Data Backup Network Formation with Heterogeneous Agents

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
CS 493 B.Tech Project

By

Harshit Jain
ID No. 140001013

Under the supervision of:

Dr. Kapil Ahuja

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2017

Declaration of Authorship

I, Harshit Jain , declare that this Undergraduate Thesis titled, 'Data Backup Network Formation with Heterogeneous Agents' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for BTP Project at IIT Indore.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

Certificate

This is to certify that the thesis entitled, "*Data Backup Network Formation with Heterogeneous Agents*" and submitted by Harshit Jain ID No. 140001013 in partial fulfillment of the requirements of CS 493 B.Tech Project embodies the work done by him under my supervision.

Supervisor
Dr. Kapil Ahuja
Associate Professor,
Indian Institute of Technology Indore
Date:

INDIAN INSTITUTE OF TECHNOLOGY INDORE

Abstract

Bachelor of Technology

Data Backup Network Formation with Heterogeneous Agents

Social Storage systems [11, 7, 10] are becoming increasingly popular compared to the existing data backup systems like local, centralized and P2P systems. A symmetric social storage model and all its aspects like the utility of each agent, bilateral stability and efficiency have proposed in [9]. We have included heterogeneity in this model by using the concept of the Social Range Matrix proposed in [8]. Now, each agents is concerned about his perceived utility which is a linear combination of other agent's utilities, with the values of the coefficients denotes whether the pair are friends, enemies or they don't care about each other. First, we modify the utility function and then derive the conditions when two agents may add or delete a link. We provide an algorithm which checks if a bilaterally stable network is possible or not and if possible, then arrives at such a network. We then take some special social range matrices and prove that under what conditions on network parameters, a bilaterally stable graph is unique, like a complete or a null graph.

Acknowledgements

I would like to thank my B.Tech Project supervisor **Dr. Kapil Ahuja** for his constant support in structuring the project and his valuable feedback throughout the course of this project. He gave me an opportunity to discover and work in such an interesting domain. His guidance proved really valuable in all the difficulties I faced in the course of this project.

I am really grateful to **Mr. Pramod Mane** who also provided valuable guidance and helped with the problems while working with the research. He provided the initial pathway for stating the project in right manner and provided useful directions to proceed along whenever necessary.

I am also thankful to my family members, friends and colleagues who were a constant source of motivation. I am really grateful to Dept. of Computer Science & Engineering, IIT Indore for providing with the necessary hardware utilities to complete the project. I offer sincere thanks to everyone who else who knowingly or unknowingly helped me complete this project.

Contents

This work is dedicated to my parents and friends.

Chapter 1

Literature Survey

1.1 Domain

In this digital era, where personal data size is growing exponentially, data backup is not a new need. Data stored on an agent's local machine is prone to loss due to diskfailure, malware, etc. Local backup, centralized on-line backup and decentralized (Peer-to-Peer) backup are some strategies available to agents. Each has its own merits and demerits. For example, maintaining data backup on a local external hard disk on a regular basis is cumbersome. As far as on-line backup systems are concerned, on the one hand, centralized on-line backup is not cost efficient, especially when the amount of data required to be backed up is huge. On the other hand, although Peer-to-Peer (P2P) backup systems are cost efficient, they require dealing with several issues like data availability, reliability and security.

In recent years, to cope up with the above issues in P2P storage systems, researchers have been focusing on on-line social network relationships. It is believed that social ties between agents (or agents) will help to build backup systems that overcome aforementioned issues. This trend that takes real world social relationships (encoded in an on-line social network) into account for constructing a data backup system is emerging as a special case of P2P backup system, and tagged as Social Storage or Friend-to-Friend (F2F) Storage1(Friendstore, FriendBox, BackupBuddy2 are a few examples).

Existing research on social storage is moving in two directions. One research trend has been focusing on various approaches to build the system, dealing with providing ways for agents to select their data-backup-partners explicitly. The other direction has been focusing on studying Quality of Service (QoS) related issues such as data availability, reliability, the cost associated with communication, data maintenance, data placement or scheduling polices, by taking online social relationships into account.

1.2 Focus

Initial work in this field has primarily focused on developing techniques to exogenously build social storage systems, and performing Quality of Services (QoS) analysis in terms of data reliability and availability in these systems. A most recent study in [9] focuses on explicit data backup partner selection, where agents themselves select their partners. This selection is studied in a strategic setting, and eventually builds a social storage network. They model a utility function, which reveals the benefit an individual receives in a social storage network. Further, they analyze the network by using bilateral stability as a solution concept, where no pair of agents add or delete a link without their mutual consent.

There are several advantages of this approach. First, this approach makes it possible to incorporate user's strategic behavior (in terms of with whom user wants to form social connections and with whom it does not). Second, this approach helps us to predict the following: which network is likely to emerge; which one is stable (in which no individual has incentives to alter the structure of the network either by forming new or deleting existing social connections); and which one is the best (contended and/ or efficient) from all the participants point of view.

1.3 Motivation

Many distributed systems have an open clientele and can only be understood when taking into account socio-economic aspects. A classic approach to gain insights into these systems is to assume that all agents are selfish and seek to maximize their utility. Often, the simplifying assumption is made that all agents have the same utility function. However, distributed systems are often "socially heterogeneous" whose participants run different clients and protocols, some of which may be selfish while others may even try to harm the system. Moreover, in a social network setting where members are not anonymous, some agents may be friends and dislike certain other agents. Thus, the state and evolution of the system depends on a plethora of different utility functions. Clearly, the more complex and heterogeneous the behavior of the different network participants, the more difficult it becomes to understand (or even predict) certain outcomes.

The above strategic setting does not consider heterogeneous behavior (e.g., selfless, selfish, etc.) in the network formation. In our thesis, we focus on this aspect. This way, we make the model close to a real-world scenario. Although, doing so makes it challenging to deal with the model and as well as predict its outcome.

To achieve the above, we incorporate the concept of Social Range Matrix introduced in [8] while exploring different social relationships between agents. We modify the utility of agents in the social storage model discussed in [9]. Further, we revisit the results regarding bilateral stability of such a model.

Chapter 2

Model

2.1 Introduction

The model is composed of four components. First, the symmetric social storage network. Second, the utility (as derived in [9]) of each agent depends on the network structure. Third, the social range matrix (SRM), which captures a social relationship between these agents. Fourth, the perceived utility obtained by modifying the above utility.

Table 2.1: Notation Summary

P2P	peer to peer
g	social storage network
N	number of players
c	cost incurred by an agent to maintain a link
β_i	worth (value) that agent i has for its data
λ	probability of failure of a disk
$\eta_i(g)$	neighborhood size of agent i in g
f_{ij}	Social ties between player i and j
F	Social range matrix

2.2 Symmetric Social Storage Network

A symmetric social storage network g, is a data backup network consists of N number of agents and a set of links connecting these agents. A link $\langle ij \rangle \in g$ represents that agent i and j are data backup partners, who store their data on each other's shared storage space. In g, pairs of agents share an equal amount of storage space.

The utility function in [9] reveals the cost and benefit that each agent i receives in g and derived as follows.

$$u_i(g) = \beta * (1 - \lambda^{n_i(g)}) - cn_i(g)$$

The utility function $u_i(g)$ consist following parameters. The neighbourhood size of agent i that is represented by $n_i(g)$. The benefit that is associated with data is represented by β. The cost c that agent i incurs to maintain its neighbours. The probability of disk failure is denoted by λ. Note that, all parameters c, β, and λ lies between 0 and 1. The utility is a combination of two objectives for each agent i, the first one is to minimize the total cost of the links which is $cn_i(g)$, and the second is to maximize the expected data backup which is $\beta_i(1 - \lambda^{n_i(g)})$. Note that, now onward, we use n_i to represent neighbourhood size of agent i in g.

2.3 Social Range Matrix

In our model, we consider three type of agents. First, where an agent helps other to maximize other's utility, second, an agent aims to decrease the other's utility, and third, where an agent does not care about other's utility. We conceived the above behaviour as follows. If a pair of agents wants to maximize each other's utility then they are friends, if they want to minimize each other's utility then they are an enemy of each other otherwise they are neutral. We represent the above kind of social relationships between pairs of agents in a social range matrix F, where each element f_{ij} denotes the social relationship between i and j.

$$f_{ij} \begin{cases} = 0 & i \text{ and } j \text{ don't care about each other} \\ > 0 & i \text{ and } j \text{ are friends to each other} \\ < 0 & i \text{ and } j \text{ are enemies to each other} \end{cases}$$

$$f_{ii} > 0$$

In our model, for the sake of simplicity we will use only three values for f_{ij} which are 0, 1, -1 indicating don't care, friends and enemies respectively. We consider that agents give more importance to other agents utilities than their own utility i.e., $f_{ii} < |f_{ij}|$ for all j.

Many such interesting matrices are possible, for example

1. A matrix with all 1's means that all pair of people are friends

2. A matrix with all -1's means that all pair of people are enemies

3. A matrix with all zeroes except the diagonal elements means weak social ties.

$$\begin{bmatrix} \epsilon & 0 & 0 & 0 & 0 \\ 0 & \epsilon & 0 & 0 & 0 \\ 0 & 0 & \epsilon & 0 & 0 \\ 0 & 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & 0 & \epsilon \end{bmatrix}$$

4. A malicious player, whose aim is to hamper the system performance has a perceived utility that consists of the negative of utilities of other players.

$$\begin{bmatrix} \epsilon & 1 & 1 & 1 & -1 \\ 1 & \epsilon & 1 & 1 & -1 \\ 1 & 1 & \epsilon & 1 & -1 \\ 1 & 1 & 1 & \epsilon & -1 \\ -1 & -1 & -1 & -1 & \epsilon \end{bmatrix}$$

2.4 Perceived Utility

Now, agents have different social relationships with others, and hence, their utility not only depend on the structure of the network but also their social relationships with others. We define the new utility as perceived utility of agent i in g as follows.

$$\widetilde{u_i}(g) = \sum_j f_{ij} u_j(g) \tag{2.1}$$

where $j \in$ the set of all agents including i.

In this setting, each agent's objective is to maximize its perceived utility (which takes care of utilities of other agents). Thus, the optimisation problem is

$$max(\widetilde{u_i}(g))$$

2.5 Network Formation Game

At any given point in time, each agent plays a dual role: that of a data owner who wants to back up its data, and that of a backup partner who provides storage for each of its backup partners. Pairs of agents may add a new link (or continue to maintain the existing link) or delete the existing link (or continue to remain without a direct link). Note that, an agent neither adds a new link nor deletes an existing link without consent of the agent with whom it wants to perform an agreement or is involved in an agreement.

In the social storage context, mutual consent is a must for deleting links too. This assumption is practical but has not been focused upon in the network formation literature.

The structure of the network is defined by actions of the agents. Firstly, the network is updated when two agents i and j add a new link $< ij >$, and we denote this by g+ $< ij >$. Secondly, the network is updated when a pair agents i and j delete an existing link $< ij >$, and we denote this by g$-$ $< ij >$. As agents explicitly decide with whom they want to perform backup partnerships and with whom they do not, this is a process of endogenous network formation (or partner selection).

There is need for a solution concept which is suitable for characterizing the storage network formation game. A strategic network formation game (NFG) is described as below. NFG consists of a set of agents A $=1, 2, \cdots, N$ who represent nodes in the network g. In this setting, pairs of agents may form new links thereby increasing their expected value of backup data, by incurring higher costs to maintain links. Pairs of agents may also delete existing links, thereby reducing the costs incurred, but reducing the probability of retrieving the data too. The shape of the network is not only defined by each agent's cost and benefit trade off, but also by limitation of resources available with the agents.

2.6 Conditions for Link Addition

Link addition between agent i and j occurs only when the perceived utilities of both the agents increases, i.e.,

$$\widetilde{u_i}(g + \langle ij \rangle) > \widetilde{u_i}(g) \tag{2.2}$$

$$and$$

$$\widetilde{u_j}(g + \langle ij \rangle) > \widetilde{u_j}(g) \tag{2.3}$$

For all agents k except i and j, the neighbourhood size (n_k) remains constant so their utility is the same and cancels out. simplifying (2.2) and (2.3) with (2.1), we get that link addition happens when

$$f_{ii}[\beta(\lambda^{n_i} - \lambda^{n_i+1}) - c] + f_{ij}[\beta(\lambda^{n_j} - \lambda^{n_j+1}) - c] > 0$$

and

$$f_{ii}[\beta(\lambda^{n_j} - \lambda^{n_j+1}) - c] + f_{ij}[\beta(\lambda^{n_i} - \lambda^{n_i+1}) - c] > 0$$

Simplifying more we get,

$$f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} > \frac{(f_{ii} + f_{ij}) * c}{(1 - \lambda) * \beta} \tag{2.4}$$

$$and$$

$$f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} > \frac{(f_{ii} + f_{ij}) * c}{(1 - \lambda) * \beta} \tag{2.5}$$

2.7 Conditions for Link Deletion

Link deletion happens when

$$\widetilde{u_i}(g - \langle ij \rangle) > \widetilde{u_i}(g)$$

$$and$$

$$\widetilde{u_j}(g - \langle ij \rangle) > \widetilde{u_j}(g)$$

Thus, equivalent to (4) and (5) we get the following :

$$f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} < \frac{(f_{ii} + f_{ij}) * c * \lambda}{(1 - \lambda) * \beta} \tag{2.6}$$

$$and$$

$$f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} < \frac{(f_{ii} + f_{ij}) * c * \lambda}{(1 - \lambda) * \beta} \tag{2.7}$$

Next, we derive the conditions on neigbourhood size, which if satisfied would ensure link addition.

2.8 Sufficiency conditions for Link Addition

2.8.1 Case 1

$$If\ n_i > n_j$$
$$\lambda^{n_i} < \lambda^{n_j} \qquad \because 0 < \lambda < 1$$
$$f_{ii}\lambda^{n_i} < f_{ii}\lambda^{n_j} \qquad \because f_{ii} > 0$$

2.8.1.1 If $f_{ij} > 0$

$$f_{ij} > 0$$
$$i.e., f_{ij}\lambda^{n_i} < f_{ij}\lambda^{n_j}$$
$$f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_i} < f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} < f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_j}$$
$$(f_{ii} + f_{ij})\lambda^{n_i} < f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} < (f_{ii} + f_{ij})\lambda^{n_j} \tag{2.8}$$
$$and$$
$$f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_i} < f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} < f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_j}$$
$$(f_{ii} + f_{ij})\lambda^{n_i} < f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} < (f_{ii} + f_{ij})\lambda^{n_j} \tag{2.9}$$

Using (2.4)−(2.5) in (2.8)−(2.9) we get sufficiency condition for link addition as below.

$$If(f_{ii} + f_{ij})\lambda^{n_i} > \frac{(f_{ii} + f_{ij}) * c}{(1 - \lambda) * \beta} \tag{2.10}$$

On Solving (2.10) we get

$$n_i < \frac{|\ln(\frac{c}{(1-\lambda)*\beta})|}{|\ln\lambda|}$$

2.8.1.2 If $f_{ij} < 0$

$$f_{ij} < 0$$

$$i.e., f_{ij}\lambda^{n_i} > f_{ij}\lambda^{n_j}$$

$$f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} > f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j}$$

So

$$(2.8) \implies (2.9)$$

$$2f_{ij}\lambda^{n_j} < f_{ij}\lambda^{n_i}+f_{ij}\lambda^{n_j} < f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j}$$

$$2f_{ij}\lambda^{n_j} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$$

$$\lambda^{n_j} > \frac{(f_{ii}+f_{ij})*c}{2f_{ij}(1-\lambda)*\beta}$$

is the sufficiency condition.

2.8.2 Case 2

If $n_i < n_j$ As (2.4) and (2.5) are symmetric interchange i and j

2.8.3 Case 3

$$n_i = n_j$$

$$\lambda^{n_i} = \lambda^{n_j}$$

$$\frac{(f_{ii} + f_{ij}) * c}{(1 - \lambda) * \beta} > (f_{ii} + f_{ij})\lambda^{n_i}$$

both sufficiency and necessary condition is

$$\lambda^{n_i} > \frac{c}{(1 - \lambda) * \beta} \qquad if\ f_{ij} > 0$$

$$\lambda^{n_i} < \frac{c}{(1 - \lambda) * \beta} \qquad if\ f_{ij} < 0$$

We have summarized our results in Table 1

value of f_2	sufficient conditions
$f_2 > 0$	$t_1 = \max(n_i , n_j)$ $t_2 = \min(n_i , n_j)$ $1) t_1 \geq t_2 \ \&\ t_1 < \frac{\lvert \ln(\frac{c}{(1-\lambda)*\beta}) \rvert}{\lvert \ln \lambda \rvert}$ $2) t_1 \geq t_2 \ \&\ t_1 < \frac{\left\lvert \ln\left(\frac{(\frac{f_1+f_2}{f_1})c}{2(1-\lambda)*\beta}\right) \right\rvert}{\lvert \ln \lambda \rvert}$
$f_2 < 0$	$t_1 \geq t_2 \ \&\ t_2 > \frac{\left\lvert \ln\left(\frac{(\frac{f_1+f_2}{f_2})c}{2(1-\lambda)*\beta}\right) \right\rvert}{\lvert \ln \lambda \rvert}$

Similarly we can derive sufficiency conditions and for link deletion also.

Chapter 3

Results and Lemmas

3.1 Bilaterally Stable Networks

A social storage network g is bilaterally stable if and only if

1. for all $< ij > \in$ g, if $u_i(g- < ij >) > u_i(g)$, then $u_j(g- < ij >) < u_j(g)$, and
2. for all $< ij > \notin$ g, if $u_i(g- < ij >) > u_i(g)$, then $u_j(g- < ij >) < u_j(g)$.

This is a network stability concept, whose first part states that no pair of agents with a link between them, wants to delete the link, and the second part states that no pair of agents has an incentive to add a new link. Note that neither link formation (addition) nor link deletion can happen without mutual consent.

In our domain of social storage we focus on the requirement of bilateral consent while deleting a link as well. For instance, let agents i and j be backup partners. That is, i provides its storage space to j for the purpose of storing j0s data, and vice versa. Now, let us assume that breaking a backup partnership without mutual consent is allowed. If agent i breaks the partnership without consent of j, then there is a threat that j will lose its data which is stored on i′s storage resource.

Hence, backup partnerships in social storage networks have to be viewed as mutual contracts which cannot be broken unilaterally.

3.2 Stability Point in a Symmetric Social Storage Network

Mane et al have derived a stability point (a neighborhood size), η where no agent wants to add or delete a link if they achieve this neighborhood size, i.e., their utility is maximized at this size. Stability point is defined as

$$\eta = \left\lceil \frac{\left|\ln(\frac{c}{\beta(1-\lambda)})\right|}{|\ln\lambda|} \right\rceil$$

As their network is symmetric, they were able to do so. But we have introduced heterogeneity in our model, so there doesn't exist such a point, but possibly a bilaterally stable graph.

3.3 Pseudo Code to arrive at a Bilaterally Stable Network.

Algorithm 1 lists the steps for reaching at a bilaterally stable network. That is, when no agent has any incentive to add or delete a link.

3.4 Case Study

Let us consider there are five agents (a, b, c, d, e). The social relationship between these agents is captured in the social range matrix F (see Table 3.1). For instance, agent a is a friend of b, but an enemy of c, and vice versa.

Let us assume that $c = 0.01, \lambda = 0.2, \epsilon = 0.1, \beta = 0.1$, and given F. Initially all agents are isolated. We follow the procedure described in Algorithm 1, and we obtain the bilaterally stable network g' shown in Fig. 3.1. With the same set of parameters and F, by taking the starting

Algorithm 1: Pseudo code to arrive at a bilaterally stable network.

Input : $c, \lambda, \beta, F, starting\ network, flag = 1$

Comment : i and j are agents, flag = 1 means network is not bilaterally stable

```
while flag == 1 do
    flag = 0
    for i = 1 to n do
        for j = 1 to n do
            if i ≠ j then
                if link is absent between < i, j > then
                    Check link addition conditions (4) and (5) for i, j and add link if they are true.
                    flag = 1
                end
                if link is present between < i, j > then
                    Check link deletion conditions (6) and (7) for i, j and delete link if they are true.
                    flag = 1
                end
            end
        end
    end
end
```

Table 3.1: Social Range Matrix F

	a	b	c	d	e
a	ϵ	1	-1	1	-1
b	1	ϵ	1	-1	-1
c	-1	1	ϵ	-1	1
d	1	-1	-1	ϵ	1
e	-1	-1	1	1	ϵ

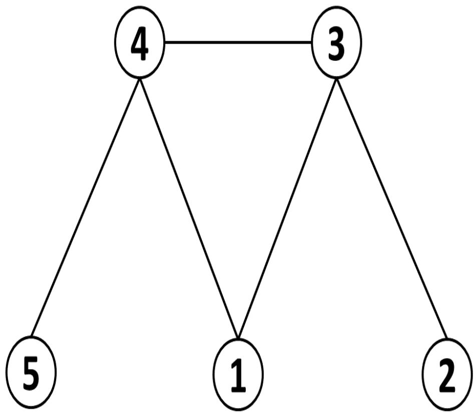

Figure 3.1: Bilaterally Stable Social Storage Network g'.

network as a random non null network, we obtain bilaterally stable network different than in Fig. 3.1. This implies that bilaterally stable networks are not necessarily unique.

By looking at the the ratio of $\frac{c}{\beta}$, where, $0 < c, \beta < 1$ and $\lambda = 0.2$, and the above F, we found that, for $0.044 < \frac{c}{\beta} < 0.089$, the procedure runs into infinite loop implying no bilaterally stable network is possible. That is, there is a no stable network we have for the specified value range. Interestingly, for all other values, we found at least one stable network.

But, if $c > \beta$ and $\lambda = 0.2$, and the above F, then we have a unique bilateral stable network in which all pair agents are enemies.

3.5 Unique Bilaterally Stable Networks

Here, we provide some general results regarding unique bilaterally stable networks.

Lemma 1. If $\frac{c}{\beta} > (1-\lambda)$ and if all agents are friends of each other (i.e., $f_{ij} = 1$) then empty network is the unique bilaterally stable network.

Proof. To prove that the stable graph is a null graph we have to prove 2 things.

1. No pair of agents form a link.
2. If a link is present between 2 agents both want to delete it.

1. Conditions for link addition.

   (a) $f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

   (b) $f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

   If max of $(L.H.S(a)\ and\ L.H.S(b))$ <R.H.S, then these equations are always false.

   Max(L.H.S (a)) is when $n_i = 0$ & $n_j = 0$ as if $n_i\ or\ n_j$ increase the value of L.H.S decreases.

   Similarly for (L.H.S (b))

   So $f_{ii} + f_{ij} < \frac{(f_{ii}+f_{ij})*c*(1-\lambda)}{\beta}$ ie.

   $\frac{c}{\beta} > (1-\lambda)$

2. Conditions for link deletion.

(a) $f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} < \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

(b) $f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} < \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

If max of $(L.H.S(a)\ and\ L.H.S(b))$ <R.H.S, then these equations are always true

For link deletion to be true

$n_i \geq 1$ & $n_j \geq 1$

So (L.H.S (a)) and (L.H.S (B)) are minimum when $n_i = n_j = 1$

So $(f_{ii} + f_{ij})\lambda < \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

$\frac{c}{\beta} > (1-\lambda)$

□

Lemma 2. If $\frac{c}{\beta} > \frac{1}{1-f_{ii}}(1-\lambda)$ and if f_{ij} is either 1 or -1 then the bilaterally stable network will be the one where all pair of enemies form links.

Proof. To prove that the stable graph is a complete graph of enemies, we have to prove 3 things.

1. Any pair of enemies agents form a link.
2. If a link is present between 2 enemies, no agent want to delete it.
3. There is no link between any pair of friends.

1. Conditions for link addition.

(a) $f_{ii}\lambda^{n_i} + f_{ij}\lambda^{n_j} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

(b) $f_{ii}\lambda^{n_j} + f_{ij}\lambda^{n_i} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$

If min of $(L.H.S(a)\ and\ L.H.S(b))$ >R.H.S, then these equations are always true.

As $f_{ij} < 0$ & $f_{ii} > 0$, min(L.H.S (a)) is when $n_j = 0$ & n_i is very large

Similarly for (L.H.S (b))

$f_{ij} > \frac{(f_{ii}+f_{ij})c}{(1-\lambda)\beta}$

So $\frac{c}{\beta} > \frac{f_{ij}}{f_{ii}+f_{ij}}(1-\lambda)$

i.e., $\frac{c}{\beta} > \frac{-1}{f_{ii}-1}(1-\lambda)$

2. Similar to (1)

3. We observe that if the condition in Lemma 2 is true, then the condition in Lemma 1 will always be true.This implies that no pair of friends will form a link.

□

Lemma 3. If $\frac{c}{\beta} < (1-\lambda)\lambda^{n-2}$ and if all agents are friends of each other (i.e., $f_{ij} = 1$) then a complete network is the unique bilaterally stable network.

Proof. First, consider link addition conditions stated in equations (2.4) and (2.5). We can observe that, as n_i and n_j increases, L.H.S of (2.4) and (2.5) decreases. This implies that agents have an incentive to increase their neighbourhood size.

If the link addition conditions (4) and (5) are true for $n_i = n_j = N-2$ (both agents have neighbourhood size one less than the maximum possible size), then they will be true for all values of n_i, $n_j \leq$ N-2 (if the smaller value of L.H.S is greater than the R.H.S bigger values will be greater). We don't consider N-1 as at that neighbourhood size agents have no incentive to add a link. So,

$$f_{ii}\lambda^{N-2} + f_{ij}\lambda^{N-2} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$$

$$f_{ii}\lambda^{N-2} + f_{ij}\lambda^{N-2} > \frac{(f_{ii}+f_{ij})*c}{(1-\lambda)*\beta}$$

$$\frac{c}{\beta} < (1-\lambda)\lambda^{n-2}$$

Similarly we prove 2.

□

3.6 Efficient Networks

Definition 1. A social storage network, g is efficient with respect to a utility profile $(u_1, u_2, \ldots u_n)$ if

$$\sum_i \widetilde{u_i}(g) > \sum_i \widetilde{u_i}(g')$$

We study efficient networks when all pairs of agents are friends to each other.

Let the graph has N agents , then

$\sum \widetilde{u_i}(g) = f_{ii}(u_1 + u_2 + \ldots u_{n-1}) + (n-1)(u_1 + u_2 + \ldots u_{n-1})$

$= (f_{ii} + n - 1)(u_1 + u_2 + \ldots u_{n-1})$

$= (f_{ii} + n - 1) \sum_j u_j$

We have to maximize $\sum_j u_j$ which is the similar analysis as in the symmetric social storage i.e., when maximum no of agents achieve stability point η.

Also, if all agents have neighbourhood size η or N-1 agents have neighbourhood size η and the remaining one has a neighbourhood size of η-1 or η+1, then the graph is bilaterally stable according to our link addition and deletion conditions.

Chapter 4

Future Work

In this work, we have extended the social storage model proposed in [9] to include heterogeneous behavior (variety of social relationships). After that, we have analyzed this model using their solution concept of bilateral stability. The preliminary results here give tremendous insight into how a endogenously built social storage system would emerge.

Future work involves further analyzing the stability of such networks as well as studying contentment (when everyone has maximized their utility) and efficiency (where the total utility of the network is maximized) in this new heterogeneous agent scenario.

In all our discussions on bilaterally stable networks as well as efficient networks, we have assumed that any pair of agents can potentially form a link. Analyzing the stability of networks where agents do not necessarily trust all agents in the network is another exciting area.

We have used the notion of bilateral stability in our model. Looking at strong and coalition-proof Nash equilibria, strong pairwise stability, and farsighted equilibrium, are also future research directions.

Bibliography

[1] Venkatesh Bala and Sanjeev Goyal. "A Noncooperative Model of Network Formation". In: *Econometrica* 68.5 (2000), pp. 1181–1230.

[2] Mohamed Belhaj, Yann Bramoulle, and Frederic Deroian. *Network Games under Strategic Complementarities.* Tech. rep. 1225. Aix-Marseille School of Economics, Marseille, France, Oct. 2012.

[3] Paul Belleflamme and Francis Bloch. "Market sharing agreements and collusive Networks". In: *International Economic Review* 45.2 (2004), pp. 387–411.

[4] Francis Bloch and Matthew Jackson. "The formation of networks with transfers among players". In: *Journal of Economic Theory* 133.1 (2007), pp. 83–110.

[5] Bhaskar Dutta and Suresh Mutuswami. "Stable Networks". In: *Journal of Economic Theory* 76.2 (1997), pp. 322–344.

[6] Robert P. Gilles and Sudipta Sarangi. "Network formation under mutual consent and costly communication". In: *Mathematical Social Sciences* 60.3 (Nov. 2010), pp. 181–185.

[7] Raúl Gracia-Tinedo et al. "FriendBox: A Hybrid F2F Personal Storage Application". In: *12th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 2012, pp. 131–138.

[8] Petr Kuznetsov and Stefan Schmid. "Towards Network Games with Social Preferences". In: *17th International Colloquium on Structural Information and Communication Complexity*

(SIROCCO). Ed. by Boaz Patt-Shamir and Tınaz Ekim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 14–28. ISBN: 978-3-642-13284-1.

[9] Pramod Mane, Kapil Ahuja, and Nagarajan Krishnamurthy. "Unique Stability Point in Social Storage Networks". 2017. URL: `https://arxiv.org/abs/1603.07689`.

[10] R. Sharma et al. "An empirical study of availability in friend-to-friend storage systems". In: *IEEE International Conference on Peer-to-Peer Computing (P2P)*. 2011, pp. 348–351.

[11] Nguyen Tran, Frank Chiang, and Jinyang Li. "Efficient Cooperative Backup with Decentralized Trust Management". In: *ACM Transactions on Storage* 8.3 (2012), 8:1–8:25. ISSN: 1553-3077.