

Using Dominating Sets to Block Contagions in Social Networks

Robert Chen Bao
Biocomplexity Institute
University of Virginia
 Charlottesville, Virginia
 cb5th@virginia.edu

Matthew Hancock
Biocomplexity Institute
University of Virginia
 Charlottesville, Virginia
 mgh3x@virginia.edu

Chris J. Kuhlman
Biocomplexity Institute
University of Virginia
 Charlottesville, Virginia
 hugo3751@gmail.com

S. S. Ravi
Biocomplexity Institute
University of Virginia
 Charlottesville, Virginia
 ssravi0@gmail.com

Abstract—There are myriad real-life examples of contagion processes on human social networks, e.g., spread of viruses, information, and social unrest. Also, there are many methods to control or block contagion spread. In this work, we introduce a novel method of blocking contagions that uses nodes from dominating sets (DSs). To our knowledge, this is the first use of DS nodes to block contagions. Finding minimum dominating sets of graphs is an NP-Complete problem, so we generalize a well-known heuristic, enabling us to customize its execution. Our method produces a prioritized list of dominating nodes, which is, in turn, a prioritized list of blocking nodes. Thus, for a given network, we compute this list of blocking nodes and we use it to block contagions for all blocking node budgets, contagion seed sets, and parameter values of the contagion model. We report on computational experiments of the blocking efficacy of our approach using two mined networks. We also demonstrate the effectiveness of our approach by comparing blocking results with those from the high degree heuristic, which is a common standard in blocking studies.

Index Terms—contagion blocking, dominating sets, threshold models, social networks, simulation, high degree heuristic

I. INTRODUCTION

A. Background

A dominating set (DS) of a graph is a subset of nodes such that every node is in the subset, or has at least one distance-1 neighbor in the subset. Dominating sets (DSs) of a graph are used to control and monitor networks in a variety of situations [1], [2].

B. Motivation, Novelty, and Overview of Work

Many techniques have been devised to block the spread of contagions on social networks. Applications include blocking the spread of disinformation, thwarting public calls for protests, and intimidating individuals from taking action. See Section II.

In this work, we evaluate the ability of subsets of DSs to block contagions spreading on (social) networks. Although there has been much work on controlling and impeding the spread of contagions, a novelty of this work is that nodes of DSs have never been used for blocking contagions.

Our approach determines a DS and a prioritized list of blocking nodes for a specified network and a set of inputs to the DS heuristic. Thus, one set of blocking nodes is used for all contagion seed sets (i.e., nodes that are activated at

time $t = 0$) and all contagion model parameter values, for a graph. This is attractive because the blocking nodes need only be computed once since in practice there are costs associated with convincing individuals (represented by nodes of social networks) to block a contagion.

An overview of our procedures is as follows. Given a social network $G(V, E)$, where V is the node set and E is the edge set of G , a near-minimum dominating set (MDS), denoted S_{MD} , for G is computed by generalizing a well-known heuristic [3] for computing DSs. (No formally efficient algorithm for computing minimum dominating sets can exist unless $\mathbf{P} = \mathbf{NP}$.) The nodes of the MDS are ranked, in decreasing order, based on the number of nodes in the network that each node $v_i \in S_{MD}$ dominates. Given a budget b on the number of blocking nodes for a computation (including $b = 0$ for a baseline), the top b nodes of the ranked list are used as blocking nodes. These nodes are removed from G . Then, agent-based simulations (ABS) are run using a Granovetter-based threshold model [4]–[7] to determine the spread fraction (i.e., the fraction of nodes that are activated by contagion transmission) emanating from the seeded nodes.

C. Contributions

1. Use of dominating sets to block contagion spreading. DSs have not been used to block contagion propagation on social networks. In this work, we quantify the ability of DSs to reduce the spread of contagions. We generalize a heuristic [3] for computing approximate MDSs for any specified dominating distance k and coverage requirement h . (These terms are defined, with examples, in Section III.) We call this heuristic DSHkh and it is presented in Section IV-A. It outputs the number of nodes that each node dominates so that nodes can be ranked in decreasing order of the number of dominated nodes. This ranked list is then used to identify the top b nodes in a graph to specify as blocking nodes.

2. Quantification of blocking performance of DSHkh. We study the blocking performance of DSHkh by running contagion simulations on two networks, ten values of threshold (i.e., for simple and complex contagions), an aggressive contagion seeding method with six values of minimum numbers $n_{s,min}$ of seed nodes (for each $n_{s,min}$, 100 seed sets per graph are generated), six values of DSHkh’s coverage requirement h and

two values of its domination distance k , and roughly 12 values of the blocking budget b .

3. Comparisons of effectiveness of DSHkh against HDH.

A common standard in blocking contagions is the high degree heuristic HDH [8]–[13]. A key finding of our work is that a performance comparison between DSHkh and HDH is nuanced. In some cases, DSHkh performs better than HDH, and in other cases, the opposite is true.

II. RELATED WORK

Approaches for blocking contagions that propagate on social networks can be broken down into the following five classes. (i) Node removal to delete nodes that can contract a contagion and pass it on (e.g., [13]); (ii) edge removal to eliminate transmission pathways, e.g., [14]; (iii) community segregation: these remove nodes or edges on the boundaries of communities to stop the spread between communities (e.g., [15]); (iv) alterations of dynamics models, i.e., changing node/edge properties for how the contagion spreads (e.g., [16]); and (v) introducing a competing contagion to stifle the undesirable contagion, e.g., [17]). References specific to the HDH are given in Section I-C.

III. PRELIMINARIES

A **dominating set** (DS) of a graph G is a set S_D of nodes of G such that each node is either in S_D or has at least one distance-1 neighbor in S_D . This definition can be generalized to include the **domination distance** k —the maximum distance over which a node dominates other nodes—and the **coverage requirement** h —the minimum number of times a node must be dominated. In the definition above, $k = 1$ and $h = 1$. The typical goal with DSs is to find a S_D of G that has a minimum number of nodes, referred to as a **minimum dominating set** (MDS) and denoted S_{MD} . Figure 1 contains a graph and several MDSs, for different values of k and h . The cases $[k, h] = [2, 1]$, $[1, 2]$, and $[2, 2]$ each demonstrate that an MDS need not be unique.

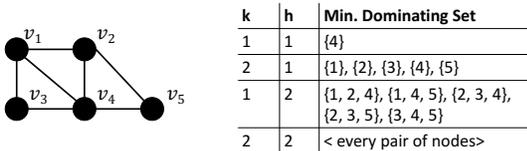


Fig. 1: A five-node connected graph and MDSs for different domination distances k and coverage requirements h . The table shows the subscript i for node v_i of the graph that is in a MDS.

In the **high degree heuristic** (HDH), nodes are ranked for blocking in decreasing order of their degrees, breaking ties arbitrarily.

IV. MODELS

A. Dominating Set Algorithm

Our algorithm for computing approximate MDSs is based on an extension of the heuristic in [3] for the $k = 1$, $h = 1$

case. Here, we generalize that heuristic for any $k \geq 1$ and $h \geq 1$. The algorithm has the following key steps. First, for each node $v_i \in V$, determine the set of all of its neighbors within distance k , including v_i itself. Second, specify a counter for each node that stores the number of times it has been dominated; initialize this counter for each node to zero. Also, initialize the node set S_{MD} to the empty set. The third step is a *while* loop on the condition that at least one node’s coverage requirement is not met and at least one node can be added to S_{MD} . In this loop, identify the node v_j with the greatest number c_j of neighbors within distance k of v_j whose counters are less than h , and where $v_j \notin S_{MD}$ (break ties among multiple v_j arbitrarily). Write this pair (v_j, c_j) to the output file of prioritized dominating nodes and add v_j to S_{MD} . For each of the c_j nodes, increment their counters by 1 and remove from further consideration all neighbors of all nodes (from step 1) whose counter equals h .

B. Graph Dynamical System (GDS)

We model the propagation of social contagions over a social network using discrete dynamical systems. We begin with the necessary definitions from [18], [19]. Let \mathbb{B} denote the Boolean domain $\{0, 1\}$. A **Synchronous Dynamical System** (SyDS) \mathcal{S} over \mathbb{B} is specified as a pair $\mathcal{S} = (G, \mathcal{F})$, where (i) $G(V, E)$ is an undirected graph with $n = |V|$ nodes, representing the underlying social network over which the contagion propagates through the node-pairs of interactions represented by the edge set E , and (ii) $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$ is a collection of functions, with f_i denoting the local transition function associated with node v_i , $1 \leq i \leq n$. Each function f_i specifies the local interaction between node v_i and its neighbors in G . Regarding these functions, we note that each $v_i \in V$ has a state value from \mathbb{B} . Nodes in state 0 (respectively, 1) are said to be **unactivated** (respectively, **activated**). Thus, in the case of information flow, for example, an activated node has received the information and will pass it on. It is assumed that once a node reaches state 1, it cannot return to state 0. We refer to a SyDS with this property as a “progressive system” [20].

We formally describe the **local transition functions**, f_i . The inputs to f_i are the state of v_i and those of the neighbors of $v_i \in V$. Function f_i maps each combination of inputs to a value in \mathbb{B} , which is the next state $s_i \in \mathbb{B}$ for v_i . For the propagation of contagions in social networks, it is natural to model each function f_i , ($1 \leq i \leq n$) as a θ_i -threshold function [4], [5] for an appropriate nonnegative integer θ_i . Such a threshold function (taking into account the progressive nature of the dynamical system) is defined as follows:

- If the state of v_i is 1, then the value of f_i is 1, regardless of the values of the other inputs to f_i .
- If the state of v_i is 0, then the value of f_i is 1 if at least θ_i of the inputs are 1; otherwise, the value of f_i is 0.

In a SyDS, at each time step, all the nodes compute and update their states *synchronously*. A **configuration** $s(t)$ of a SyDS at any time t is an n -vector (s_1, s_2, \dots, s_n) . A series of configurations, from $t = 0$ to some integer $t_{max} \geq 0$ is called a **forward trajectory**. In the simulations of the next

section, forward trajectories are computed on social networks, with and without blocking nodes.

V. COMPUTATIONAL EXPERIMENTS AND RESULTS

A. Networks Used in Simulations

Networks evaluated in this work are listed in Table I. The networks are from the SNAP data set [21].

B. Agent-Based Simulation (ABS) and Blocking Processes

Figure 2 summarizes the procedures in evaluating the DSHkh blocking method, and comparing it against the HDH. The pipeline in Figure 2 is executed per network, and each box represents a (multi-nested) loop structure to evaluate multiple parameters and/or parameter values. For each pair $[k, h]$, DSHkh is executed on a particular network. From the output, nodes are ranked in decreasing order of the number of nodes they cover. One hundred seed sets, for a given $n_{s,min}$, are generated using a modified Centola [23] seeding method. In the standard seeding method, a node is selected uniformly at random and it and all of its distance-1 neighbors (e.g., inducing a star subgraph) are selected as seed (i.e., activated) nodes at $t = 0$ in a simulation instance. Here, we extend the method such that if this number of nodes is less than the specified $n_{s,min}$, then one of the leaf nodes of the star is selected uniformly at random and all of its distance-1 neighbors are added to the seed node set. This continues until the number of seed nodes $n_s \geq n_{s,min}$. This makes the seeding process more onerous to block because the seed nodes induce a connected subgraph, which more readily spread a complex contagion [23] than does dispersing the seed nodes, e.g., by choosing them uniformly at random, particularly for smaller $n_{s,min}$. For each set of seed nodes in a simulation, for each $[k, h]$ pair, and for each blocking budget b , the blocking nodes are selected as the b nodes with the greatest coverage for DSHkh (and with the greatest degrees for HDH) subject to the constraint that no blocking node is a seed node. We are now ready to discuss agent-based simulations (ABS).

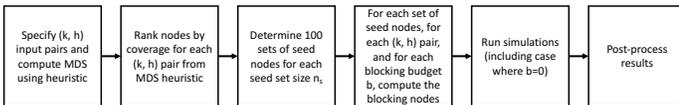


Fig. 2: Steps in numerical experiments. Each box represents (multiple) loops over parameters. This pipeline is for one network and for the DSHkh method. A similar set of steps is executed for the HDH.

A **simulation** is composed of 100 runs. A **run** is one contagion spread instance, from time $t = 0$ through $t_{max} = 100$ days. Synonyms for *run* are *iteration* and *simulation instance*. The following applies to one run. Each node v_i is assigned a threshold θ_i . The seed nodes are set to the activated state 1; all other nodes are set to state 0. The blocking nodes are removed from the network G (in actuality, their thresholds are set very high, e.g., $\theta_i = n$ so all blocking nodes remain in state 0).

Time is incremented and state changes for nodes are computed and recorded.

C. Agent-Based Simulation Blocking Results

The computations and results are such that the *lower* the curves are on plots, the *better* or *more effective* is the blocking method: i.e., lesser is better.

Baseline data. Figure 3 provides time-history results for the Enron graph. Each curve shows the fraction of activated nodes (“Frac. Spread”) as a function of time. There are four pairs of curves, corresponding to four values of threshold θ ; see legend. For each pair, one curve is for the case with no blocking nodes (i.e., $b = 0$) while the other is for the case with $b = 1000$. The blocking (solid) curve is below the non-blocking (dashed) curve. Other parameters are given in the figure caption. Because the threshold model is progressive, each curve is non-decreasing in time. Curves are higher for lesser thresholds, denoting that it is easier to propagate low-threshold contagions.

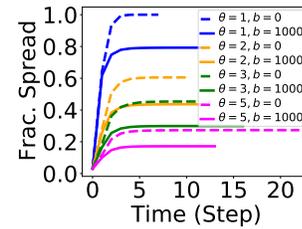


Fig. 3: Time-histories of cumulative fraction of activated nodes in ABSs for the Enron network. The minimum number of seed nodes is $n_{s,min} = 1000$. The blocking results ($b = 1000$, solid curves) are for the DSHkh algorithm with $k = 1$, $h = 8$, for comparisons with no-blocking baselines (dashed curves) of the same color.

In the plots below, we show the final (i.e., cumulative) fraction of activated nodes at $t = t_{max}$; i.e., each curve in Figure 3 gets represented as a single data point.

Effect of thresholds. Figure 4 shows the effect of node threshold on the final fraction of nodes in state 1 for the Enron network with $n_{s,min} = 10$ seed nodes. In Figure 4a, all of the curves lie on top of each other because $b = 10$ blocking nodes is too small to affect the contagion. Consistent with observations of social networks, threshold increases generally decrease contagion spread sizes. These data also serve as a baseline contagion spread size for the results in Figure 4b where the number of blocking nodes increases markedly to $b = 5000$. The same qualitative effect of threshold is observed, but now the effects of $[k, h]$ values in the DSHkh algorithm are apparent. First, the upper two curves (i.e., the least effective blocking) in Figure 4b occur for $k = 3$, compared to the lower three curves for $k = 1$. Second, for a given k , curves are lesser for increasing h .

Performance comparisons between DSHkh and HDH and comparisons of blocking budget b . We examine the performance of both blocking methods as a function of the number b of blocking nodes. For the DSHkh method, data are for $k = 1$

TABLE I: Mined networks and selected properties; properties determined with [22].

Network	Type	Num. Nodes	Num. Edges	Ave. Deg.	Max. Deg.	Max Kcore.	Ave. Clust. Coeff.	Diameter
AstroPh	collaboration	17903	196972	22.00	504	56	0.63	14
Enron	Email	33696	180811	10.73	1383	43	0.51	13

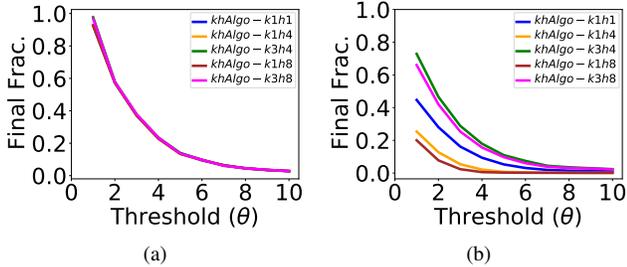


Fig. 4: Effect of node threshold θ_i on the final fraction of nodes in state 1. Enron plot for $n_{s,min} = 10$ and (a) $b = 10$, and (b) $b = 5000$. Lower curves mean more effective blocking.

and $5 \leq h \leq 8$. All curves in Figure 5 show a general decrease in the final fraction of nodes that reach state 1 as b increases.

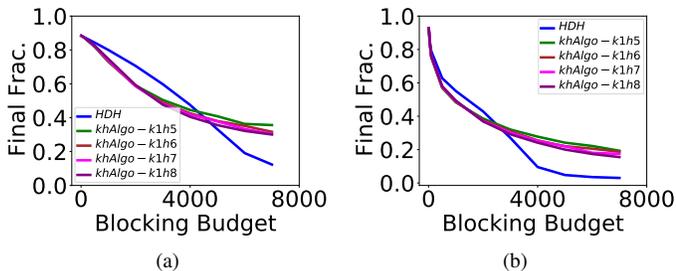


Fig. 5: Blocking performance of both methods as a function of blocking budget b . The DSHkh method uses $k = 1$ and $h = 5$ to 8. (a) AstroPh, $n_{s,min} = 1000$, $\theta = 2$. (b) Enron, $n_{s,min} = 10$, $\theta = 1$.

Figure 5 also shows comparisons between DSHkh and HDH for both networks. We choose particular plots to illustrate that each method can be superior (i.e., produce a lower curve) over ranges of parameters (here, over ranges in b). DSHkh in these plots is more effective for smaller b , but less so for larger b .

VI. SUMMARY

This is the first study to quantify the ability of dominating sets (DSs) to block contagions in social networks. We evaluate our DSHkh method for a range of parameters including different networks, threshold values, blocking budgets, and parameters in the DSHkh algorithm. We compare DSHkh to the common standard high degree heuristic (HDH) for blocking. Our contributions are given in Section I-C. Future work includes evaluating other DS algorithms (e.g., those designed to run faster).

ACKNOWLEDGMENT

This work is partially supported by University of Virginia Strategic Investment Fund Award Number SIF160, and NSF Grants: OAC-1916805 (CINES), CCF-1918656 (Expeditions), and CRISP 2.0-1832587.

REFERENCES

- [1] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*. Marcel Dekker, 1998.
- [2] J. C. Nacher and T. Akutsu, “Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control,” *New Journal of Physics*, vol. 14, 2012.
- [3] S. Eubank, V. S. A. Kumar, M. Marathe, A. Srivivasan, and N. Wang, “Structural and algorithmic aspects of massive social networks,” in *ACM-SIAM Symposium on Discrete Algorithms*, 2004, pp. 718–727.
- [4] M. Granovetter, “Threshold models of collective behavior,” *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.
- [5] T. Schelling, *Micromotives and Macrobehavior*, 1978.
- [6] P. Dodds and D. Watts, “A generalized model of social and biological contagion,” *Journal of Theoretical Biology*, vol. 232, no. 4, 2005.
- [7] P. Dreyer and F. Roberts, “Irreversible k -Threshold Processes: Graph-Theoretical Threshold Models of the Spread of Disease and Opinion,” *Discrete Applied Mathematics*, vol. 157, pp. 1615–1627, 2009.
- [8] R. Albert, H. Jeong, and A. Barabasi, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, pp. 378–382, 2000.
- [9] Z. Dezso and A. Barabasi, “Halting viruses in scale-free networks,” *Physical Review E*, vol. 65, pp. 055 103–1–055 103–4, 2002.
- [10] R. Cohen, S. Havlin, and D. Ben-Avraham, “Efficient immunization strategies for computer networks and populations,” *Physical Review Letters*, vol. 91, pp. 247 901–1–247 901–5, 2003.
- [11] D. Centola, “Failure in complex social networks,” *Journal of Mathematical Sociology*, vol. 33, pp. 64–68, 2009.
- [12] D. Siegel, “When does repression work?: Collective action and social networks,” *Journal of Politics*, vol. 73, pp. 993–1010, 2011.
- [13] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, “Gelling, and melting, large graphs by edge manipulation,” in *CIKM*, 2012, pp. 245–254.
- [14] C. J. Kuhlman, G. Tuli, M. V. Marathe, S. S. Ravi, and S. Swarup, “Blocking simple and complex contagion by edge removal,” in *ICDM*, 2013.
- [15] Y. Zhang, A. Adiga, S. Saha, A. Vullikanti, and B. A. Prakash, “Near-optimal algorithms for controlling propagation at group scale on networks,” *IEEE Trans. Knowl. Data Eng.*, pp. 3339–3352, 2016.
- [16] A. Beutel, B. A. Prakash, R. Rosenfeld, and C. Faloutsos, “Interacting viruses in networks: can both survive?” in *KDD*, 2012, pp. 426–434.
- [17] B. A. Prakash, A. Beutel, R. Rosenfeld, and C. Faloutsos, “Winner takes all: Competing viruses or ideas on fair-play networks,” in *WWW*, 2012, pp. 1037–1046.
- [18] H. Mortveit and C. Reidys, *An Introduction to Sequential Dynamical Systems*. New York, NY: Springer, 2007.
- [19] A. Adiga *et al.*, “Graphical dynamical systems and their applications to bio-social systems,” *International Journal of Advances in Engineering Sciences and Applied Mathematics*, pp. 1–19, 2018.
- [20] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the Spread of Influence Through a Social Network,” in *KDD*, 2003, pp. 137–146.
- [21] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, 2014.
- [22] N. K. Ahmed *et al.*, “net.science: A Cyberinfrastructure for Sustained Innovation in Network Science and Engineering,” in *Gateway Conference*, 2020, pp. 71–74.
- [23] D. Centola and M. Macy, “Complex contagions and the weakness of long ties,” *American Journal of Sociology*, vol. 113, pp. 702–734, 2007.