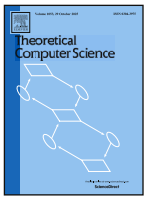
Theoretical Computer Science 1068 (2026) 115788



Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs





Dynamic debt swapping in financial networks

Henri Froese [a], Martin Hoefer [b], Lisa Wilhelmi [b,*]

[a] Dept. of Computer Science, Goethe University Frankfurt, Frankfurt am Main, 60325, Germany
[b] Dept. of Computer Science, RWTH Aachen University, Aachen, 52074, Germany

ARTICLE INFO

Section Editor: Dr. Vladimiro Sassone
Handling Editor: Katie Harris

Keywords:
Debt swap
Financial networks
Local search

ABSTRACT

A *debt swap* is an elementary edge swap in a directed, weighted graph, where two edges with the same weight swap their targets. Debt swaps are a natural and appealing operation in financial networks, in which nodes are banks and edges represent debt contracts. They can improve the clearing payments and the stability of these networks. However, their algorithmic properties are not well-understood.

We analyze the computational complexity of debt swapping. Our main interest lies in *semi-positive* swaps, in which no creditor strictly suffers and at least one strictly profits. These swaps lead to a Pareto-improvement in the entire network. We consider network optimization via sequences of v-improving debt swaps from which a given bank v strictly profits. For ranking-based clearing, we show that every sequence of semi-positive v-improving swaps has polynomial length. In contrast, for arbitrary v-improving swaps, the problem of reaching a network configuration that allows no further swaps is PLS-complete.

In global optimization, the goal is to maximize the utility of a given bank v by performing a sequence of debt swaps in the network. This problem is NP-hard to approximate for multiple types of swaps.

Moreover, we study reachability problems – deciding if a sequence of swaps exists between given initial and final networks. We design a polynomial-time algorithm to decide this question for arbitrary swaps and derive hardness results for several other types of swaps.

Many of our results can be extended to networks with arbitrary monotone clearing.

1. Introduction

The international financial system represents a complex and highly interconnected network. A prominent threat is that tight structural connections between banks, funds, and other financial institutions can yield devastating cascading effects. These systemic risks have been a major issue in the financial crisis of 2008 and several other crises since then. Mitigating these risks is an important problem, which has recently also received attention in computer science. The goal here is to understand the computational complexity of network-based interbank models for financial systems and potential interventions.

A fundamental aspect of modeling financial systems is *clearing*, a process of debt resolution by which solvency of banks in the system gets determined. In a financial network (where banks are nodes, and weighted, directed edges correspond to debt contracts) the clearing problem asks for a network flow that represents a consistent assignment of payments of debtor banks to their creditors. Based on the clearing, a bank might not receive the full amount of credit it gave to others. As a consequence, it might not have

* Corresponding author.
E-mail addresses: henri.froese@stud.uni-frankfurt.de (H. Froese), mhoefer@cs.rwth-aachen.de (M. Hoefer), wilhelmi@algo.rwth-aachen.de (L. Wilhelmi).

https://doi.org/10.1016/j.tcs.2026.115788
Received 17 April 2025; Received in revised form 21 January 2026; Accepted 26 January 2026
Available online 28 January 2026
0304-3975/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

sufficient funds to clear all her own debt. Naturally, the solution to the clearing problem is of central importance to banks since it directly affects their assets.

Banks have an obvious incentive to improve their clearing conditions. Moreover, governmental regulators, central banks, and other financial services share an interest in maintaining and improving the financial system. While allocation rules for clearing are often fixed, other interventions for improvement have the potential to significantly improve the clearing conditions.

A particularly interesting variant of network improvement is *debt swapping* [1]: There are two debt contracts among two pairs of banks (u_1, v_1) and (u_2, v_2). The contracts have the same value. In a debt swap, the creditor banks v_1 and v_2 change their role as recipients of these contracts. This leaves every bank with the same amounts of debt as before – only the network is affected by the exchange of the edges. A debt swap is a local edge-swap operation and represents a small change in the network structure. Since it involves only a small number of banks, it can be executed in a distributed fashion by the participating banks. Banks are more likely to agree to this operation if it does not deteriorate their network position, i.e., it does not decrease their assets in the clearing of the resulting network.

Debt swaps are a very natural and interesting swap operation with many conceptual advantages. For example, they are usually less demanding than the more prominent approach of portfolio compression [2,3] (for a discussion of the relation between these two operations, see [1]). Moreover, seemingly different operations proposed by regulators (such as, e.g., claims trades) can be interpreted as debt swaps [4]. From a computational perspective, debt swaps are related to a substantial body of literature on edge swap operations in graphs. Numerous variants of such swaps have attracted a large interest in computer science over the decades (see related work discussed below). The computational aspects of debt swapping, however, are not well understood.

In this paper, we strive to understand debt swap dynamics and their effects on the network. Since a debt swap is a local improvement step in the network, we first focus on local optimization (e.g., from an initial network, can we reach a configuration that allows no more profitable debt swaps?). In addition, we also consider aspects of global optimization (i.e., maximize the assets of a single bank via a sequence of debt swaps) and reachability problems (i.e., decide if a target network structure can be reached using a sequence of debt swaps). We discuss all our results consistently for networks in which banks use natural ranking-based clearing payments [5–7]. Indeed, many of our results can be extended to arbitrary monotone clearing. Throughout the paper we mention the extension explicitly whenever it applies.

Contribution and Techniques. After discussing preliminaries in Section 2, we classify debt swaps into positive (both creditors profit strictly in the clearing after the swap), semi-positive (one creditor profits strictly, the other one remains the same), or arbitrary ones. Our characterization results in Section 3 show that in every financial network, there can be no positive swap. Semi-positive swaps can exist, and, in fact, they always represent a *Pareto-improvement*: After the swap *no bank in the entire network suffers*, every bank obtains at least as many assets as before. As such, the *recovery rate* of every bank weakly increases, as well as the payments towards every debt contract. This makes semi-positive swaps very attractive. Similar results were shown by Papp and Wattenhofer [1] for proportional payments. Our paper generalizes these results substantially to *all monotone payments*.

We then concentrate on improvement dynamics with semi-positive debt swaps. Since they Pareto-improve the assets of all banks (and strictly for a creditor bank), every sequence of semi-positive swaps is inherently acyclic and terminates in a local optimum. A natural and challenging question that we study in Section 4 is whether there are short sequences that allow to reach a local optimum in polynomial time. Towards answering this question, we further refine the characterization of semi-positive swaps for ranking-based payments. Our characterization involves several classes of swaps. For each but one class, we show that there can be only a polynomial number of swaps from that class in every sequence. The existence of short sequences then depends crucially on one remaining class of so-called *active extension swaps* (for a formal statement see Definitions 4 and 5 below).

When a single bank v strictly profits from every semi-positive swap, then we prove that the number of active extension swaps is also bounded by a polynomial in the size of the network (and, thus, so is every sequence of semi-positive swaps). Instead, when we consider arbitrary swaps in which a given bank v strictly improves, then there are initial networks from which *every* sequence of such swaps has exponential length. More fundamentally, we prove that it is PLS-complete to construct any local optimum, i.e., any network where liabilities of each node are consistent with the initial network and that allows no (arbitrary) debt swap in which the assets of a given bank v strictly increase.

The problem without the requirement of strict improvement for a single bank v turns out to be extremely challenging. We also make some progress here – under a condition on the liabilities and the in-degree of partly paid edges of every bank, we can show existence of a short sequence of semi-positive swaps to a local optimum. The general problem remains as an interesting avenue for future work.

In Section 5, we consider global optimization, i.e., the problem of determining the maximal assets of a given bank that can be obtained by a sequence of debt swaps from a given initial network. We show that this problem is NP-hard to approximate, both when considering arbitrary or only semi-positive debt swaps. Similar hardness results apply when determining the maximal *sum of assets of several banks*.

In Section 6 we then turn to reachability problems: Given an initial and a target network on the same set of nodes V, the goal is to decide whether or not there is a sequence of debt swaps to turn the initial network into the target network. For arbitrary debt swaps, we show that the problem can be solved in polynomial time. In contrast, when we have to maintain a lower bound on the assets of a bank v throughout, the problem becomes PSPACE-complete. For semi-positive debt swaps, we show that the problem is NP-hard.

In Table 1 we summarize our main results with references to the respective theorems and corollaries in the technical part.

Table 1
Main results of this paper.

Local Optimization		
Swaps		v-improving
Semi-Positive	P in special cases (Theorem 4)	P (Theorem 3)
Arbitrary		PLS-complete (Theorem 5)

Global Optimization		
Swaps		v-improving
Semi-Positive	$n^{1/2-\epsilon}$-hard (Theorem 7, Corollary 8)	NP-hard (Theorem 8, Theorem 9)
Arbitrary	P (Theorem 10)	PSPACE-complete (Corollary 9/Theorem 11)

Reachability		
Swap		v-improving/with LB
Semi-Positive	NP-hard (Proposition 7)	
Arbitrary	APX-hard (Corollary 6)	APX-hard (Corollary 7)

Related Work. There has been a recent surge of interest in the algorithmic and game-theoretic aspects of financial networks. Most works adopt and extend the clearing model by Eisenberg and Noe [8] (discussed in more detail below), where banks are nodes, edges are debt contracts, and clearing is governed by proportional payments. Most closely related to our work is recent work by Papp and Wattenhofer [1], who introduce and study the concept of debt swaps for proportional payments. They show that positive swaps do not exist, semi-positive swaps lead to Pareto-improvement, discuss shock models and hardness of a number of optimization problems. Another network-altering operation was recently introduced by Kanellopoulos et al. [9]. A debt transfer describes the process of a bank transferring a debt claim to another creditor. Formally, this operation exchanges a path of length two by one single edge. The authors consider optimization problems and a game-theoretic model where the banks strategically decide whether or not to perform a debt transfer.

More generally, algorithmic aspects of (extensions of) the Eisenberg-Noe model have been subject to substantial recent interest, including, e.g., the computation of clearing states in networks with credit-default swaps (CDS) [7,10,11]. Some clearing models (e.g., using CDS'es) suffer from default ambiguity [12], and several strategies for resolving this problem have been proposed [13]. Changes of the instance in terms of adding or deleting single liabilities, donations to other banks, lending, manipulation of external assets, allocation of stimulus checks, or bailouts were analyzed in [14–17]. While most works assume the clearing state to manifest as a fixed point, the impact of *sequential* updates is studied in [18]. Further temporal aspects are investigated in [19], where liabilities are associated with individual maturity intervals. The works investigate the complexity of allocating integral and fractional payments in order to minimize defaults.

There is a growing interest in clearing problems beyond the restriction to proportional payments, using more general classes of monotone functions to model the clearing process [5]. This extension poses many interesting algorithmic and, in particular, game-theoretic problems that are very recently starting to get explored [6,20,21].

An extended abstract of this work has been published in the proceedings of the 4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2025) [22].

2. Model and preliminaries

Financial Networks. A *financial network* $\mathcal{F} = (G, \mathsf{de}, \mathsf{cr}, \mathbf{l}, \mathbf{a}^x, \mathbf{f})$ is given by a weighted, directed multi-graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ edges. Each node $v \in V$ resembles a financial institution or *bank*. Each edge $e \in E$ represents a debt contract, where a debtor bank $\mathsf{de}(e) \in V$ owes an amount of $l_e \in \mathbb{N}_{>0}$ to a creditor bank $\mathsf{cr}(e)$. Formally, e is a directed edge $(\mathsf{de}(e), \mathsf{cr}(e))$ with weight l_e. We assume that there are no loops, i.e., $\mathsf{de}(e) \neq \mathsf{cr}(e)$. The set of incoming and outgoing edges of v are denoted by $E^-(v) = \{e \in E \mid \mathsf{cr}(e) = v\}$ and $E^+(v) = \{e \in E \mid \mathsf{de}(e) = v\}$. The *total liabilities* of a bank v are defined as $L_v = \sum_{e \in E^+(v)} l_e$. Further, every bank holds *external assets* $a^x_v \in \mathbb{N}$, i.e., assets of v that are independent of the network. The payments that a bank receives from its debtors within the network are called *internal assets* and, together with external assets, form the available assets of a bank.

An *allocation rule* is a collection $\mathbf{f} = (\mathbf{f}_v)_{v \in V}$, where for bank v $\mathbf{f}_v = (f_e)_{e \in E^+(v)}$ determines a distribution of the available assets $b_v \in \mathbb{R}_{\geq 0}$ towards her liabilities. $f_e : \mathbb{R}_{\geq 0} \to [0, l_e]$ is the *payment function* for edge $e \in E^+(v)$. For every bank v, the payment functions satisfy the following properties: (1) f_e is continuous[1] and non-decreasing in b_v (monotonicity), (2) each bank pays at most the respective

[1] Monotonicity, edge capacity, and limited flow conservation together imply continuity. We postulate it explicitly for convenience.

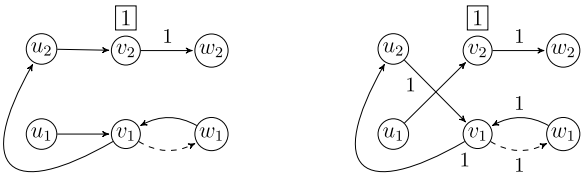


Fig. 1. The figures visualize a financial network $\mathcal{F}$ before (left) and $\mathcal{F}^\sigma$ after a debt swap (right). All edges have unit weight and edge labels indicate payments. For v_1, the solid edge is the preferred outgoing edge. v_2 has external assets of 1 as shown in the node label.

liability, i.e., $f_e(b_v) \le l_e$ (edge capacity), and (3) every bank spends all assets until all liabilities are paid off, i.e., $\sum_{e\in E^+(v)} f_e(b_v) = \min\{b_v, L_v\}$ (limited flow conservation). The available assets of v are defined as the sum of external and internal assets $b_v = a_v^x + \sum_{e\in E^-(v)} f_e(b_{\mathsf{de}(e)}) \ge 0$.

These constraints give rise to a fixed-point problem. A vector of assets $(b_v)_{v\in V}$ that fulfills all constraints is called *feasible*. For every allocation rule $\mathbf{f}$, the set of feasible vectors forms a complete lattice w.r.t. coordinate-wise comparison [5,6]. Consequently, for every allocation rule $\mathbf{f}$ there exists a supremum $(a_v)_{v\in V}$ of feasible assets and resulting payments $\mathbf{p} = (p_e)_{e\in E}$ with $p_e = f_e(a_{\mathsf{de}(e)})$ that pointwise maximize the assets of each bank and the payments on each edge. Following previous literature, we assume the clearing settles on these maximal payments, which we term the *clearing state*. Thus, the clearing state is uniquely determined for every instance. In the clearing state, bank v receives *total incoming assets* of $\sum_{e\in E^-(v)} p_e$ and holds *total assets* of $a_v = a_v^x + \sum_{e\in E^-(v)} p_e$.

In this paper, we will discuss all our results consistently in the context of a natural class of *edge-ranking* rules [5–7]: For each bank v, her outgoing edges $E^+(v)$ are ordered by a permutation π_v. First, v directs all her payments towards edge $\pi_v(1)$ until the edge is completely paid off or v has no remaining assets. Then, she pays off edge $\pi_v(2)$ until the edge is completely paid off or v has no remaining assets. This process continues until either all debt is settled or v runs out of funds.

Apart from being natural and well-motivated in applications, edge-ranking rules also have appealing computational properties. Given a financial network with edge-ranking rules, a maximal clearing state can be computed in strongly polynomial time [6]. Moreover, these favorable properties extend to general monotone integral [5,6] allocation rules in explicit representation by using edge-rankings over a suitable set of auxiliary edges [6]. In this way, *all our statements generalize* to unit-ranking rules with edge-ranking representation.

Other classes of payments have also been of interest in the literature (e.g., *proportional payments*, where every bank allocates her total assets in proportion to her total liabilities, or combinations of ranking and proportional). Indeed, many of our results extend even to arbitrary monotone allocation rules.

Debt Swaps. For a given financial network $\mathcal{F}$, a debt swap σ is a structural change to the network. For two edges e_1, e_2 among four distinct banks with the same liability (i.e., $l_{e_1} = l_{e_2} =: \ell$) we exchange their creditors. The result can be described by an adjustment in the creditor function by swapping the entries of e_1 and e_2. Note that the allocation rule $\mathbf{f}$ of the banks remains unaffected by the swap, since the debtor function remains unchanged. However, due to the structural change, a debt swap may change the payments in the clearing state. We denote the resulting financial network, creditor function, clearing state, and total assets of a bank v after the swap by $\mathcal{F}^\sigma$, cr^σ, $\mathbf{p}^\sigma$ and a_v^σ, respectively.

Definition 1 (Debt Swap)**.** Consider a financial network $\mathcal{F}$ with edges $e_1, e_2 \in E$. We use the short notation $v_1 = \mathsf{cr}(e_1)$, $v_2 = \mathsf{cr}(e_2)$, $u_1 = \mathsf{de}(e_1)$ and $u_2 = \mathsf{de}(e_2)$ throughout. Suppose that the liabilities $l_{e_1} = l_{e_2}$ and u_1, u_2, v_1, v_2 are pairwise distinct nodes. A *debt swap* σ of e_1 and e_2 creates a new network $\mathcal{F}^\sigma = (G, \mathsf{de}, \mathsf{cr}^\sigma, \mathbf{l}, \mathbf{a}^x, \mathbf{f})$ where $\mathsf{cr}^\sigma(e_1) = v_2$, $\mathsf{cr}^\sigma(e_2) = v_1$, and $\mathsf{cr}^\sigma(e) = \mathsf{cr}(e)$ for all $e \in E \setminus \{e_1, e_2\}$.

Throughout the paper, e_1 and e_2 are used to refer to the trading edges, while u_1, u_2, v_1 and v_2 refer to the respective debtor and creditor banks. Observe that a debt swap may introduce multi-edges even in initially simple graphs. The edge-ranking rule of a bank can be interpreted as a priority ranking of contracts (instead of the creditor banks). When performing a debt swap of two edges e_1 and e_2, we assume that u_1 allocates payments towards the new edge with creditor v_2 in the same way as for v_1 before the trade. Hence, besides exchanging the creditor of one outgoing edge, the permutation of outgoing edges (and thus the priority ranking) remains the same.

We proceed with a small example that shows some of the interesting effects of debt swaps.

Example 1. Consider the initial financial network in Fig. 1 (left) with edge-ranking allocation rules. Bank v_1 first pays off all debt to u_2 before directing payments to w_1, i.e, $\pi_{v_1}(1) = (v_1, u_2), \pi_{v_1}(2) = (v_1, w_1)$. All other banks pay towards their unique outgoing edge. Consider the clearing state. For bank v_2 the external assets of 1 equal her total liabilities L_{v_2}. Thus, v_2 can settle all her debt towards w_2. For the other edges, suppose there are positive payments on edge (u_2, v_2). Then, by definition, u_2 has to have these assets available, so v_1 must pay this amount to u_2. By the same argument, u_1 must have positive total assets and pay them to v_1. However, u_1 obviously has no assets, and there are no payments on these edges. Due to the allocation rule of v_1, there are no payments in the cycle of v_1 and w_1, unless v_1 fully pays off (v_1, u_2) first (for which she needs assets from u_1). The clearing state has a payment of 1 on (v_2, w_2). The total assets are 1 for v_2 and w_2, and 0 for all others.

Now consider the financial network in Fig. 1 (right) after the swap of edges (u_1, v_1) and (u_2, v_2) to (u_1, v_2) and (u_2, v_1). Obviously, v_2 can still pay off her single outgoing edge. After the swap, there is a cycle including v_1 and u_2 over prioritized edges. When the payments on both edges are raised to 1, the fixed-point properties remain satisfied. Since v_1 pays off all debt on the edge with highest

priority, she will direct any additional assets towards the edge of her second priority to w_1. Here a new cycle with w_1 over prioritized edges arises, and the fixed-point conditions allow to further raise payments along that cycle to 1.

The additional payments after the swap result from the emergence of the cycle between u_2 and v_1. The total assets of v_1, u_2 and w_1 are increased to $a_{v_1} = 2, a_{u_2} = 1$ and $a_{w_1} = 1$ after the swap. For the total assets of the other banks the swap is inconsequential.

A debt swap of two edges e_1 and e_2 maintains the local network properties of the debtors u_1 and u_2. It has no effect on the set of liability values. Moreover, it does not impact their incoming edges. As such, we are interested in the gains and losses in total assets of the creditors v_1 and v_2, since (arguably) they are the key actors to implement a debt swap. The following classification of debt swaps slightly extends the one in [1]. Observe that the debt swap explained in Example 1 is semi-positive and Pareto-improving.

Definition 2 (Swap Types)**.** A debt swap σ of edges e_1 and e_2 is called (1) *positive* if $a^\sigma_{v_1} > a_{v_1}$ and $a^\sigma_{v_2} > a_{v_2}$, (2) *semi-positive* if $a^\sigma_{v_1} \geq a_{v_1}$ and $a^\sigma_{v_2} \geq a_{v_2}$ and exactly one inequality is strict, (3) *Pareto-improving* if $a^\sigma_v \geq a_v$ for all $v \in V$ and at least one inequality is strict.

3. Characterization

To reason about debt swaps, we first provide a structural analysis of the clearing payments before and after debt swaps. We obtain a characterization of semi-positive and positive swaps. The analysis substantially generalizes techniques from [1] and yields two major insights. First, there are no positive swaps. Second, every semi-positive debt swap yields a Pareto-improvement in the *entire network*, i.e., no bank in the financial network is harmed while at least one bank strictly profits. Thus, semi-positive swaps can only improve the recovery rate of each bank (i.e., the ratio of available assets over liabilities). Consequently, the payment assigned to each debt contract does not decrease. Interestingly, the converse also holds: Every Pareto-improving debt swap is semi-positive.

Similar to [1], we use an auxiliary network with the same payments as the resulting network after the debt swap. In this auxiliary network, we replace the swap edges with auxiliary sources and sinks with appropriate assets representing the payments on the swap edges. For a contradiction, we assume that the debt swap is positive. By tracing the payments in the auxiliary network, we show that payments in the original network before the swap did not correspond to the maximum clearing state. This contradiction shows that positive swaps cannot exist. As a rather direct consequence from this characterization, we obtain that semi-positive swaps lead to a Pareto-improvement in the entire network.

The main distinction from the proofs in [1] for proportional payments is that these payments have limited structure in the auxiliary network and, as such, are easier to handle. Instead, our proofs rely only on monotonicity of the allocation rule. By arguing about a broad range of possibilities and dependencies of potential payments, we can show that these results indeed apply to all financial networks with *any monotone allocation rule*.

3.1. Structural properties

Monotonicity of the allocation rule implies that when the total assets of bank v increase, then payments on each $e \in E^+(v)$ do not decrease. Intuitively, therefore, every clearing state of a financial network can be decomposed into two clearing states of appropriate networks. We proceed in two steps: First, each bank v is assigned reduced external assets $\hat{a}^x_v \leq a^x_v$. This results in a new financial network, for which we get the first clearing state. In this state, v (partially) pays her liabilities. For the second step, we reduce each liability by the payment in the first step, and v receives external assets of $a^x_v - \hat{a}^x_v$. This again results in a new financial network, for which we get the second clearing state. The sum of the two clearing states is the clearing state of the original network.

This separability condition can be used to show that increasing the external assets of one bank Pareto-improves the total assets of every bank.

Lemma 1 (Monotonicity)**.** *Suppose two financial networks $\mathcal{F}$ and $\hat{\mathcal{F}}$ differ only in the external assets of a single bank v such that $a^x_v > \hat{a}^x_v$. For every bank $u \in V$, the total assets in the clearing state satisfy $a_u \geq \hat{a}_u$.*

The payments in a financial network emerge due to a combination of external assets and cyclic money flow. Clearly, a sink node t (with $E^+(t) = \emptyset$) is not included in any cycle. Thus, all incoming payments of a sink node originated from external assets of other banks.

Indeed, we can show that the increase of $\Delta > 0$ in external assets of a source node s (with $E^-(s) = \emptyset$) raises the total assets of all sinks in sum by at most Δ.

Lemma 2 (Non-Expansivity)**.** *For a given financial network, let $T \subset V$ be the set of all sink nodes. If the external assets of a source $s \in V \setminus T$ are increased to $a'^x_s = a^x_s + \Delta$, then the total assets of all sinks increase by at most Δ, i.e., $\sum_{t \in T} a'_t \leq \Delta + \sum_{t \in T} a_t$.*

A similar property can be shown if s is not a source bank. The argument uses an auxiliary construction that externalizes the incoming payments of bank $s = v$ by splitting v into two separate nodes s_v and t_v. The sink t_v holds the same incoming edges as v in the original graph. Similarly, source s_v has the same outgoing edges as v. Setting the external assets of s_v to the total assets a_v of v in the clearing state ensures that the payments of s_v and v are equal. We can then verify that the modified network has the same clearing state as the original network, and, in particular, t_v has the same incoming payments as v. Clearly, the nodes representing v are not contained in any cycle.

Definition 3. For a given financial network $\mathcal{F}$ with bank v, we define an adjusted network $\mathcal{H}$ as follows. We replace v by a source node s_v and a sink node t_v. Sink t_v has external assets 0 and the same set of incoming edges as v, i.e., $\mathsf{cr}_{\mathcal{H}}(e) = t_v$ for every $e \in E^-(v)$ and $E_{\mathcal{H}}{}^-(t_v) = E^-(v)$. Source s_v has external assets a_v and the same outgoing edges as v, i.e., $\mathsf{de}_{\mathcal{H}}(e) = s_v$ for every $e \in E^+(v)$, and $E_{\mathcal{H}}{}^+(s_v) = E^+(v)$. Source s_v in $\mathcal{H}$ uses the allocation rule of v in $\mathcal{F}$.

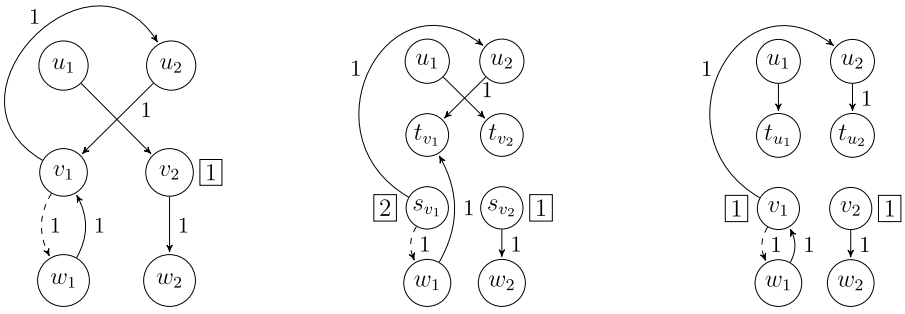


Fig. 2. The network on the left corresponds to the network $\mathcal{F}^\sigma$ in Example 1. Black edge labels indicate payments. The middle network results from applying source-sink equivalency to v_1 and v_2, whereas the right network results from applying source-sink equivalency for edges (u_2, v_1) and (v_2, u_1).

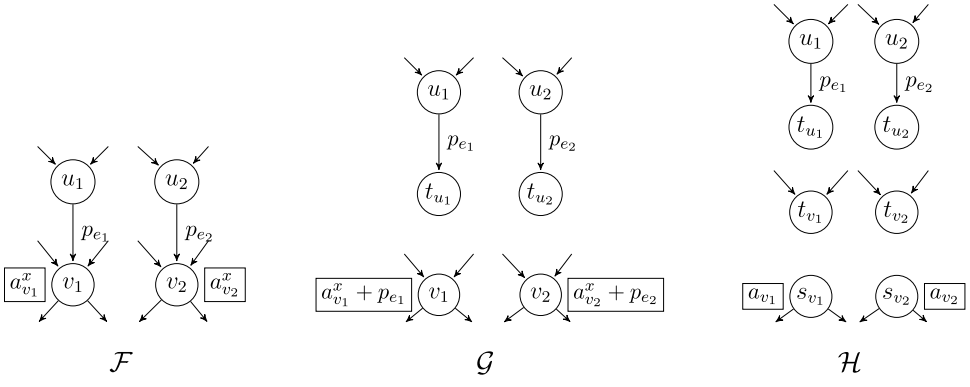


Fig. 3. Schematic illustration of networks $\mathcal{G}$ and $\mathcal{H}$ for a financial network $\mathcal{F}$.

An example construction can be seen in Fig. 2, where banks v_1 and v_2 are replaced by sources and sinks. The next lemma shows that the clearing state in the auxiliary network is equivalent to the one in the original network.

Lemma 3 (Source-Sink-Equivalency). *For a given financial network $\mathcal{F}$ and an adjusted network $\mathcal{H}$ it holds that*

(1) the networks have equivalent clearing states, i.e., $p_e^{\mathcal{H}} = p_e$ for all $e \in E$.
(2) the sink t_v has total assets $a_v - a_v^x$.

A very similar construction can be used to externalize the payments on an edge. The right image in Fig. 2 depicts an example construction.

Corollary 1 (Source-Sink-Equivalency for Edges). *For a given financial network $\mathcal{F}$ with edge e, let $u = \mathsf{de}(e)$ and $v = \mathsf{cr}(e)$. Add the sink t_v and set the target of e to $\mathsf{cr}(e) = t_v$. Increase the external assets of v by p_e. Then, the resulting financial network $\mathcal{G}$ has the same clearing state as $\mathcal{F}$, i.e., $p_e^{\mathcal{G}} = p_e$, for all $e \in E$.*

3.2. Existence of debt swaps

Example 1 shows that semi-positive and Pareto-improving debt swaps can exist, for both proportional and edge-ranking rules.

Observation 1. *For proportional payments and edge-ranking rules, financial networks with semi-positive and Pareto-improving debt swaps exist.*

Our main result in this section shows that every financial network with a monotone allocation rule has no positive debt swap.

Theorem 1. *A financial network with monotone allocation rules allows no positive debt swap.*

To prove this result, we assume for a contradiction there exists a network $\mathcal{F}$ with a positive swap of edges e_1 and e_2. To analyze the payments in $\mathcal{F}$, source-sink-equivalency is applied to the swap edges. Let $\mathcal{G}$ denote the resulting network, see Fig. 3 for an illustration. To accommodate for the reduced incoming payments $a_{v_1} - a^x_{v_1} - p_{e_1}$ of v_1, the external assets are set to $a^x_{v_1} + p_{e_1}$. The external assets of v_2 are set analogously.

The network $\mathcal{H}$ is obtained by additionally applying source-sink equivalency to v_1 and v_2 in $\mathcal{G}$. Then, t_{v_1} and t_{v_2} receive incoming payments of $a_{v_1} - a^x_{v_1} - p_{e_1}$ and $a_{v_2} - a^x_{v_2} - p_{e_2}$, respectively. This construction has no impact on the payments in the clearing state.

Analogously, the networks $\mathcal{G}^\sigma$ and $\mathcal{H}^\sigma$ are constructed for the network $\mathcal{F}^\sigma$ after the debt swap. The networks $\mathcal{G}, \mathcal{G}^\sigma$ and $\mathcal{H}, \mathcal{H}^\sigma$ have the same cycles but differ in external assets, resp.

The next technical lemma characterizes the effects of increasing external assets of s_{v_1} and s_{v_2} in $\mathcal{H}$.

Let $\mathcal{H}_\Delta$ be the resulting network when increasing the external assets of s_{v_1} and s_{v_2} in $\mathcal{H}$ by $\delta_{s_{v_1}} \in [0, a^\sigma_{v_1} - a_{v_1}]$ and $\delta_{s_{v_2}} \in [0, a^\sigma_{v_2} - a_{v_2}]$, respectively. We use Δ_w to denote the increase in total assets of bank w, i.e., $\Delta_w = a_w^{\mathcal{H}_\Delta} - a_w$. We use $\mathcal{F}_\Delta$ to denote the network when reversing the source-sink equivalency both for t_{v_1}, s_{v_1} and t_{v_2}, s_{v_2} as well as for t_{u_1} and t_{u_2}.

Lemma 4. *For $\mathcal{H}_\Delta$ and $\mathcal{F}_\Delta$ the following are true:*

(1) $\delta_{s_{v_1}} + \delta_{s_{v_2}} = \Delta_{t_{v_1}} + \Delta_{t_{v_2}} + \Delta_{t_{u_1}} + \Delta_{t_{u_2}}$.
(2) If $\delta_{s_{v_1}} > 0$ *or* $\delta_{s_{v_2}} > 0$, *then* $\delta_{s_{v_1}} \neq \Delta_{t_{v_1}} + \Delta_{t_{u_1}}$ *and* $\delta_{s_{v_2}} \neq \Delta_{t_{v_2}} + \Delta_{t_{u_2}}$.

Suppose there exists a positive debt swap for a financial network $\mathcal{F}$, in which the total assets of both creditor banks v_1 and v_2 strictly increase by $\delta_{v_1} > 0$ and $\delta_{v_2} > 0$. In $\mathcal{H}$ the debt swap is simulated by increasing the total assets of source nodes s_{v_1} and s_{v_2} by δ_{v_1} and δ_{v_2}, respectively. We will show that there exists a suitable set of values for δ_{v_1} and δ_{v_2} such that v_1 receives δ_{v_1} additional incoming payments. Intuitively, then, the pairs u_1, v_1 and u_2, v_2 are each contained in the same cycle. Hence, payments could be raised along the cycles, which contradicts maximality of the clearing state. This idea can be used to complete the proof of Theorem 1.

The theorem shows that both creditors participating in a debt swap cannot simultaneously strictly increase their total assets. In the next part, we consider the payments on the edges e_1 and e_2 before and after a generic swap σ. The first proposition shows they cannot both be strictly increased.

Proposition 1. *For every financial network, there is no debt swap σ such that $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} > p_{e_2}$.*

Proposition 2. *For every semi-positive debt swap σ, we have either $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} \leq p_{e_2}$, or $p^\sigma_{e_2} > p_{e_2}$ and $p^\sigma_{e_1} \leq p_{e_1}$.*

Towards semi-positive swaps, the total assets of v_1 and v_2 Pareto-improve. Perhaps surprisingly, the second main result of this section shows that a swap σ is semi-positive if and only if it Pareto-improves the assets of every bank in the network. In particular, no bank in the financial network is harmed while at least one bank strictly profits. The proof of the theorem uses our previous results on source-sink-equivalency and an auxiliary network $\mathcal{H}$.

Theorem 2. *For every financial network with monotone allocation rules, a debt swap is semi-positive if and only if it is Pareto-improving.*

For edge-ranking payments it is crucial that the ranking of debtors remains the same, even if, say, u_1 already had another existing edge to v_2 before the swap. Suppose that after the swap, the swap edge would get merged with the existing one to the better position in the ranking. Then creditors of edges ranked in between could suffer from lower payments after the swap. Such a swap with subsequent merge could be semi-positive without being Pareto-improving. In contrast, for proportional payments a subsequent merge of edges is unproblematic, since the payments towards each creditor are determined by the overall proportion of debt but not the splitting into individual edges.

4. Debt swap dynamics

When two creditors perform a debt swap, the structure of the underlying graph changes. This might enable further swaps which potentially further increase the assets of the involved banks. In this way, the swap operation naturally gives rise to *dynamics* where, in each step, a pair of creditors agree to perform a swap in the current network. Conceptually, semi-positive swaps are very attractive – they are equivalent to Pareto-improving swaps and improve the situation not only for the creditors but for the entire network.

In this section, we consider natural distributed dynamics of debt swaps. Given an initial financial network with monotone allocation rules, our goal is to find a small number of semi-positive debt swaps to reach a local optimum, i.e., a network structure without semi-positive swaps. We denote this as the SemiSwap problem. While all banks (weakly) profit from a semi-positive swap, we are interested in semi-positive swaps that are v-improving, i.e., *strictly increase the assets of a given bank* v. This is natural, e.g., when v is a creditor that profits strictly in every such swap, or v is a bank that is "too big to fail" and should strictly profit from network changes. We denote the problem with semi-positive v-improving swaps by SemiSwap(v). Given a financial network, can we reach a local optimum by a sequence of polynomially many semi-positive v-improving swaps?

We consider this question for the broad class of edge-ranking rules. In a given network with m edges, there are at most $\binom{m}{2}$ edge swaps. Recall that we can compute the clearing state for edge-ranking rules in strongly polynomial time [6]. Hence, for a given network, we can also enumerate all semi-positive debt swaps in strongly polynomial time.

4.1. Sequences of semi-positive swaps

Characterization. Consider a bank v with edge-ranking rule. v pays her outgoing edges in the order of a permutation π_v until she runs out of funds. Assume v cannot settle all debt. Let e denote the unique highest-ranked outgoing edge of v that is not fully paid. We call e the *active edge* of v. Clearly, all edges e' with higher priority than e are fully paid for, i.e., they are *saturated*. Additionally, all edges e' with lower priority than e receive no payments. Note that the set of all active edges is *acyclic* – we can strictly increase the feasible payments along any cycle, which is impossible when the clearing state has maximal feasible payments.



Fig. 4. Minimal example of an extension swap where all edges have weight $M > k$ and edge labels indicate payments in the clearing state. The swap extends the path of u_2's payments to reach v_2, increasing v_1's total assets of v_1 along the way.

Since the active outgoing edge of every bank is unique, the subgraph of active edges is simple and never contains multi-edges. When analyzing this subgraph, we use the notation for simple graphs, i.e., $e = (u, v)$ if $\mathsf{de}(e) = u$ and $\mathsf{cr}(e) = v$, for $e \in E$. More generally, we will use the standard notation (u, v) whenever it unambiguously describes one edge.

A semi-positive swap Pareto-improves the total assets of all banks. It can only increase the number of saturated edges. If a semi-positive swap saturates at least one additional edge, we call it a *saturating swap*. As a second class, *extension swaps* are non-saturating swaps with an interesting structural property. Recall that $v_1 = \mathsf{cr}(e_1)$ and $v_2 = \mathsf{cr}(e_2)$ before the trade.

Definition 4 (Extension Swap)**.** Consider a semi-positive debt swap σ of edges e_1 and e_2 with $p_{e_1} < p_{e_2}$. The swap is called *extension swap* if (1) no additional edge gets saturated and (2) in $\mathcal{F}$ there is an *active path* P, i.e., a path of active edges from v_1 to v_2, (3) $l_e - p_e > p_{e_2} - p_{e_1}$ for every $e \in P$, (4) in the clearing state $p_{e_1} = p^\sigma_{e_2}$ and $p_{e_2} = p^\sigma_{e_1}$.

In an extension swap, creditors v_1 and v_2 swap two edges e_1 and e_2 with different payments, where w.l.o.g. we assume $p_{e_2} > p_{e_1}$. The swap does not create a cycle of active edges, this would yield a new saturated edge along this cycle. Therefore, v_1 receives in $\mathcal{F}^\sigma$ additional incoming assets of $\delta_{v_1} = p_{e_2} - p_{e_1} > 0$. Since there exists a path P from v_1 to v_2 of active edges, each with residual liability of more than δ_{v_1}, all additional assets get routed to v_2. The total assets of v_2 after the swap remain the same (see Fig. 4 for an example).

The next proposition proves an interesting and very useful characterization.

Proposition 3. *For a given financial network with edge-ranking rules, every semi-positive debt swap is either a saturating or an extension swap.*

Proof. Consider a semi-positive swap σ with edges e_1, e_2 in a financial network $\mathcal{F}$. W.l.o.g. assume $a^\sigma_{v_1} > a_{v_1}$ and $a^\sigma_{v_2} = a_{v_2}$. Suppose σ is not a saturating swap. We show that σ is an extension swap by proving properties (1)-(4) of the definition.

Property (1) is trivial. As a consequence, σ does not close a new cycle of active edges; otherwise, since the set of active edges in $\mathcal{F}^\sigma$ is acyclic, an additional edge is saturated after the swap.

Construct the auxiliary network $\mathcal{H}$ by applying source-sink equivalency to edges e_1, e_2 and nodes v_1, v_2 and simulate the debt swap by increasing the external assets of s_{v_1} and s_{v_2} by $\delta_{v_1} = a^\sigma_{v_1} - a_{v_1}$ and $\delta_{v_2} = 0$, respectively. Due to Lemma 12 (3), s_{v_1} routes the entire additional assets of δ_{v_1} to $t_{v_1}, t_{v_2}, t_{u_1}$ and t_{u_2}. Every bank has at most one active outgoing edge. Hence, before the swap there is exactly one path $P = \{(s_{v_1}, w_1), (w_1, w_2), \dots\}$ of active edges that ends in exactly one of the four sinks. Since no edge gets saturated, s_{v_1} forwards all additional assets along P to that sink.

This sink must be either t_{v_2} or t_{u_2} due to Lemma 4, (2). Suppose it is t_{u_2}. Then (u_2, v_2) is not active in $\mathcal{F}$, since otherwise $(P \setminus \{(s_{v_1}, w_1)\}) \cup \{(v_1, w_1), (u_2, v_1)\}$ would be a cycle of active edges in $\mathcal{F}^\sigma$. However, if (u_2, v_2) is not active in $\mathcal{F}$, then neither is (u_2, t_{u_2}) in $\mathcal{H}$. Hence, t_{u_2} cannot be the sink of P.

As a consequence, t_{v_2} must be the sink of P. This implies existence of the active path and shows property (2). Since t_{u_1} and t_{u_2} are not involved in P and there is no new cycle of active edges, these sinks receive the same assets as before the swap. This implies the same payments on swapped edges, $p_{e_1} = p^\sigma_{e_2}$ and $p_{e_2} = p^\sigma_{e_1}$. As such, property (4) holds.

Since there are no new cycles of active edges, we see that $\delta_{v_1} = p^\sigma_{e_1} - p_{e_1} = p_{e_2} - p_{e_1}$, so indeed $p_{e_1} < p_{e_2}$. Thus, v_1 gains additional assets of δ_{v_1} from the swapped edge. Similarly, v_2 looses assets of δ_{v_1} from the swapped edge. Since $\delta_{v_2} = 0$, this implies that in $\mathcal{H}$ all additional assets of δ_{v_1} must get routed completely from s_{v_1} along P to t_{v_2}. Since no additional edge gets saturated, we must have $\ell_e - p_e > p_{e_2} - p_{e_1}$, i.e., property (3) follows. As a conclusion, σ must be an extension swap. □

We further classify extension swaps based on the activation status of the involved edges.

Definition 5 (Classes of Extension Swaps)**.** Consider extension swap σ of edges e_1 and e_2 with $p_{e_1} < p_{e_2}$. In a *non-active* swap both edges are nonactive. In a *semi-active* swap, either e_1 or e_2 is active. In an *active* swap, both e_1 and e_2 are active.

Observe that in a non-active swap, the edges e_1 and e_2 are each either fully saturated or receive no payments. Since $p_{e_1} < p_{e_2}$, we have both $p_{e_1} = 0$ and $p_{e_2} = l_{e_2}$. In a semi-active swap, $p_{e_1} = 0$ or $p_{e_2} = l_{e_2}$. Finally, in an active swap, $0 \le p_{e_1} < p_{e_2} < l_{e_2}$.

Periods, Phases, Active Extension Swaps. We now discuss the construction of short sequences of swaps and start with saturating swaps.

Observation 2. *Given an initial financial network with edge-ranking rules and m edges, every sequence of semi-positive debt swaps can contain at most m saturating ones.*

We define a *period* as a sequence of consecutive extension swaps. Every sequence of semi-positive swaps has at most $m + 1$ periods. We turn attention to non-active and semi-active extension swaps. Every swap has exactly one non-profiting creditor. We classify non-active and semi-active swaps in a period based on this bank.

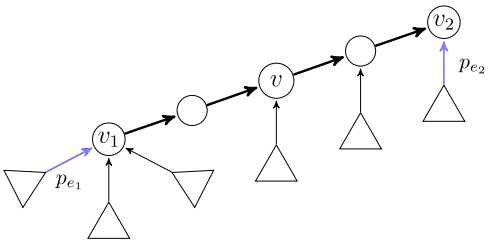


Fig. 5. Schematic picture of a v-improving semi-positive swap of active edges where payments $p_{e_2} > p_{e_1}$. Triangles indicate subtrees. When the blue edges are swapped then payments on the thick black edges strictly increase. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Lemma 5. *For every $v \in V$, every single period can contain at most $2 \cdot |E^-(v)|$ many non-active or semi-active extension swaps such that v is the non-profiting creditor.*

Proof. For each semi-active swap of this kind, one of two possible events occurs: Either (1) an incoming non-active saturated edge of v is changed to an active (non-saturated) one, or (2) an incoming active edge of v with positive payment is changed to a non-active one with 0 payment. In a non-active swap, a sequential composition of both events occurs: An incoming non-active saturated edge of v is changed directly to a non-active one with 0 payments.

In a semi-positive swap, the payment on every edge can only weakly increase. No edge gets saturated (this would end the period). Hence, no active edge can become non-active (neither saturated, nor with 0 payments). Events (1) and (2) cannot re-occur for an incoming edge of v. There are at most $2 \cdot |E^-(v)|$ events for node v in the period, and at most $2 \cdot |E^-(v)|$ many non-active or semi-active extension swaps with v as the non-profiting creditor. □

Hence, from an initial network with m edges, there are at most $O(m^2)$ saturating, non-active, or semi-active extension swaps. The challenge lies in bounding the number of active extension swaps. We define a *phase* as a sequence of consecutive active extension swaps. For bounding the length of swap sequences to a polynomial, Observation 2 and Lemma 5 allow to restrict attention to the number of swaps in a single phase.

Recall that the set of active edges is cycle-free. Every node has at most one outgoing active edge. Consequently, the set of active edges forms a forest of in-trees, where all edges are directed towards the parent node. We denote the set of these in-trees by $\mathcal{T}$. In an active extension swap, the active path P runs from v_1 to v_2. As such, we can only have active extension swaps where all four banks involved in the swap are in the same tree $T \in \mathcal{T}$, and the non-profiting creditor $v_2 \in T$ is an ancestor of all $v_1, u_1, u_2 \in T$.

We first focus on sequences of active extension swaps that strictly improve a given bank v. Afterwards, we consider the more general case of arbitrary active extension swaps, where we do not condition on a particular bank benefiting.

Improvement for a Given Bank. We analyze sequences of semi-positive swaps that are v-improving for a given bank v, i.e., in every swap v strictly profits. Note that v may not be involved in the swap as a creditor bank herself. Our main result is the following.

Theorem 3. *For every bank v in a financial network with edge-ranking rules,* every *sequence of v-improving semi-positive swaps has polynomial length. SemiSwap(v) can be solved in polynomial time.*

Proof. It suffices to prove the result for active extension swaps in a single phase. Consider the tree T containing v. Suppose there is an active extension swap, then the set of banks that strictly profit lie on the active path P, with the exception of creditor v_2. Hence, if such a swap is also v-improving, v is a node on the active path, the creditor v_1 is v or a descendant of v, and the creditor v_2 is an ancestor of v. As such, all active edges on the path P_v from v to the root of T cannot be swapped. In terms of payments, the edges whose payments are strictly increased by the swap are located in the subtree T_v rooted at v or on the path P_v. The swap keeps all other payments the same.

Note that for the swap, $p_{e_1} < p_{e_2}$, i.e., the edge e_2 with higher payment is incoming to the ancestor v_2 of v and gets swapped to $v_1 = v$ or a descendant v_1 of v. For v_2, the only incoming edge that increases in payment is the one on P_v. All other incoming edges of v_2 only keep their payments (when not being swapped) or are replaced by ones with strictly decreased payments (when being swapped). In contrast, for each edge, payments are non-decreasing over time. This implies that if edge e_2 is swapped away from v_2 for some edge e_1, edge e_2 cannot re-appear at v_2 via a swap with e_1 or any edges swapped with e_1. For a schematic illustration, see Fig. 5.

Hence, a single phase contains at most m^2 many v-improving active extension swaps. □

This result is potentially surprising, since for other allocation rules there are indeed extremely long sequences of such swaps. Indeed, with proportional rules there are exponentially long sequences of v-improving semi-positive swaps.

Proposition 4. *There is a financial network with n banks, n edges and proportional allocation rule, from which there is a sequence of $\Omega(2^n)$ many v-improving semi-positive swaps.*

Proof. We construct the initial network with a simple underlying graph as follows. There are n nodes and n edges. Every node has at most one incoming edge. For simplicity, all edges have a sufficiently large liability of 2^n. There is a single node v with out-degree 2, a

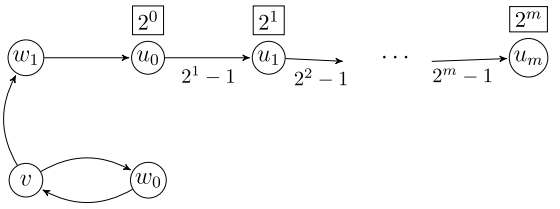


Fig. 6. Financial network with proportional allocation rules that admits a sequence of debt swaps with exponential length, where $m = n - 4$. Each edge has weight 2^n and edge labels indicate payments.

single node with outdegree 0. All nodes have in-degree 1. The node v is involved in exactly one directed cycle of two banks v and w_0 and edges (v, w_0), (w_0, v). These banks have no external assets. The second outgoing edge of v leads into a path with the remaining $n-2$ nodes. We denote the nodes in the path by $v, w_1, u_0, u_1, \dots, u_{n-4}$ in the order of their occurrence. Bank u_i has external assets of 2^i, for every $i = 0, \dots, n-4$. In the initial clearing state, there are no payments on cycle edges (v, w_0), (w_0, v), and on edges (v, w_1) (w_1, u_0). Edge (u_i, u_{i+1}) has payment $\sum_{j=0}^{i} 2^j = 2^{i+1} - 1$, for every $i = 0, \dots, n-5$. A visualization can be seen in Fig. 6.

We outline the first four steps of an exponentially long sequence for clarity. Our sequence maintains the structural invariant that the network has a single cycle involving v and a path from v to u_{n-4}. The nodes w_0 and w_1 always remain the two out-neighbors of v. They alternate in being in the cycle or on the path. The assets of v strictly improve by 2 in every swap.

First, swap edges (u_0, u_1) and (w_0, v). Then the path is $v, w_0, u_1, u_2, \dots, u_{n-4}$, and the cycle is v, w_1, u_0. Since u_0 pays its external assets of 1 in the cycle, and v splits its assets evenly among (v, w_0) and (v, w_1), the clearing state has a payment of 2 on edge (u_0, v), a payment of 1 on (v, w_0), (v, w_1) and (w_1, u_1), and a payment of $2^{i+1} - 1$ on every edge (u_i, u_{i+1}). Clearly, this is a v-improving, semi-positive swap.

Second, swap edges (u_0, v) and (u_1, u_2). Then the path is $v, w_1, u_0, u_2, \dots, u_{n-4}$, and the cycle is v, w_0, u_1. Since u_1 pays its external assets of 2 in the cycle, the clearing state has a payment of 4 on edge (u_1, v), a payment of 2 on (v, w_0), (v, w_1) and (w_1, u_0), a payment of 3 on (u_0, u_2), and $2^{i+1} - 1$ on every other edge (u_i, u_{i+1}). Clearly, this is a v-improving, semi-positive swap.

Third, swap edges (w_0, u_1) and (u_0, u_2). Then the path is $v, w_0, u_2, \dots, u_{n-4}$, and cycle is v, w_1, u_0, u_1. The clearing state has a payment of 6 on edge (u_1, v), a payment of 3 on (v, w_0), (v, w_1) and (w_1, u_0), payment 4 on (u_0, u_1). Finally, we have a payment of $2^{i+1} - 1$ on every other edge (u_i, u_{i+1}).

Forth, swap edges (u_1, v) and (u_2, u_3). Then the path is $v, w_0, u_0, u_1, u_3, \dots, u_{n-4}$, and the cycle is v, w_1, u_2. The clearing state has a payment of 8 on edge (u_2, v), a payment of 4 on (v, w_1), (w_1, u_2) and (v, w_0), (w_0, u_0). There is a payment of 5 on (u_0, u_1), 7 on (u_1, u_3) and $2^{i+1} - 1$ on every other edge (u_i, u_{i+1}).

Now at this point, we have moved u_2 into the cycle, and reset u_0 and u_1 to their original positions as the first u-nodes on the path. Thus, by repeating steps 1–3, we can bring u_0, u_1 in front of u_2 in the cycle, thus arriving at a stage with payment 14 on edge (u_2, v) and 7 on both (v, w_0) and (v, w_1). Then, by swapping (u_2, v) and (u_3, u_4), we move u_3 in the cycle and reset u_0, u_1 and u_2. Thus, we can again repeat the entire sequence. In each swap, the payment on (v, w_0) and (v, w_1) increases by exactly 1.

More generally, suppose we have nodes $v, w_1, u_0, u_1, \dots, u_j$ in the cycle in that order, and we have nodes $v, w_0, u_{j+1}, \dots, u_{n-4}$ on the path in that order. At this point, we have payments of $2^{j+1} - 1$ on both (v, w_0) and (v, w_1). Now by swapping (u_j, v) and (u_{j+1}, u_{j+2}), we have nodes v, w_0, u_{j+1} in the cycle and reset $u_0, \dots, u_j$ on the path $v, w_1, u_0, \dots, u_j, u_{j+2}, \dots, u_{n-4}$. Then we can repeat the entire sequence executed so far and move to $v, w_1, u_0, \dots, u_{j+1}$ in the cycle and $v, w_0, u_{j+2}, \dots, u_{n-4}$ on the path. The last doubling step in this exponential construction occurs when u_{n-3} moves into the cycle, and the sequence ends when the cycle is $v, w_1, u_0, \dots, u_{n-5}$ and the path v, w_0, u_{n-4}. At this point, we have made $2^{n-4} - 1$ semi-positive v-improving swaps. □

Beyond Improvement for a Given Bank. In the remainder of the section, we concentrate on arbitrary active extension swaps (without the property that they are v-improving). Bounding the length of such sequences represents a very challenging problem. We make some initial progress on this problem: For a non-trivial class of instances, we construct a *single* short sequence of such swaps in each phase.

For the remainder of the section, we focus on instances with two network conditions, namely *a constant number of active incoming edges* and *sufficient residuals*. First, suppose that throughout every node has at most d incoming active edges, for some constant d. For example, $d = 1$ means that every network created in the sequence has at most one incoming active edge for each $v \in V$. In particular, the number of active incoming edges is bounded by d when every node has an *indegree* of at most d.

Proposition 5. *Consider an initial financial network with edge-ranking rules. Every sequence of semi-positive swaps, during which every node has at most $d = 1$ active incoming edges, has polynomial length.*

Proof. In an active extension swap, the non-profiting creditor v_2 has two incoming active edges. Thus, if $d = 1$, there can be no active extension swap. All swaps are saturating, semi-active or non-active swaps. Their number is bounded by $O(m^2)$. □

When $d \geq 2$, we consider a second condition – the network should have sufficient residuals.

Definition 6 (Sufficient Residuals)**.** In a financial network $\mathcal{F}$, a *potential active swap* is a pair of active edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ such that $l_{e_1} = l_{e_2}$, $p_{e_1} < p_{e_2}$, and there is a path of active edges from v_1 to v_2. The network $\mathcal{F}$ has *sufficient residuals* if for every potential active swap, $l_e - p_e > p_{e_2} - p_{e_1}$ for all $e \in P$, i.e., the potential active swap is an active extension swap.

Intuitively, in a potential active swap all conditions for an extension swap are fulfilled, with the exception of the residual liabilities on the path P of active edges. If a network has sufficient residuals, then a potential active swap is indeed an active extension swap, since there is sufficient residual liability on P to forward the marginal increase of total assets for v_1 towards v_2. For example, if the liabilities of active edges are sufficiently high compared to clearing payments, then the increases in payments will not "exhaust" the liabilities, and we are guaranteed to maintain the property of sufficient residuals throughout the swap sequence.

Let us examine active extension swaps in networks with sufficient residuals.

Corollary 2. *For a financial network with edge-ranking rules and sufficient residuals, the pair e_1 and e_2 represents an active extension swap if and only if all conditions hold: (1) $l_{e_1} = l_{e_2}$, (2) $p_{e_1} < p_{e_2}$, (3) v_2 is an ancestor of v_1, u_1 and u_2 in an in-tree T of active edges, and (4) u_2 is not an ancestor of v_1 in T.*

Clearly, if v_2 is an ancestor of v_1, there is a path P of active edges, so conditions (1)-(4) characterize potential active swaps (and hence, active extension swaps). As a consequence, our goal now is to swap edges with small payments as high as possible in the tree. In particular, consider edges classified by their liability values. For each class, we intend to swap the active edges with lowest payments to the root of the in-tree. If there are active edges from the same class (a) at the root and (b) deeper in a different subtree, s.t. the root edge has strictly more payments, then Corollary 2 implies there exists an active extension swap.

Formally, we construct a sequence of active extension swaps in a single phase "top-down" in each in-tree. In the first stage, we only execute active swaps involving a node $v \in V_1$, where V_1 is the set containing the root of every $T \in \mathcal{T}$. If there are no more such active extension swaps (or there is a swap from another class and the phase can be ended), then the first stage ends. In the second stage, for every $v \in V_1$ with v root of T, the set V_2 contains all direct children of v in T. We consider active extension swaps that involve a node $v \in V_2$. The stage ends when no more such active extension swaps exist (or the phase can be ended by a swap of another class). More generally, in stage i we consider active extension swaps involving nodes of depth $i-1$ in the trees.

Lemma 6. *Suppose we are given an initial financial network with edge-ranking rules, sufficient residuals and constant d. For every $v \in V_i$ there is no active extension swap involving v in any stage $j > i$.*

Proof. We prove the lemma by induction. Consider stage 1 and V_1. Since $v \in V_1$ is the root of some in-tree T, it has no outgoing active edge. It can only be part of an extension swap as a non-profiting creditor. At the end of stage 1 there is no active extension swap with $v \in V_1$. Consider a liability value and suppose v has ℓ incoming edges of this liability. Then, by Corollary 2, the $\ell - 1$ edges with lowest payments in this liability class in T must be incident to v. If the edge with ℓ-smallest payment in the class is not incident to v, it must be located in a different subtree than the $\ell - 1$ other edges (otherwise there is an active extension swap with the remaining incoming edge of v from that liability class). This property holds for each class of edges with the same liabilities.

For contradiction, consider the first time at which an extension swap involving v arises in a subsequent stage $j > i$. This means there is an edge incident to v that can be swapped with one that is deeper in the tree and has less payment. However, this is impossible: In any previous extension swap the edges that received strictly increased payments lie on the path P between the creditors. Since v was not involved in any extension swap, the edges incident to v still have the payments they had at the end of stage 1. Moreover, all active edges get swapped only in the respective subtrees. As such, no active extension swap with the edges of v can arise. This proves the base case.

For the hypothesis, assume the lemma holds for all stages $k = 1, \dots, i-1$. Consider stage i and a node $v \in V_i$. At the end of stage i, there are no extension swaps involving node $v \in V_i$. Again, consider the first time at which an extension swap involving v arises in a subsequent stage $j > i$. By hypothesis, there are no extension swaps involving nodes from stages $k < i$ in this time period. Hence, in the extension swap v must be the non-profiting creditor. All other nodes must be from the subtree T_v rooted at v. We can repeat the arguments from the base case with the root and T for v and T_v. This proves the inductive step. □

It remains to construct a short sequence of active extension swaps within a single stage i. By Lemma 6, for each $v \in V_i$, no ancestor is involved in any extension swap of stage i. Hence, we restrict attention to the subtree rooted in v. We assume v is a root of a separate subtree T, and we must bound the number of active extensions swaps involving v.

Lemma 7. *For any financial network with edge-ranking rules and sufficient residuals, if every $v \in V_i$ has at most constantly many incoming active edges, then there can be at most a polynomial number of active extension swaps in stage i.*

Proof. Consider $v \in V_i$ with $d_v \le d$ incoming active edges, for some constant d. Active extension swaps always keep this number the same for every node. Since v is a non-profiting creditor in every swap of stage i, the total assets of v do not change during the stage. However, in each swap, there is at least one incoming edge of v (on the active path of the swap) that strictly increases in payments. Since payments on each edge do not decrease, the same subset of d_v incoming active edges cannot repeat during the stage. As a consequence, at the end of each swap, v has a different subset of neighbors for its incoming active edges. The number of different sets of neighbors is bounded by $\binom{n}{d_v}$, which is polynomial for constant d_v. □

Together these insights prove Theorem 4, which summarizes our result.

Theorem 4. *The SemiSwap problem for financial networks with edge-ranking rules can be solved in polynomial time if, for all financial networks encountered during the sequence, the network has sufficient residuals and every node has at most a constant number of active incoming edges.*

Open Problem. *Can the SemiSwap problem for financial networks with edge-ranking rules be solved in polynomial time?*

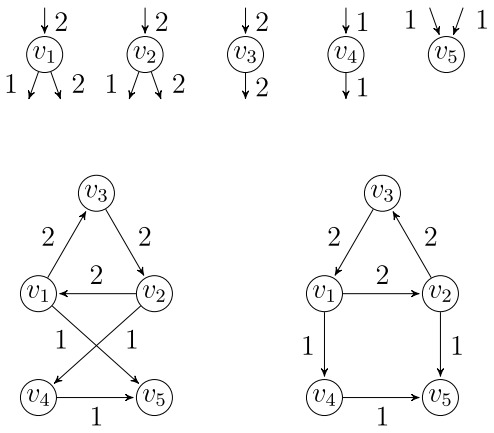


Fig. 7. Example of a set of nodes together with corresponding stubs on the top, and two graphs with consistent sets of edges below.

4.2. Sequences of arbitrary swaps

In this section, we consider sequences of arbitrary swaps. In contrast to semi-positive ones, these swaps have no natural criterion of progress. We again focus on swaps in which a single bank v has to improve strictly. Unfortunately, sequences of such arbitrary v-improving swaps can be very long, for every monotone allocation rule (including edge-ranking and proportional rules). Our results show that there are initial networks from which *every* sequence of v-improving swaps to a local optimum (in which no such swaps exist) is exponentially long.

Indeed, we obtain a more general statement. Consider a maximization problem for the total assets of a given bank v defined as follows. In an instance of the MaxAssets$_v$ problem, we are given a set of banks V. For each bank $v \in V$ we are given a non-negative value for the external assets as well as a number of incoming and outgoing *half-edges* or *stubs*. For each such stub i of v, we have a positive weight $l_i > 0$. Clearly, these stubs shall be *consistent* with at least one set of edges E among V, i.e., there must be a feasible underlying multi-graph $G = (V, E)$ in the sense that the stubs can be matched to form G. In particular, each incoming stub of weight l_i of each node v can be matched to some outgoing stub of weight l_i of some node $u \neq v$ to form an edge $e \in E$ with $\mathsf{de}(e) = u$ and $\mathsf{cr}(e) = v$. An example is depicted in Fig. 7. Consistency of stubs with *at least one* set E can be tested in polynomial time by solving a straightforward bipartite matching problem. For the remainder, we assume that it holds.

An instance of MaxAssets$_v$ has an edge-ranking rule $\mathbf{f}$ that specifies the order in which banks pay their outgoing stubs. The rule is given implicitly by a *clearing oracle* $\mathcal{O}$. A clearing oracle for rule $\mathbf{f}$ receives as input the remaining financial network $(G, \mathsf{de}, \mathsf{cr}, \mathbf{l}, \mathbf{a}^x)$ and outputs the clearing state. Note that a clearing oracle $\mathcal{O}$ can be computed in polynomial time for edge-ranking rules [6].

Observe that any set of edges E that is consistent with the stubs gives rise to a financial network and a clearing state (where the latter can be computed using $\mathcal{O}$). The goal is to find such a consistent set to maximize the total assets of a given bank v in the resulting clearing state. The objective function in the instance are the total assets for a given bank v.

Finally, for edge swaps we define a neighborhood over the set of financial networks whose edges are consistent with the stubs in the input. For a given financial network $\mathcal{F}$, the *swap* neighborhood $\mathcal{N}(\mathcal{F})$ contains all networks $\mathcal{F}'$ that can be obtained by performing an arbitrary debt swap in $\mathcal{F}$. A debt swap exchanges the matching of two pairs of stubs with pairwise disjoint incident nodes and with the same weights and thereby maintains consistency.

Proposition 6. *MaxAssets$_v$ with swap neighborhood is in the complexity class PLS whenever the clearing oracle can be computed in polynomial time.*

Clearly, as noted above, we can obtain an initial network by computing a bipartite matching (indeed, *every* network consistent with the stubs represents a bipartite matching of the stubs and vice versa). The *swap* neighborhood of a network contains $O(m^2)$ other networks. For each of these networks, the corresponding clearing state can be determined using oracle $\mathcal{O}$. Hence, the *swap* neighborhood $\mathcal{N}(\mathcal{F})$ for a given network $\mathcal{F}$ can be searched efficiently.

When locally maximizing MaxAssets$_v$ we do not necessarily restrict attention to a sequence of v-improving swaps from a given initial network to a local optimum. Instead, when given an initial network, we obtain an instance of MaxAssets$_v$ by breaking all edges into stubs. The goal is to find *any* local optimum network consistent with the stubs derived from the initial network. We show that this problem is PLS-complete [23], thus, unlikely to be solvable in polynomial time. It is easy to see that our PLS-reduction in the subsequent theorem is *tight* [24, Definition 3], i.e., it preserves solutions, neighborhoods, and objective values from the Max-2-SAT problem with *flip* neighborhood. This implies the existence of instances and starting states that require exponentially long sequences of swaps as well as PSPACE-completeness of reachability questions (see also Corollary 9 below).

Theorem 5. *There is a tight PLS-reduction from Max-2-SAT with* flip *neighborhood to MaxAssets$_v$ with swap neighborhood.*

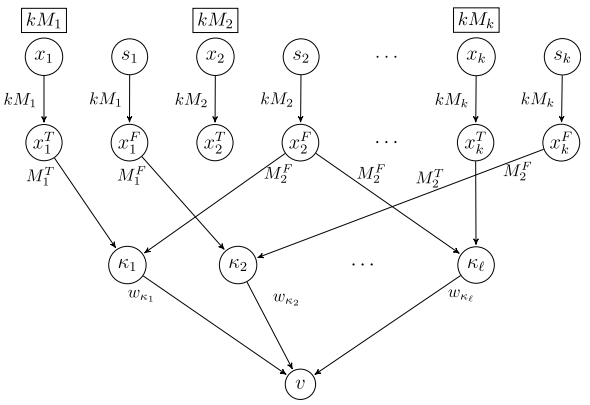


Fig. 8. Example construction for reduction with MAX-2-SAT where edge labels indicate liabilities.

Proof. An instance φ of MAX-2-SAT with the *flip* neighborhood consists of formula φ consists of k variables $x_1, x_2 \ldots, x_k$ and ℓ clauses $\kappa_1, \ldots, \kappa_\ell$ where each clause is assigned a weight w_{κ_j} and consists of at most two variables. The objective is to derive an assignment of variables that maximizes the sum of weights of satisfied clauses. In the flip neighborhood, two assignments are neighboring if they vary in the assignment of exactly one variable. We construct an input instance to MAXASSETS_v as follows. First, include banks x_i, x_i^T and x_i^F for every variable and a bank κ_j for every clause. Additionally add bank v and auxiliary bank s_i for all $i \in [k]$.

Before specifying the stubs and their weights for every bank, we describe a binary encoding for every bank corresponding to a literal. Choose integer ℓ such that $2^\ell \geq w_{\kappa_j}$ for all j and $\ell > 1 + \log k$. Then, set the ℓth lowest bit to 1. Use the bits 2 to $1 + \log k$ to encode the index i of the literal and set the first bit from the right to 0 if the literal is $\neg x_i$ and to 1 otherwise. All other bits are set to 0. Observe that the weights assigned to each literal are pairwise different and at least w_{κ_j} for all j. The use of these different weights limits the number of possible matchings of the stubs. In particular, when assigning one of the unique weights to only two stubs, the existence of a specific edge in each set E can be enforced. We discuss the exact effects of the weights below. For simplicity, denote the resulting value for each bank x_i^T by M_i^T, and for each x_i^F by M_i^F.

We construct the stubs as follows. Bank v has outdegree zero and ℓ incoming stubs such that there exists exactly one with weight w_{κ_j}. Each bank κ_j has exactly one outgoing stub with weight w_{κ_j}. Furthermore, if literal x_i appears in the clause, then we add an incoming stub with weight M_i^T – or a stub with weight M_i^F if $\neg x_i$ appears. For every bank x_i^T, there is a stub for each clause the literal x_i appears in, each with weight M_i^T. The analogous condition holds for x_i^F. Using $M_i = \max\{M_i^T, M_i^F\}$, we create a single incoming stub with weight kM_i for each x_i^T and x_i^F. Lastly, for every bank x_i and s_i, we set the indegree to zero and create one outgoing stub with weight kM_i. Observe that no set of edges E consistent with the stubs contains multi-edges.

Finally, bank x_i has external assets of kM_i, for every $i \in [k]$.

Overall, the construction of the MAXASSETS_v instance is polynomial in the size of the instance of MAX-2-SAT. For an example construction see Fig. 8.

For the clearing oracle, each bank with at most one outgoing stub has a trivial clearing payment – simply assign all assets to the outgoing stub (if any) until it is fully paid. The banks x_i^T and x_i^F can have multiple outgoing stubs. We argue below that in all consistent networks each of these banks has either 0 assets or enough assets to fully pay all outgoing stubs. This implies that for a given consistent network and a given bank, the payments of the bank are the same, for *every* allocation rule (edge-ranking, proportional, etc., even for non-monotone ones).

It is straightforward to see whether a bank has 0 assets or can pay all edges fully. Thus, defining a clearing payment is possible in polynomial time for every bank in every consistent network, for every (or, more precisely, independent of the specific) allocation rule.

We now prove correctness of the reduction. Observe that, by construction, every consistent set of edges must contain edge (κ_j, v) since for each w_{κ_j} there exists exactly one bank with an outgoing and one bank with an incoming stub. Similarly, x_i^T is the only bank that has outgoing stubs with weight M_i^T. Additionally, for every bank κ_j with an incoming stub with weight M_i^T the corresponding clause includes literal x_i. Hence, there is an edge (x_i^T, κ_j) in every consistent set of edges if and only if x_i appears in clause κ_j. The same argument extends to literals x_i^F. Thus, the consistent sets of edges only vary in the edges of banks x_i and s_i to x_i^T and x_i^F. Note that for every i it is possible to either choose (1) edges (x_i, x_i^T) and (s_i, x_i^F) or (2) (x_i, x_i^F) and (s_i, x_i^T), each with the same edge weight kM_i. In case (1) we interpret literal x_i to be assigned with value TRUE and $\neg x_i$ with value FALSE. On the other hand, in case (2) literal $\neg x_i$ is interpreted as TRUE and x_i as FALSE. In conclusion, every consistent set of edges corresponds to a truth assignment of variables in φ and vice versa.

By construction, exactly one of the banks x_i^T or x_i^F has strictly positive incoming assets. This bank defines the corresponding assignment for the variable. By construction, this bank is then able to settle all debt (while the other one settles no debt at all). As a consequence, bank κ_j has a fully saturated incoming edge if and only if there exists a satisfying literal in the corresponding clause. Then, by construction, κ_j can settle all debt towards v. As a result, for any consistent network, the incoming assets of v equal the

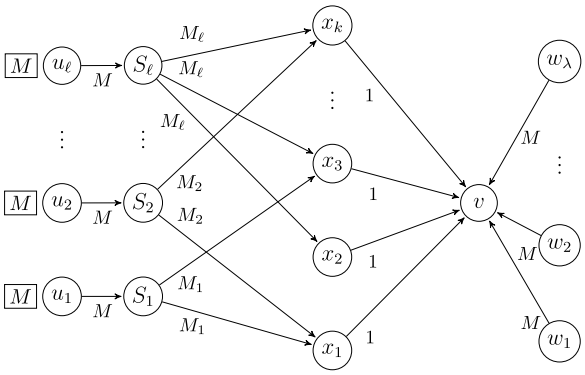


Fig. 9. Example construction for reduction with Set Cover where $\lambda = \ell - c$. Edge labels indicate liabilities.

sum of weights of satisfied clauses in φ for the corresponding assignment. In other words, the local optima for the given instance of Max-2-SAT correspond exactly to the local optimal networks w.r.t. v-improving debt swaps. □

This shows that MaxAssets$_v$ with *swap* neighborhood is PLS-complete. Moreover, tightness of the reduction implies that there exist initial states from which every execution of the standard local search algorithm requires exponentially many steps to reach a local optimum [24,25]. The standard local search algorithm for MaxAssets$_v$ yields a sequence of debt swaps.

Corollary 3. *It is PLS-complete to compute a local optimum for MaxAssets$_v$ with swap neighborhood. There is an instance of MaxAssets$_v$ and an initial network $\mathcal{F}$ such that every sequence of arbitrary v-improving swaps from $\mathcal{F}$ to a local optimum has exponential length.*

In the reduction of Theorem 5, each possible network has the property that a bank has either 0 assets or enough assets to fully pay all outgoing stubs. The clearing oracle in these networks is trivial, and all payments are completely independent of the allocation rule.

Corollary 4. *The results of Corollary 3 generalize to MaxAssets$_v$ with the swap neighborhood for any monotone allocation rule with efficient clearing oracle.*

5. Optimization

In this section, we study (global) optimization problems based on sequences of debt swaps. In particular, for a given initial network, we are interested in constructing sequences of debt swaps to a reachable network that maximize either the resulting total assets of a specific bank or the sum of total assets for all banks. We consider sequences of semi-positive swaps, v-improving swaps, arbitrary swaps, and also a restriction where a specific bank v must be involved in every swap.

Before considering sequences of swaps, we prove a slightly more general result. Therefore, consider the MaxAssets$_v$ problem defined in Section 4.2 without the neighborhood relation. Instead of local optima we are now interested in *global* ones. In the same way, we define a MaxAssets problem, where instead of the assets of v the objective function is the sum of total assets of all banks.

We again assume to have access to a clearing oracle as discussed in Section 4.2. Similar to the property in the proof of Theorem 5, our reductions always ensure that every reachable network has a clearing state that is independent of the allocation rule and can be computed in polynomial time. As such, all results in this section apply beyond edge-ranking rules to all monotone allocation rules. Additionally, none of the results in this section relies on multi-edges. For this reason, we adopt the notation for simple graphs.

Theorem 6. *MaxAssets and MaxAssets$_v$ are NP-hard.*

Proof. We first show the statement for MaxAssets$_v$. While we eventually focus on the Vertex Cover problem in Corollaries 5 and 6 below, we will describe a reduction from the more general Set Cover problem to avoid confusion in the reference to graph-theoretic concepts. An instance of the Set Cover problem is given by items $\mathcal{X} = \{x_1, x_2, \dots, x_k\}$ and sets $S_1, S_2, \dots, S_\ell$ such that $\bigcup_{j\in[\ell]} S_j = \mathcal{X}$, where $[\ell] := \{1, 2, \dots, \ell\}$, as well as an integer $c > 0$. The goal is to decide whether or not there exists a set cover, i.e., a subset $C \subset [\ell]$ with $\bigcup_{j\in C} S_j = \mathcal{X}$, that has cardinality $|C| = c$. We construct an instance of MaxAssets$_v$ with banks v, x_i for every item and S_j for every set. Moreover, we add banks $u_1, u_2, \dots, u_\ell$ and $w_1, w_2, \dots, w_{\ell-c}$ for some integer c satisfying $0 < c < \ell$.

Similar to the proof of Theorem 5 we define a value for every set S_j in binary encoding to enforce a subset of edges in every set E. First, let d be the smallest integer satisfying $2^d \geq \max_j |S_j|$ and $d > \log \ell$. Then, set the dth lowest bit to 1 and encode index j using the lowest $\log \ell$ bits. All other bits are set to 0. For simplicity, denote the resulting value by M_j for every j and let $M = \max_j M_j$. Observe that all values M_j are pairwise different and lower bounded by $\max_j |S_j|$. The number of bits is polynomial in the input size.

Banks u_j and w_j each have no incoming but one outgoing stub with weight M where the external assets of u_j are set to the same value. If item x_i is element of set S_j in the input instance add an incoming stub to bank x_i and an outgoing stub to bank S_j each with weight M_j. Moreover, every bank S_i has one incoming stub with weight M. Additionally, include an outgoing stub with weight 1 to each x_i and k incoming stubs with weight 1 to v. Finally, add $\ell - c$ incoming stubs to v with weight M.

Banks S_j are the only banks with more than one outgoing stub. We will argue below that every S_j either has no assets or enough to settle all debt. For this reason, the clearing state can be computed efficiently for every monotone allocation rule.

Clearly, the construction can be performed in polynomial time with respect to the input size of the Set Cover instance. Fig. 9 illustrates an example construction.

By construction, every consistent set of edges must contain (S_j, x_i) with weight M_i if element x_i is included in set S_j. Moreover, since only banks x_i have an outgoing stub with weight 1 and v has k incoming stubs with weight 1, edges (x_i, v) must be contained in every set of edges. Finally, banks w_i and u_i each must have an edge either towards v or a bank S_j. If bank S_j has an incoming edge from a bank w_i it has total assets of zero, or otherwise enough assets to pay off all debt if it has an incoming edge from u_i.

We claim that there exists a set cover with size c if and only if there exists a consistent set of edges such that v has total assets of $M(\ell - c) + k$ in the corresponding network.

Suppose there exists a set cover $C \subset [\ell]$ with $|C| = c$ for the input instance. Then, for every $j \in C$ include an edge from u_j to S_j. As a result, S_j has incoming payments of M. For the remaining $\ell - c$ banks S_j, choose an incoming edge from a bank w_i each. Hence, no bank S_j with $j \notin C$ has positive incoming payments. Observe that there are exactly $\ell - c$ many remaining banks u_i. For every one of them include the edge (u_i, v) yielding incoming assets of at least $M(\ell - c)$ for v in the resulting network. Observe that the set of banks S_j with incoming assets corresponds to the collection C covering all elements. Hence, by construction of M_j every bank x_i receives incoming assets of at least 1 and forwards the payments to v. In conclusion, v has total assets of $M(\ell - c) + k$.

For the other direction, assume there exists a network with consistent set of edged where v has total assets of $M(\ell - c) + k$. Because the weights of edges (x_i, v) are restricted to 1 and for sufficiently large M, bank v can only achieve incoming assets of at least $M(\ell - c)$ if $\ell - c$ banks u_i pay all their assets directly to v. The remaining assets of k then must be payments from banks x_i. Hence, every x_i receives payments of at least 1. Consequently, there exists a collection of banks S_j with cardinality c covering all x_i.

To show the statement for MaxAssets, add B banks $b_1, b_2, \dots, b_B$ where b_i has an incoming stub with weight $B + i$ and an outgoing stub with weight $B + i + 1$, for $i < B$. Add an outgoing stub with weight $B + 1$ to v and incoming stub with weight $2B$ to b_B. Observe that for a sufficiently large B there exist exactly two stubs with weight $B + i$. Hence, there exists only one consistent set of edges that joins the stubs to form a path from v to b_B. Every bank b_i on the path has the same total assets as v. When B is sufficiently large, say, $B = 3(k + \ell)$, then maximizing the sum of total assets in the network reduces to maximizing the total assets of v. □

The reduction directly implies the following hardness of approximation.

Corollary 5. *MaxAssets and MaxAssets$_v$ are APX-hard.*

Proof. Recall that Vertex Cover is APX-hard in 3-regular graphs [26]. Interpreting a 3-regular graph with n' nodes as an instance of set cover, we obtain $\ell = n'$ sets and $k = 3n'$ elements. Clearly, every feasible set cover in these instances has size $\Theta(n')$. Applying the reduction from Theorem 6, we observe that a set cover of size c corresponds to bank v obtaining assets $M(\ell - c) + k$, where $M > \ell$. As such, in these instances the assets are in $M \cdot \Theta(n')$ for every consistent network corresponding to a feasible set cover. As a consequence, the constant-factor approximation gap of Vertex Cover translates to a constant-factor gap for optimizing the assets of v. Moreover, since the adaptation to maximizing the sum of all assets requires only $\Theta(k + \ell) = \Theta(|V|)$ additional banks, the result extends to this case as well. □

We proceed to show how this result implies hardness results for sequences of swaps. In the instances constructed in the proof of Theorem 6, set S_j is included in the cover C if and only if bank S_j has positive incoming assets. Every network with consistent set of edges corresponds to some collection of sets C, and vice versa. Suppose we are given such a network for a collection C. We can perform a debt swap where the debtor of one bank S_j is swapped from a bank w_h to a bank u_i or the other way around. In this way, we remove/add S_j from/to the corresponding collection of subsets. Thus, we can reach the network corresponding to the optimal set cover C^* via a sequence of at most $|(C \cup C^*) \setminus (C \cap C^*)|$ many (arbitrary, not necessarily v-improving) debt swaps.

Corollary 6. *For a given financial network with monotone allocation rules, an efficient clearing oracle, and a bank v, it is APX-hard to compute a sequence of arbitrary debt swaps such that (1) the total assets of v are maximal or (2) the sum of total assets is maximal.*

We strengthen this result as follows. Suppose we start with the collection $[\ell]$, i.e., all sets are in the collection. The optimal set cover C^* can be obtained by iterative removal of sets from $[\ell]$. Each removal represents a debt swap in the corresponding financial network. It is straightforward to verify that each such debt swap is v-improving. Hence, from the initial network corresponding to $[\ell]$, there is a sequence of at most ℓ v-improving debt swaps to the network corresponding to C^*.

Corollary 7. *For a given financial network with monotone allocation rules, an efficient clearing oracle, and a bank v, it is APX-hard to compute a sequence of v-improving debt swaps such that (1) the total assets of v are maximal or (2) the sum of total assets is maximal.*

Semi-Positive Swaps. In contrast to v-improving swaps, the reduction in Theorem 6 does not readily extend to sequences of semi-positive swaps. In the previous section, we proved a number of positive results for semi-positive swaps (and edge-ranking rules).

Here we show hardness of approximation for maximizing assets via choosing an optimal sequence of semi-positive swaps. Our construction exploits semi-positivity more directly since it restricts the set of reachable networks from a particular initial network – solving only the MaxAssets$_v$ problem without this restriction is easy in the instances we consider.

Theorem 7. *For a given financial network with monotone allocation rules, an efficient clearing oracle and a bank v, it is NP-hard to compute a sequence of semi-positive debt swaps such that (1) the total assets of v are maximal and (2) the sum of total assets is maximal.*

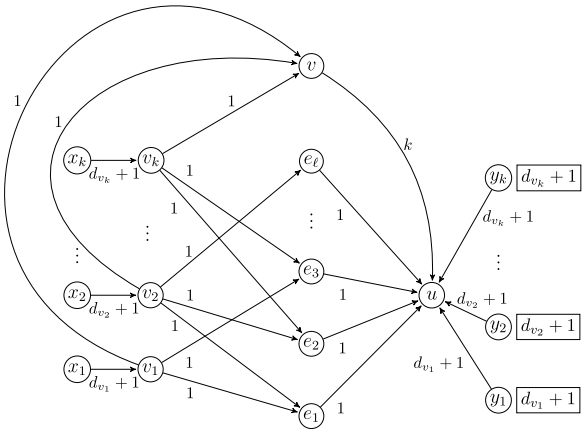


Fig. 10. Example construction for reduction with INDEPENDENT SET where edge labels indicate liabilities.

Proof. Towards (1), consider an instance of the INDEPENDENT SET problem with k nodes and ℓ edges. The objective is to derive a maximum set of nodes I such that no nodes in I are connected with an edge. We construct a financial network with banks $v_1, v_2, \dots, v_k$ representing the nodes and banks $e_1, e_2, \dots, e_\ell$ representing the edges in the given instance. Additionally, add banks $v, u, x_1, x_2, \dots, x_k$ and $y_1, y_2, \dots, y_k$. Further, if node v_i is incident to edge e_j in the input instance then include (v_i, e_j) with weight 1 in the financial network. Additionally, every bank v_i has an edge (v_i, v) with weight 1. Bank u has incoming edges (e_j, u) with weight 1 for all $j \in [\ell]$ and (v, u) with weight k. Finally, add edges (y_i, u) and (x_i, u_i) for $i \in [k]$ each with weight $d_{v_i} + 1$, respectively, where d_{v_i} is the degree of v_i in the input. Set the external assets of every bank y_i to $d_{v_i} + 1$. An example of the construction can be seen in Fig. 10.

It can be easily verified that in all networks constructed via semi-positive swaps, every bank v_i with at least two outgoing edges will either have total assets of 0 or receive sufficient assets to clear all debt. As a consequence, the clearing state of every network can be computed efficiently and is independent of the allocation rule. Thus, the result applies for all monotone allocation rules.

We now show that there exists an independent set with cardinality c if and only if there exists a sequence of semi-positive swaps with resulting total assets of c for v. Suppose there exists an independent set I with cardinality c. Swap the edges (x_i, v_i) with (y_i, u), for each $i \in I$. Then, every bank v_i pays assets of 1 to v and distributes the remaining assets of d_{v_i} fully among banks e_j. Since I is an independent set, the payments by each v_i get fully forwarded to u via the incident banks e_j. As a result, all external assets are still forwarded to u and hence all performed swaps are semi-positive. At the same time, v receives total assets of c.

For the other direction assume there exists a sequence of semi-positive debt swaps such that the resulting total assets of v are given by c. First, let us observe that no swap of edges with weight 1 can be semi-positive. The only debt swap where all creditors and debtors are pairwise distinct is given by two edges (v_i, e_j) and $(v_{i'}, e_{j'})$. However, since such a swap cannot create a new cycle and there exists no path from e_j to $e_{j'}$ and vice versa, no bank can strictly profit while the other bank is not harmed. Hence, it suffices to restrict our attention to swaps between u and a bank v_i.

Bank v can only receive payments of 1 from each bank v_i. Thus, at least c of the banks v_i must have positive total assets. This implies that u and at least c banks v_i must have executed semi-positive swaps. This only happens if all such banks v_i forward all assets to u. As a direct implication, no two distinct banks v_i and $v_{i'}$ pay assets to the same bank e_j. In conclusion, the banks represent an independent set of cardinality c in the input instance.

For statement (2) we slightly adapt the construction. Replace the edge (v, u) by a path of $B + 1$ edges and B additional banks $w_1, w_2, \dots, w_B$. Assign each edge (w_i, w_{i+1}) weight $B + i$, edge (v, w_1) with weight B and (w_B, u) with weight $2B$. Indeed, it suffices to set B to a sufficiently large value of, say, $B = 6k^2$. When executing one additional swap with a node v_i and u, the sum of assets of banks w_i increases by at least $6k^2$ whereas the sum of assets in the remaining graph is upper bounded by $5k^2 + k$. For this reason, it is always optimal to perform debt swaps to maximize the assets of v and, thus, the cardinality of the corresponding independent set. Maximizing the sum of total assets reduces to maximization of assets of v. □

The INDEPENDENT SET problem cannot be approximated within $\Omega(k^{1-\epsilon})$, unless $\mathsf{P} = \mathsf{NP}$ where k is the number of nodes [27]. In our reduction, v has total assets of c if and only if there exists an independent set of size c in the given INDEPENDENT SET instance. If the input instance of INDEPENDENT SET consists of k nodes, then the number n of banks in the construction is bounded by $O(k^2)$. Therefore, it is straightforward to argue that, for any constant $\varepsilon > 0$, we obtain NP-hardness of computing a $O(n^{1/2-\varepsilon})$-approximation to the maximal total assets of a given bank v that can be achieved using a sequence of semi-positive debt swaps. The argument applies also when maximizing the sum of all assets, since the total number of banks in the resulting network is still bounded by $O(k^2)$.

Corollary 8. *For a given financial network with monotone allocation rules, an efficient clearing oracle and a bank v, it is NP-hard to compute an $n^{1/2-\varepsilon}$-approximation to (1) the maximal total assets of v and (2) the maximal sum of total assets that can be obtained by a sequence of semi-positive debt swaps, for every constant $\varepsilon > 0$.*

The next theorem shows that the problem remains NP-hard even under the restriction that v is involved in every swap and every semi-positive swap is also v-improving.

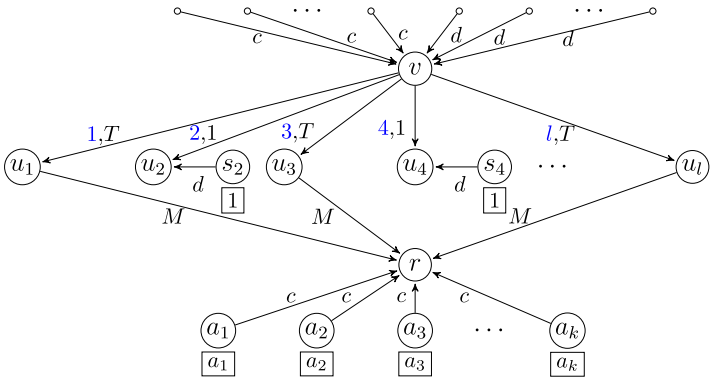


Fig. 11. Schematic construction for the reduction from 3-PARTITION where blue edge labels indicate edge-ranking rules and black labels indicate edge weights. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Theorem 8. *For a given financial network with monotone allocation rules, an efficient clearing oracle, and a bank v, it is NP-hard to compute a sequence of v-improving semi-positive debt swaps with creditor v such that (1) the total assets of v are maximal and (2) the sum of total assets is maximal.*

Proof. For statement (1), consider an instance of PARTITION with items $i \in [k]$ and integer values $a_i \in \mathbb{N}$. Moreover, denote the sum of values by A, i.e., $A = \sum_{i\in[k]} a_i$. The goal is to divide the items into two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ such that $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$, $\mathcal{A}_1 \cup \mathcal{A}_2 = [k]$ and $\sum_{i\in\mathcal{A}_1} a_i = \sum_{i\in\mathcal{A}_2} a_i$.

For our reduction, we define a financial network with banks $v, u, a_1, a_2, \dots, a_k$ and k auxiliary banks $s_1, s_2, \dots, s_k$. For every $i \in [k]$ there are edges (a_i, u) and (s_i, v) with weights $M > \max_{j\in[k]} a_j$. There is a single edge (v, u) with weight $A/2$. Moreover, bank a_i has external assets a_i, for every $i \in [k]$. Clearly, the construction can be computed efficiently in the size of the instance of PARTITION.

Every bank has at most one outgoing edge and thus pays all assets towards that edge until the debt is settled. The clearing state is the same for every monotone allocation rule and can be computed efficiently.

We show that there the instance of PARTITION has a solution if and only if there is a sequence of v-improving semi-positive debt swaps with creditor v such that the total assets of v are $A/2$.

First, assume there exists a partition in two sets $\mathcal{A}_1, \mathcal{A}_2 \subset [k]$ such that $\mathcal{A}_1 \cup \mathcal{A}_2 = [k]$, $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ and $\sum_{i\in\mathcal{A}_1} a_i = A/2 = \sum_{i\in\mathcal{A}_2} a_i$. Hence, when performing debt swaps of edges $(a_i, u), (s_i, v)$, for $i \in \mathcal{A}_1$, then v has total assets of $A/2$. By construction, all assets of v are payed to u. Thus, all swaps are semi-positive.

For the other direction, assume there exists a sequence of v-improving semi-positive debt swaps with creditor v such that the total assets of v are given by $A/2$. Then clearly, there exists a set of banks $\mathcal{A} \in [k]$ where total assets add up to $A/2$. There exists a partition $\mathcal{A}, \bar{\mathcal{A}} = \{i \notin \mathcal{A} \mid i \in [k]\}$ for the given input.

Statement (2) follows when including B additional banks w_b on the path from v to u with edge weight $A/2$ for edge (v, w_1) and $B + b$ for every edge $(w_b, \cdot)$ with $b \in [B]$. This construction does not allow any additional debt swaps. Since the total assets of every bank w_b equal those of v, maximizing the sum of total assets reduces to maximizing assets of v for sufficiently large B. □

For financial networks with edge-ranking rules we show a slightly stronger result.

Theorem 9. *For a given financial network with edge-ranking rules and a bank v, it is strongly NP-hard to compute a sequence of v-improving semi-positive debt swaps with creditor v such that (1) the total assets of v are maximal and (2) the sum of total assets is maximal.*

Proof. We show statement (1) by reduction from 3-PARTITION. An instance of 3-PARTITION consists of a set of k elements such that k is an integer multiple by 3. Each item $i \in [k]$ has a value $a_i \in \mathbb{N}$. We denote the sum of values by $A = \sum_{i\in[k]} a_i$ and let $T = \frac{3}{k}A$. For each value it holds that $\frac{T}{4} < a_i < \frac{T}{2}$. The goal is to decide whether or not the given set of k items can be partitioned into k sets $S_1, \dots, S_k$ with $\sum_{i\in S_j} a_i = T$, for each $j \in [k]$. Due to the value restriction, each set must have exactly three items.

For an instance of 3-PARTITION, we construct a financial network as follows. First, add a root r and a bank a_i with external assets a_i, for every item i. Additionally, every bank a_i has a single outgoing edge (a_i, r) with edge weight $c \geq \frac{T}{2}$. Construct bank v and $u_1, u_2, \dots, u_l$, for $l = 2\frac{k}{3} - 1$, together with edges (v, u_h). For all odd numbers h, set the weight of (v, u_h) to T and add the edge (u_h, r) with weight $M > T$. Instead, set the weight of (v, u_h) to 1, for all even numbers h, and include a source s_h with external assets of 1 and edge (s_h, u_h) and weight $d > 1$. Finally, add k auxiliary banks, each with a single outgoing edge to v and weight c, and $\frac{l}{2}$ auxiliary banks with outgoing edge to v and weight $d > 1$. The values T, M, d and c are chosen to be distinct. v is the only bank with more than one outgoing edge, and it distributes all assets according to an edge-ranking rule with $\pi_v(h) = u_h$ for all h. That means, first v fully pays off debt to u_1, then to u_2,..., until v runs out of funds. A schematic construction can be seen in Fig. 11.

Note that every bank has at most one outgoing edge. As such, the clearing state is independent of the allocation rule and can be determined easily.

We show that the 3-PARTITION instance has a solution if and only if there is a sequence of v-improving semi-positive debt swaps resulting in total assets of at least $A + \frac{l}{2} - 1$ for v.

First, assume there exists a solution to the 3-PARTITION instance. Then, swap all edges incident to banks $a_i \in S_1$ with incident edges of v. This results in total assets of T for v and thus she can exactly pay off (v, u_1). Clearly, the described swaps are all semi-positive and v is the benefiting creditor. Now, execute a swap with creditors v and u_2. Since, this results in additional assets of 1 in v and the second edge has payments zero and is active, the swap is semi-positive. When repeatedly applying this process for the remaining sets S_j, bank v has swapped all incoming edges in the initial network. Consequently, the total assets of v are given by $A + \frac{l}{2} - 1$.

For the other direction, assume there exists a sequence of v-improving semi-positive debt swaps resulting in total assets of at least $A + \frac{l}{2} - 1$ for v. Because the graph is cycle-free, all assets must originate from external assets. In particular, observe that the sum of external assets in the network is given by $A + \frac{l}{2} - 1$ and hence all banks with external assets must pay all assets to v. Clearly, this is only the case if all banks with external assets are swapped to be adjacent to v. In the initial network, due to the allocation rule v can only perform semi-positive swaps with r. Moreover, v and r can switch edges as long as the residual capacity of (v, u_1) is sufficiently large. Only when the total assets of v are exactly T, can v fully pay off the first edge towards u_1 and execute a semi-positive swap with u_2. Otherwise, a fraction of the additional assets would either be payed to u_1 or u_3. These arguments can be repeated for all remaining outgoing edges of v. As a result, v and r can only swap all their initial incoming edges if the external assets a_i can be grouped into sets with sum of value exactly T. Since $\frac{T}{4} < a_i < \frac{T}{2}$, every set has cardinality three.

Statement (2) follows by the same construction as for (1). We claim that there exists a solution to the given instance of 3-PARTITION if and only if there exists a sequence of v-improving semi-positive debt swaps resulting in at least $4A + 3(\frac{l}{2} - 1)$ sum of total assets.

If there exists a solution, then proceed as for (1). Clearly, the total assets of all sources add up to $A + \frac{l}{2} - 1$ and the total assets of v are given by same amount. Additionally, v pays all assets to banks u_h whereas all payments of A are forwarded to r. In total, assets of $4A + 3(\frac{l}{2} - 1)$ are achieved.

Suppose there exists a sequence of semi-positive swaps resulting in a sum of total assets of at least $4A + 3(\frac{l}{2} - 1)$. Clearly, this value can only be achieved when v settles all debt. By the same argument as before, this directly implies existence of a solution to 3-PARTITION. □

Observe that our constructions in the proofs of Theorems 6, 7, 8 and 9 lead to classes of cycle-free financial networks. Therefore, all NP-hardness results in this section apply even under the restriction that the underlying graph structure of the financial network is a DAG.

6. Reachability

A debt swap changes the underlying graph structure of the corresponding financial network. It is a natural reachability question whether, for two given networks, one network can be transformed into the other via a sequence of debt swaps.

More formally, in the REACH problem we are given an initial network $\mathcal{F}$ and a target network $\mathcal{G}$. Both networks have the same set V of nodes and are consistent with the stubs from an instance of MAXASSETS$_v$. We say every node $v \in V$ has the same *incidence profile* in both networks (i.e., for each value l_i the same number of outgoing edges with weight l_i in both $\mathcal{F}$ and $\mathcal{G}$ and the same number of incoming edges with l_i in both $\mathcal{F}$ and $\mathcal{G}$). Moreover, $\mathcal{F}$ and $\mathcal{G}$ have the same monotone allocation rule, e.g., v uses the same edge ranking to allocate assets to outgoing stubs, for each $v \in V$. Overall, we say the networks $\mathcal{F}$ and $\mathcal{G}$ are *consistent with each other*. If $\mathcal{F}$ and $\mathcal{G}$ are not consistent with each other, then the answer to every reachability problem is trivially negative. Hence, we assume that $\mathcal{F}$ and $\mathcal{G}$ are consistent with each other. To avoid trivialities, $\mathcal{F}$ and $\mathcal{G}$ have different edge sets.

For a finite sequence of debt swaps $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots)$ to $\mathcal{F}$, we first execute swap σ_1 in $\mathcal{F}$ and denote the resulting network by $\mathcal{F}^{\sigma_1}$. We again assume to have access to an efficient clearing oracle $\mathcal{O}$ to compute the clearing state. Then perform σ_2 in $\mathcal{F}^{\sigma_1}$, then σ_3 in $\mathcal{F}^{\sigma_2}$, etc. The process ends after all swaps in the sequence were applied. The resulting network is denoted by $\mathcal{F}^{\boldsymbol{\sigma}}$. Every financial network created during this sequence is called an *intermediate* network. We say that $\mathcal{G}$ is *reachable* from $\mathcal{F}$ if there exists a sequence of debt swaps $\boldsymbol{\sigma}$ such that $\mathcal{F}^{\boldsymbol{\sigma}} = \mathcal{G}$. The sequence $\boldsymbol{\sigma}$ is called a *reaching* sequence. In the REACH problem, we are interested in reaching sequences of *arbitrary* swaps. If we only consider sequences of v-improving arbitrary debt swaps, then the problem is denoted by REACH(v). Intuitively, instead of strict profits for a given bank v it might be sufficient that v stays solvent. Formally, in the REACH(v, ℓ) problem we are given consistent networks $\mathcal{F}$ and $\mathcal{G}$ such that v has the *same* amount ℓ of total assets in both networks. We shall decide if there exists a reaching sequence of debt swaps such that the total assets of v are at least ℓ in every intermediate network.

These problems can be interpreted as instances of the $s-t$-Connectivity (STCON) problem in a transition graph. The nodes of this transition graph are the networks consistent with $\mathcal{F}$ and $\mathcal{G}$. There is a directed edge between networks $\mathcal{F}'$ and $\mathcal{F}''$ if and only if there is a debt swap that turns $\mathcal{F}'$ into $\mathcal{F}''$. STCON is solvable in poly-logarithmic space [28]. In our case, there are less than $(m!)^{n/(2m)} < (\frac{m}{2})^m$ consistent networks [29], where the inequality holds for large m. As such, all our reachability problems lie in PSPACE.

Our first result shows that for arbitrary swaps, the problem is much easier. The consistency property is necessary *and sufficient* for reachability. A reaching sequence of polynomially many swaps can be computed in polynomial time if it exists.

Theorem 10. *The REACH problem with monotone allocation rules can be solved in time polynomial in the input size.*

Proof. Let D be the set of desired edges that are in $\mathcal{G}$ and not in $\mathcal{F}$, i.e., $D = E^{\mathcal{G}} \setminus E^{\mathcal{F}}$. There are exactly $|D| > 0$ edges in $E^{\mathcal{F}} \setminus E^{\mathcal{G}}$. First, observe that $|D| > 1$. To see this, assume for a contradiction that $D = \{e\}$ for $\mathsf{de}(e) = u$, $\mathsf{cr}(e) = v$ and $u, v \in V$ with weight $l_e > 0$. Since $\mathcal{F}$ and $\mathcal{G}$ are consistent with each other, in $\mathcal{F}$ bank v must have an incoming edge e' with $\mathsf{de}(e') = x$, and u must have an outgoing

edge e'' with $\mathsf{cr}(e'') = y$ each with weight l_e, and both these edges are not in $E^{\mathcal{G}}$. Since $|D| = 1$, banks x and y have more incident edges in $\mathcal{F}$ than $\mathcal{G}$, a contradiction. We assume $|D| \geq 2$ for the remaining proof.

We construct a sequence of debt swaps that in each step replaces an edge in $E^{\mathcal{F}} \setminus E^{\mathcal{G}}$ by one in D. Choose an arbitrary edge $\hat{e} \in D$ with $\mathsf{de}(\hat{e}) = w, \mathsf{cr}(\hat{e}) = v$. By consistency there exists an edge $e \in E^{\mathcal{F}} \setminus E^{\mathcal{G}}$, where $\mathsf{de}(e) = u, \mathsf{cr}(e) = v$, with weight $l_e = l_{\hat{e}}$. Further, w has at least one outgoing edge e' with $\mathsf{cr}(e') = x$ in $\mathcal{F}$ that is not in $\mathcal{G}$ with weight $l_{e'} = l_e$. If the banks u, v, w and x are pair-wise distinct, choose another arbitrary edge in D. Clearly, there always exists such an edge if $\mathcal{G}$ is reachable from $\mathcal{F}$. Otherwise, perform a swap σ of edges e and $\hat{e}$. The resulting network $\mathcal{F}^{\sigma}$ contains the desired edge $\hat{e} \in D$ connecting w and v and one edge in $E^{\mathcal{F}} \setminus E^{\mathcal{G}}$ was removed. We repeat this step until two desired edges in D have not been included.

For the final step with $|D| = 2$, observe that there are exactly two edges in $E^{\mathcal{F}} \setminus E^{\mathcal{G}}$. Choose an arbitrary edge $\hat{e} \in D$ with $\mathsf{de}(\hat{e}) = w, \mathsf{cr}(\hat{e}) = v$. By the same argument as before, there exist edges $e, e' \in E^{\mathcal{F}} \setminus E^{\mathcal{G}}$ where $\mathsf{de}(e) = u, \mathsf{cr}(e) = v$ and $\mathsf{de}(e') = w, \mathsf{cr}(e') = x$ each with the same weight as $\hat{e}$. By consistency, bank u must have an outgoing edge with weight $l_{\hat{e}}$ that is not in $\mathcal{F}$. x must have an incoming edge with weight $l_{\hat{e}}$ that is not in $\mathcal{F}$. Since $|D| = 2$, this implies the existence of edge from u to x in $\mathcal{G}$. Then, $\mathcal{G}$ is reachable from $\mathcal{F}$ with the debt swap of edges e and e' creating the edges in D.

This proves that the network $\mathcal{G}$ can indeed be reached via a sequence of debt swaps. In every step of this sequence, a debt swap of edges not in $\mathcal{G}$ is performed to create a new desired edge in D. This means that a desired edge created in a previous step will never be swapped again. Thus, the number of swaps in the resulting sequence is bounded by $O(m)$ where $m = |E^{\mathcal{F}}| = |E^{\mathcal{G}}|$ and $m \leq |D|$. □

The greedy algorithm described in the proof of Theorem 10 arbitrarily chooses an edge from the target network and performs a debt swap to include the edge into the current network. Note that this can lead to devastating losses of assets of a single bank v in the intermediate networks, which is not aligned with incentives and potentially problematic for the financial stability of the system. Instead, for v-improving swaps, in which a given bank v must strictly profit in every step, the tight PLS-reduction in Theorem 5 directly implies the following result [24, Theorem 16].

Corollary 9. *The* REACH(v) *problem with monotone allocation rules is PSPACE-complete.*

We strengthen this result in the following way. Instead of guaranteeing a strict profit for v in every step, we require only that the total assets of v are at least ℓ in the initial, final, and every intermediate network. Deciding whether there exists a reaching sequence of arbitrary debt swaps so that v *never has strictly less assets* in any step is PSPACE-hard (and, hence, PSPACE-complete).

Theorem 11. *The* REACH(v, ℓ) *problem with monotone allocation rules is PSPACE-complete.*

Proof. An instance of 3-SAT CONNECTIVITY consists of a CNF formula φ with k variables x_i and ℓ clauses where every clause κ_j consists of at most three variables. Every satisfying assignment of φ is represented by a node in the graph $G(\varphi)$ where two assignments are adjacent if the assignments differ for exactly one variable. For two given assignments, the objective is to decide whether there exists a path connecting the corresponding nodes in $G(\varphi)$. Equivalently, this requires to decide whether there is a sequence of flips between these satisfying assignments changing the truth assignment of exactly one variable in each step. The authors in [30] show PSPACE-completeness for 3-SAT CONNECTIVITY.

For a reduction, use a slight adaptation of the construction illustrated in Fig. 8 described in proof of Theorem 5 where all values w_j are set to one. Also, choose edges (x_i, x_i^T) and (s_i, x_i^F) if variable x_i is TRUE in the initial assignment, otherwise choose (x_i, x_i^F) and (s_i, x_i^T). Observe that every debt swap switches the incoming edges of a pair of banks x_i^T and x_i^F where the bank with incoming payments from x_i is interpreted as TRUE. As a result, every sequence of flips in the assignment translates directly to a sequence of debt swaps and vice versa.

The total assets of bank v equal the number of banks κ_j with incoming assets. Hence, if v has assets of at least l then every κ_j has incoming payments and the associated assignment is satisfying. As an implication, there exists a sequence of flips in the assignment, where every intermediate assignment is satisfying, if and only if there exists a sequence of debt swaps turning the initial into the target network, where v has assets of ℓ in every intermediate network. This concludes the proof. □

Finally, we briefly consider sequences of semi-positive debt swaps. Indeed, since they are Pareto-improving, the motivation here is similar as in the previous theorem. Deciding whether a reaching sequence exists where no bank is harmed is already challenging, even for edge-ranking rules.

Proposition 7. *Given initial network $\mathcal{F}$ and a target network $\mathcal{G}$ with edge-ranking rules that are consistent with each other, it is strongly NP-hard to decide whether $\mathcal{G}$ is reachable from $\mathcal{F}$ via any sequence of semi-positive swaps.*

Proof. The statement follows by reduction from 3-PARTITION. As initial network we choose the construction in proof of Theorem 9 (see Fig. 11). The target is the network where all incoming edges with weight c of v and r are switched. By the same arguments as in proof of Theorem 9 there exists a desired sequence of semi-positive debt swaps if and only if there exists a solution to the instance of 3-PARTITION. □

7. Conclusion

In this paper we have studied the computational complexity of problems surrounding the debt swap operation, a natural and interesting edge swap operation in financial networks. Our results characterize the convergence time of natural debt swap dynamics for edge-ranking payment rules. Moreover, our work sheds light on the complexity of optimization and reachability properties of debt swap dynamics.

Our work gives rise to a plethora of interesting future directions. Most prominently, the general SemiSwap problem remains a challenging open problem. Also, it would be interesting to identify additional meaningful cases in which reachability or optimization problems can be solved (approximately) optimally in polynomial time. More generally, debt swaps can lead to interesting subtle structural effects. For example, a sequence of debt swaps from a network $\mathcal{F}$ with edge-ranking rules can result in a financial network $\mathcal{G}$ with the same underlying graph but a different edge-ranking rule. Understanding the structural, algorithmic, and economic consequences of these effects represent fascinating open problems.

CRediT authorship contribution statement

Henri Froese: Writing - original draft; **Martin Hoefer:** Writing - original draft; **Lisa Wilhelmi:** Writing - original draft.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

Lisa Wilhelmi reports financial support was provided by Deutsche Forschungsgemeinschaft. Martin Hoefer reports financial support was provided by Deutsche Forschungsgemeinschaft. Martin Hoefer reports financial support was provided by ATHENE National Research Center for Applied Cybersecurity. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Martin Hoefer gratefully acknowledges support from DFG Research Unit ADYN under grant DFG 411362735 and ATHENE National Research Center for Applied Cybersecurity. Lisa Wilhelmi gratefully acknowledges support from DFG Research Unit ADYN under grant DFG 411362735.

Appendix A. Missing Proofs in Section 3

Definition 7. Consider a financial network $\mathcal{F}$ with external assets $\mathbf{a}^x$ and a non-negative vector $\hat{\mathbf{a}}^x \leq \mathbf{a}^x$, where $\hat{a}^x_v < a^x_v$ for at least one $v \in V$. The network $\mathcal{F}$ can be separated into two networks: $\mathcal{F}^{\text{pre}} = (G, \mathsf{de}, \mathsf{cr}, \mathbf{l}, \hat{\mathbf{a}}^x, \mathbf{f})$ is called a *pre-network* of $\mathcal{F}$; $\mathcal{F}^{\text{diff}} = (G, \mathsf{de}, \mathsf{cr}, \mathbf{l}^{\text{diff}}, \mathbf{a}^x - \hat{\mathbf{a}}^x, \mathbf{f}^{\text{diff}})$ is the *difference network* of $\mathcal{F}$ and $\mathcal{F}^{\text{pre}}$. The liabilities in $\mathcal{F}^{\text{diff}}$ are given by the remaining liabilities after clearing in $\mathcal{F}^{\text{pre}}$, (i.e., $l^{\text{diff}}_e = l_e - p^{\text{pre}}_e$), and the allocation rule is defined as $f^{\text{diff}}_e(b_{\mathsf{de}(e)}) = f_e(a^{\text{pre}}_{\mathsf{de}(e)} + b_{\mathsf{de}(e)}) - f_e(a^{\text{pre}}_{\mathsf{de}(e)})$.

The clearing state in $\mathcal{F}$ equals the sum of clearing states in pre- and difference networks.

Lemma 8 (Separability)**.** *Suppose $\mathcal{F}^{pre}$ is a pre-network of $\mathcal{F}$ and $\mathcal{F}^{diff}$ the corresponding difference network. The clearing state of $\mathcal{F}$ is given by the sum of clearing states in $\mathcal{F}^{pre}$ and $\mathcal{F}^{diff}$, i.e., $\mathbf{p} = \mathbf{p}^{pre} + \mathbf{p}^{diff}$.*

Proof. As mentioned in Section 2, the feasible asset vectors for a financial network $\mathcal{F}$ are solutions of a fixed point problem. We reformulate this property for the feasible payments – they are fixed points of a function g defined for every $e \in E$ by $g_e(\mathbf{c}) = f_e(a^x_u + \sum_{e' \in E^-(u)} c_{e'})$, where $u = \mathsf{de}(e)$. g maps the convex, compact set $\{\mathbf{c} \mid 0 \leq c_e \leq l_e\}$ to itself. The fixed points of $g^{\text{pre}}_e(\mathbf{c}) = f_e(\hat{a}^x_u + \sum_{e' \in E^-(u)} c_{e'})$ are the feasible payments in $\mathcal{F}^{\text{pre}}$, the fixed points of $g^{\text{diff}}_e(\mathbf{c}) = f^{\text{diff}}_e(a^x_u - \hat{a}^x_u + \sum_{e' \in E^-(u)} c_{e'})$ the feasible payments in $\mathcal{F}^{\text{diff}}$.

For the maximal fixed point $\mathbf{p}^{\text{diff}}$ of g^{diff} it holds that

$$\begin{aligned}
g^{\text{diff}}_e(\mathbf{p}) &= f^{\text{diff}}_e\left(a^x_u - \hat{a}^x_u + \sum_{e' \in E^-(u)} p_{e'}\right) \\
&= f_e\left(a^{\text{pre}}_u + a^x_u - \hat{a}^x_u + \sum_{e' \in E^-(u)} p_{e'}\right) - f_e\left(a^{\text{pre}}_u\right) \\
&= f_e\left(\hat{a}^x_u + \sum_{e' \in E^-(u)} p^{\text{pre}}_{e'} + a^x_u - \hat{a}^x_u + \sum_{e' \in E^-(u)} p_{e'}\right) - f_e\left(\hat{a}^x_u + \sum_{e' \in E^-(u)} p^{\text{pre}}_{e'}\right) \\
&= f_e\left(a^x_u + \sum_{e' \in E^-(u)} (p_{e'} + p^{\text{pre}}_{e'})\right) - p^{\text{pre}}_e \quad .
\end{aligned}$$

The second equality follows from the definition of f^{diff}_e. The last equality follows since $\mathbf{p}^{\text{pre}}$ is the maximal fixed point of g^{pre}.

The clearing state of $\mathcal{F}$ satisfies $\mathbf{p} \geq \mathbf{p}^{\text{pre}}$ due to monotonicity of all f_e. Indeed, for *every* fixed point $\mathbf{c}$ of g it holds that $\mathbf{c} \geq \mathbf{c}^{\text{pre}}$, since $a^x_v > \hat{a}^x_v$ for at least one v. Thus, every fixed point $\mathbf{c}$ can be separated into $\mathbf{p}^{\text{pre}}$ and $\bar{\mathbf{c}}$ such that $\mathbf{c} = \mathbf{p}^{\text{pre}} + \bar{\mathbf{c}}$. Now consider $\bar{\mathbf{c}}$. Since

$\mathbf{c} = \mathbf{p}^{\text{pre}} + \bar{\mathbf{c}}$ is a fixed point, we have $p_e^{\text{pre}} + \bar{c}_e = f_e(a_u^x + \sum_{e' \in E^-(u)}(p_{e'}^{\text{pre}} + \bar{c}_{e'}))$ for all $e \in E$ and $u = \mathsf{de}(e)$. Equivalently,

$$p_e^{\text{pre}} = f_e\left(a_u^x + \sum_{e \in E^-(u)} (p_e^{\text{pre}} + \bar{c}_e)\right) - \bar{c}_e$$

and, thus, $\bar{\mathbf{c}}$ is a fixed point of g^{diff}. Hence, every fixed point of g can be separated into $\mathbf{p}^{\text{pre}}$ and a fixed point for g^{diff}. Since the clearing state $\mathbf{p}$ is the *maximal* fixed point of g, it must use the *maximal* fixed point of g^{diff}, i.e., $\mathbf{p} = \mathbf{p}^{\text{pre}} + \mathbf{p}^{\text{diff}}$. □

Note that if edge $e \in E$ gets saturated in $\mathcal{F}^{\text{pre}}$ (i.e., $l_e = p_e^{\text{pre}}$), then $l_e^{\text{diff}} = 0$ in $\mathcal{F}^{\text{diff}}$.

Corollary 10. *If $\mathcal{F}^{pre}$ is a pre-network of $\mathcal{F}$, then for the clearing payments $p_e \geq p_e^{pre}$, for all $e \in E$, and the total assets $a_v \geq a_v^{pre}$, for every bank $v \in V$.*

The next definition establishes a relationship between the external assets of a node v and the incoming assets of a sink.

Definition 8 (Linearity)**.** If an increase of the external assets a_v^x of a node $v \in V$ by $\Delta > 0$ results in total assets $a_t' = a_t + \Delta$ for a sink $t \in V$, then v is called *t-linear* on $[a_v^x, a_v^x + \Delta]$. If the increase leads to $\sum_{t \in U} a_t' = \sum_{t \in U} a_t + \Delta$ for a set U of sink nodes, then v is called *U-linear* on $[a_v^x, a_v^x + \Delta]$.

If v is t-linear, then the benefit of t due to the increase of external assets of v is maximal.

Lemma 9 (Monotonicity)**.** *Suppose two financial networks $\mathcal{F}$ and $\hat{\mathcal{F}}$ differ only in the external assets of a single bank v such that $a_v^x > \hat{a}_v^x$. For every bank $u \in V$, the total assets in the clearing state satisfy $a_u \geq \hat{a}_u$.*

Proof. The network $\mathcal{F}$ can be seperated into the pre-network $\mathcal{F}^{\text{pre}} = \hat{\mathcal{F}}$ and difference network $\mathcal{F}^{\text{diff}}$. The external assets in $\mathcal{F}^{\text{diff}}$ are given by $a_v^x - \hat{a}_v^x$ for v and 0 for all other banks $u \neq v, u \in V$. By separability, the clearing state of $\mathcal{F}$ is the sum of clearing states of $\mathcal{F}^{\text{pre}}$ and $\mathcal{F}^{\text{diff}}$, i.e., $\mathbf{p} = \mathbf{p}^{\text{pre}} + \mathbf{p}^{\text{diff}}$. Since the total assets of every bank are non-negative, we have $a_u = a_u^{\text{pre}} + a_u^{\text{diff}} \geq a_u^{\text{pre}} = \hat{a}_u$. □

Lemma 10 (Non-Expansivity)**.** *lemmalemNonExpan For a given financial network, let $T \subset V$ be the set of all sink nodes. If the external assets of a source $s \in V \setminus T$ are increased to $a_s'^x = a_s^x + \Delta$, then the total assets of all sinks increase by at most Δ, i.e., $\sum_{t \in T} a_t' \leq \Delta + \sum_{t \in T} a_t$.*

Proof. By slight abuse of notation, we use $\mathcal{F}$ to denote the resulting financial network *after* raising the external assets of s to $a_s^x + \Delta$. By separability, we define $\mathcal{F}^{\text{pre}}$ as the pre-network with external assets a_u^x for every bank $u \in V$, i.e., the one *before* raising the external assets of s. Then, the external assets in the difference network $\mathcal{F}^{\text{diff}}$ are given by Δ for s and 0 for all other banks $u \neq s$. The statement follows when $\sum_{t \in T} a_t^{\text{diff}} \leq \Delta$.

We use p_u^- and p_u^+ to denote the incoming and outgoing assets of u in the clearing state of $\mathcal{F}^{\text{diff}}$, i.e., $p_u^- = \sum_{e \in E^-(u)} p_e^{\text{diff}}$ and $p_u^+ = \sum_{e \in E^+(u)} p_e^{\text{diff}}$. Then, $\sum_{u \in V} p_u^- = \sum_{u \in V} p_u^+$, as both are the sum of all payments in the network. Dividing the banks into sinks, source s and all other banks, it holds that

$$\sum_{u \in V \setminus (T \cup \{s\})} p_u^- + \sum_{t \in T} p_t^- + p_s^- = \sum_{u \in V \setminus (T \cup \{s\})} p_u^+ + \sum_{t \in T} p_t^+ + p_s^+ \,. \tag{A.1}$$

Since s is a source and the banks in T are sinks, $p_s^- = 0$ and $\sum_{t \in T} p_t^+ = 0$. Eq. (A.1) simplifies to

$$\sum_{t \in T} p_t^- = \sum_{u \in V \setminus (T \cup \{s\})} (p_u^+ - p_u^-) + p_s^+ \,.$$

Every bank $u \in V \setminus (T \cup \{s\})$ has external assets of 0 and, thus, the outgoing payments of u are upper bounded by the incoming ones, i.e., $p_u^+ \leq p_u^-$. As a direct consequence is $\sum_{t \in T} p_t^- \leq p_s^+ \leq \Delta$. Hence, $\sum_{t \in T} a_t^{\text{diff}} = \sum_{t \in T} p_t^- \leq \Delta$. □

Lemma 11 (Source-Sink-Equivalency)**.** *For a given financial network $\mathcal{F}$ and an adjusted network $\mathcal{H}$ it holds that*

(1) the networks have equivalent clearing states, i.e., $p_e^{\mathcal{H}} = p_e$ for all $e \in E$.
(2) the sink t_v has total assets $a_v - a_v^x$.

Proof. First, observe that (2) follows directly from (1). Hence, it remains to show property (1). As in the proof of Lemma 8, we use that the clearing state of $\mathcal{F}$ is the maximal fixed point of g with $g_e(\mathbf{c}) = f_e(a_u^x + \sum_{e' \in E^-(u)} c_{e'})$, and the clearing state of $\mathcal{H}$ is the maximal fixed point of $g^{\mathcal{H}}$ with $g_e^{\mathcal{H}}(\mathbf{c}^{\mathcal{H}}) = f_e^{\mathcal{H}}(a_u^x + \sum_{e' \in E_{\mathcal{H}}^-(u)} c_{e'}^{\mathcal{H}})$, where $u = \mathsf{de}(e)$.

First, suppose $\mathbf{p}$ is the maximal clearing state of $\mathcal{F}$. Now consider $\mathbf{p}^{\mathcal{H}}$ as in property (1) of the lemma. Then for every $e \in E$ with $\mathsf{de}(e) \neq v \neq \mathsf{cr}(e)$, clearly $g_e(\mathbf{p}) = g_e^{\mathcal{H}}(\mathbf{p}^{\mathcal{H}})$ is true. Similarly for incoming edges of $e \in E^-(v)$ with $u = \mathsf{de}(e)$,

$$g_e(\mathbf{p}) = f_e\left(a_u^x + \sum_{e' \in E^-(u)} p_{e'}\right) = f_e^{\mathcal{H}}\left(a_u^x + \sum_{e' \in E_{\mathcal{H}}^-(u)} p_{e'}^{\mathcal{H}}\right) = g_e(\mathbf{p}^{\mathcal{H}}).$$

Now consider the outgoing edges $e \in E^+(v)$. It holds that

$$p_e = g_e(\mathbf{p}) = f_e\left(a_v^x + \sum_{e' \in E^-(v)} p_{e'}\right) = f_e(a_v) = f_e^{\mathcal{H}}(a_{s_v}) = g_e(\mathbf{p}^{\mathcal{H}}) \,.$$

Hence, $\mathbf{p}^{\mathcal{H}}$ is a fixed point for the network $\mathcal{H}$ and evaluates all payments as in the maximal fixed point $\mathbf{p}$ of g.

Now, for the converse, suppose that $\mathbf{p}^{\mathcal{H}}$ is the clearing state for $\mathcal{H}$. Then, by the same equations, we obtain an equivalent fixed point of g. In conclusion, this proves (1). □

The next lemma introduces a useful connection between $\mathcal{H}$ and $\mathcal{H}^\sigma$.

Lemma 12. *Consider the financial networks $\mathcal{F}$ before and $\mathcal{F}^\sigma$ after a debt swap. If $a^\sigma_{v_1} \geq a_{v_1}$ and $a^\sigma_{v_2} \geq a_{v_2}$, then the following hold:*

(1) $\mathcal{H}$ and $\mathcal{H}^\sigma$ only differ in the external assets at s_{v_1} and s_{v_2} (and thus the clearing state).
(2) $\mathcal{H}$ is a pre-network of $\mathcal{H}^\sigma$.
(3) s_{v_1} and s_{v_2} are $\{t_{u_1}, t_{u_2}, t_{v_1}, t_{v_2}\}$-linear in $\mathcal{H}$ on intervals $[a_{v_1}, a^\sigma_{v_1}]$ and $[a_{v_2}, a^\sigma_{v_2}]$, resp.

Proof. By definition, the underlying graphs of $\mathcal{F}$ and $\mathcal{F}^\sigma$ only differ in the liabilities of u_1 and u_2 to v_1 and v_2. When applying source-sink equivalency to these edges, all other edges remain the same. The resulting networks $\mathcal{G}$ and $\mathcal{G}^\sigma$ only differ in external assets of v_1 and v_2. This still holds when applying source-sink equivalency to v_1 and v_2 to $\mathcal{G}$ and $\mathcal{G}^\sigma$. Hence, (1) is satisfied. By definition, (1) directly implies (2).

Towards (3), consider the pre-network $\mathcal{H}$. Suppose the external assets at the sources s_{v_1} and s_{v_2} are increased by $a^\sigma_{v_1} - a_{v_1}$ and $a^\sigma_{v_2} - a_{v_2}$, respectively. In total, the external assets of both banks are raised by $(a^\sigma_{v_1} + a^\sigma_{v_2}) - (a_{v_1} + a_{v_2})$.

By (1), every bank now has the same external assets and liabilities as in $\mathcal{H}^\sigma$ and, thus, the clearing state is $\mathbf{p}^\sigma$. For $\mathcal{H}^\sigma$ we use the notation $e^\sigma_1 = (u_1, t_{u_2})$ and $e^\sigma_2 = (u_2, t_{u_1})$ for the swapped edges. Then, the total assets of the sinks t_{u_1}, t_{u_2}, t_{v_1}, and t_{v_2} increase by a total amount of

$$\begin{aligned}
&(a^\sigma_{t_{u_1}} + a^\sigma_{t_{u_2}} + a^\sigma_{t_{v_1}} + a^\sigma_{t_{v_2}}) - (a_{t_{u_1}} + a_{t_{u_2}} + a_{t_{v_1}} + a_{t_{v_2}}) \\
&= \left(p^\sigma_{e^\sigma_2} + p^\sigma_{e^\sigma_1} + (a^\sigma_{v_1} - a^x_{v_1} - p^\sigma_{e^\sigma_2}) + (a^\sigma_{v_2} - a^x_{v_2} - p^\sigma_{e^\sigma_1})\right) \\
&\qquad - \left(p_{e_1} + p_{e_2} + (a_{v_1} - a^x_{v_1} - p_{e_1}) + (a_{v_2} - a^x_{v_2} - p_{e_2})\right) \\
&= (a^\sigma_{v_1} + a^\sigma_{v_2}) - (a_{v_1} + a_{v_2}).
\end{aligned}$$

Consequently, sources s_{v_1} and s_{v_2} are $\{t_{u_1}, t_{u_2}, t_{v_1}, t_{v_2}\}$-linear on $[a_{v_1}, a^\sigma_{v_1}]$ and $[a_{v_2}, a^\sigma_{v_2}]$, respectively. □

Lemma 13. *For $\mathcal{H}_\Delta$ and $\mathcal{F}_\Delta$ the following are true:*

(1) $\delta_{s_{v_1}} + \delta_{s_{v_2}} = \Delta_{t_{v_1}} + \Delta_{t_{v_2}} + \Delta_{t_{u_1}} + \Delta_{t_{u_2}}$.
(2) If $\delta_{s_{v_1}} > 0$ or $\delta_{s_{v_2}} > 0$, then $\delta_{s_{v_1}} \neq \Delta_{t_{v_1}} + \Delta_{t_{u_1}}$ and $\delta_{s_{v_2}} \neq \Delta_{t_{v_2}} + \Delta_{t_{u_2}}$.

Proof. (1) directly follows from $\{t_{u_1}, t_{u_2}, t_{v_1}, t_{v_2}\}$-linearity of s_{v_1} and s_{v_2} on intervals $[a_{v_1}, a^\sigma_{v_1}]$ and $[a_{v_2}, a^\sigma_{v_2}]$ (Lemma 12). For (2), assume that $\delta_{s_{v_1}} > 0$ or $\delta_{s_{v_2}} > 0$ and $\delta_{s_{v_1}} = \Delta_{t_{v_1}} + \Delta_{t_{u_1}}$. Then, by (1) it follows that $\delta_{s_{v_2}} = \Delta_{t_{v_2}} + \Delta_{t_{u_2}}$.

Intuitively, the additional external assets of v_1 are forwarded through the network and, eventually, are payed back to v_1. The same holds for the additional external assets of v_2. To determine the external assets of v_1 in $\mathcal{F}_\Delta$, first, observe that the incoming assets of v_1 are at least the incoming assets of v_1 in $\mathcal{F}$ of $a_{v_1} - a^x_{v_1}$. In $\mathcal{F}_\Delta$, the bank receives additional payments of $\Delta_{t_{v_1}} + \Delta_{t_{u_1}}$. Furthermore, the raised external assets could enable payments in new cycles where all payments from v_1 towards the cycle are forwarded back to v_1.

Formally, recall that the total assets of s_{v_1} in $\mathcal{H}_\Delta$ equal the total assets of v_1 in $\mathcal{F}_\Delta$. The incoming payments of v_1 in $\mathcal{F}_\Delta$ are the sum of total assets of t_{u_1} and t_{v_1} in $\mathcal{H}_\Delta$. Since $\mathcal{H}$ is a pre-network of $\mathcal{H}_\Delta$ with clearing state equivalent to $\mathcal{F}$, this sum is given by $(a_v - a^x_v) + \Delta_{t_{v_1}} + \Delta_{t_{u_1}}$. Now the external assets of v_1 in $\mathcal{F}_\Delta$ are the total assets of s_{v_1} minus the sum of total assets of t_{u_1} and t_{v_1} in $\mathcal{H}_\Delta$, i.e.,

$$\begin{aligned}
&a_{v_1} + \delta_{v_1} - \left((a_{v_1} - a^x_{v_1}) + \Delta_{t_{v_1}} + \Delta_{t_{u_1}}\right) \\
&= a^x_{v_1} + \delta_{v_1} - \Delta_{t_{v_1}} - \Delta_{t_{u_1}} \\
&= a^x_{v_1}
\end{aligned}$$

where the last equality follows by assumption. Analogously, v_2 has external assets of $a^x_{v_2}$ in $\mathcal{F}_\Delta$. Hence, every bank has the same external assets in $\mathcal{F}$ and in $\mathcal{F}_\Delta$. However, since $a_{v_1} + \delta_{v_1} > a_{v_1}$ or $a_{v_2} + \delta_{v_2} > a_{v_2}$ the payments in the clearing state in $\mathcal{F}_\Delta$ are strictly higher than in $\mathcal{F}$. This contradicts the fact that the clearing state $\mathbf{p}$ is the maximal fixed point. With the same argument, $\delta_{s_{v_2}} \neq \Delta_{t_{v_2}} + \Delta_{t_{u_2}}$ follows. □

Theorem 12. *A financial network with monotone allocation rules allows no positive debt swap.*

Proof. For a financial network $\mathcal{F}$, suppose the debt swap of e_1 and e_2 is positive, i.e., $a^\sigma_{v_1} > a_{v_1}$ and $a^\sigma_{v_2} > a^\sigma_{v_1}$. We set δ_{v_1} and δ_{v_2} to the increase in total assets of v_1 and v_2 due to the swap, i.e., $\delta_{v_1} := a^\sigma_{v_1} - a_{v_1}$ and $\delta_{v_2} := a^\sigma_{v_2} - a_{v_2}$. Now consider the financial network $\mathcal{H}$ corresponding to $\mathcal{F}$ and raise the external assets of s_{v_1} by at most δ_{v_1} and external assets of s_{v_2} by at most δ_{v_2}. Let $g_{t_{u_1}} : [0, \delta_{v_1}] \times [0, \delta_{v_2}] \to [0, \infty]$ be the increase in total assets at t_{u_1} as a function of the increase in external assets at s_{v_1} and s_{v_2} in the clearing state of network $\mathcal{H}$. We define $g_{t_{v_1}}$ similarly. The payment functions of all banks are non-decreasing and continuous, and so are $g_{t_{v_1}}$ and $g_{t_{u_1}}$. In particular, $g_{t_{v_1}}(0,0) = g_{t_{u_1}}(0,0) = 0$.

The sum $g_1 = g_{t_{u_1}} + g_{t_{v_1}}$ is non-decreasing and continuous as well, with $g_1(0,0) = 0$. By non-expansivity, we have that $g_1(x,y) \leq x + y$ for all $(x,y) \in [0, \delta_{v_1}] \times [0, \delta_{v_2}]$. We will now show that whenever the debt swap of e_1 and e_2 is positive, there exists an $(x,y) \in [0, \delta_{v_1}] \times [0, \delta_{v_2}]$ with $(x,y) > (0,0)$, i.e., $x > 0$ and $y > 0$, such that $g_1(x,y) = x$. This is a contradiction to Lemma 4 (2).

We proceed in three steps.

(i) $g_1(x,0) < x$ for all $x \in (0, \delta_{v_1}]$:
Since $g_1(x,y) \leq x + y$, it follows that $g_1(x,0) \leq x$, for all $x > 0$. Now $g_1(x,0) = x$ for some $x > 0$ would imply $\{t_{v_1}, t_{u_1}\}$-linearity of s_{v_1} (Lemma 4 (1)) and a contracticition to Lemma 4 (2).

(ii) $g_1(c,y) > c$ for some $c \in (0, \delta_{v_1}]$, $y \in (0, \delta_{v_2}]$:
Fix a value $y \in (0, \delta_{v_2}]$ with $y \leq \delta_{v_1}$. Then clearly $g_1(0,y) \geq 0$. Moreover, $g_1(0,y) > 0$ since otherwise s_{v_2} would be $\{t_{v_2}, t_{u_2}\}$-linear (Lemma 4 (1)) and we obtain a contradiction to Lemma 4 (2). Note that $g_1(0,y) \leq y \leq \delta_{v_1}$. Since $g_1(0,y) > 0$ and g_1 is monotone, we obtain $g_1\big(g_1(0,y), y\big) \geq g_1(0,y)$. Using the notation $c := g_1(0,y)$, we have $0 < c \leq \delta_{v_1}$ and $g_1(c,y) \leq c$. Since $g_1(c,y) = c$ implies a contradiction to Lemma 4 (2), we must have $g_1(c,y) > c$.

(iii) $g_1(x,y) = x$ for some $(x,y) \in (0, \delta_{v_1}] \times (0, \delta_{v_2}]$:
(i) and (ii) imply that there exists a pair $(c,y) \in [0, \delta_{v_1}] \times [0, \delta_{v_2}]$ such that $g_1(c,0) < c$ and $g_1(c,y) > c$. By the intermediate value theorem, there exists a $d \in [0,y]$ satisfying $g_1(c,d) = c$. Observe that in particular $y > 0$ holds true.

□

The theorem substantially generalizes an earlier result for proportional payments [1]. Let us note that the proof in the full version of [1] relies on a Lemma B.1 which is incorrect. The lemma claims the following: For two financial networks $\mathcal{F}^1$ and $\mathcal{F}^2$ with proportional payments on the same set of nodes, suppose there exists one source node s with external assets a^x_s while $a^x_u = 0$ for all other banks in $\mathcal{F}^1$ and $\mathcal{F}^2$. If $l^{\mathcal{F}^1}_e \leq l^{\mathcal{F}^2}_e$ for every edge, then $a^{\mathcal{F}^1} \leq a^{\mathcal{F}^2}$ follows.

As a counterexample to this claim, consider a financial network over the three banks s, w_1 and w_2 and edges (s,w_1) and (s,w_2) with $l_{(s,w_1)} = 1$ and $l_{(s,w_2)} = 2$. Let the external assets be given by $a^x_s = 2$ and $a^x_{w_1} = a^x_{w_2} = 0$. Then, s pays $\frac{1}{1+2} \cdot 2 = \frac{2}{3}$ towards w_1 and $\frac{2}{1+2} \cdot 2 = \frac{4}{3}$ towards w_2. Now assume the edge weight $l_{(s,w_1)}$ is raised to 2. Then, the payments from s to w_1 as well as w_2 are changed to $\frac{2}{2+2} \cdot 2 = 1 < \frac{4}{3}$.

Proposition 8. *For every financial network, there is no debt swap σ such that $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} > p_{e_2}$.*

Proof. For a given network $\mathcal{F}$ consider the corresponding network $\mathcal{G}$ after applying source-sink equivalency to the edges e_1 and e_2. This procedure adds the new banks t_{u_1}, t_{u_2} and changes the edges e_1 and e_2 from (u_1, v_1) and (u_2, v_2) to (u_1, t_{u_1}) and (u_2, t_{u_2}), respectively. Additionally, the external assets of v_1 and v_2 are updated to $a^x_{v_1} + p_{e_1}$ and $a^x_{v_2} + p_{e_2}$, respectively. Analogously, define $\mathcal{G}^\sigma$ for $\mathcal{F}^\sigma$. Observe that $\mathcal{G}$ and $\mathcal{G}^\sigma$ only differ in the external assets of v_1 and v_2. Hence, $\mathcal{G}$ is a pre-network of $\mathcal{G}^\sigma$ and, by monotonicity of the allocation rule, both v_1 and v_2 strictly profit from the edge swap. This is a contradiction to Theorem 1. □

Proposition 9. *For every semi-positive debt swap σ, we have either $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} \leq p_{e_2}$, or $p^\sigma_{e_2} > p_{e_2}$ and $p^\sigma_{e_1} \leq p_{e_1}$.*

Proof. Proposition 1 shows that $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} > p_{e_2}$ can never be satisfied simultaneously. If $p^\sigma_{e_1} \leq p_{e_1}$ and $p^\sigma_{e_2} \leq p_{e_2}$, then the same source-sink transformation as in the proof of Proposition 1 shows that the network after swapping $\mathcal{G}^\sigma$ is a pre-network of $\mathcal{G}$. By Corollary 10 such a swap cannot be semi-positive. Consequently, either $p^\sigma_{e_1} > p_{e_1}$ and $p^\sigma_{e_2} \leq p_{e_2}$, or $p^\sigma_{e_2} > p_{e_2}$ and $p^\sigma_{e_1} \leq p_{e_1}$. □

Theorem 13. *For every financial network with monotone allocation rules, a debt swap is semi-positive if and only if it is Pareto-improving.*

Proof. First, let the debt swap of e_1 and e_2 be Pareto-improving. For a contradiction, assume it is not semi-positive. Then, w.l.o.g. the following two cases can occur: (1) $a^\sigma_{v_1} < a_{v_1}$ and $a^\sigma_{v_2} > a_{v_2}$, or (2) $a^\sigma_{v_1} \leq a_{v_1}$ and $a^\sigma_{v_2} \leq a_{v_2}$. Towards (1), it is easy to see that v_1 is harmed by the swap. Hence, it is not Pareto-improving. For (2), consider the two networks $\mathcal{H}$ and $\mathcal{H}^\sigma$ after applying source-sink equivalency to the swapping edges and v_1 and v_2 in $\mathcal{F}$ and $\mathcal{F}^\sigma$. Since $\mathcal{H}$ and $\mathcal{H}^\sigma$ only differ in the external assets of v_1 and v_2, $\mathcal{H}$ is a pre-network of $\mathcal{H}^\sigma$. Consequently, by Corollary 10, the total assets of all banks in $\mathcal{H}$ are at least as high as those $\mathcal{H}^\sigma$. Hence, no bank strictly profits by the debt swap, a contradiction.

For the other direction, suppose the debt swap of e_1 and e_2 is semi-positive and w.l.o.g. $a^\sigma_{v_1} > a_{v_1}$ and $a^\sigma_{v_2} = a_{v_2}$. Then, by Lemma 12, $\mathcal{H}$ is a pre-network of $\mathcal{H}^\sigma$, and thus, by Corollary 10, the total assets of all banks after the swap are at least as high as those before the swap. Thus, the debt swap is a Pareto-improvement. □

References

[1] P.A. Papp, R. Wattenhofer, Debt swapping for risk mitigation in financial networks, in: Proc. 22nd Conf. Econ. Comput. (EC), 2021, pp. 765–784.
[2] S. Schuldenzucker, S. Seuken, Portfolio compression in financial networks: incentives and systemic risk, in: Proc. 21st Conf. Econ. Comput. (EC), 2020, p. 79.
[3] L. Veraart, When does portfolio compression reduce systemic risk?, Math. Finance 32 (3) (2019) 727–778.
[4] M. Hoefer, C. Ventre, L. Wilhelmi, Algorithms for claims trading, in: Proc. 41st Symp. Theoret. Aspects Comput. Sci. (STACS), 2024, pp. 42:1–42:17.
[5] P. Csóka, P. Jean-Jacques Herings, Decentralized clearing in financial networks, Manag. Sci. 64 (10) (2018) 4681–4699.
[6] N. Bertschinger, M. Hoefer, D. Schmand, Flow allocation games, Math. Oper. Res. 50 (1) (2025) 68–89.
[7] S. Ioannidis, B. De Keijzer, C. Ventre, Financial networks with singleton liability priorities, Theor. Comput. Sci. 963 (2023) 113965.
[8] L. Eisenberg, T. Noe, Systemic risk in financial systems, Manag. Sci. 47 (2) (2001) 236–249.
[9] P. Kanellopoulos, M. Kyropoulou, H. Zhou, Debt transfers in financial networks: complexity and equilibria, in: Proc. 22th Conf. Auton. Agents and Multi-Agent Syst. (AAMAS), 2023, pp. 260–268.
[10] S. Schuldenzucker, S. Seuken, S. Battiston, Finding clearing payments in financial networks with credit default swaps is PPAD-complete, in: Proc. 8th Symp. Innov. Theoret. Comput. Sci. (ITCS), 2017, pp. 32:1–32:20.
[11] S. Ioannidis, B. De Keijzer, C. Ventre, Strong approximations and irrationality in financial networks with derivatives, in: Proc. 49th Int. Colloq. Autom. Lang. Programming (ICALP), 2022, pp. 76:1–76:18.
[12] S. Schuldenzucker, S. Seuken, S. Battiston, Default ambiguity: credit default swaps create new systemic risks in financial networks, Manag. Sci. 66 (5) (2020) 1981–1998.
[13] P.A. Papp, R. Wattenhofer, Default ambiguity: finding the best solution to the clearing problem, in: Proc. 17th Conf. Web and Internet Econ. (WINE), 2021, pp. 391–409.

[14] P.A. Papp, R. Wattenhofer, Network-aware strategies in financial systems, in: Proc. 47th Int. Colloq. Autom. Lang. Programming (ICALP), 2020, p. 91.
[15] P. Kanellopoulos, M. Kyropoulou, H. Zhou, Optimal bailouts and strategic debt forgiveness in financial networks, Artif. Intell. 349 (2025) 104424.
[16] B. Egressy, A. Plesner, R. Wattenhofer, What is the price for lending in financial networks?, in: Proc. 25th Int. Conf. Princ. Pract. Multi-Agent Syst. (PRIMA), 2024, pp. 120–135.
[17] M. Papachristou, J. Kleinberg, Allocating stimulus checks in times of crisis, in: Proc. 31th World Wide Web Conf. (WWW), 2022, pp. 16–26.
[18] P.A. Papp, R. Wattenhofer, Sequential defaulting in financial networks, in: Proc. 12th Symp. Innov. Theoret. Comput. Sci. (ITCS), 2021, pp. 52:1–52:20.
[19] T. Friedetzky, D. Kutner, G. Mertzios, I. Stewart, A. Trehan, Payment scheduling in the interval debt model, Theor. Comput. Sci. 1028 (2025) 115028.
[20] P. Kanellopoulos, M. Kyropoulou, H. Zhou, On priority-proportional payments in financial networks, Theor. Comput. Sci. 1014 (2024) 114767.
[21] M. Hoefer, L. Wilhelmi, Seniorities and minimal clearing in financial network games, in: Proc. 15th Symp. Algorithmic Game Theory (SAGT), 2022, pp. 187–204.
[22] H. Froese, M. Hoefer, L. Wilhelmi, Dynamic debt swapping in financial networks, in: Proc. 4th Symp. Algorith. Found. Dynamic Networks (SAND), 2025, pp. 2:1–2:16.
[23] D. Johnson, C. Papadimitriou, M. Yannakakis, How easy is local search?, J. Comput. Syst. Sci. 37 (1) (1988) 79–100.
[24] M. Yannakakis, Computational complexity, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Princeton University Press, 2003, pp. 19–56.
[25] A. Schäffer, Simple local search problems that are hard to solve, SIAM J. Comput. 20 (1) (1991) 56–87.
[26] P. Alimonti, V. Kann, Some APX-completeness results for cubic graphs, Theor. Comput. Sci. 237 (1–2) (2000) 123–134.
[27] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Math. 182 (1999) 105–142.
[28] W. Savitch, Relationships between nondeterministic and deterministic tape complexities, J. Comput. Syst. Sci. 4 (2) (1970) 177–192.
[29] N. Alon, S. Friedland, The maximum number of perfect matchings in graphs with a given degree sequence, Electron. J. Comb. 15 (13) (2008) 1.
[30] P. Gopalan, P. Kolaitis, E. Maneva, C. Papadimitriou, The connectivity of Boolean satisfiability: computational and structural dichotomies, SIAM J. Comput. 38 (6) (2009) 2330–2355.