

Credit Freezes, Equilibrium Multiplicity, and Optimal Bailouts in Financial Networks.

Matthew O. Jackson and Agathe Pernoud*

Draft: June 2022

Abstract

We analyze how interdependencies in financial networks can lead to self-fulfilling insolvencies and multiple possible equilibrium outcomes. We show that multiplicity arises if and only if there exists a certain type of dependency cycle in the network, and characterize banks' defaults in any equilibrium. We use this analysis to show that finding the cheapest bailout policy that prevents self-fulfilling insolvencies is computationally hard (and hard to approximate), but that the problem has intuitive solutions in specific network structures. Bailouts have an indirect value as making a bank solvent improves its creditors' balance sheets and reduces their bailout costs, and we show that an algorithm that leverages indirect benefits ensures systemic solvency at a cost that never exceeds half of the overall shortfall. In core-periphery networks, indirect bailouts are optimal: bailing out peripheral banks first as opposed to targeting core banks directly.

JEL CLASSIFICATION CODES: D85, F15, F34, F36, F65, G15, G32, G33, G38

KEYWORDS: Financial Networks, Markets, Systemic Risk, Financial Crisis, Networks, Banks, Credit Freeze, Default Risk, Financial Interdependencies

*Department of Economics, Stanford University, Stanford, California 94305-6072 USA. Jackson is also an external faculty member of the Santa Fe Institute. Email: jacksonm@stanford.edu and agathep@stanford.edu. We gratefully acknowledge financial support under NSF grants SES-1629446 and SES-2018554. We thank Marco Bardoscia, Philip Bond, Celso Brunetti, Ozan Candogan, Matt Elliott, Gerardo Ferrara, David Hirshleifer, Arjun Mahalingam, Carlos Ramirez, Tarik Roukny, Alireza Tahbaz-Salehi, Erol Selman, and especially Co-Pierre Georg, for helpful conversations and comments. This paper incorporates some results from “Distorted Investment Incentives, Regulation, and Equilibrium Multiplicity in a Model of Financial Networks.” We split that paper into this one and another, Jackson and Pernoud (2021b) (SSRN: <https://papers.ssrn.com/abstract=3311839>). Both papers contain new results beyond what appeared in the original paper.

1 Introduction

Today’s financial sector is characterized by strong interdependencies, with large amounts of capital circulating among financial firms. For instance, Duarte and Jones (2017) estimate that 23% of the assets of bank holding companies come from other financial intermediaries, as well as 48% of their liabilities - *almost half*.¹ The value of any financial firm then depends on the payments it gets from its claims on other firms, which might themselves depend on the values of yet other firms, creating rich interdependencies between firms’ balance-sheets.

The literature on financial contagion has studied how interbank obligations enable a shock in one part of the system to spread widely, and how this contagion is magnified if banks incur costs upon default.² What has received less attention is how these interdependencies can lead to multiple equilibria,³ with the defaults of some banks becoming self-fulfilling. For instance, a Bank A defaulting on its obligation to another Bank B, might lead B to default as well. This can lead to further missed payments that feed back to Bank A, making its initial default self-fulfilling.

Such self-fulfilling defaults on payments are not uncommon in practice, especially in times of distress. For instance, Fleming and Keane (2021) find that almost 74% of the settlement failures that occurred in the U.S. Treasury market in March 2020 were effectively “daisy-chain” failures, which could have been avoided had all trades been centrally netted.⁴ These defaults, even if only temporary, are costly: delayed in payments can lead counterparty risk to build up, dry up liquidity in the market, restrict or stall investments, and even push some banks towards insolvency. The existence of multiple equilibria therefore contributes to the fragility of the financial system: pessimistic beliefs or uncertainty about the state of others’ balance-sheets can become self-fulfilling and lead banks to stop payments to each other—a type of credit freeze—even when another equilibrium exists in which payments are all made.

In this paper, we characterize the network structures and portfolios that lead to self-fulfilling defaults and how to prevent them at minimal cost to the regulator. We consider financial networks in which banks are linked via unsecured debt contracts.⁵ The value of a bank depends on the value of its assets outside of the network, as well as of its claims on other banks. If a bank’s assets are not enough to cover its liabilities, the bank defaults on part of its debts. Importantly, we allow for failure costs that discontinuously depress a bank’s balance-sheet upon default, and reduce the amount repaid to counterparties.

¹See their Tables 1 and 2. The difference reflects the fact that many other types of financial institutions that are not BHCs (e.g., Real Estate Investment Trusts, insurance companies, etc.) have accounts of cash, money markets, and other deposits held at BHCs that count on the liability side.

²See, for instance, Eisenberg and Noe (2001), Gai and Kapadia (2010), Elliott, Golub, and Jackson (2014), Acemoglu, Ozdaglar, and Tahbaz-Salehi (2015), and Jackson and Pernoud (2021b). For a recent survey, see Jackson and Pernoud (2021).

³We use the term “equilibrium” to keep with the literature, but note that this term only reflects the mutual consistency of banks’ values – so a fixed point in accounting balance sheets – and not any strategic behavior.

⁴Settlement failures were much lower for trades that were centrally cleared (Duffie (2020)).

⁵The analysis extends to other sorts of contracts as we discuss in the Online Appendix.

There are two main applications/interpretations of our model. One is when a bank’s default involves bankruptcy, and the failure costs are the losses associated with filing for bankruptcy and liquidating assets.⁶ The other major application of the model is to situations in which a bank’s default may only be temporary, due to an episode of illiquidity, but can still lead to a sequence of defaults or delayed payments. Failure costs then capture costs associated with delayed payments, as outlined above.⁷ Such defaults are often avoided in practice by short-term borrowing. However, as in situations such as what occurred in 2008, lenders lose confidence in the market, which then leads a lack of liquidity for short terms borrowing, which then makes a freeze self-fulfilling.

We make the following contributions.

First, we provide a comprehensive analysis of the multiplicity of equilibrium bank values. We show that there exist multiple solutions for bank values if and only if there exists a certain type of cycle in the network. Cycles allow costly defaults to feedback through the financial network, generating the possibility of self-fulfilling defaults. There are well-defined ‘best’ and ‘worst’ equilibria, ordered by banks’ values. In the best equilibrium none of the self-fulfilling cycles of defaults occur,⁸ while in the worst equilibrium they all do. There are also intermediate equilibria that involve some, but not all, of the self-fulfilling cycles defaulting. We then give necessary and sufficient conditions for banks’ solvency under both the best and worst equilibria, assuming that a bank cannot make any payments until it is fully solvent. This applies in practice whenever insolvencies lead to delays in payments, which can cascade and generate a credit freeze.⁹ All banks are solvent in the best equilibrium if and only if their portfolio satisfy an appropriate balance condition. Solvency in other equilibria is more demanding, and we show how it is precisely characterized by cycles in the network.

Second, we analyze the most cost-efficient way to inject capital into banks so as to avoid defaults and associated deadweight losses. Injecting capital into a bank has consequences beyond the bank itself: by paying back its obligations to others, it can bring some of its counterparties closer to solvency and lowering the cost of bailing them out, or even bringing them back to solvency and triggering a repayment cascade. We show that these indirect-bailout

⁶Defaults involve substantial deadweight costs, including fire sales, early termination of contracts, administrative costs of government bailouts, and legal costs, among others; some of which are born by parties others than those who are responsible for the original default. Estimates of bankruptcy recovery rates can be below 60%, and even worse in a crisis. See, for example, Branch (2002), Acharya, Bharath, and Srinivasan (2007), and Davydenko, Strebulaev, and Zhao (2012).

⁷Eisenberg and Noe (2001) and most of the ensuing literature assume that all payments are cleared *simultaneously*, circumventing this problem. A notable exception is Csoka and Herings (2018) who show that most decentralized clearing processes yield the lowest equilibrium for bank values, providing additional motivation for our analysis. Although not explicit in our model, timing on payments can lead to self-fulfilling defaults: Bank A must pay Bank B first, and Bank B then must pay Bank C, and then Bank C can pay Bank A, but Bank A cannot make the payments until it receives the payment from Bank C. We abstract from timing details in the model given the complexity of our analysis, but they offer another interpretation.

⁸There can still be some defaults in the best equilibrium, but they are not self-fulfilling. We characterize these in the results below.

⁹The insights and main results extend when there are partial payments (a fortiori), but this case provides the most direct intuition.

values are critical to understanding the minimal injection of capital needed to ensuring systemic solvency. Building on our analysis of equilibrium multiplicity, we identify the minimum bailout payments needed to ensure that all banks are solvent in *any* given equilibrium.

The minimum bailouts to ensure systemic solvency in the best equilibrium do not depend on the specific network structure, and only require bringing each bank’s portfolio into balance. In contrast, the minimum bailouts ensuring systemic solvency in any other equilibrium depend on the details of the network structure, as they require injecting enough capital so as to clear whichever self-fulfilling cycles are defaulting in that equilibrium. Characterizing the minimum bailouts needed to ensure solvency in a non-best equilibrium is thus much more complex. In fact, we prove that it is a *strongly NP-hard* problem - which implies that there are no known practical algorithms for finding approximate solutions, even with relatively small numbers of banks. Part of the complexity comes from the fact that the amount of capital the regulator has to inject in a bank to make it solvent depends on who else is already solvent in the network – in short, the order of bailouts matters, and every bank’s balance sheet can change with each sequence of bailouts. Hence the number of bailout policies to consider is of the order of $n!$, where n is the number of banks, and so it is already in the trillions with just fifteen banks and beyond 10^{18} with twenty banks, and so an exhaustive search for the optimal policy gets infeasible very quickly.

Despite this complexity, our analysis provides intuitive insights about optimal bailouts. First, indirect bailouts are part of an optimal policy: When considering the bailout of a particular bank, instead of injecting capital directly into it, it is often cheaper to inject smaller amounts into banks that owe that bank money, leveraging those banks’ capital. This can be seen as an explanation of the AIG bailout in 2008, which some argue was an indirect bailout of Goldman Sachs and others (Bernard, Capponi and Stiglitz (2017)).¹⁰ Building on this intuition, we propose a simple algorithm that bails out banks in decreasing order of their indirect bailout value to bailout cost ratio. Even though this algorithm is not optimal in some networks, we show that it guarantees systemic solvency at a total cost that never exceeds half of the total overall shortfall. Second, overlapping cycles of claims in the network drive the complexity of optimal bailouts. We derive a bound on the number of calculations needed to find the optimal bailout policy as a function of the number of cycles in the network.

We then consider some prominent network structures under which there are simpler and intuitive optimal bailout policies that prevent self-fulfilling defaults and freezes. A key example is a star network in which a core bank is linked to peripheral banks. Importantly, only the core bank lies on several cycles. Finding the optimal bailout policy is then a more tractable problem – specifically, it is only *weakly* NP-hard and collapses to what is known as a Knapsack problem. We show that it is always cheaper to start by bailing out peripheral banks as opposed to targeting the core bank directly, as it allows the regulator to leverage peripheral banks’ capital buffers.¹¹ This again highlights the value of indirect bailouts, and

¹⁰Roughly, one could view AIG as a peripheral node as it was mainly selling insurance and so was a major debtor rather than creditor, whereas Goldman Sachs is a core dealer and was a major creditor of AIG.

¹¹Note how considering the network structure yields different insights from the literature on optimal bailout

we show how some simple bailout algorithms work well in this setting.

We then discuss more general core-periphery networks, in which a similar intuition also holds: if peripheral banks are “small” relative to core banks, then it is always optimal to start by bailing out the periphery. We also show, however, that finding the overall optimal bailout policy is strongly NP-hard, as core banks are densely connected and lie on many overlapping cycles. Thus, although we show that finding optimal bailouts is computationally challenging, we also show that there are many settings in which parts of the problem can be solved efficiently, while other parts can remain computationally hard. Regardless, the indirect bailout values are important to understand and can improve over more naive bailout policies.

Overall, our results also identify the precise benefit of canceling out cycles of claims – what has become known as ‘compression’ – and we end the paper with a discussion of such netting techniques.

Our model builds on the literature that followed the interbank lending network model of Eisenberg and Noe (2001). They introduce the notion of a clearing vector, which specifies mutually consistent repayments on interbank loans for all banks in the network, and show that it is generically unique. Others have pointed out that the non-negligible failure costs that banks incur whenever they are insolvent facilitate the existence of multiple clearing vectors, and hence of multiple equilibrium for banks’ values (Rogers and Veraart (2013); Elliott, Golub, and Jackson (2014))). This multiplicity comes from the discontinuous drop in a bank’s value at default, which can create self-fulfilling combinations of defaults.¹² This multiplicity has not been examined in any detail, and instead previous studies restrict attention to equilibrium repayments that lead to the least number of defaults.

One of the only papers that emphasizes the multiplicity of clearing vectors, and hence of equilibrium values for banks, is Roukny, Battiston and Stiglitz (2018). However, the main focus of their paper is completely different from ours: they show how equilibrium multiplicity makes assessing systemic risk harder, as it means some defaults are indeterminate, and propose a method to measure this source of uncertainty. We give a new and full characterization of equilibrium multiplicity, which we then use to analyze and develop optimal bailout policies.

Our analysis of optimal bailouts relates to Demange (2016), who characterizes the optimal cash injection policy in a network of interbank lending. She defines an institution’s threat index as the marginal impact of an increase in its direct asset holdings on total debt repayments in the system, assuming the policy does *not* change the set of defaulting banks. Hence a bank’s threat index captures its *marginal* social value of liquidity. In this paper, we instead examine how much of an injection is needed to change and avoid defaults, and show

policies when banks are heterogenous. In the latter, it can make sense to bail out stronger banks first as they are closer to solvency (e.g., see Choi (2014)), but this overlooks the indirect value of bailouts.

¹²This source of equilibrium multiplicity differs from bank runs à la Diamond and Dybvig (1983). Here, it stems from reduced payments due to failure costs that become self-fulfilling, and not from the optimizing behavior of agents who, anticipating a bank’s failure, claim their assets and bring it to insolvency.

how complex that problem is, and offer insights into solving it.¹³

There is also a literature on externalities and incentives associated with bailouts in financial networks. Erol (2019) shows that public bailouts affect banks’ choice of counterparty, and hence the equilibrium structure of the financial network. Several papers study how linkages between banks can incentivize private sector bailouts, whereby solvent banks bail out insolvent ones, and how this depends on the network structure (e.g., Leitner (2005), Kanik (2019)); and Bernard, Capponi and Stiglitz (2017) analyze the interplay between public bailouts and private bail-ins. Capponi, Corell, and Stiglitz (2020) investigate how debt-financed bailouts increase the risk premia on sovereign debt, and hence depress the balance sheets of the very banks the regulator was trying to save, generating a doom loop. Our paper is orthogonal to this literature, as we examine the structure and complexity of bailouts rather than their external effects.

Finally, the literature on payment netting (e.g., Kahn and Roberds (1998), Martin and McAndrews (2008)) and portfolio compression (D’Errico and Roukny (2019) and Schuldenszucker and Seuken (2019)) is also worth noting. Portfolio compression is an increasingly popular multilateral netting technique, which allows banks to eliminate offsetting obligations with other organizations taking part in the process. We discuss how our results show novel benefits of such netting techniques in Section 6.3.

2 A Model of Financial Interdependencies

2.1 Financial Institutions and their Portfolios

Consider a set $N = \{0, 1, \dots, n\}$ of organizations involved in the network. We treat $\{1, \dots, n\}$ as the financial organizations, or “banks” for simplicity in terminology. These should be interpreted as a broad variety of financial organizations, including banks, venture capital funds, broker-dealers, central counterparties (CCPs), insurance companies, and many other sorts of shadow banks that have substantial financial exposures on both sides of their balance sheets. These are organizations that can issue as well as hold debt.

We lump all other actors into node 0 as these are entities that either hold debt in the financial organizations (for instance private investors and depositors), or borrow from the financial organizations (for instance, most private and public companies), but not both. Their balance sheets may be of interest as well, as the defaults on mortgages or other loans could be important triggers of a financial crisis. The important part about the actors in node 0 is that, although they may be the initial trigger and/or the ultimate bearers of the costs of a financial crisis, they are not the dominoes, becoming insolvent and defaulting on payments as a result of defaults on their assets. In aggregate, it may appear that there is debt going

¹³Since the first writing of this paper, others have also analyzed complexity of bailouts (Eggressy and Wattenhofer (2021), Klages-Mundt and Minca (2022)). They focus on the best equilibrium and consider other objective functions, more in line with our discussion in Section 6.2 than with our main analysis.

both in and out of node 0, but none of the individual private investors that comprise node 0 have debt coming both in and out.¹⁴

Bank portfolios are composed of investments in assets outside the system as well as financial contracts within the system. Investments in primitive assets involve some initial investment of capital and then pay off some cash flows over time, often randomly; e.g., government securities, asset backed securities, corporate and private loans, mortgages, etc. This is part of a bank's capital. We let p_i denote the current total market value of i 's investments in those assets and $\mathbf{p} = (p_i)_i$ denote the associated vector. (Analogous bold notation is used for all vectors and matrices.)

The book values of banks in the network are based not only on the capital in these outside investments, but also on assets from and liabilities to others in the financial system. In this paper, we restrict attention to interbank debt contracts, but the model extends to allow for other types of financial contracts between banks.¹⁵ A debt contract between a creditor i and a debtor j is characterized by its face value D_{ij} . As a bank cannot have a debt on itself, we set $D_{ii} = 0$ for all i . Let \mathbf{D} the matrix whose (i, j) -th entry is equal to D_{ij} , and denote a bank's total nominal debt assets and liabilities by, respectively, $D_i^A \equiv \sum_j D_{ij}$ and $D_i^L \equiv \sum_j D_{ji}$.

2.1.1 The Weighted Directed Network

The financial network generated by interbank lending contracts is thus represented by a weighted directed graph on N , where a directed edge pointing from bank i to bank j means i has a debt liability toward j with a weight of D_{ji} , so edges point from the debtor to the creditor.

A *directed path* in the network from i to j is a sequence of banks i_0, \dots, i_K , for some $K \geq 2$ such that: $i_0 = i$ and $i_K = j$ and $D_{i_{\ell+1}i_\ell} > 0$ for each $\ell < K$. Thus a directed path is such that i owes a debt to some bank, which owes a debt to another bank, and so forth, until j is reached.

A *dependency cycle*, or *cycle* for short, is a directed cycle in the network which is a sequence of banks i_0, \dots, i_K , for some $K \geq 2$ such that: $i_0 = i_K$ and $D_{i_{\ell+1}i_\ell} > 0$ for each $\ell < K$. A directed cycle is *simple* if i_0 is the only bank repeated in the sequence.

Given that node 0 is comprised of entities that cannot be involved in cycles, to avoid confusion with the network of debts, we set $D_{i0} = 0$ for all i , and any debts owed from

¹⁴Of course, this is an approximation and there is a spectrum that involves a lot of gray area. For instance, Harvard University invests tens of billions of dollars, including making large loans. At the same time it borrows money and has issued debt of more than five billion dollars. It is far from being a bank, but still has incoming and outgoing debt and other obligations. This is true of many large businesses, and all those that could become dominoes should be included in $\{1, \dots, n\}$. It is not so important for us to draw an arbitrary line through this grey area to make our points. Nonetheless, this is something a regulator does have to take a stand on when trying to address systemic risk, and in practice may even be dictated by jurisdictional rules.

¹⁵See Appendix B.4 of the online material for an extension of the model and results when banks also hold equity claims on each other.

outside of the network of banks are instead recorded in p_i . Outsiders can still default on a payment to a bank, but that is captured in a lower value of p_i .

2.2 Defaults and Equilibrium Values of Banks

A main object of interest in this paper is a bank's book value V_i . We first need to introduce defaults and their associated costs, before characterizing bank values.

If the value of bank i 's assets falls below the value of its liabilities, the bank is said to *fail* and incurs *failure costs* $\beta_i(\mathbf{V}, \mathbf{p}) \geq 0$. These costs capture the fact that the value of a bank's balance sheet can be discontinuously depressed upon insolvency, for instance due to direct costs of bankruptcy: legal and auditors' fees, fire sales, premature withdrawals, or other losses associated with halted or decreased operations. In our setting, β_i can also capture some more indirect costs: it is not necessary that the bank declares bankruptcy, but simply that its insolvency causes it to renegotiate its contracts or delay payments, leading to a freeze and imposing costs on it and its creditors. Such failure costs can depend on the degree to which i and others are insolvent as well as the value of their portfolios. Hence we allow these costs to depend on the vector of bank values \mathbf{V} and returns on outside investments \mathbf{p} .¹⁶

With the possibility of default, the realized debt payment from a bank j to one of its creditors i depends on the value of V_j , and thus its solution depends ultimately on the full vector \mathbf{V} . To make these interdependencies explicit, let $d_{ij}(\mathbf{V})$ denote the amount of debt that bank j actually pays back to i .

There are two regimes. If bank j remains solvent, it can repay its creditors in full, and then for all i

$$d_{ij}(\mathbf{V}) = D_{ij}.$$

If instead j defaults, then debt holders become the residual claimants, and are rationed proportionally to their claim on j :

$$d_{ij}(\mathbf{V}) = \frac{D_{ij}}{\sum_h D_{hj}} \max(p_j + d_j^A(\mathbf{V}) - \beta_j(\mathbf{V}, \mathbf{p}), 0), \quad (1)$$

with $d_j^A(\mathbf{V}) = \sum_i d_{ji}(\mathbf{V})$.

It is useful to introduce notation for a bank's realized failure costs, $b_i(\mathbf{V}, \mathbf{p})$, which are 0 if the bank remains solvent and equal to $\beta_i(\mathbf{V}, \mathbf{p})$ if it defaults:¹⁷

$$b_i(\mathbf{V}, \mathbf{p}) = \begin{cases} 0 & \text{if } p_i + d_i^A(\mathbf{V}) \geq D_i^L \\ \beta_i(\mathbf{V}, \mathbf{p}) & \text{if } p_i + d_i^A(\mathbf{V}) < D_i^L. \end{cases} \quad (2)$$

¹⁶We have reduced the portfolios to only track their total value, but in practice the failure costs incurred could depend on detailed information about the composition of the bank's portfolio as well as the portfolios of others.

¹⁷This allows us to use the same notation for solvent and insolvent banks, and makes equilibrium bank values easier to write down.

We add conditions on failure costs to avoid the possibility that these costs per se, and not the reduced payments they imply, induce multiple equilibria for bank values. The key assumption is that failure costs cannot increase faster than the value of a bank's assets. In particular, we assume that $d_i^A(\mathbf{V}) - \beta_i(\mathbf{V}, \mathbf{p})$ is nondecreasing in \mathbf{V} .

Note also that we have carefully written failure costs $b_i(\mathbf{V}, \mathbf{p})$ as a function of how a bank's assets $p_i + d_i^A(\mathbf{V})$ compare to its liabilities D_i^L , instead of as a function of V_i directly. This avoids having defaults or freezes driven solely by the anticipation of failure costs, even when a bank has more than enough assets, even cash on hand, to cover its liabilities. Such a self-fulfilling default would go beyond a bank run, since it would not be due to the bank not having enough cash on hand to pay its debts. It would instead be due to self-fulfilling failure costs with no interaction with the financial network or portfolio values. We rule this out as it seems of no practical interest. However, we do allow realized failure costs $\beta_i(\mathbf{V}, \mathbf{p})$ to depend on the value of other banks; e.g., a bank can incur higher costs upon default if others are defaulting as well, because of fire sales. Such a dependency is important as it can worsen contagion, and we include it in our analysis.

A canonical example of admissible failure costs corresponds to the case where

$$\beta_i(\mathbf{V}, \mathbf{p}) = b + a [p_i + d_i^A(\mathbf{V})] \quad (3)$$

with $b \geq 0$ and $a \in [0, 1]$.¹⁸ In that case, failure costs are composed of some fixed amount (e.g., legal costs), as well as some share of the value of the bank's assets. This is a reasonable assumption if, for instance, the bank only recovers some fraction of its assets upon sale (e.g., due to a markdown on a fire sale of its assets) or has a portion of its legal costs that scales with the size of the enterprise.¹⁹

It can be that failure costs exceed the value of the defaulting bank's assets; e.g., if b is large enough in the above example. These excess costs should be interpreted as real costs, for instance, debts or legal costs that are never paid, capital or labor that are idled, etc., which can be incurred by the bank itself if it does not act under limited liability, or by the government or agents outside of the network (so node 0 in our framework). What matters for our analysis is not whether failure costs can exceed the bank's assets, but that they crowd out some of the debt repayments to creditors.

Failure costs are imposed on banks' balance sheets directly, and so the book value of a

¹⁸If $b = 0$, these failure costs are similar to the ones considered in Rogers and Veraart (2013), who also observe that costs of default can generate equilibrium multiplicity. They, however, do not study which network structures generate multiple equilibria nor how to prevent such multiplicity, and instead restrict attention to the best equilibrium to study rescue consortia (see the discussion in Kanik (2019)).

¹⁹Several papers estimate marginal bankruptcy costs being around 20% or 30% of the value of a bank's assets (see for instance Davydenko et al. (2012)), suggesting a should be in that range. As mentioned above, these can be even larger, Branch (2002), Acharya, Bharath, and Srinivasan (2007), particularly in times of a financial crisis. Fixed costs associated with bankruptcies are harder to estimate, but are presumably strictly positive given the structure of legal and accounting costs, as well as the fact that marginal cost estimates are much lower than overall costs.

bank i is

$$V_i = p_i + d_i^A(\mathbf{V}) - D_i^L - b_i(\mathbf{V}, \mathbf{p}), \quad (4)$$

where $b_i(\mathbf{V}, \mathbf{p})$ is defined by (2) and $d_i^A(\mathbf{V})$ by (1) whenever there are some defaults. In matrix notation:

$$\mathbf{V} = \mathbf{p} + \mathbf{d}^A(\mathbf{V}) - \mathbf{D}^L - \mathbf{b}(\mathbf{V}, \mathbf{p}). \quad (5)$$

A vector of bank values \mathbf{V} is an *equilibrium* if it is a solution to equation (5). Note that these are banks' equilibrium values (ex post), at the time of settlement, once returns and defaults are realized. Any shock to outside investments is already realized, observed, and encoded into \mathbf{p} . Equilibrium values can be negative if a bank's liabilities exceed the value of its assets. These can then be interpreted as "hypothetical" values and not equity values. If a bank i is solvent, then its equity value coincides with V_i . However, if its "hypothetical" value V_i is negative then i defaults, and its equity value is zero. The extra negative value means that a bank's assets are not enough to cover its liabilities, and hence that some debt payments are not made in full. Coupled with failure costs, there are deadweight losses in the economy.

3 Multiplicity of Bank Values and Self-Fulfilling Defaults

Although the possibility of multiple equilibria for bank values in financial networks is well-known, the conditions under which they exist and their implications are not. In this section we characterize when there exists a multiplicity. We also derive necessary and sufficient conditions on portfolio values for all banks to be solvent, which depend on the equilibrium being considered.

All of the definitions that follow are relative to some specification of \mathbf{p}, \mathbf{D} , and we omit its mention.

3.1 The Multiplicity of Equilibrium Values

Because the value of a bank is weakly increasing in others' values, there always exists a solution to equation (5). Furthermore, there can exist multiple solutions given the interdependence in values, and in fact, the set of equilibrium values forms a complete lattice.²⁰ Thus, there exists a "best" as well as a "worst" equilibrium, in which bank values hit an overall maximum and minimum, respectively. The set of defaulting banks is hence the largest in the worst equilibrium, and the smallest in the best.

The following algorithm finds the best equilibrium. Start from bank values \mathbf{V}^0 that are

²⁰This can be seen by an application of Tarski's fixed point theorem, since banks' values depend monotonically on each other. They are bounded above by the maximum values of banks' assets $\mathbf{p} + \mathbf{D}^A$.

at least as high as the best equilibrium values (e.g., $\mathbf{V}^0 = \mathbf{p} + \mathbf{D}^A$) and then compute

$$\mathbf{V}^1 = \mathbf{p} + \mathbf{d}^A(\mathbf{V}^0) - \mathbf{D}^L - \mathbf{b}(\mathbf{V}^0, \mathbf{p}).$$

If any values are negative, the associated banks default and the values are computed again accordingly. Iterating this process yields the best equilibrium. The worst equilibrium can be found using a similar algorithm, but starting from values that must lie below the worst equilibrium values (e.g., $\mathbf{p} - \mathbf{D}^L - \bar{\mathbf{b}}$, where $\bar{\mathbf{b}}$ is a cap on how large failure costs can be).

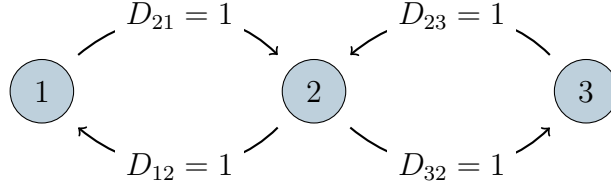


Figure 1: An Example of Equilibrium Multiplicity. Arrows point in the direction that debt is owed, that is from debtors to creditors.

Figure 1 provides a simple example, for which we compute bank values using these two algorithms.²¹ Suppose $p_1 = 1$, $p_2 = p_3 = 0$, and $\beta_i(\mathbf{V}, \mathbf{p}) = 0.5[p_i + d_i^A(\mathbf{V})]$ for $i = 1, 2, 3$. So a bank loses half of the value of its assets upon default. Let us first derive the best equilibrium for bank values using the above algorithm. We initiate at $\mathbf{V}^0 = \mathbf{p} + \mathbf{D}^A = (2, 2, 1)$. Since $\mathbf{V}^0 \geq \mathbf{0}$, no failure costs are incurred $\mathbf{b}(\mathbf{V}^0, \mathbf{p}) = \mathbf{0}$ and all debts are repaid $\mathbf{d}^A(\mathbf{V}^0) = \mathbf{D}^A$. Then $\mathbf{V}^1 = (1, 0, 0)$. Since $\mathbf{V}^1 \geq \mathbf{0}$ the algorithm stops, and bank values in the best equilibrium are $\bar{\mathbf{V}} = (1, 0, 0)$.

We now derive the worst equilibrium for bank values. Note that failure costs cannot be greater than $\bar{\mathbf{b}} = 0.5[\mathbf{p} + \mathbf{D}^A]$. We hence initiate the algorithm at $\mathbf{V}^0 = \mathbf{p} - \mathbf{D}^L - \bar{\mathbf{b}} = (-1, -3, -1.5)$. Since $p_1 + d_1^A(\mathbf{V}^0) \geq p_1 = D_1^L$, it has to be that bank 1 does not default: $b_1(\mathbf{V}^0, \mathbf{p}) = 0$ and $\mathbf{d}_{21}(\mathbf{V}^0) = D_{21}$. Debt repayments for bank 2 and 3 solve

$$\begin{cases} d_{23}(\mathbf{V}^0) = \frac{D_{23}}{D_3^L} 0.5[p_3 + D_{31} + d_{32}(\mathbf{V}^0)] = 0.5d_{32}(\mathbf{V}^0) \\ d_{32}(\mathbf{V}^0) = \frac{D_{32}}{D_2^L} 0.5[p_2 + D_{21} + d_{23}(\mathbf{V}^0)] = 0.25[1 + d_{23}(\mathbf{V}^0)] \end{cases} \iff \begin{cases} d_{23}(\mathbf{V}^0) = \frac{1}{7} \\ d_{32}(\mathbf{V}^0) = \frac{2}{7} \end{cases}.$$

Hence, $\mathbf{d}^A(\mathbf{V}^0) = (\frac{2}{7}, \frac{8}{7}, \frac{2}{7})$, and $\mathbf{V}^1 = \mathbf{p} + \mathbf{d}^A(\mathbf{V}^0) - \mathbf{D}^L - \mathbf{b}(\mathbf{V}^0, \mathbf{p}) = (\frac{2}{7}, -\frac{10}{7}, -\frac{6}{7})$. Since no new solvencies are induced – indeed $p_i + d_i^A(\mathbf{V}^1) < D_i^L$ for $i = 2, 3$ – the algorithm stops, and bank values in the worst equilibrium are $\underline{\mathbf{V}} = (\frac{2}{7}, -\frac{10}{7}, -\frac{6}{7})$.

When there are multiple equilibria, any equilibrium other than the best equilibrium must involve cycles of defaults that could have been avoided: these are defaults that are triggered either by pessimistic beliefs about the balance-sheets of others—valuing their debt at a low value—which become self-fulfilling, or by a liquidity freeze in which debts are not paid in

²¹See Section B.1 of the Appendix for an example with three different equilibria for bank values.

and thus not paid out. These avoidable defaults are essentially coordination failures: cycles of banks could have written-off (some) of their counterparties' debt so as to avoid at least some of the defaults and associated costs. Given that financial markets are prone to runs and freezes, understanding when such self-fulfilling cascades exist is of practical importance.

The following proposition highlights how equilibrium multiplicity depends on the presence of cycles of liabilities, combined with failure costs.

PROPOSITION 1. *For each i , let the failure costs $\beta_i(\cdot, \mathbf{p})$ depend only on i 's value V_i and the values V_j of banks j on which i has a (potentially indirect) debt claim,²² and be a contraction as a function of V_i (e.g., the canonical failure costs in (3)).*

- (i) *If there is no dependency cycle, then the worst and best (and thus all) equilibria coincide.*
- (ii) *Conversely, if there is a dependency cycle, then there exist failure costs (satisfying the above conditions) and values of bank investment portfolios \mathbf{p} such that the best and worst equilibria differ.*
- (iii) *Any equilibrium that differs from the best equilibrium consists of the defaults in the best equilibrium plus the defaults of all banks that lie on some set of dependency cycles. Any other banks defaulting in this equilibrium but not in the best equilibrium lie on out-pointing paths from the original defaulting banks and the newly defaulting dependency cycles.²³*

We are not the first to notice that cycles are necessary to generate multiple equilibria, as they are required for self-fulfilling feedbacks (e.g., see Elliott et al. (2014), Roukny et al. (2018)), and a similar intuition is at play here and underlies part (i). Proposition 1 however goes beyond that as it proves that cycles are also sufficient (part (ii)), and it highlights the general structure of the set of equilibria (part iii). This last observation is particularly useful, as even though there is a lattice of equilibria, figuring out what those equilibria are could be quite complex. This result ensures that one can find all equilibria by examining combinations of cycles, which can greatly simplify the calculations.

Although proving part (i) is straightforward, proving part (ii) requires finding returns \mathbf{p} and failure costs $\beta_i(\cdot, \mathbf{p})$ that generate multiple equilibria. The specific failure costs we use in the proof are $\beta_i(\mathbf{V}, \mathbf{p}) = a(p_i + d_i^A(\mathbf{V}))$ for a above some threshold; that is, a bank loses a fraction of its assets when defaulting. Hence, we do not need to construct costs that depend on other banks' values to generate multiplicity: it is enough for them to only depend on the value of the defaulting bank's assets. It is also not necessary for banks to lose the entirety of

²²More generally, if one admits equity holdings or fire sale interactions in failure costs, then one can extend this result by appropriately defining dependency cycles so as to account for all the ways j 's value can influence i 's. See the appendix for more details.

²³It is possible that the new defaults on an additional dependency cycle leads banks that defaulted in the best equilibrium to payout even less than in the best equilibrium, which can lead to additional defaults on rays outwards from the original banks, not just the newly defaulting banks.

their assets ($a = 1$) for the multiplicity to arise, though we assume they do in our analysis of minimum bailouts for the sake of transparency.

Part (iii) is intuitive, but is important to state as it is very helpful in calculating equilibria.²⁴ Essentially, beyond the best equilibrium, all other equilibria involve self-fulfilling cycles of defaults; and this can focus the search for equilibria. Those cycles can also lead to additional casualties that are on outward paths from an originally defaulting bank or an additional cycle, but the equilibria must be based on some dependency cycles. Not just any combination of dependency cycles will work (some banks might be strong enough to never default), but each equilibrium must differ from other equilibria by the inclusion/exclusion of at least one cycle.

It is clear that larger failure costs make feedback easier, and lower costs make it harder. For example, consider a “wheel” network consisting of a single debt cycle composed of n banks, such that bank i owes debt D to bank $i + 1$. To close the wheel, bank n owes debt D to bank 1. Since each bank is always able to repay all its liabilities if its counterparty does, the best equilibrium always has all banks being solvent, irrespective of \mathbf{p} . The range of primitive investment values (p_i s) for which there exists a (worst) equilibrium in which all banks default depends on the size of failure costs. For the purposes of illustration, suppose that costs are some fraction a of a bank’s assets, and that banks all have the same portfolio value $p_i = p$. For which values of p does there exist an equilibrium with mutual defaults? Supposing that such an equilibrium does exist, then the amount that banks pay each other turns out to be some $d < D$ that by (1) solves:

$$d = (1 - a)(p + d),$$

or $d = \frac{1-a}{a}p$. From (2), it must be that $p + d < D$ for banks to incur failure costs, which then requires that

$$p + \frac{1-a}{a}p < D$$

or $p < aD$. Thus, as failure costs fall, the set of asset returns \mathbf{p} that can generate self-fulfilling default cycles, and hence multiple equilibria, shrinks.

With the restriction that failure costs can only depend on a bank’s direct and indirect neighbors, the feedback that leads to multiple equilibria can only occur through a dependency cycle (Proposition 1 (i)). If instead, the costs incurred by bank i upon default are larger if j defaults as well, even though i has no network path to j , then failure costs themselves can generate “indirect” cycles. For example, if fire sales change bank values, then i ’s value could depend on whether j becomes insolvent even if i is not path-connected to j in the network of debts. It is then possible to have multiple equilibria for bank values even in the absence of any cycle in the network of debt. The following example illustrates this.

Suppose Bank 2 and Bank 1’s outside investments yield $p_2 = 1 - \varepsilon$ and $p_1 = 1$, respectively.

²⁴Even in problems with complementarities and a complete lattice of equilibria, finding all equilibria can be very challenging (see Echenique (2007) for an algorithm).



Figure 2: An Example of Indirect Cycles.

Since Bank 2 has no other assets, it must default in any equilibrium. Bank 1's outside investments are not quite enough to pay back its liabilities, but a small repayment from Bank 2 would be enough to ensure its solvency. Suppose Bank 2's failure costs are small if Bank 1 is solvent, $\beta_2(\mathbf{V}, \mathbf{p}) = \varepsilon \mathbb{1}\{V_1 \geq 0\}$, but large if it is not, $\beta_2(\mathbf{V}, \mathbf{p}) = \mathbb{1}\{V_1 < 0\}$. There exist two equilibria despite the absence of cycles in the network. In the first equilibrium, Bank 1 remains solvent and Bank 2 incurs a small cost, and thus repays $D_{12}(\mathbf{V}) = 1 - 2\varepsilon$ to Bank 1 (which then indeed has enough assets to be solvent). In the second equilibrium, Bank 1 defaults and Bank 2 incurs a large cost and repays nothing to Bank 1. One can extend Proposition 1 to take into account such fire-sale cycles that arise under more general failure costs.

For the remainder of the paper, we analyze the case in which, even if a bank has some money coming in, it cannot use that money to pay some of its debt until it is fully solvent. So $\beta_i(\mathbf{V}, \mathbf{p}) = p_i + d_i^A(\mathbf{V})$ for each bank i .

The worst equilibrium for bank values then arises whenever we take the timing of payments seriously. Indeed, the algorithm that finds the worst equilibrium starts by assuming everyone defaults, which under this specification for failure costs means that no one makes any debt payments. The only solvent banks are then those that are able to make their payments out even without receiving any payments in. This first wave of payments might enable other banks to meet their obligations, etc, and iterating on this yields the worst equilibrium for bank values. Hence this assumption applies in practice when insolvency leads to delays in payments, which can lead to cascading delays and cycles, and thus at least to a temporary freeze.

These failure costs make solvency more demanding: the set of defaulting banks can be strictly larger under this rule than when partial repayments are allowed. Nonetheless, this case provides the basic intuitions and insights without cluttering the calculations with partial payments. Some of our results do not rely on this assumption, and in particular results from Section 4 on the complexity of bailouts, which only require failure costs to be strictly positive. Our characterizations of systemic solvency and of the optimal bailout policy in some specific networks do leverage this assumption, and are more complicated to state without it, even though the underlying intuitions remain.

3.2 A Characterization of Systemic Solvency

Proposition 1 shows that cycles of debts are vital to the existence of multiple equilibria, and that such multiplicity means the defaults of some banks can be self-fulfilling. In this section, we investigate in more details which banks default, and how it depends on the equilibrium

being considered. We give necessary and sufficient conditions on portfolio values for *all* banks to be solvent, which we call *systemic solvency*.

3.2.1 Balance Conditions

To characterize solvency, and minimal bailouts, it is useful to define the following balance conditions.

We say that a bank i is *weakly balanced* if

$$p_i + D_i^A \geq D_i^L,$$

and that the network is *weakly balanced* if this holds for all i .

Weak balance requires that a bank's assets are enough to cover its debt liabilities, presuming its incoming debt assets are all fully valued. A network being weakly balanced is sufficient for all banks to be solvent in the best equilibrium. This follows since, if all banks but i honor their debt contracts, then i can also pay back its debt fully in a weakly balanced network. Essentially, all debts can be canceled out, irrespective of the network structure. Things are different in the worst equilibrium, as we know from Proposition 1.

We say that a bank i is *exactly balanced* if

$$p_i + D_i^A = D_i^L,$$

and that the network is *exactly balanced* if this holds for all i . An exactly balanced bank has no capital buffer: its assets, if fully valued, are exactly enough to cover its liabilities. Exact balance is a much stronger condition than weak balance, but is very useful as a benchmark condition in characterizing optimal bailouts, as we shall see.

We say that a bank i is *critically balanced* if it is weakly balanced and for each j for which $D_{ij} > 0$,

$$p_i + D_i^A - D_{ij} < D_i^L.$$

The network is *critically balanced* if this holds for all i .

Critical balance lies between exact balance and weak balance, and is also useful as a benchmark condition in characterizing optimal bailouts. It implies that not receiving any one of its incoming debt payments is enough to make a bank insolvent.

3.2.2 Characterizing Systemic Solvency

When there exist cycles of debt, by Proposition 1, the best and worst equilibria can differ. We now fully characterize when these cycles fail to clear.

Determining whether cycles clear at a given vector of portfolio values \mathbf{p} involves an iterative definition of solvency, since a bank being solvent can affect the solvency of other banks.

We say that a bank i is *unilaterally solvent* if $p_i \geq D_i^L$. This means that regardless of whether any of the other banks pay the debts that they owe to i , i is still able to cover its liabilities.

A set of banks S is *iteratively strongly solvent* if it is the union of sets $S = S_1 \cup \dots \cup S_K$, such that banks in S_1 are unilaterally solvent; and then banks in any S_k for $k \in \{2, \dots, K\}$ are solvent whenever they receive the debts from all banks in sets S_1, \dots, S_{k-1} :

$$p_i + \sum_{j \in S_1 \cup \dots \cup S_{k-1}} D_{ij} \geq D_i^L.$$

Note that if N is iteratively strongly solvent, then all banks are solvent in the worst equilibrium. Proposition 2 provides weaker conditions that are necessary and sufficient for systemic solvency. This then provides a base to understand optimal bailout policies.

PROPOSITION 2.

All banks are solvent in the best equilibrium if and only if the network is weakly balanced.

*All banks are solvent in the worst equilibrium if and only if the network is weakly balanced and there exists an iteratively strongly solvent set that intersects each directed (simple) cycle.*²⁵

An implication of Proposition 2 is that in a weakly balanced network, if there is an iteratively strongly solvent set that intersects each cycle, then that implies that the whole set of banks is iteratively strongly solvent. This is the crux of the proof.

The proposition is less obvious than it appears since an insolvent bank can lie on several cycles at once, and could need all of its incoming debts to be paid before it can pay any out. Solvent banks on different cycles could lie at different distances from an insolvent bank, and showing that each bank eventually gets all of its incoming debts paid before paying any of its outgoing debts is subtle. The proof is based on how directed simple cycles must work in a weakly balanced network and appears in the appendix.

Proposition 2 is useful for at least two reasons. First, it highlights the structure of the set of equilibria, which is very useful in our analysis of optimal bailouts. Second, it gives necessary and sufficient conditions to find the worst (and other) equilibria that are easier to check than preexisting algorithms. Example 3 illustrates this.

As all banks have as much debt in as out, they are weakly balanced and hence all solvent in the best equilibrium. Identifying which banks default in the worst equilibrium requires checking the iterative condition. Bank 5 is the only unilaterally solvent bank. Its solvency ensures the solvency of Bank 4, and so there is at least an iteratively strongly solvent set intersecting the left and middle cycles. However, Bank 4 paying back its debt is not enough to make Bank 6 solvent, and so we can stop the iteration there: no iteratively strongly solvent set intersects the right cycle, and all banks on that cycle must default in

²⁵A simple cycle is one that contains any bank at most once. If there is a solvent bank on each simple cycle then there is one on every cycle, since every cycle contains a simple cycle.

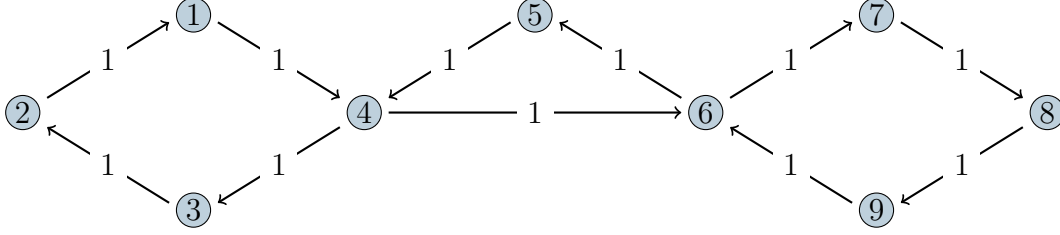


Figure 3: Arrows point in the direction that debt is owed. Let $p_1 = p_2 = p_3 = 0.2$, $p_4 = p_5 = 1$, and $p_6 = p_7 = p_8 = p_9 = 0.5$. The only iteratively strongly solvent set is $\{\{5\}, \{4\}, \{3\}, \{2\}, \{1\}\}$.

the worst equilibrium. Interestingly, this is faster than checking that the *entire* set of banks forms an iteratively strongly solvent set, as we can stop checking banks' solvencies once we have reached key banks that lie at the intersection of multiple cycles (here Banks 4 and 6). It is also faster than algorithms that have been developed to find Nash equilibria of games with strategic complements.²⁶ The worst equilibrium is usually found by iterating on best-responses starting from the lowest strategies for all agents (see Echenique (2007), and references therein), which in our setting boils down to checking whether the whole set of banks is iteratively strongly solvent.

Proposition 2 also implies that if both conditions are satisfied, then there is a unique equilibrium. Conversely, if the network is weakly balanced but there is no iteratively strongly solvent set intersecting every cycle, then there are necessarily multiple equilibria. Thus, in a weakly balanced network, there is a unique equilibrium *if and only if* there exists an iteratively strongly solvent set that intersects each directed (simple) cycle.

Clearly, without some unilaterally solvent banks, having an iteratively solvent set is precluded, and all banks must default in the worst equilibrium. This is not directly implied by Proposition 2, but still follows from the reasoning behind it. To see why this is true, recall the algorithm introduced in Section 3.1 to find the worst equilibrium. The algorithm initiates at values \mathbf{V}^0 that lie below the worst equilibrium values; e.g., $\mathbf{V}^0 = -\mathbf{D}^L$. Since we here consider full failure costs, $\mathbf{d}^A(\mathbf{V}^0) = \mathbf{0}$ and $\mathbf{b}(\mathbf{V}^0, \mathbf{p}) = (\mathbf{d}^A(\mathbf{V}^0) + \mathbf{p})\mathbb{1}\{\mathbf{d}^A(\mathbf{V}^0) + \mathbf{p} < \mathbf{D}^L\} = \mathbf{p}\mathbb{1}\{\mathbf{p} < \mathbf{D}^L\}$. That is, if we start from the pessimistic assumption that all banks are insolvent, then they get no repayments from their debtors. Thus a bank is solvent in the following step of the algorithm if and only if $p_i \geq D_i^L$, that is if and only if it is unilaterally solvent. If no bank is unilaterally solvent, the algorithm stops: assuming that all banks are insolvent is self-fulfilling.

A Sufficient Condition for Iterative Solvency. One way to ensure having an iteratively strongly solvent set intersecting each directed cycle is to have at least one unilaterally solvent bank on each cycle, but this is not always necessary, and so the iterative solvency condition is important. Nonetheless, unilateral solvency is necessary (and sufficient) whenever the banks

²⁶Equilibrium values for banks correspond to Nash equilibrium outcomes of an auxiliary game in which banks choose whether to be solvent ($s_i = 1$) or not, and best responses are $s_i = 1 \iff p_i + \sum_j D_{ij}s_j \geq D_i^L$.

that lie on multiple cycles are critically balanced.

COROLLARY 1. *If all banks that lie on multiple cycles are critically balanced, then all banks are solvent in the worst equilibrium if and only if the network is weakly balanced and there exists a unilaterally solvent bank on each cycle.*

An implication of Corollary 1 is that the iterative portion of the iterative solvency condition only matters when some of the banks that lie at the intersection of several cycles have capital buffers, so that they only need some of their incoming debts to be paid before they become solvent and can make their debt payments. In that case, those banks can be part of a repayment cascade in which payments in one cycle spread to another. As an illustration, recall the network depicted in Figure 1. Bank 2 is the only bank at the intersection of several cycles and is exactly balanced. In the worst equilibrium only Bank 1 is solvent, as there is no unilaterally solvent bank on the right cycle. Suppose instead that $p_2 = 1$. Bank 2 now has enough of a capital buffer so that, even if it gets payments from only one cycle, it can make all of its payments. All banks are then solvent in the worst equilibrium: though Bank 2 is not unilaterally solvent itself, it has enough buffer so that the debt payment initiated by Bank 1 cascades and spreads to the right cycle.

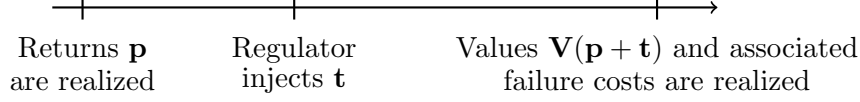
4 Minimum Bailouts Ensuring Solvency in any Given Equilibrium

The results above characterize systemic solvency in the best and worst equilibria. Next, we leverage these results to deduce the minimum bailouts needed to return the whole network to solvency whenever there are some insolvencies. These bailouts are the smallest transfers that avoid all failure costs, and other inefficiencies associated with a dysfunctional financial system, and are optimal in this sense.²⁷ Minimum bailouts depend on the network structure: just as losses can cascade and cycle through the network, the same operates in reverse and well-placed bailouts can have far-reaching consequences.

4.1 The Minimum Bailout Problem

Consider a regulator who can inject capital $(t_1, \dots, t_n) \in \mathbb{R}_+^n$ into each bank in the network, increasing the value of bank i 's balance sheet by t_i . The timeline is as follows. First, returns on outside investments \mathbf{p} are realized. Then, anticipating that some banks will be insolvent absent intervention at \mathbf{p} , the regulator can inject capital into the network. Importantly, the

²⁷If anticipated, these bailouts could distort banks' incentives and lead them to take on more risks ex ante. Such moral hazard problems are well-studied, and so we do *not* argue that the regulator *should* always ensure solvency ex post. We simply acknowledge the fact that regulators often have to intervene to restore solvency when faced with a crisis that has a potential for large cascades. This is thus an important problem, and we analyze what is the most efficient way of doing so.



capital is injected before banks officially start bankruptcy proceedings, and hence before the associated failure costs are incurred.

To ensure systemic solvency at minimum cost, the regulator solves

$$\begin{aligned}
 & \min_{(t_1, \dots, t_n) \in \mathbb{R}_+^n} \sum_i t_i & (\text{OPT}) \\
 & \text{s.t.} \quad \mathbf{V}(\mathbf{p} + \mathbf{t}) \geq \mathbf{0}.
 \end{aligned}$$

Since there can be multiple equilibria for bank values, we need to be precise about which equilibrium is selected in the constraint of (OPT). For simplicity, we suppose that the transfers \mathbf{t} do *not* impact the equilibrium selection, although they conceivably could. Thus, to examine the minimum bailouts preventing defaults in the *best* equilibrium, we always select the *best* equilibrium for bank values $\mathbf{V}(\mathbf{p} + \mathbf{t})$ in (OPT). Similarly, if the goal is to prevent defaults in the worst equilibrium, then optimal bailouts are the minimum transfers such that all banks are solvent if we select the worst equilibrium for $\mathbf{V}(\mathbf{p} + \mathbf{t})$.²⁸ Optimal bailouts are specific to an equilibrium. For instance, it can be that all banks are solvent in the best equilibrium but some default in the worst. Optimal bailouts for the best equilibrium are then null $\mathbf{t} = \mathbf{0}$ since all banks are already solvent, but are strictly positive for the worst equilibrium.

In this section we characterize the minimum bailouts needed to restore solvency of the whole network starting from any given equilibrium. Thus, we take the \mathbf{V} in (OPT) to be relative to some equilibrium selection as a function of the portfolio values, for instance always the worst, or some other selection depending on how an equilibrium is selected by the economic forces. We do not take a stand on such a selection in this paper.

We start by noting that Proposition 2 yields a first characterization of such minimum bailouts for both the best and the worst equilibria, as well as other equilibria.

The best equilibrium is relatively easy to understand. If the network is not weakly balanced then some banks must be defaulting, and each bank that is not weakly balanced needs bailouts to be brought back to solvency. It is thus necessary and sufficient for each bank i to receive its net imbalance $t_i = [D_i^L - D_i^A - p_i]^+$ (by Proposition 2).

Since the minimum injections of capital that ensure systemic solvency in the best equilibrium are fully characterized and relatively easy to calculate, for the remainder of the paper

²⁸More generally, fix any equilibrium of $\mathbf{V}(\mathbf{p})$. Let S the set of solvent banks in that equilibrium absent intervention, and $N \setminus S$ the set of defaulting banks. To keep the equilibrium selection fixed, $\mathbf{V}(\mathbf{p} + \mathbf{t})$ is computed by assuming that banks in S are all solvent, and then finding the maximum set of defaults among the originally defaulting banks.

we focus on the additional capital needed to ensure solvency in other equilibria.

Note that it is weakly optimal for the regulator to start by providing each bank with its net imbalance $[D_i^L - D_i^A - p_i]^+$, as these payments are necessary for solvency in all equilibria and may trigger some repayment cascades. To analyze minimum bailouts for an equilibrium other than the best, it is then without loss of generality to restrict attention to the banks that remain insolvent in that equilibrium once net imbalances have been injected, redefining their portfolio values to account for these transfers and any debt payments they received from solvent banks.

Proposition 2 then tells us that the additional payments needed are the smallest ones that generate an iteratively solvent set of banks that hits each defaulting cycle.

We first point out that even though (OPT) is written as a simultaneous choice of payments, $\mathbf{t} \in \mathbb{R}_+^n$, it is equivalent to specify an *ordered* list of banks to bailout. Indeed, any set of (simultaneous) payments \mathbf{t} leads banks to become solvent in a particular order: first some banks are made unilaterally solvent by the payments; then given their induced debt payments and the bailout payments, some other banks become solvent, and so forth. Reciprocally, any ordered list of bailouts can be implemented via simultaneous payments by setting t_i precisely equal to i 's shortfall given the payments it gets from already solvent banks. Hence, just knowing which banks get payments under the optimal bailout policy is not enough to characterize minimum bailouts as the order in which banks are brought back to solvency matters.

To illustrate this, consider the network depicted in Figure 4, in which all banks default in the worst equilibrium. Since Bank 2 lies on all cycles, bailing it out ensures systemic solvency and costs $D_2^L - p_2 = 5$. This is however not the optimal policy, as the regulator can significantly reduce the cost of bailing out Bank 2 if it first bails out Bank 3. Indeed, bailing out Bank 3 costs $D_3^L - p_3 = 3$, and allows for the repayment of its debt to Bank 2. Bailing out the latter then only costs 1. The optimal bailout policy is then $\mathbf{t}^* = (0, 1, 3)$, which can be equivalently expressed as a sequence of banks that are bailed out $\{3, 2\}$.

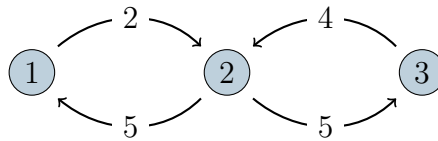


Figure 4: Let $p_1 = 0$, $p_2 = 5$, and $p_3 = 1$.

As mentioned above, the order of bailouts matters, because the bailout cost of remaining banks depends on which banks have already been bailed out. Bailing out 3 and then 2 costs less than the reverse, and both lead to systemic solvency.

4.2 The Computational Complexity of Finding Minimum Bailouts

Proposition 2 implies that ensuring systemic solvency in equilibria other than the best requires that, beyond establishing weak balance, one also needs to inject enough capital so as

to ensure the existence of an iteratively strongly solvent set intersecting all directed cycles on which banks are insolvent. We examine this in what follows.

We begin by showing that this problem is computationally complex in a well-defined way (using concepts from the computer science literature). Precisely, we prove that it is *strongly NP-hard*. The complexity comes from the fact that some of the capital that a bank needs to become solvent can come from the debts paid by others, and so it can be cheaper to first bail out a bank’s debtors rather than bailing it out directly. The number of possible combinations that could be optimal explodes factorially in the number of banks, and there are situations where it is not obvious how to focus on just a small number of those combinations.

After showing the complexity of the problem in general, we give an upper bound on the number of calculations needed to find the optimal bailout policy, and an upper bound on total capital injections. We then explore some specific, but canonical settings where we can provide more insight into the optimal policies, illustrating situations in which one can find approximately optimal policies efficiently, and others where it is not possible.

To state the complexity result, we present some standard definitions, included here for the reader who may not be fully familiar with them.²⁹

A decision problem – a problem with a yes/no answer – that takes as input a network with n banks is in the **NP** complexity class if there exists an algorithm that can *verify* whether any given policy is a solution to the problem in polynomial time, that is there exists a positive constant r such that the verification algorithm runs in $O(n^r)$ -time. Intuitively, problems in NP are problems for which a solution can be easily verified.

Consider for instance the following decision problem: *Does there exist a bailout policy that ensures systemic solvency and costs no more than some budgeted amount W ?* This problem belongs to NP: it is easy to check whether a given bailout policy $(t_i)_i$ ensures systemic solvency and costs no more than W , as this only requires computing $\mathbf{V}(\mathbf{p} + \mathbf{t})$ and $\sum_i t_i$.

A problem is **NP-hard** if it is at least as hard as any problem in NP.³⁰ If a problem is NP-hard, then there is no known polynomial-time algorithm that solves it, and it is believed by many that no such algorithm exists.

A problem is **strongly NP-hard** if it remains NP-hard even when the size of all its inputs is appropriately bounded.³¹

²⁹To keep things to a reasonable length, we do not include all background definitions of how one defines an algorithm, how to count its steps, etc., but such background is easy to find in any text on computational complexity (e.g., [Arora and Barak \(2009\)](#); [Papadimitriou \(1994\)](#)).

³⁰A problem is at least as hard as another if any instance of the second can be translated into an instance of the first in at most polynomially many steps. Alternatively, given an oracle that can solve any instance of that first problem in one step, there exists an algorithm that solves the latter problem in polynomial time.

³¹Many well-studied problems (e.g., the Knapsack Problem, or the Partition Problem) are only *weakly* NP-hard, which means that these problems become hard to solve and approximate only when the size of their inputs becomes large, in a well-defined sense. For instance the Partition Problem asks whether a given set of n positive integers can be partitioned into two subsets that have the same sum. If the possible integer values that are in the given set are sufficiently limited, then the problem can become quite easy. For example, if all these integers are 0s or 1s, then the problem is trivial as it boils down to checking whether their sum is even. More generally, if these integers are bounded above by some value cn , then each subset of the partition can

From our perspective, the important fact is that every strongly NP-hard problem has instances for which finding exact or even approximate solutions involves impractically many (more than polynomially many in the size of the problem) calculations.

PROPOSITION 3. *Finding whether there exists a bailout policy that ensures systemic solvency and costs no more than some budgeted amount is strongly NP-hard. Thus, finding a minimum cost bailout policy (**OPT**) is also strongly NP-hard.*³²

The optimal bailout policy not only accounts for the direct payments that some bank’s solvency induces, but also the value of the indirect cascades that ensue. We prove Proposition 3 by showing that for some network structures, finding whether one can make all banks solvent at no more than a certain cost enables one to solve the “Largest Minimal Vertex Cover Problem,” which is known to be strongly NP-hard and hard to even approximate.³³

Given some undirected graph, a vertex cover is a set of vertices that contains at least one endpoint of all edges in the graph. A vertex cover is minimal if it is not the superset of another vertex cover. It turns out that, for some networks, the optimal bailout policy consists in bailing out banks belonging to a minimal vertex cover of maximum size. To be more specific, consider any undirected graph, and interpret an edge between i and j in that network as bilateral claims of 1 that i and j have on each other (i.e., $D_{ij} = D_{ji} = 1$). So any edge ij generates a cycle between banks i and j in the financial network. Let $p_i = p \in (0, 1)$ for all banks, such that the network is *critically* balanced: Banks’ outside assets are not enough to absorb the default of any one of their counterparties. We know that, to ensure solvency of all, the regulator must bailout at least one bank per cycle, and so the set of bailed out banks forms a vertex cover. We furthermore show that this vertex cover has to be minimal (as otherwise the regulator is bailing out banks that need not be bailed out) and that overall bailout costs sum up to the number of edges net of the value of outside assets of all bailed out banks. Hence an optimal bailout policy consists in bailing out banks in a minimal vertex cover of *maximal* size, as this allows the regulator to leverage the outside assets of more banks. This means any Largest Minimal Vertex Cover Problem can be translated into

sum to at most $0.5cn^2$, and an algorithm could check all these possible values exhaustively fairly quickly. So at least some of the n integer values that are in consideration have to be “large” for the Partition Problem to be hard. In contrast, the Vertex-Covering Problem that we use in our proof has no integer values associated with it: it considers an undirected network on n nodes and asks what is the size of the smallest set of nodes such that every link the network has at least one of its endpoints in the set. Importantly, even though there are no integer values associated with the specification of this problem, the problem can be much harder to solve in that number of potential sets of nodes that one has to check explodes at a rate of $n!$, and unless the network is special (e.g., a lattice, bipartite) there are no known simple shortcuts.

³²Checking whether there exists a bailout policy that ensures systemic solvency and costs no more than some budgeted amount is an “easier” problem than finding a minimum cost policy since knowing an optimal policy enables one to answer the question of whether it can be done within some budget. Thus, showing that the decision problem is strongly NP-hard establishes that (**OPT**) is as well, and working with decision problems is a standard technique.

³³Boria et al. (2015) show that, unless $P = NP$, this problem cannot be approximated by a polynomial time algorithm within ratio $n^{\varepsilon-0.5}$, for any $\varepsilon > 0$.

a bailout problem on an appropriately constructed network, and so bailout problems are at least as hard as Largest Minimal Vertex Cover Problems, in a well-defined way.

We emphasize, however, that finding the optimal bailout policy is, in more general networks, even more complicated than the Largest Minimal Vertex Cover Problem. As banks are bailed out, they repay their debts and change the balance sheets of each other, thus altering the remaining network of insolvent banks.³⁴ This adds another layer of complexity,³⁵ and means that the number of bailout policies that the regulator would have to compare absent a better algorithm is of the order of $n!$, which very quickly gets too large for any computer to handle. With just 15 banks the problem already involves trillions of possible bailout strategies, and with 20 banks more than 10^{18} . Furthermore, as the order of bailouts matters, (OPT) is not easily expressed as a linear program, and standard dynamic programming algorithms that have proved effective for NP-hard problems do not directly extend.³⁶

4.3 Upper Bounds on Computation Time and Capital Injections

4.3.1 An Upper Bound on Computation Steps

Proposition 3 shows that, as the size of the network n grows, finding the optimal bailout policy becomes complex. The intuition is that a network with more banks can have more overlapping cycles, and these cycles generate interferences between bailouts. So what drives the complexity of finding the optimal bailout policy is not the number of banks n per se, but the associated possible number of cycles K . We formalize this in Proposition 4.

PROPOSITION 4. *The number of calculations needed to find the optimal bailout policy is no more than $K!n^{K+1}$, where n is the number of banks in the network and K is the number of dependency cycles.*

Proposition 4 builds from the following observation: bailing out more than one bank per cycle cannot be strictly cheaper than bailing out only one per cycle, and both ensure systemic solvency. Hence the regulator can restrict attention to policies that bail out at most one bank per cycle. Given potential overlap in cycles and resulting cascades, one needs to do more than just pick the cheapest bank per cycle. A brute-force algorithm to find the optimal bailout policy is then to compute the cost of all such policies, and pick the cheapest one. There are $K!$ different orders in which the cycles can be cleared, with at most n different banks per cycle, leading to at most $K!n^K$ policies (a bank per cycle and the order of their

³⁴Bailing out a bank also changes the indirect values of bailing out others. For example, if bailing out 1 induces a debt payment to 2, then that lowers the extent to which bailing out 3 makes 2 solvent, since some of 2's imbalance has already been taken care of.

³⁵This makes the bailout problem an interesting class of problems for further study in complexity: it is easy to check that any given bailout policy works and involves no more than a certain cost, and so this problem is strongly NP-complete, and yet provides an interesting twist on well-studied problems given that combinations of bailouts changes the cost and value of other bailouts.

³⁶One can expand the problem to have different values for every ordering, but then the linear problem has factorially many inputs as a function of n .

clearance). The computation of the cost of each policy takes at most n steps, and so this brute force algorithm can take at most this many steps (see the appendix for a formal proof). In networks with large cycles, an exhaustive search approaches this number of steps.

Proposition 4 implies that when the number of cycles is small, the problem remains tractable even if the number of banks is very large. For instance, if there is some bounded number of cycles so that K is small and fixed, then the brute-force algorithm described above runs in $O(n^{K+1})$ -time as n becomes large. In contrast, finding the optimal bailout policy is most complex when there are many overlapping cycles as for instance in cliques, and we discuss those in Section 5.3. In settings with cliques, K becomes exponential in the size of the cliques and thus can become exponential in n .

4.3.2 An Upper Bound on Bailout Costs

Even though finding the optimal bailout policy can be complex, we can provide an upper bound on the total injection of capital that is necessary to bring the network to solvency. As discussed above, we assume that net imbalances $\sum_i [D_i^L - D_i^A - p_i]^+$ have already been injected, so this is a bound on the additional injections needed to prevent self-fulfilling defaults. We also provide a simple algorithm that ensures systemic solvency, and leads to total bailout costs that never exceed this bound. This is helpful since we know from Proposition 3 that the minimum bailout problem is hard to approximate, and that no known algorithm performs well.

The algorithm that we propose leverages the idea that bailing out a bank i has an indirect value: it brings i 's counterparties closer to solvency, or makes them solvent altogether.³⁷

Define the first-step indirect bailout value of a bank i as $\sum_j \min\{D_{ji}, D_j^L - p_j\}$: it captures by how much i 's solvency reduces the bailout cost of other banks in the network. Note that this is a near-sighted measure of indirect bailout value as it only accounts for a bank's *direct* payments, but not for the fact that i 's solvency can induce the solvency of one of its creditors j , which then repays its debts, and so forth. We can thus define a k th-step indirect bailout value of bank i by all the cascades of payments that are induced by the bailout of i up to k iterations of induced solvencies. We provide formal definitions in the appendix, together with the proof of Proposition 5.

When deciding whether to bailout i , the regulator must trade-off a bank's indirect bailout value with the cost of its bailout, $(D_i^L - p_i)^+$. Consider the greedy algorithm that bails out the defaulting bank with highest ratio of its k th-step indirect bailout value to its bailout cost (for any $1 \leq k \leq n - 1$) until all are solvent, recomputing these values after each step to account for all new solvencies.³⁸ We show that this algorithm leads to total bailout costs

³⁷The indirect bailout value differs from the concept of threat index of Demange (2016), which captures the impact of marginally increasing a bank's repayments on total debt repayments, assuming the set of defaulting banks remains the same. A key difference is that our notion of indirect bailout value is not calculated on the margin, and the value of bailing out a bank propagates further if we account for induced changes in the solvency status of other banks.

³⁸If there are any ties – i.e., more than one bank with the highest indirect bailout value to bailout cost

that never exceed the following bound.

PROPOSITION 5. *For any $1 \leq k \leq n - 1$, bailing out banks in decreasing order of the ratio of their k th-step indirect bailout value over bailout cost until all are solvent leads to a cost of at most*

$$\frac{1}{2} \sum_i (D_i^L - p_i)^+.$$

Thus, the optimal bailout cost is no more than this, and this bound is reached in some networks.

When following this algorithm, the regulator is guaranteed to inject at most half of banks' total shortfall in order to ensure solvency of all banks. This bound holds for all equilibria, and can be made tighter by only summing over the banks that default in the equilibrium of interest, instead of summing over all banks. The bound is reached whenever the network is composed of $n/2$ disjoint cycles, and banks are all equally costly to bailout – so $(D_i^L - p_i)^+ = (D_j^L - p_j)^+$ for all i, j .

This upper bound is not straightforward. Consider, for instance, a clique of critically balanced banks, such that all banks have claims on each other and none of them has enough capital buffer to sustain the default of some of its counterparties. Then, to ensure systemic solvency (in the worst equilibrium) the regulator has to bailout *all banks but one*. What Proposition 5 shows is that, even in such a network, total bailout costs do not add up to more than half of banks' total shortfall: the optimal policy leverages the fact that payments from banks that have already been bailed out reduce the cost of future bailouts.

Nonetheless, Proposition 5 does not imply that following the above algorithm always finds an optimal bailout policy. For instance, in the example from Figure 4 it would begin with Bank 2 rather than Bank 3 and would overpay. In other instances it could overpay by arbitrarily large amounts and so does not approximate the optimal bailout policy well, as we show in Section 5.2 below. The same is true of an algorithm that chooses banks in the order of the indirect values that their bailouts generate.³⁹

Similarly, algorithms that bail out banks in increasing order of their bailout cost can also lead to total injections of capital arbitrarily larger than necessary. For example, the network in Figure 5 illustrates how badly a greedy algorithm based on minimizing costs can perform. The example consists of a chain of cycles of length n . Note that banks $i \geq 2$ have enough capital buffer to absorb the default of their smallest debtor, that is $i + 1$. Hence if a bank i is solvent, then this is enough to make its follower $i + 1$ solvent as well, which is then enough to make $i + 2$ solvent, and this unravels until Bank n . The reverse is, however, not true: Bank i repaying its debts is not enough to make its predecessor $i - 1$ solvent. Given this asymmetry, the optimal policy is to bailout Bank 1: this guarantees solvency of all for a total cost of

ratio – then break the ties arbitrarily.

³⁹Maximizing the indirect value divided by the cost, or just the indirect value, can lead the regulator to bail out the center bank in a star network if it has the highest debt liability/cost ratio, while bailing out some peripheral banks can lead to (arbitrarily) lower total bailout costs.

\bar{D} . Now suppose the regulator uses a simple greedy algorithm, which consists in bailing out banks in increasing order of their bailout cost. The algorithm bails out Bank n first, as this only requires injecting $D_n^L - p_n = \underline{D} < \bar{D} = D_i^L - p_i$ for all $i \neq n$. The algorithm then bails out Bank $n-1$ at a cost of $D_{n-1}^L - p_{n-1} - D_{n-1n} = \bar{D} - \underline{D}$, and then Bank $n-2$, etc. This leads the regulator to inject a total of $\underline{D} + (n-1)(\bar{D} - \underline{D})$. Hence, the performance of the greedy algorithm relative to the optimal policy can be arbitrarily bad for long enough chains.

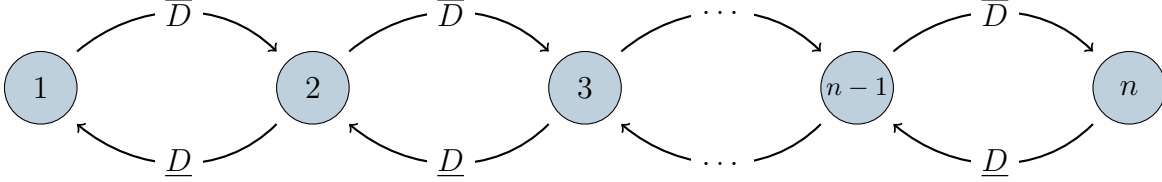


Figure 5: Let $p_1 = p_n = 0$ and $p_i = \underline{D}$ for all $i \neq 1, n$. Suppose $\bar{D} > \underline{D} > 0$.

As the example helps illustrate, the complexity arises from the fact that cycles overlap, and that a repayment cascade in one cycle can spread to another. More generally, as banks are bailed out, they can either increase or decrease each other's indirect bailout values. On the one hand, their bailouts and debt payments bring others closer to solvency and thus reduce how much they still need, but then also make it easier to cause them to return to solvency and induce subsequent payments. So, a bank's bailout can affect others' indirect bailout values in either direction. This becomes especially complex when cycles overlap.

The optimal bailout problem is easier in networks in which interferences between cycles are limited. This is for instance the case if cycles do not overlap, and we fully characterize the optimal bailout policy in such networks in Section 5.1. This is also the case if all banks that lie on multiple cycles are critically balanced – i.e., have no capital buffer – as then a repayment cascade cannot spread from one cycle to another (Corollary 1).⁴⁰

5 Some Prominent Network Structures

We showed that finding the optimal bailout policy can be both computationally hard, and hard to approximate. In some financially relevant cases, however, the analysis simplifies and provides insights regarding the structure of optimal bailouts. In this section, we first consider networks in which cycles of banks do not overlap, and then examine networks that have a core-periphery structure. A main takeaway of our analysis is the importance of indirect bailouts: to ensure the solvency of a large core bank, it is generally cheaper to bailout small peripheral banks that owe money to it rather than to bailout the core bank directly.

⁴⁰See Online Appendix B.3 for an analysis of critically balanced networks.

As in the previous section, we presume weak balance is satisfied and restrict attention to the subnetwork of insolvent banks, accounting for all the debt payments that they receive from solvent banks.

5.1 Disjoint Cycles

The first set of networks that we examine are those in which the cycles in the network are disjoint, so that no bank lies on more than one cycle.

There can still be banks that lie on no cycles, but for instance lie on directed paths coming out from some bank on a cycle. There can also exist a directed path that goes from one cycle to another, as long as there is no directed path that goes back (which would violate each bank lying on at most one cycle).⁴¹

Let c_1, \dots, c_K be the simple cycles in the network, where c_k denotes the set of banks in the k -th cycle. Order the cycles so that if $i \in c_k$ and $j \in c_{k'}$ for $k' > k$, then there is no directed path from j to i .⁴²

The optimal bailout policy is described as follows. Pick the bank on c_1 that is the cheapest to bailout:

$$\min_{i \in c_1} D_i^L - p_i.$$

This then ensures that all banks in that cycle are solvent, and can also lead to further solvencies as the banks on that cycle may owe debts to banks outside of that cycle. Let S_1 denote the set of banks that are made solvent if all banks on c_1 are solvent (which is the same regardless of which bank on c_1 is bailed out). If this includes banks on any other cycles, then that cycle will be fully solvent as well. Consider the smallest k such that $c_k \cap S_1 = \emptyset$. Accounting for all the payments that have come in from S_1 , find the cheapest bank to bailout on c_k :

$$\min_{i \in c_k} D_i^L - p_i - \sum_{j \in S_1} D_{ij}.$$

This induces solvency on c_k and additional banks, letting S_2 denote the set of banks made solvent by this bailout and its cascade. Iteratively, after h such steps, one finds the lowest indexed insolvent cycle k_h and finds the cheapest bank on that cycle to bailout:

$$\min_{i \in c_{k_h}} D_i^L - p_i - \sum_{j \in S_1 \cup \dots \cup S_h} D_{ij}.$$

After at most K steps the full network will be solvent.

⁴¹Note that there cannot exist banks with no incoming debts but with outgoing debts, since those are already solvent under weak balance.

⁴²Given that i and j lie on different cycles, there cannot be both a directed path from i to j and one from j to i , as that would imply that these banks lie on multiple cycles. Thus, there must exist such an order, and in fact there can be multiple such orders. Pick any one satisfying this condition, as all such orders will lead to the same bailout costs.

PROPOSITION 6. *Suppose that each bank lies on at most one cycle. Then, the optimal bailout policy consists of bailing out the bank closest to solvency on each simple cycle, accounting for bailouts of previous cycles, as described above.*

5.2 Star Networks

A star network is composed of one center bank, Bank n , that is exposed to all banks in the periphery: $i \in \{1, \dots, n-1\}$. Each bank i in the periphery is exposed only to the center bank. A star network is then a simple example of a core-periphery network, in which there is only one core bank. We first consider star networks as they provide part of the intuition needed for the analysis of more general core-periphery networks.

The optimal bailout policy is more tractable in star networks than in some other networks as only one bank, the center bank, lies on multiple cycles. This reduces the complexity of the problem: finding the optimal bailout policy is no longer *strongly* NP-hard and becomes equivalent to the Knapsack Problem.⁴³ The Knapsack Problem is still an NP-hard problem, but there exist several algorithms that approximate it arbitrarily well. Hence our analysis suggests that these types of algorithms should be considered when deciding which peripheral banks to bailout in core-periphery networks.

If the star network also satisfies some symmetry conditions, then the optimal bailout policy can be fully characterized and needs not be approximated. Let all peripheral banks be symmetrically exposed to the center bank: they all have a debt claim of $D_{in} = D^{out}$ on the center bank, and a liability $D_{ni} = D^{in}$ to it. So the total debt assets and liabilities of the center bank equal $D_n^A = (n-1)D^{in}$ and $D_n^L = (n-1)D^{out}$, respectively. As before, let the network be weakly balanced, and suppose none of the banks are unilaterally solvent. If the center bank were unilaterally solvent, the whole network would always clear, and regulatory intervention would be unnecessary. Similarly, if some peripheral banks were unilaterally solvent, then they could pay back their debt to the center bank for sure, and we could redefine the network to account for these payments. See Figure 6 for an example of a star network.

Bailing out the center bank costs $(n-1)D^{out} - p_n$ and clears the whole system, as the center bank lies on all cycles. However, unless all peripheral banks have $p_i = 0$, this is not optimal and can be much costlier than optimal.

To understand the optimal bailout policy, let

$$m^* = \frac{(n-1)D^{out} - p_n}{D^{in}}.$$

⁴³A Knapsack Problem consists of n items that each have a weight and a value, and a knapsack that has some limit on the total weight it can hold. The goal is to select which items to pack so as to maximize the total value, while not exceeding the weight limit. Our problem is similar as, intuitively, each peripheral bank has a value – the amount of liquidity flow into the core induced by its solvency, – and a cost – how much capital the regulator needs to inject to make it solvent. The goal is then to minimize the total cost of bailed out peripheral banks while inducing sufficient liquidity flow into the core to ensure its solvency.

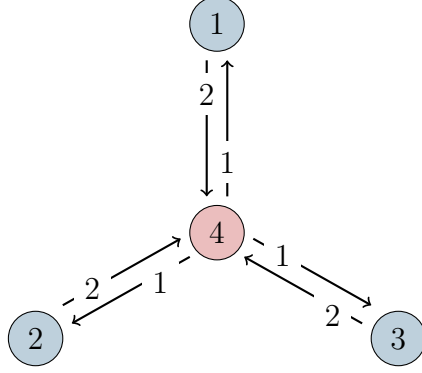


Figure 6: A star network with three peripheral banks, $D^{in} = 2$, and $D^{out} = 1$.

Without loss of generality, let the peripheral banks be indexed in decreasing order of p_i .

The optimal policy always involves first bailing out the $\lfloor m^* \rfloor$ peripheral banks with the highest portfolio values. This leverages their p_i 's and gets an amount of liquidity $\lfloor m^* \rfloor D^{in}$ into the center bank at a potentially much lower cost than paying it directly to the center. This then just leaves a comparison between the marginal shortfall of the center bank

$$(n-1)D^{out} - p_n - \lfloor m^* \rfloor D^{in}$$

and the cost of bailing out the $\lfloor m^* \rfloor + 1$ -st peripheral bank

$$D^{in} - p_{\lfloor m^* \rfloor + 1}.$$

It is optimal to do the remaining bailout via the center bank instead of the $\lfloor m^* \rfloor + 1$ -st peripheral bank if and only if⁴⁴

$$(n-1)D^{out} - p_n - \lfloor m^* \rfloor D^{in} \leq D^{in} - p_{\lfloor m^* \rfloor + 1}.$$

PROPOSITION 7. *The optimal bailout policy is: to bailout the first $\lfloor m^* \rfloor + 1$ peripheral banks if*

$$(n-1)D^{out} - p_n - \lfloor m^* \rfloor D^{in} > D^{in} - p_{\lfloor m^* \rfloor + 1},$$

and otherwise to bail out the first $\lfloor m^ \rfloor$ peripheral banks, and then inject $(n-1)D^{out} - p_n - \lfloor m^* \rfloor D^{in}$ into the center bank.*

The proposition implies that it is always best to start by bailing out peripheral banks, so as to leverage their outside assets and reduce the overall cost of bailouts. Then, once only one more peripheral bank's payment is needed to return the center bank to solvency, the regulator either injects the center bank's marginal shortfall or bails out one last peripheral bank, depending on which is cheaper.

⁴⁴Note that if m^* happens to be an integer, then the center bank is already solvent.

An implication of Proposition 7 is that mistakenly bailing out the center bank when the optimal policy is to target peripheral banks can be much costlier to the regulator than the reverse. Indeed, this can lead the regulator to waste the value of the outside assets owned by peripheral banks. In contrast, the only reason why bailing out peripheral banks may not be optimal is if this leads the regulator to bailout “one too many” of them due to the integer constraint, which implies a waste of capital of at most D^{in} .

COROLLARY 2. *Bailing out all peripheral banks leads the regulator to inject at most an extra D^{in} compared to the total amount required by the optimal bailout policy.*

Bailing out the center bank leads the regulator to inject at most an extra $\sum_{i=1}^{\lceil m^ \rceil} p_i$ compared to the total amount required by the optimal bailout policy.*

Thus, in situations where the portfolio values among the needed peripheral banks ($\sum_{i=1}^{\lceil m^* \rceil} p_i$) is much larger than the amount any one of them owes to the center bank (D^{in}), bailing out the periphery is a much safer policy in terms of wasted capital.

5.3 Core-Periphery Networks and Cliques

Many financial systems are well-approximated by a core-periphery structure: they can be decomposed into a set of densely connected core banks and a set of more sparsely connected peripheral banks. When we move to core-periphery structures, the above approach of beginning with peripheral banks carries over, but then the remaining bailout problem of the core itself can be substantially more complex (i.e., strongly NP-hard). We begin with cases in which there is some symmetry in the core, and in which intuitions can be drawn and optimal bailouts fully characterized, and then we discuss the asymmetric case.

Let N_C and N_P be the set of core and peripheral banks, respectively, with $N = N_C \cup N_P$ and $N_C \cap N_P = \emptyset$. The $n_C = \#N_C$ banks in the core form a clique and are symmetrically exposed to each other: $D_{ij} = D^{core}$ for all $i \neq j \in N_C$. Each core bank $i \in N_C$ is also exposed to a different subset of the peripheral banks $N_P^i \subseteq N_P$, and each of these peripheral banks are only exposed to the core bank i . Hence $N_P = \cup_{i \in N_C} N_P^i$ and $\cap_{i \in N_C} N_P^i = \emptyset$. Specifically, peripheral banks in N_P^i have a debt claim of D^{out} on i , and a debt liability of D^{in} to i . Suppose also that each core bank i is exposed to the same number of peripheral banks $n_P = \#N_P^i$ for all i .

The portfolio values of core banks are heterogeneous and denoted by p_i and, for simplicity, the smaller peripheral banks all have portfolios valued at p_P .

If the network is a clique, so that there is no peripheral banks and $n_P = 0$, then the optimal bailout policy is straightforward: it simply consists in bailing out banks in decreasing order of their portfolio value p_i until they are all solvent.

The added complication from the periphery comes from the fact that peripheral banks change the bailout costs of core banks. However, note that we can combine the logic from star networks, as in Proposition 7, with the logic from cliques. In particular, consider the following bailout policy.

First, pick the core bank with the highest portfolio of outside assets p_i , and let

$$m_i^* = \frac{(n_C - 1)D^{core} + n_P D^{out} - p_i}{D^{in}}.$$

We can then follow the logic of Proposition 7 to return this bank to solvency. Start by bailing out $\lfloor m_i^* \rfloor$ of its peripheral banks, and then bail out one more of its peripheral banks if

$$(n_C - 1)D^{core} + n_P D^{out} - p_i - \lfloor m_i^* \rfloor D^{in} > D^{in} - p_P,$$

and inject the remaining $(n_C - 1)D^{core} + n_P D^{out} - p_i - \lfloor m_i^* \rfloor D^{in}$ directly into bank i otherwise.⁴⁵

Iteratively, if $x > 0$ core banks have already been returned to solvency and the system is not yet at full solvency, then next consider the core bank with the highest p_i among those still insolvent. Define

$$m_i^*(x) = \frac{(n_C - 1 - x)D^{core} + n_P D^{out} - p_i}{D^{in}}.$$

Start by bailing out $\lfloor m_i^*(x) \rfloor$ of its peripheral banks, and then bail out one more of its peripheral banks if

$$(n_C - 1 - x)D^{core} + n_P D^{out} - p_i - \lfloor m_i^*(x) \rfloor D^{in} > D^{in} - p_P,$$

and otherwise inject the remaining $(n_C - 1 - x)D^{core} + n_P D^{out} - p_i - \lfloor m_i^*(x) \rfloor D^{in}$ directly into bank i .⁴⁶ Continue until all banks are solvent.

PROPOSITION 8. *The above policy is optimal.*

Hence the intuition from Proposition 7 extends to more general core-periphery networks: it is always optimal to start by bailing out a core bank's peripheral counterparties instead of targeting it directly.

For tractability, we imposed a symmetry assumption on the network: the only heterogeneity came from the difference between core and peripheral banks. When banks are heterogeneous within the core, then finding the optimal order in which to bring core banks to solvency can be quite complex, and in fact can again nest a vertex covering problem. Nevertheless, as long as core banks are sufficiently large compared to peripheral banks, it remains optimal to first target a core bank's peripheral banks instead of bailing it out directly. Hence one part of the optimal bailout policy is simple, but the other part is not. The following asymmetric example illustrates this.

⁴⁵If $\lfloor m_i^* \rfloor \geq n_P$, then first bail out all of i 's peripheral banks and then inject the remaining amount into i that is needed to make it solvent.

⁴⁶Again, if $\lfloor m_i^*(x) \rfloor \geq n_P$, then bail out all of i 's peripheral banks, and then inject the remaining amount into bank i needed to make it solvent.

5.3.1 A 2×2 Core-Periphery Network with Asymmetries

Consider a network with two core banks $i = 2, 3$, and two peripheral banks $i = 1, 4$. Core banks have claims on each other, and a claim on their peripheral bank. Each peripheral bank has a claim on its core bank. The financial network thus contains three simple cycles of claims—one between the two core banks, and one between each core bank and its peripheral bank—as depicted in Figure 7. The network is weakly-balanced, so all banks are solvent if

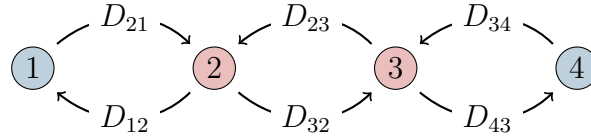


Figure 7: Arrows point in the direction that debt is owed, that is from debtors to creditors. There are two core banks (pink) and two peripheral banks (blue).

they get their debt paid back in full.

Suppose that peripheral banks are “small,” in the sense that them repaying back their debt is never enough to ensure the solvency of their core bank.⁴⁷ Then it is always cheaper to start by bailing out a peripheral bank before bailing out its core bank. Perhaps counterintuitively, it is only when a peripheral bank can induce a cascade that it might not be optimal to bail it out. Indeed, if the peripheral bank’s payment to the core bank is more than enough to make the latter solvent, then the regulator might “overpay” by bailing out the peripheral bank because the cost of bailout might be more than the shortfall in the core bank. However, if a peripheral bank is small, then the regulator cannot be overpaying by bailing it out first, but can only gain from doing so.

From now on, assume that peripheral banks are small. Then the optimal bailout policy must start by bailing out a peripheral bank. There are then four candidate policies: $(1, 2, 3)$, $(1, 2, 4)$, $(4, 3, 2)$, and $(4, 3, 1)$. Even in this simple network, there is no systematic method to determine in which order core banks should be made solvent (the first two policies make Bank 2 solvent first, and then Bank 3, whereas the last two policies do the reverse.) What is true, however, is that making a core bank a larger debtor (i.e., increasing its D_i^L) while keeping its shortfall $D_i^L - p_i$ constant always favors bringing that core bank back to solvency before other core banks. That is, if under some initial \mathbf{D}, \mathbf{p} , the optimal bailout policy makes Bank 2 solvent before Bank 3, then that order remains optimal if D_2^L and p_2 are both increased by the same amount. Indeed, this increases the indirect bailout value of Bank 2 without changing how much liquidity it needs, making it more likely that bailing it out before Bank 3 is optimal.

⁴⁷Formally, $p_2 + D_{21} < D_2^L$ and $p_3 + D_{34} < D_3^L$.

6 Discussion

6.1 Recovery of Capital Injections

Our analysis distinguishes between the minimum bailouts needed to ensure systemic solvency in the best equilibrium, and those needed to avoid additional self-fulfilling defaults in non-best equilibria. We showed that the former are easy to characterize as they do not depend on the specific network structure and only require rebalancing each bank’s portfolio, whereas the latter depend on network cycles in complex ways. Another feature that distinguishes bailing out the best equilibrium versus others is that payments made to achieve weak balance cannot be recovered by whomever intervenes in the bailout, at least in the short run.⁴⁸ In contrast, additional payments made to ensure solvency in other equilibria can be recovered, at least theoretically, once they have cycled through the network. These could thus be offered, for instance, as short-term loans.

Even though these additional bailouts can, in theory, be recovered, doing so may be infeasible in practice for a variety of reasons. Furthermore, even if the regulator does manage to recoup a large fraction of the bailout money, injecting it requires a lot of capital ex ante, which can be costly in itself. Hence finding the minimum bailouts that ensure systemic solvency in the worst equilibrium is policy relevant, even though some of the funds used in these bailouts can eventually be recuperated.⁴⁹ An extension of our analysis is to assess the risks of recovering funds from different banks, and accounting for those differences in the cost of a bailout, so that each bank has an adjusted cost of funds injected into it.

6.2 Bailouts with a Limited Budget

We considered the problem of ensuring solvency of all banks at minimum cost to the regulator. We showed that ensuring systemic solvency in the best equilibrium is fairly simple, and hence focused on studying non-best equilibria. In some settings, however, the regulator faces a related but distinct problem: it has a fixed bailout budget W to allocate across banks and wants to minimize defaults and deadweight losses in some equilibrium of interest. If the budget W is low enough such that not all defaults can be avoided, then many of the results and intuitions that we derived for non-best equilibria also apply to the best equilibrium. In particular, finding a bailout policy that minimizes defaults in the best equilibrium is (strongly) NP-hard when there is a limited budget. Indeed, because of the budget constraint, there is a trade-off between the indirect value associated with a bank’s bailout and its cost, that parallels the analysis above. Finding optimal bailouts is again challenging since the order of bailouts matters, and becomes similar to the above problem of finding the minimum amounts of payments that bring, in this case the most, banks back to solvency.

⁴⁸This is in the context of our static setting. It could be that once returned to solvency, banks’ future profits will enable them to pay back current shortfalls.

⁴⁹See [Lucas \(2019\)](#) for a detailed look at the total bailout money injected, and the amount that was not recovered, in the 2008 crisis.

The intuition underlying Proposition 7 and 8 also extends to the best equilibrium. Indirect bailouts are generally the most cost efficient: Bailing out a bank’s creditors first instead of targeting it directly leads to lower total bailout costs for the same set of induced solvencies.

6.3 The Benefits of Netting, Compression, and CCPs

Our analysis has made clear the role of cycles in generating multiple equilibria for bank values. This implies that there are potential gains from clearing cycles in the network, as this can eliminate the possibility of a market freeze and reduce this form of systemic fragility. In practice, this resonates with liquidity-saving mechanisms as well as portfolio compression. Liquidity-saving mechanisms are settlement systems that allow banks to condition their payment on the receipt of another payment (Martin and McAndrews (2008)). When offsetting payments are in the queue, the system can clear them, thus preventing a potential freeze in which none of the parties pays back the others. Portfolio compression, which has become increasingly popular since the 2008 financial crisis, allows banks to eliminate offsetting obligations with other organizations, exploiting cycles in the financial network. Portfolio compression thus reduces gross interbank exposures while keeping net exposures constant, which not only reduces systemic risk but regulatory requirements of participants as well.⁵⁰

Nevertheless, portfolio compression is still limited in practice. Compression services are currently only available for certain securities, mostly derivatives, and do not encompass multilateral nettings of all types of obligations between financial institutions. Even within markets for which compression services are available, there are large opportunities for compression that are not currently exploited. For instance D’Errico and Roukny (2019) estimate that up to three quarters of the notional value in the CDS contracts traded by European institutions could be compressed. This is all the more surprising as the complexity and entanglements of CDS contracts contributed to the turmoil and freeze of interbank debt markets in 2007.⁵¹

Several reasons help explain why financial markets are far from being fully compressed. The network is complex, with many overlapping and long cycles, and so offsetting obligations requires participation of many parties, with full opening of their books to a common entity, and banks may fail to coordinate as needed. Contracts also have staggered maturities, covenants, and priorities that further complicate netting since they are not fully comparable, and thus have to each be priced and closed out. Finally, banks may not want some positions to be fully compressed, or exposures to new counterparties to be created in the netting process, as they may prefer maintaining preexisting relationships with specific counterparties (D’Errico and Roukny (2019)).⁵² Furthermore, banks may prefer to hold offsetting long-term

⁵⁰In some cases, portfolio compression can however worsen the default cascade triggered by a shock, if failure costs are low and banks heterogeneous enough (Schuldenzucker and Seuken (2019)).

⁵¹See “Credit derivatives: The great untangling”, *The Economist*, Nov 6th 2008.

⁵²Note also that some contracts may provide incentives for counterparties to monitor the activities of another bank, and those incentives can disappear with netting.

debts with others, as such cycles allow them to insure against un-contractable liquidity shocks (Donaldson and Piacentino (2017)). Such preferences matter in practice: for example, some of the current providers of compression services ask participants their tolerance to portfolio reconfiguration beforehand and, naturally, this restricts the extent to which the network can be compressed.

In any case, our analysis provides a base to estimate the system-wide gains associated with netting: for any distribution over returns to outside investments, one can compute by how much compressing the network reduces overall expected bailout costs, for any given equilibrium.⁵³ As discussed above, there are benefits associated with having cycles in the network that would have to be valued and then would account for the other side of the calculation.

Finally, the use of Central Counterparty Clearing Houses (CCPs) can also help mitigate some of these issues, as the resulting star-like network eliminates many cycles. However, one has to worry about providing the CCPs with appropriate incentives and should be concerned about their extreme centrality and size.⁵⁴ Large government-sponsored enterprises that process huge amounts of securities have an uneven history of success, especially if one examines Fannie Mae and Freddie Mac’s failures in the 2008 crisis. More generally, considering regulations that change the network is an important topic for further research, but requires modeling the endogenous formation and benefits from the network, which is beyond the scope of our analysis (e.g., see Erol and Vohra (2018); Erol (2019)). Regardless of the precise policy that one undertakes, developing and maintaining a more complete picture of the network, and the portfolios of banks together with those of their counterparties, is a necessary first step to improving crisis management.

References

- ACEMOGLU, D., A. OZDAGLAR, AND A. TAHBAZ-SALEHI (2015): “Systemic Risk and Stability in Financial Networks,” *The American Economic Review*, 105, 564–608.
- ACHARYA, V. V., S. T. BHARATH, AND A. SRINIVASAN (2007): “Does industry-wide distress affect defaulted firms? Evidence from creditor recoveries,” *Journal of Financial Economics*, 85, 787–821.
- ARORA, S. AND B. BARAK (2009): *Computational complexity: a modern approach*, Cambridge University Press.
- BARUCCA, P., M. BARDOSCIA, F. CACCIOLI, M. D’ERRICO, G. VISENTIN, S. BATTISTON, AND G. CALDARELLI (2016): “Network valuation in financial systems,” *Available at SSRN 2795583*.

⁵³What probability distribution a regulator wants to put across equilibria is outside the scope of our model, but one might wish to work with worst case as a benchmark.

⁵⁴See for instance, Duffie and Zhu (2011).

- BERNARD, B., A. CAPPONI, AND J. E. STIGLITZ (2017): “Bail-ins and bail-outs: Incentives, connectivity, and systemic stability,” Tech. rep., National Bureau of Economic Research.
- BORIA, N., F. DELLA CROCE, AND V. T. PASCHOS (2015): “On the max min vertex cover problem,” *Discrete Applied Mathematics*, 196, 62–71.
- BRANCH, B. (2002): “The costs of bankruptcy: A review,” *International Review of Financial Analysis*, 11:1, 39–57.
- BRIOSCHI, F., L. BUZZACCHI, AND M. M. COLOMBO (1989): “Risk Capital Financing and the Separation of Ownership and Control in Business Groups,” *Journal of Banking and Finance*, 13, 747–772.
- CAPPONI, A., F. C. CORELL, AND J. E. STIGLITZ (2020): “Optimal bailouts and the doom loop with a financial network,” Tech. rep., National Bureau of Economic Research.
- CHOI, D. B. (2014): “Heterogeneity and stability: Bolster the strong, not the weak,” *The Review of Financial Studies*, 27, 1830–1867.
- CSÓKA, P. AND P. J.-J. HERINGS (2018): “Decentralized clearing in financial networks,” *Management Science*, 64(10), 4681–99.
- DAVYDENKO, S. A., I. A. STREBULAEV, AND X. ZHAO (2012): “A market-based study of the cost of default,” *The Review of Financial Studies*, 25, 2959–2999.
- DEMANGE, G. (2016): “Contagion in financial networks: a threat index,” *Management Science*, 64, 955–70.
- D’ERRICO, M. AND T. ROUKNY (2019): “Compressing over-the-counter markets,” *arXiv preprint arXiv:1705.07155*.
- DIAMOND, D. W. AND P. H. DYBVIK (1983): “Bank runs, deposit insurance, and liquidity,” *Journal of political economy*, 91, 401–419.
- DONALDSON, J. R. AND G. PIACENTINO (2017): “Netting,” Tech. rep., Working paper, Washington University in St. Louis.
- DUARTE, F. AND C. JONES (2017): “Empirical network contagion for US financial institutions,” *Federal Reserve Bank of New York Staff Reports*.
- DUFFIE, D. (2020): “Still the World’s Safe Haven?” *Redesigning the US Treasury market after the COVID-19 crisis?*. *Hutchins Center on Fiscal and Monetary Policy at Brookings*. Available online: <https://www.brookings.edu/research/still-the-worlds-safe-haven> (accessed on 28 November 2021).
- DUFFIE, D. AND H. ZHU (2011): “Does a central clearing counterparty reduce counterparty risk?” *The Review of Asset Pricing Studies*, 1, 74–95.

- ECHENIQUE, F. (2007): “Finding all equilibria in games of strategic complements,” *Journal of Economic Theory*, 135, 514–532.
- EGRESSY, B. AND R. WATTENHOFFER (2021): “Bailouts in financial networks,” *arXiv preprint arXiv:2106.12315*.
- EISENBERG, L. AND T. H. NOE (2001): “Systemic Risk in Financial Systems,” *Management Science*, 47, 236–249.
- ELLIOTT, M., B. GOLUB, AND M. O. JACKSON (2014): “Financial Networks and Contagion,” *American Economic Review*, 104(10), 3115–3153.
- EROL, S. (2019): “Network hazard and bailouts,” *Available at SSRN 3034406*.
- EROL, S. AND R. VOHRA (2018): “Network formation and systemic risk,” *mimeo: UPenn*.
- FEDENIA, M., J. E. HODDER, AND A. J. TRIANTIS (1994): “Cross-Holdings: Estimation Issues, Biases, and Distortions,” *The Review of Financial Studies*, 7, 61–96.
- FLEMING, M. J. AND F. M. KEANE (2021): “The netting efficiencies of marketwide central clearing,” *FRB of New York Staff Report*.
- GAI, P. AND S. KAPADIA (2010): “Contagion in financial networks,” *Proceedings of the Royal Society A*, 466, 2401–2423.
- JACKSON, M. O. AND A. PERNOUD (2021): “Systemic Risk in Financial Networks: A Survey,” *Annual Review of Economics*, 13, 171–202.
- (2021b): “Investment Incentives and Regulation in Financial Networks,” *SSRN*: <https://dx.doi.org/10.2139/ssrn.3311839>.
- KAHN, C. M. AND W. ROBERDS (1998): “Payment system settlement and bank incentives,” *The review of financial studies*, 11, 845–870.
- KANIK, Z. (2019): “From Lombard Street to Wall Street: Systemic Risk, Rescues, and Stability in Financial Networks,” *mimeo: MIT*.
- KLAGES-MUNDT, A. AND A. MINCA (2022): “Optimal intervention in economic networks using influence maximization methods,” *European Journal of Operational Research*, 300, 1136–1148.
- LEITNER, Y. (2005): “Financial Networks: Contagion, Commitment, and Private Sector Bailouts,” *Journal of Finance*, 60(6), 2925–2953.
- LUCAS, D. (2019): “Measuring the cost of bailouts,” *Annual Review of Financial Economics*, 11, 85–108.
- MARTIN, A. AND J. MCANDREWS (2008): “An economic analysis of liquidity-saving mechanisms,” *Federal Reserve Bank of New York Research Paper Series-Economic Policy Review*.

- ON BANKING SUPERVISION, B. C. (1995): “The supervision of financial conglomerates,” *Committee Report*.
- PAPADIMITRIOU, C. H. (1994): *Computational Complexity*, Reading, Massachusetts: Addison Welsey.
- ROGERS, L. C. AND L. A. VERAART (2013): “Failure and rescue in an interbank network,” *Management Science*, 59, 882–898.
- ROUKNY, T., S. BATTISTON, AND J. E. STIGLITZ (2018): “Interconnectedness as a source of uncertainty in systemic risk,” *Journal of Financial Stability*, 35, 93–106.
- SCHULDENZUCKER, S. AND S. SEUKEN (2019): “Portfolio Compression in Financial Networks: Incentives and Systemic Risk,” *Available at SSRN*.
- STORY, L. AND E. DASH (2009): “Bankers Reaped Lavish Bonuses During Bailouts,” *The New York Times*, July, 30, 2009.

Appendix A Proofs

Before beginning the proofs, we note that using (4) we can rewrite the expressions for the debt value (1) in a single equation:

$$d_{ij}(\mathbf{V}_j) = \frac{D_{ij}}{D_j^L} \min(D_j^L, \max\{V_j + D_j^L, 0\}), \quad (6)$$

which accounts for both possible regimes: either j is solvent and creditors share the promised payment D_j^L , or it is not and they share the book value of j ’s assets net of failure costs $V_j + D_j^L$, if positive. (6) makes it more transparent that i ’s debt assets can be written as a function of only \mathbf{V}_{-i} , so that we can write $d_i^A(\mathbf{V}_{-i})$ and then rewrite (4) as

$$V_i = p_i + d_i^A(\mathbf{V}_{-i}) - D_i^L - b_i(\mathbf{V}, \mathbf{p}). \quad (7)$$

Proof of Proposition 1: (i) Let \bar{V} and \underline{V} be the best and worst equilibrium values of banks, respectively. Since there is no dependency cycle, it has to be that a subset of banks, denoted X_0 , only derive value from their outside investments: $D_i^A = 0$ for all $i \in X_0$. Thus, by the assumption on failure costs, $\beta_i(V_i, \mathbf{p})$ only depends on V_i for $i \in X_0$. Therefore, the values of these banks are determined solely by their investments, can be written as:

$$V_i = p_i - D_i^L - \beta_i(V_i, \mathbf{p}) \mathbb{1}_{\{p_i < D_i^L\}} \quad \forall i \in X_0.$$

Since $\beta_i(\cdot, \mathbf{p})$ is a contraction, there exists a unique solution to this equation by the Contraction Mapping Theorem. Therefore, $\bar{V}_i = \underline{V}_i$ for all $i \in X_0$. Next, let X_1 be the set of banks whose debt assets only involve banks in X_0 . Thus, their value is independent of that

of banks outside X_0 , and so by (6) and (7) we can write

$$V_i = p_i + \sum_{j \in X_0} d_{ij}(V_j) - D_i^L - \beta_i((V_j)_{j \in X_0 \cup \{i\}}, \mathbf{p}) \mathbb{1}_{\{p_i + \sum_{j \in X_0} d_{ij}(V_j) < D_i^L\}} \quad \forall i \in X_1.$$

Since $\bar{V}_i = \underline{V}_i$ for all $i \in X_0$ and $\beta_i(\cdot, \mathbf{p})$ is a contraction as a function of V_i , it follows that the best and worst equilibrium values of banks in X_1 are also the same: $\bar{V}_i = \underline{V}_i$ for all $i \in X_1$. Iteratively defining X_k to be the set of banks that have debt claims on banks in $\cup_{j < k} X_j$ only, the same argument applies by induction. For some integer $K \leq n$, $\cup_{k=0}^K X_k = N$, and thus it follows that $\bar{V}_i = \underline{V}_i$ for all banks.

(ii) Next, we prove that if there is a dependency cycle in the network, then there exists \mathbf{p} and failure costs such that $\bar{V} \neq \underline{V}$. In particular, we first show that under failure costs from (3) with full failure costs (that is, when a bank loses all of its assets upon default, $a = 1$, $b = 0$), then there exists returns \mathbf{p} such that this is true. We then show that this also holds for a below 1 (and above some threshold).

Let c be the set of all banks belonging to a dependency cycle. All other banks either (i) owe debts that have value (directly or indirectly) flowing into a dependency cycle; (ii) get value from debts coming out of a dependency cycle; or (iii) none of the above. Equilibrium values of banks in c is independent from that of banks in categories (ii) and (iii), because there are no directed paths of debts from such banks that are eventually owed to any bank in c . The values of banks in (i) can impact those of banks in c , but we set their portfolio values (p_i s) to zero and thus banks in c get no value from them.⁵⁵ Thus, equilibrium values of banks in c depend only on portfolios and values of banks in c , and each one of them has a debt liability to at least one other and a debt asset from at least one other.

Therefore, we redefine $D_i^A \equiv \sum_{j \in c} D_{ij}$ for all $i \in c$. We set portfolio values $(p_i)_{i \in c}$ to be as low as possible, while ensuring all banks in c remain solvent in the best equilibrium. In particular, when all banks in c are solvent their equilibrium values are

$$V_i = p_i + D_i^A - D_i^L,$$

and therefore the smallest $(p_i)_{i \in c}$ that ensure they all remain solvent in the best equilibrium are given by

$$p_i = [D_i^L - D_i^A]^+.$$

Thus, by using these portfolio values all banks in c are solvent in the best equilibrium: $\bar{V}_i \geq 0$ for all $i \in c$.

Next, consider what happens in the worst equilibrium with these portfolio values, under full failure costs – that is under (3) with $a = 1$, $b = 0$. Recall that all banks in c must have some debt liability and some debt claim – i.e., $D_i^A > 0$, and $D_i^L > 0$ for all $i \in c$. Let us suppose that they all default so that $d_i^A(\mathbf{V}_{-i}) = 0$ for all i . Then given that $p_i = [D_i^L - D_i^A]^+$

$$V_i = p_i - D_i^L = [D_i^L - D_i^A]^+ - D_i^L = -\min\{D_i^A, D_i^L\} < 0 \quad \forall i \in c,$$

⁵⁵Note that none of these banks are part of a cycle, and so by the argument above their values are uniquely tied down, and since they have no debts coming from any banks in (ii) or (iii), it is direct that they have no assets with which to pay any of their debts.

and assuming all banks in c default is self-fulfilling. Then given these failure costs $b_i(\mathbf{V}, \mathbf{p}) = p_i$, and so $d_i^A(\mathbf{V}_{-i}) = 0$ is self-fulfilling, and the best and worst equilibrium values differ.

To complete the proof, we note that the same is true for some $a < 1$, with a sufficiently close to 1. Presuming all banks default:

$$V_i = (p_i + d_i^A(\mathbf{V}_{-i}))(1 - a) - D_i^L,$$

with $d_i^A(\mathbf{V}_{-i})$ continuous in \mathbf{V}_{-i} , as any discontinuity arises only when new defaults occur. This has a unique fixed point when $a = 1$ of $V_i = -D_i^L$, and given that the righthand-side above is continuous in \mathbf{V}_{-i} and a , then the fixed points converge to the limit fixed point as $a \rightarrow 1$. So under the presumption that all banks default there is a solution for a close enough to 1 with all $V_i < 0$, justifying that all banks default in the worst equilibrium and have values below 0, differing from the best equilibrium.

(iii) First, we know that the set of equilibria forms a complete lattice, and so banks that default in the best equilibrium must be defaulting in all other equilibria. Second, the set of banks that default in a non-best equilibrium but not in the best equilibrium must contain all banks from at least one cycle. By contradiction, suppose not; i.e., there is no cycle of banks among these additional defaults. Then some of these banks must have claims only on banks whose solvency status is the same in the non-best equilibrium and in the best equilibrium, that is claims on banks who are either solvent in both equilibria, or insolvent in both equilibria. Hence the value of these banks' assets is the same in these two equilibria, and they cannot be defaulting in one but not in the other.

We are left to prove that any additional defaults must lie on outpointing paths from the original or the newly defaulting banks. Note that a bank that does not lie on such outgoing path must be getting the same value from its debtors in both equilibria. Hence it cannot be defaulting in the non-best equilibrium while remaining solvent in the best equilibrium. ■

Proof of Proposition 2: First, note that weak balance is a necessary condition for a bank to be solvent in any equilibrium. That is, $p_i + D_i^A < D_i^L$ for some i , then it must default in every equilibrium, since it then defaults even if it gets all its incoming debts paid, and thus regardless of the solvencies of other banks.

Next, note that if $p_i + D_i^A - D_i^L \geq 0$ for all i – i.e., the network is weakly balanced – then it is an equilibrium for all banks to be solvent, and so the best equilibrium has all banks solvent. Thus, weak balance is necessary and sufficient for solvency of all banks in the best equilibrium.

The characterization of solvency in the worst equilibrium requires more work. First, weak balance of the network is also a necessary condition for full solvency in the worst equilibrium, as argued above. Weak balance is, however, no longer sufficient. Thus, for the following, we suppose that the network is weakly balanced, and show that all banks are solvent in the worst equilibrium if and only if there is an iteratively strongly solvent set intersecting each directed cycle.

We first show that having an iteratively strongly solvent set intersecting each directed cycle (in addition to weak balance) implies all banks are solvent.

Suppose there is an iteratively strongly solvent set that intersects each directed cycle, and call it B . By the definition of an iteratively strongly solvent set, all banks in B must

be solvent in the worst equilibrium, which means that there is at least one solvent bank on each cycle. We prove that this implies all banks in the network are solvent by induction on the number of cycles in the network K .

If there are no cycles, then the best and worst equilibrium coincide (see the proof of Proposition 1), and then by the arguments above weak balance alone guarantees solvency of all banks (in fact, then it is easy to check that $B = N$ along the iterative arguments discussed above). If there is exactly one cycle, then consider some bank i on that cycle that is in B . This bank i is solvent in the worst, and thus in every, equilibrium given that all of B is solvent. So, consider the following modified network. For each j reset p_j to be $p_j + D_{ji}$, so presume that all debts from i are paid. Then reset D_{ji} to be 0. Note that we have not changed the structure of any equilibrium since i would have been solvent in every equilibrium, and also that weak balance still holds. The new network has no cycles, and thus all banks are solvent in all equilibria. More generally, consider a network with n cycles and some bank $i \in B$ that lies on some cycle. Via the same argument, we end up with a modified network that has the same equilibria and fewer cycles, and so by induction all banks are solvent in all equilibria.

Finally, we argue that if there does not exist an iteratively strongly solvent set that intersects every cycle, then some banks default in the worst equilibrium. First note that the union of iteratively strongly solvent sets is also an iteratively strongly solvent set: hence there exists a maximum one which, by assumption, does not intersect every cycle and hence does not include all banks. One can then check that the maximum iteratively strongly solvent set is actually the set of solvent banks in the worst equilibrium: presuming those banks are all solvent and none of the others are solvent is self-fulfilling, and so $N \setminus B$ are not solvent in the worst equilibrium. ■

Proof of Proposition 3: We show that finding the size of the largest minimal vertex cover, which is known to be a strongly NP-hard problem,⁵⁶ can be reduced to the decision problem associated with (OPT) for some specific network structures.

The largest minimal vertex cover problem is stated as follows: Given a graph, a vertex cover is a set of vertices that includes at least one endpoint of all edges in the graph. A vertex cover is minimal if it is not a superset of another vertex cover. The problem is to find the maximum size of a minimal vertex cover.

We show that if we can solve the decision version of the bailout problem for some specific network structures, then we can solve any instance of the decision version of the largest minimal vertex cover problem.

For any undirected graph, build an interbank network as follows: all vertices are banks, and all edges represent bilateral claims of size 1; i.e., if ij is an edge then $D_{ij} = D_{ji} = 1$. Hence any edge $ij \in E$ in the network generates a cycle of claims between Banks i and j . Finally suppose $p_i = 0.5$ for all i .

Note that the network is critically balanced: each bank's total liabilities equal its total debt assets $D_i^A = D_i^L$, and their outside assets are not enough to compensate one missing payment since $p_i < 1$. Hence all banks default in the worst equilibrium.

⁵⁶This is slightly different from the usual minimum (rather than largest minimal) cover problem, but is still strongly NP-hard. See [Boria et al. \(2015\)](#).

We first argue that an optimal bailout policy has to bailout all banks in a minimal vertex cover. One bank on each cycle must be bailed out to ensure systemic solvency, so the set of bailed out banks must form a vertex cover. If it does not, then it means one edge; i.e., one cycle, is not cleared by the policy as none of the banks involved in the cycle is bailed out. For the bailout policy to be optimal, the vertex cover must be minimal. If it isn't, then there exists a bank that is bailed out and whose counterparties are all bailed out as well. Bailing out this bank is not necessary since it is made solvent by receiving payments from all of its counterparties.

We next show that bailing out a minimal vertex cover of cardinality m costs $|E| - 0.5m$, where $|E|$ is the number of edges, and hence that a bailout policy must bailout a minimal vertex cover of maximum size to be optimal. The bailout policy cannot cost less than $|E| - 0.5m$, since all cycles must be cleared, which requires injecting at least 1 into all $|E|$ cycles, and that the total value of banks' assets that the regulator leverages is $0.5m$. Furthermore, the bailout policy cannot cost more than $|E| - 0.5m$. If it did, the regulator would be injecting more than 1 in some cycle, which cannot be optimal as injecting 1 in all cycles ensures systemic solvency. Let W be the minimum total bailout costs. Hence there exists a minimal vertex cover of size m if and only if $W \leq |E| - 0.5m$: if we can solve the bailout problem, then we can solve any instance of the largest minimal vertex cover problem. ■

Proof of Proposition 4: We first show that an optimal bailout policy must be bailing out at most one bank per cycle. Although we know from Proposition 2 that bailing out one bank per cycle is enough to guarantee systemic solvency, this does not directly imply that bailing out more than one bank per cycle cannot be cost-efficient as it could reduce the cost of a subsequent bailout.

Take any bailout policy $(i_\ell)_\ell$ that ensures systemic solvency, and let \mathcal{K}_ℓ be the set of cycles on which i_ℓ lies. A bailout policy bails out more than one bank per cycle if there exists ℓ such that $\mathcal{K}_\ell \setminus (\cup_{j=1}^{\ell-1} \mathcal{K}_j) = \emptyset$. That is, at some step, the policy bails out a bank that only lies on cycles on which another bank has already been bailed out. We show that for any such policy, there exists another that also ensures systemic solvency, that bails out at most one bank per cycle, and that is weakly cheaper. There are two cases: either i_ℓ 's bailout cost $C_{i_\ell|(i_1, \dots, i_{\ell-1})}$ is zero, or it is not. In the first case, then the "bailout" of i_ℓ is inconsequential as, given previous bailouts, the bank is already solvent. So $(i_{-\ell})$ is a cost-equivalent bailout policy that bails out at most one bank per cycle. The more interesting case is when i_ℓ 's bailout is actually costly to the regulator. This means that at least one of i_ℓ 's debtor is defaulting. Let X_1 be the set of all i_ℓ 's debtors that are defaulting despite already made bailouts $(i_1, \dots, i_{\ell-1})$. It has to be that each bank in X_1 also has a debtor that is defaulting. Iteratively construct X_k has the set of banks that owe money to banks in X_{k-1} and that are defaulting, and let $X = \cup_k X_k$. Three things are worth noting: (i) by construction, all banks in X are insolvent, (ii) the subnetwork $(X, (D_{ij})_{i,j \in X})$ must have at least one cycle, and (iii) bank i_ℓ is not in X . Indeed if $i_\ell \in X$, it must belong to a cycle of defaulting banks, and hence must belong to a cycle that has not been cleared out yet, which contradicts our initial assumption. Since the bailout policy ensures systemic solvency, it has to bailout some banks in X after i_ℓ 's bailout. Construct an alternative bailout policy by moving all bailouts of banks in X before i_ℓ 's bailout. That cannot increase the cost of these bailouts as banks

in X do not expect additional capital from i_ℓ since $i_\ell \notin X$. That makes i_ℓ 's bailout free – i.e. unnecessary, – and hence strictly reduces the cost of the overall bailout policy. It is thus without loss to only consider bailout policies that bailout at most one bank per cycle.

We now argue that computing the cost of all such policies requires no more than $K!n^{K+1}$ steps. A bailout policy that bails out one bank per cycle is characterized by (i) the order in which the cycles are cleared, and (ii) the identity of the bank on each cycle that is bailed out. There are $K!$ different orders in which cycles can be cleared. If m_1, \dots, m_K denote the number of banks on each cycle, there are $K! \prod m_k$ bailout policies to consider. The worst case is when cycles are as large as possible. Since $m_k \leq n$ for all k , this means there are at most $K!n^K$ different bailout policies to check. The least costly policy among those must be an optimal policy. Since computing the cost of each policy takes at most n steps, this brute-force algorithm finds the optimal bailout policy in no more than $K!n^{K+1}$ steps.

Proof of Proposition 5: We prove – by induction on the number of banks n – that the total injection of capital needed to ensure systemic solvency is no greater than $0.5 \sum_i (D_i^L - p_i)^+$. We provide the proof for the worst equilibrium, which then ensures that this works for any equilibrium. Also, we provide the proof for the first-step indirect value over cost ratio, and then discuss how it extends to the k th-step indirect value over cost ratio, for $k \geq 2$.

Given the set of banks S that are already solvent, let $x_{ij} \equiv \min\{D_i^L - p_i - \sum_{k \in S} D_{ik}, D_{ij}\}$ be the liquidity flow that j 's solvency induces in bank i . Let $x_j \equiv \sum_i x_{ij}$ be the overall indirect bailout value of bank j , that is the overall liquidity flow of interest that j 's solvency induces in the network. Finally, let $c_i \equiv (D_i^L - p_i - \sum_{j \in S} D_{ij})^+$ be the cost of bailing out bank i . Consider the algorithm that bails out the defaulting bank with highest x_i/c_i , recomputing these values after each step to account for all new solvencies.

It is simple to show that the claim holds for $n = 2$, as there is then a single network configuration to consider in which each bank has a claim on the other.⁵⁷ Bailing out any of the two banks is enough to clear the cycle, and so $x_1 = D_2^L - p_2$ and $x_2 = D_1^L - p_1$. Then the algorithm bails out the bank closest to solvency, which costs $\min\{(D_1^L - p_1)^+, (D_2^L - p_2)^+\} \leq 0.5[(D_1^L - p_1)^+ + (D_2^L - p_2)^+]$.

Now suppose the induction hypothesis holds for a network of n banks, and consider a network with $n + 1$ banks. Let $i^* = \arg \min_i x_i/c_i$ be the first bank picked by the algorithm, such that i^* is the bank with highest indirect bailout to bailout cost ratio. We give a bound on total bailout costs if the regulator starts by bailing out i^* .

The regulator first bails out i^* at a cost of $D_{i^*}^L - p_{i^*}^* = c_{i^*}$. We can then restrict attention to the remaining network of n banks, but we need to account for the payments they got from i^* . In aggregate, the remaining banks get a payment of $\sum_j \min\{D_j^L - p_j, D_{ji^*}\} = x_{i^*}$ from i^* . In that remaining subnetwork of n banks, the induction hypothesis tells us that the optimal bailout policy costs at most $0.5 \sum_{i \neq i^*} (D_i^L - p_i)^+ - x_{i^*}$. In total, accounting for the cost of i^* 's bailout, ensuring solvency then costs at most

$$0.5 \sum_{i \neq i^*} (D_i^L - p_i)^+ - x_{i^*} + c_{i^*}.$$

⁵⁷If not, then there is no cycle in the network and all banks are already solvent as the network of interest is assumed to be weakly balanced.

We then have to argue that $x_{i^*} \geq c_{i^*}$. By contradiction, suppose that $x_{i^*}/c_{i^*} < 1$. By definition of i^* , this means $x_i/c_i < 1$ for all i . However, we know that, since the network is weakly balanced, each banks getting its payments back in full has to be solvent. So

$$\sum_i x_{ij} = \sum_i \min\{D_j^L - p_j, D_{ji}\} \geq \min\{D_j^L - p_j, D_j^A\} \geq c_j.$$

Hence $\sum_j \sum_i x_{ij} \geq \sum_j c_j$, or equivalently $\sum_j x_j \geq \sum_j c_j$. There must then be at least one bank with $x_j \geq c_j$, and so $x_{i^*} \geq c_{i^*}$. The overall cost is then at most

$$0.5 \sum_{i \neq i^*} (D_i^L - p_i)^+ \leq 0.5 \sum_i (D_i^L - p_i)^+,$$

and the claim is true.

Extension to any k th-step indirect bailout value. As mentioned in the body of the paper, indirect bailout values can be defined at various levels to accounts for cascades of indirect payments induced by a bank's solvency, not only in terms of its payments, but further levels of solvencies and subsequent payments that its payments induce. Let $I_i^0 \equiv \{i\}$ and, taking as given some set of banks S that are already solvent, i 's bailout induces the solvency of banks in

$$I_i^1 \equiv \{j \notin S \cup I_i^0 \mid D_j^L - p_j \leq \sum_{\ell \in S \cup I_i^0} D_{j\ell}\}.$$

These banks paying back their debts leads to the following additional "indirect" solvencies

$$I_i^2 \equiv \{j \notin S \cup I_i^0 \cup I_i^1 \mid D_j^L - p_j \leq \sum_{\ell \in S \cup I_i^0 \cup I_i^1} D_{j\ell}\}.$$

Define recursively the set of banks made indirectly solvent at step k as

$$I_i^k \equiv \{j \notin S \cup I_i^0 \cup \dots \cup I_i^{k-1} \mid D_j^L - p_j \leq \sum_{\ell \in S \cup I_i^0 \cup \dots \cup I_i^{k-1}} D_{j\ell}\}.$$

This process must terminate at some step $k \leq n - 1$, given the finite number of banks. A bank i 's K -th step indirect bailout value, for $2 \leq K \leq n - 1$, is then

$$\sum_{j \notin S \cup I_i^0} \min\{D_{ji}, D_j^L - p_j - \sum_{l \in S} D_{jl}\} + \sum_{k=1}^{K-1} \sum_{h \in I_i^k} \sum_{j \notin S \cup I_i^0 \cup \dots \cup I_i^k} \min\{D_{jh}, D_j^L - p_j - \sum_{m \in S \cup I_i^0 \cup \dots \cup I_i^{k-1}} D_{jm}\}.$$

The proof of Proposition 5 works similarly for any k th-step indirect bailout value. The case with $n = 2$ is unchanged as these notions of indirect bailout value are all equivalent with only two banks. Note that a bank's k th-step indirect bailout value is always at least as high as the $k - 1$ th-step indirect value, and so if it is impossible that $x_i/c_i < 1$ for all i under the first-step indirect value definition of x_i , then it is also impossible if we weakly increase the x_i 's. ■

Proof of Proposition 6: Given that banks lie on at most one cycle, cycles can be partitioned into tiers. First, there are cycles that have no directed path coming in from any other cycle – call these \mathcal{C}_0 . Next, there are cycles that only have directed paths coming in from \mathcal{C}_0 , call these \mathcal{C}_1 . Then there are cycles that only have directed paths coming in from \mathcal{C}_0 and \mathcal{C}_1 , and so forth. Remaining banks either lie on directed paths between cycles, or on dead-end paths that come out of some cycle(s). Given that at least one bank must be made solvent on each directed cycle, it follows that finding the cheapest bank on each cycle in \mathcal{C}_0 is necessary for solvency, and cannot be made cheaper by any further bailouts. Under the ordering, these cycles will all be listed first and none will affect the bailing out of any other within this tier. Iterating on this logic, the result is easily verified, noting that any bank i that is not on any cycle will be cleared once all banks that have directed paths leading to it are bailed out, which necessarily has to be done before any cycles that lie on directed paths that point out from bank i are bailed out. ■

Proof of Proposition 7: First note that since the network is weakly balanced, and the center bank lies on all cycles, all banks are solvent in the worst equilibrium if and only if the center bank is. Hence the optimal bailout policy is the one ensuring the solvency of the center bank at minimum cost.

The center bank needs an amount of liquidity $(n-1)D^{out} - p_n \leq (n-1)D^{in}$ to be brought back to solvency. Bailing out a peripheral bank i costs $D^{in} - p_i$ and leads to a liquidity flow of D^{in} into the center bank. Hence it is always weakly cheaper (strictly if $p_i > 0$) to do so instead of injecting D^{in} directly into the center bank.

The only time at which it may not be optimal to bail out a peripheral bank is then when the center bank is less than D^{in} -close to solvency. This is the case if and only if $\lfloor m^* \rfloor$ peripheral banks have already been bailed out. Then the regulator can either bail out one additional peripheral bank, which costs $D^{in} - p_{\lfloor m^* \rfloor + 1}$, or inject the center bank's marginal shortfall directly into it. ■

Proof of Proposition 8: Let $x \in \{0, 1, \dots, n_C\}$ be the number of solvent core banks. The following reasoning holds for any x . Without loss, suppose there are still some insolvent core banks.

Core banks are completely identical except for the value of their outside assets. Hence bailing out any core bank induces the same liquidity flow into the network, but the cost of such bailout can differ across core banks. It is then always optimal to bring core banks back to solvency in decreasing order of their p_i .

What is the cheapest way to bring a core bank i back to solvency? By the previous argument, it cannot be cost-efficient to bailout its core counterparties, as these are more expensive to bailout than i . So the optimal way must only involve bank i 's peripheral banks, or injecting capital into i directly. This is the same problem as studied in Section 5.2, and Proposition 7 applies. The only slight difference is that, for some parameter values, bailing out all of i 's peripheral banks may not be enough to ensure its solvency – i.e. $m_i^*(x) > n_P$. The same logic however still holds: it is still optimal to leverage the peripheral banks' outside assets, and the optimal policy is then to bailout all of i 's peripheral banks, and then inject whatever additional capital is needed directly into i . ■