Blocking the Propagation of Two Simultaneous Contagions over Networks

Henry L. Carscadden, Chris J. Kuhlman, Madhav V. Marathe,
S. S. Ravi, and Daniel J. Rosenkrantz

[1] Henry L. Carscadden: University of Virginia. hlc5v@virginia.edu.
[2] Chris J. Kuhlman: University of Virginia. cjk8gx@virginia.edu.
[3] Madhav V. Marathe: University of Virginia. marathe@virginia.edu.
[4] S. S. Ravi, Daniel J. Rosenkrantz: University of Virginia and University at Albany – SUNY. {ssravi0, drosenkrantz}@gmail.com.

Abstract We consider the simultaneous propagation of two contagions over a social network. We assume a threshold model for the propagation of the two contagions and use the formal framework of discrete dynamical systems. In particular, we study an optimization problem where the goal is to minimize the total number of infected nodes subject to a budget constraint on the total number of nodes that can be vaccinated. While this problem has been considered in the literature for a single contagion, our work considers the simultaneous propagation of two contagions. Since the optimization problem is **NP**-hard, we develop a heuristic based on a generalization of the set cover problem. Using experiments on three real-world networks, we compare the performance of the heuristic with some baseline methods.

1 Introduction

Contagion models have been used to explain a host of observed phenomena in human populations (e.g., the spread of diseases, fads, opinions, information, actions such as joining a group) [8, 16, 19]. In this paper, we treat contagions as undesirable entities (such as infectious diseases) propagating through a network. Network models of contagion propagation capture complex patterns of interaction missed by models that assume homogeneous mixing. These interactions present interesting combinatorial optimization problems such as seed selection and contagion blocking. Our focus in this paper is on blocking. Previous work on blocking focuses on the case where only a single contagion is propagating through a network (see, e.g., [5, 11] and the references cited therein). We seek to extend prior work from the single contagion setting to the multiple contagion setting. To understand the landscape of the area, we consider two independent contagions propagating under the threshold model [9]. Under this model, an individual (i.e., node in a social network) gets infected because it has at least a sufficient number (called the **threshold**) of infected neighbors. In addition to disease propagation, threshold models [4, 9, 17, 21] have also been used to capture other social contagions (such as information, opinion and fads). In this paper, we consider disease propagation and use vaccinating nodes as the blocking strategy. The goal is to reduce the number of newly infected nodes under a budget on the number of nodes that can be vaccinated. Following [11], we use the synchronous dynamical system (SyDS) as the formal model for contagion propagation; see Section 2.

Summary of Results: We discuss a general threshold-based model for the simultaneous propagation of two contagions through a network. As this general model (which requires the specification of five threshold values for each node) is somewhat complex, we consider a simplified model that uses only two threshold values for each node. Using that model, we formulate the problem of minimizing the number of new infections in a network by vaccinating some nodes. In practice, there is a budget constraint on the number of vaccinations. We observe that the resulting budget-constrained optimization problem is computationally intractable using a known result for the case of a single contagion [11]. Therefore, we develop an efficient heuristic algorithm called MCICH for the problem. This heuristic is based on a generalized version of the Minimum Set Cover (MSC) problem [7]. Through computational experiments, we compare the performance of MCICH with two baseline methods using three real-world social networks. Our results indicate that MCICH is able to block the two contagions effectively even with a small vaccination budget, and performs far better than the other two methods.

Related Work: Reference [11] treats the single contagion blocking problem under the threshold model. The goal is again to minimize the number of new infections subject to a budget on the number of nodes that can be vaccinated. It is shown that if the budget cannot be violated, even obtaining an approximation algorithm with any provable performance guarantee is **NP**-hard. Two efficient heuristics for the problem are introduced and their performance is evaluated on several social networks. Although single contagion epidemic models have been studied for years, study of the multiple contagion context is newer. For example, conditions for the coexistence of two contagions in compartmental models are explored in [3]. A number of references (see e.g., [10,14,15] and the references cited therein) have considered the propagation of competing contagions (where infection by one contagion prevents or reduces the likelihood of infection by another), and cooperating contagions (where infection by one contagion makes it easier to get infected by another contagion). While our work uses the deterministic threshold model, reference [18] discusses a general framework for a probabilistic multiple-contagion model, namely the Susceptible-Infected-Recovered (SIR) model.

2 Definitions and Analytical Results

Model Description: We use the **synchronous dynamical system** (SyDS) model studied in the literature (see e.g., [2]). A (SyDS) $\mathbb{S}$ over a domain $\mathbb{B}$ is specified as a pair $\mathbb{S} = (G, \mathbb{F})$, where (a) $G(V,E)$, an undirected graph with $|V| = n$, represents the underlying graph of the SyDS, with node set V and edge set E, and (b) $\mathbb{F} = \{f_1, f_2, \ldots, f_n\}$ is a collection of functions in the system, with f_i denoting the **local function** associated with node v_i, $1 \leq i \leq n$. Each node of G has a state value from $\mathbb{B}$. Each function f_i specifies the local interaction between node v_i and its neighbors in G. The inputs to function f_i are the state of v_i and those of the neighbors of v_i in G; function f_i maps each combination of inputs to a value in $\mathbb{B}$. This value becomes the next state of node v_i. It is assumed that each local function can be computed efficiently.

For a single contagion, the domain $\mathbb{B}$ is usually chosen as $\{0,1\}$, with 0 and 1 representing that a node is uninfected and infected respectively. Since we have two contagions (denoted by $\mathbb{C}_1$ and $\mathbb{C}_2$) propagating through the underlying network, we have four possible states for each node, denoted by 0, 1, 2 and 3; thus, $\mathbb{B} = \{0, 1, 2, 3\}$.

The interpretation of these state values is shown in Table 1. An easy way to think of these states is to consider the 2-bit binary expansion of the state values 0 through 3. The least (most) significant bit indicates whether the node has been infected by $\mathbb{C}_1$ ($\mathbb{C}_2$).

Table 1: Possible states for each node

State	Interpretation
0	Not infected by either $\mathbb{C}_1$ or $\mathbb{C}_2$
1	Infected by $\mathbb{C}_1$ only
2	Infected by $\mathbb{C}_2$ only
3	Infected by both $\mathbb{C}_1$ and $\mathbb{C}_2$

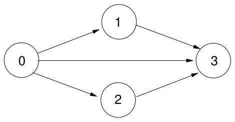


Fig. 1: Possible state transitions for each node

We assume that the system is **progressive** with respect to each of the contagions [6]; that is, once a node is infected by a contagion, it remains infected by that contagion. Using this assumption, Figure 1 shows the possible state transitions for each node.

State transition rules: Each node v is associated with a **local transition function** f_v that determines the next state of v given its current state and the states of the neighbors of v. Such a function may be deterministic or stochastic (as in SIR systems). Here, we will consider a simple class of deterministic functions called **threshold functions**.

A general form of local functions: We first discuss a very general (but somewhat complex) form of local functions for the propagation of two contagions in a network and then present a simpler form that will be used in the paper. In the general form, for each node v and each of the five possible state transition x to y (shown in Figure 1), there is a threshold value $\theta(v,x,y)$. Let $N(v,j)$ denote the number of neighbors of v in state j, $0 \leq j \leq 3$. (If the state of node v is j, then v is included in the count $N(v,j)$.) For any node v, the rules for each possible state transition which collectively specify the local function f_v are shown in Table 2.

Table 2: Transition rules to specify the local function f_v

Transition	Condition
$0 \longrightarrow 1$	$(N(v,1)+N(v,3) \geq \theta(v,0,1))$ and $(N(v,2)+N(v,3) < \theta(v,0,2))$
$0 \longrightarrow 2$	$(N(v,1)+N(v,3) < \theta(v,0,1))$ and $(N(v,2)+N(v,3) \geq \theta(v,0,2))$
$0 \longrightarrow 3$	$(N(v,1)+N(v,3) \geq \theta(v,0,1))$ and $(N(v,2)+N(v,3) \geq \theta(v,0,2))$
$1 \longrightarrow 3$	$N(v,2)+N(v,3) \geq \theta(v,1,3)$
$2 \longrightarrow 3$	$N(v,1)+N(v,3) \geq \theta(v,2,3)$

We briefly explain two of the state transition conditions shown in Table 2. The conditions for other state transitions are similar. Consider the condition for the "$0 \longrightarrow 1$" transition. For this transition to occur at a node v, the number of neighbors of v in state 1 or state 3 must be at least $\theta(v,0,1)$ (i.e., $(N(v,1)+N(v,3) \geq \theta(v,0,1))$) and the number of neighbors of v in state 2 or state 3 must be *less than* $\theta(v,0,2)$ (i.e., $(N(v,2)+N(v,3) < \theta(v,0,2))$). Likewise, for the "$1 \longrightarrow 3$" transition to occur at v, the number of neighbors of v in state 2 or state 3 must be at least $\theta(v,1,3)$ (i.e., $(N(v,2)+N(v,3) \geq \theta(v,1,3))$). At any state $j \in \{0,1,2,3\}$, if none of the conditions for transitions out of j hold, the node remains in state j.

The above general model is powerful as it allows the two contagions to interact. Many references have considered cooperating and competing contagions (e.g., [10, 12, 15]). For example, in the case of cooperating contagions, if a node has already contracted $\mathbb{C}_1$, it may be easier for it to contract $\mathbb{C}_2$. This can be modeled by choosing a low value for $\theta(v,1,3)$. However, the model is also complex since it requires the specification of five threshold values for each node. In this paper, we consider a simpler model which uses only two threshold values for each node.

A simpler form of local functions: In the general form discussed above, each node was associated with five threshold values, one corresponding to each of the five transitions shown in Figure 1. In the simpler model, for each node v, we use two threshold values, denoted by $\theta(v,1)$ and $\theta(v,2)$. The parameter $\theta(v,1)$ is used when v is in state 0 or state 2 (i.e., has not contracted contagion $\mathbb{C}_1$); it specifies the minimum number of neighbors of v whose state is either 1 or 3 so that v can contract contagion $\mathbb{C}_1$. Similarly, $\theta(v,2)$ is used when v is in state 0 or 1, and it specifies the minimum number of neighbors of v whose state is either 2 or 3 so that v can contract contagion $\mathbb{C}_2$. Unlike the general model, the simpler model does not permit other interactions between the two contagions. However, the simpler model facilitates the development of analytical and experimental results.

Additional definitions concerning SyDSs: At any time τ, the **configuration** $\mathscr{C}$ of a SyDS is the n-vector $(s_1^\tau, s_2^\tau, \ldots, s_n^\tau)$, where $s_i^\tau \in \mathbb{B}$ is the state of node v_i at time τ $(1 \leq i \leq n)$. Given a configuration $\mathscr{C}$, the state of a node v in $\mathscr{C}$ is denoted by $\mathscr{C}(v)$. As mentioned earlier, in a SyDS, all nodes compute and update their next state *synchronously*. Other update disciplines (e.g., sequential updates) have also been considered in the literature (e.g., [2, 13]). Suppose a given SyDS transitions in one step from a configuration $\mathscr{C}'$ to a configuration $\mathscr{C}$. Then we say that $\mathscr{C}$ is the **successor** of $\mathscr{C}'$, and $\mathscr{C}'$ is a **predecessor** of $\mathscr{C}$. Since the SyDSs considered in this paper are deterministic, each configuration has a *unique* successor. However, a configuration may have zero or more predecessors. A configuration $\mathscr{C}$ which is its own successor is called a **fixed point**. Thus, when a SyDS reaches a fixed point, no further state changes occur at any node.

Example: The underlying network of a SyDS in which two contagions are propagating under the simpler model discussed above is shown in Figure 2. For each node v, the two threshold values $\theta(v,1)$ and $\theta(v,2)$ are both chosen as 1. Suppose the initial states of nodes a, b, c and d are 1, 2, 0 and 0 respectively; that is, the initial configuration of the system is $(1,2,0,0)$. The local function f_a at a is computed as follows. Since a is in state 1, we need to check if it can contract contagion $\mathbb{C}_2$. Since $\theta(a,2) = 1$ and a has a neighbor (namely b) in state 2, a can indeed contract contagion $\mathbb{C}_2$. Therefore, the value of the local function f_a is 3; that is, the next state of a is 3. In a similar manner, it can be seen that the local functions f_b and f_c (at nodes b and c respectively) also evaluate to 3. For node d, whose current state is 0, there is one neighbor (namely, b) whose state is 2. Therefore, the local function f_d at d evaluates to 2. Thus, the configuration of the system at time 1 is $(3,3,3,2)$. Since the system is progressive, the states of nodes a, b and c will continue to be 3 in subsequent time steps. However,

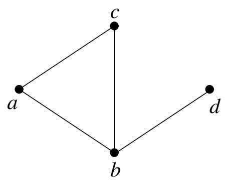


Fig. 2: The underlying network of a SyDS with two contagions. For each node v, both the threshold values are 1.

the state of node d changes to 3 at time step 2 since d has a neighbor (namely, b) whose state at time step 1 is 3. Thus, the configuration of the system at the end of time step 2 is $(3,3,3,3)$. In other words, the sequence of configurations at times 0, 1 and 2 of the system is:

$$(1,2,0,0) \longrightarrow (3,3,3,2) \longrightarrow (3,3,3,3)$$

Once the system reaches the configuration $(3,3,3,3)$, no further state changes can occur. Thus, the configuration $(3,3,3,3)$ is a fixed point for the system.

In this example, the SyDS reached a fixed point. Using our assumption that the system is progressive, one can show that every such SyDS reaches a fixed point.

Proposition 1. *Every progressive SyDS under the two contagion model reaches s fixed point from every initial configuration.*

Proof: Consider any progressive SyDS on $\mathbb{B} = \{0,1,2,3\}$. Let n denote the number of nodes in the underlying graph of the SyDS. In any transition from a configuration to a different configuration, at least one node changes state. Because the system is progressive, each node may change state at most twice: once from 0 to 1 (or 0 to 2) and then from 1 to 3 (or 2 to 3). Thus, after at most $2n$ transitions where the states of one or more nodes change, there can be no further state changes. In other words, the system reaches a fixed point after at most $2n$ transitions. ■

Problem Formulation: The focus of this paper is on a method for containing the propagation of two simultaneous contagions by appropriately vaccinating a subset of nodes. Before defining the problem formally, we state the assumptions used in our formulation.

Following [11], we assume that only those nodes that are initially uninfected by either contagion (i.e., nodes whose initial state is 0) can be vaccinated for $\mathbb{C}_1$ and/or $\mathbb{C}_2$. When a node is vaccinated for a certain contagion, the node cannot get infected by that contagion; as a consequence, such a node cannot propagate the corresponding contagion. For $i = 1,2$, one can think of vaccinating a node v for a contagion $\mathbb{C}_i$ as increasing the threshold $\theta(v,i)$ of the node v to degree(v)+1 so that the number of neighbors of v that are infected by $\mathbb{C}_i$ will always be *less than* $\theta(v,i)$. If a node v is vaccinated for both $\mathbb{C}_1$ and $\mathbb{C}_2$, then it plays no role in propagating either contagion. In such a situation, one can think of the effect of vaccination as removing node v and all the edges incident on v from the network.

The optimization problem studied in this paper is a generalization of a problem studied in [11] for a single contagion. This problem deals with choosing a small set of nodes to vaccinate so that the total number of resulting new infections when the system reaches a fixed point is a minimum. Given a set C of nodes to be vaccinated, a **vaccination scheme** specifies for each node $w \in C$, whether w is vaccinated against $\mathbb{C}_1$, $\mathbb{C}_2$ or both. The total number of vaccinations used by a vaccination scheme for a set of nodes C is the sum $N_1 + N_2$, where N_i is the number of nodes vaccinated against $\mathbb{C}_i$, $i = 1,2$. Note that if a node w is vaccinated against both $\mathbb{C}_1$ and $\mathbb{C}_2$, then it is included in both N_1 and N_2. Also, after a vaccination scheme is chosen and the contagions spread through a network, the number of new infections is measured as the total number of state transitions, because each state transition means a node acquires a new contagion. A formal statement of this optimization problem is as follows.

Vaccination Scheme to Minimize the Total Number of New Infections (VS-MTNNI)
Given: A social network represented by the SyDS $\mathbb{S} = (G, \mathbb{F})$ over $\mathbb{B} = \{0, 1, 2, 3\}$, with each local function $f_v \in \mathbb{F}$ at node v represented by two threshold values $\theta(v,1)$ and $\theta(v,2)$; the set I of **seed** nodes which are initially infected (i.e., the state of each node in I is from $\{1,2,3\}$); an upper bound β on the total number of vaccinations.
Requirement: A set $C \subseteq V - I$ of nodes to be vaccinated and a vaccination scheme for C so that (i) the total number of vaccinations is at most β and (ii) among all subsets of $V - I$ which can be vaccinated to satisfy (i), the set C and the chosen vaccination scheme lead to the smallest number of newly infected nodes.

It is straightforward to show that VS-MTNNI is computationally intractable. To do this, we state a problem and a result from [11].

Smallest Critical Set to Minimize the number of Newly Affected nodes (SCS-MNA)
Given: A SyDS represented by a graph $G(V,E)$ through which a single contagion is propagating, a threshold value $\theta(v)$ for each node v, a set $I \subseteq V$ of initially infected nodes, a vaccination budget β and an upper bound Q on the number of new infections.
Requirement: A subset $C \subseteq V$ such that $|C| \leq \beta$ and after vaccinating the nodes in C, the number of new infections in G is at most Q.

The following result is from [11].

Theorem 1. *The* SCS-MNA *problem is **NP**-hard even when each threshold value is 2. Further, if the vaccination budget cannot be violated, the problem cannot be approximated*[5] *to within any factor* $\rho \geq 1$*, unless* ***P** = **NP***. ■

It is easy to show that the result of Theorem 1 also holds for the VS-MTNNI problem.

Proposition 2. *The* VS-MTNNI *problem is **NP**-hard even when each threshold value is 2. Further, if the vaccination budget cannot be violated, the problem cannot be approximated to within any factor* $\rho \geq 1$*, unless* ***P** = **NP***.

Proof: The SCS-MNA problem can be easily reduced to the VS-MTNNI problem as follows. Let an instance of SCS-MNA be given by a graph $G(V,E)$, a subset $I \subseteq V$ of initially infected nodes (by the only contagion), a vaccination budget β and an upper bound Q on the number of new infections. From the graph $G(V,E)$ of the SCS-MNA instance, we create a new graph $G'(V',E)$ by adding a new node v to V such that v has no edges incident on it. In the VS-MTNNI instance, the initial state of each node in I is chosen as 1 and the initial state of the new node v is chosen as 2. The two threshold values for each node in G' are chosen as 2. It is now easy to see that only $\mathbb{C}_1$ can spread in the SyDS represented by G'. Therefore any vaccination scheme for G' which vaccinates at most β that causes at most Q new infections is also a solution to the SCS-MNA instance, and vice versa. ■

[5] An algorithm for the SCS-MNA problem provides a factor ρ approximation if for every instance of the problem, the number of new infections is at most ρQ^*, where Q^* is the minimum number of new infections.

Proposition 2 points out that in the worst-case, even obtaining an efficient approximation algorithm with a provable performance guarantee for the VS-MTNNI problem is computationally intractable. Therefore, we now focus on designing a heuristic algorithm that works well in practice. This heuristic relies on a known approximation algorithm for a generalized version of the Set Cover problem, called the Set Multicover problem [20]. In this problem, we are given a universal set $U = \{u_1, u_2, \ldots, u_n\}$ of elements, a collection $C^* = \{C_1, C_2, \ldots, C_m\}$ of subsets of U, an integer coverage requirement $r_i \geq 1$ for each $u_i \in U$, $1 \leq i \leq n$, a budget $\beta \leq m$. The goal is to find a subcollection $C' \subseteq C^*$ such that $|C'| \leq \beta$ and for each $u_i \in U$, the number of sets in C' that contain u_i is at least r_i, $1 \leq i \leq n$. When $r_i = 1$, $1 \leq i \leq n$, then we have the usual Set Cover problem [7]. An iterative greedy heuristic (which in each iteration picks a set which covers the largest number of elements whose coverage requirement has not yet been met) is known to provide a performance guarantee of $O(\log n)$ for the Set Multicover problem [20]. As discussed in Section 3 this heuristic is useful in developing our heuristic for the VS-MTNNI problem.

3 Experimental Results

In this section, we provide the networks tested; descriptions of the key elements of the analysis process—simulation and the contagion blocking heuristics (including the new MCICH); a summary of the overall analysis steps; and results of the contagion blocking numerical experiments. Throughout this section, we use the words “activated” and “infected” as synonyms, and also “block” and “vaccinate” as synonyms.

Networks: The three networks of Table 3 are evaluated. We use only the giant components from the networks.

Table 3: Networks used in experiments, and selected properties. All properties are for the giant component of each graph. These properties were computed using the `net.science` system [1].

Network	Num. Nodes	Num. Edges	Ave. Degree	Ave. Clust. Coeff.	Diameter
Astroph	17,903	196,972	22.0	0.633	14
FB-Politicians	5,908	41,706	14.1	0.385	14
Wiki	7,115	100,762	28.3	0.141	7

Simulation Process: A **simulation** consists of a set of iterations. Each **iteration** consists of software execution of contagion propagation from a **seed set** I, where seed nodes states are 1, 2, or 3. The total number of seed nodes is 20 in all iterations, and are chosen from the 20-core of each graph. (The 20-core of a graph G is the subgraph of G in which every node has a degree of at least 20 [6].) Each of the seed nodes has a probability of 1/3 of being set to each of states 1, 2, and 3. (All iterations were also done with 10 seed nodes, but results are not reported here.) An iteration starts at $t = 0$ with the seed nodes as the only activated nodes. From these nodes, contagion propagates in discrete times $t \in [1 \,..\, t_{max}]$ as described for the SyDS in Section 2. All state transitions, x to y, are recorded for all $v \in V$. In this work, all iterations within one simulations use *uniform* thresholds for all nodes and all state transitions, so we abbreviate the thresholds below by setting $\theta = \theta(v,1) = \theta(v,2)$. In this work, we run 10 iterations per simulation, where the differences among the iterations is the composition of the seed node sets. Simulations are run with and without blocking nodes.

Blocking Heuristics: We present three methods (heuristics) for blocking a contagion. The first two are well studied, and serve as baselines for comparison. The third method is the covering heuristic MCICH that is a contribution of this work. For a simulation involving two distinct contagions, the corresponding method is applied for each contagion individually.

Random heuristic. For a given budget β_i on the number of blocking nodes for contagion $\mathbb{C}_i$, select β_i nodes from among all nodes, uniformly at random.

High degree heuristic. For a given budget β_i on the number of blocking nodes for contagion $\mathbb{C}_i$, select the β_i nodes with the greatest degrees (break ties arbitrarily).

New Multi-Contagion Independent Covering Heuristic (MCICH). We devise a set cover heuristic to identify a subset of nodes that are activated at time t, to set as blocking nodes, such that no nodes will activate at time $t+1$. If this is accomplished, then the contagion is halted at t, and our goal is achieved.

A key idea is that any node v_i that is activated at time $t+1$ does so because it receives influence from nodes activated at time t, for otherwise, v_i would have activated at an earlier time. Thus, for a node v_i that gets activated at time $t+1$, vaccinating or blocking nodes at time t will halt contagion propagation to v_i. This idea is used in the algorithm as follows. Consider the sets S_t and S_{t+1} of nodes that get infected or activated at times t and $t+1$, respectively. We identify nodes from S_t, one at a time, iteratively, where the node v_k that is removed from S_t has the most edges in the graph G to nodes that are still infected in S_{t+1}. Each time a v_k is removed from S_t, the "covering requirement" for each neighbor $v_j \in S_{t+1}$ is reduced by 1, and when v_js requirement is 0, by removal of one or more nodes from S_t, that means v_j can no longer be infected for contagion $\mathbb{C}_i$.

The algorithm for the MCICH is presented in Algorithm 1. The algorithm computes the set C of blocking nodes for contagion $\mathbb{C}_i$ for one iteration.

Summary of Analysis Process: The steps of the full analysis follow. Step 1: simulations are preformed without consideration of blocking nodes, as described above. Step 2: using the simulation outputs, blocking nodes are determined using the blocking heuristics and specified blocking node budget β_i for contagion $\mathbb{C}_i$. Step 3: the simulations are repeated, with all conditions the same as in Step 1, except that now the blocking nodes are added (these blocking nodes remain in state 0). Note that the simulation and blocking methods, models and codes can handle—as they currently exist—non-uniform thresholds across nodes, different thresholds per contagion for each node, and heterogeneities in other parameters. We are reporting uniform threshold results owing to space limitations and because it is important to understand baseline behaviors.

Simulation and Blocking Results: Unless otherwise stated, all results are averages over all 10 iterations of a simulation.

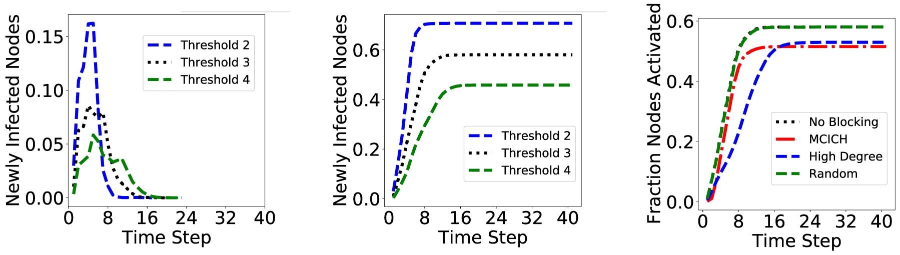
Basic simulation data and temporal blocking effects. Figure 3 provides three types of results for the FB-Politicians network. The first two plots show temporal data on the spread or propagation of both contagions $\mathbb{C}_1$ and $\mathbb{C}_2$ simultaneously *without* blocking. The third plot shows temporal effects of blocking nodes on the propagation of both contagions. Figure 3a shows the number of newly activated nodes at each time step. The curves rise as uniform threshold decreases from 4, to 3, to 2, since contagion propagates

Algorithm 1: Steps of the node blocking algorithm MCICH.

```
Input: Threshold $\theta = \theta(v,i)$ for contagion $\mathbb{C}_i$. A network $G(V,E)$. A set $I$ of initially
  activated nodes (at time $t = 0$). Budget $\beta_i$ on the number of blocking nodes for
  contagion $\mathbb{C}_i$. Maximum number $t_{max}$ of time steps to run simulation.
Output: The set $C$ of blocking nodes such that $|C_i| \leq \beta_i$ and such that the number of
  (newly) activated nodes is small.
Steps:
  (A) Run simulation of contagion propagation.
      (i) Compute the activated nodes at each time step from $t = 1$ through $t_{max}$.
      (ii) The output is a set $S_t$ of newly activated nodes at each time $t \in \{0,1,2,\ldots,t_{max}\}$,
           where $S_0 = A$.
  (B) Run the MCICH to obtain blocking node set $C$.
      (i) for $t = 1$ to $t_{max}$:
          (1) if $|S_t| \leq \beta_i$ then set $C = S_t$ and return $C$. Stop.
          (2) Initialize the candidate set of blocking nodes $T_t$ for this $t$ to $T_t = \emptyset$.
          (3) Set $Q_{t+1} = S_{t+1}$; $Q_{t+1}$'s elements will be removed iteratively.
          (4) for each $v_k \in S_{t+1}$, compute the number $\rho_k$ of neighbors that must be
              un-activated in order to prevent $v_k$ from being activated. Here, $\rho_k = n_k - \theta_k + 1$,
              where $n_k$ is the number of neighbors of $v_k$ in $G$ that are activated at any $t^*$,
              $0 \leq t^* < t$.
          (5) while $Q_{t+1}$ not empty and $|T_t| < \beta_i$ do:
              (a) for each $v_j \in S_t$, let $H_j$ be the subset of nodes in $S_{t+1}$ for which $v_j$ is a
                  neighbor in $G$.
              (b) Select the node $v_k$ such that $\max_k |H_k \cap Q_{t+1}|$. Break ties arbitrarily.
              (c) Add $v_k$ to $T_t$, the candidate set of blocking nodes.
              (d) For each node $v_j$ in $H_k$, reduce $\rho_j$ by 1. if $\rho_j = 0$ then remove $v_j$ from all
                  $H_k$ and remove $v_j$ from $Q_{t+1}$.
          (6) if $Q_{t+1}$ is empty then set $C = T_t$ and return $C$. Stop.
      (ii) No blocking set was found to completely stymie the contagion. Iterate through all
           $Q_{t+1}$ for all $t \in [1 \,..\, t_{max} - 1]$ and set $C = T_t$ for the smallest $|Q_{t+1}|$; if ties, choose the
           one at the earliest $t$. Return $C$. Stop.
```

more readily for lesser thresholds. Figure 3b shows the corresponding plots of *total* or cumulative number of nodes activated for both contagions as a function of time. Roughly 40% to 70% of FB-Politicians nodes are activated by $t_{max} = 24$, depending on θ. Figure 3c uses the $\theta = 3$ data from Figure 3b as a baseline, and shows three additional curves, one for each of the three blocking methods discussed above. These data show that for a blocking budget $\beta_i = 0.02$ fraction of nodes, the MCICH performs best (i.e., the curve is the lowest). For blocking contagions "lesser" (or "lower") is better. However, this budget is not sufficient to completely halt the contagion.

Efficacy and comparisons of all blocking methods across all networks. Figure 4 depicts the efficacy of all of the blocking methods for the three networks, for threshold values $\theta = 2$, 3, and 4. Data for one network are in a row, and data for one threshold are in one column. Each plot presents the cumulative fraction of activated nodes, as a function of the blocking budget in terms of fraction of network nodes. Note that the y-axis is



(a) $\theta = 2,3,4$, no blocking (b) $\theta = 2,3,4$, no blocking (c) $\theta = 3$ with blocking

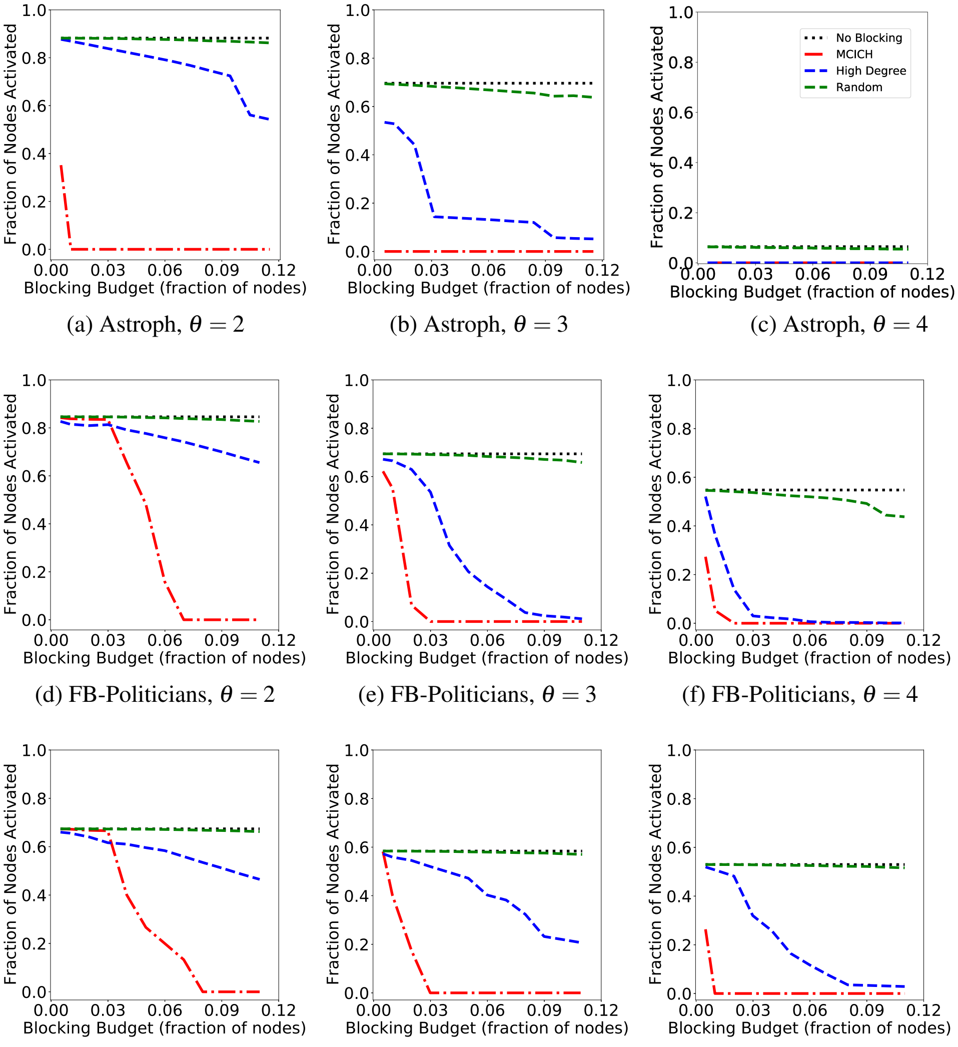
Fig. 3: Simulation results for the FB-Politicians network, where results are averages over 10 iterations. (a) shows time histories of the average number of *newly* activated nodes at each time step for contagions $\mathbb{C}_1$ and $\mathbb{C}_2$ combined, for three thresholds. (b) shows time histories of the average number of *cumulative* activated nodes at each time step for contagion $\mathbb{C}_1$ and $\mathbb{C}_2$ combined, for the same thresholds. (c) provides data for $\theta = 3$, for no blocking, and for each of the three blocking methods, where the blocking node budget $\beta_i = 0.02$ fraction of nodes. No method completely blocks the contagion (a greater budget is required), but MCICH performs best over the entire time history.

the total number of activations, so that, for example, if a node has contracted $\mathbb{C}_1$ and $\mathbb{C}_2$, then that counts as two activations. The cumulative fraction of activated nodes corresponds to the points at t_{max} in curves such as those presented in Figure 3c, for the respective blocking methods, thresholds, and networks. There is a "no blocking" curve, and three curves for each of the random blocking nodes heuristic, high degree blocking nodes heuristic, and MCICH in each plot. Since lower curves represent more effective blocking, it is clear that MCICH performs far better, in the great majority of cases, than do the random and high-degree blocking heuristics. The blocking budget β is currently allocated between the two contagions using proportion of nodes infected by contagions $\mathbb{C}_1$ and $\mathbb{C}_2$ when there is no blocking. For example, suppose n_1 and n_2 denote the number of newly infected nodes by $\mathbb{C}_1$ and $\mathbb{C}_2$ respectively, we use $n_1/(n_1+n_2)$ fraction of the budget for blocking $\mathbb{C}_1$ and the remaining budget for $\mathbb{C}_2$. If the algorithm needs less than the allocated budget for blocking $\mathbb{C}_1$, the remaining allocation is used to increase the budget for $\mathbb{C}_2$.

4 Future Research Directions

There are several directions for future work. For example, it is of interest to evaluate the MCICH heuristic under several other scenarios; examples include graphs with non-uniform threshold values for nodes, different ways of selecting seed sets and skewed distributions of seed nodes between the two contagions. It is also of interest to investigate the sensitivity of our heuristic with respect to the choice of seed sets. In our model, the two contagions are independent. It is of interest to investigate models where the contagions interact; that is, a node that is infected one contagion may make it easier or harder for the node to be infected by the other contagion.

Acknowledgments: We thank the reviewers for their comments. This work is partially supported by NSF Grants ACI-1443054 (DIBBS), IIS-1633028 (BIG DATA), CMMI-1745207 (EAGER), OAC-1916805 (CINES), CCF-1918656 (Expeditions), CRISP 2.0-1832587 and IIS-1908530.



(a) Astroph, $\theta = 2$

(b) Astroph, $\theta = 3$

(c) Astroph, $\theta = 4$

(d) FB-Politicians, $\theta = 2$

(e) FB-Politicians, $\theta = 3$

(f) FB-Politicians, $\theta = 4$

(g) Wiki, $\theta = 2$

(h) Wiki, $\theta = 3$

(i) Wiki, $\theta = 4$

Fig. 4: Simulation and blocking results of applying all three blocking methods to block two-contagion spreading in three networks, using different threshold values for contagion propagation for each network. Results for each network are in one row. From the top to bottom rows, the networks are: Astroph, FB-Politicians, and Wiki. Each column contains data one threshold: left to right, $\theta = 2$, 3, and 4. Each plot displays the fraction of nodes contracting either contagion in diffusion simulations, as a function of the fraction of nodes used as blocking nodes, employed to stop the contagions. In each plot, there are four curves. Contagion spreading without blocking (black dots) is the reference curve, and is a horizontal line. Results from the random selection of blocking nodes is the green curve. Results from selecting the highest degree nodes as blocking nodes is the blue dashed curve. Results from MCICH method is the red dash-dot curve. The lower the curve, the better the performance in blocking contagion. The MCICH method does significantly better in all cases.

References

1. Ahmed, N.K., Alo, R.A., Amelink, C.T., et al.: net.science: A cyberinfrastructure for sustained innovation in network science and engineering. In: Gateway (2020)
2. Barrett, C., Hunt III, H.B., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., Thakur, M.: Predecessor existence problems for finite discrete dynamical systems. Theoretical Computer Science 386(1), 3–37 (2007)
3. Beutel, A., Prakash, B.A., Rosenfeld, R., Faloutsos, C.: Interacting viruses in networks: Can both survive? In: KDD. pp. 426–434 (2012)
4. Centola, D., Macy, M.: Complex contagions and the weakness of long ties. American Journal of Sociology 113(3), 702–734 (2007)
5. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. TISSEC 10, 1–33 (2008)
6. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press (2010)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman & Co., San Francisco (1979)
8. González-Bailón, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. Scientific Reports 1, 7 pages (2011)
9. Granovetter, M.: Threshold models of collective behavior. American Journal of Sociology pp. 1420–1443 (1978)
10. Karrer, B., Newman, M.E.J.: Competing epidemics on complex networks. Physical Review E 84(3) (Sep 2011)
11. Kuhlman, C.J., Kumar, V.A., Marathe, M.V., Ravi, S., Rosenkrantz, D.J.: Inhibiting diffusion of complex contagions in social networks: Theoretical and experimental results. Data Mining and Knowledge Discovery 29(2), 423–465 (2015)
12. Kumar, P., Verma, P., Singh, A., Cherifi, H.: Choosing optimal seed nodes in competitive contagion. Frontiers in Big Data 2, 1–6 (June 2019)
13. Mortveit, H., Reidys, C.: An Introduction to Sequential Dynamical Systems. Springer Science & Business Media, New York, NY (2007)
14. Myers, S.A., Leskovec, J.: Clash of the contagions: Cooperation and competition in information diffusion. In: ICDM. pp. 539–548 (2012)
15. Newman, M.E.J., Ferrario, C.R.: Interacting epidemics and coinfection on contact networks. PLOS ONE 8, 1–8 (2013)
16. Romero, D., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In: WWW. pp. 695–704 (2011)
17. Schelling, T.C.: Micromotives and Macrobehavior. Norton, New York, NY (1978)
18. Stanoev, A., Trpevski, D., Kocarev, L.: Modeling the spread of multiple concurrent contagions on networks. PloS one 9(6), e95669 (2014)
19. Ugander, J., Backstrom, L., Marlow, C., Kleinberg, J.: Structural diversity in social contagion. Proceedings of the National Academy of Sciences 109(16), 5962–5966 (2012)
20. Vazirani, V.V.: Approximation Algorithms. Springer (2001)
21. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of the National Academy of Sciences 99, 5766–5771 (2002)