

Decentralized and Distributed solutions for epidemic control

Final Project of Distributed Optimization Course



February 2, 2019

University of TEHRAN

Hossein Bijanrostami 810196370

Mohammad Mahmoodian 810196391

Contents

[System Model 2](#_Toc11718)

[SIS Model 3](#_Toc11719)

[SIS without vital dynamics 3](#_Toc11720)

[Spreading Model in Arbitrary Networks 4](#_Toc11721)

[Epidemic Control Problem 6](#_Toc11722)

[Geometric Programming formulation for Resource Allocation 7](#_Toc11723)

[Decentralized ADMM Algorithm 8](#_Toc11724)

[Distributed ADMM Algorithm 9](#_Toc11725)

[Numerical Results 11](#_Toc11726)

[Graph Generation 11](#_Toc11727)

[Network Configuration 12](#_Toc11728)

[Final Results 13](#_Toc11729)

[Reference 16](#_Toc11730)

# System Model

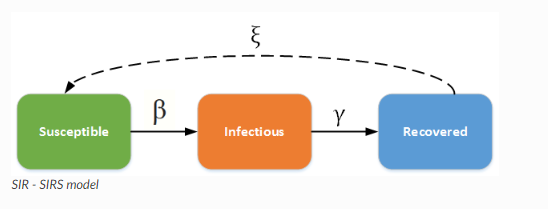
Epidemic outbreaks occur when there are reported cases of a contagious ailment beyond what is traditionally expected within a population.

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

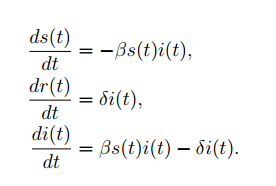
A significant mathematical contribution to understanding the spread dynamics of epidemics came from the work of William Hamer and Ronald Ross.

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 . They can also recover with probability .



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 ; and respectively represent the fraction of susceptible, infected and recovered individuals, then , and the population of each group evolves as follows:



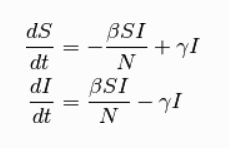
A key result from the epidemic process with dynamics is that a significantly large fraction of the population is infected by the epidemic if and only if

# SIS Model

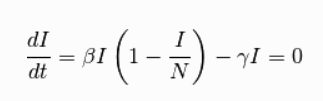
The infected individuals return to the susceptible state after infection. This model is appropriate for diseases that commonly have repeat infections, for example, the common cold (rhinoviruses) or sexually transmitted diseases like gonorrhea or chlamydia.

## [SIS without vital dynamics](https://institutefordiseasemodeling.github.io/Documentation/general/model-si.html#id12)

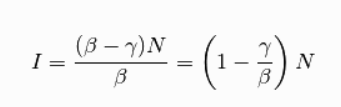
Because individuals remain susceptible after infection, the disease attains a steady state in a population, even without vital dynamics. The ODE for the SIS model without vital dynamics can be analytically solved to understand the disease dynamics. The ODE is as follows:



At equilibrium, solving:



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

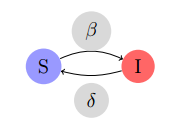


For disease to spread, . Therefore, when , the disease will spread and approach the second steady state; otherwise, it will eventually reach the disease-free state.

# Spreading Model in Arbitrary Networks

We consider a network comprising 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 changes according to a stochastic process parameterized by its infection rate , and curing rate .

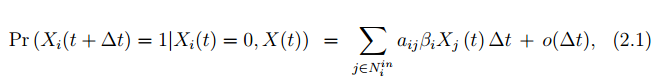
As part of our model, we assume the respective infection and curing rates and at the different agents can be tuned by injecting vaccine and treatment resources.



The state of node at time is a binary random variable . The state indicates that node is in the Susceptible state, which we denote as and the state indicates that node is in the Infected state, denoted .

Let the vector of states be defined as . Next, we describe the state evolution of each node

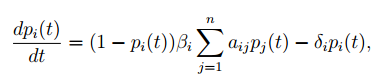
* Suppose node is susceptible to the infection at time , it can transition to infected state during a small time interval with a probability that depends on its infection rate , the strength of its incoming connections from its neighbors , as well as the states of its in-neighbors .



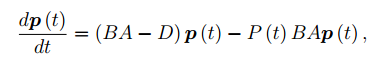
* If node is infected, its probability of transitioning to a susceptible state in the time interval depends on its recovery rate and is given by



Suppose we denote the probability of infection at node as . Then, and taking into account the fact that , the Markov differential  
equation for state can be approximated by



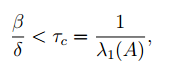
Which we can more compactly write as



where , is a diagonal matrix comprising the infection rates across the nodes, is a diagonal matrix comprising the curing rates across the nodes, and .

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*



where and are respectively the infection and curing rates, and 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 , , . Then, if the eigenvalue with largest real part of satisfies

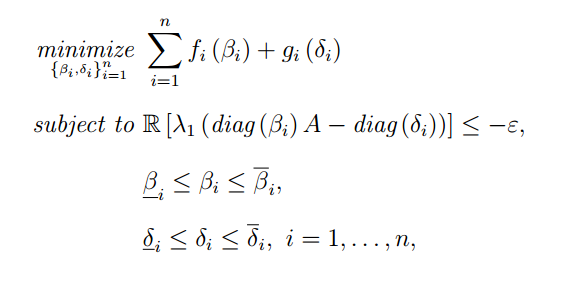


for some , the disease-free equilibrium is globally exponentially stable,i.e., , for some .

# Epidemic Control Problem

An assumption in our model and problem set up is that the infection rate and recovery rate for node can be adjusted at a cost. We assume preventive resources (vaccines) at node reduces its infection rate within feasible intervals, with associated cost ; and that treatment resources (antidotes) at node ups its recovery rate within feasible intervals and accrues a cost . Our choices for the associated cost functions is such that is monotonically decreasing w.r.t. and is monotonically increasing w.r.t. .

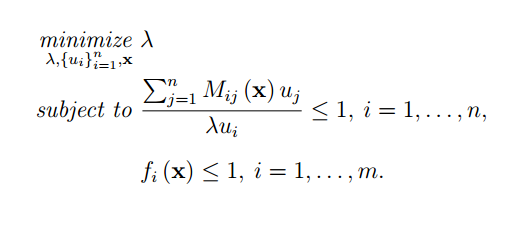
**Problem 1.** Given a vaccination cost function , for within a feasible interval, and treatment cost function , for within a feasibleinterval, determine the optimal allocation of vaccines and treatmentresources to control the spread of an epidemic outbreak with an asymptotic exponential decaying rate for a minimum cost. Mathematically, this problem can beformulated as follows:



Where is the vaccination cost incurred at node , is the treatment costat node , is the adjacency matrix associated with the network.

**Lemma 1.** (Perron-Frobenius) Let be a nonnegative, irreducible matrix. Then,the following statements about its spectral radius, , hold:1. is a simple eigenvalue of ,2. , for some , and3. .

**Proposition 3.** Consider the nonnegative, irreducible matrix with entries being either or posynomials with domain , where isdefined as , being posynomials. Then, we canminimize by solving the following GP:



# 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 . To overcome this challenge for **Problem 1**, an equivalent reformulation is derived.

**Theorem 1.** Given a strongly connected graph with adjacency matrix, posynomial cost functions ; bounds on the infection and recovery rates and , , and adesired exponential decay rate . Then, the optimal investment on vaccines andantidotes for node to solve **Problem 1** are and , where and are the optimal solution for and in the following GP:

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 representing a local copy of the global variable at each node .

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

So the Lagrangian function of problem above is:

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:

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**

1. Initializing the parameters

At iteration k:

1. Each agent optimize its own problem and then sends the related to the coordinator.

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

1. In this step coordinator updates and Lagrange multiplier using the information collected from the agents.
2. Go to b until

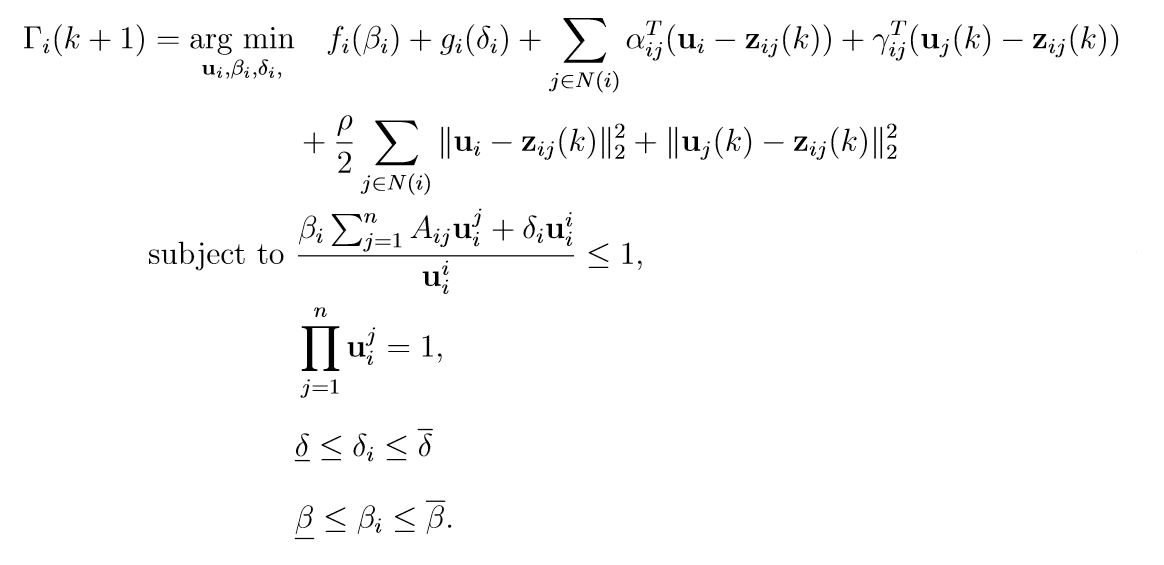
# Distributed ADMM Algorithm

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.

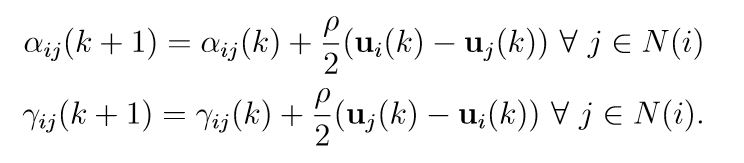
So given the explanation above the distributed ADMM algorithm for each node can be expressed as:

The constraints and imply that for all pairs of agents that form an edge, the feasible set is such that . Assuming a strongly connected contact network, these local consensus constraints imply that feasible solutions must satisfy for all, not necessarily neighboring, pairs of agents and .

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



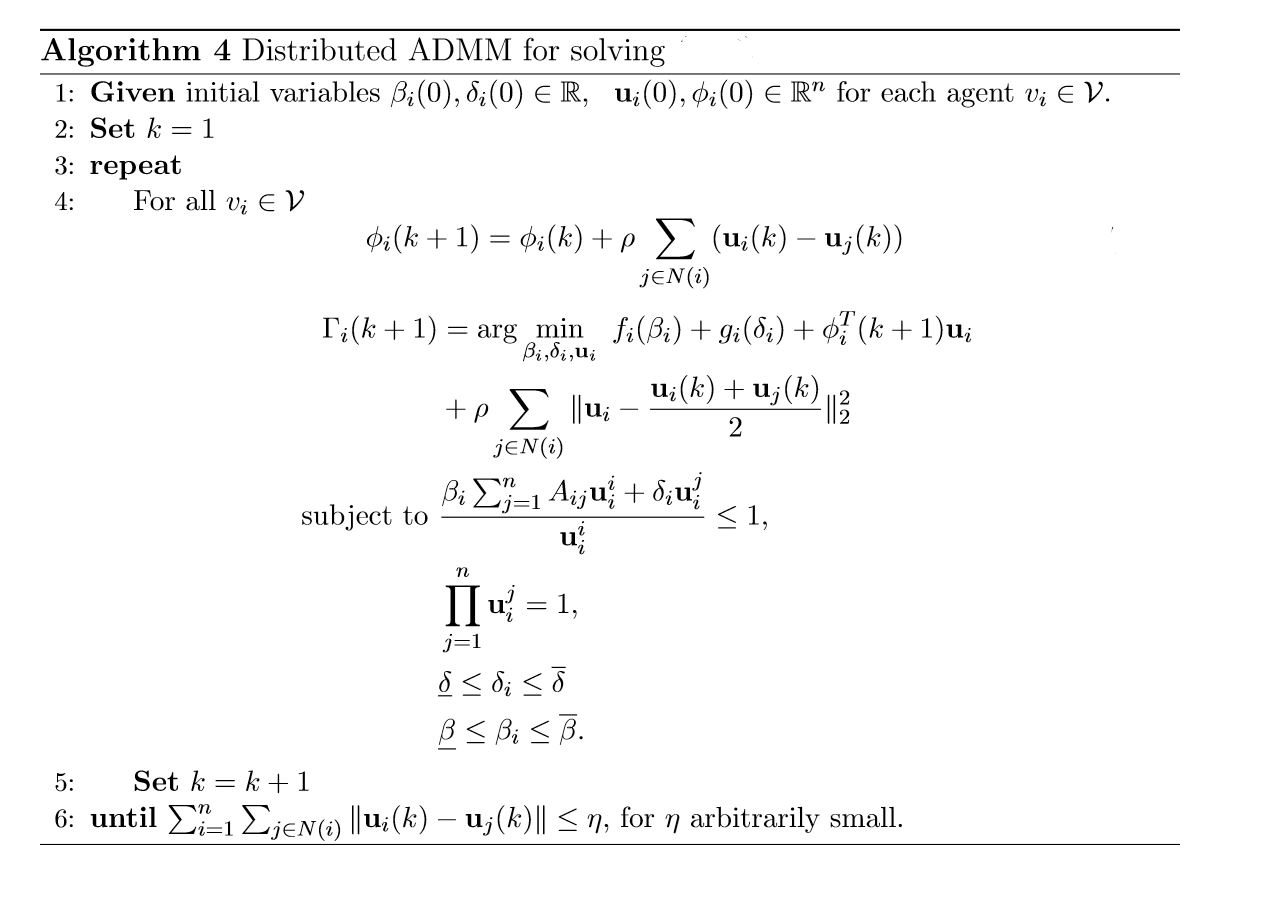
And the dual variables updates are:



Now the paper has introduced a variable

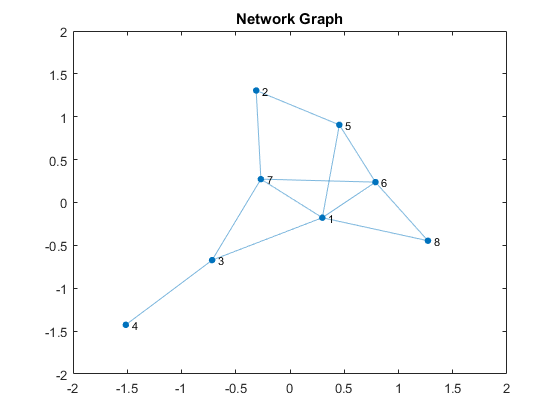
Therefore by using the new introduced variable and augmented Lagrangian the Distributed ADMM algorithm can be expressed:

**Algorithm 2. Distributed Algorithm**



# 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))))

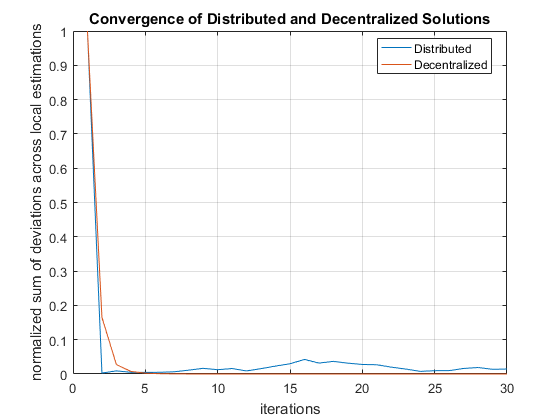
ans = 0.7200  
ans = -4.4176e-16  
ans = 0.1600  
ans = -0.5600

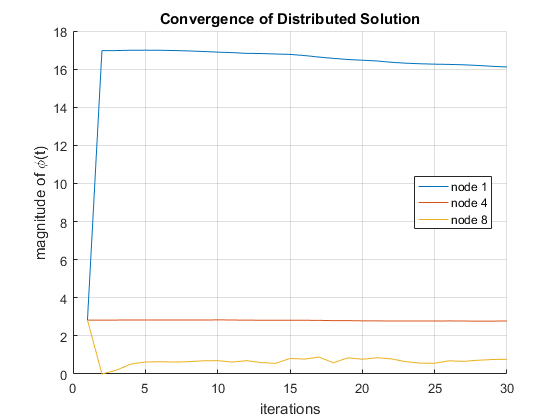
## Final Results

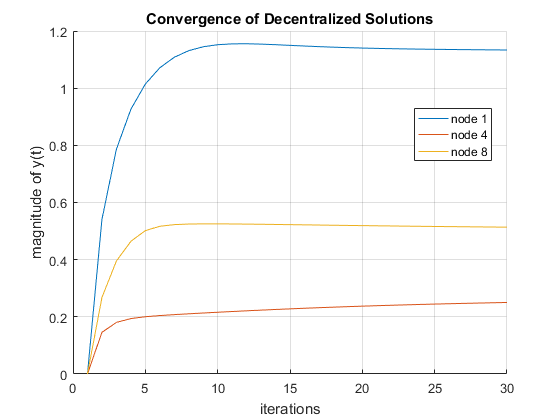
For above network we show convergence rate and analyze three special nodes that one of them has least degree and one of them has most degree and another one is a typical node. As you can see the node with highest degree is a central node so it has important role, so we should assign most vaccines to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

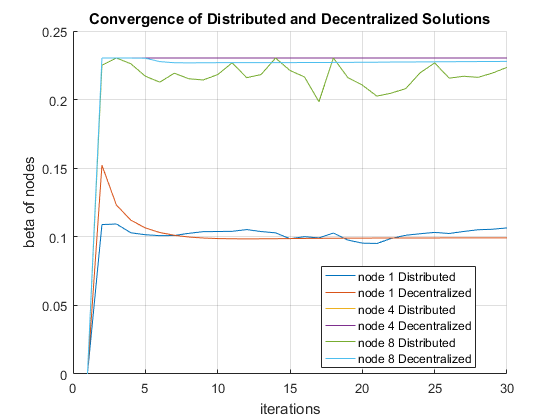
For comparison of Distributed algorithm and Decentralized we can say that Decentralized approach has better performance based on speed of convergence and accuracy of optimization but the main bottleneck of decentralized algorithm is the need of extra agent named ‘coordinator’, That forces extra costs.

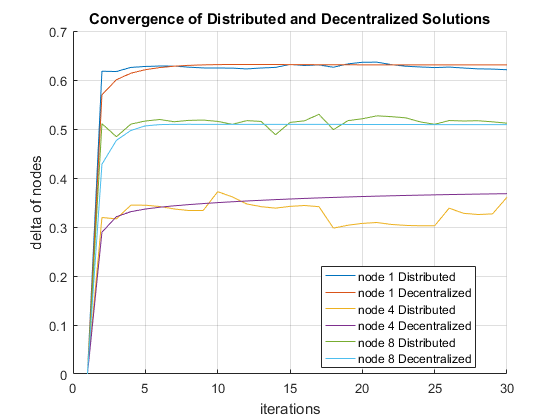
Epsilon is 0.1000  
DFE parameter of distributed solution that should be less than minus epsilon is: -0.0969   
DFE parameter of decentralized solution that should be less than minus epsilon is: -0.1001











# Reference

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