Abstract

Contagion is a process whereby the collapse of a node in a network leads to the collapse of neighboring nodes and thereby sets off a chain reaction in the network. It thereby creates a special type of time-dependent network. Such processes are studied in various applications, e.g., in financial network analysis, infection diffusion prediction, supply chain management or gene regulation. Visual Analytics methods can help analysts examine contagion effects. For this purpose, network visualizations need to be complemented with specific features to illustrate the contagion process. Moreover, new visual analysis techniques for comparison of contagion need to be developed.

We propose a system geared to the visual analysis of contagion. It includes the simulation of contagion effects as well as their visual exploration. We present new tools able to compare the evolution of the different contagion processes. In this way, propagation of disturbances can be effectively analyzed. We focus on financial networks, however, our system can be applied to other use cases as well.

Keywords: Visual Analytics, Network Visualization, Time-Dependent Networks, Financial Networks, Financial Stability, Systemic Relevance

1 Introduction

The outbreak of the global financial crisis in 2007 changed the view on the importance of network effects within the global financial system [1]. A key event in this crisis was the default of a single financial company – Lehman Brothers. The shock wave from this event spread across the globe, generating financial stress and pushing some other banks into default or receivership. Following such a trigger event in a contagion process, the collapse would propagate as a chain reaction throughout the network (see Figure 1). Elements that cause large effects are called systemically relevant [1].

The analysis of contagion effects is important not only for financial stability purposes but also in other application areas such as for the analysis of gene regulatory networks, food webs, infection diffusion or supply-chain networks. Usually, the analysts simulate contagion under application-specific models and then analyze the results [2–5].

![Figure 1: Contagion process in a network. Institutions (nodes) are connected by their dependences. The red nodes collapse.](image)

There is a strong need for tools that help analyze contagion. Visual Analytics tools can be helpful to this end as they
combine interactive graph visualization with graph analysis. For the purpose of analyzing contagion, the visual analysis methods need to be enhanced in several ways:

1. Integrate the simulation of contagion processes for a given model with the visual display of the results.
2. Show information on contagion strength in the system.
3. Convey information on contagion propagation.
4. Allow for a comparison of contagion processes under various simulation scenarios.

In this paper, we focus on the analysis of financial scenarios.

We present a system that provides new techniques for

- Simulation of contagion effects under user-defined conditions using interactive visual interfaces.
- New visualization of the contagion process, which combines animation with a static graph of causal default relationships.
- Visual-analytical methods for the evaluation of contagion effects and the selection of important simulations for a detailed analysis.
- Visual comparison of several contagion effects.

A contagion develops over several steps and thereby creates time-dependent graphs. Our approach thus also constitutes a novel method for the analysis and the comparison of time-dependent graphs.

The paper is structured as follows: Section 2 introduces contagion effects and defines the main tasks and challenges. Section 3 presents related work. Section 4 provides details of our approach. Section 5 shows a use case for our approach. Section 6 discusses the contributions and limitations of our approach as well as the outline of possible future extensions. Section 7 concludes.

2 Analysis of Contagion Effects

In the following, we define and describe the contagion process and the main tasks tackled.

2.1 Financial Networks

A financial system consists of a set of financial institutions (e.g., banks) connected via reciprocal asset/liability positions (e.g., credit, contract payments outstanding). The companies are nodes of the network and their financial strength (e.g., their capital ratio or total assets) is indicated by the node weight (see Figure 1). Financial claims of a bank on another bank form the edges. The edge weight captures the relationship’s strength (e.g., credit amount) and the edge direction indicates its nature. An edge pointing from bank A to bank B denotes that A holds claims on B.

2.2 Contagion Process

The contagion effects in a network resulting from a default are best described by an iterative process [2–5] (see Figure 1). It begins with the default of a financial institution as an initial default node. A company defaults when it does not have enough financial assets, e.g., cash, to redeem its debt coming due. After the default, the entity is removed from the network. As the entity is not able to pay for its debt, its creditors incur losses, i.e., a reduction in their assets, and the debt is removed from the network. In turn, this weakens creditors’ balance sheets. If their financial position is sufficiently weakened, these companies will, in turn, default. This invokes a new iteration of the contagion process. Depending on the financial stability of the institutions in the network, this process may consist of one or more iterations. Serial defaults lead to the collapse of an increasing number of entities.
The contagion process is a *time-dependent graph*, formed by the network states in each iteration. Node weight is non-increasing over iterations (stays stable or reduces). The graph size (number of nodes and edges) is diminishing over time (reduced by the deleted defaulted nodes and the removed edges).

Analysts often simulate contagion effects based on an assumption for a propagation model [2–5], which states how the contagion propagates and when a node collapses (see Section 5.1). Such a simulation allows to evaluate hypothetic contagion effects for policy purposes, e.g., identifying systemically relevant banks.

### 2.3 Analytic Tasks

Financial analysts need to perform a set of tasks when they analyze the stability of financial systems. In order to identify these tasks, we have analyzed the related financial literature, e.g., [2,3,6–9] and discussed with analysts from two banks. The tasks can be categorized according to two criteria: 1) the number of simulations: examining one or comparing several simulations; 2) the target: the final state or the process (see Table 1). It is important to say that these tasks are interwoven.

<table>
<thead>
<tr>
<th></th>
<th>Process</th>
<th>Final State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze one simulation</td>
<td>T1</td>
<td>T3</td>
</tr>
<tr>
<td>Compare multiple simulations</td>
<td>T2</td>
<td>T4</td>
</tr>
</tbody>
</table>

Table 1: Overview of the analytical tasks.

The comparison of several institutions can involve deeper examination of each of the processes and the final state is part of the process (its final stage). So it is important to support these tasks in one system with the possibility to combine the analysis.

**T1 Analyze the contagion process initiated by an institution.**

The analysts wish to examine the process and the causal relationships between the defaults within the process.

**T2 Compare the contagion processes of several institutions or of an institution under different conditions.**

This task enables to analyze whether entities cause similar effects in a process, whether similar institutions fail due to different initial settings, identify whether there are common processes in the system.

**T3 Analyze the final effect of a default.**

This task focuses on the comparison of the final state to the initial state of the financial network, e.g., loss of strength of individual companies and of the whole network.

**T4 Compare the final effects of a default of several institutions or default effects of an institution under different conditions.** This step allows to compare the strength of the effects for individual companies and for the whole network.

It should be noted that our task assessment also showed that the analysis of contagion effects is a lively research area in the financial domain and therefore new tasks may arise in the future.

### 2.4 Requirements and Technical Challenges

Based on the above-mentioned analytical tasks, we define the following objectives for the visual analysis system.

**R1 Support whole analysis process.** There is a need to integrate the simulation and the result examination in one system. The challenge is the design of a new visual-analytics system as the current systems either do not focus on contagion analysis or do not offer the breadth of analytic functions needed for the tasks T1-T4 [10–12].

**R2 Offer visual analysis of contagion process.** The users wish to analyze specific details of the contagion propagation within the network. For this purpose, a new method for time-dependent graph visualization is needed. The
use of animation in graph visualization is limited by human perception capabilities, while static views easily get overcrowded [13, 14]. Moreover, specific contagion information needs to be conveyed (such as the strength and the source of contagion).

R3 Support comparison of contagion processes. The users wish to compare both the contagion process and their total effects from various simulations. The main challenge is the scalability as many simulations need to be compared at the same time (e.g., simulations for all nodes in the network as initial nodes). There is a need to provide a view on the similarities among the steps of the contagion process across the different simulations and across moments in time. However, the current methods either do not consider time-dependence or rely on a manual comparison of graphs [15–17].

R4 Provide information on final effects of a simulation. The users are interested in identifying the strength of the contagion effect. The main challenge is twofold: to define how to measure the effect and to design the visual interface showing the results. This requirement is specific to the visual analysis of contagion and it is therefore not covered in general graph visualization systems [10, 11, 18].

3 Related Work

In this section, we present an overview of visual graph analysis systems and techniques related to our work. We first concentrate on graph visualization systems and their capabilities and, subsequently, on specific areas of visual graph analysis. For a comprehensive overview, we refer to [13].

3.1 Visual Graph Analysis Systems

We review related systems divided into three broad categories: 1) general graph visualization systems, 2) specialized systems and 3) tools used by financial analysts to examine contagion.

1. There are many general graph visualization systems and libraries (e.g., Pajek [19], GraphViz [10], GVS [18], CGV [20], Gephi [11], Tulip [21], JUNG [22], etc.). They concentrate mainly on interactive graph rendering. Many systems also offer analytical functions such as the calculation and the visualization of graph measures (e.g., node centrality). These tools, however, do not offer the possibility to simulate and visualize contagion effects or other network simulation models.

2. There are two specialized systems, which are somewhat related to our approach: Financial Network Analysis (FNA) [23] and Network Workbench Tool (NWT) [12]. FNA [23] concentrates on the analysis of financial networks, in particular on the analysis of financial transactions. It offers a variety of functions for this purpose and recently was extended with simple visualization possibilities. It, however, requires programming skills in order to perform an analysis. FNA can analyze financial networks in order to detect systemically relevant institutions, but the analysis is only performed in two specific network models and a visualization of the result is not provided. Therefore, it does not offer a visual analysis of the contagion process as required by the tasks stated in Section 2.3. NWT [12] offers the possibility to run models of discrete network dynamics, which are popular in biology. The tool offers interfaces for defining model parameters and for the visualization of model results. The visualization, however, does not provide insight into the model process and it does not allow to compare several models. Moreover, it is applicable only to discrete models only, while the use cases tackled in this paper require continuous models.

3. In the financial area, some of the tasks are currently performed using general data analysis softwares such as Matlab. These data analysis softwares require a high level of programming skills and do not provide interactive visual analysis capabilities. The researchers employ simple static network visualizations provided by the software packages.

In summary, there is no comprehensive system suitable for the visual analysis of contagion processes.
3.2 Visualization of Time-Dependent Graphs

As the contagion process is represented by a time-dependent graph, we focus on its visualization using two categories of techniques: animated and static displays.

Animated node-link diagrams require a stable graph layout, which changes minimally over time [24–28]. A stable layout preserves the user’s mental map, i.e., it facilitates following changes on the screen [26, 29]. It thus supports the analysis of graph changes. Although animation is a natural way of conveying data changes, its effectiveness is limited by human perception capabilities. Usually, the user is only able to recognize and remember larger changes in the data. This limitation can be mitigated by highlighting changes in the graph structure between successive points in time [30].

Static views are preferred for a more detailed analysis of data changes. However, these displays are often complex and suffer from overplotting. Several static approaches have been proposed: Saraiya et al. present a technique, where the development of node values over time is shown as a time plot within the node [31]. This approach is restricted to graphs without structural changes (i.e., nodes and edges are constant over time). Brandes and Corman [32] use the third dimension to show the evolution over time, but this rapidly leads to overplotting. Additionally, GraphDice [33] can be used for showing time-varying descriptive graph measures. It shows scatterplots of these descriptive measures, where time can be mapped to one dimension. GraphDice uses interaction to switch between projections. The approach focuses on measure development over time, but not on structural graph changes.

3.3 Visual Graph Comparison

An important analytical task is the examination of graph similarity resulting from the contagion propagation. There are several methods for visual graph comparison, which focus on structural differences [34]. The methods can be divided according to 1) the time-dependence of the graph (static or time-dependent), 2) the number of graphs they can compare (only two or several graphs), and 3) the type of graph (general graph vs. only trees).

Most comparison techniques deal with static graphs. A comparison of two graphs employs either linked views of the corresponding nodes in the two graphs [35, 36], or visually underscores the similar parts of the two graphs. The latter visualization may use either color denoting matching parts of the graphs [37, 38] or it may collapse the identical parts of the graph, thereby emphasizing the differences between the graphs [39].

When analyzing several graphs, the above-mentioned techniques can be used for pairwise comparison leading to \( n \times (n - 1)/2 \) graph pairs. Other methods first calculate graph similarity and then allow to explore the results. Both [16] and [17] use a set of graph measures for determining graph similarity. These measures are either directly shown in a table [16] or are used for visual clustering [17]. Brandes et al. [40] use visual statistics for comparing clustered graphs. Most recently, Bremm et al. [41] presented a new tree similarity measure used in the visual tree comparison as well as for highlighting of the corresponding nodes among trees. This technique, however, cannot be employed for general graphs.

Visual comparison of time dependent graphs was presented by Pohl et al. [14]. They use multiple linked windows, each showing a graph in a selected time step. On demand, for selected nodes, a time line of node properties is shown in an extra window. The analysis of these graphs is cumbersome as it relies solely on manual inspection of all time moments. The graph visualization in multiple windows also requires the identification of node correspondence by the user as the layouts in both windows may differ. Moreover, for several graphs, the view will get cluttered.

4 Approach

We present a system for the integrated analysis of contagion effects. It provides four views (see Figure 2)

1. Simulation interface: Interactive interface for setting and running simulations
2. **Visual analysis of a contagion process**: This view details effects on individual nodes and displays the evolution of the contagion process.

3. **Visual comparison of processes**: This view shows the similarities between the steps of a large number of processes and their evolution.

4. **Interface for analysis and comparison of final effects**: This view provides a most abstract overview of contagion effects based on contagion measures. It offers an interactive selection of interesting processes using these measures.

Depending on the data analysis tasks, the user can follow several analytical paths, which all start with the simulation (1). For example, if the user wished to analyze the contagion emanating in one institution, she can in detail explore the process and the outcome in (2). If she wants to compare effects stemming from various initial defaults, she can either first compare the final effects (4) or compare the processes (3) and then select several simulations for a detailed analysis (2).

The integrated system also offers a feedback loop for calculating and analyzing more/other simulations.

---

**4.1 Definitions for Contagion in Networks**

A graph is a pair \( G = (V, E) \); \( E \subseteq [V]^2 \), where elements of \( V \) are vertices (or nodes) and elements of \( E \) are edges (or links) \( e = (v_1, v_2), e \in E, v_1, v_2 \in V \) [42]. Directed graphs have edges with ordered vertices – the first one is a parent node and the second one is a child node. The graphs have numeric attributes (weights), which are attached to both vertices and edges. The edge attributes are defined by a function \( w^E : E \to \mathbb{R}^+ \) and node attributes by the function \( w^V : V \to \mathbb{R}^+ \). In this paper, the word graph and network is used interchangeably.

Contagion results in the network evolving over time. Such graphs are called *dynamic graphs* (or time-dependent graphs). Based on [15], it is an ordered set of graphs \( GT = (G_{t_0}, \ldots, G_{t_n}) \), where \( (t_0, \ldots, t_n) \in T \) are consecutive time steps (or time moments) and \( G_t \) is a graph \( G \) in step \( t \). In our case, the time-dependent changes affect also the numeric attributes. Therefore the weight function is time-dependent for nodes and edges: \( w(e, t) = w^E(e) \), \( e \in G_t \) and \( w(v, t) = w^V(v) \), \( v \in G_t \).

In the distressed contagion effect scenarios analyzed in this paper, specific time dependent graphs are formed. In these graphs, the set of edges and vertices is stable or diminishing over steps as well as the edge and node weights are stable or decreasing. This is denoted as \( G_{t_1} \subseteq G_{t_2} : V_{t_1} \subseteq V_{t_2} \land E_{t_1} \subseteq E_{t_2} \forall t_1 < t_2 \).

Weights \( w^E(v, t_2) \leq w^E(v, t_1) \land w^E(e, t_2) \leq w^E(e, t_1) \forall t_1 < t_2, v \in GT, e \in GT \)

---

**4.2 Simulation of a Contagion Process**

The simulation entails the calculation of the contagion process, based on a model of contagion. The calculation methodology used in this paper is described in detail in Section 5.1.
The model requires the setting of parameters (e.g., the starting node, the default threshold), which the user inputs in an interactive interface (see Figure 3). It consists of three parts. In the center, an initial representation of the network is shown. It allows the user to select a starting node for the simulation. On the right, the option panel permits the user to set further parameters, such as the default threshold, needed for the simulation. On the left, an overview of past performed simulations is displayed in a list, where each simulation can be revisited. The simulations are color-coded. This color-coding is consistent over all views.

The user may select one or more starting nodes for the simulation. When several nodes are selected, each node serves as an initial node for a simulation. In addition to this manual selection of a starting node, we also provide automatic calculations for all nodes in the graph (i.e. contagion simulations starting in each node of the network without the need to manually select each node separately). In this case, the same calculation settings are used for all nodes.

### 4.3 Visual Analysis of Contagion Process

The analysis of the contagion process is supported by a new view. This view shows the steps in the contagion process with additional information on the strength of the contagion effect. Two versions of this view are offered: 1) a basic view for the analysis of one process and 2) an extended view showing the outcome of several processes.

#### 4.3.1 Contagion Process View

The contagion process in a network creates a time-dependent graph. A natural way of illustrating the evolving state of a graph is animation. However, animation suffers from limited human perception capabilities [43]. By contrast, a static view of an evolving graph can easily get overplotted. Our approach combines the two methods in a new way. We use two linked changed which offer different displays the data. The “Animation View” on the left shows the current state of the graph as an animation and the “Causal View” on the right shows the past steps of the contagion process in a reduced form (it shows only default causalities -the deleted nodes and their connections). The latter view was deemed necessary as our initial experiments showed that the understanding of causal default relationships and the analysis of the sequence of defaulting nodes was very difficult for processes consisting of several steps.

The Animation View shows the contagion process as an animated node-link graph with a special design for nodes and edges, which conveys the information on the state of the contagion. The stability of a node is shown using a bar glyph: a full colored bar denotes a healthy node, while a low value denotes a node close to default (see Figure 4 left). The companies defaulting in a particular step are emphasized by a round highlight. This type of highlight was deemed important as more subtle highlighting (e.g., by fill color or border) was often overlooked by the users during animation.
The impact of a default is shown by highlighting its outgoing edges (i.e., liabilities impacting other institutions in the network). The view is generated with a user-chosen graph layout (e.g., Kamada-Kawai [44]). Importantly, the layout does not change throughout the animation, so that the correspondence of nodes stays stable. The user can replay the process and pause it at interesting steps for more details.

The Causal View shows past steps of the contagion process focusing on the causal relationship between nodes: which node default leads to a default of which subsequent nodes. The default causes are represented by a causality graph where the impacted nodes are children of the initial defaulted nodes (see Figure 4 right). During animation, this causal graph is automatically built from top to bottom, with the top node being the initial default. The recursive default process creates several levels of default causalities. The number of levels in the graph corresponds to the duration of the causal process. In this view, the analyst can examine which institutions induce a default of other entities and in which order.

4.3.2 Extended Contagion Process View

The Extended Animation View allows to compare the stability of nodes across several simulations. This view uses color coding to distinguish simulations (see Figure 5a). However, in this view, it may be challenging to compare several node values at the same time.

The Extended Causal View shows several simulations simultaneously (see Figure 5b). An important aspect is to determine which nodes are affected in several simulations and at which step in the process. The number processes in which a node defaults is displayed by color, with a darker coloring suggesting a larger number of defaults. Node correspondence is indicated by node border and connection lines.

Both these views are very good for comparing a small number of processes in detail. However, for larger data sets, these views may get overplotted. Therefore, we developed an alternative compact view for the comparison of a larger number of processes, which is explained in the next section.

4.4 Comparison of Contagion Processes

A comparison of contagion processes entails a comparison of time-dependent graphs. The main challenge hereby lies in ensuring scalability (number of time steps, number of simulations, graph size). One way of dealing with the scalability
Figure 5: a) Extended Animation View. The colors denote simulations. Bar loading means vulnerability to default. A full colored bar means a healthy node. b) Extended Causal View. Each graph shows past causal relationships. Node color denotes the number of times an institution defaulted (darker is more). The node correspondence is shown on demand by highlighting (thick black border and the connecting red lines).

issue is filtering the number of simulations. This is implemented by an interactive filtering of simulations by their total effects (see Section 4.5). However, this could still leave many contagion processes to be analyzed. Alternatively, the user may wish to examine all simulations.

We propose a new approach for the comparison of several simulations ("Process Comparison View"). With this view, we address the main tasks of contagion process comparison: identification of similarities among process steps in a manifold of simulations.

More specifically, the view allows:

1. The identification of similarities/differences among graphs in various points in time: For example, the view makes it possible to see that one step of a simulation (a graph $G^1_i$ in time moment $t_i$) is structurally similar another moment in time of another simulation (i.e., another graph $G^2_j$ in time moment $t_j$).

2. The analysis of the graph similarity over time: The user can recognize whether simulation steps (i.e., graphs) become more similar or less similar over time.

Note that these aspects were not covered in previous studies (see Section 3.3).

We represent each simulation $s$ (time-dependent graph $GT^s = \{G^s_i\}, i = 0, ..., \tau^s$) as a sequence of points (nodes), where each point represents one graph $G^s_i$ in the simulation. The points are connected by directed edges in time order $G^s_i \rightarrow G^s_{i+1}$. This allows the user to follow the progress of contagion (see Figure 6). This compact data representation also addresses the scalability problem when comparing several dynamic graphs with many nodes and edges. For a better readability, the time moment is shown as a label. The simulations being compared are distinguished by color (see Figure 7 left). In this regard, it should be noted that the same colors are used across all views for consistency and easier user perception.

The positions of points on the screen indicate the similarity of steps in the contagion processes (i.e., close points represent similar process steps across simulations). We use the multi-dimensional scaling algorithm (MDS) [45] for the calculation of point positions (see Figure 7). In this way, we can see that two graphs (yellow and blue) behave similarly throughout the simulation, however at different steps. At the same time, the green and the red simulation stay very similar to both the yellow and the blue simulation, but finish earlier and in different final states. The turquoise simulation is similar to the blue simulation in the first step and then differs significantly. This novel representation allows for a
Visual Analysis of Contagion in Networks

Figure 6: The idea of contagion comparison. Each node corresponds to one simulation step. Lines connect steps.

Figure 7: Process comparison view. Left: Compact view on simulation steps. Color denotes simulations. Right: Compared simulation steps using MDS layout.

The MDS-based layout depends strongly on the definition of graph similarity used. There are two main ways of defining graph similarity: edit-based graph distance [46] and feature-based graph distance [17]. The edit-based graph distance calculates the similarity by the number of graph edit actions needed in order to transform one graph into the other graph. This requires a correspondence of the two graphs being compared, which is in our case directly given by node labels and their adjacent edges. The feature-based graph distance calculates the similarity of two graphs on the basis of a set of metrics such as the number of nodes, the number of edges, or the eccentricity. This approach is more suitable to graphs without labels, as two graphs with the same structure but different node labels would be taken as identical. This, in our case, would be disadvantageous and therefore we employ the edit-based graph distance approach.

Our implementation of the edit-based graph distance reflects the similarity of two graphs by their common elements (relative number of the same edges and vertices). In this way, we assume that two contagion effect graphs are similar if the same elements remain after default. The graphs that we compare have a direct correspondence by the node labels and their adjacent edges.

We calculate the distance as follows:

$$\text{dist}(G_{s1}^k, G_{s2}^k) = \alpha \cdot \frac{\sum_{v \in V(G)} d(V)}{|V(G)|} + (1.0 - \alpha) \cdot \frac{\sum_{e \in E(G)} d(E)}{|E(G)|},$$

$$d(V)(v) = 0.0, \forall v_1, v_2, \forall v_1 \in G_{s1}^k \land v_2 \notin G_{s2}^k, d(V)(v) = 1.0, \forall v_1, v_2, v_1 \in G_{s1}^k \land v_2 \notin G_{s2}^k \lor v_1 \notin G_{s1}^k \land v_2 \in G_{s2}^k, d(V)(v) = \frac{|w(v_1) - w(v_2)|}{w(v_0)}, \forall v_1, v_2, v_1 \in G_{s1}^k \land v_2 \in G_{s2}^k, v_0 \in G_0,$$

where $v_i$ are corresponding nodes in the compared simulation steps. $d(E)(e)$ is calculated in an analogue way.

$\alpha$ is a user-defined parameter, where $0 \leq \alpha \leq 1$. In our experiments, we found out that $\alpha = 0.5$ provides good results.

Axes of the MDS layout generally do not have a direct meaning, however, in some cases they can be interpreted by experts. As shown in Section 5, the possible meaning of the X axis in our case is the strength of the contagion step (i.e., its impact on system stability) and the Y axis corresponds to the impact on the individual system structure (contagion steps far from each other on the Y axis contain different nodes and edges).

Analysis of similarity among contagion steps is further enhanced by the possibility to group two or more contagion steps (i.e., the similarity-based merging). The steps are automatically merged if graph similarity is larger then a user-
defined threshold (see Fig. 9). The important advantage of this method is that similar steps are identified in a comparison across the simulations and across the moments of time. Thus, for example high similarity among simulation A in time moment $t_i$ with simulation B in time moment $t_j$ and simulation C in time moment $t_k$ can automatically be detected. Note that these groups can also be built manually, if the user wants to compare specific steps in the contagion process.

The group of similar steps is shown as a merged node colored in black (see Figure 8). The number of merged nodes is shown in the label and the exact group members are seen in tooltips.

Labels of black nodes denote the number of simulation steps that were merged. Tooltip shows the merged simulation steps, where color means simulation and the number depicts the simulation step. The blue simulation in step 4 is similar to the yellow simulation in step 6. The two simulation steps are compared in detail showing their high structural similarity. Similar nodes defaulted Contagion steps in selected groups can be compared in detail (see Figure 8 right). The view shows a table with all graph nodes (rows) and all selected simulation steps (columns). The table is sorted according to the node weights. The node color shows the similarity of a node in graph $G_j$ to a reference graph $G_i$ shown at most left. Missing cells denote defaulted institutions. Figure 8 right shows an example of the detailed view for two simulation steps: the yellow simulation in step 6 and the blue simulation in step 4), whereby the rows are graph nodes. We see that the yellow and blue simulations are very similar in steps 6 and 4, although one node has not defaulted yet in the blue simulation. The red color

Figure 9: Comparison of many simulations. Left: Without merging. Right: With merging.

The similarity-based merging also allows for a simplification of the view, which is especially useful when analyzing many points in time or many simulations. As shown in Figure 9, the merged view can show an overview of similarities among simulations of a large number of simulations. For this purpose, the simulations are color-coded. As for a large
number of simulations, the color differences may be difficult to distinguish, we also offer tooltips showing the simulation details (the simulation name and the step number). In Figure 9, 50 simulations are compared. These simulations were calculated for all nodes in the network as initial nodes using the same threshold. The merging identifies groups of similar simulation steps and similar final states. It also reveals major “contagion process ways” (black paths) and outliers (e.g., the orange and the green paths).

4.5 Analysis and Comparison of Total Effects

The analysis of the strength of the total effects of a contagion caused by a default of a node is important for assessing the importance of a node in the network. The stronger the destructive effect the more important the process starting in this node is. For assessing the total effect, we propose a set of measures, that capture various aspects of the contagion (see Section 4.5.1). Based on these measures, several contagion effects can be compared and the most important nodes can be identified for a detailed comparison in the Contagion Comparison and the Contagion Process Views (see Section 4.5.2).

4.5.1 Analysis of Total Effects

We propose the following set of measures in order to examine the strength of the contagion process. The set includes new measures (M4, M5) complementing the network measures used in [3].

We provide several measures which cover various different analytical aspects. The measures are normalized to the interval [0,1] for comparability reasons. A value close to 1.0 identifies a very relevant simulation. Based on these measures, systemically relevant institutions can be identified (i.e., those with significant impact on the network).

The proposed measures are:

M1 Duration of contagion process: The higher the value of M1 the longer lasting is the impact of an initial default. We measure it as follows: 

\[ m_1 = \frac{\tau}{|V(G_\tau)|} \]

where \( \tau \) is the number of steps of the contagion process. It is normalized by the maximum possible duration of a contagion process, which assumes that one node defaults in each step until all nodes default.

M2 Number of defaulted institutions: This measure focuses on the extent of defaults in the system. The higher the value of M2, the larger is the number of defaulted institutions. The measure is defined as:

\[ m_2 = 1.0 - \frac{|V(G_\tau)|}{|V(G_0)|} \]

M3 Asset loss: This measure focuses on the financial system strength after the contagion process. The higher the value of asset loss, the weaker is the system after the contagion. The measure is calculated as follows:

\[ m_3 = 1.0 - \frac{\sum_{v \in V(G_\tau)} w(v)}{\sum_{v \in V(G_0)} w(v)} \]

where \( w(v) \) is the balance sheet value (i.e., total assets) of an institution \( v \).

M4 Loss of connectivity: Well functioning financial systems are often highly connected, as financial transactions are performed among many participants in the systems. By contrast, labile systems depend on a few major players. Therefore, we propose to measure the loss of the financial network stability due to contagion using the connectivity measure M4 as:

\[ m_4 = 1.0 - \frac{|E(G_\tau)|}{|E(G_0)|} \]

M5 Vulnerability to default: This measure describes, how stable the nodes are in the final state after contagion. It reflects, whether the institutions are financially weakened (close to the default threshold \( h \)) or have not been influenced at all. The measure M5 is calculated as follows:

\[ m_5 = 1.0 - \frac{\sum_{v \in V(G_\tau)} h(v, \tau)}{|V(G_\tau)|} \]

where \( h(v, \tau) = \frac{w(v, \tau)}{w(v, 0)} \). Note that we assume all institutions to be 100% stable at the beginning and to have a weight of 0.0 when defaulted.

For gaining knowledge about the state of the financial system after a process of contagion, these measures should be analyzed in combination. For instance, there can be a long-lasting default process [M1] with a low number of defaulting institutions [M2], or a short process with a large number of defaulting institutions.
4.5.2 Comparison of Total Effects

The above mentioned measures can be used for both the comparison of the final states of many simulations and for the selection of important simulations for further detailed comparison within the views presented in Sections 4.3 and 4.4).

We propose to use a combination of the above-mentioned criteria for the comparison of total effects, resulting in a weighted linear combination of the measures. The weights are user-specified in order to address the current user’s analytical focus. The weights are defined in an interactive interface using a set of sliders (see Figure 10 top). Please note that the sum of the weights is automatically adjusted to 1.0).

![Figure 10: Interactive selection of the interesting simulations. Top: Weighting of measures for the relevance function. Bottom: Histogram for interactive setting of the filtering threshold.](image)

The selection of interesting simulations is supported by an interactive histogram view with an interactive filter threshold (see Figure 10 bottom). The histogram shows the distribution of the simulations by their importance and allows the user to interactively adjust the filter threshold according to the data. In this way, the users can make informed selections of most relevant simulations. The selected simulations are highlighted in the list of all performed simulations on the left.

5 Use Case

We show how our system is applied to analyze the stability of a financial system. We first explain the mathematics of contagion, then describe the underlying data set and finally present an illustrative scenario analysis as developed by a financial expert.

5.1 Calculation of a Contagion Process

There are several models of contagion propagation in financial networks, e.g., [2, 5]. We use the model of Chan-Lau et al. [2] from the Bank of International Settlements, which assumes that a bank defaults if the ratio of its assets after defaulting of another bank to its original assets is less than a model threshold. We use this model as it is meaningful and has been used before [47]). For other applications or analytical scenarios, the contagion propagation model can be modified.

We employ a simplified balance sheet of a financial institution as a starting point. The balance sheet of an institution \( v_i \) is composed of the sum of incoming transactions \( \sum w^E(in(v_i)) \) and its capital \( c(v_i) \) (see Eq. 1). Sum of the balance sheet components corresponds to institution’s total assets/liabilities \( w(v_i) \).

\[
w(v_i) = \sum w^E(in(v_i)) + c(v_i)
\]

As described in Section 2.2, the contagion starts with a default of an initial node \( v_0 \) in the initial network \( G \). Afterwards, all the transactions of the institution \( v_0 \) are deleted from the network and the balance sheets of all dependent
institutions \( \{ v_i \} \) are adjusted according to the Eq. 2. An institution \( v_i \) defaults as a consequence of the default of \( v_0 \), if its balance sheet decreases by more than a certain threshold \( h \), where \( 0 \leq h < 1 \) (see Eq. 3). If at least one institution defaults, a next round of default starts. The process finishes, when no institution defaults.

\[
\begin{align*}
    w(v_i)^* & \leftarrow w(v_i) - \mathcal{W}(e(v_i, v_0)), \forall v_i \in \text{in}(v_0), v_i \neq v_0 \\
    w(v_i)^* & \leq h \cdot w(v_i)
\end{align*}
\] (2) (3)

5.2 Data

As supervisory data necessary to run real world simulations are highly confidential, we simulate a prototypical banking system. We have implemented several graph generators as mentioned in the literature [5, 48, 49], as banking systems across countries differ substantially [50]. We would like to emphasize that our aim is to show the application of our system to a financial analysis process not to perform a perfect simulation of economic reality.

![Figure 11: The financial network used in our use case.](image)

In this use case, we employ Moussa-Cont generator for financial graph structure [5]. The parameters of the generator were set according to the example described in that work. We generate edge and node weights based on a random distribution, where we assume that highly connected banks are larger and have higher asset/liability positions than smaller banks. The network consists of 40 banks (see Figure 11). We see that, by construction, large banks are highly connected in the center of the graph (FraB, UHB, TruB, SavB and PosB).

We run simulations for all banks using a threshold of 0.92, because banks need to hold a capital buffer equal to 8% of risk-weighted assets according to the financial regulation.

5.3 Example Analytical Process

We present an example for the analytical process, which was developed together with a financial expert from a European bank. The analyst first looks at the distribution of the total effects according to various criteria (see Figure 12). She can see that there are many banks with little impact and several banks with a large impact. Among the banks with a large impact are the big banks mentioned before. Interestingly, there are also some small banks (e.g., CreB, SocB, OreB), which are rather on the “periphery” of the banking system. In this view, the analyst filters banks with a large impact for further analysis.
She then compares the contagion processes of all banks with high impact (see Figure 13). The MDS layout together with automatic merging reveals several contagion paths and groups of similar contagion steps. As expected, all processes start in the same point. Moreover, they end in one point, meaning that they lead to similar final results. However, the paths between the start and the end of these processes differ: For example, the large banks UHB and SavB have very different paths from other large banks FraB and TruB as well as from PosB. The analyst is then interested in examining the details of these processes. What happens in detail? Why are they (dis)similar?

The analyst now studies the Process Analysis View for the details of the contagion processes for two small banks (OreB in blue and CreB in green) (see Figure 14). The view shows that at the beginning of the process, the two contagions behave very differently. First, different banks default (see the causal graphs on the right) and, second, different parts of the network are impacted (see the highlighted green and blue links in the network on the left). The view also reveals an interesting finding that in this step, both processes lead to the collapse of UHB bank (black circle). The expert continues the analysis with the examination of the next steps of the two processes (see Figure 15). Moreover, the default of UHB causes many banks to collapse directly afterwards. So UHB is a kind of “multiplier”. It therefore seems that the default of UHB is crucial to the financial system. The highlighting function enables the analyst to identify that UHB causes in both cases the default of the same banks. So all these banks are directly threatened by a default of UHB.

Figure 16 shows the final state of the two contagion processes. This view reveals that both processes lead to default of similar banks, although in a different number of process steps. It also leads to interesting insights. First, two small banks (CreB and SocB, white color) default only in the CreB process, and, second, ComB and SweB collapse at last in both contagion processes.

6 Discussion

We next turn to a discussion of the strengths and weaknesses of our approach with regard to the three aspects: 1) scalability of our approach, 2) the support of the comparison tasks and 3) extendability to other financial analysis scenarios.
Figure 14: Details for the contagion step 2 for two small banks (OreB in blue and CreB in green). We see that they have different impacts up to this step.

Figure 15: Details for the contagion step 4 for two small banks (OreB in blue and CreB in green). The collapse of UHB leads to a kind of “multiplicator” effect, which is the same in both cases (see the highlighted nodes connected by red lines.)
Scalability  The scalability of our approach can be viewed from two perspectives: 1) the graph size (number of nodes and edges) and 2) the number of performed simulations.

1. Graph size influences the visualization possibilities and also the number of simulations (e.g., simulating contagion for all nodes). Visualization of large graphs in the animation view may lead to clutter. One possibility would be to filter the view only for the most relevant nodes. These nodes can be identified using the function for the degree of interest presented in Section 4.5. This extension is planned in the future.

2. A large number of simulations poses two challenges: the need for substantial computational time and the need to compare many simulations. The aspect of large computational time can be addressed by the calculation of simulations in parallel. The aspect relating to the comparison of many simulations is tackled by the possibility to select only the most relevant simulations for a detailed analysis.

Comparison  Our approach provides various possibilities for comparing simulations (the Extended Process View, the Process Comparison View and the comparison of total effects).

- The Extended Process View offers the possibility to compare contagions in detail. It however relies on manual inspection of the data, which the users deemed possibly cumbersome for many simulations.

- The Process Comparison View can currently show dozens of simulations. The view can get cluttered when comparing a large number long-lasting simulations. We address this issue with the automatic merging of similar simulation steps and showing details for user-selected simulation steps. However, this may not scale up for many simulations (hundreds or thousands).

- The comparison of total effects allows for the filtering of the most interesting simulations for a detailed inspection in the previously mentioned views. This enhances the scalability of our approach. This notwithstanding, there may be cases when the filtering still leads to an extensive number of simulations needing to be presented.

Financial Analysis  Our approach offers a wide variety of functions supporting the tasks presented in Section 2.3. It could, however, still be extended in various ways. First, our work is based on the model of Chan-Lau et al. [2]. As an extension, we could include other models and support their comparison. This could lead to the need to extend our system. In particular, simulation calculation and visualization could be affected (e.g., by new parameters and additional information to be shown to the user). Second, the users suggested the possibility to mark one bank as systemically relevant and analyze what would happen if this bank was “saved”, i.e. it would not be allowed to collapse.
7 Conclusions and Future Work

In this paper, we proposed a new approach for the visual analysis of contagion effects. Our approach provides the users with the possibility to interactively simulate, analyze and compare contagion processes. For this purpose, we developed several interactive visualizations and analytical measures. Our new approach for comparison of simulations can also be used for comparing time-dependent graphs. It allows to automatically identify similar steps across simulations and time moments.

The presented system can be adapted to similar tasks investigating the stability of networks in other domains such as the analysis of infection propagation, gene regulation or supply-chain networks.

In the future, we would like to improve our system in several ways. We would like to support the analysis of causal relationships with a large number of simulations. We intend to improve the scalability of process visualization with filters and by showing the most relevant nodes, followed by an interactive expansion of the graph depending on user preferences. Additionally, we intend to provide the user with the possibility to compare simulation results using various simulation models.

Acknowledgements

We would like to thank B. B. from [anonymous1] Bank and J. L. from the [anonymous2] Bank for their insights into financial analysis, and helpful comments on the prototype, scenario and the paper. This work has been partially supported by German BMBF within Software Campus 2013.

References

Visual Analysis of Contagion in Networks


