Forgiving Debt in Financial Network Games

Panagiotis Kanellopoulos[1] , **Maria Kyropoulou**[1] and **Hao Zhou**[1]
[1]University of Essex, UK
{panagiotis.kanellopoulos, maria.kyropoulou, hz18065}@essex.ac.uk

Abstract

We consider financial networks, where nodes correspond to banks and directed labeled edges correspond to debt contracts between banks. Maximizing systemic liquidity, i.e., the total money flow, is a natural objective of any financial authority. In particular, the financial authority may offer bailout money to some bank(s) or forgive the debts of others in order to maximize liquidity, and we examine efficient ways to achieve this. We study the computational hardness of finding the optimal debt-removal and budget-constrained optimal bailout policy, respectively, and we investigate the approximation ratio provided by the greedy bailout policy compared to the optimal one.

We also study financial systems from a game-theoretic standpoint. We observe that the removal of some incoming debt might be in the best interest of a bank. Assuming that a bank's well-being (i.e., utility) is aligned with the incoming payments they receive from the network, we define and analyze a game among banks who want to maximize their utility by strategically giving up some incoming payments. In addition, we extend the previous game by considering bailout payments. After formally defining the above games, we prove results about the existence and quality of pure Nash equilibria, as well as the computational complexity of finding such equilibria.

1 Introduction

A financial system comprises a set of institutions, such as banks, that engage in financial transactions. The interconnections showing the liabilities (financial obligations or debts) among the banks can be represented by a network, where the nodes correspond to banks and the edges correspond to liability relations. Each bank has a fixed amount of external assets (not affected by the network) which are measured in the same currency as the liabilities. A bank's total assets comprise its external assets and its incoming payments, and may be used for (outgoing) payments to its lenders. If a bank's assets are not enough to cover its liabilities, that bank will be in default and the value of its assets will be decreased (e.g., by liquidation); the extent of this decrease is captured by *default costs* and essentially implies that the corresponding bank will have only a part of its total assets available for making payments.

On the liquidation day (a.k.a. clearing), each bank in the system has to pay its debts in accordance with three principles of bankruptcy law (see, e.g., [Eisenberg and Noe, 2001]): i) *absolute priority*, i.e., banks with sufficient assets pay their liabilities in full, ii) *limited liability*, i.e., banks with insufficient assets to pay their liabilities are in default and pay all of their assets to lenders, subject to default costs, and iii) *proportionality*, i.e., in case of default, payments to lenders are made in proportion to the respective liability. Payments that satisfy the above properties are called *clearing payments* and maximal clearing payments, i.e., ones that point-wise maximize all corresponding payments, are known to exist and can be efficiently computed [Rogers and Veraart, 2013].

The total liquidity of a financial system is measured by the sum of payments made at clearing, and is a natural metric for the well-being of the system [Lee, 2013]. Financial authorities, e.g., governments or other regulators, wish to keep the systemic liquidity as high as possible and they might interfere, if their involvement is necessary and would considerably benefit the system. For example, in the not so far past, the Greek government (among others) took loans in order to bailout banks that were in danger of defaulting, to avert collapse. In this work, we study the possibility of a financial regulating authority performing cash injections (i.e., bailouts) to selected bank(s) and/or forgiving debts selectively, with the aim of maximizing the total liquidity of the system. Similarly to cash injections, it is a fact that debt removal can have a positive effect on systemic liquidity. Indeed, the existence of default costs can lead to the counter-intuitive phenomenon whereby removing a debt/edge from the financial network might result in increased money flow, e.g., if the corresponding borrower avoids default costs because of the removal.

Even more surprising than the increase of liquidity by the removal of debts, is the fact that the removal of an edge from borrower b to lender l might result in l receiving more incoming payments, e.g., if b avoids default costs and there is an alternative path in the network where money can flow from b to l. This motivates the definition of an edge-removal game on financial networks, where banks act as strategic agents who wish to maximize their total assets and might intentionally give-up a part of their due incoming payments towards this goal. This strategic consideration is meaningful both in the context where a financial authority performs cash injections or not. We consider the existence, quality, and computation of equilibria that arise in such games.

1.1 Our contribution

We consider computational problems related to maximizing systemic liquidity, when a financial authority can modify the network by appropriately removing debt, or by injecting cash into selected agents. We also consider financial network games where agents can choose to forgive incoming debts.

We show how to compute the optimal cash injection policy in polynomial time when there are no default costs, by solving a linear program; the problem is NP-hard when non-trivial default costs apply. As our LP-based algorithm requires knowledge of the available budget and leads to non-monotone payments, we study the approximation ratio of a greedy cash injection policy. Regarding debt removal, we prove that finding the set of liabilities whose removal maximizes systemic liquidity is NP-hard, and so are relevant optimization problems.

Regarding edge-removal games, with or without bailout, we study the existence and the quality of Nash equilibria, while also addressing computational complexity questions. Apart from arguing about well-established notions, such as the Price of Anarchy and the Price of Stability, we introduce the notion of the Effect of Anarchy (Stability, respectively) as a new measure on the quality of equilibria in this setting.

1.2 Related work

Our model is based on the seminal work of Eisenberg and Noe [2001] who introduced a widely adopted model for financial networks, assuming debt-only contracts and proportional payments. This was later extended by Rogers and Veraart [2013] to allow for default costs. Additional features have been since introduced, see e.g., [Schuldenzucker *et al.*, 2020] and [Papp and Wattenhofer, 2020]. We follow the model of Eisenberg and Noe and consider proportional payments; we note that a recent series of papers introduced different payment schemes [Bertschinger *et al.*, 2020; Papp and Wattenhofer, 2020; Kanellopoulos *et al.*, 2021].

When the financial regulator has available funds to bailout each bank of the network, Jackson et al. [2020] characterize the minimum bailout budget needed to ensure systemic solvency and prove that computing it is an NP-hard problem. When the financial authority has limited bailout budget, Demange [2018] proposes the threat index as a means to determine which banks should receive cash during a default episode and suggests a greedy algorithm for this process. Egressy and Wattenhofer [2021] focus on how central banks should decide which insolvent banks to bailout and formulate corresponding optimization problems. Dong et al. [2021] introduce an efficient greedy-based clearing algorithm for an extension of the Eisenberg-Noe model, while also studying bailout policies when banks in default have no assets to distribute. We note that the problem of injecting cash (as subsidies) in financial networks has been studied (in a different context) in microfinance markets [Irfan and Ortiz, 2018].

Further work includes [Schuldenzucker and Seuken, 2020] that considers the incentives banks might have to approve the removal of a set of liabilities forming a directed cycle in the financial network, while [Schuldenzucker *et al.*, 2017] considers the complexity of finding clearing payments when Credit Default Swap (3-party) contracts are allowed. In a similar spirit, [Ioannidis *et al.*, 2021] studies the clearing problem from the point of view of irrationality and approximation strength, while [Papp and Wattenhofer, 2021] studies which banks are in default, and how much of their liabilities these defaulting banks can pay.

2 Preliminaries

A *financial network* $N = (V, E)$ consists of a set $V = \{v_1, \dots, v_n\}$ of n banks and a set E containing directed edge (v_i, v_j) among these banks, where each bank v_i initially has some non-negative *external assets* e_i corresponding to income received from entities outside the financial system. Banks have payment obligations, i.e., *liabilities*, among themselves. In particular, a *debt contract* creates a liability l_{ij} of bank v_i (the borrower) to bank v_j (the lender); we assume that $l_{ij} \geq 0$ and $l_{ii} = 0$. Note that $l_{ij} > 0$ and $l_{ji} > 0$ may both hold simultaneously. Also, let $L_i = \sum_j l_{ij}$ be the total liabilities of bank v_i. Banks with sufficient funds to pay their obligations in full are called *solvent* banks, while ones that cannot are *in default*. Then, the relative liability matrix $\Pi \in \mathbb{R}^{n \times n}$ is defined by

$$\pi_{ij} = \begin{cases} l_{ij}/L_i, & \text{if } L_i > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Let p_{ij} denote the actual payment[1] from v_i to v_j; we assume that $p_{ii} = 0$. These payments define a payment matrix $\mathbf{P} = (p_{ij})$ with $i, j \in [n]$, where by $[n]$ we denote the set of integers $\{1, \dots, n\}$. We denote by $p_i = \sum_{j \in [n]} p_{ij}$ the total outgoing payments of bank v_i. A bank in default may need to liquidate its external assets or make payments to entities outside the financial system (e.g., to pay wages). This is modeled using *default costs* defined by values $\alpha, \beta \in [0, 1]$. A bank in default can only use an α fraction of its external assets and a β fraction of its incoming payments (the case without default costs is captured by $\alpha = \beta = 1$). The absolute priority and limited liability regulatory principles, discussed in the introduction, imply that a solvent bank must repay all its obligations to all its lenders, while a bank in default must repay as much of its debt as possible, taking default costs also into account. Summarizing, it must hold that $p_{ij} \leq l_{ij}$ and, furthermore, $\mathbf{P} = \Phi(\mathbf{P})$, where

$$\Phi(\mathbf{x})_{ij} = \begin{cases} l_{ij}, & \text{if } L_i \leq e_i + \sum_{j=1}^{n} x_{ji} \\ (\alpha e_i + \beta \sum_{j=1}^{n} x_{ji}) \cdot \pi_{ij}, & \text{otherwise.} \end{cases}$$

Payments $\mathbf{P}$ that satisfy these constraints are called *clearing payments*. *Proportional payments* have been frequently studied in the financial literature (e.g., in [Demange, 2018; Eisenberg and Noe, 2001; Rogers and Veraart, 2013]). Given

[1] Note that the actual payment need not equal the liability, i.e., the payment obligation.

clearing payments $\mathbf{P}$, in order to satisfy proportionality, each p_{ij} must also satisfy $p_{ij} = l_{ij}$ when v_i is solvent, and $p_{ij} = \left(\alpha e_i + \beta \sum_{j\in[n]} p_{ji}\right) \pi_{ij}$, when v_i is in default.

Given clearing payments $\mathbf{P}$, the *total assets* $a_i(\mathbf{P})$ of bank v_i are defined as the sum of external assets plus incoming payments, i.e.,

$$a_i(\mathbf{P}) = e_i + \sum_{j\in[n]} p_{ji}.$$

Maximal clearing payments, i.e., ones that point-wise maximize all corresponding payments (and hence total assets), are known to exist [Eisenberg and Noe, 2001; Rogers and Veraart, 2013] and can be computed in polynomial time.

We measure the total *liquidity* of the system (also refered to as systemic liquidity) $\mathcal{F}(\mathbf{P})$ as the sum of payments traversing through the network, i.e.

$$\mathcal{F}(\mathbf{P}) = \sum_{i\in[n]} \sum_{j\in[n]} p_{ji}.$$

We assume that there exists a financial authority (a *regulator*) who aims to maximize the systemic liquidity. In particular, the regulator can decide to remove certain debts (edges) from the network or inject cash to some bank(s). In the latter case, we assume the regulator has a total *budget* M available in order to perform *cash injections* to individual banks. We sometimes refer to the total increased liquidity, $\Delta\mathcal{F}$, (as opposed to total liquidity) which measures the difference in the systemic liquidity before and after the cash injections. A *cash injection policy* is a sequence of pairs of banks and associated transfers $((i_1, t_1), (i_2, t_2), \dots (i_L, t_L)) \in (V \times \mathbb{R})^L$, such that the regulator gives capital t_1 to bank i_1, t_2 to bank i_2, etc. These actions naturally define two corresponding optimization problems on the total (increased) liquidity, i.e., *optimal cash injection* and *optimal debt removal*.

We will also find useful the notion of the *threat index*[2], μ_i, of bank v_i, which captures how many units of total increased liquidity will be realized if the financial authority injects one unit of cash into bank v_i's external assets [Demange, 2018]; a unit of cash represents a small enough amount of money so that the set of banks in default would not change after the cash injection. We remark that for the maximum total increased liquidity it holds $\Delta\mathcal{F} \le M \cdot \mu_{\max}$, where $\mu_{\max}$ is the maximum threat index. Naturally, the threat index of solvent banks is 0, while the threat index of banks in default will be at least 1. Formally, the threat index is defined as

$$\mu_i = \begin{cases} 1 + \sum_{j\in D} \pi_{ij}\mu_j, & \text{if } a_i(\mathbf{P}) < L_i \\ 0, & \text{otherwise,} \end{cases}$$

where $D = \{j | a_j(\mathbf{P}) < L_j\}$ is the set of banks in default.[3]

An example. Figure 1 provides an example of a financial network, inspired by an example in [Demange, 2018]. The clearing payments are as follows: $p_{21} = 4.4$, $p_{32} = 3.2$, and $p_{43} = p_{45} = 1$, implying that banks v_2, v_3 and v_4 are in default. We assume that there are no default costs, i.e., $\alpha = \beta = 1$. The threat indexes are computed as follows: $\mu_1 = \mu_5 = 0$, $\mu_2 = 1 + \mu_1$, $\mu_3 = 1 + \mu_2$, and $\mu_4 = 1 + \frac{1}{2}\mu_3 + \frac{1}{2}\mu_5$, implying that $\mu_3 = \mu_4 = 2$, $\mu_2 = 1$, while $\mu_1 = \mu_5 = 0$.

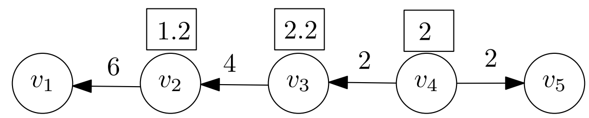

Figure 1: A simple financial network. Nodes correspond to agents, edges are labelled with the respective liabilities, while external assets are in a rectangle above the relevant agents.

3 Computing and approximating optimal outcomes

In this section we present algorithmic and complexity results regarding the problems of computing optimal cash injection (see Section 3.1) and debt removal (Section 3.2) policies. Note that we omit referring to default costs in our statements for those results that hold when $\alpha = \beta = 1$.

3.1 Optimal cash injections

We begin with a positive result about computing the optimal cash injection policy when default costs do not apply.

Theorem 1. *Computing the optimal cash injection policy can be solved in polynomial time.*

Proof. The proof follows by solving a linear program that computes the optimal cash injections and accompanying payments. In particular, we denote by x_i the cash injection to bank i and by p_{ij} the payment from i to j. We aim to maximize the total liquidity, i.e., the total payments, subject to satisfying the limited liability and absolute priority principles. Recall that M is the budget, l_{ij} is the liability of i to j, e_i is the external assets of bank i, and π_{ij} is the (i, j)-th entry of relative liability matrix Π.

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_i \sum_j p_{ij} & \\ \text{subject to} & \sum_i x_i \le M, & \forall i, j \\ & p_{ij} \le l_{ij}, & \forall i, j \\ & p_{ij} \le (x_i + e_i + \sum_k p_{ki}) \cdot \pi_{ij}, & \forall i, j \\ & x_i \ge 0, & \forall i \\ & p_{ij} \ge 0, & \forall i, j \end{array}$$

The first constraint corresponds to the budget constraint, while the second and third sets of constraints guarantee that no bank pays more than her total assets or more than a given liability; hence, the limited liability principle is satisfied. It remains to argue about the absolute priority principle, i.e., a bank can pay strictly less than her total assets only if she fully repays all outstanding liabilities.

[2] The term threat index aims to capture the “threat” posed to the network by a decrease in a bank's cash-flow or even the bank's default; this index can be thought of as counting all the defaulting creditors that would follow a potential default of the said bank.

[3] We note that threat indexes can be efficiently computed.

Consider the optimal solution corresponding to a vector of cash injections and payments p_{ij}; we will show that this solution satisfies the absolute priority principle as well. We distinguish between two cases depending on whether a bank is solvent or in default. In the first case, consider a solvent bank i, i.e., $x_i + e_i + \sum_j p_{ji} \geq L_i$, for which $p_{ik} < l_{ik}$ for some bank k. By replacing p_{ik} with $p'_{ik} = l_{ik}$, we obtain another feasible solution that strictly increases the objective function; a contradiction to the optimality of the starting solution. Similarly, consider a bank i with $x_i+e_i+\sum_j p_{ji} < L_i$ for which $\sum_j p_{ij} < x_i + e_i + \sum_j p_{ji}$. Then, there necessarily exists a bank k for which $p_{ik} < (x_i + e_i + \sum_j p_{ji}) \cdot \pi_{ik}$ and it suffices to replace p_{ik} with $p'_{ik} = (x_i + e_i + \sum_j p_{ji}) \cdot \pi_{ik}$ to obtain another feasible solution that, again, strictly increases the objective function. Hence, we have proven that the optimal solution to the linear program satisfies the absolute priority principle and the claim follows by providing each bank i a cash injection of x_i. □

Note that the optimal policy does not satisfy certain desirable properties. In particular, as observed in [Demange, 2018], cash injections are not monotone with respect to the budget. To see that, consider the financial network in Figure 1 and note that when $M = 0.5$, the optimal policy would give all available budget to bank v_3, while under an increased budget of 1.6, the entire budget would be allocated to v_4, hence v_3 would get nothing. Furthermore, our LP-based algorithm crucially relies on knowledge of the available budget.

In an attempt to alleviate these undesirable properties, we turn our attention to efficiently approximating the optimal cash injection policy by a natural and intuitive greedy algorithm, and we compute its approximation ratio under a limited budget, when we care about the total *increased* liquidity.

Definition 1 (GREEDY and its approximation ratio). *According to* GREEDY *(the Greedy cash injection policy), banks receive their cash injections in sequence, so that i_k, for $k = 1, \ldots, L$, is the bank with the highest threat index after the cash injection at round $k-1$ (round 0 is defined to be the starting configuration), while t_k is the minimum amount that would cause a change in the vector of threat indexes at the time it is transferred (it would lead to some previously defaulting bank to become solvent).*[4] *This process is repeated until the budget runs out.*

The approximation ratio *of* GREEDY *shows how smaller the total* increased *liquidity (or money flow) can be, compared to the optimal total increased liquidity, and is computed as*

$$\mathcal{R}_{Greedy} = \min_{N,M} \frac{\Delta\mathcal{F}_{Greedy}}{\Delta\mathcal{F}_{OPT}},$$

where the minimum is computed over all possible networks and budgets.

Let us revisit the example in Figure 1, assuming a budget $M = 1.6$. Initially banks v_3 and v_4 have the highest threat index of $\mu_3 = \mu_4 = 2$ compared to $\mu_1 = \mu_5 = 0$, and $\mu_2 = 1$. We can assume[5] that bank v_3 would receive the first cash injection ($i_1 = v_3$) and in fact this will be equal to $t_1 = 0.8$. Indeed, a cash injection of 0.8 to v_3 will result in v_3 becoming solvent (notice that v_3 receives 1 from v_4), while a smaller cash injection would not impose any change on the threat index vector. At this stage, the threat index of each bank is as follows $\mu'_1 = \mu'_3 = \mu'_5 = 0$ and $\mu'_2 = \mu'_4 = 1$. At this round, $i_2 = v_2$ would receive the remaining budget of $t_2 = 0.8$. Hence, the *total increased liquidity* achieved by GREEDY at this instance is $\Delta\mathcal{F}_{Greedy} = 2.4$ (t_1 will traverse edges (v_3, v_2) and (v_2, v_1), while t_2 will traverse edge (v_2, v_1)). However, the optimal cash injection policy is to inject the entire budget $M = 1.6$ to bank v_4 resulting in $\Delta\mathcal{F}_{OPT} = 3.2$. Therefore, this instance reveals $\mathcal{R}_{\text{GREEDY}} \leq \frac{2.4}{3.2} = 3/4$.

Theorem 2. GREEDY*'s approximation ratio is at most $3/4$. For inputs satisfying $M \leq t_1 \frac{\mu_v}{\mu_v - 1}$, this ratio is tight.*

We conclude this section with some hardness results.

Theorem 3. *The following problems are NP-hard: a) compute the optimal cash injection policy under the constraint of integer payments, b) compute the optimal cash injection policy with default costs $\alpha \in [0, 1)$ and $\beta \in [0, 1]$, and c) compute the minimum budget so that a given agent becomes solvent, with default costs $\alpha \in [0, 1/2)$ and $\beta \in [0, 1]$.*

Proof Sketch. We only present here a sketch of the proof of parts b and c; both follow by a reduction from the PARTITION problem, a well-known NP-complete problem. Recall that in PARTITION, an instance $\mathcal{I}$ consists of a set X of positive integers $\{x_1, x_2, \ldots, x_k\}$ and the question is whether there exists a subset X' of X such that $\sum_{i \in X'} x_i = \sum_{i \notin X'} x_i = \frac{1}{2} \sum_{i \in X} x_i$.

The reduction works as follows. Starting from $\mathcal{I}$, we build an instance $\mathcal{I}'$ by adding an agent v_i for each element $x_i \in X$ and allocating an external asset of $e_i = x_i$ to v_i; we also include three additional agents S, T and L. Each agent v_i has liability equal to $\frac{4e_i}{3}$ to S and equal to $\frac{2e_i}{3}$ to T, while S has liability $\frac{2+\alpha}{3} \sum_i e_i$ to L; see also Figure 2. We assume the presence of default costs $\alpha \in [0, 1)$, and $\beta \in [0, 1]$, while the budget is $M = \frac{1}{2} \sum_i e_i$; clearly, the reduction requires polynomial-time.

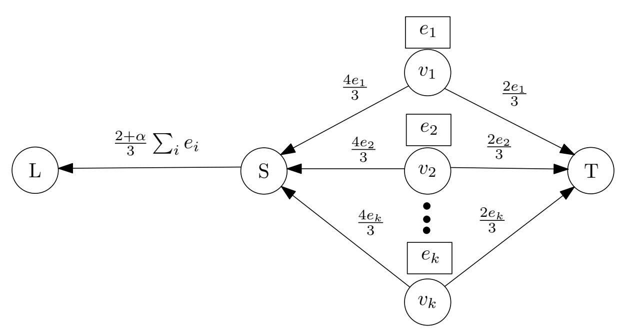

Figure 2: The reduction used to show hardness of computing the optimal cash injection policy when $\alpha < 1$.

[4] Without loss of generality we assume ties are broken in favor of the smallest index.

[5] This is consistent to our tie breaking assumption that favors the least index.

Part b follows by first showing that if $\mathcal{I}$ is a yes-instance for PARTITION, then the total liquidity is $\mathcal{F} = \frac{5\alpha+10}{6}\sum_i e_i$, as well as showing that any cash injection policy that leads to a total liquidity of at least $\frac{5\alpha+10}{6}\sum_i e_i$ leads to a solution for instance $\mathcal{I}$ of PARTITION.

Part c follows by arguing that any cash injection policy with a budget of $\frac{1}{2}\sum_i e_i$ that can make agent S solvent leads to a solution for instance $\mathcal{I}$ when the default costs are $\alpha \in [0, 1/2), \beta \in [0, 1]$, and vice versa. □

3.2 Optimal debt removal

In this section, we focus on maximizing systemic liquidity by appropriately removing edges/debts. As an example, consider again Figure 1, where the central authority can increase systemic liquidity by removing the edge between v_4 and v_5.

Theorem 4. *The problem of computing an edge set whose removal maximizes systemic liquidity is NP-hard.*

We note that the objective of systemic solvency, i.e., guaranteeing that all agents are solvent, can be trivially achieved by removing all edges. However, adding a liquidity target, makes this problem more challenging.

Theorem 5. *In networks with default costs, the following three problems are NP-hard. Compute an edge set whose removal: a) ensures systemic solvency and maximizes systemic liquidity, b) ensures systemic solvency and minimizes the amount of deleted liabilities, and c) guarantees that a given agent is no longer in default and minimizes the amount of deleted liabilities.*

4 The edge-removal game

In this section, we consider the case of strategic agents who have the option to forgive debt.

Edge-removal games and the Effect of Anarchy/Stability. Consider a financial network N of n banks who act strategically. The strategy set of a bank is the power set of its incoming edges and a strategy denotes which of its incoming edges that bank will remove, thus erasing the corresponding debt owed to itself. The edge-removal game can be defined with and without cash-injections. A given strategy vector will result in realized payments through maximal clearing payments including possible cash injections through a predetermined cash injection policy. Our results hold for both the optimal policy and GREEDY.

A bank is assumed to strategize over its incoming edges in order to maximize its utility, i.e., its total assets, where we remark that a possible cash injection can be seen as increasing one's external assets. The objective of the financial authority is to maximize the total liquidity of the system, i.e., the social welfare is the sum of money flows that traverse the network.

We consider the central notion of *Nash equilibrium* strategy profile, under which no bank can unilaterally increase its utility by changing strategy; let $\mathcal{S}_{eq}$ denote the set of strategy profiles at equilibrium. The inefficiency of Nash equilibria in terms of liquidity is measured by the *Price of Anarchy* (*Stability*, respectively) which is equal to optimal systemic liquidity, denoted by $\mathcal{F}_{OPT}$, over that of the worst (best, respectively) pure Nash equilibrium. Note that the optimal systemic liquidity corresponds to the maximal one when the financial authority can dictate everyone's actions (edge-removals).

$$\text{PoA} = \max_{N,M,\mathbf{s}\in\mathcal{S}_{eq}} \frac{\mathcal{F}_{OPT}}{\mathcal{F}_{\mathbf{s}}} \qquad \text{PoS} = \min_{N,M,\mathbf{s}\in\mathcal{S}_{eq}} \frac{\mathcal{F}_{OPT}}{\mathcal{F}_{s}}$$

The Price of Anarchy/Stability notions provide indications regarding the extent to which the individual objectives of the banks and the objective of the regulator are (not) aligned. We, here, introduce a new notion that we use to measure the discrepancy between the systemic liquidity of the original network (no edge removal), denoted by $\mathcal{F}_N$, and that of worst (best) Nash equilibrium. We call this the *Effect of Anarchy, (Stability, respectively)* and define it as follows.

$$\text{EoA} = \max_{N,M,\mathbf{s}\in\mathcal{S}_{eq}} \frac{\mathcal{F}_N}{\mathcal{F}_{\mathbf{s}}} \qquad \text{EoS} = \min_{N,M,\mathbf{s}\in\mathcal{S}_{eq}} \frac{\mathcal{F}_N}{\mathcal{F}_{\mathbf{s}}}$$

We investigate properties of Nash equilibria in the edge-removal game with respect to their existence and quality, while we also address computational complexity questions under different assumptions on whether default costs and/or cash injections apply. Our results on the Effect of Anarchy of edge removal games imply that, rather surprisingly, in the presence of default costs even the worst Nash equilibrium can be arbitrarily better than the original network in terms of liquidity. However, the situation is reversed in the absence of default costs, where we observe that the original network can be considerably better in terms of liquidity than the worst equilibrium; in line with similar Price of Anarchy results. We begin with some results for the basic case, that is, without default costs; recall that we do not refer to default costs in the statements for results holding for $\alpha = \beta = 1$.

Our first result exploits the fact that, for edge-removal games without cash injections, the strategy profile where all edges are preserved is a (not necessarily unique) Nash equilibrium.

Theorem 6. *Edge-removal games without cash injections always admit Nash equilibria.*

Theorem 7. *In edge-removal games without cash injections, the Effect of Anarchy is unbounded and the Effect of Stability is at most* 1.

Our next result shows that Nash equilibria may not exist once we allow for cash injections.

Theorem 8. *There is an edge-removal game with cash injections that does not admit Nash equilibria.*

Theorem 9. *The Price of Stability in edge-removal games (with or without cash injections) is unbounded.*

Theorem 10. *The Effect of Anarchy in edge-removal games with cash injections is at least* $n - 1$.

Theorem 11. *The Effect of Stability in edge-removal games with cash injections is* $\Omega(n)$.

Proof. Consider the network in Figure 3 where the budget $M = 1$, $k = n/2$ and H is arbitrarily larger than k. We start by noticing that $\mu_1 = 1$, while for $i = 2, \ldots, k$, it holds that $\mu_i = 1 + \frac{H-1}{H}\mu_{i-1} \approx 1 + \mu_{i-1}$, for sufficiently large H; all other banks are solvent. Hence, the optimal total liquidity

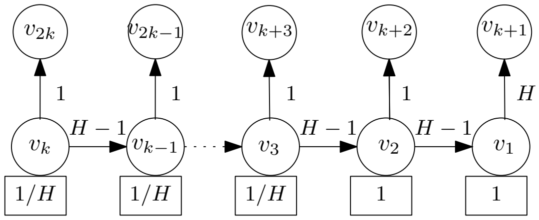

Figure 3: A network that yields EoS $= \Omega(n)$ for budget $M = 1$, $n = 2k$, and arbitrarily large H.

is achieved when v_k receives the entire budget of $M = 1$ as a cash injection, and is roughly $kM = n/2$, when H is sufficiently large.

We now claim that under any Nash equilibrium, v_2 will receive the budget and all edges (v_i, v_{i-1}) for $i \in \{3, \ldots, k\}$ are removed. This would complete the proof, as the total liquidity would be at most $\frac{1}{H}(k-2)+2+2 \leq 5$. We now prove this claim. Consider any equilibrium and observe that v_i, for $i = k+1, \ldots, 2k$, must have their unique incoming edge present. Now, assume for a contradiction that some bank v_i with $i \in \{3, \ldots, k\}$ gets a cash injection; this implies that the edge (v_i, v_{i-1}) is present as, otherwise, the result holds trivially. Then, bank v_{i-1} has total assets $1 - 1/H^2 + e_{i-1}$, but can increase them to $1 + e_{i-1}$ by strategically removing its incoming edge. So, under any Nash equilibrium, either v_2 or v_1 receives a cash injection. In the former case, where edge (v_2, v_1) is present, $a_1 = 3 - 2/H$, while the assets of v_1 would be 2 if it removed its incoming edge and received the cash injection.

So far, we have proven that v_2 gets the cash injection and it remains to show no other edge (v_i, v_{i-1}) for $i \in \{3, \ldots, k\}$ exists in a Nash equilibrium. Now, observe that if such an edge exists, then neighboring edges on the horizontal path cannot exist as that would contradict that v_2 gets the cash injection. Then, when $i > 4$, bank v_{i-2} would have an incentive to add edge (v_{i-1}, v_{i-2}), thus, making bank v_i the recipient of the budget (for both optimal and greedy) and strictly increase its own total assets. The cases $i \in \{3, 4\}$ can be easily ruled out as well. Our proof is complete. $\square$

We now present a series of results for the case where default costs exist, but cash injections are not allowed. Contrary to the case with neither default costs nor cash injections, we show that a Nash equilibrium is no longer guaranteed to exist; the next result is complementary to Theorem 8.

Theorem 12. *There is an edge-removal game with default costs but without cash injections that does not admit Nash equilibria.*

For some restricted topologies, however, the existence of Nash equilibria is guaranteed; in particular, keeping all edges is a Nash equilibrium.

Theorem 13. *Edge-removal games with default costs but without cash injections always admit Nash equilibria if the financial network is a tree or a cycle.*

The following result demonstrates that the positive impact of (individually benefiting) edge removals dominates the negative impact of reducing the number of edges through which money can flow, hence, edge removals are in line with the regulator's best interest too.

Lemma 1. *Edge-removal games with default costs but no cash injections satisfy the following: given any network and any strategy profile, any unilateral removal of any edge(s) that weakly improves the total assets of the corresponding bank, also weakly improves the total assets of every other bank in the network. Consequently, the total liquidity of the system is increased.*

In fact, the systemic liquidity of even the worst Nash equilibrium can be arbitrarily higher than at the original network. To see this, consider the network in the proof of Theorem 14, which admits a unique Nash equilibrium with arbitrarily higher total liquidity than that of the original network.

Theorem 14. *When default costs apply but there are no cash injections, the Effect of Stability is arbitrarily close to* 0.

Proof. Consider a network with n nodes, $v_1, \ldots, v_n$, and edges (v_i, v_{i+1}), for $i = 1, \ldots, n-1$, as well as edge (v_1, v_n), all with unit liability. Only bank v_1 has one unit of external asset, while we assume default costs $\alpha = \beta = \epsilon$ for some arbitrary small positive constant ϵ.

If no edge is removed, then all banks except v_n are in default and the following payments are realized: $p_{12} = p_{1n} = \epsilon/2$ and $p_{i,i+1} = \epsilon \cdot p_{i-1,i} = \epsilon^i/2$. The systemic liquidity is then $\mathcal{F}_N = \epsilon/2 + \sum_{i=1}^{n-1} \frac{\epsilon^i}{2} < \frac{\epsilon}{1-\epsilon}$. On the other hand, the unique Nash equilibrium is achieved when v_n removes the edge pointing from v_1 to itself. The systemic liquidity in this case is $n-1$, and the proof follows. $\square$

We conclude with our results on computational complexity for the setting with default costs.

Theorem 15. *In edge-removal games with default costs, the following problems are NP-hard: a) decide whether a Nash equilibrium exists or not, b) compute a Nash equilibrium, when it is guaranteed to exist, c) compute a best-response strategy, and d) compute a strategy profile that maximizes systemic liquidity.*

5 Conclusions

We considered problems arising in financial networks, when a financial authority wishes to maximize the total liquidity either by injecting cash or by removing debt. We also studied the setting where banks are rational strategic agents that might prefer to forgive some debt if this leads to greater utility, and we analyzed the corresponding games with respect to properties of Nash equilibria. In that context, we also introduced the notion of the Effect of Anarchy (Stability, respectively) that compares the liquidity in the initial network to that of the worst (best, respectively) Nash equilibrium.

Our work leaves some interesting problems unresolved. Given the computational hardness of some of the optimization problems, it makes sense to consider approximation algorithms. From the game-theoretic point of view, one can also consider the problems from a mechanism design angle, i.e., to design incentive-compatible policies where banks weakly prefer to keep all incoming liabilities.

References

[Bertschinger *et al.*, 2020] Nils Bertschinger, Martin Hoefer, and Daniel Schmand. Strategic payments in financial networks. In *Proceedings ofthe 11th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 46:1–46:16, 2020.

[Demange, 2018] Gabrielle Demange. Contagion in financial networks: a threat index. *Management Science*, 64(2):955–970, 2018.

[Dong *et al.*, 2021] Zhi-Long Dong, Jiming Peng, Fengmin Xu, and Yu-Hong Dai. On some extended mixed integer optimization models of the Eisenberg-Noe model in systemic risk management. *International Transactions in Operational Research*, 28(6):3014–3037, 2021.

[Egressy and Wattenhofer, 2021] Beni Egressy and Roger Wattenhofer. Bailouts in financial networks. *arXiv:2106.12315*, 2021.

[Eisenberg and Noe, 2001] Larry Eisenberg and Thomas H. Noe. Systemic risk in financial systems. *Management Science*, 47(2):236–249, 2001.

[Ioannidis *et al.*, 2021] Stavros D. Ioannidis, Bart de Keijzer, and Carmine Ventre. Strong approximations and irrationality in financial networks with financial derivatives. *arXiV*, abs/2109.06608, 2021.

[Irfan and Ortiz, 2018] Mohammad T. Irfan and Luis E. Ortiz. Causal strategic inference in a game-theoretic model of multiplayer networked microfinance markets. *ACM Transactions on Economics and Computation*, 6(2), 2018.

[Jackson and Pernoud, 2020] Matthew O. Jackson and Agathe Pernoud. Credit freezes, equilibrium multiplicity, and optimal bailouts in financial networks. *arXiv:2012.12861*, 2020.

[Kanellopoulos *et al.*, 2021] Panagiotis Kanellopoulos, Maria Kyropoulou, and Hao Zhou. Financial network games. In *Proceedings of the 2nd ACM International Conference on Finance in AI (ICAIF)*, 2021.

[Lee, 2013] Seung Hwan Lee. Systemic liquidity shortages and interbank network structures. *Journal of Financial Stability*, 9(1):1–12, 2013.

[Papp and Wattenhofer, 2020] Pál András Papp and Roger Wattenhofer. Network-aware strategies in financial systems. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 91:1–91:17, 2020.

[Papp and Wattenhofer, 2021] Pál András Papp and Roger Wattenhofer. Default ambiguity: finding the best solution to the clearing problem. In *Proceedings of the 17th Conference on Web and Internet Economics (WINE)*, 2021.

[Rogers and Veraart, 2013] Leonard CG Rogers and Luitgard Anna Maria Veraart. Failure and rescue in an interbank network. *Management Science*, 59(4):882–898, 2013.

[Schuldenzucker and Seuken, 2020] Steffen Schuldenzucker and Sven Seuken. Portfolio compression in financial networks: Incentives and systemic risk. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, page 79, 2020.

[Schuldenzucker *et al.*, 2017] Steffen Schuldenzucker, Sven Seuken, and Stefano Battiston. Finding clearing payments in financial networks with credit default swaps is PPAD-complete. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 32:1–32:20, 2017.

[Schuldenzucker *et al.*, 2020] Steffen Schuldenzucker, Sven Seuken, and Stefano Battiston. Default ambiguity: Credit default swaps create new systemic risks in financial networks. *Management Science*, 66(5):1981–1998, 2020.