SPARSIFYING DEFAULTS: OPTIMAL BAILOUT POLICIES FOR FINANCIAL NETWORKS IN DISTRESS

Zhang Li and Ilya Pollak

School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47906
li424@purdue.edu and ipollak@ecn.purdue.edu

ABSTRACT

We propose a quantitative framework for constructing optimal policies to manage systemic risk in financial networks. We analyze borrower-lender networks where all the loan amounts and cash flows are known, and where some nodes may default in the absence of external intervention. Given a fixed amount of cash to be injected into the system, we address the problem of allocating it among the nodes to minimize the overall amount of unpaid liabilities. We show that this problem is equivalent to a linear program. In addition, we address the problem of allocating the cash injection amount so as to minimize the number of nodes in default. For this problem, we develop an approximate algorithm which uses reweighted ℓ_1 minimization. We illustrate this algorithm using two synthetic network structures for which the optimal solution can be calculated exactly. We show through numerical simulations that the solutions calculated by our algorithm are close to optimal.

Index Terms— Risk, contagion, networks, sparsity, financial systems, optimal resource allocation.

1. MOTIVATION, PRIOR LITERATURE, AND OUR CONTRIBUTIONS

The events of the last few years revealed an acute need for tools to systematically model and analyze large financial networks. Many applications of such tools include the forecasting of systemic failures and analyzing probable effects of economic policy decisions.

We consider the problem of optimizing the amount and structure of a bailout in a borrower-lender network. Two broad application scenarios motivate our work: day-to-day monitoring of financial systems and decision making during an imminent crisis. Examples of the latter include the decision in September 1998 by a group of financial institutions to rescue Long-Term Capital Management, and the decisions by the Treasury and the Fed in September 2008 to rescue AIG and to let Lehman Brothers fail. The deliberations leading to these and other similar actions have been extensively covered in the press. These reports suggest that the decision making processes could benefit from quantitative methods for analyzing potential repercussions of contemplated actions. In addition, such methods could help avoid systemic crises in the first place, by informing day-to-day actions of financial institutions and governments.

Forecasting and preventing systemic failures is an open problem, despite a surge in the research literature during the last four years. There are two main difficulties. First, the data on borrower-lender relationships and capital structure of financial institutions is largely unavailable to academic researchers. Even the data available to regulators is far from exhaustive and perfect. Second, the network of financial relationships is very large, complex, and dynamic.

Notable examples of network topology analysis based on real data are [1] and [2]. Estimation of the network structure is used in [3] and [4] to propose a new approach for assessing systemic financial stability of a banking system.

Often, systemic failures are caused by an epidemic of defaults whereby a group of nodes unable to meet their obligations trigger the insolvency of their lenders, leading to the defaults of lenders' lenders, etc, until this spread of defaults infects a large part of the system. For this reason, many studies have been devoted to discovering network structures conducive to default contagion [5, 6, 7, 8, 9]. The relationships between the probability of a systemic failure and the average connectivity in the network are investigated in [6, 7, 8]. In addition, [9] examines other features such as the distribution of degrees and the structure of the subgraphs of contagious links.

While potentially useful in policymaking, these references do not provide specific policy recipes. Literature on quantitative models for optimizing policy decisions has focused on analyzing the efficacy of bailouts and understanding the behavior of firms in response to bailouts. To this end, game-theoretic models are proposed in [10] and [11] that have two agents: the government and a single private sector entity. However, the current state of the art lacks quantitative frameworks to develop policies for preventing systemic failures in financial networks that consist of a large number of private-sector agents. The main contribution of our paper is to propose such a framework. Specifically, we are interested in addressing the following problem, given a financial network model.

> **Problem I:** Given a fixed amount of cash C to be injected into the system, how should it be distributed among the nodes in order to achieve the smallest overall amount D of unpaid liabilities?

An alternative, Lagrangian, formulation of the same problem, is to both select C and determine how to distribute it in order to minimize $C + \lambda D$, where λ is the cost associated with every dollar of unpaid liabilities. In this formulation, λ can be used to model the trade-off between the costs of a bailout (direct costs as well as moral hazard) and the costs of defaults.

In this work, we consider a static model with a single maturity date, and with a known network structure. Specifically, we assume that we know both the amounts owed by every node in the network to every other node, and the amounts of cash available at every node. Even for this relatively simple model, Problem I is far from straightforward, because of a nonlinear relationship between the cash injection amounts and the loan repayment amounts. Building upon the results from [12], we construct algorithms for computing exact solutions for Problem I and its Lagrangian variant, by showing that both formulations are equivalent to linear programs.

We also consider another problem where the objective is to minimize the number of defaulting nodes rather than the overall amount of unpaid liabilities:

> **Problem II:** Given a fixed amount of cash C to be injected into the system, how should it be distributed among the nodes in order to minimize the number of nodes in default, N_d?

For Problem II, we develop an approximate algorithm using a reweighted ℓ_1 minimization approach inspired by [13]. We illustrate our algorithm using two synthetic network structures for which the optimal solutions can be calculated exactly, and show through numerical simulations that the solutions calculated by our algorithm are close to optimal.

In Section 2 we describe our model and the results from prior literature that we use. Our own results—the equivalence of Problem I to a linear program and the approximate algorithm for Problem II—are described in Section 3.

2. NOTATION, MODEL, AND BACKGROUND

Our network model is a directed graph with N nodes where a directed edge from node i to node j with weight $L_{ij} > 0$ signifies that i owes \$$L_{ij}$ to j. This is a one-period model with no dynamics—i.e., we assume that all the loans are due on the same date and all the payments occur on that date. We use the following notation:

- any inequality whose both sides are vectors is componentwise;
- $\mathbf{0}$, $\mathbf{1}$, $\mathbf{e}$, $\mathbf{c}$, $\bar{\mathbf{p}}$, $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{r}$ are all vectors in $\mathbb{R}^N$ defined in Table 1;
- $D = \mathbf{1}^T(\bar{\mathbf{p}} - \mathbf{p})$ is the overall amount of unpaid liabilities in the system;
- N_d is the number of nodes in default, i.e., the number of nodes i whose payments are below their liabilities, $p_i < \bar{p}_i$;
- Π_{ij} is what node i owes to node j, as a fraction of the total amount owed by node i,

$$\Pi_{ij} = \begin{cases} \frac{L_{ij}}{\bar{p}_i} & \text{if } \bar{p}_i \neq 0, \\ 0 & \text{otherwise;} \end{cases}$$

- Π and L are the matrices whose entries are Π_{ij} and L_{ij}, respectively.

Following [12], we make the following assumptions.

- If i's total funds are at least as large as its liabilities (i.e., $r_i \geq \bar{p}_i$) then all i's creditors get paid in full.
- If i's total funds are smaller than its liabilities, then i pays all its funds to its creditors.

Table 1. Notation for several vector quantities.

VECTOR	i-TH COMPONENT
$\mathbf{0}$	0
$\mathbf{1}$	1
$\mathbf{e} \geq \mathbf{0}$	cash on hand at node i
$\mathbf{c} \geq \mathbf{0}$	external cash injection to node i
$\bar{\mathbf{p}}$	the amount node i owes to all its creditors
$\mathbf{p} \leq \bar{\mathbf{p}}$	the total amount node i actually repays all its creditors on the due date of the loans
$\bar{\mathbf{p}} - \mathbf{p}$	node i's total unpaid liabilities
$\mathbf{q}$	the total amount node i actually receives from all its borrowers
$\mathbf{r} = \mathbf{q} + \mathbf{e} + \mathbf{c}$	the total funds available to i for making payments to its creditors

- All i's debts have the same seniority. This means that, if i's liabilities exceed its total funds (i.e., $r_i < \bar{p}_i$) then each creditor gets paid in proportion to what it is owed. This guarantees that the amount actually received by node j from node i is always $\Pi_{ij} p_i$. Therefore, the total amount received by any node i from all its borrowers is $q_i = \sum_{j=1}^{N} \Pi_{ji} p_j$.

As defined in [12], a *clearing payment vector* $\mathbf{p}$ is a vector of borrower-to-lender payments that is consistent with these conditions for given L, $\mathbf{e}$, and $\mathbf{c}$. It is shown in [12] (Theorem 2) that a unique $\mathbf{p}$ exists for any network that satisfies a mild technical assumption. We restrict our attention to models that satisfy this assumption and therefore have a unique clearing payment vector $\mathbf{p}$.

3. RESULTS

3.1. Minimizing the amount of unpaid liabilities

Consider a network with a known structure of liabilities L and a known cash vector $\mathbf{e}$. Using the notation established in the preceding section, we can see that Problem I seeks to find a cash injection allocation vector $\mathbf{c}$ to minimize the total amount of unpaid liabilities,

$$D = \mathbf{1}^T(\bar{\mathbf{p}} - \mathbf{p}),$$

subject to the constraint that the total amount of cash injection is some given number C:

$$\mathbf{1}^T\mathbf{c} = C.$$

Our first result establishes the equivalence of Problem I and a linear programming problem.

Theorem 1. *Assume that the liabilities matrix L, the cash-on-hand vector $\mathbf{e}$, and the total cash injection amount C are fixed and known. Assume that the network satisfies all the conditions listed above. Then Problem I has a solution which can be ob-*

tained by solving the following linear program:

$$\begin{aligned}&\textit{find } \mathbf{c} \textit{ and } \mathbf{p} \textit{ to maximize } \mathbf{1}^T\mathbf{p} \\ &\textit{subject to} \\ &\mathbf{1}^T\mathbf{c} = C, \\ &\mathbf{c} \geq \mathbf{0}, \\ &\mathbf{0} \leq \mathbf{p} \leq \bar{\mathbf{p}}, \\ &\mathbf{p} \leq \Pi^T\mathbf{p} + \mathbf{e} + \mathbf{c}.\end{aligned} \tag{1}$$

Proof. Since the constraints on $\mathbf{c}$ and $\mathbf{p}$ in linear program (1) form a closed and bounded set in $\mathbb{R}^{2N}$, a solution exists. Moreover, for any fixed $\mathbf{c}$, it follows from Lemma 4 in [12] that the linear program has a unique solution for $\mathbf{p}$ which is the clearing payment vector for the system.

Let $(\mathbf{p}^*, \mathbf{c}^*)$ be a solution to (1). Suppose that there exists a cash injection allocation that leads to a smaller total amount of unpaid liabilities than does $\mathbf{c}^*$. In other words, suppose that there exists $\mathbf{c}' > \mathbf{0}$, with $\mathbf{1}^T\mathbf{c}' = C$, such that the corresponding clearing payment vector $\mathbf{p}'$ satisfies $\mathbf{1}^T(\bar{\mathbf{p}} - \mathbf{p}') < \mathbf{1}^T(\bar{\mathbf{p}} - \mathbf{p}^*)$, or, equivalently,

$$\mathbf{1}^T\mathbf{p}^* < \mathbf{1}^T\mathbf{p}'. \tag{2}$$

Note that $\mathbf{c}'$ satisfies the first two constraints of (1). Moreover, since $\mathbf{p}'$ is the corresponding clearing payment vector, the last two constraints are satisfied as well. The pair $(\mathbf{p}', \mathbf{c}')$ is thus in the constraint set of our linear program. Therefore, Eq. (2) contradicts the assumption that $(\mathbf{p}^*, \mathbf{c}^*)$ is a solution to (1). This completes the proof that $\mathbf{c}^*$ is the allocation of C that achieves the smallest possible amount D of unpaid liabilities. □

In the Lagrangian formulation of Problem I, we are given a weight λ and must choose the total cash injection amount C and its allocation $\mathbf{c}$ to minimize $C + \lambda D$. This is equivalent to the following linear program:

$$\begin{aligned}&\text{find } C, \mathbf{c}, \text{ and } \mathbf{p} \text{ to maximize } \lambda\mathbf{1}^T\mathbf{p} - C \\ &\text{subject to the same constraints as in (1).}\end{aligned} \tag{3}$$

This equivalence follows from Theorem 1: denoting a solution to (3) by $(C^*, \mathbf{p}^*, \mathbf{c}^*)$, we see that the pair $(\mathbf{p}^*, \mathbf{c}^*)$ must be a solution to (1) for $C = C^*$. At the same time, the fact that C^* maximizes the objective function in (3) means that it minimizes $C + \lambda D = C + \lambda\mathbf{1}^T(\bar{\mathbf{p}} - \mathbf{p})$, since $\bar{\mathbf{p}}$ is a fixed constant.

3.2. Minimizing the number of defaults

Given that the total amount of cash injection is C, Problem II seeks to find a cash injection allocation vector $\mathbf{c}$ to minimize the number of defaults N_d, i.e., the number of nonzero entries in the vector $\bar{\mathbf{p}} - \mathbf{p}$.

We adapt the reweighted ℓ_1 minimization strategy approach from Section 2.2 of [13]. Our algorithm solves a sequence of weighted versions of the linear program (1), with the weights designed to encourage sparsity of $\bar{\mathbf{p}} - \mathbf{p}$. In the following pseudocode of our algorithm, $\mathbf{w}^{(m)}$ is the weight vector during the m-th iteration.

1. $m \leftarrow 0$.
2. Select $\mathbf{w}^0$ (e.g., $\mathbf{w}^0 \leftarrow \mathbf{1}$).
3. Solve linear program (1) with objective function replaced by $\mathbf{p}^T\mathbf{w}^{(m)}$.
4. Update the weights: for each $i = 1, \cdots, N$,
$$w_i^{(m+1)} \leftarrow \frac{K}{\exp\left(\bar{p}_i - p_i^{*(m)}\right) - 1 + \epsilon},$$
where $K > 0$ and $\epsilon > 0$ are constants, and $\mathbf{p}^{*(m)}$ is the clearing payment vector obtained in Step 3.
5. If $\|\mathbf{w}^{(m+1)} - \mathbf{w}^{(m)}\|_1 < \delta$, where $\delta > 0$ is a constant, stop; else, increment m and go to Step 3.

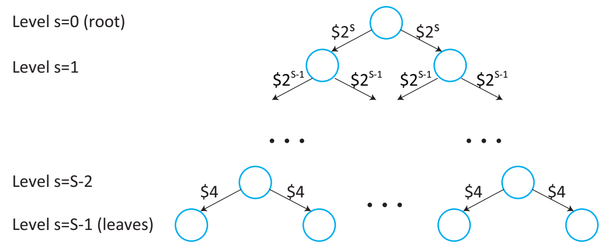

Fig. 1. Binary tree network.

Note that nodes for which $\bar{p}_i - p_i^{*(m)}$ is very small require very little additional resources to avoid default. This is why Step 4 is designed to give more weight to such nodes, thereby encouraging larger cash injections into them. On the other hand, nodes for which $\bar{p}_i - p_i^{*(m)}$ is very large require a lot of cash to become solvent. The algorithm essentially "gives up" on such nodes by assigning them small weights.

We test the algorithm on two networks for which we know the optimal solution. First, we use a full binary tree with S levels and $N = 2^S - 1$ nodes. As shown in Fig. 1, levels 0 and $S-1$ correspond to the root and the leaves, respectively. Every node at level $s < S-1$ owes $\$2^{S-s}$ to each of its two creditors (children). We set $\mathbf{e} = \mathbf{0}$.

If $C < 8$, then all $2^{S-1} - 1$ non-leaf nodes are in default, and the 2^{S-1} leaves are not in default. In aggregate, the nodes at any level $s < S - 1$ owe $\$2^{S+1}$ the nodes at level $s + 1$. Therefore, if $C \geq \$2^{S+1}$, then $N_d = 0$ can be achieved by allocating the entire amount to the root node.

For $8 \leq C < 2^{S+1}$, we first observe that if $C = 2^{S+1-s}$ for some integer s, then the optimal solution is to allocate the entire amount to a node at level s. This would prevent the defaults of this node and all its $2^{S-s-1} - 2$ non-leaf descendants, leading to $2^{S-1} - 2^{S-s-1}$ defaults. If C is not a power of two, we can represent it as a sum of powers of two and apply the same argument recursively, to yield the following optimal number of defaults:

$$N_d = T(S) - \sum_{u=4}^{U} b(u) \cdot T(u-2),$$

where $T(x) = 2^{x-1} - 1$ is the number of non-leaf nodes in an x-level complete binary tree, $b(u)$ is the u-th bit in the binary representation of C (right to left) and U is the number of bits. To summarize, the smallest number of defaults N_d, as a function of the cash injection amount C, is:

$$N_d(C) = \begin{cases} T(S) & \text{if } C < 8, \\ T(S) - \sum_{u=4}^{U} b(u)T(u-2) & \text{if } 8 \leq C < 2^{S+1}, \\ 0 & \text{if } C \geq 2^{S+1}. \end{cases} \tag{4}$$

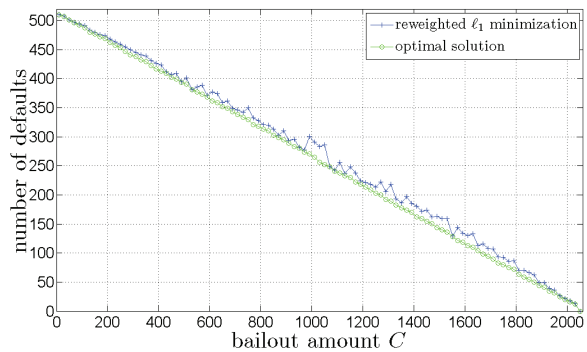

Fig. 2. Our algorithm for minimizing the number of defaults vs the optimal solution, for the binary tree network of Fig. 1.

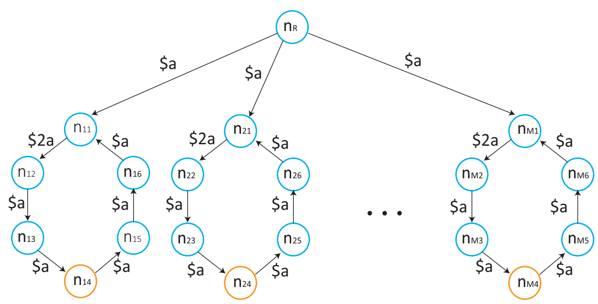

Fig. 3. Network topology with cycles.

In our test, we set $S = 10$. The green line in Fig. 2 is a plot of the minimum number of defaults as a function of C from Eq. (4). The blue line is the solution calculated by our reweighted ℓ_1-minimization algorithm with $K = 1000$, $\epsilon = 0.001$ and $\delta = 0.001$. The algorithm was run using six different initializations: five random ones and $\mathbf{w^{(0)}} = \mathbf{1}$. Among the six solutions, the one with the smallest number of defaults was selected. As evident from Fig. 2, the results are very close to the optimal for the entire range of C.

We also test our algorithm on a network with cycles shown in Fig. 3. The network contains M cycles with six nodes each. The nodes in the k-th cycle are denoted $n_{k1}, n_{k2}, \cdots, n_{k6}$. Node n_{k1} owes \$$2a$ to n_{k2}. Node n_{k6} owes \$$a$ to n_{k1}. For $i = 2, \cdots, 5$, n_{ki} owes \$$a$ to $n_{k(i+1)}$. The root node, denoted as n_R, owes \$$a$ to n_{k1}, for every $k = 1, 2, \cdots, M$. In order to satisfy the unique clearing payment vector condition in [12], we set $e = \$0.01 < a$ for nodes n_{k4} (shown in orange in Fig. 3), and $e = \$0$ for other nodes.

If $C < a$, then the root node and all M nodes connected to the root, $n_{k1}(k = 1, 2, \cdots, M)$, are in default. The remaining $5M$ nodes are not in default.

If $C \geq aM$, then allocating the entire amount C to the root yields zero defaults.

If $a \leq C < aM$, then giving \$$a$ to node n_{k1} will prevent it from defaulting. Thus, the total number of defaults in this case is $M + 1 - [C/a]$.

Summarizing, for this network structure, the smallest number of defaults N_d, as a function of the cash injection amount

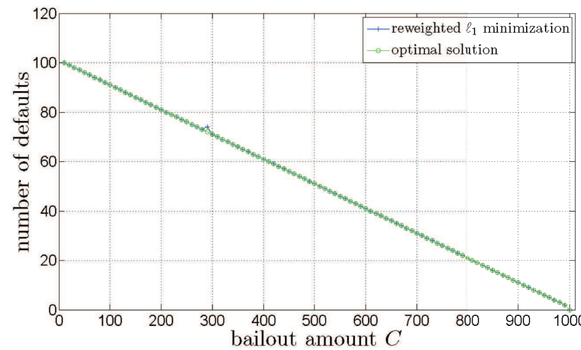

Fig. 4. Our algorithm for minimizing the number of defaults vs the optimal solution, for the network of Fig. 3.

C, is:

$$N_d(C) = \begin{cases} M+1 & \text{if } C < a, \\ M+1-[C/a] & \text{if } a \leq C < aM, \\ 0 & \text{if } C \geq aM. \end{cases} \tag{5}$$

In Fig. 4, $a = 10$, $M = 100$. The green line is a plot of the minimum number of defaults as a function of C and the result of the reweighted ℓ_1-minimization algorithm for the network with cycles is illustrated as the blue line with $K = 1000$, $\epsilon = 0.001$, $\delta = 0.001$. And as same as binary tree example, we run the algorithm with six different initializations: $\mathbf{w^{(0)}} = \mathbf{1}$ and five random ones. From Fig. 4, the results of reweighted ℓ_1-minimization algorithm are also very close to the optimal for the topology with cycles.

4. FUTURE WORK

No regulatory body ever has full and accurate information about the books of all financial institutions. Therefore, in most practical applications, it is unrealistic to assume that the network structure is known. A more realistic model might be that the parameters $\mathbf{e}$ and L are random variables which are only partially—and perhaps indirectly—observed. The total amount of unpaid liabilities and the number of defaults are then random variables as well, and one might address the problem of allocating C among the nodes so as to minimize either the expectations of these random variables or some other cost functions that depend on the distributions of these random variables.

5. REFERENCES

[1] K. Soramäki, M.L. Bech, J. Arnold, R.J. Glass, and W.E. Beyeler, "The topology of interbank payment flows," *Physica A: Statistical Mechanics and its Applications*, vol. 379, no. 1, pp. 317–333, 2007.

[2] M. Boss, H. Elsinger, M. Summer, and S. Thurner, "The network topology of the interbank market," *arXiv preprint cond-mat/0309582*, 2003.

[3] H. Elsinger, A. Lehar, and M. Summer, "Risk assessment for banking systems," *Management science*, vol. 52, no. 9, pp. 1301–1314, 2006.

[4] H. Elsinger, A. Lehar, and M. Summer, "Using market information for banking system risk assessment," *Available at SSRN 787929*, 2005.

[5] J. Lorenz, S. Battiston, and F. Schweitzer, "Systemic risk in a unifying framework for cascading processes on networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 441–460, 2009.

[6] P. Gai and S. Kapadia, "Contagion in financial networks," *Proceeding of Royal Society A: Mathematical, Physical and Engineering Science*, vol. 466, no. 2120, pp. 2401–2423, August 2010.

[7] S. Battiston, D.D. Gatti, M. Gallegati, B. Greenwald, and J. E. Stiglitz, "Liaisons dangereuses: Increasing connectivity, risk sharing, and systemic risk," *Journal of Economic Dynamics and Control*, vol. 36, no. 8, pp. 1121 – 1141, 2012.

[8] F. Allen and D. Gale, "Financial contagion," *Journal of Political Economy*, vol. 108, no. 1, pp. 1–33, 2000.

[9] H. Amini, R. Cont, and A. Minca, "Resilience to contagion in financial networks," *Available at SSRN 1865997*, 2010.

[10] G. Mundaca, "Optimal bailout during currency and financial crises: A sequential game analysis," Tech. Rep., Department of Economics, University of Oslo, 2002.

[11] A. Bernardo, E. Talley, and I. Welch, "A model of optimal government bailouts," Tech. Rep.

[12] L. Eisenberg and T.H. Noe, "Systemic risk in financial systems," *Management Science*, vol. 47, no. 2, pp. 236–249, February 2001.

[13] E.J. Candès, M.B. Wakin, and S.P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.