomous Agents; Distributed Algorithms; Multi-agent Systems; Graph Theory**

I. Introduction and Motivation

In relevant studies, to solve large systems of linear equations there were techniques like parallel processing proposed. Like such approach, this paper aims to provide a faster and more accurate computation of the linear equation. The method of distributed algorithms, which is implemented in this paper, is effective in the cases of sensor networking and filtering applications where information is processed amongst a system with some connectivity.

II. Problem Formulation

When there are $m > 1$ autonomous agents and neighbors denoted by i, such network can be described as $\mathcal{N}_i(t)$ at some time t. There are two types of graphs: directed and undirected graph. In this paper the directed graph is denoted as $\mathbb{N}(t)$. In this graph there are m vertices and arcs or edges that connect certain pairs of nodes. Now with the network the objective is the multi-agents to solve the linear system equation

$$Ax = b \tag{1}$$

Purdue University AAE

where

$$A = \begin{bmatrix} A_1' & A_2' & A_3' & \cdots & A_m' \end{bmatrix}'_{n \times \bar{n}}$$
$$b = \begin{bmatrix} b_1' & b_2' & b_3' & \cdots & b_m' \end{bmatrix}'_{1 \times \bar{n}}$$
$$\bar{n} = \sum_{i=1}^{m} n_i.$$

The problem to be solved is a distributed parameter estimation problem where the b_i is the observation made by each agent and x_i is the estimation that we require. In the paper there are a several proposed methods which aim to solve the equation based on a set of conditions.

1) for any pair of real matrices (A, b)
2) solutions are at least exponentially fast
3) largest possible class of time-varying directed graphs with guaranteed convergence
4) solution is obtained even without the round off and communication errors
5) can solve for any graph sequence which is repeatedly jointed or strongly connected
6) solved with a n dimensional state vector received at each time
7) can solve without unrealistic assumptions
8) can operate asynchronously

This paper reformulates the distribution optimization problem and applies preexisting algorithms to it. Moreover they reformulate the problem into a least square problem. The last approach taken is the view the problem as a consensus problem and have the states converge to some goal that is constrained convexly. The last method employs distributed averaging to devise an algorithm.

III. Main Results

By introducing a orthogonal projection matrix to the null space of A_i denoted as P_i, this paper reformulates the problem as follows.

$$x_i(t+1) = x_i(t) - P_i\Big(x_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_i} x_j(t)\Big) \tag{2}$$

where $d_i(t)$ is the number of neighbors of node i at some time t. Through analysis, this paper concludes that for a unique solution case there is a geometric convergence rate for the worst case that characterize a specific expression which is derived in this paper. Furthermore, with small changes to the algorithm, it has been proven that the algorithm can track the solution to the linear equation problem with sufficiently small rates. Given that there are no lag in the communication amongst the agents, it has also been proven that the algorithm can operate asynchronously.

IV. Your Ideas of Further Improvements

As the paper mentions on the concluding remarks, there are a few questions open to who have read this paper for further research. However, this paper did provide me with a new technique of implementing consensus theory to solve a large scale and network based linear equation that has the potential of being applied to many applications. In particular, for the current research I am doing for this class, I am first formulating a convex weight that forms the adjacency matrix A for a PRM (Probabilistic Road Map) planning algorithm that optimizes the point generations. Then supposing that each generated points for each iteration forms a graph $\mathcal{N}_i$, I will solve some kind of linear equation to have the nodes update their location to optimize the path. The algorithm can be implemented to solve the linear equation that I will have to deal in my own research; however, I do believe that I will have to probe the robustness of this algorithm so that I can reject a sufficient amount of noise and irregularities that may occur.

References

[1] Shaoshuai Mou, Ji Liu, and A. Stephen Morse. A distributed algorithm for solving a linear algebraic equation. *IEEE Transactions on Automatic Control*, 60(11):2863–2878, 2015.