Chapter 1

1.1 Inspiration

While working in the math department I was assigned to help Independent Study customers resolve their issues. Independent Study is loosely organized into a network of interconnected organizations. Many of the sub-units know very little about other parts of the organization. I spent a lot of time on the phone trying to find a person who could answer my questions. Needless to say, while suffering through hours on hold I thought a lot about how much time I could save if I knew the right people to call. This experience led me to consider ways to fix the problem, and speculate that other companies probably have the same issue.

1.2 Network Organizations

Below are some quotes taken from *Causes of Failure in Network Organizations* by Raymond E. Miles and Charles C. Snow:

Properties of Network Organizations

Network organizations are different from previous organizations in several respects. First over the past several decades firms using older structures preferred to hold in house... all the assets required to produce a given good or service. In contrast many networks use the collective assets of many firms located at various points along the value chain...

Second, networks rely more on market mechanisms than administrative processes to manage resource flows...

The various components of the network are recognize their interdependence and are willing to share information and cooperate with each other... (Page 55)

Failures in Network Organizations

The most common threat to the effectiveness of a stable network is an extension that demands the complete utilization of the supplier's or distributor's assets for the benefit of the core firm... Unless suppliers sell to other firms the price and quality of their output is not subject to market test.

1.3 Model

Model I Assume a firm has a network structure with p units or nodes and that the firm is capable of completing n tasks.

Let $\mathbf{x}(k) \in \mathbb{R}^{pn}$ and let $\mathbf{u}(k)$ be a vector of length pn where each $u(k)_{ij} \in \{0, 1\}$. We will index them as follows:

$$\mathbf{x}(k) = \begin{bmatrix} x(k)_{11} \\ x(k)_{12} \\ \vdots \\ x(k)_{1n} \\ x(k)_{21} \\ \vdots \\ x(k)_{2n} \\ \vdots \\ x(k)_{pn} \end{bmatrix} \quad \mathbf{u}(k) = \begin{bmatrix} u(k)_{11} \\ u(k)_{12} \\ \vdots \\ u(k)_{1n} \\ u(k)_{21} \\ \vdots \\ u(k)_{2n} \\ \vdots \\ u(k)_{pn} \end{bmatrix}$$

Let $x(k)_{ij}$ represent the experience the ith unit has at completing task j at time k and let each $u(k)_{ij}$ be a binary variable representing if task j is assigned to node i at time k where $i \in 1, 2, \cdots, p$ and $j \in 1, 2, \cdots, n$. For example, a value of 1 at $u(5)_{23}$ represents the assignment of task 3 to unit 2 at time $k = 5$.

We will model changes in the states and measure efficiency by creating a dynamic model:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k)$$

$$y(k) = \mathbf{c}^{\mathrm{T}}\mathbf{x}(k)$$

To proceed we make an assumption that certain tasks are delegated to certain units. The vector $\mathbf{c}$ represents this intended delegation structure of the firm. If node i is assigned to complete task j C_{ij} will equal zero. Otherwise, it will equal one.

The scalar $y(k) \in \mathbb{R}$ is an output measurement of unit specialization. It represents the dot product of $\mathbf{c}$ with the experience vector, $\mathbf{x}(k)$. This way only the experience of unit i at completing tasks that it is **not** assigned to do will be factored into the specialization measurement. A higher specialization value represents an inability of units to pass tasks to the appropriate unit and therefore a greater the degree of inefficiency in the firm. If $y(k) = 0$ for all k, the firm accomplishes it's intended delegation structure perfectly.

In this model B is matrix of zeros and ones representing the ability of the firm to transfer tasks to a different unit.

The matrix A represents a "forgetting rate". We assume that each unit forgets how to do tasks they were not intended to do at a rate of $\alpha \in (0, 1)$. Setting $\mathrm{A} = \mathrm{I} - (1 - \alpha)diag(\mathbf{c})$ achieves the desired result.

Example Assume a firm offers three services and has two units. Then $\mathbf{x}(k)$ and $\mathbf{u}(k)$ have length 6.
If the first unit is assigned to preform only task one and the second unit is assigned to preform only tasks two and three, $\mathbf{c}^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$.

Assume that unit one always transfers tasks two and three to unit two, but unit two does not transfer task one to unit one. This gives a B matrix with the following structure:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

If at time $k = 1$ each unit has one experience completing their assigned tasks, we have $\mathbf{x}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^{\mathrm{T}}$.

If at $k = 1$ unit one receives task three and unit two receives task one, we have and input given by $\mathbf{u}(1) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$. We plug these values into the system as follows:

$$\mathbf{x}(2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

This gives an output of

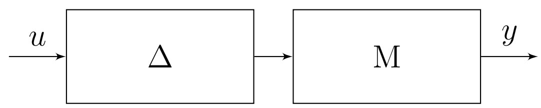
$$\mathbf{y}(2) = \mathrm{C}\mathbf{x}(2) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

demonstrating the degree of inefficiency introduced by unit two's inability to transfer task one to unit one.

Analysis One issue to consider is scalability. Though matrix and vector size increase relative to $\mathrm{O}(pn)$, the model should still scale well due to the use of sparse, time invariant matrices.

A second issue to work through is answering the question of how to set α. While this depends on the input stream, a high α could easily lead to an unbounded $y(k)$.

Third, it is important to consider where input comes from. This model is best suited to explaining how a firm handles a perturbed input stream. Ideally, all requests for a certain service would be directed to the appropriate unit in the firm, but in reality this in not the case. (i.e. Customers calling IT when they should really be calling Customer Service) This creates a system diagram as illustrated below, where M is the controlled system.



Finally, there is some interesting work to be done investigating the B matrix. The model currently simulates each unit knowing where to pass a task or doing the task because they don't know where to send it. It fails to address a network of passing a task between multiple units.

Furthermore, restrictions could be imposed on the B matrix, such as allowing each unit to have a limited number of connections to other units. It could be interesting to determine which connections create the most efficiency in this limited setting.