Statistical Ensembles of Financial Networks and the Dynamics of Cascading Defaults

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Author :	Camille de Valk
Student ID :	S1633597
Supervisor :	Dr. D. Garlaschelli
Company supervisor :	Dr. F. Jansen
2^{nd} corrector :	Dr. P.J.H. Denteneer

Leiden, The Netherlands, July 14, 2021

Statistical Ensembles of Financial Networks and the Dynamics of Cascading Defaults

Camille de Valk

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

July 14, 2021

Abstract

Statistical physicists are recently focusing on the network structure of financial systems in order to model cascading defaults in those systems. Active research is on the characteristics of financial networks. These networks are inferred from statistical ensembles constructed from partial information, due to the fact that researchers usually don't have full access to the entire network underlying the financial system, because of privacy reasons. Other research simulates the propagation of shocks such as defaults through financial systems and focuses on the characteristics of the network governing the dynamics. This research combines the two areas of research using a unique data set containing all transactions of commercial Dutch ING accounts from the year 2019. A state-of-the-art random network ensemble is tested in the research, whereafter networks are sampled from this ensemble to run the cascading defaults simulation on. Furthermore, the simulation of cascading defaults is improved by implementing non-trivial payment strategies, motivated by experiences from bankers dealing with defaulted companies. The simulation using the empirical network yields similar results to the networks sampled from the ensemble, indicating that the random network ensemble captures information governing the dynamics.

Contents

Chapter 1

Introduction

A financial system consists of companies having connections, e.g. input-output relationships, supply chain relationships or financial relationships. When these connections weaken or break, for example if a product or service essential for production is not delivered, the system is shocked. This shock can propagate through the system. Companies in the system that were not involved in the shock, can still be affected by the after-effects of the shock, because of the interconnectedness of the financial system. The shock could also affect the company that caused the shock: company A could fail to fulfil its obligations to company B, which thereby fails to deliver (products/services/money) to company C. Company C might have a connection to company A, creating a feedback loop on company A. In short, the very own structure of the financial system can create (systemic) risk for companies.

One of the possible shocks in a financial system is a *default*: a failure to fulfil a financial obligation. In 2001, Eisenberg and Noe [1] proposed a way of simulating such defaults in a financial system in order to measure the systemic risk. In the method, defaults can shock the system and cause a cascade of defaults, where one default follows the other because of a connection. Hazama and Uesugi [2] showed empirically that such default propagation exists in the Japanese interfirm trade credit network. They showed that the cascade of defaults can be modelled and used to detect prospective defaulters and even bankruptcies.

Both Eisenberg and Noe, and Hazama and Uesugi use a network representation of the financial system to simulate the cascade of defaults. The question remains what properties of the financial network cause and control the cascade of defaults. To try to answer this question, this research reaches to random network ensembles. Random network ensembles can

be used to observe higher-order behaviour (e.g. certain patterns) in real networks, as shown by Squartini and Garlaschelli [3, 4]. Examples of such higher-order patterns are the clustering coefficient, and the occurrence of triangular patterns [5]. The simulation of a cascade of defaults is also higher-order behaviour.

Random network models can be constructed in such a way that the realisations recreates certain constrained properties on average in the ensemble and that these realisations are otherwise maximally random [6]. For example, the number of neighbours (degrees) can be exactly reconstructed by the so-called Configuration Model [7], or both the flow of money (strength) and degrees can be constrained (on average) using the so-called Fitness-induced Configuration Model (FiCM) [6, 8]. When the higher-order behaviour of the real network and the behaviour of the random ensemble is not similar, one can conclude that the behaviour is not (only) the cause of the constrained properties.

The aim of this research is twofold. First, the simulation algorithm proposed by Eisenberg and Noe [1] is adjusted in such a way that defaulting companies employ non-trivial payment strategies and then run on the Dutch ING interfirm transaction network of 2019. In the new simulation, companies pay their obligations in such a way that they have fewer creditors, by paying their largest creditor last. This strategy is motivated by professional experiences of bankers in an arrears department, who hypothesised that this the actual behaviour of defaulting companies. The opposite strategy, paying the largest creditor first is also researched.

Second, to interpret the results of the cascading defaults simulation, a state of the art interfirm network reconstruction model called the Stripe Fitness model [9], is implemented on the 2019 Dutch ING interfirm transaction network. On top of the reconstruction of expected degrees and strengths, the Stripe Fitness model is developed such that the expected sector-sector connections are realised on average in the ensemble.

In chapter 2, the background of the research is discussed, further diving into the research of Eisenberg and Noe and on the random network models. Following that, chapter 3 discusses the data and preprocessing. The methods are explained in chapter 4 on page 27, whereafter the results are displayed and discussed in chapter 5. The discussion of the entire research is done in chapter 6. The thesis ends with a conclusion in chapter 7 looking back at the entire project.

Version of July 14, 2021– Created July 14, 2021 - 12:35

Note that this thesis is the physics part of a physics research and a business studies research project at ING. Some important parts of the project had more focus on business studies and are discussed more elaborate in that thesis [10]. There may also be some overlap in order to construct a complete thesis.
This thesis, the physics part, aims at testing the performance of reconstructed network ensembles built from statistical physics, both on the structure of the network ensembles and the dynamics, i.e. the cascading defaults simulation. This testing requires data of a real-world network, which is hard to find, because of confidentiality issues. The data which are used, are the transaction data at ING, made available for the internship. Therefore, the same data (and data description, chapter 3 on page 21) are used.

Chapter 2

Background

In this section, the (mathematical) foundations of graphs will be explained, the work of Eisenberg and Noe is discussed [1], following the internship report [10], and the basics of random network models are presented.

2.1 Financial networks

A *network* (or *graph* in mathematical context) G consists of *nodes* $n \in \mathcal{N}$ and *links* $g_{i\to j} \in \mathcal{E}$. The size of $\mathcal{N}$ is the number of nodes, N. The size of $\mathcal{E}$ is the number of links $\mathcal{L}$. Only simple graphs are considered in this thesis, which means there is at most one link from node i to node j and there are no links from node i to itself. Furthermore, the networks that are considered are *weighted graphs*, i.e. there is a positive weight $w_{i\to j}$ attached to link $g_{i\to j}$. The unweighted representation of the graph is captured in the adjacency matrix A with entries $a_{i\to j}$ where

$$a_{i\to j} = \begin{cases} 1 & \text{if } w_{i\to j} > 0 \\ 0 & \text{if } w_{i\to j} = 0. \end{cases} \tag{2.1}$$

2.1.1 Clearing vectors

In the context of this research, a financial network is a network L with positive-weight directed links $L_{i\to j}$ that represent an obligation from company i to company j. The payments that happen at time t are captured in matrix $\mathcal{P}(t)$ with positive-weight directed links $\mathcal{P}_{i\to j}(t)$. The sum of all entries of $\mathcal{P}$ is called the *total flow*. The interest is in finding how much each company will pay after some time.

Following Eisenberg-Noe [1], for node i, its obligations $L_{i\to j}$ and its total payables $L_i \equiv \sum_j L_{i\to j}$ are constant. Its total receiving cash at time t is $R_i(t) \equiv \sum_j \mathcal{P}_{j\to i}(t)$, which can change in time. L_i and $R_i(t)$ may not be in balance, which can create risk in the system, as explained in the introduction on page 7 and shown by Eisenberg and Noe.

In this system, it is defined that node i, at time t, can never pay more than is had incoming at time $t-1$. This is called *limited liability*, i.e.

$$P_i(t) \leq \sum_j \mathcal{P}_{j\to i}(t-1) = R_i(t-1), \tag{2.2}$$

where $P_i(t)$ is the sum of payments of node i at time t, i.e. $P_i(t) \equiv \sum_j P(t)_{i\to j}$.

Furthermore, (for now) we define that there is *absolute priority*. This means that a node either pays its obligations in full, i.e. $P_i(t) = \sum_j L_{i\to j} = L_i$, or it pays as much as it can:

$$P_i(t) = \sum_j \mathcal{P}_{j\to i}(t-1) = R_i(t-1). \tag{2.3}$$

These two conditions, *limited liability* and *absolute priority* define a *payment vector* $P(t) = \{P_0(t), P_1(t), \dots P_N(t)\}$ of total payments. When $P^*(t) = P^*(t+1)$, P^* is called a *clearing payment vector* or *clearing vector* for short because it 'clears' the financial system.

Note that these two conditions don't prescribe a way of paying obligations when a node is in default. As long as the two conditions are met, $P_{i\to j}(t)$ can be anything, under the condition that a node i doesn't pay more than it has to. Therefore, in general

$$0 \leq \mathcal{P}_{i\to j}(t) \leq L_{i\to j}. \tag{2.4}$$

Eisenberg-Noe's way of finding clearing vectors

In the method of Eisenberg-Noe, a few extra assumptions are made in order to develop a way to find clearing vectors. First, a matrix of relative obligations Π is defined as follows:

$$\Pi_{ij} \equiv \begin{cases} \frac{L_{i\to j}}{L_i} & \text{if } L_i > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

From the definition of L_i, it follows that

$$\sum_j \Pi_{ij} = \sum_j \frac{L_{i\to j}}{L_i} = \frac{\sum_j L_{i\to j}}{\sum_j L_{i\to j}} = 1, \tag{2.6}$$

i.e. Π is row-normalised. Note that, since Π is defined by the obligations matrix L, which is constant in time, the relative obligations matrix is also constant in time.

Now, Eisenberg and Noe assume all debts have equal priority, and the payment of node i to node j $\mathcal{P}_{i\to j}$ is given by

$$\mathcal{P}_{i\to j} = P_i \Pi_{ij}, \tag{2.7}$$

i.e. every payment $\mathcal{P}_{i\to j}$ is a fraction of the total payments P_i, where the fraction is given by the size of the obligation, compared to all other obligations (equation (2.5)).

With equation (2.5) and (2.7), limited liability (equation (2.2)) and absolute priority (equation (2.3)) become $\forall i \in \mathcal{N}$

$$P_i^* \leq \sum_j P_j^* \Pi_{ij}^T, \tag{2.8}$$

and $\forall i \in \mathcal{N}$, node i pays its obligations in full, $P_i(t) = L_i$ or

$$P_i^* = \sum_j P_j^* \Pi_{ij}^T. \tag{2.9}$$

These representations of clearing vectors allow for an analytical way of proving a) there exists a greatest and least clearing vector and b) the value of equity of each node in the financial system is the same under all clearing vectors (theorem 1 in [1]). Value of equity is defined as

$$\sum_j P_j \Pi_{ij}^T - P_i, \tag{2.10}$$

the difference between outgoing and incoming. Note that equity is strictly non-negative.

2.2 Random networks

Throughout this thesis, multiple notions on random networks will be made. In section 4.2 on page 30 the explicit methods for this research are explained. The results are presented in section 5.1 on page 37 and discussed in section 6.1 on page 51. This section will lay the foundations of these sections by going over maximum entropy and the maximum likelihood method, whereafter the Configuration Model is explained [7].

Version of July 14, 2021– Created July 14, 2021 - 12:35

2.2.1 The Maximum Entropy Principle

By 'borrowing' methods from statistical physics, one can find a way to sample random networks in an unbiased manner [11]. In statistical physics, Shannon's entropy is a measure of unpredictability. It is defined using a probability $\mathbb{P}$ of an event happening. In network theory, the entropy S is therefore defined (up to a constant) using the graph probability $\mathbb{P}(G)$. Thus

$$S \equiv -\sum_{G \in \mathcal{G}} \mathbb{P}(G) \ln \mathbb{P}(G), \tag{2.11}$$

where $\mathcal{G}$ is the ensemble of possible graphs G. If entropy is maximal, the unpredictability and therefore the unbiasedness is maximised. Note that finding the functional form of $\mathbb{P}(G)$ is precisely the exercise, which is why it is not defined yet.

In [11], Jaynes has developed the Maximum Likelihood Principle using the Shannon's entropy, which Squartini et al. [3] showed can be used to generate random network ensembles that will realise (on average) some pre-determined constraints and are otherwise *maximally unbiased*. The constraints are enforced only on an expected value over the ensemble. The constraints can be chosen to be a set of m observables $\{x_\alpha(G)\}_{\alpha=1}^m$, measured from a network G. When measuring from an empirical network G^*, the constraints can be canonically enforced by setting the expected value $\langle x_\alpha \rangle$ of observable x_α to be

$$\langle x_\alpha \rangle \equiv \sum_{G \in \mathcal{G}} x_\alpha(G) \mathbb{P}(G) \stackrel{!}{=} x_\alpha^*, \tag{2.12}$$

with x_α^* the empirical measurement of the α-th property.

Now, in order to maximise entropy (equation (2.11)), under the constraints given by equation (2.12) and $\sum_{G \in \mathcal{G}} \mathbb{P} = 1$, the Lagrangian becomes

$$\mathfrak{L}(\mathbb{P}(G), \vec{\theta}) \equiv S + \sum_{\alpha=0}^{m} \theta_\alpha \big(-\sum_{G \in \mathcal{G}} x_\alpha(G) \mathbb{P}(G) + \langle x_\alpha \rangle \big) \tag{2.13}$$

with $\vec{\theta} = \{\theta_\alpha\}_{\alpha=1}^m$ the Lagrange multipliers corresponding to the constraints from equation (2.12) and θ_0 an additional Lagrange multiplier for the normalisation. The Lagrangian is maximised by taking the functional derivative with respect to $\mathbb{P}$ and setting this to zero. The resulting maximum entropy $\mathbb{P}$ is

$$\mathbb{P}(G|\vec{\theta}) = \frac{e^{-H(G,\vec{\theta})}}{Z(\vec{\theta})}, \tag{2.14}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

with a defined *Hamiltonian*

$$H(G,\vec{\theta}) \equiv \sum_{\alpha=0}^{m} \theta_\alpha x_\alpha(G), \tag{2.15}$$

being a linear combination of the constraints and where $Z(\vec{\theta})$ is the *partition function* defined as

$$Z(\vec{\theta}) \equiv \sum_{G\in\mathcal{G}} e^{-H(G,\vec{\theta})}, \tag{2.16}$$

normalising the probability. Equation (2.14) can be recognised as the canonical distribution as commonly used in statistical physics.

Equation (2.14), together with the definitions of the Hamiltonian and the partition functions, can be used to completely define unbiased models with constraints of the form in equation (2.12). To fully use these equations to construct models, a useful relation can be constructed from equations (2.14), (2.15) and (2.16), which is

$$\langle x_\alpha \rangle = -\frac{1}{Z(\vec{\theta})}\frac{\partial Z(\vec{\theta})}{\partial \theta_\alpha} = \frac{\partial \Omega(\vec{\theta})}{\partial \theta_\alpha}, \tag{2.17}$$

where the *free energy* $\Omega(\vec{\theta})$ is defined as

$$\Omega(\vec{\theta}) \equiv -\ln Z(\vec{\theta}). \tag{2.18}$$

The proof of equation (2.17) is presented in Appendix A.1 on page 59.

2.2.2 Constructing models

In general, one could try to measure the presence of every link $g_{i\to j}$ from a network G by measuring $a_{i\to j}$ from the adjacency matrix A and constrain it to be realised on average. In this case, the Hamiltonian from equation (2.15) becomes

$$H(G,\vec{\theta}) = \sum_{i,j} \theta_{ij} a_{i\to j}. \tag{2.19}$$

Here θ_{ij} can be considered the 'cost' of placing a link from i to j and the Hamiltonian is the full cost function. With this Hamiltonian, the partition function from equation (2.16) becomes

$$Z(\vec{\theta}) = \sum_{G\in\mathcal{G}} e^{-\sum_{i,j}\theta_{ij}a_{i\to j}} = \sum_{G\in\mathcal{G}}\prod_{i,j} e^{-\theta_{ij}a_{i\to j}} = \prod_{i,j}\sum_{a_{i\to j}=0,1} e^{-\theta_{ij}a_{i\to j}} = \prod_{i,j}(1+e^{-\theta_{ij}}) = \prod_{i,j} z(\theta_{ij}), \tag{2.20}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

where it is used that summing over all graphs $G \in \mathcal{G}$ is the same (for unweighted graphs with adjacency matrix A) as summing over all possibilities $a_{i\to j} = \{0,1\}$ for all node-pairs $g_{i\to j} \forall i,j \in \mathcal{N}$ and the node-pair partition function z is defined as

$$z(\theta) \equiv (1 + e^{-\theta}). \tag{2.21}$$

Now the free energy from equation (2.18) is

$$\Omega(\vec{\theta}) = -\ln Z(\vec{\theta}) = -\ln \prod_{i,j} z(\theta_{ij}) = -\sum_{i,j} \ln z(\theta_{ij}) = \sum_{i,j} \omega(\theta_{ij}), \tag{2.22}$$

with similarly as above a node-pair free energy defined as

$$\omega(\theta) = -\ln z(\theta). \tag{2.23}$$

Using equation (2.17) and the equations above, one can calculate the probability that there is a link from i to j, which is the expected value of measurable $a_{i\to j}$, as follows

$$p_{i\to j} = \langle a_{i\to j} \rangle = \frac{\partial \Omega(\vec{\theta})}{\partial \theta_{ij}} = \frac{\partial \omega(\theta_{ij})}{\partial \theta_{ij}}, \tag{2.24}$$

which means that

$$p_{i\to j} = \frac{1}{1 + e^{\theta_{ij}}}, \tag{2.25}$$

and the expected total number of links is

$$\langle \mathcal{L} \rangle = \langle \sum_{i,j} a_{i\to j} \rangle = \sum_{i,j} p_{i\to j}. \tag{2.26}$$

In principle, the full Hamiltonian can be used in order to find a model that recreates every link $a_{i\to j}$ in the ensemble average, but it should be noted that the Hamiltonian depends on the vector $\vec{\theta}$ which is $N(N-1)$-dimensional. This makes the problem unfeasible for most applications. Therefore, usually not the individual links are constrained (by (2.12)), but another measure. Furthermore, realising every link on average is not of much interest, as it is very biased, in fact: maximally biased.

Erdős-Rényi

An Erdős-Rényi random graph is a graph where a link is placed from i to j with a constant probability $p_{i\to j} \equiv p$. This random graph can be recreated

Version of July 14, 2021– Created July 14, 2021 - 12:35

using the above methodology when only constraining the number of links from equation (2.26) and setting $\theta_{ij} \equiv \theta$. This reduces the problem from $N(N-1)$-dimensional to 1-dimensional.

The Hamiltonian now becomes

$$H(G, \vec{\theta}) = \theta \sum_{i,j} a_{i\to j} = \theta L(G) \tag{2.27}$$

and by tuning the only free parameter θ, one can realise an empirical measured number of links L^*. The probability of a link from i to j now is

$$p_{i\to j} = p = \frac{1}{1+e^{\theta}}, \tag{2.28}$$

which is a constant as expected.

Configuration Model

The Configuration Model is developed to recreate the expected degrees of nodes [4]. The out and in degree of node i, is the number of out links $k_i^{out} \equiv \sum_j a_{i\to j}$ and the number of in links $k_i^{in} \equiv \sum_j a_{j\to i}$ respectively. In this section, the functional from of the probability $p_{i\to j}$ that there is a link from i to j is derived. Under the constraints

$$\langle k_i^{out} \rangle = \sum_{G\in\mathcal{G}} k_i^{out}(G)\mathbb{P}(G) \stackrel{!}{=} (k_i^{out})^* \tag{2.29}$$

$$\langle k_i^{in} \rangle = \sum_{G\in\mathcal{G}} k_i^{in}(G)\mathbb{P}(G) \stackrel{!}{=} (k_i^{in})^*, \tag{2.30}$$

the Hamiltonian from equation (2.15) on page 15 is

$$H(G, \vec{\theta}) = \sum_{\zeta=0}^{2N} \theta_\zeta x_\zeta(G) = \sum_i \left(\eta_i k_i^{out}(G) + \mu_i k_i^{in}(G)\right), \tag{2.31}$$

where η_i is the langrange multiplier to control for k_i^{out} and μ_i is the langrange multiplier to control for k_i^{in}. By using the definitions of out and in degrees, the Hamiltonian is

$$H(G, \vec{\theta}) = \sum_{i,j} (\eta_i + \mu_j) a_{i\to j}, \tag{2.32}$$

which is a particular case of equation (2.19) where $\theta_{ij} = \eta_i + \mu_j$.

Version of July 14, 2021– Created July 14, 2021 - 12:35

From what's derived in the section above, this Hamiltonian leads, by equation (2.24), to

$$p_{i\to j} = \frac{1}{1+e^{\eta_i+\mu_j}}. \tag{2.33}$$

This leads to the original formulation of the Configuration Model by defining $x_i \equiv e^{-\eta_i}$ and $y_j \equiv e^{-\mu_j}$ as

$$p_{i\to j} = \frac{x_i y_j}{1+x_i y_j}. \tag{2.34}$$

Models for link weights

The methods described above can be used to find the topology of binary graphs. When dealing with weighted graphs, as in this thesis, there are several approaches to construct the weights of links in the random graphs. Some methods, like the Weighted Configuration Model, proceed as above with the entropy given by equation (2.11) on page 14, but constrain the strengths instead of the degrees [12].

In this thesis however, a different method to find the link weights is used as a basis, called MaxEnt. MaxEnt uses a different definition of entropy, which interprets the weights of the graph as probabilities of independent events happening, in contrast to the entropy of equation (2.11), which accounts for the existence of the entire graph [3, 12, 13]. The entropy is redefined in order to find a form for the weights $w_{i\to j} \in (0,\infty)$, and

$$S(W) \equiv -\sum_{i,j} w_{i\to j} \ln w_{i\to j}, \tag{2.35}$$

which must be maximised under the constraints

$$\langle s_i^{out}\rangle = \sum_j w_{i\to j} \stackrel{!}{=} (s_i^{out})^*, \tag{2.36}$$

$$\langle s_i^{in}\rangle = \sum_j w_{j\to i} \stackrel{!}{=} (s_i^{in})^*. \tag{2.37}$$

In appendix A.2 on page 60, it is shown that this definition with these constraints lead to a deterministic way of finding the weights $w_{i\to j}$ from a maximum entropy argument:

$$w_{i\to j} = \frac{(s_i^{out})^*(s_i^{in})^*}{W}, \tag{2.38}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

where $W \equiv \sum_{i,j} w_{i\to j} = \sum_i (s_i^{out})^* = \sum_i (s_i^{in})^*$. This result necessarily leads to a network where all the nodes i with positive $(s_i^{out})^*$ and $(s_i^{in})^*$ are connected, which is not a favourable outcome. This method is later modified, where a variant of the Configuration Model (section above) is used to sample the existence of links and a variant of equation (2.38) is used to place the weights of the realised links.

Chapter 3

Data

As mentioned, the data that are used are made available for the physics and business studies project for the internship. The following chapter describes the data, along the lines of the internship report [10].

3.1 Transactions

The data used for the research are the transaction data available at ING's Wholesale Banking Advanced Analytics (WBAA) tribe. The data are formatted as a table of all (SEPA) transactions that are processed from or to a Dutch ING account, excluding accounts with a foreign currency. On top of those transactions, the table also contains Swift transactions, which are international. Roughly speaking, these are transactions from accounts in the EU countries and their neighbours to accounts in another of these countries. For privacy reasons, the private individuals (PI's) are filtered out*.

The transactions in this table started in October 2018 and the table is updated regularly. This means there are approximately $7 \cdot 10^{10}$ transactions in the transaction table, for $4 \cdot 10^5$ nodes. For every transaction, there are several features available: amount in EUR, the date and for both payer and beneficiary the account, economic and legal ultimate parent, the name and the country where the account holder is established. The sector specification of the accounts (as classified by the North American Industry Classification System, see section 4.2.2 on page 34) is added from another table,

* From ING accounts it is known whether they belong to private individuals. For non-ING accounts, WBAA's Possible Private Individuals (PPI) algorithm classifies accounts as PI or not-PI.

which contains information about ING clients.

In a sense, this transaction table already is an edge list, which could be used to create a graph. This graph could contain multi-edges, where there is more than one transaction between the same nodes. To go from this table to a simple graph, the transactions are grouped by ID[†] and summed, such that every edge occurs only once. For this research, only the transactions in 2019 are considered. This is because 1 year will discard most of the seasonality, and because the data from 2018 wouldn't give a full year (the table starts in October 2018). 2020 was an unusual year, which is why that year was not considered.

After the grouping and filtering, the table is an edge list of a simple graph and it is stored as a Compressed Sparse Row (CSR) sparse matrix, with approximately $3.1 \cdot 10^6$ edges and $3.7 \cdot 10^5$ nodes. This is the graph that is considered to be 'the transaction network' throughout this thesis.

3.2 The Transaction Network

Only a subset of the nodes in the transaction network are companies with a Dutch ING account with relevant transactions, 'Dutch ING' meaning they have an account at ING and are in the internal database. From these accounts, every SEPA and Swift transaction from October 2018 until now is in the data set. That is no guarantee, however, that everything about these companies is known. We do know that more is known about them, than about e.g. ABN accounts. For these non-ING accounts, only the transactions from ING to those accounts and from those accounts to ING are observable. There is a blind spot for transactions from and to these accounts to and from other banks (i.e. Rabobank, ABN, Triodos, etc.) or foreign accounts.

Because of this fact, a distinction is made between Dutch ING companies and the other nodes. Using IBAN, the internal ING database, sector information (saved as a NAICS-code, see section 4.2.2 on page 34), the distinction can be made. An account is considered a Dutch ING account when 1) it has 'NL' and 'INGB' in its IBAN, 2) its country, according to our database, is 'NL' and 3) it has an ID in our database, which 4) does not correspond to a private individual account. The other nodes are grouped and considered as separate group nodes. The group nodes are

- Public administrations (e.g. 'Belastingdienst', the Dutch tax office).

[†]The ID roughly corresponds to an economic entity in an internal ING table, however it is not necessarily the case that 1 company only has 1 ID.

Version of July 14, 2021– Created July 14, 2021 - 12:35

This is based on the NAICS code 92.

- Private individuals, based on a flag in the internal database. See also footnote $*$ on page 21.
- Foreign accounts, based on the country as present in the internal database.
- Financial institutions (e.g. insurance companies and banks), based on having NAICS code 521, 522 or 523.
- Dutch non-ING companies, which have a country 'NL' according to our database and have a defined sector in the database. Only accounts with a known sector are considered because of those accounts we're sure they're companies and not private individuals.
- Other, where all of the above requirements are not met. This group unfortunately contains batch payments.

The subgraph of only ING nodes is named the 'ING network' or 'Internal Network' and has approximately $2.7 \cdot 10^6$ edges and $2.8 \cdot 10^5$ nodes. The other (group) nodes are considered to be 'exogenous forces'. Figure 3.1 on the next page displays the presence of the external nodes, compared to the Internal Network, in terms of the total flow of money and count of transactions.

Characteristics of the Transaction Network

In figure 3.2 on page 25, the transaction network is described by 4 histograms. The properties that are displayed are the (out and in) degree k_i^{out}, k_i^{in}, and the weighted (out and in) degree s_i^{out}, s_i^{in}. The latter is also called the strength. The properties can be directly measured for any graph G with entries $w_{i\to j}$ and its respective adjacency matrix entries $a_{i\to j}$ (see equation (2.1)), by the following

$$k_i^{out} = \sum_{j\neq i} a_{i\to j}, \qquad k_i^{in} = \sum_{j\neq i} a_{j\to i} \tag{3.1}$$

$$s_i^{out} = \sum_{j\neq i} w_{i\to j}, \qquad s_i^{in} = \sum_{j\neq i} w_{j\to i}. \tag{3.2}$$

Generally speaking, the *distribution* of degrees and strengths is more informative than the individual degrees and strengths. In real-world networks, these distributions are highly non-trivial, as shown in [4]. It is

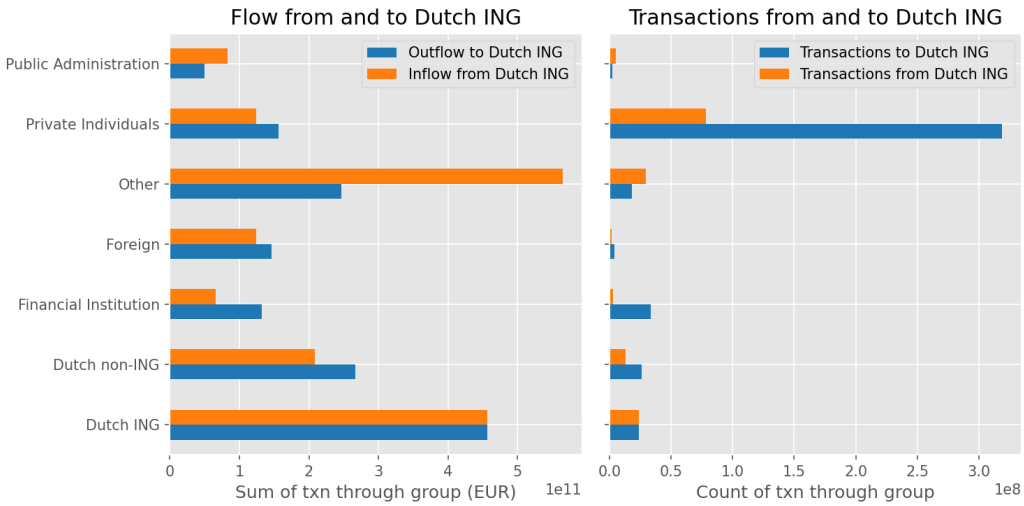


Figure 3.1: *The figure shows which type of transactions are present in the transaction network. The figure should help understand what the distribution of internal-external transaction is and what type of transactions are filtered. A figure with the strength of all groups can be found in the appendix (figure B.1 on page 64).*

shown that the degree distributions of real-world networks have a long tail, i.e. there are more nodes with a large degree than expected from a completely random graph as described by Erdős-Rényi [14]. The long tails make measuring the distribution with a histogram somewhat tricky, but an accepted way to overcome this is by using a log-log scale and exponential bins. The degree distributions in figure 3.2 appear to be power-law distributions, which is to be expected of real-world networks [4].

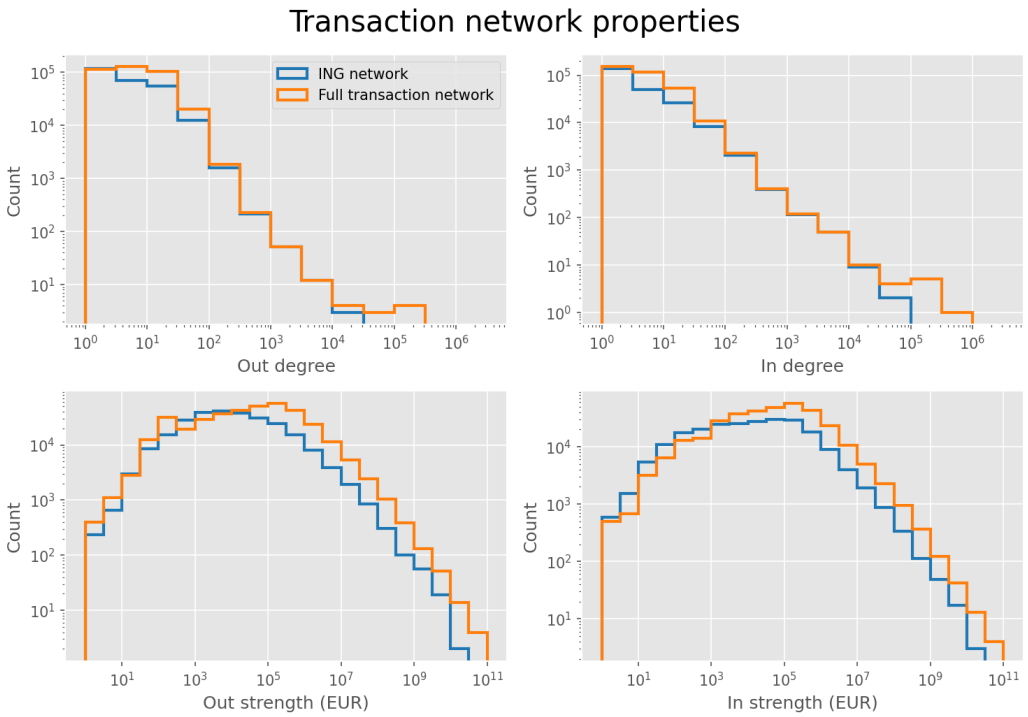


Figure 3.2: *The figure shows the network properties of the transaction network used for the research. Note the logarithmic scales. The grouped nodes (foreign accounts, other banks, etc.) are considered only in the full transaction network. Because they represent a large number of accounts, they have an extremely large out and in degree and are filtered for the ING network. When not considering the external nodes, there are fewer nodes with large strengths or degrees.*

Chapter 4

Methods

In this chapter, all the methods used in the project are presented. In section 4.1, all the aspects of the simulation of cascading defaults and default strategies, as explained in the internship report [10], are discussed. Then, in section 4.2 on page 30, the network models and their construction are explained.

4.1 Simulation

The goal of this project is to research the dynamics of cascading defaults on networks reconstructed from statistical ensembles. To fully reach this goal, we first need to look at how the simulation is done. Following the internship report [10], the domino effect (or 'cascade') of defaults in a financial network, as introduced in the Introduction, is investigated. Section 2.1.1 on page 11 explained a way of finding clearing vectors of a financial system, under certain assumptions as introduced by Eisenberg and Noe [1]. This is done using a simulation called the fictitious default algorithm.

In Eisenberg and Noe, this algorithm is nothing more than a matrix multiplication under the two constraints of limited liability and absolute priority and the assumption that all debts have equal priority. Here, a more general approach of simulating defaults is proposed where 1) not all debts have equal priority, but a debtor can distinguish creditors by the size of the respective obligation and 2) the absolute priority constraint is relaxed, as a node can choose to save money to reserves instead of paying.

4.1.1 Fictitious Default Algorithm

To simulate the defaults, the *fictitious default algorithm* as proposed by Eisenberg and Noe [1] is adapted. The algorithm allows measuring a node's systemic risk exposure: the earlier it defaults, the more it is at risk. The thinking behind the simulation is relatively simple:

First, each node pays its obligations if it can. If it cannot, the node is by definition 'in default'. Then, the node pays what it can, according to a certain defaulting strategy (see the next section). At every time step, the weighted matrix of payments $\mathcal{P}(t)$ is observed and the sum of all entries $\sum_{i,j} \mathcal{P}_{i \to j}(t)$ gives a measure for the *total flow*. The algorithm is written in pseudo-code in algorithm 1. Note that debts don't carry over time and are forgiven after every iteration.

Algorithm 1: Fictitious default algorithm

for $n \in \mathcal{N}$ **do**
  $\texttt{obligations}_n = \sum_m L_{n \to m}$
  $\texttt{incoming}_{0,n} = \sum_m L_{m \to n}$
  $\texttt{reserves}_{0,n} = \max\left(0, \left(\texttt{incoming}_{0,n} - \texttt{obligations}_n\right)\right)$
$\texttt{P}_0 = [[0, .., 0], ..[0, .., 0]]$
for $t \in range(1, max\_iterations)$ **do**
  $\texttt{incoming}_{t,n} = \sum_m P_{(t-1)_{m \to n}} + \texttt{reserves}_{(t-1),n}$
  for $n \in \mathcal{N}$ **do**
    if $incoming_{t,n} \geq obligations_n$ **then**
      $\texttt{payments}_{t,n} = \texttt{obligations}_n$
    else
      $\texttt{payments}_{t,n} = \texttt{payments\_strategy}_n$
    $\texttt{reserves}_{t,n} = \left(\texttt{incoming}_{t,n} - \texttt{payments}_{t,n}\right)$
  $\texttt{P}_t = stack(\texttt{payments}_{t,n})$

4.1.2 Defaulting Strategies

When a node is in default, the node could still pay part of its obligations. The node does this according to a certain defaulting strategy. Here, the investigated strategies are described and briefly motivated.

Eisenberg Noe [1]

In 2001, Eisenberg and Noe described a mechanism to analytically find clearing vectors (see subsection 2.1.1 on page 11). Their thinking can be

adapted to describe a defaulting strategy. In this defaulting strategy, a defaulted node n pays all their incoming money to their creditors, weighted by the relative liabilities matrix Π_{ij}, see equations (2.5) and (2.7) on page 13.

The motivation behind this default strategy is mainly simplicity. With the relative liabilities matrix Π_{ij} described as above, the algorithm becomes an analytically solvable problem, i.e. the system always converges to a steady-state which can be found using Π_{ij}, as shown in [1]. One could also argue that from an economic perspective, this defaulting strategy is fair: if a debtor is in default, all their creditors suffer proportionally to the size of the obligation.

Largest Creditor

A defaulting strategy that has more motivation from reality is the *Largest Creditor* strategy. It is hypothesised by bankers from within an arrears department of ING that defaulted companies pay obligations according to their respective size, such that a company has fewer creditors to deal with. There are two variants of this strategy: (Pay) Largest Creditor First and Largest Creditor Last. When using this strategy, a node that is in default sorts all its obligations (on size in EUR) and pays obligations from that sorted list until it runs out of money (see Algorithm 2).

Algorithm 2: Largest Creditor

$\texttt{sorted\_obligations}_n = sort(\texttt{obligations}_n)$
$\texttt{payments}_n = 0$
$i = 0$
while $payments_n + sorted\_obligations_n[i] < incoming_n$ **do**
  $\texttt{payments}_n += \texttt{sorted\_obligations}_n[i]$
  $i += 1$
return $\texttt{payments}_n$

The result of Largest Creditor First for a defaulting node, is, by construction, that the node has fewer obligations of 'large' size, in popular language 'no big bills lying around'. The result of Largest Creditor Last is fewer obligations in terms of number of creditors and therefore, in popular language 'fewer parties to deal with'.

4.1.3 Reserves (R vs. XR)

For the Largest Creditor First and Largest Creditor Last strategies, a defaulting node may pay less than it has incoming. This also happens when

making a profit. When a node pays less than it has incoming, there are two options to force conservation of money. The money can be saved to a 'reserve', which is considered income in the next time step (strategy **R**), or, for nodes in default, the money is paid to a creditor, even when this payment doesn't fulfil the entire obligation (strategy **XR**). For simplicity, the excess money is always paid to the creditor that would've received the next payment of the defaulting node.

Note that using the **XR** strategy coincides with the assumption of absolute priority and that the **R** strategies no longer obey absolute priority. Convergence, which was guaranteed for clearing vectors, i.e. obeying limited liability and absolute priority is not guaranteed anymore when using the **R** strategy.

When building reserves, an unpaid obligation can become a paid obligation after a few iterations when a defaulted node saves money until it has enough incoming again to fully pay an obligation. When not building reserves and having a fixed order of preference for paying, it can be shown that the total flow of money (per node) is strictly non-increasing, by the following argument.

When a node is healthy, i.e. it has at least as much incoming as obligations, it will pay its obligations. The total flow of this node is exactly the sum of its obligations (a constant in time). When, for some reason, at a later point in time the incoming money of this node is less than its obligations, it will spend everything it has at that point. All of its neighbours will therefore receive at most what they already received, and thus never more. The same goes for all other nodes and thereby: when not building reserves, the total flow of money (per node and in total) is strictly non-increasing.

4.2 Random Networks

In order to make sense of the measurements on the transaction network, the results are compared to a null model. An ensemble of random networks is sampled and used as a comparison. The random networks are generated in such a way that statistical properties are realised in the ensemble averages. This allows the ensemble to act as a *null model*: when behaviour of the empirical network, e.g. the dynamical behaviour of the cascading defaults simulation, result in different outcomes than the random network ensemble, the behaviour is not fully explained by the constrained statistical properties.

In this research, the random network models are designed to replicate

the total number of links and the individual strengths of all the nodes. A random network model that fits for this purpose is the Fitness-induced Configuration Model (FiCM) [7, 8], possibly enhanced with labelled edges and nodes in order to replicate flows between sectors [9].

4.2.1 Fitness-induced Configuration Model

Finding the topology

The configuration model (CM) [7] can realise an empirical degree distribution, by having two Langrange multipliers $\{x_i, y_i\}$ for every node $i \in V$ in the network. The model is generally fitted from a maximum entropy argument, using each of the node's empirical out and in degree $(k_i^{out})^*$ and $(k_i^{in})^*$ respectively, as the expected out and in degree $\langle k_i^{out} \rangle$ and $\langle k_i^{in} \rangle$ respectively of that node in the ensemble. See section 2.2.2 on page 17 for the details. In general, the configuration model gives rise to the functional form

$$p_{i \to j} = \frac{x_i y_j}{1 + x_i y_j}. \tag{4.1}$$

In the Fitness-induced Configuration Model (FiCM) [8], it is *assumed* that the Langrange multipliers x_i and y_i correlate linearly with some (measurable) node-specific fitness χ_i and ψ_i, determining the out and in degrees, respectively, through universal parameters α and β, i.e.

$$x_i \equiv \sqrt{\alpha}\chi_i \qquad\qquad y_i \equiv \sqrt{\beta}\psi_i. \tag{4.2}$$

This is called the fitness ansatz. Now equation (4.1) becomes

$$p_{i \to j} = \frac{z\chi_i\psi_j}{1 + z\chi_i\psi_j}, \tag{4.3}$$

where $z \equiv \sqrt{\alpha\beta}$ [15]. Given the (empirical) node-specific fitnesses $\{\chi_i, \psi_i\}_{i \in V}$, which can be any node attributes that are assumed to affect the degree by equation (4.2), e.g. the out and in strengths, the only free parameter is z.

The parameter z can be found by an algebraic equation. The CM realises the empirical degree distribution by design. The sum of the out and in degree distributions is the number of links $\mathcal{L}$ in the networks. The empirical degree distributions are realised, thus the total number of links $\mathcal{L}^*$ should also be realised. Now z can be found by solving

$$\mathcal{L}^* \stackrel{!}{=} \langle \mathcal{L} \rangle = \sum_{i,j} p_{i \to j} = \sum_{i,j} \frac{z\chi_i\psi_j}{1 + z\chi_i\psi_j}. \tag{4.4}$$

When z is found, a binary topology $\tilde{A}$ of a random sample $\tilde{G}$ is realised by placing an edge in the random network when a randomly generated number is smaller than $p_{i\to j}$. This means for an edge $\tilde{a}_{i\to j}$ in the sample:

$$\tilde{a}_{i\to j} = \begin{cases} 1, & \text{with probability } p_{i\to j} = \frac{z\chi_i\psi_j}{1+z\chi_i\psi_j} \\ 0, & \text{with probability } 1 - p_{i\to j}. \end{cases} \tag{4.5}$$

Finding the weights

As explained in section 2.2.2 on page 18, the empirical strengths are realised by the random networks from a maximum entropy viewpoint when

$$\tilde{w}_{i\to j} = \frac{(s_i^{out})^*(s_j^{in})^*}{W}, \tag{4.6}$$

with $(s_i^{out})^* = \sum_j w^*_{i\to j}$, $(s_i^{in})^* = \sum_j w^*_{j\to i}$ the empirical out and in strength of node i, respectively, and $W = \sum_{i,j} w^*_{i\to j}$ the sum of empirical weights in the network. This functional form, however, assumes that the network is *fully connected* and is therefore unfeasible for the proposed random topology (see section 2.2.2 on page 18).

Two modifications to equation (4.6) can be made to make it feasible. First, rather than placing the weights deterministic like in the equation, the weights can be recreated in the ensemble average by sampling from a positive support probability distribution with a mean $\mu = \tilde{w}_{i\to j}$. The exponential distribution is used for this, because it is the continuous distribution that maximises the entropy under the constraint of the mean. The exponential distribution is given by

$$\mathbb{P}(x|\mu) = \frac{1}{\mu} e^{-x/\mu}. \tag{4.7}$$

Second, when trying to create a sparse (not fully connected) network, by sampling the edges, e.g. by equation (4.5), the value $p_{i\to j}$ can be added to the denominator in order to still recreate the expected strengths. The expected weights now become

$$\langle \tilde{w}_{i\to j} \rangle = \frac{(s_i^{out})^*(s_j^{in})^*}{W p_{i\to j}} \tilde{a}_{i\to j}, \tag{4.8}$$

which realises the empirical strengths in the ensemble average. This can explicitly be shown by the following:

$$\langle s_i^{out} \rangle = \Big\langle \sum_j \tilde{w}_{i\to j} \Big\rangle = \frac{(s_i^{out})^*}{W} \sum_j \frac{(s_j^{in})^*}{p_{i\to j}} \langle \tilde{a}_{i\to j} \rangle = (s_i^{out})^*, \tag{4.9}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

and analogous for $\langle s_i^{in} \rangle$.

In summary, using equation (4.5) to create the binary topology of a random network, with a z found by solving equation (4.4) and by using equation (4.8) to place the weights, an ensemble of random networks can be made where, on average, the number of links, out and in degree distributions and out and in strength distributions are realised and where the other properties are maximally unbiased and random. This is under the assumption that the strengths correlate with the Lagrange multipliers and therefore with the degrees of nodes [8].

4.2.2 Stripe Fitness model

The Fitness-induced Configuration Model assumes that there exist universal parameters α and β that couple the fitnesses to the Lagrange multipliers. This universality can constrain the model, as it does not allow for other (node) properties that control the degrees: all nodes experience the same relation described in equation (4.2) on page 31.

The Stripe Fitness model [9] generalises the FiCM [8] and allows for different relations between the Lagrange multipliers and fitnesses to coexist. Nodes get a label g (e.g. a sector) and for every label that is present in the model, a z_g is fitted by the procedure described above. This means that equation (4.3) on page 31 generalises to:

$$p_{i\to j} = \frac{z_{g_i} s_i^{out} s_{g_i\to j}}{1 + z_{g_i} s_i^{out} s_{g_i\to j}}, \tag{4.10}$$

and again

$$\tilde{a}_{i\to j} = \begin{cases} 1, & \text{with probability } p_{i\to j} \\ 0, & \text{with probability } 1 - p_{i\to j}, \end{cases} \tag{4.11}$$

where

$$s_i^{out} = \sum_j w_{i\to j} \qquad\qquad s_{g_i\to j} = \sum_{k\in g_i} w_{k\to j}. \tag{4.12}$$

The expected weights are generalised as follows:

$$\langle \tilde{w}_{i\to j} \rangle = \frac{s_i^{out} s_{g_i\to j}}{W_{g_i} p_{i\to j}} \tilde{a}_{i\to j}, \tag{4.13}$$

with

$$W_{g_i} = \sum_{k\in g_i} s_k^{out} = \sum_j s_{g_i\to j} \tag{4.14}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

The quantity $s_{g_i \to j}$ can be thought of as the in strength from a certain labelled group. The motivation is that a node might *need* a certain influx from nodes with a certain label (e.g. sector) g_i* . Using the equations from the Stripe Fitness model, the node has no bias to any node in the group g_i, and in the ensemble, the node is certain to have on average influx $s_{g_i \to j}$.

This can be shown using a similar argument as above in (4.9). For a node i, its influx from group g_j is, averaged over the ensemble:

$$\langle \tilde{s}_{g_j \to i} \rangle = \langle \sum_{k \in g_j} w_{k \to i} \rangle = \langle \sum_{k \in g_j} \frac{s_k^{out} s_{g_j \to i}}{W_{g_j} p_{k \to i}} \tilde{a}_{k \to i} \rangle, \tag{4.15}$$

which is equal to

$$\langle \tilde{s}_{g_j \to i} \rangle = \frac{s_{g_j \to i}}{W_{g_j}} \sum_{k \in g_j} \frac{s_k^{out}}{p_{k \to i}} \langle \tilde{a}_{k \to i} \rangle. \tag{4.16}$$

Now $\sum_{k \in g_j} s_k^{out}$ and W_{g_j} cancel, as well as $\langle \tilde{a}_{k \to i} \rangle$ and $p_{k \to i}$, resulting in

$$\langle \tilde{s}_{g_j \to i} \rangle = s_{g_j \to i}. \tag{4.17}$$

The out strength of node i, averaged over the ensemble, can be found using a similar procedure:

$$\langle \tilde{s}_i^{out} \rangle = \frac{s_i^{out}}{W_{g_i}} \sum_j \frac{s_{g_i \to j}}{p_{i \to j}} \langle \tilde{a}_{i \to j} \rangle = s_i^{out}, \tag{4.18}$$

where equation (4.14) is implicitly used in the last step.

Choosing number of groups

In the research, the nodes of the financial network are companies. For the companies i, a natural group g_i is the sector. The Stripe Fitness model replicates (on average) for every node i, the influx from every group g_j. In this case, the model replicates for every company the incoming cash from a certain sector. 'Sector', however, is not trivial to define.

*Note that technically, one should consider this to be 'outflux of money' or 'influx of product', since a link corresponds to a monetary transaction from node i to j and a product from j to i. In the Stripe Fitness model, one should pay close attention to which party (payer or beneficiary) is labelled with the sector. In this research, the beneficiary of the transaction determines the group that is considered, which corresponds to the seller of a product.

Version of July 14, 2021– Created July 14, 2021 - 12:35

Codes	Titles
11	Agriculture, Forestry, Fishing and Hunting
1111	Oilseed and Grain Farming
111110	Soybean Farming
111120	Oilseed (except Soybean) Farming
111130	Dry Pea and Bean Farming
111140	Wheat Farming
111150	Corn Farming
111160	Rice Farming
111191	Oilseed and Grain Combination Farming
111199	All Other Grain Farming
1112	Vegetable and Melon Farming
111211	Potato Farming
111219	Other Vegetable (except Potato) and Melon Farming

Table 4.1: *A small piece of the NAICS-table. NAICS provides a way of hierarchically grouping sectors [16].*

For sectors, the North American Industry Classification System (NAICS) is used, because it is available in the data [16]. The system provides a way of hierarchically classifying sectors into subsectors. The first two digits provide a broad description of the sector, and with every next digit, there is more granularity. See table 4.1 for an example.

One advantage of using NAICS is that sectors can be grouped, which can be useful to ensure enough companies in one group. When for example, there are only 5 companies in the entire data set with NAICS-code *111191: Oilseed and Grain Combination Farming*, they can be considered as being part of the larger group *1111: Oilseed and Grain Farming*, decreasing granularity. This way, companies get attributed a sector label in such a way that there are at least l nodes with the same sector.

The factor l can be used to control for granularity. l can range from 1 to N, with N the number of nodes in $\mathcal{N}$. When $l = 1$, the sector of every node will be its 6-digit NAICS-code and some sectors can have only 1 node. When $l \geq N$, there is only one sector in the data and the Stripe Fitness model reduces back to the FiCM. When $l = 15$, there are 737 different sectors measurable with more than 15 companies.

Version of July 14, 2021– Created July 14, 2021 - 12:35

Chapter 5

Results

This chapter will try to cover all the results of the research. First, the random network ensemble is compared to the empirical network to check whether it can be used as a null model for the second part, the cascading defaults simulations.

5.1 Random networks

An important step in generating random network ensembles is checking its realisations with the empirical network. Remember that the FiCM [8] and Stripe Fitness model [9] both were based on maximum entropy, which ensures maximally randomness, and the assumption that the strengths (albeit strength of a group g) were a good proxy for the Lagrange multipliers in the Configuration model [7]. Because this is an assumption, analytical results should be checked numerically.

5.1.1 Number of links

In table 5.1 on the following page, it can be seen that for all the random network ensembles the number of links is perfectly realised, with a minimal error. This result was to be expected, as the expected number of links of the ensemble is equal to the empirical number of links, by design (equation (4.4) on page 31).

Network	# of sectors	Number of links	Relative error
Empirical	n/a	$2.73160 \cdot 10^6$	n/a
Stripe Fitness model	60	$2.73168 \cdot 10^6 \pm 1.319 \cdot 10^3$	0.059σ
Stripe Fitness model	204	$2.73175 \cdot 10^6 \pm 1.216 \cdot 10^3$	0.120σ
Stripe Fitness model	563	$2.73155 \cdot 10^6 \pm 1.353 \cdot 10^3$	-0.036σ
Stripe Fitness model	737	$2.73176 \cdot 10^6 \pm 1.399 \cdot 10^3$	0.115σ
FiCM	1	$2.73162 \cdot 10^6 \pm 1.504 \cdot 10^3$	0.010σ

Table 5.1: *The empirical number of links and ensemble averages of the number of links. The errors are given by the standard deviation over all 50 realisations.*

5.1.2 Degrees

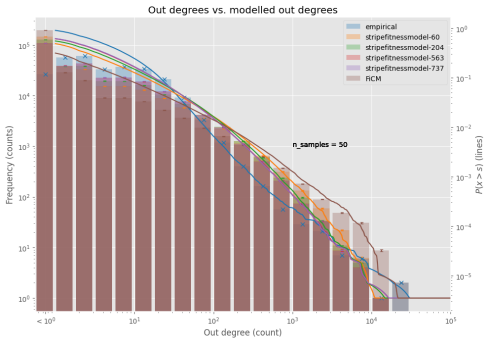
The Configuration Model should be able to realise the degree distribution of the empirical network. In the proposed models, i.e. the FiCM and Stripe Fitness model, the Configuration Model is used as an inspiration. There is no analytical guarantee, however, that the expected degrees are equal to the empirical ones.

After sampling networks from the ensemble, the degree (out and in) distributions and inverse cumulative distribution functions (icdf) are inspected using figure 5.1 on the facing page. The top panels (figures 5.1a and 5.1b) show the degree distribution and icdf of an average realisation. For every realisation a degree distribution and icdf are made. Then, for every sample in the ensemble, the average height of the bin (and the standard deviation for the error bar) and average of the icdf is calculated and displayed.

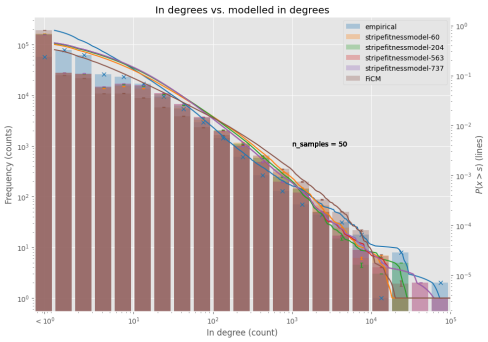
The bottom panels (figures 5.1c and 5.1d) show the ensemble-averaged degree distributions and icdf. These are calculated by averaging the degree of every node over the ensemble and then creating the distribution and icdf for the ensemble-averaged degree distribution. The error bar is $\sqrt{N}$ with N the height of the bar in the histogram, to indicate the error.

When inspecting the distributions and icdf's, several point can be made. First, the number of nodes with a low in degree ($< \ 5 \cdot 10^1$) is greatly underestimated (note the log-scale) by the models. Second, the number of nodes with a medium to large out degree ($5 \cdot 10^1 \ < \ k^{out} \ < \ 5 \cdot 10^3$), is overestimated by the models. These points can be made by comparing the height of the bins of the models to the height of the bins of the empirical network (in blue).

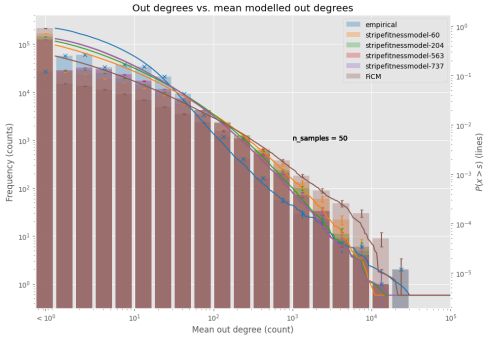
Furthermore, the number of nodes with no outgoing connections ($k^{out} = 0$) or no incoming connections ($k^{in} = 0$), which is displayed in the left bar of the histogram, is far greater ($\mathcal{O}(10^4)$) in the random networks than in



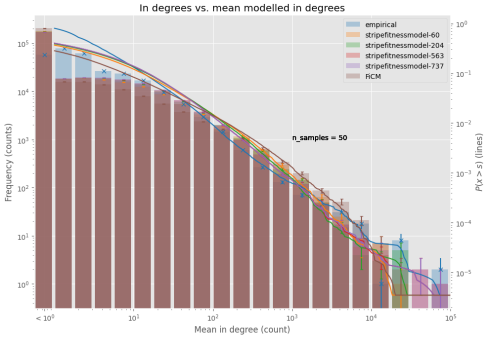
(a) *Empirical out degrees (blue) and other out degrees (histograms averaged over all realisations).*



(b) *Empirical in degrees (blue) and other in degrees (histograms averaged over all realisations).*



(c) *Empirical out degrees (blue) and ensemble mean out degrees.*



(d) *Empirical in degrees (blue) and ensemble mean in degrees.*

Figure 5.1: *Degrees of empirical and random networks. The bars display the count (left axis), the lines display the inverse cumulative distribution function (right axis). Figures (a) and (b) show the average distributions (i.e. first make a distribution, then average and standard deviation for the error bars). Figures (c) and (d) show the ensemble averages, with $\sqrt{N}$ error bars. The models are the Stripe Fitness model with different number of labels (section 4.2.2 on page 33) and the Fitness-induced Configuration Model (FiCM) (section 4.2.1 on page 31).*

the empirical network, which is expected of random networks sampled from a canonical ensemble (see section 6.1 on page 51).

The last point is that the icdf of the observed out degrees is not clearly from a power-law, which would have produced a straight line in log-log plot. This behaviour is not replicated by the models, which have a more power-law-like shape for the icdf of the out degrees.

5.1.3 Strengths

Distributions

When inspecting the strength distributions (figure 5.2 on the facing page), the strengths and limitations of the model become even more visible than when analysing the degrees. In the figures, which are similar to the figures shown in section 5.1.2 on page 38 but for strengths, the empirical strength distributions are compared to the ensemble strength distributions.

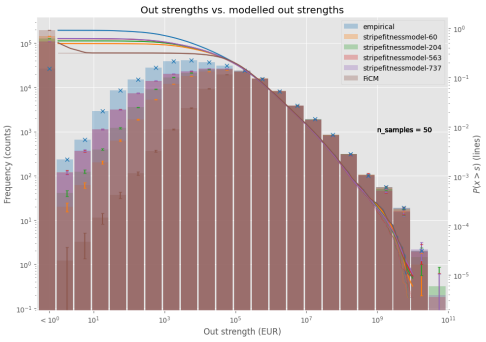
As expected by equation (4.18) on page 34, the distribution of the ensemble averaged out strengths are (approximately) equal to that of the empirical out strengths, as can be seen in figure 5.2c. The number of low in strength nodes (figure 5.2d) is underestimated by the models, and the number of zero in strength nodes is overestimated by the models.

The top panels 5.2a and 5.2b, where the strength distributions of an average realisation are displayed show that in the average realisation, the number of low-strength nodes is massively underestimated (for in strengths a factor around $\mathcal{O}(10^1)$) and the number of zero out or in strength node is overestimated by a number of $\mathcal{O}(10^4)$, which is $\mathcal{O}(10)\%$ of the total number of nodes. This result and its implications are discussed in section 6.1 on page 51.

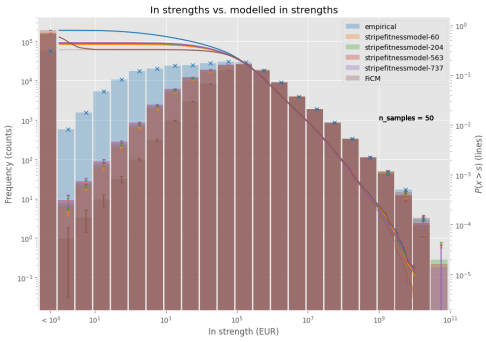
Expected values

To get a sense of how the models recreate the strength distributions, one can sample a number of random networks and calculate the mean strength of every node. A distribution of these strengths is depicted in figure 5.2c on the facing page, as already discussed above. Figures 5.3 on page 42 show the mean value of the strengths versus the empirical strength. There are therefore $\mathcal{N} \approx 2.8 \cdot 10^5$ points in these figures and the figure is displayed as a 2D histogram.

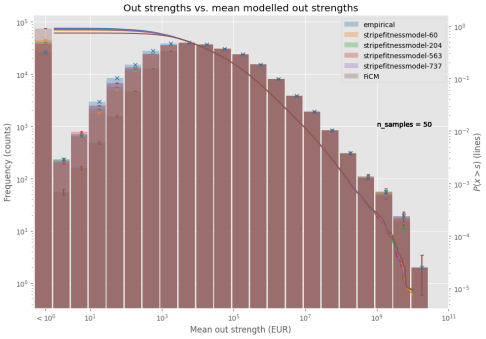
From these figures, it can be concluded that generally, nodes with a large empirical strength ($s_i > 10^5$) have the same mean strength in the



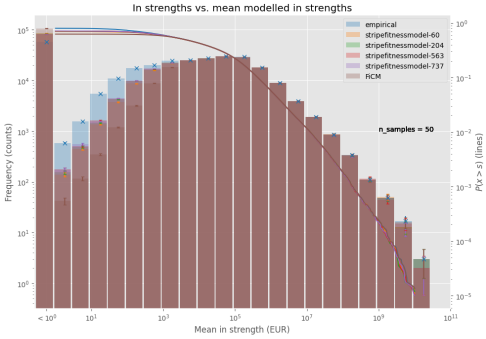
(a) *Empirical out strengths (blue) and other out strengths (histograms averaged over all realisations).*



(b) *Empirical in strengths (blue) and other in strengths (histograms averaged over all realisations).*

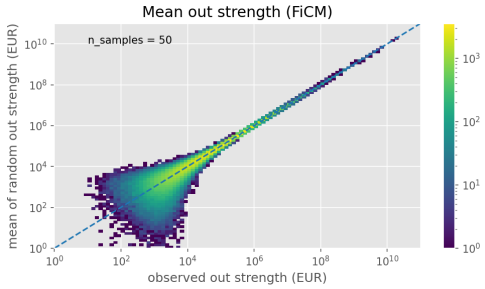


(c) *Empirical out strengths (blue) and ensemble mean out strengths.*

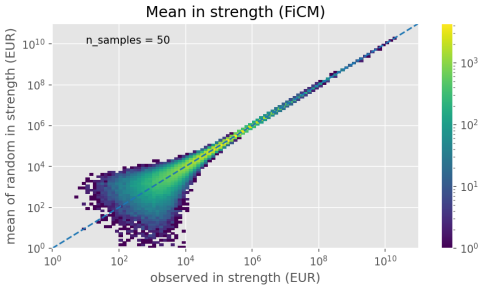


(d) *Empirical in strengths (blue) and ensemble mean in strengths.*

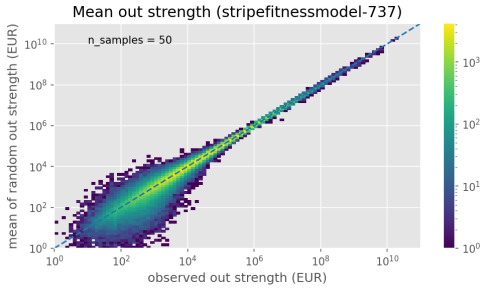
Figure 5.2: *Strengths of empirical random networks. Figures (a) and (b) show the average distributions (i.e. first make a distribution, then average and standard devation for error bars). Figures (c) and (d) show the ensemble averages with a $\sqrt{N}$ error bar. The left axis corresponds to the bars (frequency), the right axis corresponds to the lines (inverse cumulative distribution function). The models are the FiCM (see section 4.2.1 on page 31) and the Stripe Fitness model with different number of labels (see section 4.2.2 on page 33).*



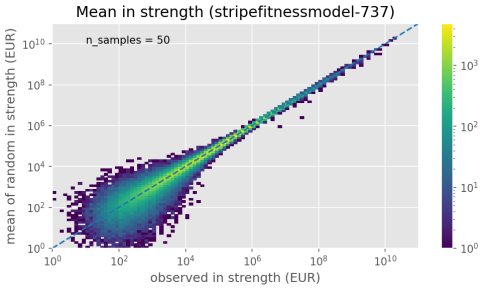
(a) *Comparison between empirical out strengths and the mean modelled out strengths, using the FiCM.*



(b) *Comparison between empirical in strengths and the mean modelled in strengths, using the FiCM.*



(c) *Comparison between empirical out strengths and the mean modelled out strengths, using the Stripe Fitness model with 737 labels.*



(d) *Comparison between empirical in strengths and the mean modelled in strengths, using the Stripe Fitness model with 737 labels.*

Figure 5.3: *In the figures, the empirical strengths are plotted against the mean sampled strength. The Stripe Fitness model with 737 labels and the FiCM, which is just a Stripe Fitness model with 1 label, are used to show the result. The results of the intermediate Stripe Fitness models (60, 204 and 563 labels) gradually shift in shape from the FiCM to the Stripe Fitness 737. The figure with all the strength realisations can be found in figure B.3 on page 66.*

ensemble. For smaller strength nodes, the spread is somewhat larger. This is also discussed in section 6.1 on page 51.

FiCM vs. Stripe Fitness model

In general, an observation on the Stripe Fitness model can be made from figures 5.1 and 5.2. The Stripe Fitness model replicates both the degrees and the strengths, both for the individual realisations and for the ensemble-averaged distributions, better than the FiCM. The more labels that are used, the better the realisations.

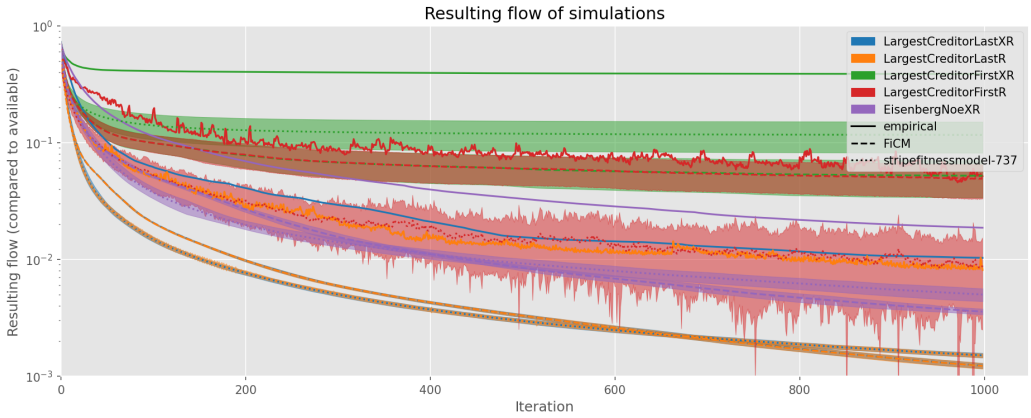


Figure 5.4: *(Full size in appendix figure B.2 on page 65). The flow of money in time, depicted for different defaulting strategies (colours) when the simulation is run for different networks (line styles). 100 random networks are sampled from the ensemble and the flow is recorded at each time. For that time, the mean is depicted with the line and the area is $\pm 1\sigma_t$ with σ_t the standard deviation of the flow of the ensemble at that time.*

5.2 Cascading defaults

In this section, the results of the cascading defaults simulations from the internship report [10] are reiterated. The simulations are done using the empirical network and 100 random networks sampled from a certain ensemble (FiCM or Stripe Fitness model with 737 labels, i.e. $l \geq 15$).

5.2.1 Flow and not-defaulted companies

Dynamics

The first observed quantity is the change of the total flow of money in the network $\sum_{i,j} P_{i\to j}(t = t')$, relative to the total obligations $\sum_{i,j} L_{i\to j}$. Figure 5.4 shows some interesting results. The first observation is that for **XR** strategies (always paying as much as you can, see section 4.1.3 on page 29), the total flow is strictly non-increasing, as expected. For *LargestCreditorFirst XR* (in green), the total flow even reaches a non-zero equilibrium in $t < 1000$, also in the random networks sampled from the Stripe Fitness model ensemble.

On the other hand, the total flow for **R** strategies, where you can build a reserve, even when you're in default, fluctuates for both the empirical and the random networks. It is clear, however, that the trend is overall

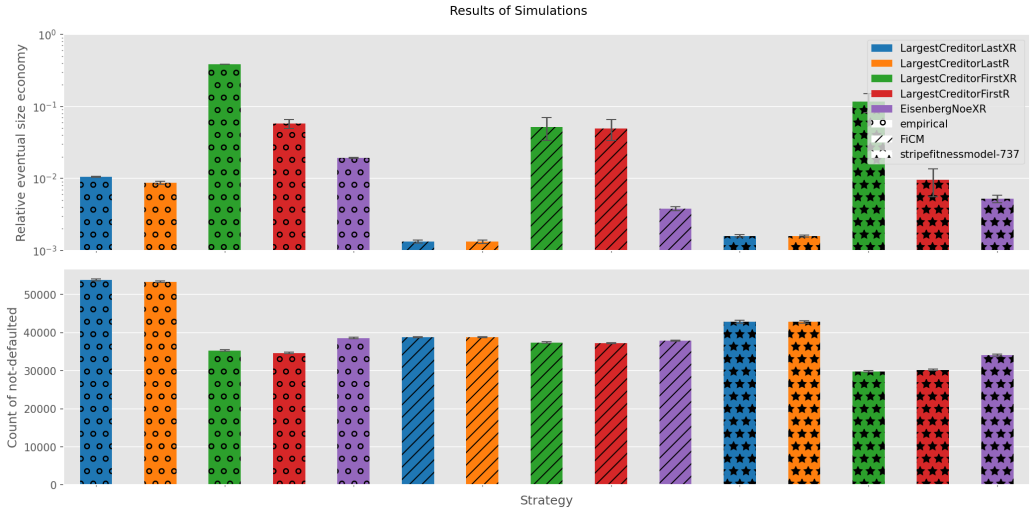


Figure 5.5: *The results at the end of the simulations, sorted by defaulting strategy (colour) and network (hatches). The top figure shows the eventual flow $\sum_{i,j} P_{i\to j}$ relative to the total obligations $\sum_{i,j} L_{i\to j}$ on a log scale. From this, it is clear that LargestCreditorFirst strategies (in green and red) yield the highest eventual flow and LargestCreditorLast strategies (in blue and orange) the lowest, for all networks (empirical and random).*

decreasing.

Final state

When combing this with figure 5.5, more becomes apparent. In the top figure, the height of the bars represents the relative eventual size of the economy. They correspond to the height of the lines in figure 5.4 at time $t = 1000$. From this, some observations can be made. First, the LargestCreditorFirst strategies (in green and red), have the highest flow for all networks.

This is explaind by observing figure 5.6 on the facing page, which displays the payments done in the last iteration of the simulation. It is obvious that for the LargestCreditorFirst strategies, the payments are higher than for the LargestCreditorLast strategies (because all the companies first pay their large obligations). The area under this distribution is equal to the height of the corresponing bar in figure 5.5, and because of the skewed distribution in figure 5.6 (note the log-log scale), the total flow is higher.

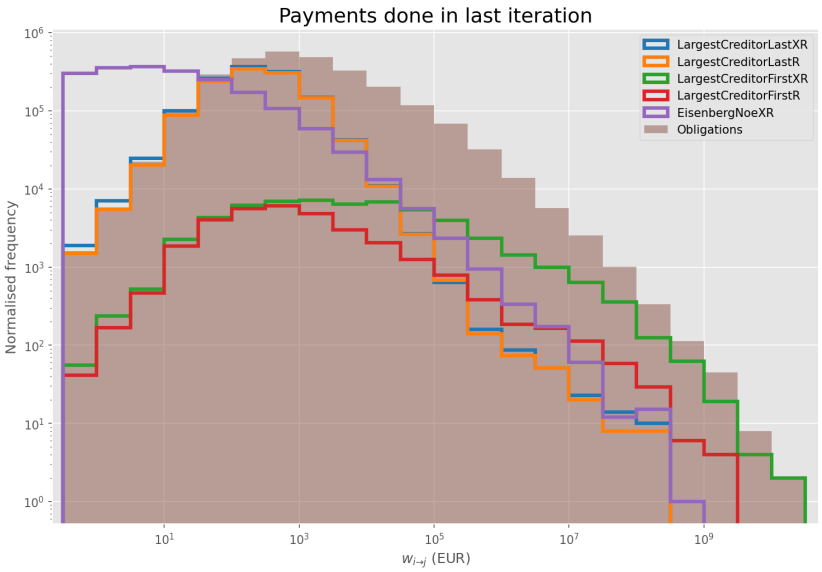


Figure 5.6: *Distribution of the payments (weights of the payment matrix) that are done in the last iteration for the empirical network. The original obligations are also displayed (brown area). The area under the distribution equals the height of the total flow in figure 5.4 on page 43 in the last iteration (and the height of the bars in figure 5.5). The area under the green distribution (LargestCreditorFirstXR) is orders of magnitude higher than that of the other lines (note the log-scales). Note that the LargestCreditorLast payments (blue and orange) are more skewed to the lower payments (because the large obligations are paid last) and the LargestCreditorFirst payments (green and blue) are more skewed to the higher payments, as the nodes first pay the largest obligations. A similar figure for the randomly generated models can be found in appendix figure B.4 on page 67.*

5.2.2 Building reserves when in default

A second important observation from figures 5.4 and 5.5 on page 44 is that the strategies where all the money is paid according to absolute priority (the **XR**-cases) always result in a higher total flow than strategies where nodes are allowed to build reserves when they are in default (in the **R**-case). This can be explained by the following reasoning.

Because of conservation of money, all the money must be either paid to a creditor or saved to a reserve. The only difference between **R** and **XR** is that nodes that are in default can choose to *not* pay a creditor. Therefore, this money is saved and not flowing. When it happens that debtors of these nodes stop paying, this node is not able to save more money and it stops saving, but with some non-zero amount in its reserves. This money will then never flow, which is equivalent to it being removed from the system. Now, since this is the only difference between **R** and **XR**, it is likely that the strategies **XR** will always result in a higher flow, as observed. This effect is amplified in the LargestCreditorFirst strategies, where nodes try to save for a larger obligation, but never quite reach this level. This effect is not present when using networks sampled from the FiCM, because there, in an average realisation, there are fewer nodes with small strengths (see section 6.1 on page 51).

A second point is the difference between the random networks models. In terms of relative differences between strategies, the Stripe Fitness model with 737 labels recreates similar behaviour as observed with the empirical network. The ratio of flows between simulations of the Stripe Fitness model and the empirical network, for the five strategies, is $21.4\% \pm 6.0\%$. This indicates that the simulation on the random networks recreates some of the behaviour (differences between strategies), but not all (total flow). For the FiCM, this ratio is $29.4\% \pm 28.4\%$, which is a much higher variance, indicating that the FiCM recreates the behaviour of the empirical network worse.

When comparing the number of not-defaulted nodes ('healthy' companies), similar results appear. For the Stripe Fitness model, the number of not-defaulted nodes is on average $84.0\% \pm 3.6\%$ compared to the empirical. In the bottom figure, one can see that there is almost no difference in terms of number of not-defaulted nodes between strategies for the FiCM. For all the strategies, the number of not-defaulted nodes is on average $3.80 \cdot 10^4 \pm 7.42 \cdot 10^2$, a relative difference of about 2%.

Version of July 14, 2021– Created July 14, 2021 - 12:35

5.3 Sectors

After the simulation, the sector of the nodes is inspected. Each node is part of a sector (or grouped sector, see section 4.2.2 on page 34) and within these sectors, it is observed *when* the nodes default on average and *how many* nodes in the sector stay healthy during the simulation. Figure 5.7 on the next page shows the results (averages in the sectors) of this analysis.

From this figure, one can observe several points. First, in contrast to the rather naive way of interpreting figures 5.4 and 5.5, the *LargestCreditorFirst* strategies (in blue and orange) seem to produce more favourable results than the *LargestCreditorLast* strategies (in red and green): for *LargestCreditorFirst*, the nodes default on average later in the simulation, and, also confirmed by figure 5.5, there are fewer defaults in general (for all the sectors).

Second, there seems to be a difference between the sectors. For example, the nodes in the sectors starting with 54, i.e. Professional, Scientific, and Technical Services, default on average later than nodes in the sectors starting with 44, i.e. Retail Trade. There are also *more* defaults in the latter sectors. It should be noted, however, that, given the errors and the shortcomings of the simulation as discussed in 6.2 on page 52, this research cannot draw strong conclusions on the risk of the mentioned sectors face in reality.

Version of July 14, 2021– Created July 14, 2021 - 12:35

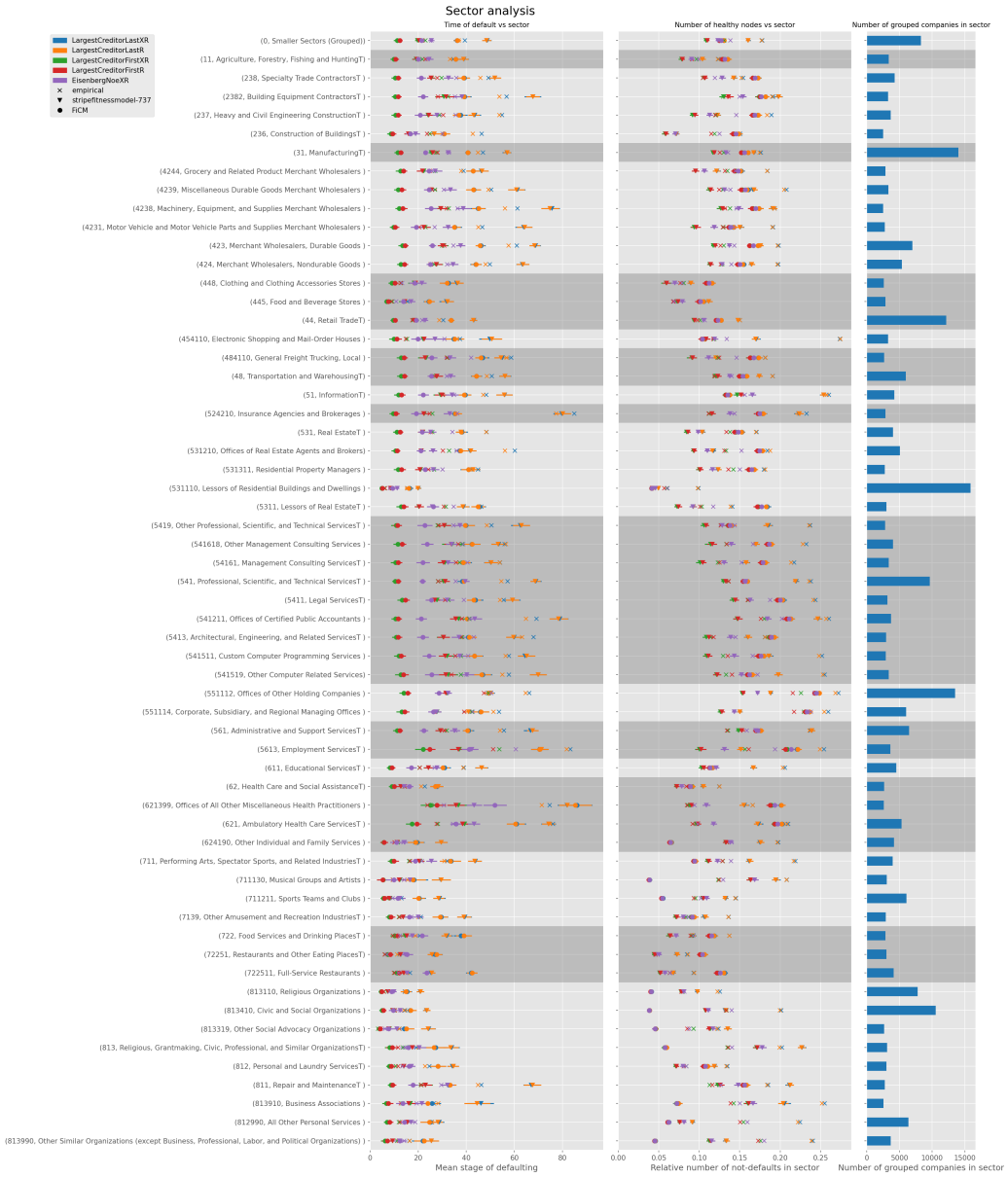


Figure 5.7: *(Caption next page.)*

Figure 5.7: *(Previous page.) The sector analysis of the simulations. The left figure shows when the nodes in that group default, averaged over all nodes in the group. For the random ensembles, 100 networks were sampled and the mean of these 100 data points is plotted, with an error for the standard deviation. The middle figure shows the relative number of nodes that did not default in $t < 1000$, which can be observed directly with only 1 network (the empirical). For the random networks, the relative number of not-defaulted nodes is averaged over the 100 networks, with error bars for the standard deviation. The right plot shows how many companies are in the different groups. The sectors are grouped using NAICS (see section 4.2.2 on page 34) in such a way that every sector contains at least 2500 nodes. Note that this grouping is done separately from the grouping for the Stripe Fitness model. The grey area's in the figure indicate the 2-digit NAICS grouping, which means that neighbouring sectors in the same shade of grey are part of the same broader group.*

Chapter 6

Discussion

In chapter 5, some interesting results were presented. In this chapter, the results are discussed. First, some shortcomings of the random networks are discussed. Second, the simulation is discussed.

6.1 Stripe Fitness model

6.1.1 Low strengths

In the figures presented in section 5.1.3 on page 40 on the strength distributions, two points became clear. First, in an average realisation of the Stripe Fitness model, the number of nodes with a small strength was underestimated, especially for the Stripe Fitness models with a small number of labels. This can be explained by first inspecting the equations that govern the FiCM (equations (4.3) and (4.8) on page 32) (and also the equations of the Stripe Fitness model, equations (4.10) and (4.13) on page 33).

The first equation is increasing in χ and ψ (for $\chi \geq 0, \psi \geq 0$), indicating that links between smaller-strength nodes have a smaller probability of being realised (by construction). This is precisely what is observed in the figures. As explained previously, the average strengths are reproduces by construction. The fact that an average realisation of the random models is not equal in terms of strength distributions could have an effect on the cascading defaults simulations, which are done using individual realisations of the random networks.

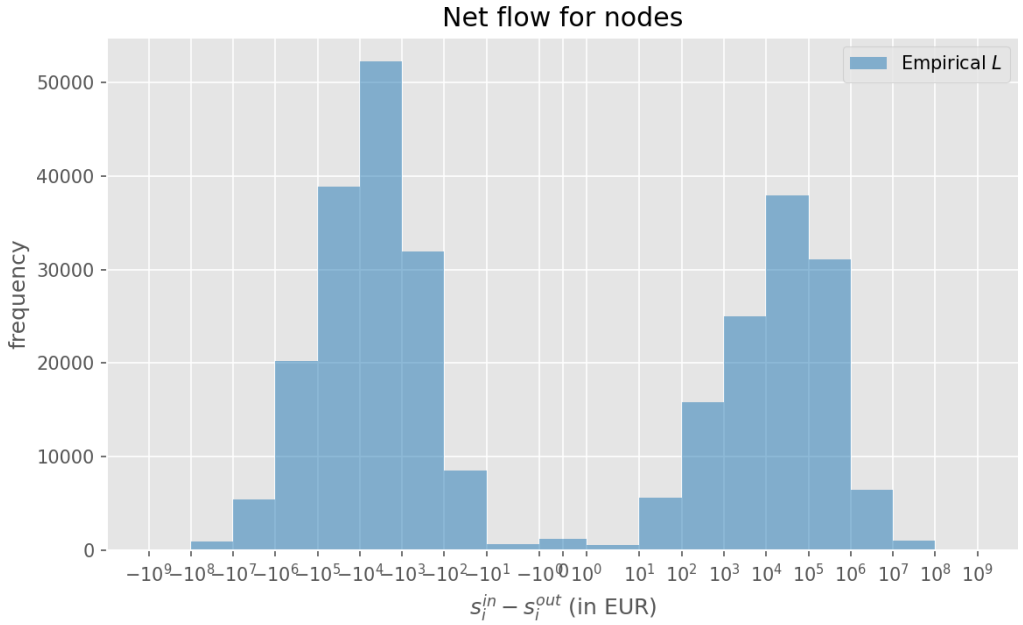


Figure 6.1: *The net flow of the empirical transaction network. Net flow can be calculated by subtracting the out strength from the in strength for all the nodes. From this figure, one can see that many nodes start the simulation in default (i.e.* $s_i^{in} - s_i^{out} < 0$*).*

6.2 Cascading defaults

Imbalanced transactions

Since the transactions from 2019 form the matrix of obligations and thereby the incoming flow in the first iteration, any company that had more outgoing transactions than incoming transactions (after filtering), will default in the first iteration. In other words, the network of obligations is highly unbalanced, see figure 6.1. In reality however, this shouldn't be the case necessarily, as companies may have reserves and/or other (unobserved) accounts.

This imbalance will not result in a 'shock' in the network, as is for example the case in research on risk contagion [17–19], but rather it will cause a spread of money. When every node has exactly as much incoming as outgoing money (i.e. net flow $s_n^{in} - s_n^{out} = 0 \forall n \in \mathcal{N}$), the entire network will relax. This can be checked by inspecting equations (2.2) and (2.3) on page 12.

To overcome this, the nodes could start with a certain 'reserve' r. This

would make it such that the starting capital is no longer $s_n^{in} - s_n^{out}$, but rather $s_n^{in} - s_n^{out} + r$. The value of this reserve could be estimated from account balances or balance sheet information.

Other shortcomings

In the business research thesis on this project [10], more shortcomings of the cascading defaults simulation are discussed, including the calculation of obligations and the exogenous filtered transactions.

6.3 Cascading defaults and random networks

In section 5.2 on page 43, some of the results are already discussed. It is for example discussed that the flow is decreasing for most simulations, that the strategies **XR** (not saving when in default) result in higher flow and that *LargestCreditorFirst* results in higher total flow, but also more defaults. These results are also explained in that section. In this section, figure 5.7 on page 48 is discussed.

6.3.1 Stripe Fitness and FiCM

Recall figure 5.7 on page 48 on the sectors of nodes in the cascading defaults simulations. In the figure for every strategy (e.g. LargestCreditor, EisenbergNoe) and for every sector the mean stage of defaulting and the relative number of defaults is depicted. This is shown for the different networks (empirical, ensemble of Stripe Fitness and ensemble of FiCM). The goal of this section is to explore the differences between the random network models in term of agreement with the results of the empirical model.

In the Stripe Fitness model, sector information is explicitly used to create the random networks. One would therefore expect that the results of the simulation on sector level agree better with the empirical than an ensemble of random networks where only strengths are conserved (FiCM). In the figure (figure 5.7), two properties are measured. First, the mean stage of defaulting for nodes in that sector. Second, the number of nodes that did not default during the entire simulation (divided by the number of nodes in the sector).

In figure 6.2 on the following page, the difference between the mentioned measurements for the models and for the empirical network is depicted for the different ensembles. The mean of the difference indicates a bias and the standard deviation gives a spread. Both models result in error

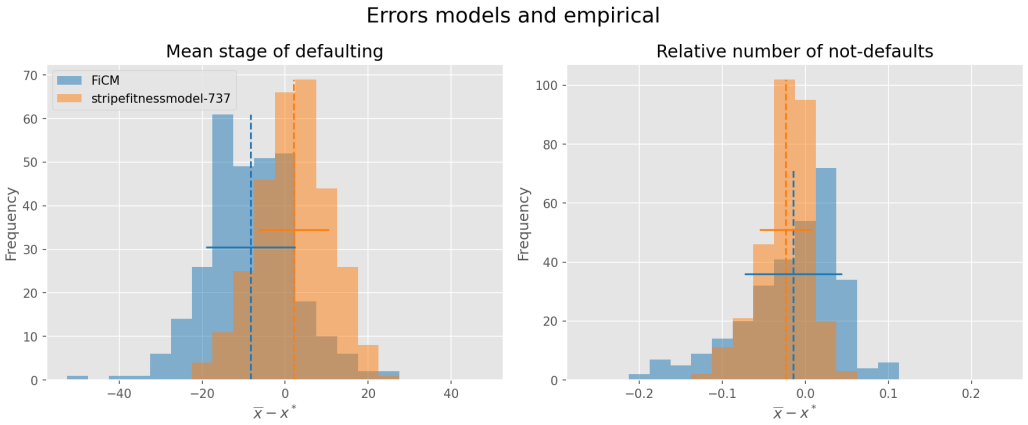


Figure 6.2: *Histograms of the difference between the mean observed values $\overline{x}$ for the random models and the observed values x^* for the empirical network. The vertical lines depict the means of the distribution and the horizontal lines the standard deviation of the distribution. For the FiCM, the mean of the error between stage of defaulting (left figure) is -8.29 ± 10.8 and this is 2.10 ± 8.43 for the Stripe Fitness model with 737 labels. The mean of the error of relative number of defaults is -0.0145 ± 0.0588. For the Stripe Fitness model with 737 labels, this is -0.0237 ± 0.0312. A similar figure where the different strategies are not grouped can be found in figure B.5 on page 68.*

that have 0 within 1σ, indicating that in general the results are replicated. It should be noted that the variance is smaller in the Stripe Fitness model than the FiCM. A possible interpretation is that the Stripe Fitness model recreates the behaviour observed on the sector level better than the FiCM. However, this conclusion is not statistically rigorous.

Chapter 7

Conclusion

Statistical physics can be used in order to analyse the structure and dynamics of networks. Specifically for financial systems, where the underlying network is usually not fully known, statistical ensembles can be used to sample random networks that share specific properties with some empirical network. This research applies methods from statistical physics to focus on two topics: cascading defaults and random network generation. In the sections on random networks, the Fitness-induced Configuration Model (FiCM, [8]) and the Stripe Fitness model [9] were used to create an ensemble of random networks from which can be sampled. The ensembles were created such that the ensemble samples would recreate the number of links and the individual (out and in) strengths of nodes on average.

The research confirmed that these measures were indeed reconstructed on average, but not in individual samples from the ensemble. When using more labels for the Stripe Fitness model, the individual strength distributions coincide more with the empirical. In general, the Stripe Fitness model reproduces the strengths better than the FiCM. The degree distributions were not realised exactly, also not in the ensemble average. The general shape of the distribution was reproduced, but certainly not the exact degree distribution (as hypothesised by the Fitness ansatz [15]).

This indicates that the out and in strengths as used in the FiCM are not a perfect fitness (i.e. proxy for the Lagrange multiplier controlling the degree). Using sector strengths as used in the Stripe Fitness model and a parameter z per group results is a degree distributions that is more in line with the empirical network.

From the simulations, some conclusions were drawn in the internship report [10]. First, the strategies LargestCreditorFirst produce the highest flow, which is explained by the fact that on average, larger obligations

are paid. The LargestCreditorLast strategy results in more companies that don't default in the simulation. This results in a lower total flow because the obligations that are paid are the small obligations.

These results are confirmed by the random networks sampled from the Stripe Fitness model. The FiCM only reproduces the observed behaviour regarding the total flow. The number of not-defaulting nodes is almost the same for all different strategies when using networks sampled from the FiCM.

With this research, the start of a representative simulation of cascading defaults is made. By the Stripe Fitness model, it is confirmed that using sector information and strengths more realistic networks can be sampled than by example the FiCM.

Version of July 14, 2021– Created July 14, 2021 - 12:35

Bibliography

[1] L. Eisenberg and T. H. Noe, *Systemic Risk in Financial Systems*, Management Science **47**, 236 (2001).

[2] M. Hazama and I. Uesugi, *Measuring the systemic risk in interfirm transaction networks*, Journal of Economic Behavior and Organization **137**, 259 (2017).

[3] T. Squartini and D. Garlaschelli, *Analytical maximum-likelihood method to detect patterns in real networks*, New Journal of Physics **13** (2011).

[4] M. E. Newman, *The structure and function of complex networks*, SIAM Review **45**, 167 (2003).

[5] T. Squartini, R. Mastrandrea, and D. Garlaschelli, *Unbiased sampling of network ensembles*, New Journal of Physics **17** (2015).

[6] R. Mastrandrea, T. Squartini, G. Fagiolo, and D. Garlaschelli, *Enhanced reconstruction of weighted networks from strengths and degrees*, New Journal of Physics **16** (2014).

[7] J. Park and M. E. Newman, *Statistical mechanics of networks*, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **70**, 13 (2004).

[8] G. Cimini, T. Squartini, D. Garlaschelli, and A. Gabrielli, *Systemic Risk Analysis on Reconstructed Economic and Financial Networks*, Scientific Reports **5**, 1 (2015).

[9] L. Ialongo and D. Garlaschelli, *Stripe Fitness model*, unpublished.

[10] C. Q. de Valk, *Cascading defaults on ING transaction network*, Master's thesis, Leiden University, 2021, Internship report Business Studies specialisation.

[11] E. T. Jaynes, *Information Theory and Statistical Mechanics*, Phys. Rev. **106**, 620 (1957).

[12] M. Á. Serrano and M. Boguñá, *Weighted configuration model*, AIP Conference Proceedings **776**, 101 (2005).

[13] F. Parisi, T. Squartini, and D. Garlaschelli, *A faster horse on a safer trail: Generalized inference for the efficient reconstruction of weighted networks*, New Journal of Physics **22** (2020).

[14] P. Erdős and A. Rényi, *On Random Graphs I*, Publicationes Mathematicae Debrecen **6**, 290 (1959).

[15] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Muñoz, *Scale-Free Networks from Varying Vertex Intrinsic Fitness*, Physical Review Letters **89**, 1 (2002).

[16] U.S. Office of Management and Budget, *North American Classification System (NAICS)*, page 963 (2017).

[17] P. Barucca, M. Bardoscia, F. Caccioli, M. D'Errico, G. Visentin, G. Caldarelli, and S. Battiston, *Network valuation in financial systems*, Mathematical Finance **30**, 1181 (2020).

[18] S. J. Wells, *Financial Interlinkages in the United Kingdom's Interbank Market and the Risk of Contagion*, SSRN Electronic Journal (2005).

[19] E. Nier, J. Yang, T. Yorulmazer, and A. Alentorn, *Network models and financial stability*, Journal of Economic Dynamics and Control **31**, 2033 (2007).

Appendix A

Derivations

A.1 Free energy relation

Given a function $f(x_1, x_2, \dots) = \ln(y(x_1, x_2, \dots))$, one can see that the partial derivative w.r.t. x_i is

$$\frac{\partial f(x_1, x_2, \dots)}{\partial x_i} = \frac{df}{dy}\frac{\partial y(\vec{x})}{\partial x_i}. \tag{A.1}$$

Now, since $\frac{df}{dy} = \frac{1}{y(\vec{x})}$,

$$\frac{\partial f(\vec{x})}{\partial x_i} = \frac{1}{y(\vec{x})}\frac{\partial y(\vec{x})}{\partial x_i}. \tag{A.2}$$

Now by renaming $\vec{x} \equiv \vec{\theta}$ the parameters, $y \equiv Z(\vec{\theta})$ the partition function and $f \equiv -\Omega(\vec{\theta})$ the free energy as defined by equation (2.18) on page 15, the relation becomes

$$\frac{\partial \Omega(\vec{\theta})}{\partial \theta_\alpha} = -\frac{1}{Z(\vec{\theta})}\frac{\partial Z(\vec{\theta})}{\partial \theta_\alpha}, \tag{A.3}$$

which is the right part of equation (2.17). Now by equations (2.15) and (2.16),

$$\begin{aligned}\frac{\partial Z(\vec{\theta})}{\partial \theta_\alpha} = \frac{\partial}{\partial \theta_\alpha}\Big(\sum_{G\in\mathcal{G}} e^{-\sum_{\beta=1}^{m}\theta_\beta x_\beta(G)}\Big) &= -\sum_{G\in\mathcal{G}} x_\alpha(G) e^{-\sum_{\beta=1}^{m}\theta_\beta x_\beta(G)} \\ &= -\sum_{G\in\mathcal{G}} x_\alpha(G) e^{-H(G,\vec{\theta})},\end{aligned} \tag{A.4}$$

and therefore equation (A.3) is

$$\frac{\partial\Omega(\vec{\theta})}{\partial\theta_\alpha} = \frac{1}{Z(\vec{\theta})}\Big(\sum_{G\in\mathcal{G}} x_\alpha(G)e^{-H(G,\vec{\theta})}\Big). \tag{A.5}$$

Here, equation (2.14) for $\mathbb{P}$ can be recognised and

$$\frac{\partial\Omega(\vec{\theta})}{\partial\theta_\alpha} = \sum_{G\in\mathcal{G}} x_\alpha \mathbb{P}(G|\vec{\theta}) = \langle x_\alpha\rangle, \tag{A.6}$$

where in the last step, the definition of $\langle x_\alpha\rangle$ (equation (2.12)) is used. Hereby,

$$\langle x_\alpha\rangle = -\frac{1}{Z(\vec{\theta})}\frac{\partial Z(\vec{\theta})}{\partial\theta_\alpha} = \frac{\partial\Omega(\vec{\theta})}{\partial\theta_\alpha}, \tag{A.7}$$

which is equation (2.17) on page 15.

A.2 MaxEnt

Recall equations (2.35), (2.36) and (2.37) on page 18. Analogous to the binary case described in subsection 2.2.1 The Maximum Entropy Principle on page 14, the entropy and constraints result in a Lagrangian

$$\mathfrak{L}(w_{i\to j},\vec{\pi},\vec{v}) \equiv S + \sum_i \pi_i(\sum_j w_{i\to j} - (s_i^{out})^*) + \sum_i v_i(\sum_j w_{j\to i} - (s_i^{in})^*), \tag{A.8}$$

which must be maximised, i.e.

$$\frac{\partial\mathfrak{L}}{\partial w_{i\to j}} = -1 - \ln w_{i\to j} + \pi_i + v_j \stackrel{!}{=} 0, \forall i,j \tag{A.9}$$

$$\frac{\partial\mathfrak{L}}{\partial\pi_i} = \sum_j w_{i\to j} - (s_i^{out})^* \stackrel{!}{=} 0, \forall i \tag{A.10}$$

$$\frac{\partial\mathfrak{L}}{\partial v_i} = \sum_j w_{j\to i} - (s_i^{in})^* \stackrel{!}{=} 0, \forall i. \tag{A.11}$$

From equation (A.9), it follows that

$$w_{i\to j} = e^{\pi_i + v_j - 1}, \tag{A.12}$$

which can be used in equations (A.10) and (A.11) to find that

$$e^{\pi_i} = \frac{(s_i^{out})^*}{\sum_j e^{v_j - 1}} \tag{A.13}$$

$$e^{v_j} = \frac{(s_i^{in})^*}{\sum_i e^{\pi_i - 1}}. \tag{A.14}$$

Now, by rearranging (A.12), it can be found that

$$w_{i\to j} = \frac{(s_i^{out})^*(s_i^{in})^*}{(\sum_j e^{v_j-1})(\sum_i e^{\pi_i-1})e} = \frac{(s_i^{out})^*(s_i^{in})^*}{\sum_{i,j} e^{\pi_i+v_j-1}} = \frac{(s_i^{out})^*(s_i^{in})^*}{W}, \tag{A.15}$$

where W is defined to be

$$W \equiv \sum_{i,j} w_{i\to j}, \tag{A.16}$$

which is by construction equal to

$$W = \sum_i (s_i^{out})^* = \sum_i (s_i^{in})^*. \tag{A.17}$$

Version of July 14, 2021– Created July 14, 2021 - 12:35

Appendix B

Supplementary results

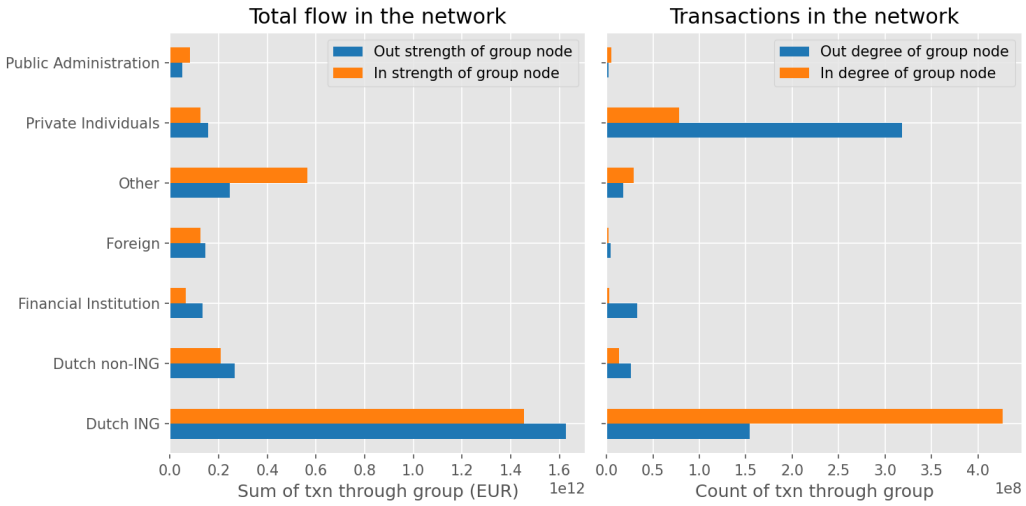


Figure B.1: *[Figure from the internship report [10]].The figure shows what the strengths are of the groups node. This figure is similar to figure 3.1 on page 24, which displays the strength from ING companies. For all the bars, except the bar 'Dutch ING', the numbers are therefore the same. In this figure, the Dutch ING bars contain both exogenous → Dutch ING and Dutch ING → Dutch ING (and the other way around).*

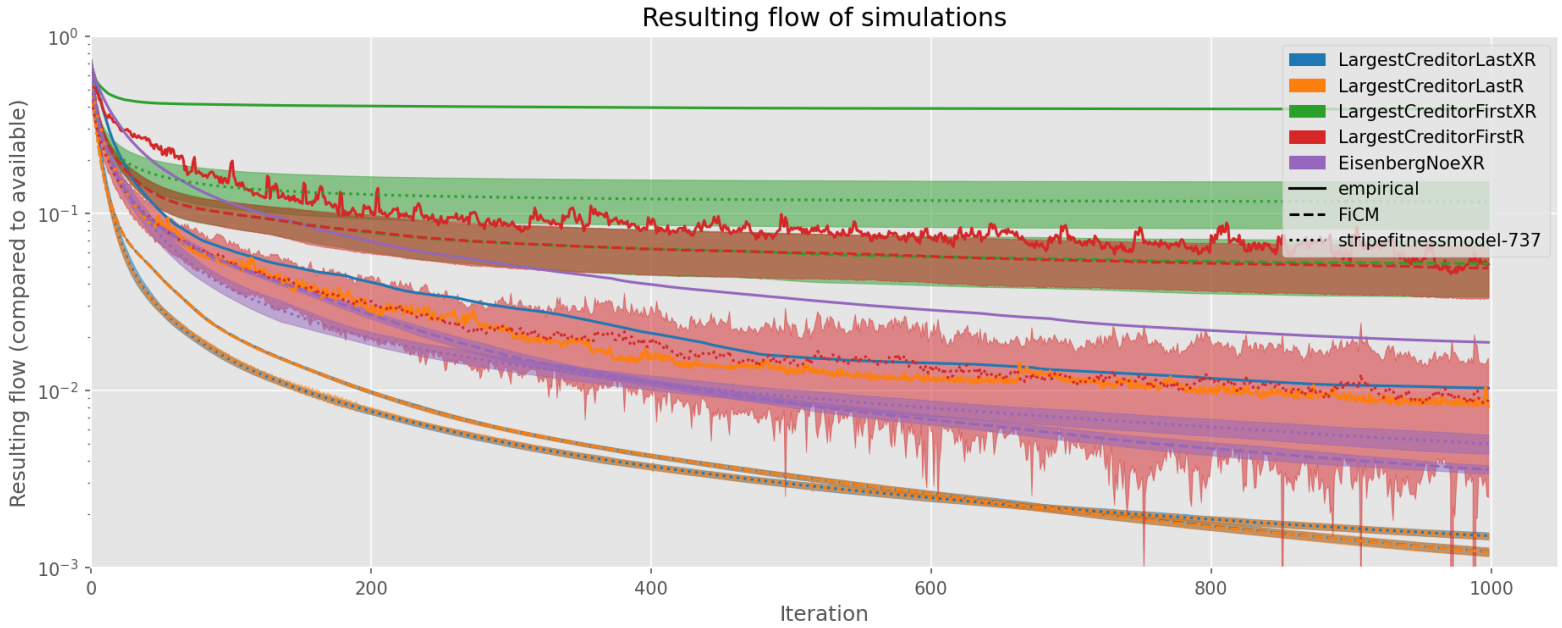
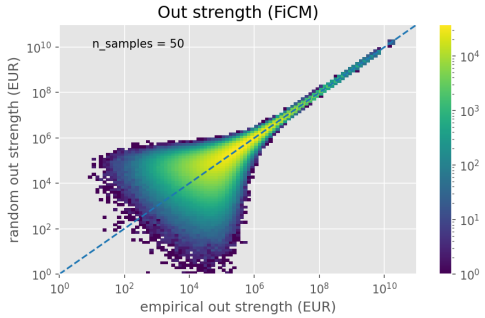
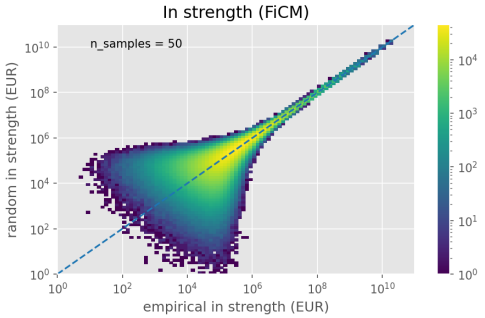


Figure B.2: *(Full size display of figure 5.4 on page 43). The flow of money in time, depicted for different defaulting strategies (colors) when the simulation is ran for different networks (linestyles). 100 random networks are sampled from the ensemble and the flow is recorded at each time. For that time, the mean is depicted with the line and the area is $\pm 1\sigma_t$ with σ_t the standard deviation of the flow of the ensemble at that time.*

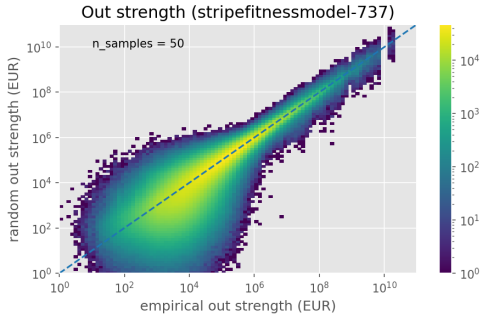
65



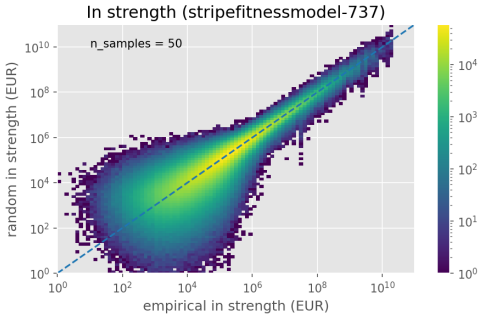
(a) *Comparison between empirical out strengths and the mean modelled out strengths, using the FiCM.*



(b) *Comparison between empirical in strengths and the mean modelled in strengths, using the FiCM.*



(c) *Comparison between empirical out strengths and the mean modelled out strengths, using the Stripe Fitness model with 737 labels.*



(d) *Comparison between empirical in strengths and the mean modelled in strengths, using the Stripe Fitness model with 737 labels.*

Figure B.3: *In the figures, the empirical strengths are plotted against the sampled strength. The Stripe Fitness model with 737 labels and the FiCM, which is just a Stripe Fitness model with 1 label, are used to show the result. The results of the intermediate Stripe Fitness models (60, 204 and 563 labels) gradually shift in shape from the FiCM to the Stripe Fitness 737. The figure with mean sampled strengths can be found in figure 5.3 on page 42.*

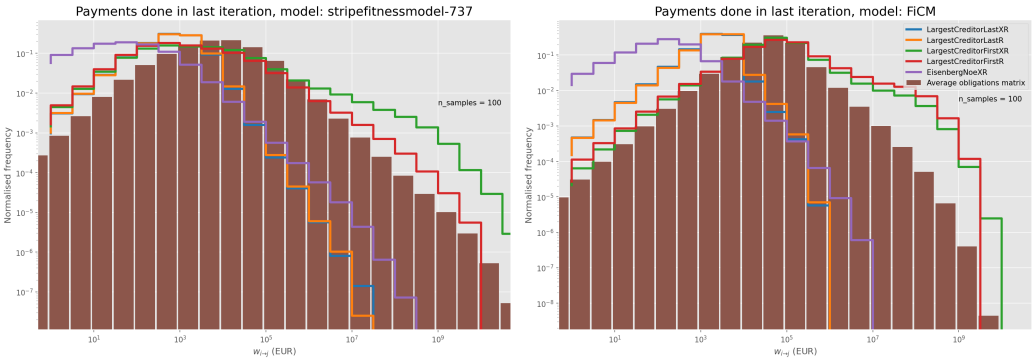


Figure B.4: *[Figure from the internship report [10]]. Similar figure as figure 5.6 on page 45, but for random networks. The distribution of the payments (weights of the payment matrix) that are done in the last iteration is displayed. The height of the bars is normalised so they add up to 1. For both random network models, the LargestCreditorLast payments (blue and orange) are generally lower than the payments with LargestCreditorFirst. This is due to the fact that nodes pay the largest obligations last or first, respectively.*

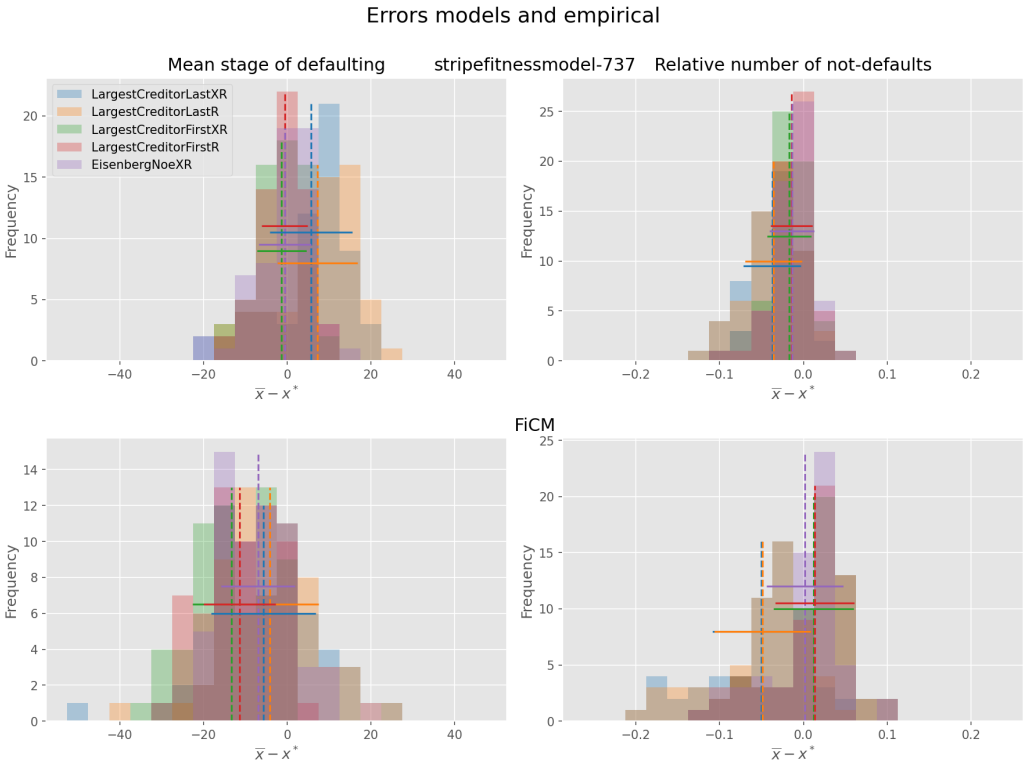


Figure B.5: *An extension of figure 6.2 on page 54. Histograms of the difference between the mean observed values $\overline{x}$ for the random models and the observed values x^* for the different strategies and the different models (top stripe fitness model, bottom FiCM). The vertical lines depict the means of the distributions and the horizontal lines depict the standard deviations of the distributions.*