CERIAS

The Center for Education and Research in Information Assurance and Security

Distributed Algorithms for Solving Linear Equations

Xuan Wang, Jingqiu Zhou

Advisor: Shaoshuai Mou (mous@purdue.edu)

AutoMous Lab

Introduction

The goal of our research is to propose distributed algorithms for solving linear equations by ***multi-agent networks***. In the network, each agent only knows part of these linear equations and is able to communicate with its nearby neighbors. The key idea behind the algorithms is a so-called ``***agreement principle***'', in which each agent limits the update of its state to satisfy its own equation meanwhile trying to reach a consensus with its nearby neighbors' states.

Motivation

- Requirement for data security: Each agent knows part of equation and the sub-equation is not allowed to be shared through the network.
- Requirement on calculation Efficiency when the size of equation are extremely large.
- Requirement for real time data processing when the data acquisition units are physically distant from each other.

The Algorithms

- When the equation has **at least one solution**, to get it:

Initial $x_i(t)$ as an arbitrary solution of $A_i x = b_i$, then follow the update:

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

Where P_i is the orthogonal projection matrix to the kernel of A_i, that is $P_i = I - A_i'(A_i A_i')^{-1}A_i$.

- Furthermore, when equation has **multiple solutions**, to get a solution with **minimum l_2 norm**.

By a special initialization:

$$x_i(0) = (M_i' M_i)^{-1} M_i' \begin{bmatrix} b_i \\ 0 \end{bmatrix}$$

where

$$M_i = \begin{bmatrix} A_i \\ P_i \end{bmatrix}$$

- When the equation **does not have solution**, to get its **least square solution**, by any initial point and the update:

$$\begin{bmatrix} x_i(t+1) \\ z_i(t+1) \end{bmatrix} = Q_i \begin{bmatrix} x_i(t) + \frac{1}{d_i}\sum_{j\in\mathcal{N}_j} z_j(t) + \frac{1}{d_i}A_i' b_i \\ z_i(t) - \frac{1}{d_i}\sum_{j\in\mathcal{N}_j} x_j(t) \end{bmatrix}$$

Where $z_i(t) \in \mathbb{R}^n$ is an extra state hold by each agent and

$$Q_i = \begin{bmatrix} I_n + \frac{1}{d_i}A_i' A_i & I_n \\ -I_n & I_n \end{bmatrix}^{-1}$$

The Problem

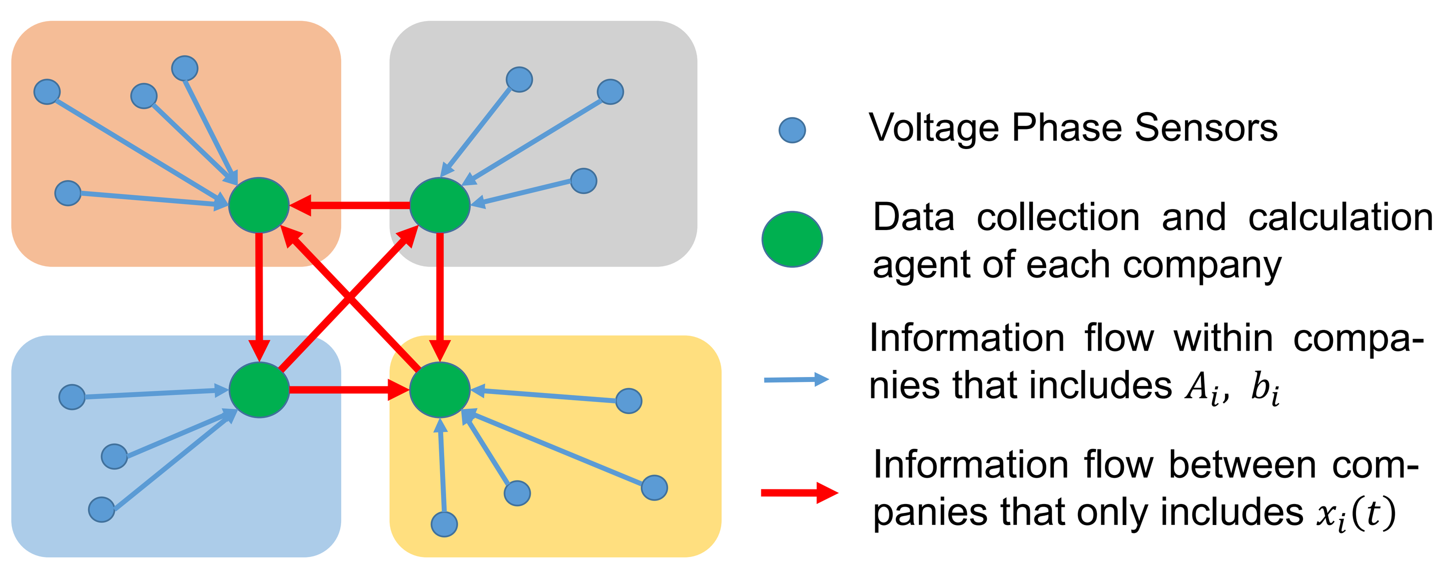
To solve a linear equation $Ax = b$ by a network of m agents, each agent i knows a sub-equation $A_i x = b_i$

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

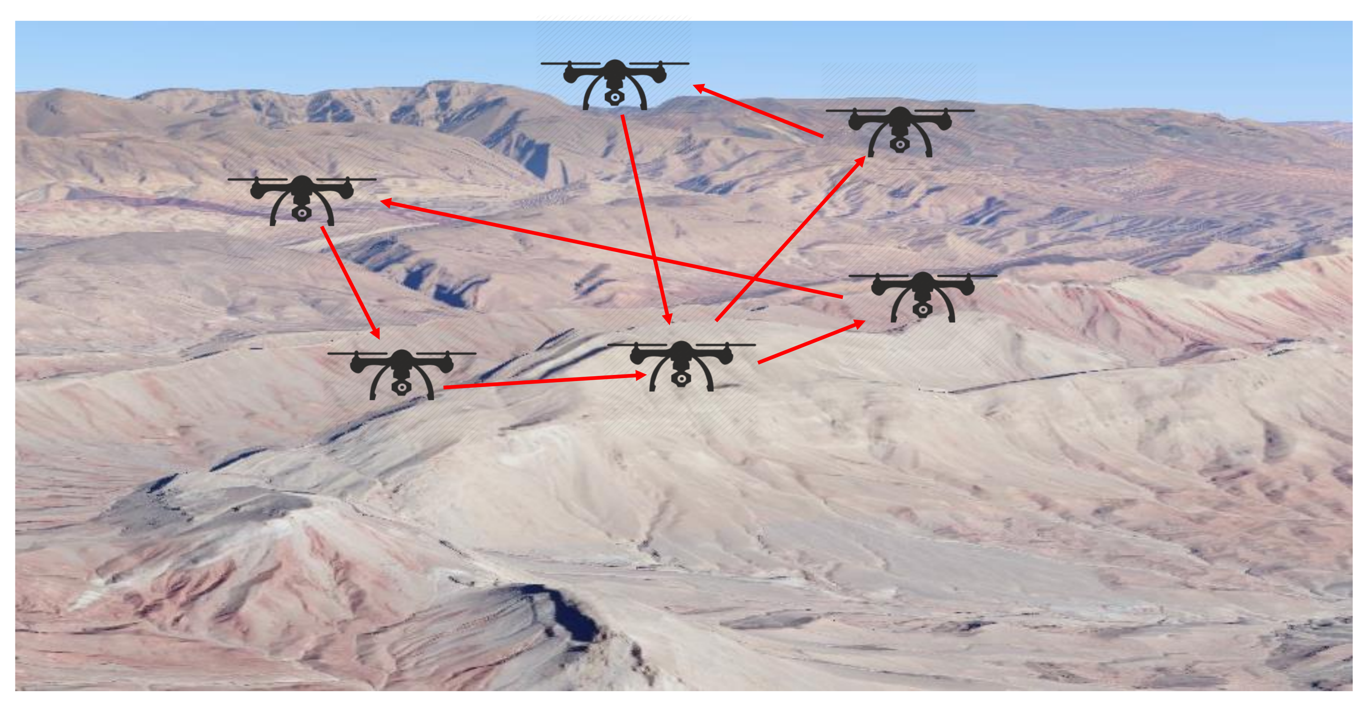
Each agent controls a state vector $x_i(t) \in \mathbb{R}^n$, which could be looked as an estimate of agent i to the solution of $Ax = b$. At time t, each agent only shares its state $x_i(t)$ to its current neighbors denoted by set $\mathcal{N}_i(t)$.

Application Examples

- The mode estimation of voltage oscillation In power networks
  - The phase sensors to monitor the power network is privately owned by different companies.
  - The companies don't want to share their information with their competitors.



- Large scaled distributed terrain-matching by UAVs



PURDUE UNIVERSITY