Distributed Optimization

Polina Alexeenko, Jiahe Chen, Zhaopeng Xu

Cornell University, Electrical and Computer Engineering
pa357@cornell.edu, jc3472@cornell.edu, zx273@cornell.edu

Motivation

- In many engineering applications, multiple intelligent agents are deployed and coordinated to achieve a common goal.
- Such multi-agent optimization system can be implemented in a decentralized manner, which is more robust and may require much less communication resources.
- Some motivating applications:
    1. Decentralized estimation.
    2. Big data and machine learning.
    3. Swarm robotics.
- How to design such system so that decisions of all agents will converge to the optimum as quick as possible?
- Many factors have been studied, including: the network topology, updating rules, communication protocols, noises, etc.

Outline

1. Distributed subgradient methods for multi-agent optimization (Nedic and Ozdaglar).
   - Basic setting.
   - Describe quality of solution to which agents converge.
2. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization (Nedic et al.).
   - Lazy Metropolis iteration.
   - Lazy Metropolis based subgradient method.
3. Decentralized averaging and optimization over directed graphs. (Nedic and Olshevsky)
   - Push-Sum iteration.
   - Push-Sum based subgradient method.

Model

Setting:

- Network with n agents
- Each agent has a convex function $f_i : \mathbb{R}^m \to \mathbb{R}$
- Agents want to cooperatively solve

$$\min \sum_{i=1}^{n} f_i\left(x\right) \quad \text{subject to } x \in \mathbb{R}$$

Procedure:

- at time k, each agent has estimate $x^i\left(k\right)$ of the optimal decision
- at time $k+1$ agents update estimate based on local information
- communication is asynchronous, local, and with time varying connectivity

Goal: Show that procedure converges to approximate global optimum

Information exchange model

Assumptions:

- There exists a scalar η with $0 < \eta < 1$ such that for all $i \in \{1, \ldots, n\}$
  1. $a_i^i(k) \geq \eta$ for all $k \geq 0$
  2. $a_j^i(k) \geq \eta$ for all $k \geq 0$ and all agents j communicating directly with agent i in the interval (t_k, t_{k+1})
  3. $a_j^i(k) = 0$ for all $k \geq 0$ and j otherwise
- Matrices $A(k) = \left[a^1(k), \ldots, a^n(k)\right]$ are doubly stochastic for all k.
- Agent communicate with neighbors at least once every B time slots

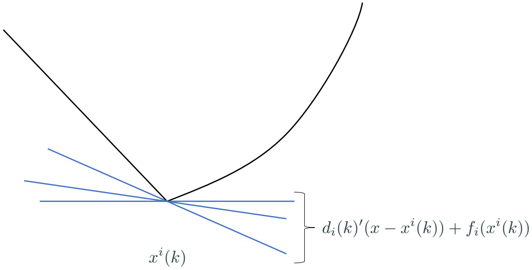
Optimization model

- Denote the optimal value: $f^* = \min \sum_{i=1}^{n} f_i(x)$
- Denote the optimal solution set

$$X^* = \left\{ x \in \mathbb{R}^n \middle| \sum_{i=1}^{n} f_i(x) = f^* \right\}$$

- Denote by $d_i(k)$ the subgradient of $f_i(x)$ at $x^i(k)$, i.e.,

$$f_i\left(x^i(k)\right) + d_i(k)'\left(x - x^i(k)\right) \leq f_i(x) \quad \text{for all } x \in \mathbb{R}^n$$



Optimization model

- Each agent updates their estimates according to

$$x^{i}\left(k+1\right)=\sum_{j=1}^{n}a_{j}^{i}\left(k\right)x^{j}\left(k\right)-\alpha d_{i}\left(k\right)$$

  where α is a stepsize used by all agents

- Defining the transition matrix $\Phi\left(k,s\right)$ as

$$\Phi\left(k,s\right)=A\left(s\right)A\left(s+1\right)\dots A\left(k-1\right)A\left(k\right)$$

  we can rewrite this as

$$\begin{aligned}x^{i}\left(k+1\right)=&\sum_{j=1}^{n}\left[\Phi\left(k,s\right)\right]_{j}^{i}x^{j}\left(s\right)\\&-\sum_{r=s+1}^{k}\left(\sum_{j=1}^{n}\left[\Phi\left(k,r\right)\right]_{j}^{i}\alpha d_{j}\left(r-1\right)\right)-\alpha d_{i}\left(k\right)\end{aligned}$$

Convergence of transition matrices

Theorem (Proposition 1 from (Nedic and Ozdaglar, 2007))

The entries $[\Phi(k,s)]_j^i$ *converge to* $\frac{1}{n}$ *as* $k \to \infty$ *with a geometric rate uniformly with respect to* i *and* j*, i.e., for all* $i, j \in \{1, \ldots, n\}$*,*

$$\left| [\Phi(k,s)]_i^j - \frac{1}{n} \right| \leq 2 \frac{1+\eta^{-B_0}}{1-\eta^{B_0}} \left(1-\eta^{B_0}\right)^{\frac{k-s}{B_0}}$$

where $B_0 = (n-1)B$

- Proof of convergence similar in spirit to (Tsitsiklis, 1984)
- Next step: how good is the consensus to which opinions converge?

Analysis

- Consider a "stopped" model where agents stop computing subgradients at some time $t_{\bar{k}}$ but keep exchanging information so that

$$\bar{x}^i(k) = x^i(k) \qquad \text{for all } k \leq \bar{k}$$

and

$$\bar{x}^i(k) = \sum_{j=1}^{n} [\Phi(k-1,0)]_j^i x^j(0) - \alpha \sum_{s=1}^{k} \left(\sum_{j=1}^{n} [\Phi(k-1,s)]_k^i d_j(s-1) \right)$$

- Denoting $\lim_{k\to\infty} \bar{x}^i(k) = y\left(\bar{k}\right)$ and relabeling, we can write

$$y(k+1) = y(k) - \frac{\alpha}{n} \sum_{j=1}^{n} d_j(k)$$

- Denote by $\{g_j(k)\}$ the sequence of subgradients of f_j at $y(k)$
- The bound on the objective value will be in terms of time-averaged vectors:

$$\hat{y}(k) = \frac{1}{k} \sum_{h=0}^{k-1} y(h) \quad \text{and} \quad \hat{x}^i(k) = \frac{1}{k} \sum_{h=0}^{k-1} x^i(h)$$

- Idea: We analyze $y(k)$ and show that $x(k)$ is close to $y(k)$

Analysis

Assumptions:

- The optimal solution set X^* is nonempty
- The subgradient sequences $\{d_j(k)\}$ and $\{g_j(k)\}$ are bounded by L
- The subgradients $\hat{g}_{ij}(k)$ of f_j at $\hat{x}^i(k)$ are bounded uniformly by $\hat{L}$
- Assume $\max_{1 \le j \le m} \left\| x^j(0) \right\| \le \alpha L$

Theorem (Proposition 3 in (Nedic and Ozdaglar, 2007))

(a) For every $i \in \{1, \ldots, n\}$, we have

$$\left\| y(k) - x^i(k) \right\| \leq 2\alpha L C_1 \quad \textit{for all } k \geq 0$$

(b) An upper bound on the objective value for each i is

$$f\left(\hat{x}^i(k)\right) \leq f^* + \frac{n dist\left(\frac{1}{n}\sum_{j=1}^{n} x^j(0), X^*\right)}{2\alpha k} + \alpha L\left(\frac{LC}{2} + 2n\hat{L}_1 C_1\right)$$

where $C_1 = 1 + \dfrac{n}{1-(1-\eta^{B_0})^{1/B_0}} \dfrac{1+\eta^{-B_0}}{1-\eta^{B_0}}$, $C = 1 + 8nC_1$

Proof sketch

Part (a): Show this using bounds on $\max_{1 \le j \le m} \left\| x^j(0) \right\|$, bounds on subgradients, and Proposition 1

Part (b):

Lemma (Lemma 5b from (Nedic Ozdaglar))

Let $\{g_j(k)\}$ be a sequence of subgradients such that $g_j(k) \in \partial f_j(y(k))$ for all $j \in \{1, \dots, n\}$ and $k \ge 0$. We then have

$$
\begin{aligned}
dist^2(y(k+1), X^*) \le\ & dist^2(y(k), X^*) \\
& - \frac{2\alpha}{n}\left[f(y(k)) - f(x)\right] + \frac{\alpha^2}{n^2}\sum_{j=1}^{n}\|d_j(k)\|^2 \\
& + \frac{2\alpha}{n}\sum_{j=1}^{n}\left(\|d_j(k)\| + \|g_j(k)\|\right)\left\|y(k) - x^j(k)\right\|
\end{aligned}
$$

Proof sketch (continued)

- Using the Lemma and Part (a) have for all $k \geq 0$

$$f\left(y\left(k\right)\right) - f^{*} \leq \frac{\mathrm{dist}^{2}\left(y\left(k\right), X^{*}\right) - \mathrm{dist}^{2}\left(y\left(k+1\right), X^{*}\right)}{2\alpha/n} + \frac{\alpha L^{2} C}{2}$$

- Summing over $k-1$ and dividing by k, we get

$$\frac{1}{k}\sum_{k=0}^{k-1} f\left(y\left(h\right)\right) - f^{*} \leq \frac{\mathrm{dist}^{2}\left(y\left(0\right), X^{*}\right)}{2\alpha/n} + \frac{\alpha L^{2} C}{2}$$

- By the convexity of f, we have

$$\frac{1}{k}\sum_{k=0}^{k-1} f\left(y\left(h\right)\right) \geq f\left(\hat{y}\left(k\right)\right)$$

which gives us

$$f\left(\hat{y}\left(k\right)\right) \leq f^{*} + \frac{m\mathrm{dist}^{2}\left(y\left(0\right), X^{*}\right)}{2\alpha k} + \frac{\alpha L^{2} C}{2}$$

Proof sketch (continued)

- From the definition of the subgradient, we have

$$f\left(\hat{x}^i(k)\right) \leq f(\hat{y}(k)) + \sum_{j=1}^{n} \hat{g}_{i,j}(k)' \left(\hat{x}^i(k) - \hat{y}(k)\right)$$

- Since the subgradients are bounded, we get

$$f\left(\hat{x}^i(k)\right) \leq f(\hat{y}(k)) + n\hat{L}_1 \left\|\hat{x}^i(k) - \hat{y}(k)\right\|$$

- Using the estimate in part (a), we have

$$f\left(\hat{x}^i(k)\right) \leq f^* + \frac{n\mathrm{dist}\left(\frac{1}{n}\sum_{j=1}^{n} x^j(0), X^*\right)}{2\alpha k} + \alpha L\left(\frac{LC}{2} + 2n\hat{L}_1 C_1\right)$$

Summary

- Nedic and Ozdaglar show that the iteration converges to a consensus in a simple setting
- Provide bounds on the quality of the result as a function of the number of iteration
- Next up: Consider a more sophisticated setting
  - Alternative update algorithms
  - Time-varying step-size

Notation

- Consider a time-varying network described by a sequence of stochastic matrices A^0, A^1, A^2,...
- Agents are represented by the vertex set $\{1, ..., n\}$ and links among agents are represented by the edge set $E_A = \{(i, j)|A_{i,j} > 0\}$.
- $[A_\alpha]$ denotes the threshold matrix of A, which is obtained from A by setting every element smaller than α to zero.
- At iteration k, each agent i sends messages to its out-neighbors $N_i^{out,k} = \{j|A_{j,i}^k > 0\}$ and receives messages from its in-neighbors $N_i^{in,k} = \{j|A_{i,j}^k > 0\}$. Each agent has out-degree $d_i^{out,k} = |N_i^{out,k}|$ and in-degree $d_i^{in,k} = |N_i^{in,k}|$.

Assumption

Assumption (Strong-connectivity condition)

- *The sequence of directed graphs G^0, G^1, G^2,... is B-strongly-connected. Namely a graph with the same vertex set and edge set as* $\bigcup_{k=lB}^{(l+1)B-1} E_{A^k}$ *is strongly connected for each* $l = 0, 1, 2, \ldots$.
- *Each node has a self-loop in every graph G^k.*

Lazy Metropolis iteration

Update rules:

- Consider a network with undirected graph: $(i, j) \in E_A \Leftrightarrow (j, i) \in E_A$.
- Consider a linear consensus process: $x^{k+1} = A^k x^k, \;\; k = 0, 1, ...,$ where A^k is stochastic.
- **Lazy Metropolis iteration:** each agent updates its vector with:

$$x_i^{k+1} = x_i^k + \sum_{j \in N_i^k} \frac{1}{2\max\{d_i^k, d_j^k\}} (x_j^k - x_i^k).$$

A^k associated with the Lazy Metropolis iteration is doubly stochastic.

Theorem (Consensus convergence over time-varying graphs)

For a sequence of doubly stochastic matrices A^0, A^1,..., if there exists an $\alpha > 0$ such that $G_{[A^0]_\alpha}$, $G_{[A^1]_\alpha}$,... satisfy the strong-connectivity condition, then $x(t)$ converges to consensus on the average:

$$\lim_{k \to \infty} x^k = \frac{1}{n} \sum_{i=1}^{n} x_i^0$$

Lazy Metropolis iteration

Convergence rate:

- **Convergence time** $T(n, \epsilon, \{A^0, A^1, ...\})$ is defined as the first k such that:

$$\|x^k - \overline{x}\| \leq \epsilon \|x^0 - \overline{x}\| \text{ with } \overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i^0.$$

Proposition (Convergence rate)

For a linear consensus process with doubly stochastic matrices A^k, the convergence time is:

$$T(n, \epsilon, \{A^0, A^1, ...\}) = O\Big(\frac{1}{1-\lambda} \ln \frac{1}{\epsilon}\Big) \;\; \textit{with} \;\; \lambda = \sup_{l \geq 0} \sigma_2(A^l),$$

where $\sigma_2(A^l)$ denotes the second-largest singular value of the matrix A^l.

- For the lazy Metropolis matrices on connected graphs $\lambda \leq 1$.
- Spectral gap $\dfrac{1}{1-\lambda}$ scales with network size n.

Convergence rate of lazy Metropolis iteration for different graph families:

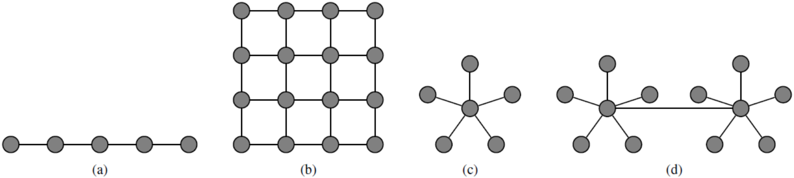


Figure 1: Examples of some graph families.

(a). $T(n, \epsilon, \{A^0, A^1, ...\}) = O(n^2 \log(1/\epsilon))$

(b). $T(n, \epsilon, \{A^0, A^1, ...\}) = O(n \log(n) \log(1/\epsilon))$

(c). $T(n, \epsilon, \{A^0, A^1, ...\}) = O(n^2 \log(1/\epsilon))$

(d). $T(n, \epsilon, \{A^0, A^1, ...\}) = O(n^2 \log(1/\epsilon))$

Lazy Metropolis based subgradient method

- A vector $g \in \mathbb{R}^d$ is called a subgradient of a convex function $h : \mathbb{R}^d \to \mathbb{R}$ at the point x if

$$h(y) \geq h(x) + g^T(y - x), \text{ for all } y \in \mathbb{R}^d.$$

- **Centralized subgradient method:**

$$x^{k+1} = x^k - \alpha^k g^k,$$

where g^k is a subgradient at the point x^k and α^k is a non-negative step-size.

- **Lazy Metropolis based subgradient method:**

$$x_i^{k+1} = \sum_{j \in N_i^k} A_{i,j}^k x_j^k - \alpha^k g_i^k,$$

where A^k is the adjacency matrix associated with the lazy Metropolis iteration and g_i^k is the subgradient of $f_i(\cdot)$ at x_i^k.

Lazy Metropolis based subgradient method

Theorem (Convergence for decentralized subgradient method)

Let $\mathcal{X}^*$ denote the set of minimizers of the function f. We assume that: (i) each f_i is convex; (ii) $\mathcal{X}^*$ is nonempty; (iii) g_i is bounded by constant L; (iv) A^k is double stochastic and satisfies strong-connectivity assumption; and (v) x_i^0 are the same for all nodes. Then we have:

- *If α^k satisfies:*

$$\sum_{k=0}^{\infty} \alpha^k = +\infty \text{ and } \sum_{k=0}^{\infty} [\alpha^k]^2 < \infty,$$

then for any $x^* \in \mathcal{X}^*$, we have that for all $i = 1, ..., n$,

$$\lim_{t\to\infty} f\left(\frac{\sum_{l=0}^{t} \alpha^l x_i^l}{\sum_{l=0}^{t} \alpha^l}\right) = f(x^*)$$

- *If we run for T steps with $\alpha^k = 1/\sqrt{T}$, then we have for all $i = 1, ..., n$,*

$$f\left(\frac{\sum_{l=0}^{T-1} y^l}{T}\right) - f(x^*) \leq \frac{(y^0 - x^*)^2 + L^2}{2\sqrt{T}} + \frac{L^2}{\sqrt{T}(1-\lambda)},$$

where $y^l = (1/n)\sum_{i=1}^{n} x_i^l$.

Lazy Metropolis based subgradient method

Convergence rate

- Define the ϵ-**convergence time** $T(n, \epsilon, \{A^0, A^1, ...\})$ as the first time when

$$f\left(\frac{\sum_{l=0}^{T-1} y^l}{T}\right) - f(x^*) \leq \epsilon.$$

- For lazy Metropolis based subgradient method, the ϵ-**convergence time** can be upper bounded as:

$$O\left(\frac{\max((y^0 - x^*)^4, L^4 P_n^2)}{\epsilon^2}\right).$$

- For some common graph families:
  1. path graph: $P_n = O(n^2)$.
  2. 2-dimensional grid: $P_n = O(n \log(n))$.
  3. complete graph: $P_n = O(1)$.
  4. star graph: $P_n = O(n^2)$.

Assumption

- At each time t, node i can only send messages to its out-neighbors in some directed graph $G(t)$.
- The sequence $G(t)$ is uniformly strongly connected (or, as it is sometimes called, B-strongly-connected).
- Notations of $N_i^{in}(t)$ and $N_i^{out}(t)$ is given below:

$$N_i^{in}(t) = \{j|(j,i) \in E(t)\} \cup \{i\},$$

$$N_i^{out}(t) = \{j|(i,j) \in E(t)\} \cup \{i\}$$

and $d_i(t)$ for the out-degree of node i, i.e.,

$$d_i(t) = |N_i^{out}(t)|.$$

Crucially, we will be assuming that every node i knows its out-degree $d_i(t)$ at every time t.

Push-Sum based subgradient method

Every node i maintains vector variables $x_i(t), w_i(t) \in R^d$, as well as a scalar variable $y_i(t)$. These quantities are updated according to the following rules: for all $t \geq 0$ and all $i = 1, ..., n$,

$$w_i(t+1) = \sum_{j \in N_i^{in}(t)} \frac{x_j(t)}{d_j(t)},$$

$$y_i(t+1) = \sum_{j \in N_i^{in}(t)} \frac{y_j(t)}{d_j(t)},$$

$$z_i(t+1) = \frac{w_i(t+1)}{y_i(t+1)}),$$

$$x_i(t+1) = w_i(t+1) - \alpha(t+1) g_i(t+1), \tag{1}$$

where $g_i(t+1)$ is a subgradient of the function $f_i(z)$ at $z = z_i(t+1)$. It is initiated with an arbitrary vector $x_i(0) \in R^d$ at node i, and with $y_i(0) = 1$ for all i. The stepsize $\alpha(t+1) > 0$ satisfies the following decay conditions

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \sum_{t=1}^{\infty} \alpha^2(t) < \infty, \tag{2}$$

$$\alpha(t) \leq \alpha(s) \text{ for all } t > s \geq 1.$$

Push-Sum based subgradient method

Theorem (Theorem 1 from (Angelia and Alex, 2014))

Suppose that:

a. *The graph sequence $G(t)$ is uniformly strongly connected.*
b. *Each function $f_i(z)$ is convex over R^d and the set $Z^* = Argmin_{z \in R^d} F(z)$ is nonempty.*
c. *The subgradients of each $f_i(z)$ are uniformly bounded, i.e., there exists $L_i < \infty$ such that for all $z \in R^d$,*

$$\|g_i\| \leq L_i \text{ for all subgradients } g_i \text{ of } f_i(z).$$

Then, the distributed subgradient-push method of Eq.(3) with the stepsize satisfying the conditions in Eq.(4) has the following property

$$\lim_{t \to \infty} z_i(t) = z^* \text{ for all } i \text{ and for some } z^* \in Z^*$$

Push-Sum based subgradient method

Theorem (Theorem 2 from (Angelia and Alex, 2014))

Suppose that all the assumptions of Theorem 1 hold, and let $\alpha(t) = \frac{1}{\sqrt{t}}$ *for* $t \geq 1$. *Moreover, suppose that every node* i *maintains the variable* $\tilde{z}_i(t) \in R^d$ *initialized at time* $t = 0$ *with any* $\tilde{z}_i(0) \in R^d$ *and updated by*

$$\tilde{z}_i(t+1) = \frac{\alpha(t+1)z_i(t+1) + S(t)\tilde{z}_i(t)}{S(t+1)} \text{ for } t \geq 0$$

where $S(0) = 0$ *and* $S(t) = \sum_{s=0}^{t-1} \alpha(s+1)$ *for* $t \geq 1$. *Then, we have for all* $t \geq 1, i = 1, ..., n$, *and any* $z^* \in Z^*$,

$$F(\tilde{z}_i(t+1)) - F(z^*) \leq \frac{n}{2}\frac{\|\bar{x}(0) - z^*\|_1}{\sqrt{t+1}} + \frac{L^2(1 + \ln t + 1)}{2n\sqrt{t+1}} +$$

$$\frac{24L\sum_{j=1}^{n}\|x_j(0)\|_1}{\delta(1-\lambda)\sqrt{t+1}} + \frac{24dL^2(1 + \ln t)}{\delta(1-\lambda)\sqrt{t+1}}$$

where $\bar{x}(0) = \frac{1}{n}\sum_{i=1}^{n} x_i(0)$.

Push-Sum based subgradient method

For this equation:

$$F(\tilde{z}_i(t+1)) - F(z^*) \leq \frac{n}{2}\frac{\|\bar{x}(0) - z^*\|_1}{\sqrt{t+1}} + \frac{L^2(1+\ln t+1)}{2n\sqrt{t+1}} +$$

$$\frac{24L\sum_{j=1}^{n}\|x_j(0)\|_1}{\delta(1-\lambda)\sqrt{t+1}} + \frac{24dL^2(1+\ln t)}{\delta(1-\lambda)\sqrt{t+1}}$$

where $\bar{x}(0) = \frac{1}{n}\sum_{i=1}^{n} x_i(0)$. It implies that, along the time-averages $\tilde{z}_i(t)$ for each node i, the network objective function $F(z)$ converges to the optimal objective value F^*, i.e.,

$$\lim_{t\to\infty} F(\tilde{z}_i(t)) = F^* \text{ for all i}$$

In this equation, the scalars λ and δ are functions of the graph sequence. Moreover, the closeness of λ to 1 measures the speed at which the (connectivity) graph sequence $G(t)$ diffuses the information among the nodes over time. Additionally, δ is a measure of the imbalance of influences among the nodes. Time-varying directed regular networks are uniform in influence and will have $\delta = 1$.

Push-Sum based subgradient method

- Define the matrix $A(t)$ that captures the weights used in the construction of $w_i(t+1)$ and $y_i(t+1)$ in Equation (3)

$$A_{ij}(t) = \begin{cases} 1/d_j(t) & \text{whenever } j \in N_i^{in}(t), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Lemma (Lemma 4 from (Angelia and Alex, 2014))

Given a graph sequence $G(t)$, define

$$\delta' \triangleq \inf_{t=0,1,\ldots} \left(\min_{1 \le i \le n} [1' A'(t) \ldots A'(0)]_i \right).$$

if the graph sequence $G(t)$ is uniformly strongly connected, then $\delta' \geq \frac{1}{n^{nB}}$. If each $G(t)$ is regular, then $\delta' = 1$.

Summary

- Consider the setting where a group of n agents collectively minimize

$$\sum_{i=1}^{n} f_i(x)$$

- Present algorithms for achieving consensus optimization on directed and undirected time-varying graphs.
- Present convergence results for each algorithm.
- Next steps:
  - Is it possible to achieve sub-quadratic convergence time for any graph without prior knowledge about the network?
  - What are the trade-offs involved in implementing the decentralized algorithm? How to efficiently allocate communication resources?