[1]BALANCE SHEET ANALYSIS OF CREDIT AND DEBT NETWORKS

PAULO GARRIDO

Escola de Engenharia, Universidade do Minho
4800 Guimarães, Portugal
pgarrido@dei.uminho.pt

PEDRO CAMPOS

LIAAD INESC TEC and Faculdade de Economia
Universidade do Porto, Rua Dr. Roberto Frias
4200-465 Porto, Portugal
pcampos@fep.up.pt

ANDRÉ DIAS

LIAAD INESC TEC and Faculdade de Economia
Universidade do Porto, Rua Dr. Roberto Frias
4200-465 Porto, Portugal
diasandre@gmail.com

In this paper, we present a computing procedure to analyze a network of credit and debt among agents (nodes) from a standpoint of balance sheet dependencies. The aim is to develop a method to assess thoroughly the sensitivity of the network to potential individual insolvencies. For this purpose, given a state of the network, the insolvency of an agent is assumed and the cascade of provoked insolvencies is simulated. Exploring the matrix definition of the network, this is made systematically for all agents. Therefore, in only one run of the procedure, all the possible trajectories of insolvencies, each beginning in a different agent, are calculated. This allows spotting at a glance which agents are "systemically riskier". Determination of adequate capital levels can be made on a case basis by running the procedure repeatedly. This work contributes with two new aspects to the existing literature. First, given a known situation of a credit and debt network, a computing procedure is presented that allows to assess the network sensitivity to the exogenous insolvency of any of its nodes. Second, continued monitoring of a credit and debt network is computationally feasible. This "proof of concept" software can be extended into a tool useful for research and regulation, if the relevant information is made available.

Keywords: credit and debt networks, balance-sheet insolvency, insolvency propagation, banking networks, systemic risk, financial crisis, Julia programming.

1 Introduction

Financial crises and bank failures have been topics of interest for many researchers [9, 10, 18]. The crisis of 2008 prompted for scientific developments in the field, increasing

---

[1] *Electronic version of an article published as* [Advances in Complex Systems, 18, 5-6, 2015, 1550025] [DOI: 10.1142/S0219525915500253] © [copyright World Scientific Publishing Company] [https://www.worldscientific.com/worldscinet/acs]

the contributions and new approaches on both subjects. In a previous work [15], we proposed a model, based on the agents' paradigm, for an inter-banking market network. Our main goal was to assess how the structure of credits and debts, assumed a scale-free network [19, 28], would behave in the presence of credit and liquidity shocks [11] producing bank failures.

The study of inter-banking markets is of great practical importance and theoretically rich [22]. However, one can generalize from such inter-banking markets to abstract networks of credits and debts. In such a generalization, each node or agent in the network can represent an institution of any kind one may be interested in studying as a bank, a firm or even a state. For an example, Battiston *et al.* present in [6] a study of a production network where insolvencies arise because of delayed payments and supply failure costs.

In a generalized credit and debt network, each agent is in debt and (or) in credit with respect to some other agents. Solvency or insolvency of an agent can be classified as cash-flow insolvency or balance-sheet insolvency [30]. The first arrives when an agent does not have a form of payment to settle a debt that falls due. The second happens when the value of an agent's assets becomes less than the value of its liabilities. The two types of insolvency need not happen simultaneously. The work reported in this paper refers to the propagation of balance-sheet insolvency in a credit and debt network.

The situation initially assumed for the network is all agents being solvent in a balance-sheet or accounting sense, that is, their equity, or assets minus liabilities, is positive or zero. Then, one supposes that for some reason external to the network an agent loses assets to the point that its equity becomes negative. Consequently, the agent becomes insolvent –in the balance-sheet sense considered – and agents in the network having credits over the insolvent agent lose a fraction (or all) of these assets and eventually also become insolvent. Questions arise on the susceptibility of the network to the propagation of insolvencies and on which capital levels and other regulations can prevent or lower the probability of such propagation.

The approach most described in the literature to solve the problem consists in establishing a connection between properties of the network and some measure of its susceptibility to the propagation of insolvencies. References [5], [7], [8] and [29] are examples of it. We propose to supplement this approach with a more direct one. Exploring the fact that a network can have a matrix representation, we describe here a procedure to compute all the possible sequences of insolvency, each sequence beginning at an agent or node. Given that a sequence has been calculated, one can also compute the number of nodes in it and the associated equity or capital losses. This means that by running the procedure one can associate to each state of the network the set of conceivable sequences of insolvency and losses that the state is hypothetically prone.

For simplicity of analysis we assume that a sequence of insolvency begins in one and only one node. Yet, the simulation of sequences of insolvency with several agents defaulting simultaneously at the initial instant just requires a different initialization of data. So, in fact, the procedure can be used to assess the effects of joint initial defaults.

We are aware that the proposed approach has a character of brute force. However, we believe that such an approach is a rational one for this problem, based on two arguments. On one hand, the propagation of insolvencies follows a (strong) non-linear transition rule; therefore, we can suspect that the dynamical process of propagation will show little predictability from knowledge of the network state. Given these properties, we expect that this procedure will become useful in the work of non-linear analysis of credit and debt networks, by providing fast and complete maps of what can happen. On the other hand, determining all possible trajectories of insolvency propagation and associated losses appears to be a feasible computational task for interesting numbers of nodes in a network. This, besides making it a practical analytical tool, raises the question of developing a procedure-based system to monitor actual credit and debt networks.

Use of the procedure beyond the basic one is possible. For example, the procedure can be run with different levels of capital assigned to each agent. Running the procedure assuming that all the agents have zero capital corresponds to the most dangerous situation. Increasing the capital levels step by step generates increasingly safer situations up to the point where no more insolvency propagation happens. This use of the procedure would allow a regulator to determine which capital levels would be adequate for a network on a near real-time fashion. One could envisage other uses as the theoretical study of network properties. Therefore, we believe that we are contributing to the subject that together with existing theory may become a useful tool.

The plan of the paper is as follows. In the next section, we present a literature review. Most of the work on credit and debt networks centers on banks and the interbanking market, yet it is clearly relevant to the more abstract approach taken here. Section three describes the representation of a credit and debt network in terms of matrices and some properties of such network. In section four, we present the computational procedure used to calculate all the possible insolvency sequences. Section five discusses implementation with a focus on execution time varying with programming language, parameterizations of the network and fraction of recoverable assets. Section six concludes depicting some future work.

2 Literature Review

Literature on financial crises supplies context and a departure point for the analysis of credit and debt networks, from which we have already pointed some important references. Economists as Friedman and Schwartz [18], Bernanke [9], Bernanke and Gertler [10] produced research relating bank failures and financial crises. Study of the Great Depression (GD) was a main concern of theirs, as avoiding a repetition of the GD became a central goal of policy. Between 1929 and 1933, the United States of America saw the number of banks decline from over 25,000 to less than 15,000 [18]. The 2008 crisis showed a relationship between bank failures and financial crises similar to the GD. It is commonly referred that the 2008 crisis was triggered by the failure of the

Lehman Brothers and Bear Stearns investment banks. The reduction in confidence originated a severe shrinking of bank lending, affecting all industries including banking itself [23] and leading to a strong decrease in economic activity.

Analysis of a bank's balance sheet is the starting point in the simulation of banking networks [12, 13]. The main inspiration for many authors is the model of Allen and Gale [2], where one of the first models of a banking network was presented and the conclusion that systemic resilience depends on the structure of relationships was drawn. Further studies on network contagion have been produced. Some focus on the asset side of the balance sheet – credit shocks – like [27] who studied the resilience of a network in case of a general loss of assets among all the banks in the system. Also, [19] studies the ability of a central bank to maintain stability.

Other studies focus on the liability side of the balance sheet – liquidity shocks – like [14] who discusses the main function of banks as being to transform deposit maturities. Liquidity shocks happen when the funding sources of banks are at risk. Market stress levels or fire sales have also been one topic of interest [1]. They happen when there are liquidity shortages and a big amount of the same type of asset is put for sale [21]. Another topic worth mentioning is the growth of complex financial instruments in the form of derivatives. It has been pointed by authors as [26] that their complexity was one of the reasons why so many players were unaware of the possible devastating consequences of a default. A recent study of the systemic risk in Europe is given in [16].

The general study of information propagation in social networks has attracted much attention, a landmark paper on the subject being [24]. Insolvency propagation can be understood as a particular case of constrained information propagation.

The non-linear character of insolvency propagation in credit and debt networks can give rise to somewhat surprising results. In [7] and [8], Battiston *et al.* observe that diversifying risk can, counter-intuitively, lead to increased systemic risk. The more recent papers [5], Battiston and Caldarelli, and [29], Roukny *et al.*, summarize many results of applying network analysis to the issue of financial stability. They point that in order to better control systemic risk contagion, besides creating sufficient capital ratios, it is necessary to increase monitoring of the interrelated dimensions of interconnectedness, complexity and correlation all together, including more detailed information on who owes who. In fact, as we hope to show, having such detailed information would allow assessing the stability properties of a credit and debt network in real-time.

3 Representing a Credit and Debt Network as a Matrix

It is well known that a directed or undirected graph can be equivalently represented as a square matrix with as many lines as nodes of the graph. If the i, j entry of the matrix is non-null, it represents the (i, j) edge from node i to node j. If the graph is undirected then the (i, j) edge equals the (j, i) edge and both entries must be equal.

A network of debt (credit) can be represented by a simple, directed graph, or a matrix, if one takes in account between two entities in the network one relation of the form:

$$\text{“}x \text{ owes } y \text{ the amount of } -X\text{” or owes } (x, y, -X) \tag{1}$$

The relation can be represented as an arrow from x to y with an associated non-null value of $-X$. In a matrix, it would be entered as the $-X$ value at the (x, y) cell.

A debt relation from a to b is also a credit relation from b to a:

$$\text{owes}(x, y, -X) \Leftrightarrow \text{has\_credit}(y, x, X) \tag{2}$$

In a credit matrix collecting the credits of the agents in the network, the value X would be entered at the (y, x) cell.

Let us assume that among five agents, a, b, c, d and e, the following instances of the relation hold at a given moment:

owes(a, c, –66), owes(a, d, –98), owes(a, e, –53),
owes(b, c, –69), owes(b, e, –67),
owes(c, a, –97), owes(c, d, –83),
owes(d, b, –89), owes(d, e, –65),
owes(e, a, –69), owes(e, b, –68),

This can be seen as a network of credit and debt among the four agents. The debt matrix is as follows:

$$D = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & 0 & -66 & -98 & -53 \\ 0 & 0 & -69 & 0 & -67 \\ -97 & 0 & 0 & -83 & 0 \\ 0 & -89 & 0 & 0 & -65 \\ -69 & -68 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

We observe that each entry in the debt matrix would appear in a balance sheet as a liability. In the case of a bank, it could be a loan of reserves from another bank.

The credit matrix corresponding to (3) is as follows:

$$C = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & 0 & 97 & 0 & 69 \\ 0 & 0 & 0 & 89 & 68 \\ 66 & 69 & 0 & 0 & 0 \\ 98 & 0 & 83 & 0 & 0 \\ 53 & 67 & 0 & 65 & 0 \end{bmatrix} \tag{4}$$

We observe that each entry in a credit matrix appears in a balance sheet as an asset. In the case of a bank, it could be a loan to another bank.

In all generality, we have that the following equalities hold for credit and debt matrices:

$$\begin{aligned} C &= -D^T \\ D &= -C^T \end{aligned} \tag{5}$$

The above definitions of the debt and credit matrices allow that two agents simultaneously lend to and borrow from each other; in the example, this happens for the pairs (*a, c*), (*a, e*) and (*b, e*). Summing matrices *C* and *D*, one gets what we call the net position matrix *NP* of agents:

$$NP = C + D = C - C^T = D - D^T \tag{6}$$

We observe that such matrix must be skew-symmetric. The example would yield:

$$NP = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & 0 & 31 & -98 & 16 \\ 0 & 0 & -69 & 89 & 1 \\ -31 & 69 & 0 & -83 & 0 \\ 98 & -89 & 83 & 0 & -65 \\ -16 & -1 & 0 & 65 & 0 \end{bmatrix} \tag{7}$$

The matrix *NP* summarizes all the net credits and debts among agents. In Figure 1, we represent a corresponding network. To each arrow we associate a pair of symmetrical numbers: the first one, positive, is the credit held by the agent or node that the arrow points to; the second one, negative, is the debt of the agent or node that the arrow departs from.

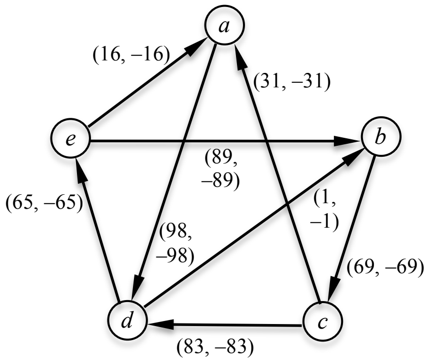

Fig. 1- Graph representation of the credit-debt network with net position matrix in (7).

With the representation above, the sum of assets and liabilities or net worth *with respect to the network* of each agent equals the sum of the corresponding row in the *NP* matrix. Assuming that the net worth of all agents is represented by a vector ***nw*** and the existence of an operation 'sum_rows', we would get for the example:

$$\boldsymbol{nw} = \text{sum\_rows}(NP) = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} -51 \\ 21 \\ -45 \\ 27 \\ 48 \end{bmatrix} \tag{8}$$

If we assume now that the *out of the network* equities of the agents is registered in a vector ***on***, it follows that the accounting state, ***as***, of all agents is given by:

$$\boldsymbol{as} = \text{sum\_rows}(NP) + \boldsymbol{on} = \boldsymbol{nw} + \boldsymbol{on} \tag{9}$$

Let us assume that the out of the network equities or capital positions of the agents in the example are given by ***on*** = $[51, -21, 45, -27, -48]^T$. Then, we would have:

$$\boldsymbol{as} = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} -51 \\ 21 \\ -45 \\ 27 \\ 48 \end{bmatrix} + \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 51 \\ -21 \\ 45 \\ -27 \\ -48 \end{bmatrix} = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

An agent will be insolvent if its ***as*** value is negative and solvent otherwise. As it stands out, if one assumes that the state of a credit and debt network is given by the pair (*NP*, ***on***), then determining the accounting situation of all the agents or nodes amounts to the computation indicated in (9). However, there is an aspect that deserves clarification.

Over a credit-debt network, the sum of assets must equal the symmetric of the sum of liabilities. Let the network have I agents or nodes, $A(i)$ and $L(i)$ be the assets and liabilities of agent or node i. Summing over the I nodes yields:

$$\sum_{i=1}^{I} A(i) = -\sum_{i=1}^{I} L(i) \tag{11}$$

This must be so because any single asset of an agent i must conversely be a liability for an agent j and a single liability of an agent i must conversely be an asset for an agent j.

Now, let us assume that the network completely represents the credit and debt relations of all agents or, equivalently, no credit or debt relation of any agents fails to be represented in the network. If all agents in the network are solvent, or have non-negative equity, $\boldsymbol{nw}(i) = A(i) - L(i) \geq 0$, it follows that equity must be zero for all agents:

$$\sum_{i=1}^{I} A(i) = -\sum_{i=1}^{I} L(i) \rightarrow \left(\forall i, A(i) - L(i) \geq 0 \rightarrow \forall i, A(i) - L(i) = 0\right) \tag{12}$$

Another way to see this is to observe that the equality of absolute values of total assets and total liabilities implies that if an agent has positive equity another one must have negative equity:

$$\sum_{i=1}^{I} A(i) = -\sum_{i=1}^{I} L(i) \rightarrow \left(\exists i, A(i) - L(i) > 0 \rightarrow \exists j, A(j) - L(j) < 0\right) \tag{13}$$

In this study, we want to consider departing situations where all the agents are solvent and some may have positive equity – as in fact agents do. This implies that one assumes that the network represents the agents only partially, i.e., it represents completely the credit-debt relations among agents, but not all the relations that the agents may have. The vector ***on*** referred above stores the values of external or out of the network equity. One observes that for the accounting state of all agents to be zero, ***on*** must be the symmetrical of ***nw***, as in the example (10).

4 Calculating all the Possible Sequences of Insolvency Propagation

We now ask: given a state of the network (*NP*, ***on***), can we assess the associated risk? We propose to answer the question by simply calculating all the possible sequences of insolvency propagation that can begin with each agent *i* becoming insolvent because its ***on***(*i*) value changes so that the sum ***as***(*i*) = ***nw***(*i*) + ***on***(*i*) becomes negative.

4.1 The accounting state of agents

The computing procedure that we present calculates the accounting state of each agent *i* in discrete time *k* as ***as***(*i*, *k*) At instant $k = 1$, all agents are supposed to be solvent, that is $\forall i, \boldsymbol{as}(i,1) \geq 0$. At instant $k = 2$, one agent *j* is supposed to become insolvent by an exogenous change in ***on***(*j*), that is $\exists ! i = j, \boldsymbol{as}(j,2) < 0$.

Beginning in instant $k = 3$, first the *NP* matrix is updated to reflect the changes in assets and liabilities resulting from the insolvencies occurring at $k - 1$. Second, the accounting state of agents that were solvent at $k - 1$ is recalculated according to Eq. (9). More precisely, assuming that the *NP* matrix is now understood as a function of time *NP*(*k*):

$$\boldsymbol{as}(i,k-1) \geq 0 \rightarrow \boldsymbol{as}(i,k) = \text{sum\_row}\left(i, NP(k)\right) + \boldsymbol{on}(i) \tag{14}$$

The accounting state of agents that were insolvent at $k - 1$ is kept unchanged. Calculations are repeated until no more insolvencies show up.

It is useful to consider the out of network equity values ***on***(*i*) as the sum of two parcels:

$$\boldsymbol{on}(i) = \boldsymbol{oe}(i) + \boldsymbol{p}(i) \tag{15}$$

We define $\boldsymbol{oe}(i)$, offset equity, to be the symmetrical of the net worth $\boldsymbol{nw}(i)$ of agent i at instant $k = 1$:

$$\boldsymbol{oe}(i) = -\text{sum\_row}\left(i, NP(1)\right) \tag{16}$$

We name $\boldsymbol{p}(i)$ the "initial capital" of agent i because the value of $\boldsymbol{p}(i)$ is the amount of "cushioning" against insolvency that agent i disposes if it loses net worth as a result of other agents becoming insolvent.

4.2 Reassignment of assets after insolvency

We must clarify which assumption is to be used regarding what happens to assets and liabilities of other agents when an agent becomes insolvent or how the matrix $NP(k)$ is going to be updated. We assume that if an agent i becomes insolvent at instant k then its creditors can recuperate only a fraction α of the assets owned by i at instant $k-1$, with $\alpha \in [0, 1]$; its debtor agents experience a corresponding gain of liabilities, if any.

If an agent i becomes insolvent then the representing node in the network is deleted together with the arrows coming from and going to the node. In terms of the matrix $NP(k)$, this translates in zeroing all non-zero entries in the ith row and the ith column. According to the value of α considered, other arrows (or the corresponding entries in the matrix NP) may need to be changed.

If α equals zero, no changes need to be done: creditors lose all the assets corresponding to credits on agent i and debtors gain the symmetrical of all liabilities corresponding to debts to agent i.

If α is greater than zero, we suppose that the assets are going to be divided by the creditors. Let us suppose that at instant $k-1$ agent i had $M(i)$ assets and $N(i)$ liabilities, each represented by an arrow:

$$A(i) = \sum_{m=1}^{M(i)} a_m(i) = -\sum_{m=1}^{M(i)} l_i(m) \tag{17}$$

$$L(i) = \sum_{n=1}^{N(i)} l_n(i) = -\sum_{n=1}^{N(i)} a_i(n) \tag{18}$$

With an abuse of notation, one takes m and n as indexes, based on i, for debtor and creditor nodes. Therefore, in expression (17) $a_m(i)$ represents the value of the asset that agent i owns from agent m. Likewise, $l_i(m)$ represents the corresponding symmetrical liability that agent m owes to agent i. In expression (18) $l_n(i)$ represents the value of the liability that agent i owes to agent n. Likewise, $a_i(n)$ represents the corresponding symmetrical asset that agent n owns to agent i.

Upon deletion of node i, the absolute values of arrows from each m node to the n nodes in instant k will be increased according to:

$$
\begin{aligned}
&\alpha A(i) \le -L(i) \rightarrow a_m(n) = \alpha a_m(i)\frac{a_i(n)}{A(i)} = -l_n(m) \\
&\alpha A(i) > -L(i) \rightarrow a_m(n) = a_m(i)\frac{a_i(n)}{A(i)} = -l_n(m)
\end{aligned} \tag{19}
$$

If the recoverable assets are less than or equal to liabilities, then each recoverable asset – $\alpha a_m(i)$ – is divided among creditors according to the credit fraction of each – $a_i(n)/A(i)$. If the recoverable assets are greater than liabilities, then each asset is divided among creditors according to the ratio of the credit of each to total assets.

For an example, consider that at instant $k = 1$ agent d in the network represented in Fig. 1 becomes insolvent. The matrix $NP(1)$ being given in (7), it follows that the the matrix $NP(2)$ would get the following values corresponding to $\alpha = 0$ and $\alpha = 0.5$:

$$
\begin{aligned}
&NP(1) = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & 0 & 31 & -98 & 16 \\ 0 & 0 & -69 & 89 & 1 \\ -31 & 69 & 0 & -83 & 0 \\ 98 & -89 & 83 & 0 & -65 \\ -16 & -1 & 0 & 65 & 0 \end{bmatrix} \\
&NP(2)_{|\alpha=0} = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & 0 & 31 & 0 & 16 \\ 0 & 0 & -69 & 0 & 1 \\ -31 & 69 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -16 & -1 & 0 & 0 & 0 \end{bmatrix} \quad NP(2)_{|\alpha=0.5} = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{bmatrix} 0 & -28 & 31 & 0 & 48 \\ 28 & 0 & -45 & 0 & 1 \\ -31 & 45 & 0 & 0 & -18 \\ 0 & 0 & 0 & 0 & 0 \\ -48 & -1 & 18 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{20}
$$

4.3 The iterating loops

We now give a general account of the procedure. Its main data structure is a three dimensional array $AS(i, j, k)$ where accounting states are recorded along agents in the network (i dimension), along agents initially assumed to become insolvent as a result of an exogenous loss (j dimension) and along discrete time (k dimension).

If the network has N agents or nodes, then AS dimensions must have sizes of N, N, and $N + 2$, the last one being required by the longest possible sequence of insolvency propagation being of length N (although that is seldom the case). The main computational work of the procedure is to "populate" the array, fully on dimensions i and j and up to the length required in dimension k.

In Fig. 2 we present a possible visualization of an AS array as an example.

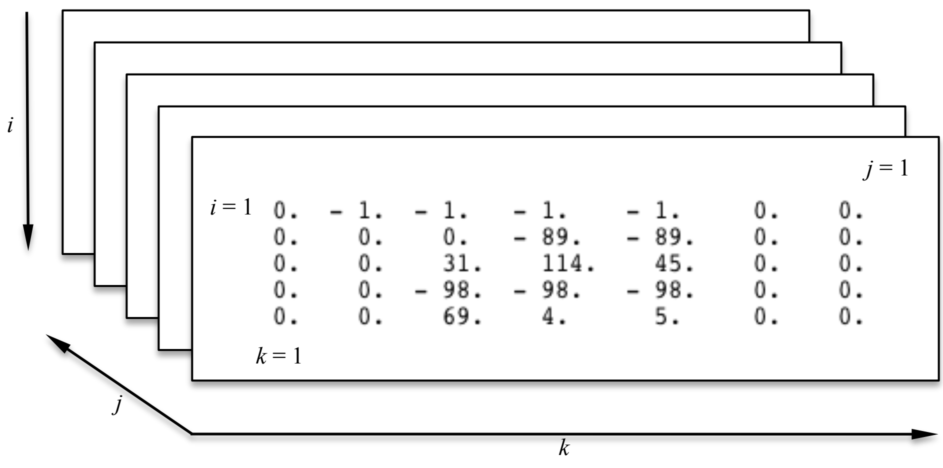

Figure 2 – Graphical depiction of an AS array for a network with five agents. In the visible "sheet", corresponding to j = 1, evolution of the network with NP matrix in (7) is shown, for α = 0 and zero initial capital for all agents, implying $AS(:,1,1)= 0$. At instant k = 2 agent i = 1 – named 'a' in (7) – becomes insolvent. Further accounting states of agents are computed until k = 5, when propagation of insolvency stops.

While calculation of the accounting state values is necessary, other information may be deemed more interesting. We have considered three items:

– The N sequences of insolvency propagation corresponding to the N possible initial exogenous insolvencies. These are stored in a vector of strings S, each string containing the sequence of nodes becoming insolvent.

– The number of nodes in each sequence, stored in a vector nF.

– The loss in equity in each sequence, stored in a vector mL.

For the exemplifying computation partially depicted in Fig. 2, the values of these items are:

$$S = \begin{bmatrix} 1;4;2; \\ 2,3;1;4; \\ 3;1,4,2; \\ 4;2;5; \\ 5;1,2;3,4; \end{bmatrix} \qquad nF = \begin{bmatrix} 3 \\ 4 \\ 4 \\ 3 \\ 5 \end{bmatrix} \qquad mL = \begin{bmatrix} -138 \\ -188 \\ -130 \\ -112 \\ -136 \end{bmatrix} \tag{21}$$

Inside strings, a semicolon separates two nodes that became insolvent at successive time instants; commas separate nodes that become insolvent at the same time instant.

We present now a pseudo-code description of the procedure, assuming that all the variables referred in the code have been initialized.

```
for j=1 to N
  set AS(:,j,1)
  set AS(:,j,2)
  S(j)='j'
  k=3
  while (sign(AS(:,k-1,j)) != sign(AS(:,k-2,j)))
    compute NP(j,k)
    for i=1 to N
      compute AS(i,k,j)
      append to S(j) string(i) if node i became insolvent
    end i
```

```
12      k=k+1
13    end while
14    compute nF(j)
15    compute mL(j)
16  end j
```

The outer ‘for’ loop started in line 1 computes in sequence the evolutions of the network beginning with exogenous insolvency of node j. Lines 2 and 3 set the values for the AS columns for instants 1 and 2. Line 4 records the first insolvent node in $S(j)$. The ‘while’ loop started in line 6 is executed if the signs of columns $k-1$ and $k-2$ of AS in the jth “sheet” being computed are different, signaling that propagation of insolvencies did not end yet. In line 7 the value of the $NP(k)$ matrix is determined according to subsection 4.2 above. Then in line 8 a ‘for’ loop starts that goes through all nodes establishing the accounting status for each at instant k and updating the $S(j)$ string if it is the case.

Calculation of the AS values is necessary, but, depending on the output one wants, actually storing all of them in memory can be avoided, drastically reducing memory requirements for large N. The above description of the procedure was written for clarity of presentation – actual implementation can be more complicated.

5 Aspects of Implementation

In this section we discuss implementation with a focus on the execution time of the procedure. If this is going to be used as a tool for the study and monitoring of real networks, execution time is a critical property.

The procedure has three inputs to consider: the NP matrix with size N^2, N being the number of agents or nodes, the $\boldsymbol{p}$ vector with size N and the parameter α. This last one can be given as a vector of size N, one value for each node, but in this reported implementation we used a single value for all nodes. The reason is that considering an α different from zero actually introduces a significant overhead as the values of the arrows going to an insolvent node must be recalculated and “spread” over several nodes. Nevertheless, it makes little difference if the procedure reads the α value from a single memory position rather than from a vector. Furthermore, recoding the implementation to have an α value for each node is a simple task.

Execution time depends on α in an on-off fashion. If α is different from zero, there is overhead, but this is independent of the α value. That is not the case for N: as fundamentally the procedure computes the values of a three-dimensional array whose size grows with N in all dimensions, it is to expect that the execution time grows proportional to N^3 or higher. Therefore we realized a set of tests to get an indication of the execution times of the procedure along several values of number of agents N, average degree, fraction of recoverable assets α and capital $\boldsymbol{p}$. In one of the tests, we made an empirical study of computational complexity.

5.1 Test conditions

All the tests were made in a MacBookAir6,2 with an Intel i5 2-core processor and 64-bit OSX. The Geekbench Browser [20] gives to this machine a score of 5162 points. In comparison, the highest scored Apple machine in the Geekbench is a MacPro with an Intel Xeon 12-core processor scoring 32140 points. This indicates that substantial room for improving execution time exists just by upgrading the hardware, without considering parallel computation schemes.

We conducted several tests to get an assessment of the relation between execution time versus the number of nodes and the average degree of the network. The network topology considered in the tests was of the random type [17]. We chose random networks because they are easy to generate. They also appear as a benchmark against which to assess the question of execution time in more specialized topologies of interest in the analysis of credit and debt networks, as the scale-free [4] and community [25] ones.

Given that networks will be represented by matrices, it does not appear that execution times be sensitive to the topology of the network being studied with this procedure. A network is represented as entries in a square matrix, therefore the number of rows and the number of non-null entries should mostly determine the computational load.

In all the network configurations used for tests of execution time, the random generator seed was fixed, so that the values generated for the network were the same for the same conditions. Values of credits and debts were constrained to the interval [–100, 100].

Execution times were read from timing instructions available in the languages used for coding the procedure and set in seconds. Their values showed a wide dispersion, therefore they must be considered an indicative sampling. Except where noted, the presented values were calculated as an average of four runs.

The language and environment firstly used for coding was Scilab [32]. After some tests, it became obvious that execution times in this environment were unsatisfactory even for a moderate number of nodes. This led us to recode the procedure in Julia [31], a young language already tested for an economic study [3]. After comparing the execution times reported in Table 1, we made all subsequent tests with the Julia implementation.

Table 1: Sampled execution times of the procedure for implementations in Julia and Scilab. The networks generated had an average degree of 20 and we set $\alpha = 0$ and $\boldsymbol{p} = 0$.

Language	Execution time for 100 nodes (s)	Execution time for 200 nodes (s)
Julia	0.12	0.90
Scilab	2.7	12.

5.2 *Execution times for the implementation in Julia*

The goal of the first test, reported in Table 2, was to assess the sensitivity of execution time to varying average degree. Using a network with 100 nodes and varying average degree from 20 to 100, we concluded that for $\alpha = 0$ execution time is fairly constant and for $\alpha \neq 0$ it increases mildly with average degree.

Table 2: Sampling of execution time of the procedure for a network with 100 nodes, varying average degree, two values of α and $\boldsymbol{p} = 0$.

Average degree	20	40	60	80	100
Execution time for $\alpha = 0$ (s)	0.12	0.12	0.12	0.12	0.11
Execution time for $\alpha = 0.5$ (s)	6.1	8.1	9.2	9.7	10.

In the second test we considered a network with constant average degree of 20 and capital of zero. Values for the number of nodes going from 100 to 1000, for $\alpha = 0$, and from 100 to 400, for $\alpha \neq 0$, are presented in Table 3.

Table 3: Execution time of the procedure for a network with varying number of nodes, average degree of 20, two values of α and $\boldsymbol{p} = 0$.

Number of nodes	100	200	400	800	1000
Execution time for $\alpha = 0$ (s)	0.12	0.75	5.3	51.	110.
Execution time for $\alpha = 0.5$ (s)	5.9	85. [(1)]	1300. [(1)]	—	—

[(1)] Execution time of one run only.

The execution times registered are plotted in Figure 2 for $\alpha = 0$ and for $\alpha \neq 0$. In both graphics, parabolic curves of expression $a_n \cdot nodes^n$, where $n = 3, 4, 5$, are also plotted. These curves include the last sample. The samples obtained suggest that for $\alpha = 0$, execution time grows faster than N^3, but slower than N^4. For $\alpha \neq 0$, execution time appears to grow with N^4.

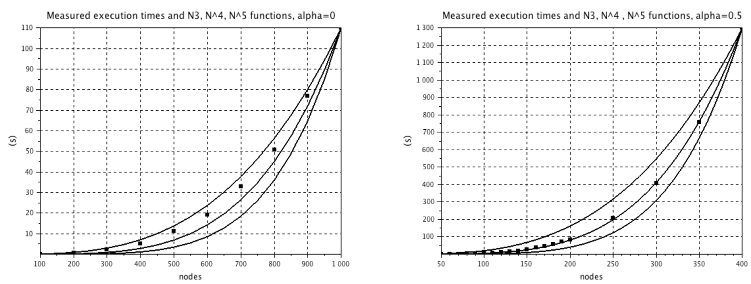

Fig. 2 – Execution times plotted for $\alpha = 0$ (left) and $\alpha \neq 0$ (right), together with curves $a_3 \cdot nodes^3$ (up), $a_4 \cdot nodes^4$ (middle), and $a_5 \cdot nodes^5$ (down).

The tests above supposed capital $\boldsymbol{p}$ being equal to zero for all agents. This is a computing worst case as the number of insolvencies is maximal. In the last test we assessed what happens when capital is increased for all agents from zero to the maximal absolute value generated for credits and debts (100). Table 4 records execution times for $\alpha = 0$ and $\alpha \neq 0$ together with average number of insolvencies for each case, again for a network of 100 nodes and average degree of 20. When capital is 100, insolvencies are limited to the initial ones, determined exogenously to the network. Execution times diminish markedly with the number of insolvencies if $\alpha \neq 0$. An interesting behavior to be studied elsewhere is the suggestion that, in the same capital conditions, reassigning assets lowers the number of insolvencies.

Table 4: Sampling of the procedure's execution times for a network with 100 nodes, average degree of 20, two values of α and varying $\boldsymbol{p}$.

Value of capital $\boldsymbol{p}$	0	20	40	60	80	100
Execution time for $\alpha = 0$ (s)	0.12	0.14	0.15	0.17	0.18	0.04
Average number of insolvencies	72.	67.	61.	53.	38.	1.0
Execution time for $\alpha = 0.5$ (s)	6.1	4.9	2.0	0.40	0.07	0.07
Average number of insolvencies	63.	42.	21.	4.8	1.2	1.0

6 Conclusions and Future Work

The aim of the research reported in this paper was to develop a computing procedure to assess the sensitivity of a credit and debt network to individual insolvencies originated outside the network. For this purpose, given a state of the network, all the possible trajectories of insolvencies beginning in a single insolvency are computed and relevant information from each trajectory is recorded.

We use a representation of the network state consisting of a matrix *NP*, recording the net credit and debt positions of agents, and a vector $\boldsymbol{p}$, recording their capital positions. With this data as input, the possible evolutions of the accounting states of agents, starting in single, exogenous insolvencies, are computed and the associated sequences of insolvency and values of equity variation are recorded. The computed evolutions depend on the assumed fraction, α, of recoverable assets belonging to insolvent nodes.

It comes out that the programming of the procedure allows for considering simultaneous initial defaults of agents by suitable setting initial data. Therefore, this situation can also be studied with this tool.

While the procedure potentially generates most relevant information, it is of a brute force character. Therefore, its execution times were tested in a range of situations. In the tests realized, it showed up that the procedure can be used for the study or monitoring in a real-time fashion of networks with up to 200 nodes if $\alpha \neq 0$ and up to 1000 nodes if $\alpha = 0$, without special hardware provisions and with the present level of refinement. An empirical study of computational complexity suggests that execution time grows faster than N^3, but slower than N^4 for $\alpha = 0$; it grows with N^4 for $\alpha \neq 0$.

Three lines of future research appear now. The first will be to make a formal study of the procedure convergence. The second will be to assess the possibility of computing only changes to already computed evolutions on the basis of point information arrival. So, assuming that information about two agents changing their credit and debt relation arrives, or a new node in the network is created, one would only "propagate" the "perturbations", rather than recalculating all the possible evolutions.

A third line of future research, of a more conceptual character, will be to investigate the extension of the procedure to a more general framework that will consider cash-flow insolvency in addition to the balance-sheet insolvency considered here.

Acknowledgements

We would like to thank the reviewers for helpful comments and criticisms.

The work of Paulo Garrido was financially supported by FCT, Fundação Portuguesa para a Ciência e a Tecnologia, through the Program PEst, Strategic Program of the Algoritmi Research Center, Project FCOMP-01-0124-FEDER-022674. The work of Pedro Campos was financially supported by Sibila and Smartgrids research projects (NORTE-07-0124-FEDER-000056/59), financed by North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the Development Fund (ERDF), and by national funds, through Fundação para a Ciência e a Tecnologia (FCT).

References

[1] Allen, F. and Gale, D., Optimal financial crises, *J. Financ.* **53** (1998) 1245–1248.

[2] Allen, F. and Gale, D., Financial contagion, *J. Polit. Econ.* **108** (2000) 1–34.

[3] Aruoba, S. B. and Fernandez-Villaverde, J., A comparison of programming languages in economics, Working paper, Penn University, (2014). Retrieved from http://economics.sas.upenn.edu/~jesusfv/comparison_languages.pdf

[4] Barabasi, A.-L. and Albert R., Emergence of scaling in random networks *Science*, **286** (1999) 509–512.

[5] Battiston, S. and Caldarelli, G., Systemic risk in financial networks, *J. Finan. Manage. Markets Institut.* **1** (2013) 129–154.

[6] Battiston, S., Delli Gatti, D., Gallegati, M., Greenwald, B. and Stiglitz, J. E., Credit chains and bankruptcy propagation in production networks, *J. Econ. Dyn. Control* **31** (2007) 2061–2084.

[7] Battiston, S., Delli Gatti, D., Gallegati, M., Greenwald, B. and Stiglitz, J. E., Default cascades: when does risk diversification increase stability?, ETH Risk Center Working Paper

Series (2011). Retrieved from https://www.sg.ethz.ch/ethz_risk_center_wps/pdf/ETH-RC-11-006.pdf
[8] Battiston, S., Delli Gatti, D., Gallegati, M., Greenwald, B. and Stiglitz, J. E., Liaisons dangereuses: Increasing connectivity, risk sharing, and systemic risk, *J. Econ. Dyn. Control* **36** (2012) 1121–1141.
[9] Bernanke, B. S., Nonmonetary effects of the financial crisis in propagation of the Great Depression, *Am. Econ. Rev*. **73** (1983) 257–276.
[10] Bernanke, B. S. and Gertler, M., Agency costs, net worth, and business fluctuations, *Am. Econ. Rev.* **79** (1989) 14–31.
[11] Blavarg, M., and Nimander, P., Interbank exposures and systemic risk, in *Risk Measurement and Systemic Risk*, Bank for International Settlements (2002).
[12] Chan-Lau, J. A., The global financial crisis and its impact on the chilean banking system, Mimeo, International Monetary Fund, Washington DC (2009).
[13] Chan-Lau J. A., Espinosa M., Giesecke K., and Solé J., Assessing the systemic implications of financial linkages, in *Global Financial Stability Report*, International Monetary Fund (2009).
[14] Diamond, D. and Dybvig, P., Bank runs, deposit insurance, and liquidity, *J. Polit. Econ.* **91** (1983) 401–419.
[15] Dias, A., Campos, P. and Garrido, P., An agent based model of bank failures, in Amblard, F., Miguel, F., Blanchet, A. and Gaudou, B. (eds.), *Advances in Artificial Economics, LNEMS – 676*, (Springer, 2014) pp. 119-130.
[16] Engle, R. F., Jondeau, E. and Rockinger, M., Systemic risk in Europe, *Review of Finance* **19** (2015) 145-190.
[17] Erdős, P., Rényi, A., On random graphs I, *Publicationes Mathematicae* **6** (1959) 290–297.
[18] Friedman, M. and Schwartz, A. J., *A Monetary History of the United States, 1867-1960* (Princeton University Press, Princeton, 1963).
[19] Georg, Co-Pierre, An agent based approach to interbank networks and monetary policy, Working paper, Jena University (2011).
[20] Geekbench Browser, Mac benchmarks, Primate Labs Inc (2014). Retrieved from https://browser.primatelabs.com/mac-benchmarks
[21] Hagan, W. D., *House of Cards: A Tale of Hubris and Wretched Excess on Wall Street* (Doubleday, New York, 2009).
[22] Iori, G., Jafarey, S. and Padilla, A., Systemic risk on the interbank market, *J. Econ. Behav. Organ.* **61** (2006) 525–542.
[23] Ivashina, V. and Scharftein, D., Bank lending during the financial crisis of 2008, *J. Fin. Econ.* **97** (2009) 319–338.
[24] Kleinberg, J., Cascading behavior in networks: algorithmic and economic issues, in Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds), *Algorithmic Game Theory* (Cambridge University Press, New York, 2007)
[25] Liu, W., Pellegrini, M. and Wang, X., Detecting communities based on network topology. *Sci. Rep.* **4**. (2014). Retrieved from http://dx.doi.org/10.1038/srep05739
[26] Markose, S., Shaghaghi, A. R., and Giansante, S. Too interconnected to fail financial network of US CDS market: topological fragility and systemic risk, *J. Econ. Behav. Organ.* **89** (2012) 627–646.
[27] Nier, E., Yang, Y., Yorulmazer, T,. and Alerton, A., Network models and financial stability, Working paper, Bank of England (2007).
[28] Provenzano, D., Contagion and bank runs in a multi-agent financial system, in *Managing Market Complexity, LNMCS – 662*, (Springer, 2013), pp. 27–38.
[29] Roukny, T., Bersini H., Pirotte, H., Caldarelli, G. and Battiston, S., Default cascades in complex networks: topology and systemic risk, *Scient. Reports* **3** (2013) art. 2759.
[30] Wikipedia, Insolvency, (2015). Retrieved from https://en.wikipedia.org/wiki/Insolvency
[31] Wikipedia, Julia (programming language), (2015). Retrieved from https://en.wikipedia.org/wiki/Julia_%28programming_language%29

[32] Wikipedia, Scilab. (2015) Retrieved from https://en.wikipedia.org/wiki/Scilab