DECENTRALIZED AND DISTRIBUTED SOLUTIONS FOR EPIDEMIC CONTROL

Final Project of Distributed Optimization Course

Hossein Bijanrostami 810196370

Mohammad Mahmoodian 810196391

FEBRUARY 2, 2019

UNIVERSITY OF TEHRAN

Contents

System Model

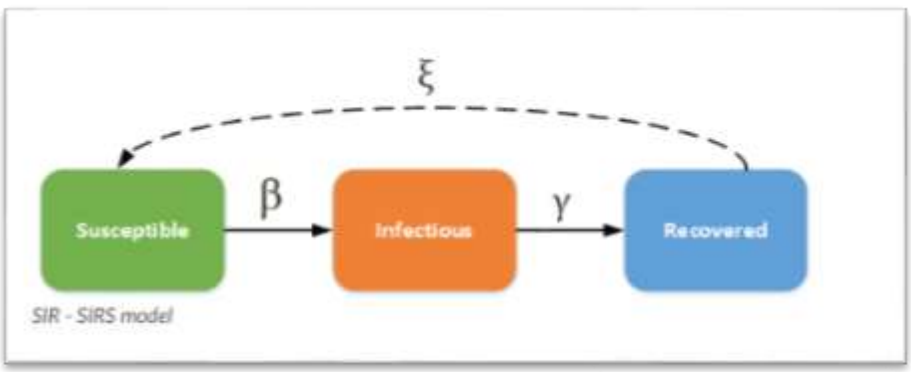
Epidemic outbreaks occur when there are reported cases of a contagious ailment beyond what is traditionally expected within a population.

Controlling the spread of epidemic outbreaks in populations is an age-old problem in human existence.

A significant mathematical contribution to understanding the spread dynamics of epidemics came from the work of William Hamer and Ronald Ross.

Kermack and McKendrik developed epidemic models involving ordinary differential equations based on a population model.
Population models partitioned a population of *N* individuals into 3 groups based on their state - Infected *I*, Susceptible *S*, and Recovered *R*.

In Kermack and Kendrick's model, infected individuals can independently infect susceptible individuals with some probability β. They can also recover with probability γ.



If *S*(*t*); *I*(*t*) and *R*(*t*) respectively denote the number of susceptible, infected and recovered individuals in the population at time *t*, and $s(t) = \frac{S(t)}{N}$; $i(t) = \frac{I(t)}{N}$ and $r(t) = \frac{R(t)}{N}$ respectively represent the fraction of susceptible, infected and recovered individuals, then $s(t) + i(t) + r(t) = 1$, and the population of each group evolves as follows:

$$\frac{ds(t)}{dt} = -\beta s(t)i(t),$$
$$\frac{dr(t)}{dt} = \delta i(t),$$
$$\frac{di(t)}{dt} = \beta s(t)i(t) - \delta i(t).$$

A key result from the epidemic process with dynamics is that a significantly large fraction of the population is infected by the epidemic if and only if $\tau_c = \frac{\beta}{\delta} > 1$

SIS Model

The infected individuals return to the susceptible state after infection. This model is appropriate for diseases that commonly have repeat infections, for example, the common cold (rhinoviruses) or sexually transmitted diseases like gonorrhea or chlamydia.

SIS without vital dynamics

Because individuals remain susceptible after infection, the disease attains a steady state in a population, even without vital dynamics. The ODE for the SIS model without vital dynamics can be analytically solved to understand the disease dynamics. The ODE is as follows:

$$\frac{dS}{dt} = -\frac{\beta SI}{N} + \gamma I$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$

At equilibrium, solving:

$$\frac{dI}{dt} = \beta I \left(1 - \frac{I}{N}\right) - \gamma I = 0$$

There are two equilibrium states for the SIS model, the first is $I = 0$ (disease free state), and the second is:

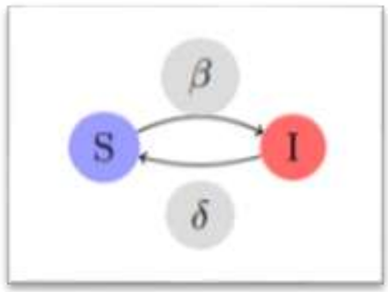
$$I = \frac{(\beta - \gamma)N}{\beta} = \left(1 - \frac{\gamma}{\beta}\right) N$$

For disease to spread, $\frac{dI}{dt} > 0$. Therefore, when $\frac{\beta}{\gamma} > 1$, the disease will spread and approach the second steady state; otherwise, it will eventually reach the disease-free state.

Spreading Model in Arbitrary Networks

We consider a network comprising N agents, where each agent can be in one of two states *susceptible* to the infectious disease or *infected* by the disease. As time evolves, the state of each agent $v_i \in V$ changes according to a stochastic process parameterized by its infection rate β_i, and curing rate δ_i.

As part of our model, we assume the respective infection and curing rates β_i and δ_i at the different agents v_i can be tuned by injecting vaccine and treatment resources.



The state of node v_i at time $t \geq 0$ is a binary random variable $X_i(t) \in \{0, 1\}$. The state $X_i(t) = 0$ indicates that node v_i is in the Susceptible state, which we denote as S and the state $X_i(t) = 1$ indicates that node v_i is in the Infected state, denoted I.

Let the vector of states be defined as $X(t) = \left(X_1(t); \dots; X_n(t)\right)^T$. Next, we describe the state evolution of each node

- Suppose node v_i is susceptible to the infection at time t, it can transition to infected state during a small time interval $(t;\ t + \Delta t)$ with a probability that depends on its infection rate $\beta i > 0$, the strength of its incoming connections from its neighbors $\{a_{ij};\ for\ j \in N_i^{in}\}$, as well as the states of its in-neighbors $\{Xj\ (t)\ ;\ for\ j \in N_i^{in}\}$.

$$\Pr\left(X_i(t+\Delta t) = 1 | X_i(t) = 0, X(t)\right) = \sum_{j \in N_i^{in}} a_{ij}\beta_i X_j(t)\,\Delta t + o(\Delta t), \quad (2.1)$$

- If node v_i is infected, its probability of transitioning to a susceptible state in the time interval $[t;\ t + \Delta t)$ depends on its recovery rate δ_i and is given by

$$\Pr(X_i(t+\Delta t) = 0 | X_i(t) = 1, X(t)) = \delta_i \Delta t + o(\Delta t).$$

Suppose we denote the probability of infection at node v_i as $p_i(t)$. Then, $pi(t) \triangleq \Pr(Xi(t) = 1)$ and taking into account the fact that $1 - p_i(t) = \Pr\{X_i(t) = 0\}$, the Markov differential equation for state $X_i(t) = 1$ can be approximated by

$$\frac{dp_i(t)}{dt} = (1 - p_i(t))\beta_i \sum_{j=1}^{n} a_{ij}p_j(t) - \delta_i p_i(t),$$

Which we can more compactly write as

$$\frac{d\boldsymbol{p}(t)}{dt} = (BA - D)\,\boldsymbol{p}(t) - P(t)\,BA\boldsymbol{p}(t),$$

where $\mathbf{p}\,(\mathrm{t}) = \left(\mathrm{p}_1\,(\mathrm{t}); \dots; \mathrm{p}_\mathrm{n}\,(\mathrm{t})\right)^\mathrm{T}$, $\mathrm{B} = \mathrm{diag}(\beta_\mathrm{i})$ is a diagonal matrix comprising the infection rates across the nodes, $\mathrm{D} = \mathrm{diag}\,(\delta_\mathrm{i})$ is a diagonal matrix comprising the curing rates across the nodes, and $\mathrm{P(t)} = \mathrm{diag}(\mathrm{p_i}\,(\mathrm{t}))$.

Of interest to us is to derive a sufficient condition, based on the dynamics in recent equation, that guarantee the probability of infections converge to zero exponentially fast across the network.

Proposition 1. *Given the SIS epidemic model with uniform infection and recovery rates across all agents, the probability of infection (from an initial infection), converges to zero exponentially fast if*

$$\frac{\beta}{\delta} < \tau_c = \frac{1}{\lambda_1(A)},$$

where β and δ are respectively the infection and curing rates, and $\lambda_1(\mathrm{A})$ is the maximum eigenvalue of the network adjacency matrix *A*.

The DFE is the equilibrium at which the probability of infection is zero; that is, the expected number of agents in infected state is zero and all agents in the network are only susceptible to the infection.

Proposition 2. *Consider the heterogeneous networked SIS model, with $A \geq 0$, B, $D > 0$. Then, if the eigenvalue with largest real part of $BA - D$ satisfies*

$$\mathbb{R}\left[\lambda_1\left(BA - D\right)\right] \leq -\varepsilon,$$

for some $\epsilon > 0$, the disease-free equilibrium $(\mathrm{p}^* = 0)$ is globally exponentially stable, i.e., $\|\mathrm{p(t)}\| \leq \|\mathrm{p}(0)\|\,\mathrm{K}\exp(-\epsilon \mathrm{t})$, for some $\mathrm{K} > 0$.

Epidemic Control Problem

An assumption in our model and problem set up is that the infection rate β_i and recovery rate δ_i for node v_i can be adjusted at a cost. We assume preventive resources (vaccines) at node v_i reduces its infection rate β_i within feasible intervals $0 < \underline{\beta_i} \le \beta_i \le \overline{\beta_i}$, with associated cost $f_i(\beta_i)$; and that treatment resources (antidotes) at node v_i ups its recovery rate δ_i within feasible intervals $0 < \underline{\delta_i} \le \delta_i \le \overline{\delta_i}$ and accrues a cost $g_i(\delta_i)$. Our choices for the associated cost functions is such that $f_i(\beta_i)$ is monotonically decreasing w.r.t. β_i and $g_i(\delta_i)$ is monotonically increasing w.r.t. δ_i.

Problem 1. *Given a vaccination cost function* $f_i(\beta_i)$*, for* β_i *within a feasible interval* $0 < \underline{\beta_i} \le \beta_i \le \overline{\beta_i}$*, and treatment cost function* $g_i(\delta_i)$*, for* δ_i *within a feasible interval* $0 < \underline{\delta_i} \le \delta_i \le \overline{\delta_i}$*, determine the optimal allocation of vaccines and treatment resources to control the spread of an epidemic outbreak with an asymptotic exponential decaying rate* ϵ *for a minimum cost. Mathematically, this problem can be formulated as follows:*

$$\underset{\{\beta_i,\delta_i\}_{i=1}^n}{minimize} \sum_{i=1}^{n} f_i(\beta_i) + g_i(\delta_i)$$

$$subject\ to\ \mathbb{R}\left[\lambda_1\left(diag(\beta_i)A - diag(\delta_i)\right)\right] \le -\varepsilon,$$

$$\underline{\beta}_i \le \beta_i \le \overline{\beta}_i,$$

$$\underline{\delta}_i \le \delta_i \le \overline{\delta}_i,\ i = 1, \ldots, n,$$

Where $f_i(\beta_i)$ is the vaccination cost incurred at node v_i, $g_i(\delta_i)$ is the treatment cost at node v_i, A is the adjacency matrix associated with the network.

Lemma 1. *(Perron-Frobenius) Let* M *be a nonnegative, irreducible matrix. Then, the following statements about its spectral radius,* $\rho(M)$*, hold:*
1. $\rho(M) > 0$ *is a simple eigenvalue of* M*,*
2. $Mu = \rho(M)u$*, for some* $u \in R_{++}^n$*, and*
3. $\rho(M) = \inf\{\lambda \in R : Mu \le \lambda u \text{ for } u \in R_{++}^n\}$.

Proposition 3. *Consider the* $n \times n$ *nonnegative, irreducible matrix* $M(x)$ *with entries being either* 0 *or posynomials with domain* $x \in S \subseteq R_{++}^k$*, where* S *is defined as* $S = \prod_{i=1}^{m} \{x \in R_{++}^n :$

$f_i(x) \leq 1\}$, f_i *being posynomials. Then, we can minimize* $\lambda_1(M(x))$ for $x \in S$ *by solving the following GP:*

$$\begin{aligned} &\underset{\lambda,\{u_i\}_{i=1}^n,\mathbf{x}}{\text{minimize}}\ \lambda \\ &\text{subject to}\ \frac{\sum_{j=1}^n M_{ij}(\mathbf{x})\,u_j}{\lambda u_i} \leq 1,\ i=1,\ldots,n, \\ &\qquad f_i(\mathbf{x}) \leq 1,\ i=1,\ldots,m. \end{aligned}$$

Geometric Programming formulation for Resource Allocation

Before stating the solution within the GP framework, note that the first constraint in **Problem 1** cannot be directly expressed as a set of posynomial functions as was done in **Proposition 3** because of the negative coefficient of the term $diag(\delta_i)$. To overcome this challenge for **Problem 1**, an equivalent reformulation is derived.

Theorem 1. *Given a strongly connected graph* G *with adjacency matrix* A*, posynomial cost functions* $\{f_I(\beta_i);\ g_I(\delta_i)\}_{i=1}^n$ *; bounds on the infection and recovery rates* $0 < \underline{\beta_i} \leq \beta_I \leq \overline{\beta_i}$ *and* $0 < \underline{\delta_I} \leq \delta_i \leq \overline{\delta_I}$, $I = 1, \ldots, n$*, and a desired exponential decay rate* ϵ*. Then, the optimal investment on vaccines and antidotes for node* v_I *to solve* ***Problem 1*** *are* $f_I(\beta_i^*)$ *and* $g_I(\Delta + 1 - \hat{\delta}_i^*)$*, where* $\Delta \triangleq \max\{\epsilon, \overline{\delta_\imath}$ for $i = 1, \ldots, n\}$ *and* $\beta_i^*, \overline{\delta}_i^*$ *are the optimal solution for* β_i *and* $\overline{\delta}_i$ *in the following GP:*

$$\text{minimize} \sum_{i=1}^{N} f_i(\beta_i) + g_i(\Delta + 1 - \hat{\delta}_i)$$

$$s.t. \qquad \frac{\beta_i \sum_{j=1}^N A_{ij}u_j + \hat{\delta}_i u_i}{(\Delta + 1 - \bar{\epsilon})u_i} \leq 1 \quad \forall\ all\ i_s \tag{1}$$

$$\underline{\beta_i} \leq \beta_i \leq \bar{\beta}_i \qquad \forall\ all\ i_s \tag{2}$$

$$\Delta + 1 - \bar{\delta}_i \leq \hat{\delta}_i \leq \Delta + 1 - \underline{\delta_i} \qquad \forall\ all\ i_s \tag{3}$$

Given the equations above, we have reached a well-behaved centralized optimization problem to apply the goal of this project.

Now based on what we mentioned at the beginning, we need to solve the problem above using Decentralized and Distributed algorithms. According what we had in seminars during the semester, we conclude to utilize both decentralized ADMM and distributed ADMM in order to receive better and more sophisticated results.

Decentralized ADMM Algorithm

Based on what we had in seminars, Decentralized ADMM algorithms tries to make an agreement among all the Agents using a Coordinator, which means all the Agents minimize their own cost function by using the parameters of other Agents which are provided by the Coordinator. The common parameter of all the agents is vector ***u.***

we introduce *n*-dimensional variables $u_i \in R^n$ representing a local copy of the global variable $u = (u_1, \dots, u_n)^T$ at each node v_i.

So by applying Decentralized ADMM to the well-behaved centralized optimization problem, the new decentralized problem is as follows:

$$\text{minimize} \sum_{i=1}^{N} f_i(\beta_i) + g_i\left(\Delta + 1 - \hat{\delta}_i\right)$$

$$s.t. \qquad \frac{\beta_i \sum_{j=1}^{N} A_{ij} u_i^j + \hat{\delta}_i u_i^i}{(\Delta + 1 - \bar{\epsilon}) u_i^i} \le 1 \quad \forall\, all\ i_s \qquad (1)$$

$$\underline{\beta_i} \le \beta_i \le \bar{\beta}_i \qquad \forall\, all\ i_s \qquad (2)$$

$$\Delta + 1 - \bar{\delta}_i \le \hat{\delta}_i \le \Delta + 1 - \underline{\delta_i} \qquad \forall\, all\ i_s \qquad (3)$$

$$u_i = z \quad \forall\, all\ i_s \qquad (4)$$

So the Lagrangian function of problem above is:

$$L_\rho = \sum_{i=1}^{N} f_i(\beta_i) + g_i\left(\Delta + 1 - \hat{\delta}_i\right) + y_i^T (u_i - z) + \frac{\rho}{2} \left|\left|u_i - z\right|\right|^2$$

Now as it is obvious the Lagrange function of each agent can be decoupled from the other agent's Lagrange function. So for each agent we have:

$$L_\rho^i = f_i(\beta_i) + g_i\left(\Delta + 1 - \hat{\delta}_i\right) + y_i^T (u_i - z) + \frac{\rho}{2} \left|\left|u_i - z\right|\right|^2$$

Hence, each agent can optimize its own problem by receiving the ***z*** vector from the coordinator***.*** After each agent optimizes its own problem, all the agents send their copy of ***u*** vector to the coordinator. Accordingly the coordinator will update ***z*** and ***y*** (Lagrange multiplier) using the information that is received from the agents.

Finally the Decentralized ADMM algorithm is as follows:

Algorithm 1. Decentralized ADMM algorithm

a. Initializing the parameters $z^0, u_i^0, y_i^0, \beta_i^0, \delta_i^0$
   At iteration k:
b. Each agent optimize its own problem and then sends the related u_i to the coordinator.

$$\beta_i^{k+1}, \delta_i^{k+1}, u_i^{k+1} = \arg\min L_\rho^i(\beta_i, \delta_i, u_i, y_i^k, z^k)$$
$$= f_i(\beta_i) + g_i\left(\varDelta + 1 - \hat{\delta}_i\right) + {y_i^k}^T(u_i - z^k) + \frac{\rho}{2}\left\| u_i - z^k \right\|^2$$

$$s.t. \qquad \frac{\beta_i \sum_{j=1}^{N} A_{ij} u_i^j + \hat{\delta}_i u_i^i}{(\varDelta + 1 - \bar{\epsilon}) u_i^i} \leq 1 \qquad (1)$$

$$\underline{\beta_i} \leq \beta_i \leq \bar{\beta}_i \qquad (2)$$

$$\Delta + 1 - \bar{\delta}_i \leq \hat{\delta}_i \leq \Delta + 1 - \underline{\delta_i} \qquad (3)$$

$$\prod_{j=1}^{n} u_i^j = 1 \qquad (4)$$

Normalization of the vectors u_i is to ensure that the local estimates u_i have the same direction.

c. In this step coordinator updates z and Lagrange multiplier y using the information collected from the agents.

$$z^{k+1} = \frac{\sum_{i=1}^{N} u_i^{k+1}}{N} + \frac{1}{\rho}\frac{\sum_{i=1}^{N} y_i^k}{N}$$

$$y_i^{k+1} = y_i^k + \rho(u_i^{k+1} - z^{k+1})$$

d. Go to b until $\left\| z^{k+1} - z^k \right\| < \epsilon$

Distributed ADMM Algorithm

According what we had in previous section in decentralized method, each agent performs its own optimization problem and interact the results with a coordinator, actually the coordinator was responsible for the management of the agents and the constraints. But in distributed algorithms there is nothing like a coordinator and all the nodes must interact with themselves.

So given the explanation above the distributed ADMM algorithm for each node can be expressed as:

$$\text{minimize} \sum_{i=1}^{N} f_i(\beta_i) + g_i\big(\Delta + 1 - \hat{\delta}_i\big)$$

$$s.t. \qquad \frac{\beta_i \sum_{j=1}^{N} A_{ij} u_i^j + \hat{\delta}_i u_i^i}{(\Delta + 1 - \bar{\epsilon}) u_i^i} \le 1 \quad \forall\ all\ i_s \qquad (1)$$

$$\underline{\beta_i} \le \beta_i \le \bar{\beta}_i \qquad \forall\ all\ i_s \qquad (2)$$

$$\Delta + 1 - \bar{\delta}_i \le \hat{\delta}_i \le \Delta + 1 - \underline{\delta_i} \qquad \forall\ all\ i_s \qquad (3)$$

$$\prod_{j=1}^{n} u_i^j = 1 \qquad (4)$$

$$u_i = z_{ij} \quad and \quad u_j = z_{ij} \qquad (v_i, v_j) \in E$$

The constraints $u_i = z_{ij}$ and $u_j = z_{ij}$ imply that for all pairs of agents (v_i, v_j) that form an edge, the feasible set is such that $u_i = u_j$. Assuming a strongly connected contact network, these local consensus constraints imply that feasible solutions must satisfy $u_i = u_j$ for all, not necessarily neighboring, pairs of agents v_i and v_j.

So based on problem above, the augmented Lagrangian is as follows:

$$\begin{aligned}
\Gamma_i(k+1) = \arg\min_{\mathbf{u}_i, \beta_i, \delta_i,} \quad & f_i(\beta_i) + g_i(\delta_i) + \sum_{j \in N(i)} \alpha_{ij}^T(\mathbf{u}_i - \mathbf{z}_{ij}(k)) + \gamma_{ij}^T(\mathbf{u}_j(k) - \mathbf{z}_{ij}(k)) \\
& + \frac{\rho}{2} \sum_{j \in N(i)} \|\mathbf{u}_i - \mathbf{z}_{ij}(k)\|_2^2 + \|\mathbf{u}_j(k) - \mathbf{z}_{ij}(k)\|_2^2 \\
\text{subject to} \quad & \frac{\beta_i \sum_{j=1}^{n} A_{ij} \mathbf{u}_i^j + \delta_i \mathbf{u}_i^i}{\mathbf{u}_i^i} \le 1, \\
& \prod_{j=1}^{n} \mathbf{u}_i^j = 1, \\
& \underline{\delta} \le \delta_i \le \overline{\delta} \\
& \underline{\beta} \le \beta_i \le \overline{\beta}.
\end{aligned}$$

And the dual variables updates are:

$$\alpha_{ij}(k+1) = \alpha_{ij}(k) + \frac{\rho}{2}(\mathbf{u}_i(k) - \mathbf{u}_j(k)) \ \forall\ j \in N(i)$$
$$\gamma_{ij}(k+1) = \gamma_{ij}(k) + \frac{\rho}{2}(\mathbf{u}_j(k) - \mathbf{u}_i(k)) \ \forall\ j \in N(i).$$

Now the paper has introduced a variable $\phi_i(k) = \sum_{j\in N(i)} \Big(\alpha_{ij}(k) + \gamma_{ji}(k) \Big) \quad \forall\, v_i \in V$

Therefore by using the new introduced variable and augmented Lagrangian the Distributed ADMM algorithm can be expressed:

Algorithm 2. Distributed Algorithm

Algorithm 4 Distributed ADMM for solving

1: **Given** initial variables $\beta_i(0), \delta_i(0) \in \mathbb{R}, \quad \mathbf{u}_i(0), \phi_i(0) \in \mathbb{R}^n$ for each agent $v_i \in \mathcal{V}$.

2: **Set** $k = 1$

3: **repeat**

4: For all $v_i \in \mathcal{V}$

$$\phi_i(k+1) = \phi_i(k) + \rho \sum_{j\in N(i)} (\mathbf{u}_i(k) - \mathbf{u}_j(k))$$

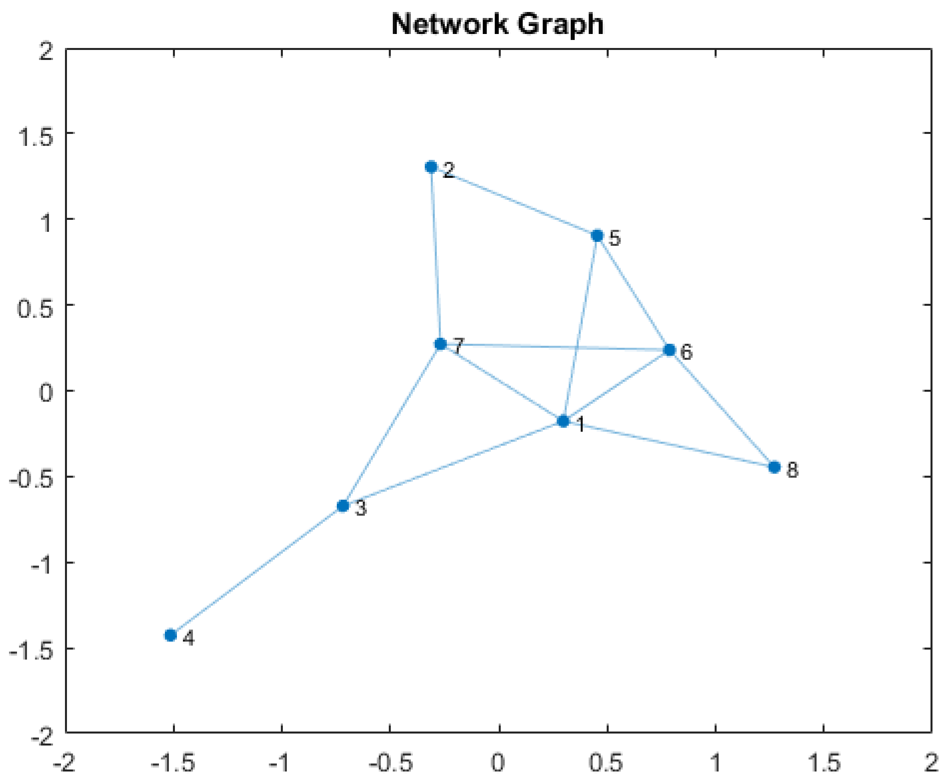
$$\begin{aligned}\Gamma_i(k+1) = \arg\min_{\beta_i,\delta_i,\mathbf{u}_i} \quad & f_i(\beta_i) + g_i(\delta_i) + \phi_i^T(k+1)\mathbf{u}_i \\ & + \rho \sum_{j\in N(i)} \|\mathbf{u}_i - \frac{\mathbf{u}_i(k)+\mathbf{u}_j(k)}{2}\|_2^2 \\ \text{subject to} \quad & \frac{\beta_i \sum_{j=1}^n A_{ij}\mathbf{u}_i^i + \delta_i \mathbf{u}_i^j}{\mathbf{u}_i^i} \leq 1, \\ & \prod_{j=1}^n \mathbf{u}_i^j = 1, \\ & \underline{\delta} \leq \delta_i \leq \overline{\delta} \\ & \underline{\beta} \leq \beta_i \leq \overline{\beta}.\end{aligned}$$

5: **Set** $k = k+1$

6: **until** $\sum_{i=1}^n \sum_{j\in N(i)} \|\mathbf{u}_i(k) - \mathbf{u}_j(k)\| \leq \eta$, for η arbitrarily small.

Numerical Results

Graph Generation



Network Configuration

```
specRadA = max(eig(A));
deltaMax = 0.8;
deltaMin = 0.08;
epsilonBar = 0.1;
DeltaTilda = max(epsilonBar,deltaMax);
epidTresh = (1-deltaMax)/specRadA;
betaMax = 4*epidTresh;
betaMin = 0.3*betaMax;
max(eig(diag(betaMax*ones(1,N))*A - diag(deltaMin*ones(1,N))))
max(eig(diag(betaMax*ones(1,N))*A - diag(deltaMax*ones(1,N))))
max(eig(diag(betaMin*ones(1,N))*A - diag(deltaMin*ones(1,N))))
max(eig(diag(betaMin*ones(1,N))*A - diag(deltaMax*ones(1,N))))
```

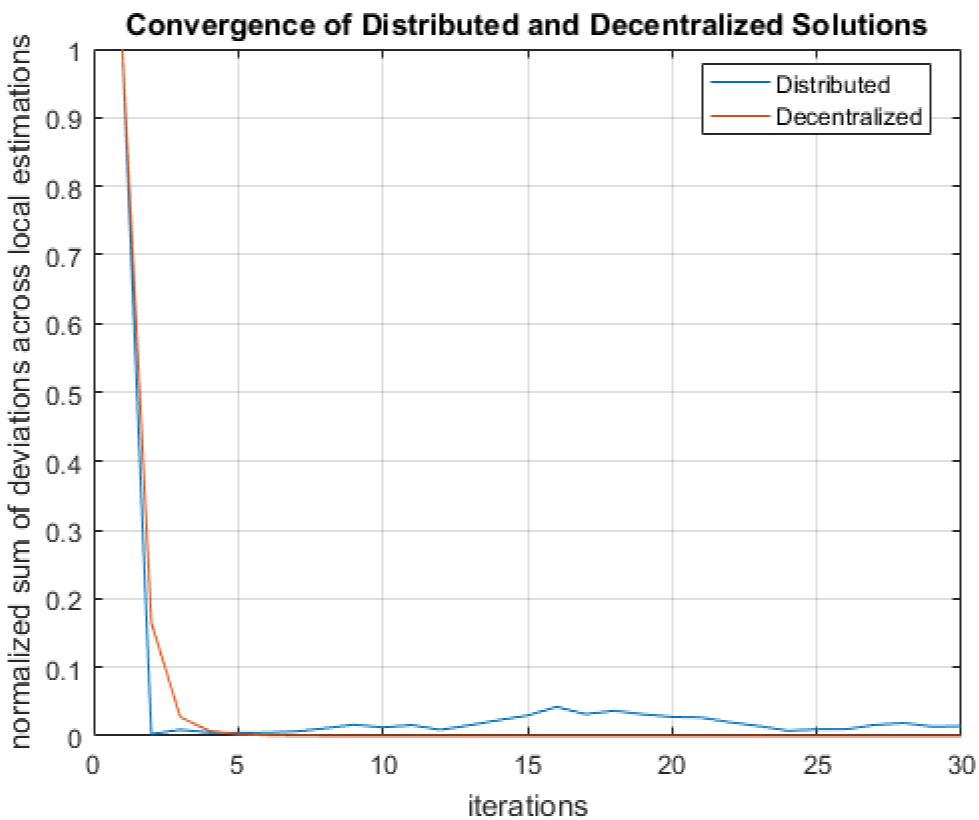
ans = 0.7200
ans = -4.4176e-16
ans = 0.1600
ans = -0.5600

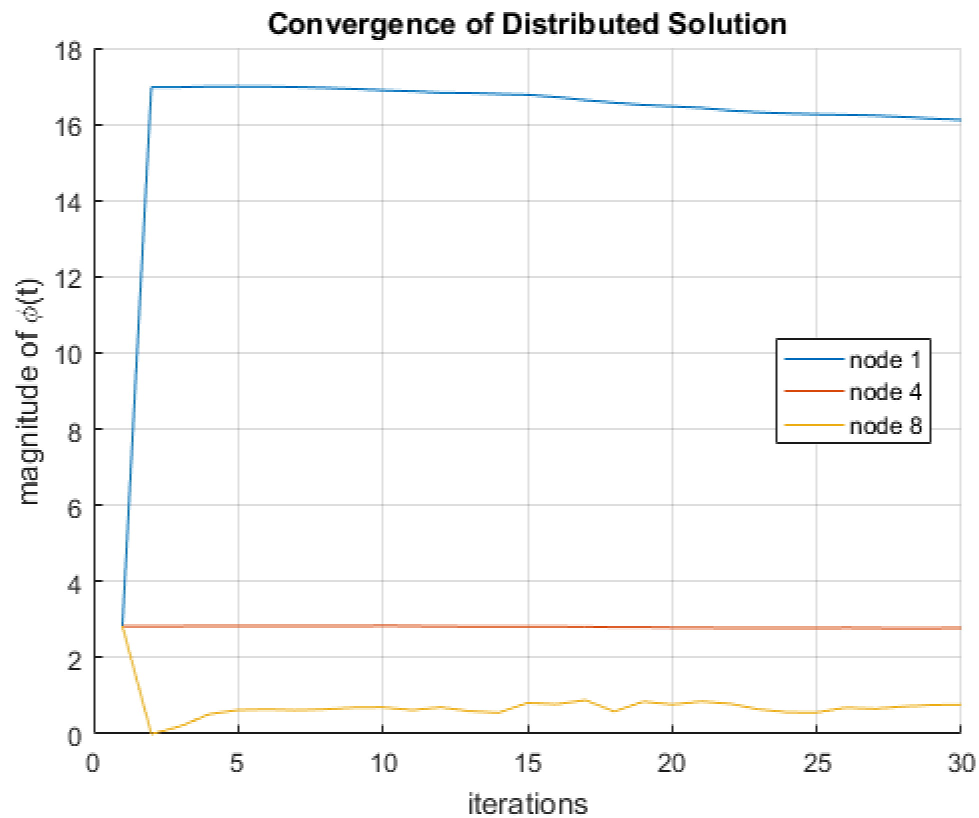
Final Results

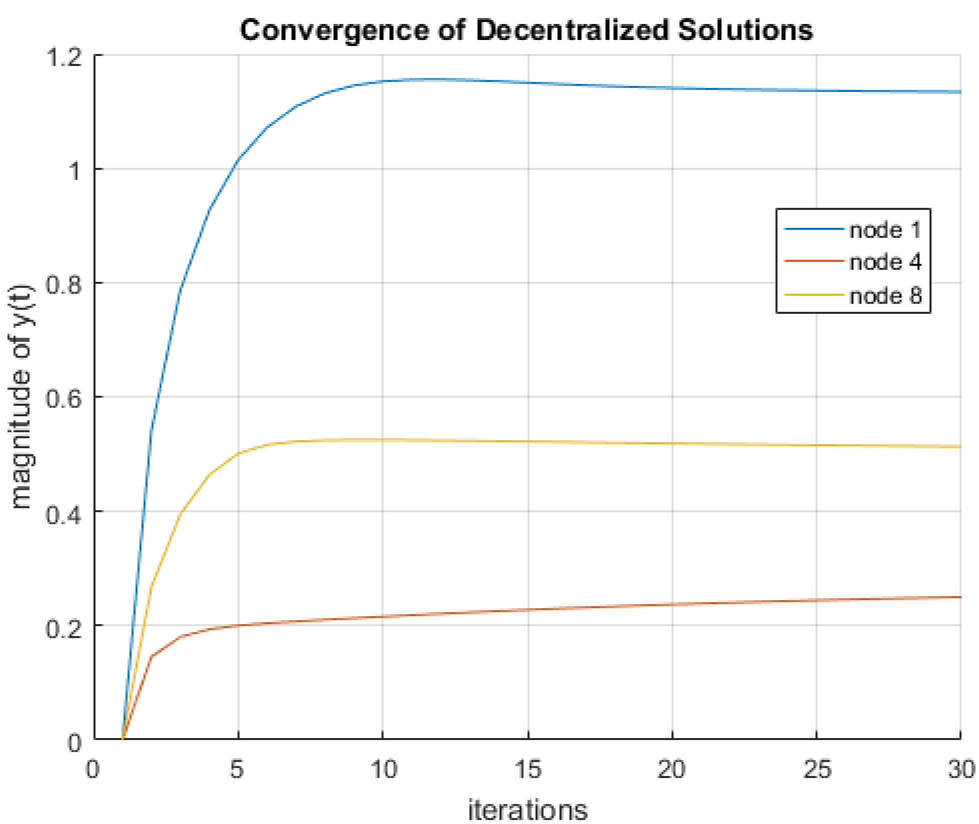
For above network we show convergence rate and analyze three special nodes that one of them has least degree and one of them has most degree and another one is a typical node. As you can see the node with highest degree is a central node so it has important role, so we should assign most vaccines to this node and in our resource allocation it will gain the most resources. For the lowest degree node everything is reversed.

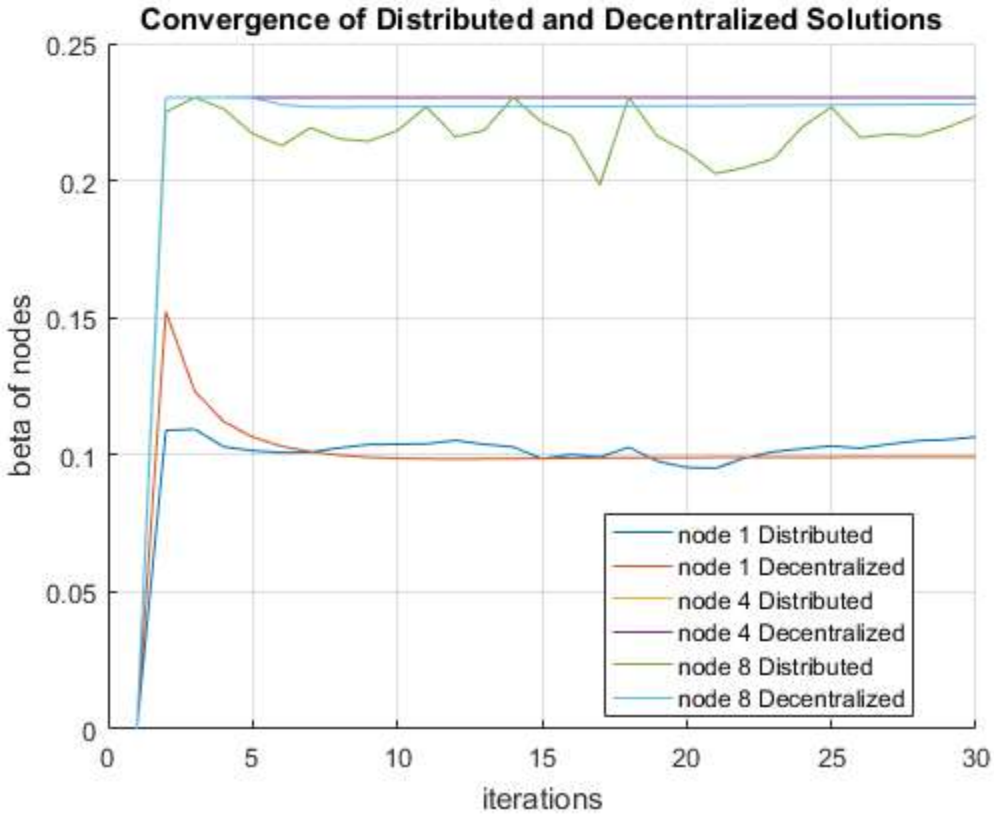
For comparison of Distributed algorithm and Decentralized we can say that Decentralized approach has better performance based on speed of convergence and accuracy of optimization but the main bottleneck of decentralized algorithm is the need of extra agent named 'coordinator', That forces extra costs.

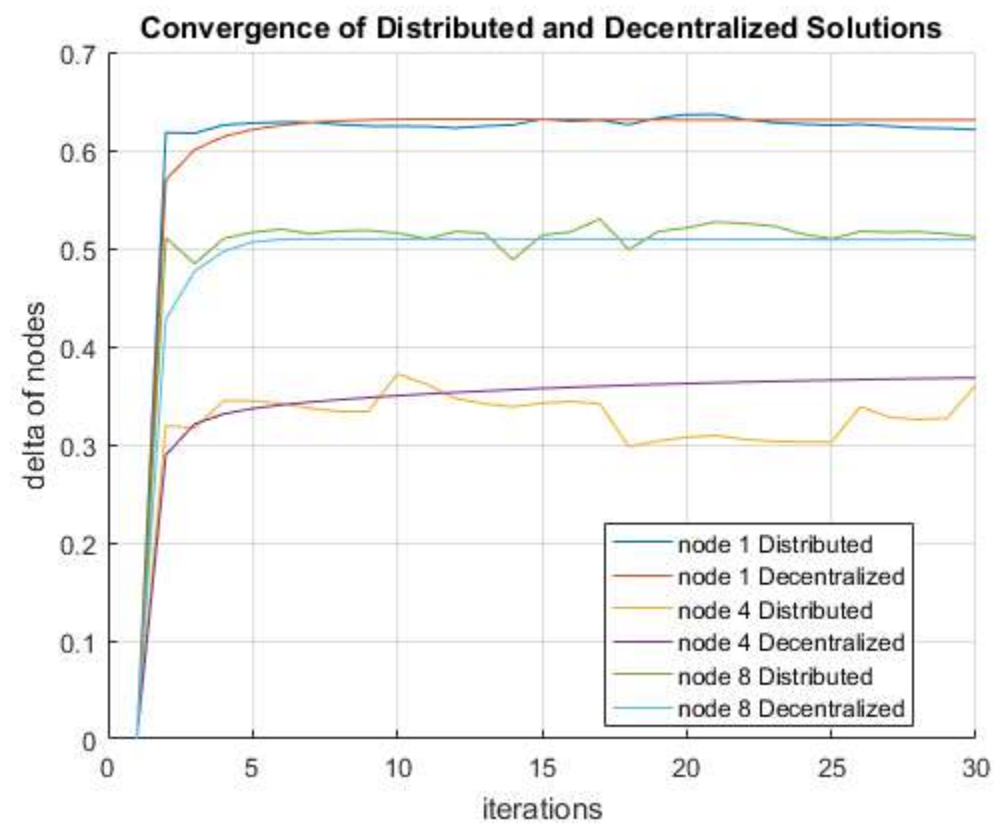
```
Epsilon is 0.1000
DFE parameter of distributed solution that should be less than minus epsilon is: -0.0969
DFE parameter of decentralized solution that should be less than minus epsilon is: -0.1001
```




Convergence of Distributed Solution
magnitude of ϕ(t)
iterations
node 1
node 4
node 8
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25
30


Convergence of Decentralized Solutions
magnitude of y(t)
iterations
node 1
node 4
node 8
0
0.2
0.4
0.6
0.8
1
1.2
0
5
10
15
20
25
30


Convergence of Distributed and Decentralized Solutions
beta of nodes
iterations
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25
30
node 1 Distributed
node 1 Decentralized
node 4 Distributed
node 4 Decentralized
node 8 Distributed
node 8 Decentralized


Convergence of Distributed and Decentralized Solutions
delta of nodes
iterations
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25
30
node 1 Distributed
node 1 Decentralized
node 4 Distributed
node 4 Decentralized
node 8 Distributed
node 8 Decentralized

Reference

[1] C. Enyioha, A. Jadbabaie, V. M. Preciado, and G. J. Papappas, “Distributed resource allocation for epidemic control,”, arXive:1501.01701,2015.