

Interactive Exploration and Understanding of Contagion Dynamics in Networked Populations

Sherif E. Abdelhamid, Chris J. Kuhlman, Madhav V. Marathe

Biocomplexity Institute and Department of Computer Science

Virginia Tech

Email: {sherief}, {ckuhlman}, {mmarathe}@vbi.vt.edu

S. S. Ravi

Department of Computer Science

University at Albany—SUNY

Email: sravi@albany.edu

Abstract—Modeling and simulation of contagion processes on networked populations are used to understand protests, social unrest, the spread of information, and virus and disease epidemics, among other phenomena. Network structure and attributes of vertices and edges are often useful in explaining contagion spreading processes. However, particularly for larger networks (e.g., those with hundreds of thousands or millions of vertices), reasoning about and making sense of contagion propagation results is difficult owing to the scale of these simulations. We present a web application called NEMO for assisting an analyst in understanding contagion processes and in establishing causality. It has several features to query and visualize networks, subnetworks, and their properties. In addition to explaining NEMO’s features, we provide a real case study of the spread of Ebola on a 4-million-vertex social network of Liberia, Africa. We demonstrate how NEMO can be used to explore interactively networks to understand the reasons for the effectiveness of different interventions.

Keywords—interactive sense-making; knowledge discovery; web application; modeling; contagion dynamics

I. INTRODUCTION

A. Background and Motivation

Computing the spread of contagions on populations is useful for understanding how local behaviors give rise to global or population-level outcomes. Examples include person-to-person transmission of Ebola and how it produces large numbers of outbreaks [1]. Other contagions, such as Twitter tweet hashtag propagation in Spain in 2011, provide insights into people’s reactions to government austerity measures [2]. Information diffusion and contagious influence can lead to collective action [3] and social movements of the type well-studied in sociology (e.g., [4]). Models of contagion dynamics can also guide how to control undesirable contagions through interventions [5].

Contagion studies are often performed on network representations of populations, where vertices and edges of a network represent agents (e.g., people) and pairwise interactions, respectively. Network structure is known to have a large effect on the contagion dynamics [2]. Thus, many intervention methods are based on network structure; e.g., removing vertices or edges from a graph to inhibit contagion spreading [5], [6]. Many of these studies (e.g., [1], [5]) use agent-based modeling (ABM) and simulation.

Despite the large number of ABM studies of contagion dynamics on networks, we know of no tools available to interactively probe results to assist in understanding computed dynamics (e.g., why particular outcomes are generated). In effect, our goal is to establish causality in the results.

There are several considerations for understanding contagion propagation on social networks. Almost always, simulations involve multiple runs (i.e., diffusion instances) to account for stochasticity, for example. Parametric studies are also quite common, in order to understand sensitivities of results to various inputs, which may involve initial conditions, interventions, model parameters, and combinations thereof. These realities make it imperative that explorations of simulation data be *interactive* and *iterative* because one may have to explore successively different “threads” in order to achieve an understanding.

There are significant differences between our goals and those for network visualization. Our focus here is contagion dynamics *on* networks, and not the analysis of dynamic (e.g., time-varying) networks. Also, the majority of agent-based contagion models are based on local behavior; i.e., an agent changes state based on local interactions with its nearest neighbors. Consequently, it is important to mine simulation results for these local properties. Thus, large-scale network visualizations such as community structure are often insufficient to understand dynamics; e.g., contagions can propagate between communities. Furthermore, while it is possible to pre-compute many network-based measures a priori, this is not the case for contagion dynamics, where interactive probing may be driven by initial conditions, model parameters, and other dynamical considerations. As one example, if one “seeds” (i.e., initially assigns contagion to) 20 agents of a one million agent system, there are $C(10^6, 20) \gg 10^{50}$ possible seed sets. Different sets will produce different dynamics and require different interpretations of simulation data. There is no possibility of pre-computing all of these results.

In the past, we have produced special-purpose scripts to post-process particular simulation results data, which is time-consuming, not conducive to iterative exploration, and not helpful for domain experts with no computing background.

In this work, we introduce and illustrate the use of a web application called NEMO that enables domain experts (e.g., network scientists, social scientists) to reason about the

results of contagion simulations on social networks in terms of network properties. These properties can be (i) domain-specific attributes, such as the age and gender of humans that are represented by network vertices or weights of interactions that are represented by edges; and (ii) network structure measures, such as degree and clustering coefficient distributions. NEMO uses a combination of data mining, computational analyses, and scientific data and network visualizations to enable knowledge discovery about networks that helps in understanding computed contagion dynamics.

B. Contributions

A summary of our main contributions follows.

1. NEMO functionality for understanding contagion diffusion on large networked populations. NEMO is a web application with the following features: uploading attributed networks into the system; computing structural characteristics and measures of networks; browsing networks and their properties; querying networks to return sets of vertices and edges, and other information; visualizing networks and their attributes, computed data (e.g., measures), and data returned from queries; and sharing of data across users. We describe these features in more detail below. The system is designed for use by researchers, domain scientists, policy planners, and others with no computing expertise. It is an extensible system; e.g., it can readily incorporate new network measures.

2. Description of NEMO in the context of contagion dynamics. NEMO's purpose is to enable one to understand contagion diffusion on networks, through knowledge discovery about network properties using scientific computation and visual analytics. It is a distinct web application that can be run in isolation. Nonetheless, it provides a set of services that can be profitably used in conjunction with other external systems. One example is the EDISON web application [7] that computes the diffusion of a contagion on networks using discrete-time agent-based simulation. We discuss this interoperability in Section IV.

3. Case study of the Ebola Outbreak in Liberia, Africa in 2014. We provide a detailed case study to illustrate the ability of NEMO to store and process large attributed graphs. (The social contact network of Liberia has 4.08 million people and 85 million daily interactions.) Results from multiple simulations are presented, most of which evaluate different interventions to halt the spread of Ebola. The purpose of this case study is to understand why some interventions are more effective than others. The goal is to establish causality between intervention inputs and contagion spread sizes. We describe how NEMO is used to iteratively and interactively explore different network properties and attributes in order to understand contagion spreading.

II. RELATED WORK

Gephi [8] is an open source tool for visualizing networks and properties of networks. We use Gephi as a service within NEMO to render graphs and provide information about them. However, Gephi does not have the

capabilities of NEMO for understanding contagion dynamics. There are many other graph visualization tools; see, for example, <http://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html>.

SocialNetSense [9] is designed to help analysts understand domain-based attributes and structural parameters of networks. Their work appears to be confined to networks with perhaps a few thousand vertices, whereas we seek to understand dynamics on networks that are 2 to 3 orders of magnitude greater in size. Other software uses machine learning [10] and logic programming [11] to understand graphs.

Network Workbench (NWB) (e.g., [12]) is a tool for visualizing networks for a wide range of domains. It can investigate dynamics to some extent, but the tool is not used to explore and query networks, and to visualize networks and their properties, in order to explain contagion spreading on networks.

Contagion in networks is visualized in [13]. This work differs greatly from ours in that their definition of contagion is the process of removing vertices from graphs, not the spread of contagions like information on a network. Also, they primarily study small financial networks (roughly 40 vertices). Furthermore, they are not concerned with explaining dynamics, only showing the dynamics (in terms of how a graph is reduced in size—for their type of contagion).

Analysis tools to understand brain network data are called for in [14], where some of the stated needs are not only structural, but also related to brain functioning (i.e., dynamics).

The SNAP system [15] provides a number of graph measures, but not within a system that performs agent-based simulation.

Most tools (beyond those identified here) appear to be focused on smaller graphs.

III. DISCRETE DYNAMICAL SYSTEMS MODEL

We provide an overview of discrete dynamical systems, which is our framework for computing contagion dynamics. We then present a concrete example, which illustrates some aspects of agent-based simulation. The final subsection expands on the first two, to motivate the need for a tool like NEMO to understand contagion dynamics.

A. Formulation

Our goal is to provide intuition about the discrete dynamical systems that we use for simulation. More detailed treatments are found in [16], [17]. A **graph dynamical system** (GDS) \mathcal{S} is a 4-tuple $\mathcal{S}(G(V, E), K, F, W)$, where $G(V, E)$ is the **social network** of the population, with vertex set V and edge set E ; let $n = |V|$. Agents and interactions are represented by, respectively, vertices and edges. Each vertex $i \in V$ is assigned precisely one value from the **set of vertex states** K , at each time t of a simulation. In the example below, we will take $K = \{0, 1\}$, where 0 (resp., 1) means that a vertex is not (resp., is) activated. The state x_i of vertex i at t is denoted $x_i(t)$. Let F be the set of all **vertex functions** or **local transition functions** f_i , for each $i \in V$. That is, the

next state of i is computed using $f_i \in F$. Specifically, given the states of all vertices at t , the next state of i is given by $x_i(t+1) = f_i(x[i](t))$, where $x[i](t)$ is the sequence of states of i and its (distance-1) neighbors at t . For example, in the 6-vertex network in Figure 1, if $i = 5$, then vertices 4 and 6 are its distance-1 neighbors and $x[5](t) = (x_4(t), x_5(t), x_6(t))$; i.e., it is the sequence of states of vertices 4, 5, and 6 at t . Finally, W is the **update scheme**; it describes the order in which the vertex functions f_i , $1 \leq i \leq n$ are executed. Here, we use a **synchronous** update method, where all f_i execute simultaneously at each time step t . Such a GDS is often called a **synchronous dynamical system** (SyDS) [17].

B. Illustration of Contagion Dynamics

We now provide an example, illustrated in Figure 1. On the left, the state transition diagram for a vertex is shown. The vertex function is a **threshold** function, where each vertex j is assigned a threshold θ_j . In words, f_j at time t is defined as follows: (i) if j is in state 0, and at least θ_j of its neighbors are in state 1, then j transitions from state 0 to 1 (i.e., $x_j(t+1) = 1$); otherwise, $x_j(t+1) = 0$; and (ii) if j is in state 1, then its next state is 1. In this example, all vertices have $\theta = 2$, except for vertex 5, where $\theta_5 = 3$.

At $t = 0$, we specify that vertices 2 and 6 (seed nodes) are in state 1, and the system state $x = (x_1, x_2, x_3, x_4, x_5, x_6) = (0, 1, 0, 0, 0, 1)$. At $t = 1$, vertex 4 has two neighbors in state 1, namely 2 and 6, so it transitions to 1; no other vertex transitions. At $t = 2$, vertex 1 has two neighbors in state 1 (2 and 4), and hence transitions to state 1. Note that vertex 5 does not transition because its threshold is 3; it can never transition from 0 to 1 because it does not have 3 neighbors. Finally, at $t = 3$, vertex 3 changes state. This results in a **fixed point**; i.e., no more vertex state transitions can occur.

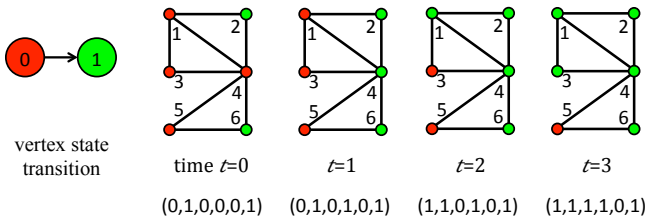


Fig. 1. An example GDS. Left: the single vertex state transition $0 \rightarrow 1$. Right: the 6-vertex graph has initially two vertices (2 and 6) in state 1 at time $t = 0$. These are seed nodes. Each vertex has threshold $\theta = 2$, except vertex 5, for which $\theta_5 = 3$. The system state $x = (x_1, x_2, x_3, x_4, x_5, x_6)$ at times $t = 0$ through 3 is given. At the end of $t = 3$, the system has reached a fixed point; i.e., no vertex can change state.

C. Extensions of the Basic GDS Formulation

The dynamics of the intentionally simple example of Section III-B belie the power of GDS, which is Turing equivalent for appropriate complexity classes [18]. There are many directions of increased sophistication of a GDS from the example above; we cite four. First, networks can be any size (e.g., any number of vertices), often reaching tens of thousands or millions of vertices. Second, the vertex functions can be much

more sophisticated, such as one for the spread of Ebola [19], which has six vertex states, eight different state transitions, and requires 11 properties for each agent and one property for each edge. Third, the interactions can be complex. For example, there may be multiple edges between any pair of vertices in Figure 1, each representing a different mechanism such as face-to-face, social media, email, and cell phone interactions. Fourth, there may be different types of agents in the system, each with a different type of vertex function. These complications further motivate the need for tools to understand contagion simulation results.

IV. ILLUSTRATIVE LARGER SYSTEM OVERVIEW

Our focus of this paper is on NEMO and its capabilities from a user's perspective. However, it is useful to put NEMO in context to see its utility in a larger scope. This description illustrates, in a concrete way, the value of NEMO.

Figure 2 provides an over-arching view of our system for computational modeling and agent-based simulation (ABS) of contagion processes on network representations of populations, as overviewed in Section III. We refer to this system as EDISON (shown in the figure as one of the components), and has been described in [7]. EDISON is a web application and is intended for computing novices and experts alike. Illustrative applications include the spread of influenza, and of social contagions such as information, rumors, and addiction.

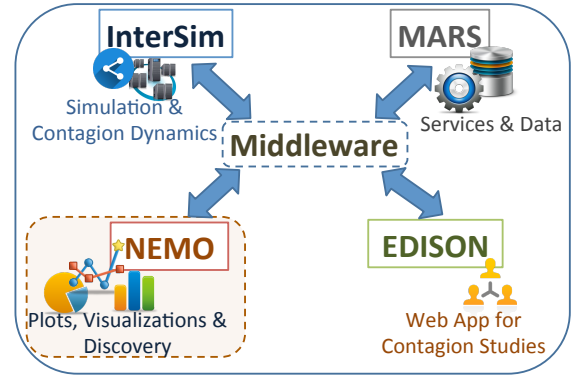


Fig. 2. Overarching view of a software capability of which NEMO is a component. These components are distinct, and several can be integrated with systems other than the EDISON system, identified here. Our focus in this paper is the highlighted NEMO system for interactive knowledge discovery and visualization.

Each of the four components is distinct, and can interoperate with other software systems. This is an example of our approach to design and implement pluggable, loosely-coupled components that can operate within service oriented architectures (SOAs). The middleware is key for communications, passing metadata, data, authentication, and service requests, including quality of service (QoS) specifications.

InterSim is the agent-based simulation framework that performs the contagion dynamics computations; an earlier version is described in [20]. MARS [21] is a set of services

that manipulates networks and their attributes, and computes structural properties of networks, among other things. NEMO has features that are completely distinct from those of the other components, although it does use the MARS query service, which has been enhanced for the needs of NEMO.

V. NEMO OVERVIEW

Figure 3 summarizes features available to users through NEMO’s user interface. A user can upload networks into the system, and specify structural parameters (e.g., number of connected components) and distributions (e.g., degree distribution) to be computed. Users can browse networks, including their attributes. Attributes may be domain-based, such as the ages of network vertices that represent people and duration of contact on an edge between two vertices, or structural parameters such as k-cores or betweenness centrality. Users can perform queries on these networks, returning sets of vertices or edges. These data may be plotted. Other types of plots include network visualizations and data associated with them.

Data can be shared among users. For example, a network uploaded by one user can be used by another; e.g., to generate plots of properties. These users may be remote collaborators.

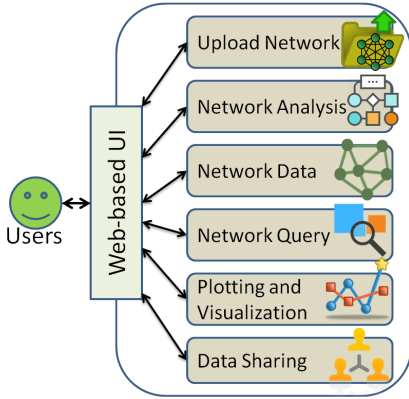


Fig. 3. User functionality through NEMO.

VI. CASE STUDY: EBOLA OUTBREAK IN LIBERIA, AFRICA IN 2014

We now describe an actual case study based on an ABS of the Ebola outbreak in Liberia, Africa in 2014. Different interventions are evaluated by determining their effects on computed outbreak sizes. The main goal of this description is to demonstrate how NEMO can be used to understand contagion dynamics in terms of network structure and attributes. We first discuss the social contact network of Liberia. Then we summarize the simulations and results. Finally, we describe the use of NEMO.

A. Social Contact Network of Liberia

For ABS of the Ebola outbreak, a social *contact* network is required. The social network of Liberia consists of human agents or vertices, and edges between pairs of humans that

indicate contact in the sense of being co-located for a prescribed time interval. Each person is endowed with traits such as age, gender, home location, size and composition of the person’s household, and a set of daily activities with locations and start and end times. The contact network represents human interactions in a single, normal day. In this network, activities are primarily home, work, and school, but any number and type of activities can be included. Each edge is labeled with the duration of contact based on start and end times, the activities of the two people interacting, and the IDs of these people. The social network that we use is produced from a baseline population consisting of 4.08M attributed people and 84.8M labeled edges. Attributed vertices and edges of the network are entered into MARS through the NEMO web portal.

B. Agent-Based Simulations and Results

Simulations were completed with EDISON [7]. The social contact network described above was used in all simulations. The state transmission model describing the progression of illness in a human is provided in Figure 4, and comes from [19]. A healthy person that has never contracted Ebola is susceptible. Exposed means that the person has contracted Ebola, but is not yet contagious. The infectious state means that the person is out interacting with the population, performing her normal activities, but is infectious. Hospitalized individuals and even the deceased (in the funeral state) remain infectious but do not have activities. Removed means either recovered (with subsequent immunity) or deceased and taken out of the population. Note that Figure 4 reflects cultural aspects of select African countries because of hospital- and funeral-related customs [19]. Equations for disease transmission are omitted due to their complexity, but follow the model described in [19].

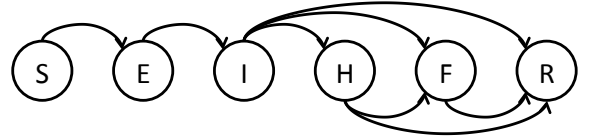


Fig. 4. Finite state machine showing the permissible states for each human agent and its state transitions for Ebola [19]. States are susceptible (S), exposed (E), infectious (I), hospitalized (H), in funeral state after death (F), and removed or recovered (R).

Simulations were performed by seeding five randomly chosen people per run and computing transmission among agents per day, for 400 days. Final fractions of infected people over 400 simulated days are plotted in Figure 5 for various conditions. Each bar represents the average of 20 individual simulation instances. The base results correspond to about 9500 infections over this period with no interventions. Additional simulations were performed to investigate interventions. These consist of vaccinating people with 70% efficacy, thus reducing the baseline disease transmissibility τ to τ' for vaccinated people, according to $\tau' = (1 - 0.7)\tau = 0.3\tau$. Vaccinated people are correspondingly less likely to be infected. Different simulations were used to vaccinate people in 10-year intervals, as depicted in Figure 5. The results show that vaccinating those

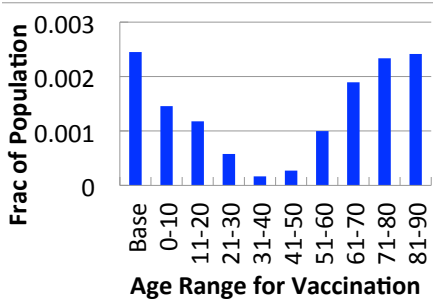


Fig. 5. Agent-based simulation results for the spread of Ebola in Liberia, Africa. Fraction of total population infected after 400 days, averaged over 20 contagion instances, for the baseline case (of no intervention) and the nine interventions; from [7].

in the 31-40 year age range was most effective. The purpose of NEMO, in this case, is to aid an analyst in understanding why some interventions are more effective than others, which is where our focus now turns.

C. Understanding the Effects of Interventions Using NEMO

Figure 6 conveys the operations of NEMO for the purpose of understanding why some interventions are more effective than others in Figure 5. The analyses described here represent one session with NEMO; i.e., a user logs into the system and completes all analyses in one sitting (although these operations may be broken up among multiple sessions).

First, it is reasonable to suspect that there are simply more people in the network in the 31-40 age range, so vaccinating these people produces the greatest vaccine coverage. To test this hypothesis, the user constructs a workflow within NEMO using point-and-click operations. The workflow counts the number of social network vertices that are in each age range, and constructs and displays the bar chart shown by workflow 1 (the orange circle with “1” inside), which contains the green bars. From this plot, we see that approximately 0.5 million people are in the 31-40 age range, but that other groups have far more people.

These data do not provide an explanation, so the analyst constructs a second workflow, denoted by the orange circle with the “2” inside, in Figure 6. This workflow generates the accompanying plot and is motivated by the idea that people in the 31-40 age range may have more total interactions than those in other age ranges. But the plot of sum-of-degrees by age bin shows that, again, this is not the case. This age group has slightly more than 10 million interactions, but there are two other groups with more contacts.

Workflow 3 is used to determine whether the average number of interactions per person in each age range explains the effects of interventions. In this case, the 31-40 and 41-50 age ranges have the greatest average degrees, but only slightly more than the bins 21-30 and 51-60. These four bins represent the most effective age ranges for interventions in Figure 5. While informative, the average degrees among these four bins do not explain the differences in the efficacies of the interventions.

Workflows 4 and 5 are used to compute k -shell distributions, and for each k -shell, the number of vertices in the shell for the different age bins is computed. A k -core is a subgraph of a graph in which all vertices have degree at least k . A k -shell is the set of vertices in the k -core subgraph, and not in the $(k+1)$ -core subgraph. For each age bin, the more vertices that are in greater shells, the more well-connected those vertices are to other high-degree vertices. Workflow 4 produces the counts of network vertices in each k -shell, by age bin, for the five age groups that are least effective in stopping the outbreak when vaccinated, and generates the plot shown. Workflow 5 generates analogous data for the four age groups that are most effective to vaccinate and the plot generated in this workflow is also shown. From this latter plot, we see that for some of the greatest shells, around $k = 30$ (see the red arrow), the ordering by age bin for the number of vertices in shells 30 through 35, exactly matches the ranking of the efficacy of vaccination. The largest shell in the Liberia network is 37, so these shells are large. This suggests that the vertices in the social network with high degree that are most well-connected to other high degree vertices, are most effective in reducing the Ebola spread.

After workflow 5, the analyst terminates her session. Note that for all plots, the user can customize the plotted ranges on the x- and y-axes, the axis labels and all font sizes, and the colors and types of lines for each data line. The goal is to enable a user to generate publication-quality graphics while interactively exploring the data.

This case study does not prove that vaccinating high k -shell vertices are most effective in thwarting Ebola outbreaks. Typically, the best one can do is use inductive reasoning: given the evidence (i.e., simulation results), identify likely causes that produce the result. Additional simulations, perhaps on other networks, could be used to test this hypothesis.

This study also illustrates that one must look in detail at data. High-level graphics, such as a network visualization with vertices colored by community, does not yield useful information, although identifying the vertices in communities that are most connected to other communities could in some cases. Social networks with high-degree hub nodes often produce communities of very large size, which are not helpful in understanding how contagion is transmitted. Other possibilities are computing the durations of contact for the vertices of different age bins (not shown), since the duration of contact affects the probability of disease transmission.

VII. CONCLUSIONS AND FUTURE WORK

We have described a new web application called NEMO for interactive and iterative knowledge discovery. In particular, the primary purpose of NEMO is to assist domain experts and analysts in understanding the causes of observed contagion dynamics phenomena. Iterative and interactive use of NEMO facilitates data exploration. The system is designed for extensibility, and new features are under development.

Acknowledgments. We thank the anonymous reviewers for their useful comments and suggestions. We thank our external

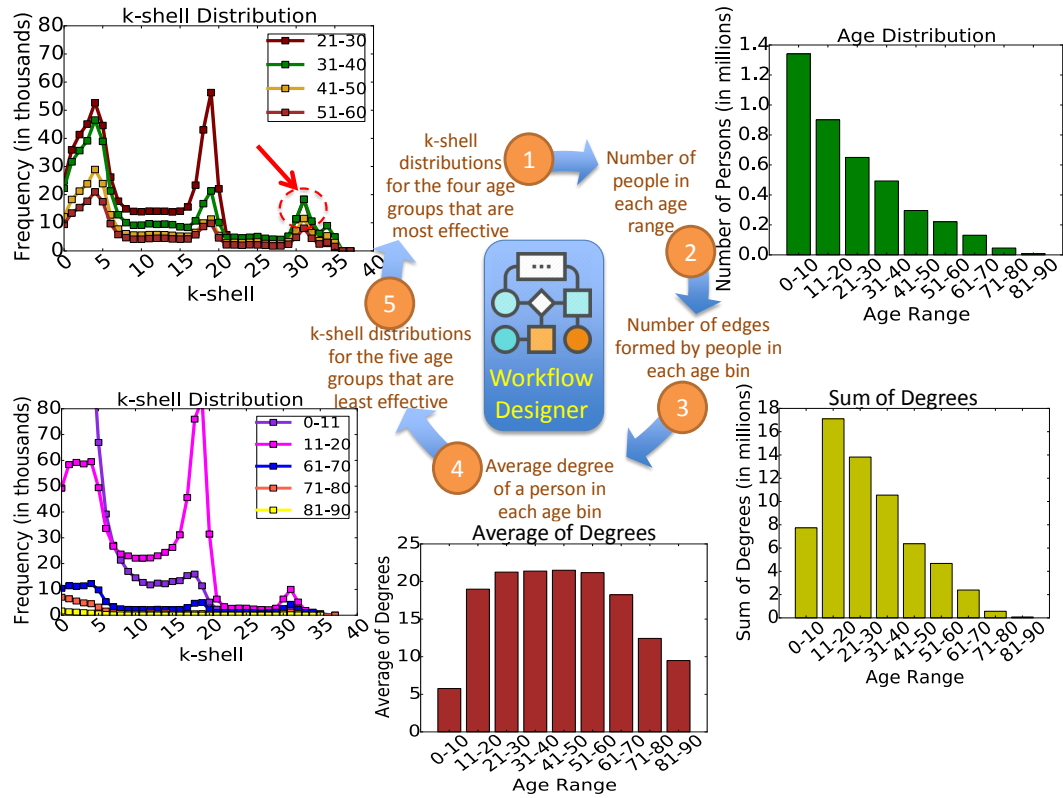


Fig. 6. Iterative and interactive use of NEMO to generate different plots on-the-fly through the web console, in order to understand the effects of interventions on simulated Ebola outbreaks in Liberia, Africa. Given the simulation results in Figure 5, the goal is to understand the reasons for the effectiveness of different interventions. The data here were generated during one user session with NEMO. Each plot is the result of a different workflow that is specified within NEMO by a user. Between workflows, the human-in-the-loop evaluates the results and decides on the next analysis (workflow) to be completed, if any. Here, the results of five analyses (workflows labeled 1 to 5) are provided. The utility of these results is described in the text.

collaborators and members of the Network Dynamics and Simulation Science Laboratory (NDSSL). This work was partially supported by DTRA Grant HDTRA1-11-1-0016, DTRA CNIMS Contract HDTRA1-11-D-0016-0001, and NSF NetSE Grant CNS-1011769.

REFERENCES

- [1] C. Siettos, C. Anastassopoulou *et al.*, “Modeling the 2014 ebola virus epidemic—agent-based simulations, temporal analysis and future predictions for liberia and sierra leone,” *PLOS Currents Outbreaks*, 2015.
- [2] S. Gonzalez-Bailon, J. Borge-Holthoefer, A. Rivero, and Y. Moreno, “The dynamics of protest recruitment through an online network,” *Nature Scientific Reports*, pp. 1–7, 2011.
- [3] S. Lohmann, “Collective action cascades: An informational rationale for the power in numbers,” *Journal of Economic Surveys*, 2000.
- [4] M. Diani, “Introduction: Social Movements, Contentious Actions, and Social Networks: ‘From Metaphor to Substance’?” in *Comparative Politics: Social Movements and Networks*. Oxford, 2003, pp. 1–20.
- [5] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, “Gelling, and Melting, Large Graphs by Edge Manipulation,” in *Proc. CIKM*, 2012, pp. 245–254.
- [6] R. Albert, H. Jeong, and A. Barabasi, “Error and attack tolerance of complex networks,” *Nature*, 2000.
- [7] S. E. M. Abdelhamid, C. J. Kuhlman, G. Korkmaz, M. V. Marathe, and S. Ravi, “Edison: a web application for computational health informatics at scale,” in *ACM-BCB*, 2015, pp. 413–422.
- [8] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: An open source software for exploring and manipulating networks,” 2009.
- [9] L. Gou, X. Zhang, A. Luo, and P. F. Anderson, “Socialnetsense: Supporting sensemaking of social and structural features in networks with interactive visualization,” in *VAST*, 2012, pp. 133–142.
- [10] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos, “Apolo: Making sense of large network data by combining rich user interaction and machine learning,” in *SIGCHI*, 2011, pp. 167–176.
- [11] P. Shakarian, G. I. Simari, and D. Callahan, “Reasoning about complex networks: A logic programming approach,” in *TPLP*, vol. 13, 2013.
- [12] V. Colizza, R. Pastor-Satorras, and A. Vespignani, “Reaction–diffusion processes and metapopulation models in heterogeneous networks,” *Nature Physics*, vol. 3, no. 4, pp. 276–282, 2007.
- [13] T. von Landesberger, S. Diel, S. Bremm, and D. W. Fellner, “Visual analysis of contagion in networks,” *Information Visualization*, 2013.
- [14] O. Sorns, “Making sense of brain network data,” *Nature Methods*, 2013.
- [15] J. Leskovec, “SNAP: Stanford network analysis project,” <http://snap.stanford.edu/>, 2016.
- [16] H. Mortveit and C. Reidys, *An introduction to sequential dynamical systems*. Springer Science & Business Media, 2007.
- [17] D. Rosenkrantz, M. Marathe, H. H. III, S. S. Ravi, and R. Stearns, “Analysis problems for graphical dynamical systems: A unified approach through graph predicates,” in *AAMAS*, 2015.
- [18] C. Barrett, H. B. Hunt III *et al.*, “Complexity of reachability problems for finite discrete dynamical systems,” *J. of Comp. and Syst. Sci.*, vol. 72, pp. 1317–1345, 2006.
- [19] J. Legrand, R. F. Grais, P. Y. Boelle, A. J. Valleron, and A. Flahault, “Understanding the dynamics of ebola epidemics,” *Epidemiology and Infection*, vol. 135, 2007.
- [20] C. J. Kuhlman, V. S. A. Kumar *et al.*, “A general-purpose graph dynamical system modeling framework,” in *WSC*, 2011.
- [21] S. Abdelhamid, C. J. Kuhlman, M. V. Marathe, and S. S. Ravi, “Network services and their compositions for network science applications,” in *ICCS*, 2016.